A. Kaveh

Applications of Metaheuristic Optimization Algorithms in Civil Engineering

Springer

Applications of Metaheuristic Optimization Algorithms in Civil Engineering

A. Kaveh

Applications of Metaheuristic Optimization Algorithms in Civil Engineering

Springer

A. Kaveh
Iran University of Science and Technology
Centre of Excellence for Fundamental Studies
in Structural Engineering
Tehran, Iran

ISBN 978-3-319-48011-4 ISBN 978-3-319-48012-1 (eBook)
DOI 10.1007/978-3-319-48012-1

Library of Congress Control Number: 2016959614

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Recent advances in civil engineering technology require greater accuracy, efficiency, and speed in the analysis and design of the corresponding systems. It is therefore not surprising that new methods have been developed for optimal analysis and design of real-life systems and models with complex configurations and a large number of elements.

This book can be considered as an application of metaheuristic algorithms to some important optimization problems in civil engineering. This book is addressed to those scientists and engineers, and their students, who wish to explore the potential of newly developed metaheuristics by some practical problems. The concepts presented in this book are not only applicable to civil engineering problems but can equally be used for optimizing the problems involved in mechanical and electrical engineering.

The author and his graduate students have been involved in various developments and applications of various metaheuristic algorithms to structural optimization in the last two decades. This book contains part of this research suitable for various aspects of optimization in civil engineering.

The book is likely to be of interest to civil, mechanical, and electrical engineers who use optimization methods for design, as well as to those students and researchers in structural optimization who will find it to be necessary professional reading.

In Chap. 1, a short introduction is provided for the goals and contents of this book. Chapter 2 discusses optimum design of laterally supported castellated beams using tug-of-war optimization algorithm. Chapter 3 provides optimum design of multi-span composite box girder bridges using the well-known cuckoo search algorithm. In Chap. 4, the sizing optimization of skeletal structures using the recently developed enhanced whale optimization algorithm is presented. Examples are chosen from both trusses and frame structures. Chapter 5 contains the size and geometry optimization of double-layer grids from the family of space structures using the colliding bodies optimization (CBO) and Enhances colliding bodies optimization (ECBO) algorithms. Chapter 6 presents the sizing and geometry optimization of different mechanical system of domes via the ECBO algorithm. Special domes are discussed in the chapter. Chapter 7 presents improved magnetic charged system search method for optimal design of single-layer barrel vault. Chapter 8 contains optimal design of double-layer barrel vaults using the CBO and ECBO algorithms. In Chap. 9, optimum design of steel floor system is performed using ECBO. In Chap. 10, optimal design of the monopole structures is performed using the CBO and ECBO algorithms. Chapter 11 deals with damage detection in skeletal structures based on the charged system search (CSS) optimization using incomplete modal data. Such a study is an important issue in earthquake engineering. In Chap. 12, modification of the ground motions is performed utilizing the ECBO. In Chap. 13, a combinatorial optimization is considered and the bandwidth, profile, and wavefront of sparse matrices are optimized using four metaheuristic algorithms consisting of the PSO, CBO, ECBO, and TWO. In Chap. 14, optimal analysis and design of large-scale domes with frequency constraints is presented. Here, the importance of using optimal analysis in optimal design of structures for large-scale domes is illustrated. In Chap. 15, an accurate and efficient technique, so-called multi-DVC cascade optimization, is presented for optimal design of 3D truss towers with a large number of design variables to illustrate its applicability to optimum design of practical structures. Chapter 16 utilizes the vibrating particles system algorithm for truss optimization with frequency constraints. Five examples are used for evaluating this algorithm. In Chap. 17, the cost and CO_2 emission optimization of reinforced concrete frames is performed employing the ECBO algorithm. Nowadays, this is an important environmental issue in civil engineering. Chapter 18 presents a study of the construction site layout planning problem using the CBO and ECBO algorithms. This chapter shows the use of optimization methods in construction management.

I would like to take this opportunity to acknowledge a deep sense of gratitude to a number of colleagues and friends who in different ways have helped in the preparation of this book. My special thanks are due to Ms. Silvia Schilgerius, the senior editor of the Applied Sciences of Springer, for her constructive comments, editing, and unfailing kindness in the course of the preparation of this book. My sincere appreciation is extended to our Springer colleagues, in particular Mr. R.R. Pavan Kumar, the project manager, who prepared the careful page design of this book.

I would like to thank my former and present Ph.D. and M.Sc. students, Dr. A. Zolghadr, Dr. T. Bakhshpoori, Mr. V.R. Mahdavi, Mr. M. Ilchi Ghazaan, Mr. M.H. Ghafari, Mrs. Sh Bijari, Mr. M. Alipour, Mr. F. Shokohi. Mr. M. Kamalinejad, Mr. M. Maniat, Mr. M. Moradveisi, Mr. M. Rezaei, and Mr. M. Rasteghar Moghaddam, for using our joint papers and for their help in various stages of writing this book. I would like to thank the publishers who permitted some of our papers to be utilized in the preparation of this book, consisting of Springer, John Wiley and Sons, and Elsevier.

My warmest gratitude is due to my son, Mr. B. Kaveh, for his help in editing the first chapter, and my wife, Mrs. L. Kaveh, for her continued support in the course of preparing this book.

Every effort has been made to render the book error free. However, the author would appreciate any remaining errors being brought to his attention through his e-mail address: alikaveh@iust.ac.ir.

Tehran, Iran A. Kaveh
January 2017

Contents

Chapter 1
Introduction

1.1 Metaheuristic Algorithms for Optimization

Much has been made of the parallels between engineering and art, and yet a unique economy of parts and adherence to a plethora of constraints from cost to market trends, from maintainability to robustness, and from project schedules safely distinguish engineering design from the arts and engineering projects from artworks. At the heart of this distinction lies the concept of "optimization" – the science of choosing design variable values within given constraints such that a function, e.g., total system cost is minimized, or overall system reliability is maximized.

While the last three decades has seen an explosion in new methodologies applied to the problem of optimization, there is also evidence for a resurgence of improved classical algorithms and a growing number of engineering problems where heuristic and algorithmic optimization has overtaken and, in some cases, replaced the engineering graybeards and rule-of-thumb optimization methods.

Some of the most commonly used classical algorithmic optimization techniques were gradient based and allowed a search of the solution space near a given parameter point where gradient information about the target function was available [1, 2]. Gradient-based methods, in general, converge faster and can obtain solutions of higher accuracy than more modern stochastic approaches. However, the acquisition of gradient information for the target function can be either costly or even impossible. Moreover, these types of algorithms are only guaranteed to converge to local minima. Furthermore, a good starting point can be vital for the successful execution of these methods. In many optimization problems, prohibited zones, side limits, and non-smooth or non-convex functions need to be taken into consideration, increasing the difficulty of obtaining optimal solutions.

There is a slew of more recently developed optimization methods, known as metaheuristic algorithms, that are not restricted in the aforementioned manner. These methods are suitable for global searches over the entire search space due to

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_1

their capability of exploring and finding promising regions in the search space with reasonable computational effort. Ultimately, metaheuristic algorithms tend to perform rather well for most optimization problems [3, 4]. This is because these methods refrain from simplifying or making assumptions about the original problem. Evidence of this can be seen in their successful application to a vast variety of fields, such as engineering, physics, chemistry, arts, economics, marketing, genetics, operations research, robotics, social sciences, and politics.

The word *heuristic* has its origin in the old Greek work *heuriskein*, which means the art of discovering new strategies or rules to solve problems. The suffix *meta*, also a Greek prefix, has come to mean a higher level of abstraction in the English language. The term *metaheuristic* was introduced by Glover in the paper [5] and denotes a strategy of solving a problem using higher levels of abstractions and to guide a heuristic search of the solution space.

A heuristic method can be considered as a procedure that is likely to discover a very good feasible solution, but not necessarily an optimal solution, for a considered specific problem. In most cases no guarantee is provided for the quality of the solution obtained, but a well-designed heuristic method usually can provide a solution that is nearly optimal. The procedure also should be sufficiently efficient to deal with very large problems. Heuristic methods are often *iterative algorithms*, where each iteration involves conducting a search for a new solution that might be better than the best solution found in a previous iteration. After a reasonable amount of time when the algorithm is terminated, the solution it provides is the best one found during all iterations. A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring (global search) and exploiting (local search) the search space in order to efficiently find near-optimal solutions [6]. Learning strategies can be employed to add the "intelligence" to such guided search heuristics.

Metaheuristic algorithms have found many applications in different areas of applied mathematics, engineering, medicine, economics, and other sciences. Within engineering, these methods are extensively utilized in the design stages of civil, mechanical, electrical, and industrial projects.

1.2 Optimization in Civil Engineering and Goals of the Present Book

In the area of civil engineering that is the main concern of this book, one tries to achieve certain objectives in order to optimize weight, construction cost, geometry, layout, topology, construction time, and computational time satisfying certain constraints. Since resources, fund, and time are always limited, one has to find solutions to optimize the usage of these resources.

The main goal of this book is to apply some well established and most recently developed metaheuristic algorithms to optimization problems in the field of civil engineering, as detailed in the subsequent section. The subjects considered in this book are structural design of various types of structures such as trusses, frames, space structures, castellated beams, floor system, monopole structures, and multi-span composite box girder bridges. From earthquake engineering, modification of ground motions and damage detection in skeletal structures are studied. For optimal analysis, bandwidth, profile, and wavefront optimization is performed using different metaheuristic algorithms. From optimization with frequency constraints, large-scale domes are studied using optimal analysis. For optimal design of large-scale three-dimensional truss structures, an accurate and efficient technique, so-called multi-DVC cascade optimization, is presented, and examples with large number of design variables are investigated to illustrate the applicability of the presented method for optimum design of practical structures. From concrete structures the objective function of algorithm is considered as the construction material costs of reinforced concrete structural elements and carbon dioxide (CO_2) emissions through different phases of a building life cycle. From construction management, the construction of site layout planning problem is presented.

1.3 Organization of the Present Book

The remaining chapters of this book are organized in the following manner:

Chapter 2 introduces the recently developed metaheuristic so-called tug of war optimization and applies this method to the *optimal design of castellated beams*. Two common types of laterally supported castellated beams are considered as design problems: beams with hexagonal openings and beams with circular openings. In this chapter, castellated beams have been studied for two cases: beams without filled holes and beams with end-filled holes. Here, tug of war optimization algorithm is utilized for obtaining the solution of these design problems. For this purpose, the cost is taken as the objective function, and some benchmark problems are solved from literature [7].

Chapter 3 presents an integrated metaheuristic-based optimization procedure for discrete size *optimization of straight multi-span steel box girders* with the objective of minimizing the self-weight of girder. The selected metaheuristic algorithm is the *cuckoo search (CS) algorithm*. The optimum design of a box girder is characterized by geometry, serviceability, and ultimate limit states specified by the American Association of State Highway and Transportation Officials (AASHTO). Size optimization of a practical design example investigates the efficiency of this optimization approach and leads to around 15 % of saving in material (Kaveh et al. [8]).

Chapter 4 addresses a new nature-inspired metaheuristic optimization algorithm, called *whale optimization algorithm* (WOA), and utilizes this algorithm for size *optimization of skeletal structures*. This method is inspired by the bubble-net hunting strategy of humpback whales. WOA simulates hunting behavior with random or the best search agent to chase the prey and the use of a spiral to simulate bubble-net attacking mechanism of humpback whales. In this chapter, EWOA is also compared with WOA and other metaheuristic methods developed in the literature using four skeletal structure optimization problems. Numerical results compare the efficiency of the WOA and EWOA with the latter algorithm being superior to the standard implementation [9].

Chapter 5 applies the optimum design procedure, based on *colliding bodies optimization* (CBO) method and its *enhanced* version (ECBO), to optimal design of two commonly used configurations of *double-layer grids*, and optimum span–depth ratios are determined. Two ranges of spans as small and large sizes with certain bays of equal lengths in two directions and different types of element grouping are considered for each type of square grids. These algorithms obtain minimum weight grid through appropriate selection of tube sections available in AISC load and resistance factor design (LRFD). The comparison is aimed in finding the depth at which each of different configurations shows its advantages. Finally, the effect of support locations on the weight of the double-layer grids is investigated [10].

Chapter 6 introduces a finite element model based on geometrical nonlinear analysis of different mechanical systems of large-scale domes consisting of *double-layer domes*, *suspen-domes*, and *single-layer domes* with rigid connections. The suspen-dome system is a new structural form that has become a popular structure in the construction of long-span roof structures. Suspen-dome is a kind of new prestressed space grid structure which is a spatial prestressed structure and has complex mechanical characteristics. In this chapter, an optimum geometry and sizing design is performed using the *enhanced colliding bodies optimization* algorithm. The length of the strut, the cable initial strain, the cross-sectional area of the cables, the cross-sectional size of steel elements, and the height of dome are adopted as design variables for domes, and the minimum volume of each dome is taken as the objective function. A simple approach is defined to determine the configurations of the dome structures. The design algorithm obtains minimum volume domes through appropriate selection of tube sections available in AISC load and resistance factor design (LRFD). This chapter explores the efficiency of Lamella suspen-dome with pin-jointed and rigid-jointed connections and compares them with single-layer Lamella dome and double-layer Lamella dome [11].

Chapter 7 optimizes two *single-layer barrel vault frames* with different patterns via the *improved magnetic charged system search* (IMCSS). In the process of optimization, contrary to size variables, shape is a continuous variable. In the case of shape optimization of this type of space structures, since all of the nodal coordinates as the shape variables are dependent on the height-to-span ratio of the barrel vault, height is considered as the only shape variable in a constant span of barrel vault. In comparison, the best height-to-span ratios of barrel vaults under static loading conditions obtained from CSS, MCSS, and IMCSS algorithms are approximately close to the value of 0.17 from a comparative study carried out by Parke. Furthermore, as seen from the results, different patterns of barrel vaults have different effects on the value of the best height-to-span ratio. Moreover, in comparison to CSS and MCSS algorithms, IMCSS found better values for the weight of the structures with a lower number of analyses [12].

Chapter 8 implements the recently developed metaheuristic algorithms *colliding bodies optimization* (CBO) and its *enhanced* version (ECBO) for the optimization of *double-layer barrel vaults*. Two kinds of double-layer barrel vaults are optimized considering the weight of the structure as the objective function, where the design constraints are imposed according to the provisions of AISC-ASD. The numerical results show the successful performance of the CBO and ECBO algorithms in large-scale structural optimization problems such as double-layer barrel vaults [10].

Chapter 9 considers a *steel floor system* consisting of decks, interior beams, edge beams, and girders. Optimal design of a deck without considering beam optimization is simple. However, a deck with a higher cost may increase the composite action of the beams and decrease the beam cost, thus reducing the total expense. Also different number of floor divisions can improve the total floor cost. Increasing beam capacity by using castellated beams is another efficient method to save the costs. In this study, floor optimization is performed and these three issues are discussed. Floor division number and deck sections are some of the variables. For each beam, profile section of the beam, beam-cutting depth, cutting angle, spacing between holes, and number of filled holes at the ends of castellated beams are other variables. The objective function is the total cost of the floor, consisting of the steel profile, cutting and welding, concrete, steel deck, shear stud, and construction costs. Optimization is performed by *enhanced colliding bodies optimization* (ECBO). Results show that using castellated beams, selecting a deck with higher price and considering different number of floor divisions can decrease the total cost of a floor [13].

Chapter 10 studies a *tubular steel monopole structure* widely used for supporting antennas in telecommunication industries. This chapter utilizes the two recently developed metaheuristic algorithms, so-called colliding bodies optimization (CBO) and enhanced colliding bodies optimization (ECBO), for size optimization of monopole steel structures. The design procedure aims to obtain minimum weight of monopole structures subjected to the TIA-EIA222F specification. Two monopole structure examples are examined to verify the suitability of the design procedure and to demonstrate the effectiveness and robustness of the CBO and ECBO in creating optimal design for this problem. The outcomes of the ECBO are also compared to those of the standard CBO to illustrate the importance of the enhancement [14].

Chapter 11 studies the *damage detection in structures* by alteration in the dynamic behavior of the structures. Observation of these changes has often been viewed as a means to identify and assess the location and severity of damages in structures. Among the responses of a structure, natural frequencies and natural modes are both relatively easy to obtain and independent from external excitation and, therefore, can be used as a measure of the structural behavior before and after an extreme event which might have led to damage in the structure. This chapter applies the *charged system search* algorithm to the problem of damage detection using vibration data. The objective is to identify the location and extent of multi-damage in a structure. Both natural frequencies and mode shapes are used to form the required objective function. To moderate the effect of noise on measured data, a penalty approach is applied. Numerical examples consisting of beams, frames, and trusses are examined. The results show that the present methodology can reliably identify damage scenarios using noisy measurements and incomplete data [15].

Chapter 12 presents a simple and robust approach for *spectral matching of ground motions* utilizing the wavelet transform and an improved metaheuristic optimization technique. For this purpose, wavelet transform is used to decompose the original ground motions to several levels, where each level covers a special range of frequency and then each level is multiplied by a variable. Subsequently, the *enhanced colliding bodies optimization* technique is employed to calculate the variables such that the error between the response and target spectra is minimized. The application of the proposed method is illustrated through modifying 12 sets of ground motions [16].

Chapter 13 employs three recently developed metaheuristic optimization algorithms, known as *colliding bodies optimization* (CBO), *enhanced colliding bodies optimization* (ECBO), and *tug of war optimization* (TWO), for *optimum nodal ordering* to reduce bandwidth, profile, and wavefront of sparse matrices. The bandwidth, profile, and wavefront of some graph matrices, which have equivalent pattern to structural matrices, are minimized using these methods. Comparison of the achieved results with those of some existing approaches shows the robustness of the utilized algorithms for bandwidth, profile, and wavefront optimization [17].

Chapter 14 involves the structural *optimization of domes* with a large number of structural analyses using the *democratic particle swarm optimization*. When optimizing large structures, these analyses require a considerable amount of computational time and effort. However, there are specific types of structure for which the results of the analysis can be achieved in a much simpler and quicker way due to their special repetitive patterns. In this chapter, frequency constraint optimization of cyclically repeated space trusses is considered. An efficient technique is used to decompose the large initial eigenproblem into several smaller ones, thus decreasing the required computational time significantly. Some examples are presented in order to illustrate the efficiency of the presented method [18].

Chapter 15 performs optimum design of real-world structures with high number of design variables, large size of the search space, and control of a great number of design constraints in a reasonable time. This chapter presents an accurate and efficient technique, so-called multi-DVC cascade optimization, for optimal design of *three-dimensional truss towers* with large number of design variables to illustrate its applicability to optimum design of practical structures [19].

Chapter 16 includes application of the recently developed physically inspired non-gradient algorithm for structural optimization with frequency constraints. The algorithm being called vibrating particles system (VPS) mimics the free vibration of single degree of freedom systems with viscous damping. Truss optimization with frequency constraints is believed to represent nonlinear and non-convex search spaces with several local optima and therefore is suitable for examining the capabilities of the new algorithms. A set of five truss design problems are considered for evaluating the VPS in this article. The numerical results demonstrate the efficiency and robustness of the new method (Kaveh and Ilchi Ghazaan [20]).

Chapter 17 investigates discrete design *optimization of reinforcement concrete frames* using the recently developed metaheuristic called *enhanced colliding bodies optimization* (ECBO) and the *non-dominated sorting enhanced colliding bodies optimization* (NSECBO) algorithm. The objective function of algorithms consists of construction material costs of reinforced concrete structural elements and carbon dioxide (CO_2) emissions through different phases of a building life cycle that meets the standards and requirements of the American Concrete Institute's building code. The proposed method uses predetermined section database (DB) for design variables that are taken as the area of steel and the geometry of cross sections of beams and columns. The use of ECBO algorithm for designing reinforced concrete frames indicates an improvement in the computational efficiency over the designs performed by Big Bang–Big Crunch (BB–BC) algorithm. The analysis also reveals that the two objective functions are quite relevant, and designs focused on mitigating CO_2 emissions could be achieved at an acceptable cost increment in practice [21].

Chapter 18 employs two newly developed metaheuristic algorithms called *colliding bodies optimization* and *enhanced colliding bodies optimization* to solve construction *site layout planning problem*. Results show that both of these algorithms have the capability of solving this kind of problem. Two case studies are presented to illustrate the applicability and performance of the utilized methods [22].

Finally, it should be mentioned that most of the metaheuristic algorithms are attractive, because each one has its own striking features. However, the one which is simple, less parameter dependent, and easy to implement, has a good balance between exploration (diversification) and exploitation (intensification), has higher capability to avoid being trapped in local optima and higher accuracy, is applicable to wider types of problems, and can deal with higher number of variables can be considered as the most attractive for engineering usage.

The type of problems to be optimized is also important. An algorithm can be more suitable for a group of problems, while it might not be very efficient to another group of problems. Therefore, unlike what some people argue, the author thinks no restriction should be imposed on researchers in relation with developing new algorithms. Unfortunately, there is no solid approach for characterizing the metaheuristic algorithms and therefore one cannot easily identify the best ones.

References

1. Majid KI (1974) Optimum design of structures. Newness-Butterworth, UK
2. Kirsch U (1993) Structural optimization: fundamentals and applications. Springer, Berlin, Heidelberg
3. Gonzalez TF (ed) (2007) Handbook of approximation algorithms and metaheuristics, Computer and information science series. Chapman & Hall/CRC, Boca Raton, FL
4. Yang X-S (2010) Nature-inspired metaheuristic algorithms, 2nd edn. Luniver Press, UK

5. Glover F, Kochenberger GA (eds) (2003) Handbook of metaheuristics. Kluwer Academic Publishers, Boston, MA
6. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures, 1st edn. Springer, Switzerland
7. Kaveh A, Shokohi F (2016) Optimum design of laterally-supported castellated beams using tug of war optimization algorithm. Struct Eng Mech 58(3):533–553
8. Kaveh A, Bakhshpoori T, Barkhori M (2014) Optimum design of multi-span composite box girder bridges using Cuckoo Search algorithm. Steel Compos Struct Int J 17(5):705–719
9. Kaveh A, Ilchi Ghazaan M (2016) Enhanced whale optimization algorithm for sizing optimization of skeletal structures. Mech Based Des Struct Mach Int J (Published online: 21 July 2016)
10. Kaveh A, Moradveisi M (2016) Optimal design of double-layer barrel vaults using CBO and ECBO algorithms. Iran J Sci Technol Trans Civil Eng. doi:10.1007/s40996-016-0021-4
11. Kaveh A, Rezaei M (2015) Optimum topology design of geometrically nonlinear suspended domes using ECBO. Struct Eng Mech Int J 56(4):667–694
12. Kaveh A, Mirzaei B, Jafarvand A (2014) Shape-size optimization of single-layer barrel vaults using improved magnetic charged system search. Int J Civil Eng 12(4):447–465
13. Kaveh A, Ghafari MH (2016) Optimum design of steel floor system: effect of floor division number, deck thickness and castellated beams. Struct Eng Mech 59(5):933–950
14. Kaveh A, Mahdavi VR, Kamalinejad M (2016) Optimal design of the monopole structures using CBO and ECBO algorithms. Periodica Polytech Civil Eng (Published online)
15. Kaveh A, Maniat M (2014) Damage detection in skeletal structures based on charged system search optimization using incomplete modal data. Int J Civil Eng IUST 12(2):291–298
16. Kaveh A, Mahdavi VR (2016) A new method for modification of ground motions using wavelet transform and enhanced colliding bodies optimization. Appl Soft Comput 47:357–369
17. Kaveh A, Bijary Sh (2016) Bandwidth, profile and wavefront reduction using PSO, CBO, ECBO and TWO algorithms. Iran J Sci Technol Trans Civ Eng. doi:10.1007/s40996-016-0026-z
18. Kaveh A, Zolghadr A (2016) Optimal analysis and design of large-scale domes with frequency constraints. Smart Struct Syst 18(4):733–754
19. Kaveh A, Ilchi Ghazaan M (2016) Optimum design of large-scale truss towers using cascade optimization. Acta Mech. doi:10.1007/s00707-016-1588-3
20. Kaveh A, Ilchi Ghazaan M (2016) Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. Acta Mech. doi:10.1007/s00707-016-1725-z. First online: 20 Sept 2016
21. Kaveh A, Ardalani S (2016) Cost and CO_2 emission optimization of reinforced concrete frames using ECBO algorithm. Asian J Civil Eng 17(6):831–858
22. Kaveh A, Khanzadi M, Alipour M, Rastegar Moghaddam M (2016) Application of two new metaheuristic algorithms for construction site layout planning problem. Iran J Sci Technol 40 (4):263–275

Chapter 2
Optimum Design of Castellated Beams Using the Tug of War Algorithm

2.1 Introduction

In this chapter, the *tug of war algorithm* is applied to optimal design of castellated beams. Two common types of laterally supported castellated beams are considered as design problems: beams with hexagonal openings and beams with circular openings. Here, castellated beams have been studied for two cases: beams without filled holes and beams with end-filled holes. Also, tug of war optimization (TWO) algorithm is utilized for obtaining the solution of these design problems. For this purpose, the cost is taken as the objective function, and some benchmark problems are solved from literature (Kaveh and Shokohi [1]).

Since the 1940s, the manufacturing of structural beams with higher strength and lower cost has been an asset to engineers in their efforts to design more efficient steel structures. Due to the limitations on maximum allowable deflections, using section with heavyweight and high capacity in the design problem cannot always be utilized to the best advantage. As a result, several new methods have been created for increasing the stiffness of steel beams without increase in the weight of steel required. Castellated beam is one of the basic structural elements within the design of building, like a wide-flange beam (Konstantinos and D'Mello [2]).

A castellated beam is constructed by expanding a standard rolled steel section in such a way that a predetermined pattern (mostly circular or hexagonal) is cut on section webs and the rolled section is cut into two halves. The two halves are shifted and connected together by welding to form a castellated beam. In terms of structural performance, the operation of splitting and expanding the height of the rolled steel sections helps to increase the section modulus of the beams.

The main initiative for manufacturing and using such sections is to suppress the cost of material by applying more efficient cross-sectional shapes made from standard rolled beam. Web-openings have been used for many years in structural steel beams in a great variety of applications because of the necessity and economic advantages. The principal advantage of steel beam castellation process is that

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_2

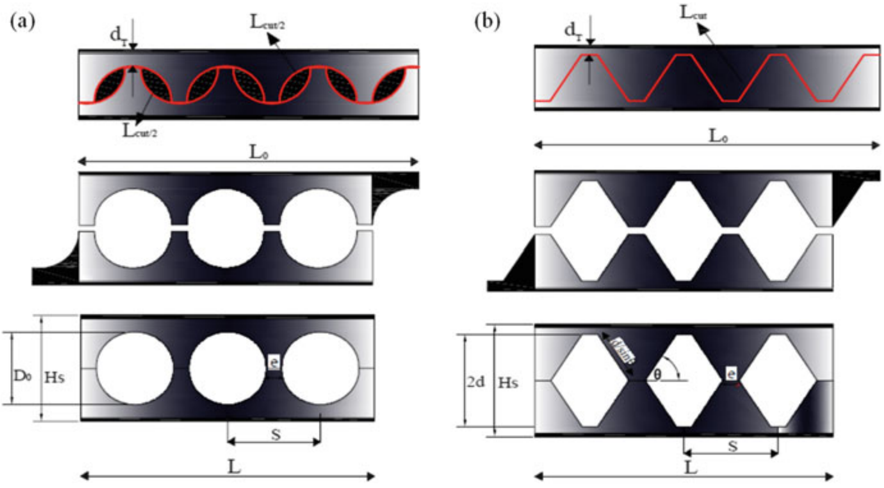


Fig. 2.1 (**a**) A castellated beam with circular opening. (**b**) A castellated beam with hexagonal

designer can increase the depth of a beam to raise its strength without adding steel. The resulting castellated beam is approximately 50 % deeper and much stronger than the original unaltered beam (Soltani et al. [3], Zaarour and Redwood [4], Redwood and Demirdjian [5], Sweedan [6], Konstantinos and D'Mello [7]).

In recent years, a great deal of progress has been made in the design of steel beams with web-openings, and a cellular beam is one of them. A cellular beam is the modern form of the traditional castellated beam, but with a far wider range of applications in particular as floor beams. Cellular beams are steel sections with circular openings that are made by cutting a rolled beam web in a half circular pattern along its centerline and re-welding the two halves of hot rolled steel sections as shown in Fig. 2.1. An increase in beam depth provides greater flexural rigidity and strength to weight ratio.

In practice, in order to support high shear forces close to the connections, sometimes it becomes necessary to fill certain openings. In cellular beams, this is achieved by inserting discs made of steel plates and welding from both sides (Fig. 2.2). The openings are usually filled for one of two reasons:

(i) At positions of higher shear, especially at the ends of a beam or under concentrated loads
(ii) At incoming connections of secondary beams

It should be noted that for maximum economy infills should be avoided whenever possible, even to the extent of increasing the section mass.

In the last two decades, many metaheuristic algorithms have been developed to help solve optimization problems that were previously difficult or impossible to solve using mathematical programming algorithms. Metaheuristic algorithms provide mechanisms to escape from local optima by balancing exploration and exploitation phases, being based either on solution populations or iterated solution paths,

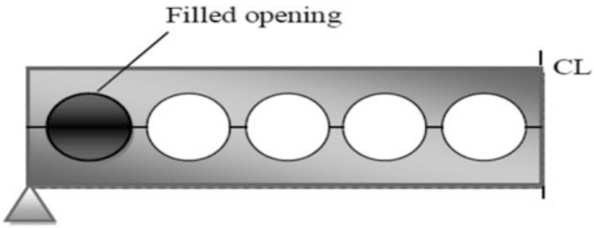


Fig. 2.2 Example of a beam with filled opening

for instance, by using neighborhoods. In general, these algorithms are simple to implement and present (near) optimal solutions in acceptable computational times even in complex search spaces. TWO is a multi-agent metaheuristic algorithm, which considers each candidate solution $X_i = \{x_{i,j}\}$ as a team engaged in a series of tug of war competitions.

The main aim of this study is to optimize the cost of castellated beams with and without end-filled openings. For this purpose, the tug of war optimization approach is utilized for design of such beams with circular and hexagonal holes.

The present chapter is organized as follows: In the next section, the design of castellated beam is introduced. In Sect. 2.3, the problem formulation including the mathematical model is presented, based on the Steel Construction Institute Publication Number 100 and Eurocode3. In Sect. 2.4, the algorithm is briefly introduced. In Sect. 2.5, numerical examples are studied, and finally the concluding remarks are provided in Sect. 2.6.

2.2 Design of Castellated Beams

The theory behind the castellated beam is to reduce the weight of the beam and to improve the stiffness by increasing the moment of inertia resulting from increased depth without usage of additional material. Due to the presence of holes in the web, the structural behavior of castellated steel beam is different from that of the standard beams. At present, there is no prescribed design method due to the complexity of the behavior of castellated beams and their associated modes of failure (Soltani et al. [3]). The strength of a beam with different shapes of web-openings is determined by considering the interaction of bending moment and shear at the openings. There are many failure modes to be considered in the design of a beam with web-opening, consisting of lateral-torsional buckling, Vierendeel mechanism, flexural mechanism, rupture of welded joints, and web post buckling. Lateral-torsional buckling may occur in an unrestrained beam. A beam is considered to be unrestrained when its compression flange is free to displace laterally and rotate. In this chapter it is assumed that the compression flange of the castellated beam is restrained by the floor system. Therefore, the overall buckling strength of the castellated beam is omitted from the design considerations. These modes are closely associated with beam geometry, shape parameters, type of loading, and provision of lateral supports. In the design of castellated beams, these criteria should be considered (EN 1993-1-1 [8], Ward [9], Erdal et al. [10], Saka [11], Raftoyiannis and Ioannidis [12], British Standards [13], AISC-LRFD [14]):

2.2.1 Overall Flexural Capacity of the Beam

This mode of failure can occur when a section is subjected to pure bending. In the span subjected to pure bending moment, the tee sections above and below the openings yield in a manner similar to that of a standard webbed beam. Therefore, the maximum moment under factored dead and imposed loading should not exceed the plastic moment capacity of the castellated beam (Soltani et al. [3], Erdal et al. [10]).

$$M_{\mathrm{U}} \leq M_{\mathrm{P}} = A_{\mathrm{LT}} P_{\mathrm{Y}} H_{\mathrm{U}} \tag{2.1}$$

where A_{LT} is the area of lower tee, P_{Y} is the design strength of steel, and H_{U} is the distance between center of gravities of upper and lower tees.

2.2.2 Shear Capacity of the Beam

In the design of castellated beams, two modes of shear failure should be checked. The first one is the vertical shear capacity, and the upper and lower tees should undergo that. The vertical shear capacity of the beam is the sum of the shear capacities of the upper and lower tees. The factored shear force in the beam should not exceed the following limits:

$$\begin{aligned} P_{\mathrm{VY}} &= 0.6 P_{\mathrm{Y}} (0.9 A_{\mathrm{WUL}}) \qquad \text{circular opening} \\ P_{\mathrm{VY}} &= \frac{\sqrt{3}}{3} P_{\mathrm{Y}} (A_{\mathrm{WUL}}) \qquad \text{hexagonal opening} \end{aligned} \tag{2.2}$$

The second one is the horizontal shear capacity. It is developed in the web post due to the change in axial forces in the tee section as shown in Fig. 2.3. Web post with too short mid-depth welded joints may fail prematurely when horizontal shear exceed the yield strength. The horizontal shear capacity is checked using the following equations (Soltani et al. [3], Erdal et al. [10]):

$$\begin{aligned} P_{\mathrm{VH}} &= 0.6 P_{\mathrm{Y}} (0.9 A_{\mathrm{WP}}) \qquad \text{circular opening} \\ P_{\mathrm{VH}} &= \frac{\sqrt{3}}{3} P_{\mathrm{Y}} (A_{\mathrm{WP}}) \qquad \text{hexagonal opening} \end{aligned} \tag{2.3}$$

where A_{WUL} is the total area of the web-opening and A_{WP} is the minimum area of web post.

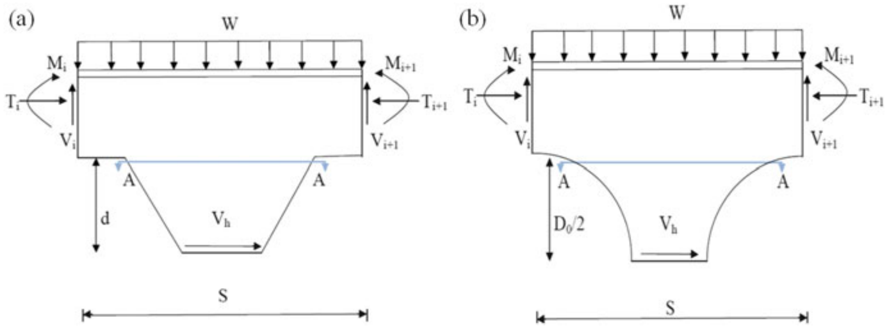


Fig. 2.3 Horizontal shear in the web post of castellated beams. (**a**) Hexagonal opening. (**b**) Circular opening

2.2.3 Flexural and Buckling Strength of Web Post

In this study, it is assumed that the compression flange of the castellated beam is restrained by the floor system. Thus the overall buckling of the castellated beam is omitted from the design consideration. The web post flexural and buckling capacity in a castellated beam is given by Soltani et al. [3] and Erdal et al. [10]):

$$\frac{M_{\mathrm{MAX}}}{M_E} = \left[C_1 \times \alpha - C_2 \times \alpha^2 - C_3\right] \tag{2.4}$$

where M_{MAX} is the maximum allowable web post moment and M_E is the web post capacity at critical section A–A shown in Fig. 2.3. C_1, C_2, and C_3 are constants obtained by the following expressions

$$C_1 = 5.097 + 0.1464\beta - 0.00174\beta^2 \tag{2.5}$$

$$C_2 = 1.441 + 0.0625\beta - 0.000683\beta^2 \tag{2.6}$$

$$C_3 = 3.645 + 0.0853\beta - 0.00108\beta^2 \tag{2.7}$$

where $\alpha = \frac{S}{2d}$ is for hexagonal openings and $\alpha = \frac{S}{D_0}$ is for circular openings, also $\beta = \frac{2d}{t_w}$ is for hexagonal openings, and $\beta = \frac{D_0}{t_w}$ is for circular openings and S is the spacing between the centers of holes, d is the cutting depth of hexagonal opening, D_0 is the hole diameter, and t_w is the web thickness.

2.2.4 Vierendeel Bending of Upper and Lower Tees

Vierendeel mechanism is always critical in steel beams with web-openings, where global shear force is transferred across the opening length, and the Vierendeel moment is resisted by the local moment resistances of the tee sections above and below the web-openings. This mode of failure often occurs in web-expanded beams with long horizontal opening lengths.

Vierendeel bending results in the formation of four plastic hinges above and below the web-opening. The overall Vierendeel bending resistance depends on the local bending resistance of the web-flange sections. This mode of failure is associated with high shear forces acting on the beam. The Vierendeel bending stresses in the circular opening obtained by using the Olander's approach. The interaction between Vierendeel bending moment and axial force for the critical section in the tee should be checked as follows (Erdal et al. [10]):

$$\frac{P_0}{P_U} + \frac{M}{M_P} \leq 1.0 \tag{2.8}$$

where P_0 and M are the force and the bending moment on the section, respectively. P_U is equal to the area of critical section $\times P_Y$, and M_P is calculated as the plastic modulus of critical section $\times P_Y$ in plastic section or elastic section modulus of critical section $\times P_Y$ for other sections.

The plastic moment capacity of the tee sections in castellated beams with hexagonal opening is calculated independently. The total of the plastic moment is equal to the sum of the Vierendeel resistances of the above and below tee sections (Soltani et al. [3]). The interaction between Vierendeel moment and shear forces should be checked by the following expression:

$$V_{OMAX} \times e - 4M_{TP} \leq 0 \tag{2.9}$$

where V_{OMAX} and M_{TP} are the maximum shear force and the moment capacity of tee section, respectively.

2.2.5 Deflection of Castellated Beam

Serviceability checks are of high importance in the design, especially in beams with web-opening where the deflection due to shear forces is significant. The deflection of a castellated beam under applied load combinations should not exceed span/360. Methods for calculating the deflection of castellated beam with hexagonal and circular openings are shown in Raftoyiannis and Ioannidis [12], and Erdal et al. [10], respectively.

2.3 Problem Formulation

In optimization problem of castellated beams, the objective is to minimize the manufacturing cost of the beam while satisfying certain constraints. In a castellated beam, there are many factors that require special considerations when estimating the cost of beam, such as man-hours of fabrication, weight, price of web cutting, and welding process. In this study, it is assumed that the costs associated with man-hours of fabrication for hexagonal and circular openings are identical. Thus, the objective function comprises of three parts: the beam weight, price of the cutting, and price of the welding. The objective function can be expressed as

$$F_{\text{cost}} = \rho A_{\text{initial}}(L_0) \times p_1 + L_{\text{cut}} \times p_2 + L_{\text{weld}} \times p_3 \tag{2.10}$$

In practice, in order to support high shear forces close to the connection or for reasons of fire safety, sometimes it becomes necessary to fill certain openings using steel plates. In this case, the price of plates is added to the total cost. Therefore, the objective function can be expressed as

$$\mathrm{F}_{\text{cost-filled}} = \rho(A_{\text{initial}}(L_0) + 2A_{\text{hole}} \times t_{\text{w}}) \times p_1 + L_{\text{cut}} \times p_2 + (L_{\text{weld}}) \times p_3 \tag{2.11}$$

where p_1, p_2, and p_3 are the price of the weight of the beam per unit weight, length of cutting, and welding per unit length, L_0 is the initial length of the beam before castellation process, ρ is the density of steel, A_{initial} is the area of the selected universal beam section, A_{hole} is the area of a hole, and L_{cut} and L_{weld} are the cutting length and welding length, respectively. The length of cutting is different for hexagonal and circular web-openings. The dimension of the cutting length is described by the following equations:

For circular opening,

$$L_{\text{cut}} = \pi D_0 \times NH + 2e(NH + 1) + \frac{\pi D_0}{2} + e \tag{2.12}$$

$$L_{\text{cut-infill}} = \pi D_0 \times NH + 2e(NH + 1) + \frac{\pi D_0}{2} + e + 2 \times P_{\text{hole}} \tag{2.13}$$

For hexagonal opening,

$$L_{\text{cut}} = 2NH\left(e + \frac{d}{\sin(\theta)}\right) + 2e + \frac{d}{\sin(\theta)} \tag{2.14}$$

$$L_{\text{cut-infill}} = 2NH\left(e + \frac{d}{\sin(\theta)}\right) + 2e + \frac{d}{\sin(\theta)} + 2 \times P_{\text{hole}} \tag{2.15}$$

where NH is the total number of holes, e is the length of horizontal cutting of web, D_0 is the diameter of holes, d is the cutting depth, θ is the cutting angle, and P_{hole} is the perimeter of hole related to filled opening.

Also, the welding length for both of circular and hexagonal openings is determined by Eqs. (2.16) and (2.17).

$$L_{\text{weld}} = e(NH + 1) \tag{2.16}$$

$$L_{\text{weld-infill}} = e(NH + 1) + 4 \times P_{\text{hole}} \tag{2.17}$$

2.3.1 Design of Castellated Beam with Circular Opening

Design process of a cellular beam consists of three phases: the selection of a rolled beam, the selection of a diameter, and the spacing between the center of holes and total number of holes in the beam as shown in Fig. 2.1 (Erdal et al. [10], Saka [11]). Hence, the sequence number of the rolled beam section in the standard steel sections' tables, the circular holes diameter, and the total number of holes are taken as design variables in the optimum design problem. This problem is formulated by considering the constraints explained in the previous sections and can be expressed as the following:

Find an integer design vector $\{X\} = \{x_1, x_2, x_3\}^T$, where x_1 is the sequence number of the rolled steel profile in the standard sections list, x_2 is the sequence number for the hole diameter which contains various diameter values, and x_3 is the total number of holes for the cellular beam (Erdal et al. [10]). Hence the design problem can be expressed as follows:

Minimize Eqs. (2.10) and (2.11)

Subjected to

$$g_1 = (1.08 \times D_0) - S \le 0 \tag{2.18}$$

$$g_2 = S - (1.60 \times D_0) \le 0 \tag{2.19}$$

$$g_3 = (1.25 \times D_0) - H_{\text{S}} \le 0 \tag{2.20}$$

$$g_4 = H_{\text{S}} - (1.75 \times D_0) \le 0 \tag{2.21}$$

$$g_5 = M_{\text{U}} - M_{\text{P}} \le 0 \tag{2.22}$$

$$g_6 = V_{\text{MAXSUP}} - P_{\text{V}} \le 0 \tag{2.23}$$

$$g_7 = V_{\text{OMAX}} - P_{\text{VY}} \le 0 \tag{2.24}$$

$$g_8 = V_{\text{HMAX}} - P_{\text{VH}} \le 0 \tag{2.25}$$

$$g_9 = M_{A\text{–AMAX}} - M_{\text{WMAX}} \le 0 \tag{2.26}$$

$$g_{10} = V_{\text{TEE}} - (0.50 \times P_{\text{VY}}) \le 0 \tag{2.27}$$

$$g_{11} = \frac{P_0}{P_{\text{U}}} + \frac{M}{M_{\text{P}}} - 1.0 \le 0 \tag{2.28}$$

$$g_{12} = Y_{\mathrm{MAX}} - {}^{L}\!/_{360} \leq 0 \tag{2.29}$$

where t_{W} is the web thickness, H_{S} and L are the overall depth and the span of the cellular beam, and S is the distance between centers of holes. M_{U} is the maximum moment under the applied loads, M_{P} is the plastic moment capacity of the cellular beam, V_{MAXSAP} is the maximum shear at support, V_{OMAX} is the maximum shear at the opening, V_{HMAX} is the maximum horizontal shear, and $M_{A-A\mathrm{MAX}}$ is the maximum moment at A–A section shown in Fig. 2.3. M_{WMAX} is the maximum allowable web post moment, V_{TEE} represents the vertical shear on top of the hole, P_0 and M are the internal forces on the web section, and Y_{MAX} denotes the maximum deflection of the cellular beam (Erdal et al. [10], AISC-LRFD [14]).

2.3.2 Design of Castellated Beam with Hexagonal Opening

In design of castellated beams with hexagonal openings, the design vector includes four design variables: the selection of a rolled beam, the selection of a cutting depth, the spacing between the center of holes and total number of holes in the beam, and the cutting angle as shown in Fig. 2.1. Hence the optimum design problem is formulated by the following expression:

Find an integer design vector $\{X\} = \{x_1, x_2, x_3, x_4\}^T$ where x_1 is the sequence number of the rolled steel profile in the standard sections' list, x_2 is the sequence number for the cutting depth which contains various values, x_3 is the total number of holes for the castellated beam, and x_4 is the cutting angle. Thus, the design problem turns out to be as follows:

Minimize Eq. (2.10), Eq. (2.11)

Subjected to

$$g_1 = d - \frac{3}{8}(H_{\mathrm{S}} - 2t_{\mathrm{f}}) \leq 0 \tag{2.30}$$

$$g_2 = (H_{\mathrm{S}} - 2t_{\mathrm{f}}) - 10 \times (d_{\mathrm{T}} - t_{\mathrm{f}}) \leq 0 \tag{2.31}$$

$$g_3 = \frac{2}{3} d \cot\theta - e \leq 0 \tag{2.32}$$

$$g_4 = e - 2d \cot\theta \leq 0 \tag{2.33}$$

$$g_5 = 2d \cot\theta + e - 2d \leq 0 \tag{2.34}$$

$$g_6 = 45^\circ - \theta \leq 0 \tag{2.35}$$

$$g_7 = \theta - 64^\circ \leq 0 \tag{2.36}$$

$$g_8 = M_{\mathrm{U}} - M_{\mathrm{P}} \leq 0 \tag{2.37}$$

$$g_9 = V_{\mathrm{MAXSUP}} - P_{\mathrm{V}} \leq 0 \tag{2.38}$$

$$g_{10} = V_{\mathrm{OMAX}} - P_{\mathrm{VY}} \leq 0 \tag{2.39}$$

$$g_{11} = V_{\mathrm{HMAX}} - P_{\mathrm{VH}} \leq 0 \tag{2.40}$$

$$g_{12} = M_{A-A\mathrm{MAX}} - M_{\mathrm{WMAX}} \leq 0 \tag{2.41}$$

$$g_{13} = V_{\mathrm{TEE}} - (0.50 \times P_{\mathrm{VY}}) \leq 0 \tag{2.42}$$

$$g_{14} = V_{\mathrm{OMAX}} \times e - 4M_{\mathrm{TP}} \leq 0 \tag{2.43}$$

$$g_{15} = Y_{\mathrm{MAX}} - {}^{L}\!/_{360} \leq 0 \tag{2.44}$$

where t_{f} is the flange thickness, d_{T} is the depth of the tee section, M_{P} is the plastic moment capacity of the castellated beam, $M_{A-A\mathrm{MAX}}$ is the maximum moment at A–A section shown in Fig. 2.3, M_{WMAX} is the maximum allowable web post moment, V_{TEE} is the vertical shear on the tee, M_{TP} is the moment capacity of the tee section, and Y_{MAX} denotes the maximum deflection of the castellated beam with hexagonal opening (Soltani et al. [3]).

2.4 Optimization Algorithm

In this section, the new metaheuristic algorithm developed by Kaveh and Zolghadr [15, 16] is briefly introduced. The TWO is a population-based search method, where each agent is considered as a team engaged in a series of tug of war competitions. The weight of the teams is determined based on the quality of the corresponding solutions, and the amount of pulling force that a team can exert on the rope is assumed to be proportional to its weight. Naturally, the opposing team will have to maintain at least the same amount of force in order to sustain its grip of the rope. The lighter team accelerates toward the heavier team, and this forms the convergence operator of the TWO. The algorithm improves the quality of the solutions iteratively by maintaining a proper exploration/exploitation balance using the described convergence operator. A summary of this method is provided in the following steps.

Step 1: Initialization
The initial positions of teams are determined randomly in the search space:

$$x_{ij}^{0} = x_{j,\min} + rand\left(x_{j,\max} - x_{j,\min}\right) \qquad j = 1, 2, \ldots, n \tag{2.45}$$

where x_{ij}^{0} is the initial value of the jth variable of the ith candidate solution; $x_{j,\max}$ and $x_{j,\min}$ are the maximum and minimum permissible values for the jth variable, respectively; *rand* is a random number from a uniform distribution in the interval [0, 1]; and n is the number of optimization variables.

Step 2: Evaluation of Candidate Designs and Weight Assignment The objective function values for the candidate solutions are evaluated and sorted. The best solution so far and its objective function value are saved. Each solution is considered as a team with the following weight:

$$W_i = 0.9\left(\frac{fit(i) - fit_{\text{worst}}}{fit_{best} - fit_{\text{worst}}}\right) + 0.1 \qquad i = 1, 2, \ldots, N \tag{2.46}$$

where $fit(i)$ is the fitness value for the ith particle. The fitness value can be considered as the penalized objective function value for constrained problems; fit_{best} and fit_{worst} are the fitness values for the best and worst candidate solutions of the current iteration. According to Eq. (2.46) the weights of the teams range between 0.1 and 1.

Step 3: Competition and Displacement In TWO each team competes against all the others one at a time to move to its new position. The pulling force exerted by a team is assumed to be equal to its static friction force ($W\mu_s$). Hence the pulling force between the teams i and j ($F_{p,ij}$) can be determined as $\max\{W_i\mu_s, W_j\mu_s\}$. Such a definition keeps the position of the heavier team unaltered.

The resultant force affecting team i due to its interaction with heavier team j in the kth iteration can then be calculated as follows:

$$F^k_{r,ij} = F^k_{p,ij} - W^k_i \mu_k \tag{2.47}$$

where $F^k_{p,ij}$ is the pulling force between teams i and j in the kth iteration and μ_k is coefficient of kinematic friction.

$$a^k_{ij} = \left(\frac{F^k_{r,ij}}{W^k_i \mu_k}\right) g^k_{ij} \tag{2.48}$$

in which a^k_{ij} is the acceleration of team i toward team j in the kth iteration and g^k_{ij} is the gravitational acceleration constant defined as

$$g^k_{ij} = X^k_j - X^k_i \tag{2.49}$$

where X^k_j and X^k_i are the position vectors for candidate solutions j and i in the kth iteration. Finally, the displacement of team i after competing with team j can be derived as

$$\Delta X^k_{ij} = \frac{1}{2} a^k_{ij} \Delta t^2 + \alpha^k (X_{\max} - X_{\min}) \circ (-0.5 + rand(1, n)) \tag{2.50}$$

The second term of Eq. (2.50) induces randomness into the algorithm. This term can be interpreted as the random portion of the search space traveled by team i before it stops after the applied force is removed. Here, α is a constant chosen from the interval [0,1]; $X_{\max}$ and $X_{\min}$ are the vectors containing the upper and lower bounds of the permissible ranges of the design variables, respectively; $\circ$ denotes element by element multiplication; and $rand(1, n)$ is a vector of uniformly distributed random numbers.

It should be noted that when team j is lighter than team i, the corresponding displacement of team i will be equal to zero (i.e., ΔX_{ij}^k). Finally, the total displacement of team i in iteration k is equal to

$$\Delta X_i^k = \sum_{j=1}^{N} \Delta X_{ij}^k \tag{2.51}$$

The new position of team i at the end of the kth iteration is then calculated as

Step 4: Handling of Side Constraints

It is possible for the candidate solutions to leave the search space, and it is important to deal with such solutions properly. This is especially the case for the solutions corresponding to lighter teams for which the values of ΔX are usually bigger. Different strategies might be used in order to solve this problem. In this study, it is assumed that such candidate solution can be simply brought back to their previous permissible position (Flyback strategy) or they can be regenerated randomly.

Step 5: Termination

Steps 2 through 5 are repeated until a termination criterion is satisfied.

Flowchart of the TWO algorithm is shown in Fig. 2.4.

The pseudo-code for design of castellated beam using the tug of war optimization algorithm is shown in Fig. 2.5. It should be noted that each team is considered a beam.

2.5 Test Problems and Optimization Results

In this section, numerical results are presented to demonstrate the efficiency of the new metaheuristic method (TWO) for design of castellated beams. For this purpose, three beams are selected from literature that have previously been optimized by other algorithms. Among the steel sections' list of British Standards, 64 universal beam (UB) sections starting from 254 × 102 × 28 UB to 914 × 419 × 388 UB are chosen to constitute the discrete set of steel sections from which the design algorithm selects the cross-sectional properties for the castellated beams. In the design pool of holes diameters, 421 values are arranged which vary between 180 and 600 mm with an increment of 1 mm. Also, for cutting depth of hexagonal opening, 351 values are considered which vary between 50 and 400 mm with an increment of 1 mm and cutting angle changes from 45 to 64. Another discrete set is arranged for the number of holes. Likewise, in all the design problems, the modulus of elasticity is equal to 205 GPa and Grade 50 is selected for the steel of the beam

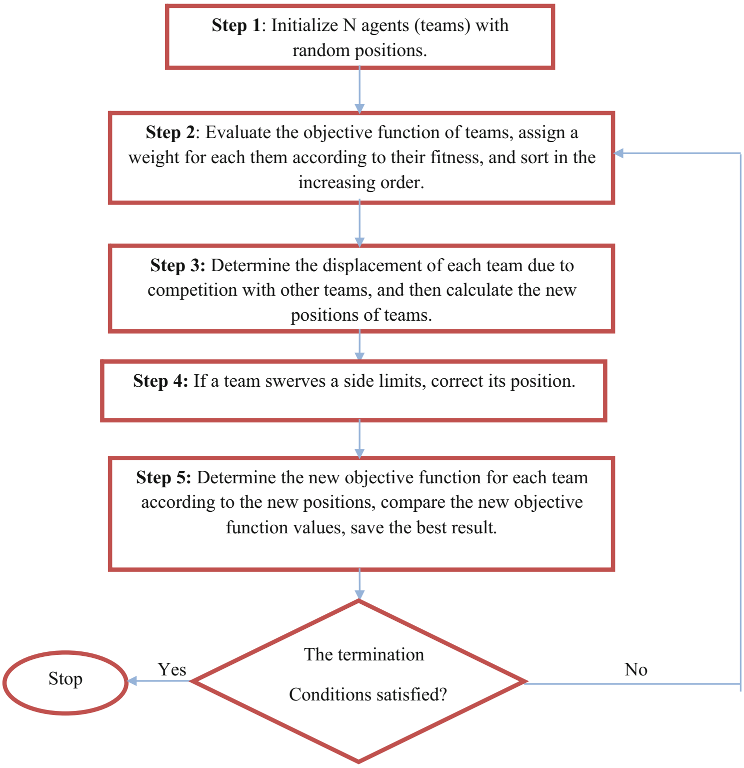


Fig. 2.4 Flowchart of the TWO algorithm

which has the design strength of 355 MPa. The coefficients P_1, P_2, and P_3 in the objective function are considered as 0.85, 0.30, and 1.00, respectively (Kaveh and Shokohi [17–20]). A maximum number of iterations of 200 are used as the termination criterion in all the examples, and α is taken as 0.1 for all design problems. Also, all design problems have been solved in two cases, with and without filled holes.

2.5.1 Castellated Beam with 4 m Span

A simply supported beam with a span of 4 m is considered as the first test problem, shown in Fig. 2.6. The beam is subjected to 5 kN/m dead load including its own weight. A concentrated live load of 50 kN also acts at mid-span of the beam, and the allowable displacement of the beam is limited to 12 mm. For this problem the number of agents (teams) is taken as 20.

procedure Design of a Castellated Beam using the Tug of War Optimization algorithm
begin
Initialize parameters**;** Such as NOA, NOV, ROV, …% NOA=Number of Agent(Team), NOV=Number of Variable, ROV=Range of Variable.
Generate a population of NOA random candidate solutions (Beams);
while (not termination condition) **do**
Analyze beams and evaluate the objective function values for them.
Define the weights of the teams (Beams) W_i based on fit(X_i)
Sort the solutions and save the best one so far.
for each team i
for each team j
if ($W_i < W_j$)
Move team i towards team j using Eq. (2.50);
end if
end for
Calculate the total displacement of team i using Eq. (2.51);
Determine the final position of team i using $X_i^{k+1} = X_i^k + \Delta X_i^k$
Use the side constraint handling technique to regenerate violating variables
Determine the new objective function for each team according to the new positions and save the best result.
end for
end while
end

Fig. 2.5 The pseudo-code for design of castellated beam using the TWO algorithm

Castellated beams with hexagonal and circular openings are separately designed with TWO. These beams are designed for two cases. In case 1, it is assumed that the end of the beams is not filled. Thus the objective function for this case is obtained from Eq. (2.10). In the second case, it is assumed that the holes in the end of the beam are filled with steel plate, and Eq. (2.11) is utilized for the objective function. The optimum results obtained by TWO are given in Table 2.1. It is apparent from the same table that the optimum cost for castellated beam with hexagonal hole is equal to 89.73$ which is obtained by TWO. Also, according to the results, the tug of war optimization algorithm has good performance in design of cellular beam. These results indicate that the castellated beam with hexagonal opening has less cost in comparison to the cellular beam. The same conclusion can be drawn for the filled opening configuration from the results listed in Table 2.1.

Figure 2.7 shows the convergence curves of the TWO algorithm for design of castellated beams with different shapes for the openings.

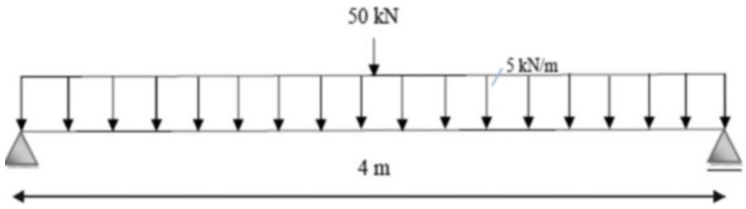


Fig. 2.6 Simply supported beam with a span of 4 m

2.5.2 Castellated Beam with 8 m Span

In the second problem, the tug of war optimization algorithm is used to design a simply supported castellated beam with a span of 8 m. Similar to the first example, this beam is also designed for two different cases. The beam carries a uniform dead load 0.40 kN/m, which includes its own weight. In addition, it is subjected to two concentrated loads as shown in Fig. 2.8. The allowable displacement of the beam is limited to 23 mm, and the number of agents is taken as 20.

This beam is designed by TWO, and the results are compared to those of the other optimization algorithms as shown in Table 2.2. In design of the beam with hexagonal hole, the corresponding cost obtained by the TWO is equal to 718.2$ which is the lowest value among all the methods. Therefore, the performance of the tug of war optimization is better than other approaches (Kaveh and Shokohi [17–20]) for this design example. According to the obtained results, the designed beam with hexagonal opening has less cost in comparison with the cellular beam, and it is a better option in this case. In design of end-filled case, it is obvious that the presented method has the same performance. Furthermore, the maximum value of the strength ratio is equal to 0.99 for both hexagonal and circular beams, and it is shown that these constraints are dominant in the design process.

Figure 2.9 shows the convergence history for optimum design of hexagonal beam which is obtained by different metaheuristic algorithms.

2.5.3 Castellated Beam with 9 m Span

The beam with 9 m span is considered as the last example of this study in order to compare the minimum cost of the castellated beams. The beam carries a uniform load of 40 kN/m including its own weight and two concentrated loads of 50 kN as shown in Fig. 2.10. The allowable displacement of the beam is limited to 25 mm, and the number of agent is taken as 20.

Table 2.3 compares the results obtained by the TWO with those of the other algorithms. In the optimum design of castellated beam with hexagonal hole, TWO algorithm selects 684 × 254 × 125 UB profile, 16 holes, and 231 mm for the cutting

Table 2.1 Optimum designs of the castellated beams with 4 m span

	Algorithm	Optimum UB section	Hole diameter – cutting depth (mm)	Total number of holes	Cutting angle	Minimum cost ($)	Type of the hole
Case 1	ECSS [17]	UB 305 × 102 × 25	125	14	57°	89.78	Hexagonal
	CBO [18]	UB 305 × 102 × 25	125	14	57°	89.78	
	CBO-PSO [20]	UB 305 × 102 × 25	125	14	57°	89.78	
	TWO [1]	UB 305 × 102 × 25	126	13	61°	89.73	
	ECSS [17])	UB 305 × 102 × 25	248	14	–	96.32	Circular
	CBO [18])	UB 305 × 102 × 25	244	14	–	91.14	
	CBO-PSO [20])	UB 305 × 102 × 25	243	14	–	91.08	
	TWO [1]	UB 305 × 102 × 25	249	14	–	91.15	
Case 2	ECSS [19]	UB 305 × 102 × 25	125	14	60°	96.45	Hexagonal
	CBO [19]	UB 305 × 102 × 25	125	14	64°	96.61	
	CBO-PSO [19]	UB 305 × 102 × 25	125	14	56°	96.04	
	TWO [1]	UB 305 × 102 × 25	125	14	56°	96.33	
	ECSS [19]	UB 305 × 102 × 25	244	14	–	98.62	Circular
	CBO [19]	UB 305 × 102 × 25	243	14	–	98.70	
	CBO-PSO [19]	UB 305 × 102 × 25	243	14	–	98.58	
	TWO [1]	UB 305 × 102 × 25	244	14	–	98.62	

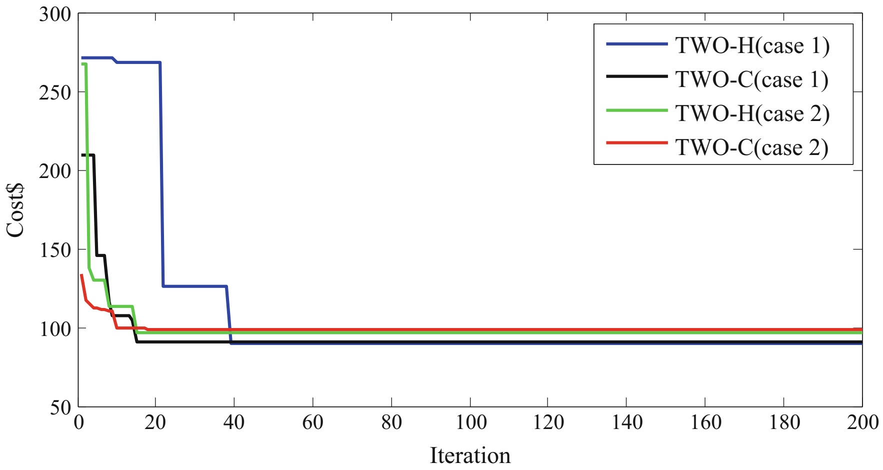


Fig. 2.7 Convergence curves recorded in the 4 m span beam problem for the TWO best optimization runs [1]

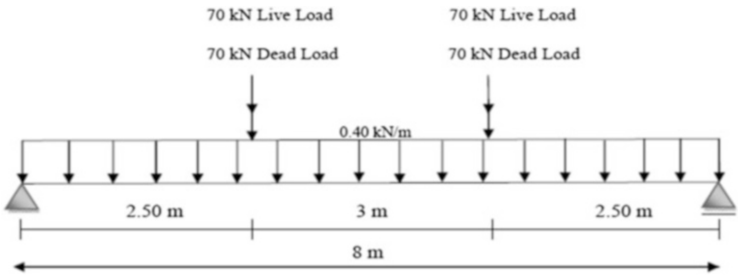


Fig. 2.8 Simply supported beam with a span of 8 m

depth and 57° for the cutting angle. The minimum cost of the design beam is equal to 991.04$. Also, in the optimum design of cellular beam, the TWO algorithm selects 610 × 229 × 125 UB profile, 14 holes of diameter 490 mm. It can be observed from Table 2.3 that the optimal design has the minimum cost of 990.33 $ for beam with hexagonal holes which is obtained by the CBO-PSO algorithm; however, the TWO results in better design for cellular beam. In the design of beam with filled holes, the obtained results using the tug of war optimization algorithm are slightly different from each other. This shows that in the case of holes filled with steel plates, where the beam span is large, using cellular beams can be a good design strategy. Similar to the previous example, the strength criteria are dominant in the design of this beam, and it is related to the Vierendeel mechanism. The maximum ratio of these criteria is equal to 0.99 for both hexagonal and cellular cases.

The optimum shapes of the hexagonal and circular openings with unfilled holes are separately shown in Fig. 2.11. Also, the convergence histories of metaheuristics

Table 2.2 Optimum designs of the castellated beams with 8 m span

	Algorithm	Optimum UB section	Hole diameter – cutting depth (mm)	Total number of holes	Cutting angle	Minimum cost ($)	Type of the hole
Case 1	ECSS [17]	UB 610 × 229 × 101	246	14	59°	719.47	Hexagonal
	CBO [18]	UB 610 × 229 × 101	243	14	59°	718.93	
	CBO-PSO [20]	UB 610 × 229 × 101	244	14	55°	718.33	
	TWO [1]	UB 610 × 229 × 101	243	14	56°	718.20	
	ECSS [18]	UB 610 × 229 × 101	487	14	–	721.55	Circular
	CBO [18]	UB 610 × 229 × 101	487	14	–	721.55	
	CBO-PSO [20]	UB 610 × 229 × 101	487	14	–	721.55	
	TWO [1]	UB 610 × 229 × 101	487	14	–	721.55	
Case 2	ECSS [19]	UB 610 × 229 × 101	246	14	56°	744.65	Hexagonal
	CBO [19]	UB 610 × 229 × 101	246	14	58°	745.48	
	CBO-PSO [19]	UB 610 × 229 × 101	246	14	55°	744.42	
	TWO [1]	UB 610 × 229 × 101	246	14	55°	744.42	
	ECSS [19]	UB 610 × 229 × 101	478	14	–	753.74	Circular
	CBO [19]	UB 610 × 229 × 101	479	14	–	754.02	
	CBO-PSO [19]	UB 610 × 229 × 101	478	14	–	753.74	
	TWO [1]	UB 610 × 229 × 101	478	14	–	753.74	

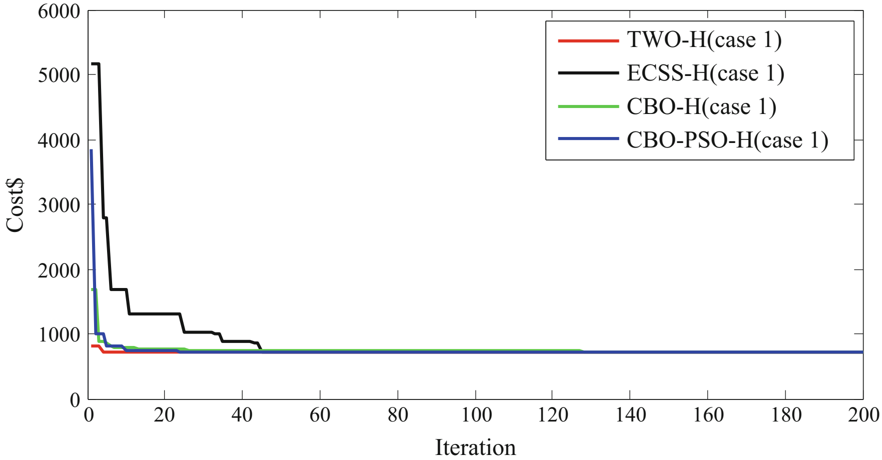


Fig. 2.9 Comparison of best run convergence curves recorded in the 8 m span beam problem (unfilled hexagonal holes) for different metaheuristic algorithms [1]

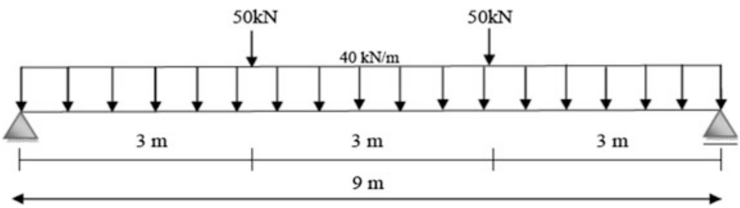


Fig. 2.10 Simply supported beam with 9 m span

are shown in Fig. 2.12 for design of cellular beam with filled openings. It is apparent from the figure that TWO has good convergence rate in design of this problem and finds better solution for cellular beam.

2.6 Concluding Remarks

In this chapter, the newly developed metaheuristic algorithm so-called tug of war optimization is utilized for optimum design of castellated beams. Three benchmark problems are solved in order to assess the robustness and efficiency of the TWO. These beams are designed for two cases, with filled openings and unfilled openings, where the hexagonal and circular holes are considered as the types of the web-openings. Comparing the results obtained by TWO with those of other optimization methods demonstrates that TWO has a better performance in the ability of finding the optimum solution. Also, the convergence rate of this algorithm to the optimal solution is quite good for most of problems, and it requires a less number of

Table 2.3 Optimum designs of the castellated beams with 9 m span

	Algorithm	Optimum UB section	Hole diameter – cutting depth (mm)	Total number of holes	Cutting angle	Minimum cost ($)	Type of the hole
Case 1	ECSS [17]	UB 684 × 254 × 125	277	13	56°	995.97	Hexagonal
	CBO [18]	UB 684 × 254 × 125	233	15	64°	993.79	
	CBO-PSO [20]	UB 684 × 254 × 125	230	16	56°	990.33	
	TWO [1]	UB 684 × 254 × 125	231	16	57°	991.04	
	ECSS [17]	UB 684 × 254 × 125	539	14	–	998.94	Circular
	CBO [18]	UB 684 × 254 × 125	538	14	–	997.57	
	CBO-PSO [20]	UB 684 × 254 × 125	538	14	–	998.58	
	TWO [1]	UB 610 × 229 × 125	490	14	–	995.89	
Case 2	ECSS [19]	UB 684 × 254 × 125	277	14	61°	1033.32	Hexagonal
	CBO [19]	UB 684 × 254 × 125	277	14	60°	1034.07	
	CBO-PSO [19]	UB 684 × 254 × 125	276	14	58°	1031.92	
	TWO [1]	UB 684 × 254 × 125	277	14	57°	1031.98	
	ECSS [19]	UB 684 × 254 × 125	539	14	–	1041.71	Circular
	CBO [19]	UB 684 × 254 × 125	539	14	–	1041.79	
	CBO-PSO [19])	UB 684 × 254 × 125	539	14	–	1041.68	
	TWO [1]	UB 610 × 229 × 125	489	15	–	1033.34	

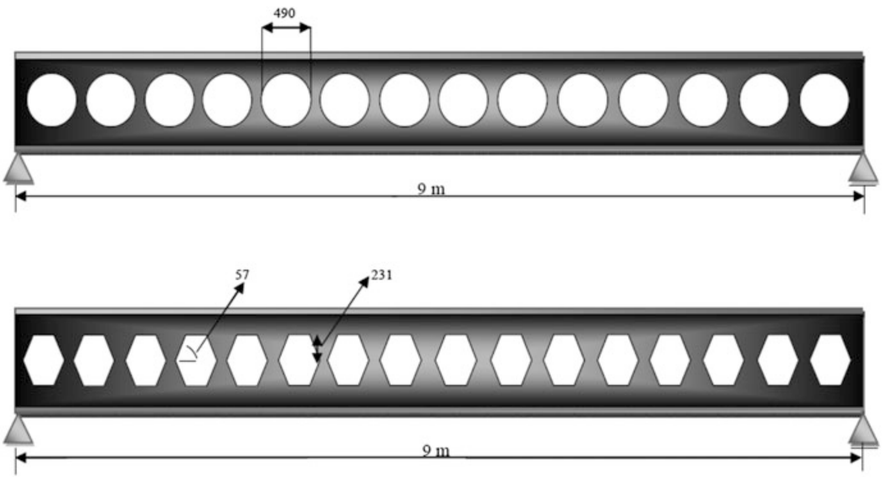


Fig. 2.11 Optimum profiles of the castellated beams with unfilled cellular and hexagonal openings for beam with 9 m span

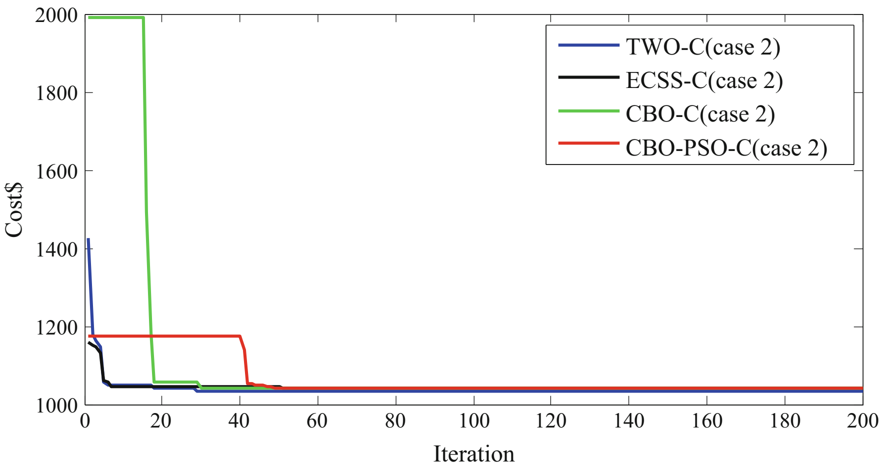


Fig. 2.12 Comparison of best run convergence curves recorded in the 9 m span beam problem (*filled circular holes*) for different metaheuristic algorithms [1]

analyses to find better solution making TWO computationally more efficient. From the results obtained in this chapter, it can be concluded that the use of the beam with hexagonal openings leads to the use of less steel material and it is a better choice than cellular beam in unfilled cases. For design of castellated beam with large spans, especially in filled cases, it is observed that the cellular beam has a better performance and it can be used as an alternative to castellated beam with hexagonal opening.

References

1. Kaveh A, Shokohi F (2016) Optimum design of laterally-supported castellated beams using tug of war optimization algorithm. Struct Eng Mech 58(3):533–553
2. Konstantinos T, D'Mello C (2012) Optimisation of novel elliptically-based web opening shapes of perforated steel beams. J Constr Steel Res 76:1605–1620
3. Soltani MR, Bouchair A, Mimoune M (2012) Nonlinear FE analysis of the ultimate behavior of steel castellated beams. J Constr Steel Res 70:101–114
4. Zaarour W, Redwood RG (1996) Web buckling in thin webbed castellated beams. J Struct Eng ASCE 122(8):860–866
5. Redwood R, Demirdjian S (1998) Castellated beam web buckling in shear. J Struct Eng ASCE 124(10):1202–1207
6. Sweedan MI (2011) Elastic lateral stability of I-shaped cellular steel beams. J Constr Steel Res 67(2):151–163
7. Konstantinos T, D'Mello C (2011) Web buckling study of the behavior and strength of perforated steel beam with different novel web opening shapes. J Constr Steel Res 67 (10):1605–1620
8. EN 1993-1-1 (2005) Eurocode 3: design of steel structures part 1–1: general rules and rules for building. CEN
9. Ward JK (1990) Design of composite and non-composite cellular beams. The Steel Construction Institute Publication, UK
10. Erdal F, Dogan E, Saka MP (2011) Optimum design of cellular beams using harmony search and particle swarm optimization. J Constr Steel Res 67(2):232–237
11. Saka MP (2009) Optimum design of steel skeleton structures. Stud Comput Intell 191:87–112
12. Raftoyiannis I, Ioannidis G (2006) Deflection of castellated I-beams under transverse loading. J Steel Struct 6(1):31–36
13. British Standards, BS 5950 (2000) Structural use of steel works in building. Part 1. Code of practice for design in simple and continuous construction, hot rolled sections. British Standard Institute, London
14. LRFD-AISC (1986) Manual of steel construction-load and resistance factor design, SA
15. Kaveh A, Zolghadr A (2016) A novel metaheuristic algorithm: tug of war optimization. Int J Optim Civil Eng 6(4):469–492
16. Kaveh A, Zolghadr A (2017) Truss shape and size optimization with frequency constraints using Tug of War Optimization. Asian J Civil Eng 7(2):311–333
17. Kaveh A, Shokohi F (2014) Cost optimization of castellated beams using charged system search algorithm. Iran J Sci Technol Trans Civil Eng 38(C1+):235–249
18. Kaveh A, Shokohi F (2015) Optimum design of laterally-supported castellated beams using CBO algorithm. Steel Compos Struct 18(2):305–324
19. Kaveh A, Shokohi F (2015) Cost optimization of end-filled castellated beams using metaheuristics algorithms. Int J Optim Civil Eng 5(3):335–256
20. Kaveh A, Shokohi F (2016) A hybrid optimization algorithm for the optimal design of laterally-supported castellated beams. Sci Iran 23(2):508–519

Chapter 3
Optimum Design of Multi-span Composite Box Girder Bridges Using Cuckoo Search Algorithm

3.1 Introduction

Composite steel–concrete box girders are frequently used in bridge construction for their economic and structural advantages. An integrated metaheuristic based optimization procedure is proposed for discrete size optimization of straight multi-span steel-box girders with the objective of minimizing the self-weight of the girder. The selected metaheuristic algorithm is the *cuckoo search (CS) algorithm*. The optimum design of a box girder is characterized by geometry, serviceability, and ultimate limit states specified by the American Association of State Highway and Transportation Officials (AASHTO). Size optimization of a practical design example investigates the efficiency of this optimization approach and leads to around 15 % of saving in material (Kaveh et al. [1]).

For every product designed to satisfy human needs, the creator tries to achieve the best solution for the task in hand (safety and serviceability) and therefore performs optimization. This chapter is concerned with discrete size optimization of straight multi-span steel-box girders with the objective of minimizing the self-weight of girder. Composite steel-box girders in the form of built-up steel-box sections and concrete deck slabs have become very frequent due to some positive structural features such as high torsional and wrapping rigidity, aesthetical appeal with regard to relatively large span-depth ratio, and economical advantages in fabrication and maintenance (Chen and Yen [2]). Developments in computer hardware and software, advances in computer-based analysis and design tools, and advances in numerical optimization methods make it possible to formulate design of complicated discrete engineering problems as optimization problems and solve them by one of the optimization methods (Rana et al. [3]). Further developments on box girders can be found in the works of Ding et al. [4] and Ko et al. [5].

Many optimization methods have been developed during the last decades pioneered by the traditional mathematical-based methods which use the gradient

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_3

information to search the optimal solutions with drawbacks such as complex derivatives, sensitivity to initial values, being limited to continuous search spaces, and the large amount of numerical memory required (Lee and Geem [6]). Although some mathematical programming-based methods have been developed for discrete optimum design problems, they are not very efficient for obtaining the optimum solution of the large size practical design problems (Saka [7]). In recent years, the other class of optimization techniques, stochastic optimization algorithms inspired by natural mechanisms, has been produced for overcoming these disadvantages which which make it possible to optimize complicated discrete engineering optimization problems.

Due to the presence of large number of design variables, discrete values of variables, large size of search space, difficulties of modeling and analyzing methods, and many constraints including stress, deflection, and geometry limitations under various load types, size optimization of multi-span steel-box girders has not been attempted. In practice several techniques with various degrees of consistency are available for analysis. These range from the elementary or engineer's beam theory to complex-shell finite element analyses (Razaqpur and Li [8]). One of the most prevalent analysis and design tools, the SAP2000, is employed in this study and also took advantage of its open application programming interface (OAPI) feature to model as practical and detailed as possible. To take full advantage of the enhancements offered by the new multi-core hardware era, the MATLAB software with its Parallel Computing Toolbox is used in this research (Luszczek [9]). A population-based algorithm namely cuckoo search (CS), inspired by the behavior of some cuckoo species in combination with Lévy flight behavior (Yang and Deb [10, 11]), is selected to optimize straight multi-span composite steel-box girders under self-weight. This population-based algorithm like the other ones can benefit the features of parallel computing and has been used successfully for discrete optimum deign of truss structures, 2D and 3D frames (Kaveh et al. [12], Saka and Dogan [13], Saka and Geem [14], Kaveh and Bakhshpoori [15]). In order to verify the efficiency of the CS, two other algorithms are also used to determine the solution of the considered discrete optimization problem. These are the harmony search method (HS) (Kochenberger and Glover [16]) and particle swarm optimization (PSO) (Kennedy et al. [17]) algorithm.

Taking into account all restrictions imposed by American Association of State Highway and Transportation Officials (AASHTO [18]), a practical design example is optimized using the proposed integrated parallel optimization procedure. The results reveal a saving of around 15 % of material for the considered bridge girder.

The remaining sections of this chapter are organized as follows. Section 3.2 states the design optimization problem. Section 3.3 outlines the details of parallel CS-based optimization procedure. Section 3.4 contains a comprehensive practical design optimized by the proposed method, to illustrate the features of the design method. The chapter is concluded in Sect. 3.5.

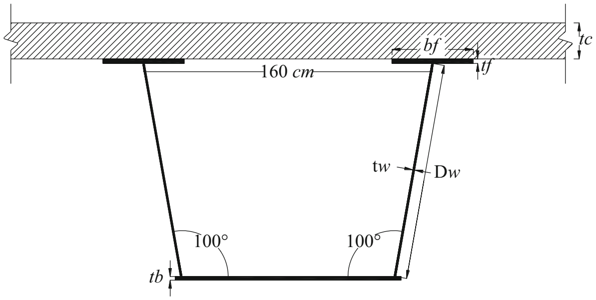


Fig. 3.1 A typical section of steel-box girder

3.2 Design Optimization Problem

After the topology and support conditions are established, the girder is divided into some segments along the girder length. The process of division is based on fabrication requirements. The main design effort involves sizing the individual girder sections for the predetermined segments with the objective of minimizing the self-weight of the girder. A typical section for composite steel–concrete box girder is shown in Fig. 3.1. As it is depicted, the design variables in each section are slab thickness (t_c), top flange width (b_f), top flange thickness (t_f), web depth (D_w), web thickness (t_w), and bottom flange thickness (t_b). The center to center distance of the top flanges and the inclination angle of web from the vertical direction are fixed to 160 cm and 100°, respectively, for the entire girder because of fabrication conditions. As a result, the width of bottom flange is a function of other variables.

The design procedure based on the AASHTO Division I [18] provisions can be outlined as follows:

3.2.1 Loading

Maximum compressive and tensile stresses in girders that are not provided with temporary supports during the placing of the permanent dead load are the sum of the stresses produced by the dead loads acting on the steel girders alone and the stresses produced by the superimposed loads acting on the composite girder. Therefore, two different dead loads should be considered. In the first case, the dead load is exerted on the non-composite section ($L1$). This load involves self-weight of the steel girder and weight of the concrete deck. The second case is applied on the composite section which includes the pavement, curb, pedestrian, and guard fence loads ($L2$). The highway live loads on the roadways of bridges or incidental structures shall consist of standard trucks or lane loads that are equivalent to truck trains. AASHTO HS loading is applied in this study. The live load for each box girder ($L3$) shall be determined by applying to the girder the fraction W_L of a wheel load determined by the following equation:

$$W_{\mathrm{L}} = 0.1 + 1.7R + 0.85/N_{\mathrm{w}}, R = N_{\mathrm{w}}/\text{Number of girders} \tag{3.1}$$

in which N_{w} is the number of lanes. Dynamic effects of live load should be taken into account as an impact coefficient based on Article 3.8.2 from the AASHTO [18].

3.2.2 Geometric Constraints

According to Section 10 of the AASHTO [18], the following geometry limitations are imposed on the section:

$$\begin{cases} g_1 : \dfrac{t_{\mathrm{w}} \times 1.5}{t_{\mathrm{f}}} - 1 \le 0 \\ g_2 : \dfrac{D_{\mathrm{w}} \times 0.2}{b_{\mathrm{f}}} - 1 \le 0 \\ g_3 : \dfrac{b_{\mathrm{f}}}{t_{\mathrm{f}} \times 23} - 1 \le 0 \\ g_4 : \dfrac{D_{\mathrm{w}}}{t_{\mathrm{w}} \times 327} - 1 \le 0 \end{cases} \tag{3.2}$$

3.2.3 Strength Constraints

The flanges of section, both top and bottom, should be designed for flexural resistance as follows:

$$\begin{gathered} g_5 : \frac{\sigma_{\mathrm{top}}}{\sigma_{\mathrm{all}}(\mathrm{top})} - 1 \le 0 \\ g_6 : \frac{\sigma_{\mathrm{top}}}{\sigma_{\mathrm{all}}(\mathrm{bot})} - 1 \le 0 \end{gathered} \tag{3.3}$$

The flexural stresses of top and bottom flanges, σ(top) and σ(bot), are calculated under three loading conditions: the section without considering concrete slab under $L1$, the composite section under $L2$ with creep and shrinkage effects, and the composite section under live loads without long-term effects. Creep and shrinkage effects are taken into account by dividing concrete elastic modulus by 3 based on 10.38.1.4 (AASHTO [18]). The allowable stress of top flange, σ_{all}(top), and tensile allowable stress of bottom flange, σ_{all}(bot), are equal to 0.55 F_{y}. The bottom flange allowable compressive stress is supplied on the 10. 39. 4. 3.

Concrete compressive stress under $L2$ and $L3$ loads should satisfy the following constraint:

$$g_7 : \frac{\sigma_{\text{concrete}}}{0.4f'_{\text{c}}} - 1 \leq 0 \tag{3.4}$$

in which f'_{c} is concrete cylindrical compressive strength.

Shear stresses in the web should be bounded by allowable shear stress as follows:

$$g_8 : \frac{\left(f_{\text{v}} = \frac{V}{2D_{\text{w}}t_{\text{w}}\cos\theta}\right)}{F_{\text{v}}} - 1 \leq 0 \tag{3.5}$$

where V is the shear under dead and live loads (all three load conditions) and θ is the inclination angle of the web, f_{v} is the shear stress, and F_{v} is the allowable shear stress which is obtained by 10.39.3.1.

3.2.4 Serviceability Constraints

Complying with Sect. 10.6, the composite girder deflections under live load plus the live load impact (Δ_{L+I}) for each span shall not exceed 1/800 span length (S) which can be presented as follows:

$$g_9 : \frac{800 \times \Delta_{L+I}}{S} - 1 \leq 0 \tag{3.6}$$

3.3 Parallel Metaheuristic Based Optimization Technique

3.3.1 Cuckoo Search Algorithm

Cuckoo search is a metaheuristic algorithm inspired by some species of a bird family called cuckoo because of their special lifestyle and aggressive reproduction strategy (Yang and Deb [11]). These species lay their eggs in the nests of other host birds with amazing abilities like selecting the recently spawned nests and removing existing eggs that increase hatching probability of their eggs. The host takes care of the eggs presuming that the eggs are its own. However, some of host birds are able to combat with this parasite behavior of cuckoos and throw out the discovered alien eggs or build their new nests in new locations. The cuckoo breeding analogy is used for developing new design optimization algorithm. A generation is represented by a set of host nests. Each nest carries an egg (solution). The quality of the solutions is improved by generating a new solution from an existing solution and modifying certain characteristics. The number of solutions remains fixed in each generation. In this study the later version of the CS algorithm is used, which is first introduced for optimum design of frames (Yang and Deb [11]). The pseudo-code of the optimum design algorithm is as follows (Kaveh and Bakhshpoori [15]):

3.3.1.1 Initialize the Cuckoo Search Algorithm Parameters

The CS parameters are set in the first step. These parameters consist of the number of nests (n), the step size parameter (α), the discovering probability (pa), and the maximum number of frame analyses as the stopping criterion.

3.3.1.2 Generate Initial Nests or Eggs of Host Birds

The initial locations of the nests are determined by the set of values randomly assigned to each decision variable as

$$\text{nest}_{i,j}^{(0)} = \text{ROUND}\left(x_{j,\min} + rand.\left(x_{j,\max} - x_{j,\min}\right)\right) \tag{3.7}$$

where $\text{nest}_{i,j}{}^{(0)}$ determines the initial value of the jth variable for the ith nest, $x_{j,\min}$ and $x_{j,\max}$ are the minimum and the maximum allowable values for the jth variable, and $rand$ is a random number in the interval [0, 1]. The rounding function is utilized due to the discrete nature of the problem.

3.3.1.3 Generate New Cuckoos by Lévy Flights

In this step, all the nests except for the best one are replaced based on quality by new cuckoo eggs produced with Lévy flights from their positions as

$$\text{nest}_i^{(t+1)} = \text{nest}_i^{(t)} + \alpha.S.\left(\text{nest}_i^{(t)} - \text{nest}_{\text{best}}^{(t)}\right).r \tag{3.8}$$

where $\text{nest}_i{}^t$ is the ith nest current position, α is the step size parameter, r is a random number from a standard normal distribution and $\text{nest}_{\text{best}}$ is the position of the best nest so far, and S is a random walk based on the Lévy flights. The Lévy flight essentially provides a random walk while the random step length is drawn from a Lévy distribution. In fact, Lévy flights have been observed among foraging patterns of albatrosses, fruit flies, and spider monkeys. One of the most efficient and yet straightforward ways of applying Lévy flights is to use the so-called Mantegna algorithm. In Mantegna algorithm, the step length S can be calculated by

$$S = \frac{u}{|v|^{1/\beta}} \tag{3.9}$$

where β is a parameter between [1, 2] interval and considered to be 1.5; u and v are drawn from normal distribution as

$$u \sim N\left(0,\ \sigma_u^2\right), \quad v \sim N\left(0, \sigma_v^2\right) \tag{3.10}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\ \sin\left(\pi\beta/2\right)}{\Gamma[(1+\beta)/2]\ \beta\ 2^{(\beta-1)/2}} \right\}^{1/\beta}, \qquad \sigma_v = 1 \tag{3.11}$$

3.3.1.4 Alien Egg Discovery

The alien egg discovery is performed for each component of each solution in terms of probability matrix such as

$$P_{ij} = \begin{cases} 1 & \text{if } rand < pa \\ 0 & \text{if } rand \geq pa \end{cases} \tag{3.12}$$

where $rand$ is a random number in [0, 1] interval and pa is the discovering probability. Existing eggs are replaced considering quality by the newly generated ones from their current positions through random walks with step size such as

$$\begin{aligned} &S = rand\,.(\text{nests}(\text{randperm1}(n),\ :) - \text{nests}(\text{randperm2}(n),\ :)) \\ &\text{nest}^{t+1} = \text{nest}^{t} + S.{*}P \end{aligned} \tag{3.13}$$

where randperm1 and randperm2 are random permutation functions used for different row permutations applied on nest matrix and P is the probability matrix.

3.3.1.5 Termination Criterion

The generating new cuckoos and discovering alien eggs steps are alternatively performed until a termination criterion is satisfied. The maximum number of analyses is considered as termination criterion of the algorithm.

3.3.2 Parallel Computing System

A visit to the neighborhood PC retail store provides ample proof that we are in the multi-core era. This created demand for software infrastructure to utilize mechanisms such as parallel computing to exploit such architectures. In this respect, the MathWorks introduced Parallel Computing Toolbox software and MATLAB® Distributed Computing Server (Luszczek [9]). Regarding that our individual designs proposed by population-based metaheuristic algorithms are evaluated independently, electing one of MATLAB's most basic programming paradigms, the parallel for loops (Luszczek [9]), makes it easy for user to handle such optimization problem.

Since the parallel computing technique enables us to perform several actions at the same time, it is needed to adjust the analysis and design assumptions for a prime model of box girder in the SAP2000 environment. Once the optimization algorithm invokes the model, a set of sections are assigned to the predefined segments. A certain feasible number of proposed solutions get invoked for analysis, and evaluating the penalized fitness value following the PARFOR conditional command the next set of agents is generated. The iteration continues until a stopping criterion is attained.

3.4 Design Example

3.4.1 A Three-Span Continuous Composite Bridge

In this section, a practical example is provided to investigate the application of the presented parallel integrated optimization approach. The example bridge deck is composed of three composite trapezoidal box girders which are continuous over three spans of the lengths 15, 34, and 21 m. Figure 3.2a and b shows the topology, support conditions, and segments of a girder and the cross section of the bridge, respectively. The girder is divided to eight pre-built segments (S_i, $i = 1, 2, \ldots, 8$) in a way to satisfy fabrication limitations and minimize material waste. Considering the concrete slab thickness as a constant value (t_c), Table 3.1 presents design variables of the problem in which the second column states different cross sections for each segment. Segments on the middle supports are shaped as non-prismatic due to the presence of large negative moments. Plate thicknesses and widths are constant along each segment; also the concrete slab thickness and the top flange width are fixed for the entire girder. Altogether this problem contains 30 design variables. The range of variables is tabulated in Table 3.2.

The optimum design problem can be expressed as follows:

$$\begin{aligned}
&\text{Considering concrete slab thickness as a constant value } (t_c): \\
&\text{find}\{X\} = [b_f, t_{f1}, t_{f2}, \ldots, t_{f8}, D_{w1}, D_{w2}, \ldots, D_{w5}, t_{w1}, t_{w2}, \ldots, t_{w8}, t_{b1}, t_{b2}, \ldots, t_{b8}]_{1\times 30} \\
&\text{to minimize } W(\{X\}) \\
&\text{Subject to: } g_1, g_2, g_3, \ldots, g_9
\end{aligned} \tag{3.14}$$

where $\{X\}$ is the set of design variables and its components are sized from the discrete sets presented in Table 3.2 and $W(\{X\})$ is the self-weight of girder obtained by SAP2000. Optimum design of composite steel-box girders is one of those issues for which the conventional objective function is not applicable. Considering concrete slab,

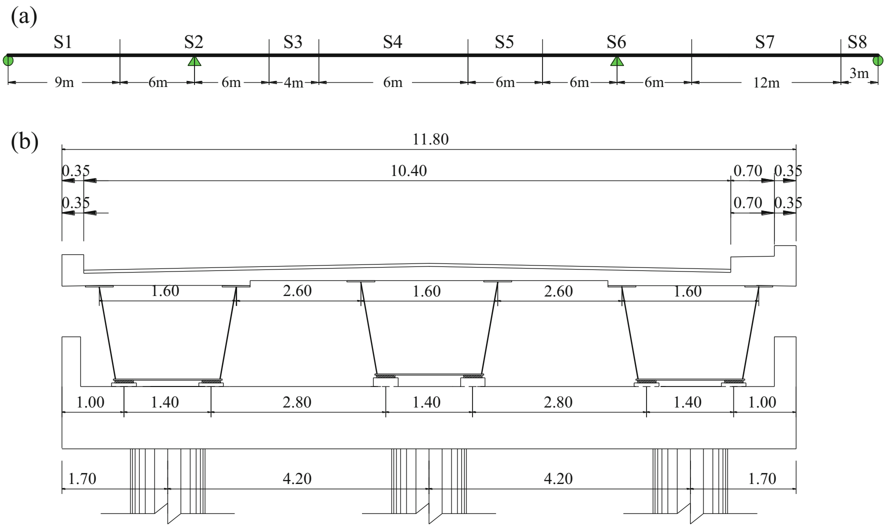


Fig. 3.2 The practical design example. (**a**) Longitudinal view and (**b**) transverse view

Table 3.1 Segments and related variables

Segment	Section	t_c	b_f	t_f	D_w	t_w	t_b
S1	A1	t_c	b_{f1}	t_{f1}	D_{w1}	t_{w1}	t_{b1}
S2	A2	t_c	b_{f1}	t_{f2}	D_{w1}	t_{w2}	t_{b2}
	A3	t_c	b_{f1}	t_{f2}	D_{w2}	t_{w2}	t_{b2}
	A4	t_c	b_{f1}	t_{f2}	D_{w3}	t_{w2}	t_{b2}
S3	A5	t_c	b_{f1}	t_{f3}	D_{w3}	t_{w3}	t_{b3}
S4	A6	t_c	b_{f1}	t_{f4}	D_{w3}	t_{w4}	t_{b4}
S5	A7	t_c	b_{f1}	t_{f5}	D_{w3}	t_{w5}	t_{b5}
S6	A8	t_c	b_{f1}	t_{f6}	D_{w3}	t_{w6}	t_{b6}
	A9	t_c	b_{f1}	t_{f6}	D_{w4}	t_{w6}	t_{b6}
	A10	t_c	b_{f1}	t_{f6}	D_{w5}	t_{w6}	t_{b6}
S7	A11	t_c	b_{f1}	t_{f7}	D_{w5}	t_{w7}	t_{b7}
S8	A12	t_c	b_{f1}	t_{f8}	D_{w5}	t_{w8}	t_{b8}

Table 3.2 Design variable range

Variable	Lower bound (m)	Upper bound (m)	Increment (m)
t_c	0.20	0.35	0.05
b_f	0.25	0.8	0.05
t_f, t_w and t_b	0.01	0.05	0.005
D_w	0.5	4.6	0.1

shear connectors, and reinforcement cost seems to be necessary. Cost of the shear connectors is negligible in comparison to the overall cost. Higher strength shear connectors are considered to satisfy the complete composite action. According to

Articles 3.24.10.2 and 3.24.3.1 provided by AASHTO [18] for designing the longitudinal and transverse reinforcement, the reinforcement depends only on the slab thickness and the distance of the girders. Thus reinforcement is not considered as a design variable. Considering the concrete slab thickness as a design variable, the proposed objective function is not representative and needs to be modified. Instead of the total weight (concrete slab weight and steel section weight altogether), the sum of the total cost of the concrete material and the total cost of the steel section material should be used. Modification can be made using unit cost coefficients for each item. The choice of the unit cost parameters can influence the properties of the most cost-efficient design (Fragiadakis and Lagaros [19]). In addition slab thickness as a design variable has a profound effect on the model stiffness matrix and dead load. Considering t_c as a design variable simultaneously with design variables representing the steel section can lead the algorithm to unfeasible designs. In these regards, the CS is applied to find the optimum design considering the slab thickness as a constant value from a certain practical interval [0.2, 0.35] with 0.05 m increment to achieve the optimum thickness. The lower bound is considered according to the provisions of AASHTO [18] (Table 3.8.9.2).

The design should be carried out in such a way that the girder satisfies the strength, displacements, and geometric requirements presented in the second section. In order to handle the constraints, a penalty approach is utilized. In this method, the aim of the optimization is redefined by introducing the cost function as

$$f_{\text{cost}}(\{X\}) = (1 + \varepsilon_1 . N)^{\varepsilon_2} \times W(\{X\}) \tag{3.15}$$

where N is the constraint violation function. For generating the total penalty, each segment is divided into five equal parts, and all the constraints, g_1 to g_8, are checked for each part. In this way the constraint violation function can be obtained as follows:

$$\begin{aligned} N &= \sum_{i=1}^{8} v_i, v_i = \max(\mu_j), j = 1, 2, .., 5 \\ \mu_j &= \sum_{k=1}^{9} \max[g_k, 0] \end{aligned} \tag{3.16}$$

in which ν_i is the penalty of each segment and μ_j is the penalty value for jth part of ith segment. ε_1 and ε_2 are penalty function exponents which are selected considering the exploration and the exploitation rate of the search space. Here, ε_1 is set to unity; ε_2 is selected in a way that, in the first steps of the search process, it is equal to 1 and ultimately increased to 3.

In modeling, analysis, and design procedures, the fundamental assumptions are made to idealize the results as follows: Material property for all sections is considered as A36 steel material with weight per unit volume of $\rho = 7849$ kg/m^3 (0.2836 lb/in^3), modulus of elasticity of $E = 199{,}948$ MPa (29,000 ksi) and a yield stress of $f_y = 248.2$ MPa (36 ksi), and concrete material with the strength of $f_c' = 24$ MPa (ksi) and $\rho = 2500$ t/m^3 (lb/in^3); the spacing of transverse stiffeners is assumed 2 m and the bottom flange is longitudinally stiffened. As it was

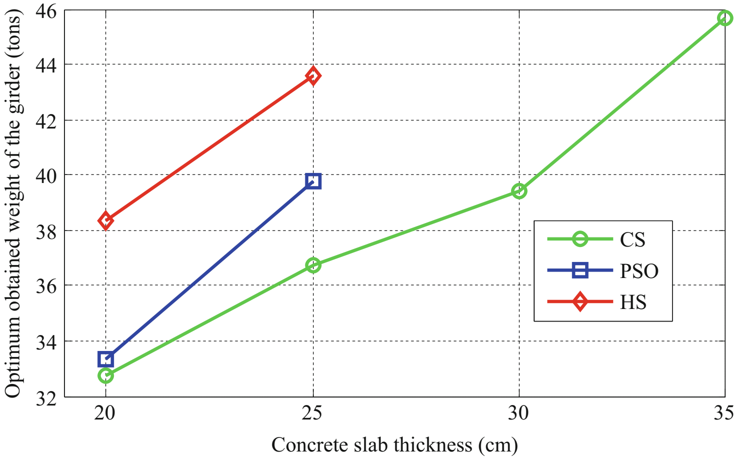


Fig. 3.3 Best results obtained by three algorithms

mentioned, the girder carries three types of loads (ton/m) as follows: $L1 = \text{slab weight} + \text{self-weight of girder}$, $L2 = 1.22$, and $L3 = 1.326$ (HS loading on a girder).

In order to verify the efficiency of the CS, two other algorithms are used to determine the solution of the considered discrete optimization problem, which are harmony search method (HS) (Kochenberger and Glover [16]) and PSO (Kennedy et al. [17]) algorithm. These algorithms have been frequently used in multicriteria and constrained optimization, typically associated with practical engineering problems. For example, Erdal et al. [20] have utilized these algorithms for optimum design of cellular beams. The author and colleagues have used these algorithms for discrete optimum design problem similar to the work by Erdal et al. [20]. Additional details can be found in Erdal et al. [20]. Here the PSO, HS, and CS algorithms are used for obtaining the optimum slab thickness and two adjacent depths. Considering the effect of the initial solution on the final results and the stochastic nature of the metaheuristic algorithms, each algorithm is independently solved for five times with random initial designs. Then the best run is chosen for performance evaluation of each technique. The maximum number of box girder evaluations are considered as 7000 for the termination criteria. The parameters of the CS algorithm are considered as $n = 7$, $\alpha = 0.1$, and $pa = 0.3$. The parameters of the PSO algorithm are tuned as $\text{NPT} = 50$, $C_1 = C_2 = 2$, $\omega = 1.2$, and $V_{\max} = \Delta t = 1.3$, and the parameters of the HS algorithm are tuned as $\text{hms} = 70$, $\text{hmcr} = 0.8$, and $\text{par} = 0.2$.

3.4.2 Discussions

Figure 3.3 shows the obtained optimum weight for various concrete slab thicknesses by the algorithms. All three algorithms result in the optimum thickness of concrete slab as 0.2 m. It can be concluded that in this test problem, considering the

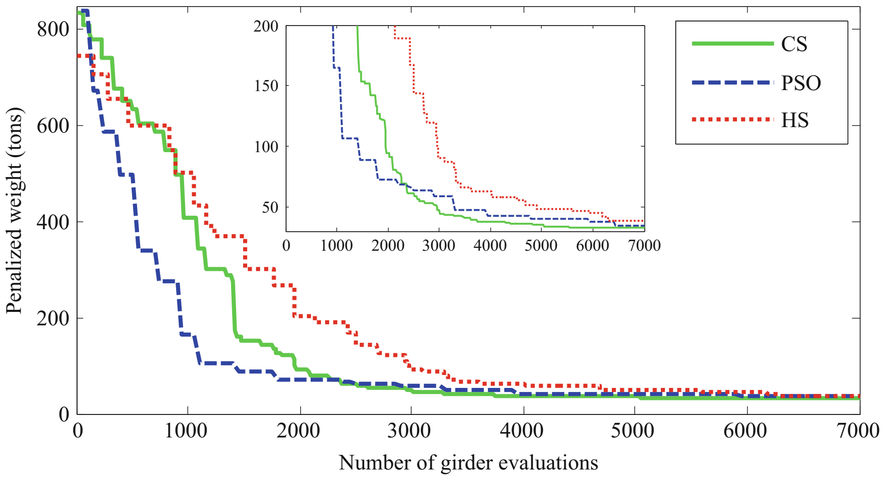


Fig. 3.4 Best convergence history obtained by three metaheuristic algorithms ($t_c = 20$ cm)

concrete slab thickness equal to the minimum value provided by the AASHTO [18] provisions leads to the optimum design. The optimum feasible designs obtained by CS, PSO, and HS algorithms weighted 32.77, 33.34, and 38.36 t, respectively. For graphical comparison of algorithms, the convergence histories for the best result of five independent runs in the case of $t_c = 0.2$ m are shown in Fig. 3.4. PSO and CS act far better than the HS algorithm. PSO algorithm shows the fastest convergence rate compared to other methods and this is because of the good global search ability of PSO. It is obvious that PSO cannot perform efficiently in the local search stage of the algorithm. However, PSO results in the same practical design as the CS but needs higher number of girder evaluations (6450). Continuous step like movements of the CS algorithm demonstrates its ability in balancing the global and local search in this optimization test problem. The optimum design obtained by cuckoo search algorithm is weighted 32.77 t which is approximately 15 % lighter than the conventional design. Related cross-sectional properties and mass per length of sections for each segment are summarized in Table 3.3. The cross-sectional properties based on the conventional design, considering the concrete slab thickness equal to 0.2 cm, are also presented in this table.

Geometry constraint values of sections for each segment are listed in Table 3.4. As it can be seen, the first constraint (g_1) with the aim of controlling the top flange thickness to the web thickness is the most active limitation. The last row exhibits optimum design controlling priority with respect to the geometry constraints. The serviceability and strength performance of the resulted optimum girder are illustrated in Fig. 3.5. Based on this figure, in spite of relatively long middle span, the effect of deflection constraint is not notable here. Such a performance is also observed for shear stress ratio constraint. Figure 3.5c shows the available and allowable flexural stress ratios for the top and bottom flanges and the concrete deck. It can be observed that the stress ratio of top and bottom flanges have more

Table 3.3 Sectional designations of the best optimum design obtained by the CS

Segment	Section	b_f	t_f	D_w	t_w	t_b	Mass per length (kg/m)
S1	A1	0.3 (0.45)	0.02 (0.025)	0.7 (1.5)	0.01 (0.015)	0.015 (0.025)	363.83 (741.55)
S2	A2	0.3 (0.45)	0.02 (0.025)	0.7 (1.5)	0.01 (0.015)	0.025 (0.025)	470.33 (741.55)
	A3	0.3 (0.45)	0.02 (0.03)	1.8 (2)	0.01 (0.01)	0.025 (0.025)	568.05 (703.55)
	A4	0.3 (0.45)	0.02 (0.02)	1.7 (1.5)	0.01 (0.01)	0.025 (0.02)	559.16 (588.49)
S3	A5	0.3 (0.45)	0.015 (0.02)	1.7 (1.5)	0.01 (0.01)	0.01 (0.02)	416.75 (546.14)
S4	A6	0.3 (0.45)	0.02 (0.02)	1.7 (1.5)	0.01 (0.01)	0.025 (0.02)	559.16 (546.14)
S5	A7	0.3 (0.45)	0.02 (0.02)	1.7 (1.5)	0.01 (0.01)	0.015 (0.02)	479.92 (546.14)
S6	A8	0.3 (0.45)	0.02 (0.02)	1.7 (1.5)	0.01 (0.01)	0.02 (0.02)	519.54 (546.14)
	A9	0.3 (0.45)	0.02 (0.03)	2.0 (2)	0.01 (0.01)	0.02 (0.025)	550.28 (703.55)
	A10	0.3 (0.45)	0.02 (0.025)	0.8 (1.5)	0.01 (0.015)	0.02 (0.025)	427.33 (741.55)
S7	A11	0.3 (0.45)	0.015 (0.025)	0.8 (1.5)	0.01 (0.015)	0.015 (0.025)	351.89 (741.55)
S8	A12	0.3 (0.45)	0.02 (0.02)	0.8 (1.5)	0.01 (0.01)	0.01 (0.02)	323.55 (546.14)

The values in parentheses are the at hand design using the conventional design procedure

Table 3.4 Geometry constraint value of each section for optimum design obtained by the CS

Segment	Section	g_1	g_2	g_3	g_4
S1	A1	0.750	0.350	0.652	0.214
S2	A2	0.750	0.350	0.652	0.214
	A3	0.750	0.900	0.652	0.550
	A4	0.750	0.850	0.652	0.520
S3	A5	1.000	0.850	0.870	0.520
S4	A6	0.750	0.850	0.652	0.520
S5	A7	0.750	0.850	0.652	0.520
S6	A8	0.750	0.850	0.652	0.520
	A9	0.750	1.000	0.652	0.612
	A10	0.750	0.400	0.652	0.245
S7	A11	1.000	0.400	0.870	0.245
S8	A12	0.750	0.400	0.652	0.245
Min		0.750	0.350	0.652	0.214
Max		1.000	1.000	0.870	0.612
Average		0.792	0.671	0.688	0.410
SD		0.097	0.261	0.085	0.159
CP		1	3	2	4

CP optimum design controlling priority with respect to geometry constraints

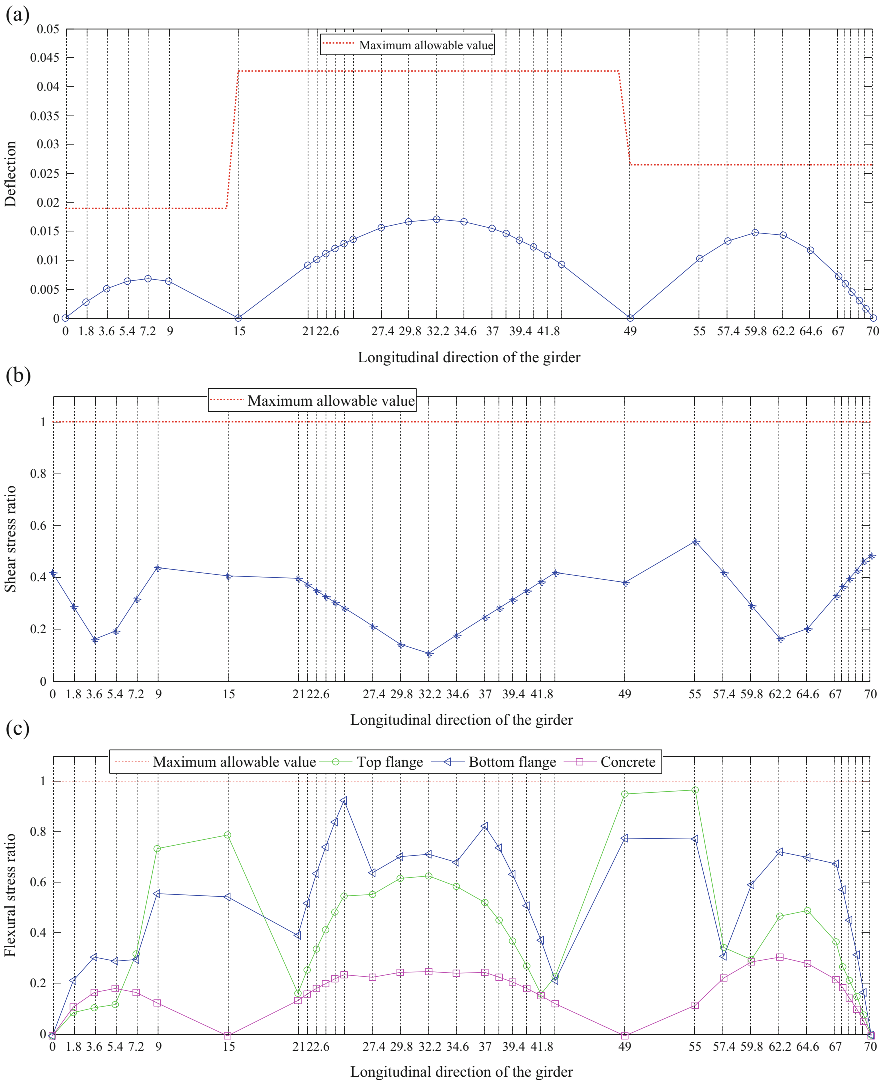


Fig. 3.5 Performance evaluation of the best achieved optimum design via CS. (**a**) Deflection; (**b**) shear stress ratio; (**c**) flexural stress ratios

effect in controlling the optimum design than the shear and concrete slab stress ratios. Also it can be interpreted that the bottom and the top flange stress ratios are dominant at the middle of spans and on the supports, respectively. This can be due to the contribution of the concrete slab in carrying the loads in a composite manner at the middle of spans.

3.5 Concluding Remarks

In this study, size optimization of composite continuous multi-span steel-box girders is performed based on AASHTO code of practice for loading and designing of bridges. The metaheuristic algorithm of choice is the cuckoo search algorithm. This algorithm optimizes the self-weight of a girder by interfacing SAP2000 and MATLAB software in the form of parallel computing. In order to verify the efficiency of the CS, two other algorithms consisting of the PSO and HS are used to determine the solution of the considered discrete optimization problem.

The results of this study reveal that the cuckoo search has a good ability in finding acceptable 3 feasible designs in terms of accuracy and convergence rate. In the case of size optimization of a box girder with 30 design variables and conditions similar to practical design, the integrated parallel metaheuristic based optimization procedure resulted in around 15 % reduction of weight compared to the conventional non-optimized design. The dominance of the constraints in controlling the final optimized results is also investigated. Despite a relatively long middle span, the effect of deflection constraint has not been notable here. Based on the present study, it can be concluded that the geometry, the top and bottom flange flexural strength, the middle span deflection, and the shear and concrete slab strength constraints are effective in optimum design of a typical multi-span continuous straight steel-box girders.

References

1. Kaveh A, Bakhshpoori T, Barkhori M (2014) Optimum design of multi-span composite box girder bridges using Cuckoo Search algorithm. Steel Compos Struct Techno Press 17 (5):705–719
2. Chen YS, Yen BT (1980) Analysis of composite box girders. Fritz Laboratory Reports. Report No. 380.12(80)
3. Rana S, Islam N, Ahsan R, Ghani SN (2013) Application of evolutionary operation to the minimum cost design of continuous prestressed concrete bridge structure. Eng Struct 46:38–48
4. Ding Y, Jiang K, Shao F, Deng A (2013) Experimental study on ultimate torsional strength of PC composite box-girder with corrugated steel webs under pure torsion. Struct Eng Mech Int J 46(4):519–531
5. Ko H-J, Moon J, Shin Y-W, Lee H-E (2013) Non-linear analyses model for composite box-girders with corrugated steel webs under torsion. Steel Compos Struct Int J 14(5):409–429
6. Lee KS, Geem W (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–798
7. Saka MP (2009) Optimum design of steel sway frames to BS5950 using harmony search algorithm. J Constr Steel Res 65(1):36–43
8. Razaqpur AG, Li HG (1991) Thin walled multi-cell box girder finite element. J Struct Eng 117:2953–2971
9. Luszczek P (2009) Parallel programming in MATLAB. Int J High Perform Comput Appl 21:336–359
10. Yang XS, Deb S (2008) Nature-inspired metaheuristic algorithms. Luniver Press, UK

11. Yang XS, Deb S (2009) Engineering optimisation by cuckoo search. Int J Math Model Numer Optim 1:330–343
12. Kaveh A, Bakhshpoori T, Ashoori M (2012) An efficient optimization procedure based on cuckoo search algorithm for practical design of steel structures. Int J Optim Civil Eng 2 (1):1–14
13. Saka MP, Dogan E (2012) Design optimization of moment resisting steel frames using a cuckoo search algorithm. In Topping BHV (ed) Proceedings of the eleventh international conference on computational structures technology. Civil-Comp Press, Stirlingshire, Paper 71. doi:10.4203/ccp.99.71
14. Saka MP, Geem ZW (2013) Mathematical and metaheuristic applications in design optimization of steel frame structures: an extensive review. Math Prob Eng Article ID 271031:33
15. Kaveh A, Bakhshpoori T (2013) Optimum design of steel frames using Cuckoo Search algorithm with Lévy flights. Struct Des Tall Spec Build 22(13):1023–1036
16. Kochenberger GA, Glover F (2003) Handbook of metaheuristics. Kluwer, Boston, MA
17. Kennedy J, Eberhart R, Shi Y (2001) Swarm intelligence. Morgan Kaufmann, San Francisco, CA
18. American Association of State Highway and Transportation Officials (AASHTO) (2002) Standard specifications for highway bridges, 17th edn. AASHTO, Washington, DC
19. Fragiadakis M, Lagaros ND (2011) An overview to structural seismic design optimisation frameworks. Comput Struct 89:1155–1165
20. Erdal F, Doğan E, Saka MP (2011) Optimum design of cellular beams using harmony search and particle swarm optimizers. J Constr Steel Res 67:237–247

Chapter 4
Sizing Optimization of Skeletal Structures Using the Enhanced Whale Optimization Algorithm

4.1 Introduction

The *whale optimization algorithm* (WOA) is a recently developed swarm-based optimization algorithm inspired by the hunting behavior of humpback whales. This chapter attempts to enhance the original formulation of the WOA in order to improve solution accuracy, reliability, and convergence speed. The new method, called *enhanced whale optimization algorithm* (EWOA), is tested in the sizing optimization of skeletal structures. In this chapter, EWOA is also compared with WOA and other metaheuristic methods developed in the literature using four skeletal structure optimization problems. Numerical results compare the efficiency of the WOA and EWOA with the latter algorithm being superior to the standard implementation [1].

In this chapter, a new nature-inspired metaheuristic optimization algorithm, called WOA, is utilized in sizing optimization of skeletal structures. This method is introduced by Mirjalili and Lewis [2], and it is inspired by the bubble-net hunting strategy of humpback whales. WOA simulates hunting behavior with random or the best search agent to chase the prey and the use of a spiral to simulate bubble-net attacking mechanism of humpback whales. Here, the original formulation of WOA is modified in order to improve its convergence behavior. The new algorithm, named EWOA, is tested in four structural optimization problems: two truss optimization problems (spatial 72-bar truss and spatial 582-bar tower) and two frame optimization problems (3-bay 15-story frame and 3-bay 24-story frame). The four test problems are solved with both EWOA and WOA, and optimization results are compared with the literature.

The remainder of the chapter is organized as follows: The mathematical model of structural optimization is presented in Sect. 4.2. Section 4.3 describes the EWOA algorithm together with a brief introduction to the basic WOA. In order to show the capability of the proposed algorithms, four numerical examples are studied in Sect. 4.4. Finally, some conclusions are derived in Sect. 4.5.

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_4

4.2 Statement of the Optimization Problem

Sizing optimization of skeletal structures can be stated as follows:

$$\begin{array}{ll} \text{Find} & \{X\} = \left[x_1, x_2, .., x_{ng}\right] \\ \text{to minimize} & W(\{X\}) = \sum_{i=1}^{nm} \rho_i A_i L_i \\ \text{subjected to :} & \begin{cases} g_j(\{X\}) \leq 0, & j = 1, 2, \ldots, nc \\ \mathrm{x_{i\ min}} \leq \mathrm{x_i} \leq x_{i\max} \end{cases} \end{array} \tag{4.1}$$

where $\{X\}$ is the vector containing the design variables; ng is the number of design variables; $W(\{X\})$ is the weight of the structure; nm is the number of elements of the structure; ρ_i, A_i, and L_i denote the material density, cross-sectional area, and the length of the ith member, respectively; $x_{i\min}$ and $x_{i\max}$ are the lower and upper bounds of the design variable x_i, respectively; $g_j(\{X\})$ denotes design constraints; and nc is the number of constraints.

To handle the constraints, the well-known penalty approach is employed. Thus, the objective function is redefined as follows:

$$f(\{X\}) = (1 + \varepsilon_1 . \upsilon)^{\varepsilon_2} \times W(\{X\}), \quad \upsilon = \sum_{j=1}^{nc} \max\left[0, g_j(\{X\})\right] \tag{4.2}$$

where υ denotes the sum of the violations of the design constraints. The constant ε_1 is set equal to 1 while ε_2 starts from 15 and then linearly increases to 3.

4.3 Optimization Algorithms

4.3.1 Whale Optimization Algorithm

A recent addition to metaheuristic algorithms is the WOA, which was introduced by Mirjalili and Lewis [2]. The WOA is inspired by the humpback whale hunting method that is called bubble-net hunting strategy. They prefer to hunt school of krill or small fishes close to the surface. Therefore, humpback whales swim around the prey within a shrinking circle and along a spiral-shaped path simultaneously to create distinctive bubbles along a circle or "9"-shaped path. To simulate this behavior in WOA, there is a probability of 50 % to choose between the encircling mechanism and the spiral model to update the position of whales during optimization. Their formulations are designed as follows:

1. Shrinking encircling preys: In WOA, the currently best candidate solution is assumed as the target prey, and the other search agents try to update their positions toward it. This behavior is represented by the following formula:

$$\vec{X}(t+1) = \vec{X}^*(t) - A.\vec{D} \tag{4.3}$$

$$\vec{D} = \left|C.\vec{X}^*(t) - \vec{X}(t)\right| \tag{4.4}$$

$$A = 2.a.r - a \tag{4.5}$$

$$C = 2.r \tag{4.6}$$

where $\vec{X}^*$ is the historically best position, $\vec{X}$ is a whale position and t indicates the current iteration, a is linearly decreased from 2 to 0 over the course of iterations, and r is a random number uniformly distributed in the range of [0,1]. The sign "||" denotes the absolute value.

2. Spiral bubble-net feeding maneuver: A spiral equation is used between the position of whale and prey to mimic the helix-shaped movement of humpback whales as follows:

$$\vec{X}(t+1) = e^{bk}.\cos(2\pi k).\vec{D}' + \vec{X}^*(t) \tag{4.7}$$

$$\vec{D}' = \left|\vec{X}^*(t) - \vec{X}(t)\right| \tag{4.8}$$

where b is a constant for defining the shape of the logarithmic spiral and k is a random number uniformly distributed in the range of [−1,1].
In order to have a global optimizer, when A is >1 or A <−1, the search agent is updated according to a randomly chosen search agent instead of the best search agent:

$$\vec{X}(t+1) = \vec{X}_{rand} - A.\vec{D}'' \tag{4.9}$$

$$\vec{D}'' = \left|C.\vec{X}_{rand} - \vec{X}(t)\right| \tag{4.10}$$

where $\vec{X}_{rand}$ is selected randomly from whales in the current iteration. For further details, the reader may refer to Mirjalili and Lewis [2].

4.3.2 Enhanced Whale Optimization Algorithm

The WOA is simple in concept and effective to explore global solutions. In order to improve the solution accuracy, reliability of search, and convergence speed of

WOA, a new algorithm is introduced in this chapter, which is called the EWOA. A key point in improving an algorithm is to preserve the simplicity of the original method.

A random number in the [0,1] range is extracted for each whale in each iteration. If it is >0.5, Eq. (4.7) is selected; otherwise, Eq. (4.12) is chosen for updating whale's position.

In exploration phase of EWOA, one component of each whale is changed with the random value in the search space with a probability like p instead of Eq. (4.9).

$$p = 0.3(1 - iter/iter_{max}) \tag{4.11}$$

where $iter$ and $iter_{max}$ are current iteration number and the total number of the iterations for optimization process, respectively.

For a selected whale, an integer random number is extracted in the interval [1, ng] to choose which design variable should be randomly changed. At this point, another random number q is extracted in the interval [0,1] and compared with the probability threshold p. The selected variable x_j is changed if $q < p$, according to $x_j = x_{j\min} + random.\left(x_{j\max} - x_{j\min}\right)$, where $random$ is a random number uniformly distributed in the interval [0, 1].

The modified algorithm should be capable of maintaining proper balance between the diversification and the intensification inclinations. According to this point and the above change, Eq. (4.3) is redefined as follows:

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A}\circ\vec{D}''' \tag{4.12}$$

$$\vec{D}''' = \vec{r}\circ\left|\vec{X}(t)\right| \tag{4.13}$$

$$\vec{A} = 2.\vec{a}\circ\vec{r} - \vec{a} \tag{4.14}$$

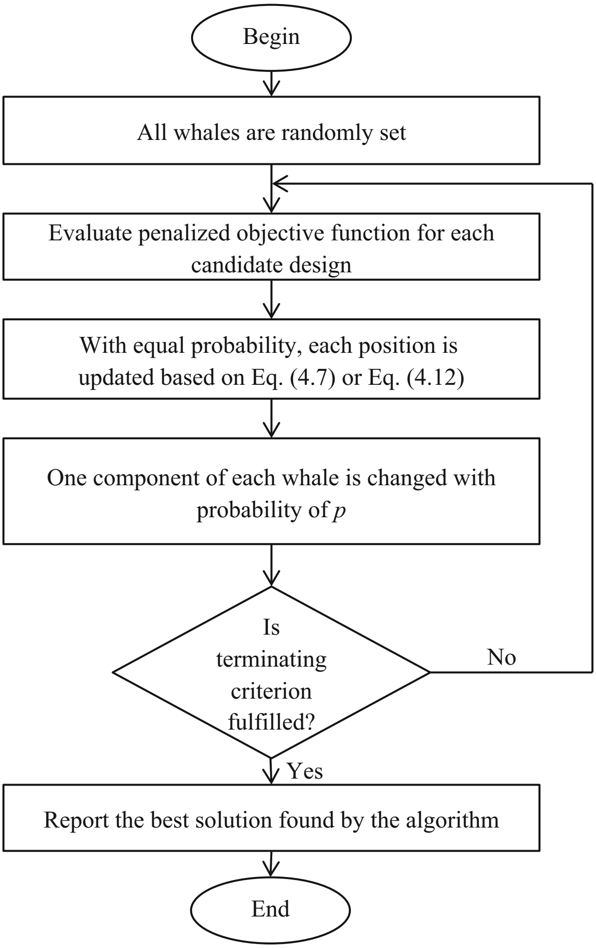
where $\vec{r}$ is a random vector that has each component uniformly distributed in the range of [0,1] and $\vec{a}$ is a vector that has each component equal to a. The sign "∘" denotes an element-by-element multiplication.

Flowchart of EWOA is shown in Fig. 4.1.

4.4 Test Problems and Optimization Results

In this section, four benchmark examples are provided to demonstrate the effectiveness, robustness, and efficiency of the WOA and EWOA. In order to reduce statistical errors, each test is repeated 20 times independently. In all problems, agents are allowed to select discrete values from the permissible list of cross sections (real numbers are rounded to the nearest integer in the each iteration). The algorithms are coded in MATLAB, and the structures are analyzed using the direct stiffness method by our own codes.

Fig. 4.1 Flowchart of the EWOA algorithm [1]



4.4.1 Spatial 72-Bar Truss Problem

Figure 4.2 shows the schematic of a spatial 72-bar truss structure. The material density is 0.1 lb/in^3 (2,767,990 kg/m^3) and the modulus of elasticity is 10^7 psi (6895 GPa). The elements are divided into 16 groups, because of structural symmetry: (1) A_1–A_4, (2) A_5–A_{12}, (3) A_{13}–A_{16}, (4) A_{17}–A_{18}, (5) A_{19}– A_{22}, (6) A_{23}–A_{30}, (7) A_{31}–A_{34}, (8) A_{35}–A_{36}, (9) A_{37}–A_{40}, (10) A_{41}–A_{48}, (11) A_{49}–A_{52}, (12) A_{53}–A_{54}, (13) A_{55}–A_{58}, (14) A_{59}–A_{66} (15), A_{67}– A_{70}, and (16) A_{71}–$A_{72.}$ The structure is subject to the two independent loading conditions listed in Table 4.1. The maximum stress developed in the elements must be less than ±25 ksi (±172,375 MPa). Maximum displacement of the uppermost nodes cannot exceed ±0.25 in (±635 mm), for each node, in all directions. In this case, the discrete sizing

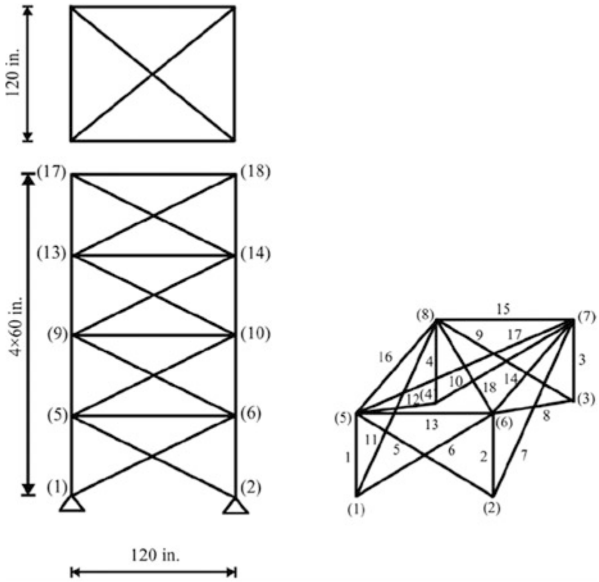


Fig. 4.2 Schematic of the spatial 72-bar truss structure

variables can be selected from a list of 64 discrete sections from 0.111 to 335 in^2 (71,613–21,612,860 mm^2) (Kaveh and Ilchi Ghazaan [3]).

This example is also used for adjusting *b* [a constant for defining the shape of the logarithmic spiral in Eq. (4.7)], number of whales, and $iter_{max}$ (total number of iterations). In order to adjust the value of *b*, a number of whales and $iter_{max}$ are, respectively, set to 20 and 1000, and different amounts of b are considered as 0.5, 1, 15, and 2. The results shown in Table 4.2 demonstrate that the algorithm is not very sensitive to the values of *b*; however, statistical results indicate that 0.5 is the most efficient value. In order to adjust the number of whales, the value of iter_{max} is set to 1000, and various numbers of whales are selected as 10, 20, 30, and 40. Comparison of the results is shown in Table 4.3, and it can be seen that 20 is a quite suitable number. Different $iter_{max}$ are tested (500, 750, 1000, 1250, and 1500) to adjust this variable. Table 4.4 summarizes the results and it can be concluded 1000 is the most suitable value for $iter_{max}$.

Table 4.5 represents the results obtained by different optimization algorithms. The lightest designs obtained by discrete heuristic particle swarm ant colony optimization (DHPSACO) (Kaveh and Talatahari [4]), imperialist competitive algorithm (ICA) (Kaveh and Talatahari [5]), and colliding bodies optimization (CBO) (Kaveh and Ilchi Ghazaan [3]) are 393,380 lb, 39,284 lb, and 39,123 lb, respectively. The best designs of improved ray optimization (IRO) (Kaveh et al. [6]), adaptive elitist differential evolution (aeDE) (Ho-Huu et al. [7]), WOA, and EWOA are identical (i.e., 38,933 lb). EWOA was the most robust optimizer, achieving the lowest average weight over the independent optimization runs. Figure 4.3 shows the convergence curves of the best and average results obtained

Table 4.1 Loading conditions for the spatial 72-bar truss problem

	Condition 1			Condition 2		
Node	F_x kips (kN)	F_y kips (kN)	F_z kips (kN)	F_x kips (kN)	F_y kips (kN)	F_z kips (kN)
17	0.0	0.0	−5.0 (−22.25)	−5.0 (−22.25)	5.0 (−22.25)	−5.0 (−22.25)
18	0.0	0.0	−5.0 (−22.25)	0.0	0.0	0.0
19	0.0	0.0	−5.0 (−22.25)	0.0	0.0	0.0
20	0.0	0.0	−5.0 (−22.25)	0.0	0.0	0.0

Table 4.2 Sensitivity of EWOA to the *b* parameter studied for the 72-bar truss problem

	Results		
b	Weight (lb)	Average optimized weight (lb)	Standard deviation on average weight (lb)
0.5	389.33	389.64	0.74
1	389.33	389.98	1.58
1.5	389.33	389.89	1.29
2	389.33	389.81	0.78

Table 4.3 Sensitivity of EWOA to the number of whales studied for the 72-bar truss problem

	Results		
Number of whales	Weight (lb)	Average optimized weight (lb)	Standard deviation on average weight (lb)
10	389.33	390.03	1.36
20	389.33	389.64	0.74
30	389.33	389.73	0.71
40	389.33	389.86	0.97

Table 4.4 Sensitivity of EWOA to the $iter_{max}$ parameter studied for the 72-bar truss problem

	Results		
$iter_{max}$	Weight (lb)	Average optimized weight (lb)	Standard deviation on average weight (lb)
500	389.33	390.28	1.89
750	389.33	390.49	1.53
1000	389.33	389.64	0.74
1250	389.33	389.90	0.95
1500	389.33	389.93	1.33

Table 4.5 Optimized designs found by different algorithms in the 72-bar truss problem

Element group	Members	Cross-sectional areas (in^2)						
		DHPSACO (Kaveh and Talatahari [4])	ICA (Kaveh and Talatahari [5])	IRO (Kaveh et al. [6])	CBO (Kaveh and Ilchi Ghazaan [3])	aeDE (Ho-Huu et al. [7])	Kaveh and Ilchi Ghazaan [1]	
							WOA	EWOA
1	A_1–A_4	1.800	1.99	1.99	2.13	1.99	1.99	1.99
2	A_5–A_{12}	0.442	0.442	0.563	0.563	0.563	0.563	0.563
3	A_{13}–A_{16}	0.141	0.111	0.111	0.111	0.111	0.111	0.111
4	A_{17}–A_{18}	0.111	0.141	0.111	0.111	0.111	0.111	0.111
5	A_{19}–A_{22}	1.228	1.228	1.228	1.228	1.228	1.228	1.228
6	A_{23}–A_{30}	0.563	0.602	0.563	0.442	0.442	0.442	0.442
7	A_{31}–A_{34}	0.111	0.111	0.111	0.141	0.111	0.111	0.111
8	A_{35}–A_{36}	0.111	0.141	0.111	0.111	0.111	0.111	0.111
9	A_{37}–A_{40}	0.563	0.563	0.563	0.442	0.563	0.563	0.563
10	A_{41}–A_{48}	0.563	0.563	0.442	0.563	0.563	0.563	0.563
11	A_{49}–A_{52}	0.111	0.111	0.111	0.111	0.111	0.111	0.111
12	A_{53}–A_{54}	0.250	0.111	0.111	0.111	0.111	0.111	0.111
13	A_{55}–A_{58}	0.196	0.196	0.196	0.196	0.196	0.196	0.196
14	A_{59}–A_{66}	0.563	0.563	0.563	0.563	0.563	0.563	0.563
15	A_{67}–A_{70}	0.442	0.307	0.391	0.391	0.391	0.391	0.391
16	A_{71}–A_{72}	0.563	0.602	0.563	0.563	0.563	0.563	0.563
Weight (lb)		393.380	392.84	389.33	391.23	389.33	389.33	389.33
Average optimized weight (lb)		N/A	N/A	408.17	456.69	390.913	392.52	389.64
Worst optimized weight (lb)		N/A	N/A	N/A	N/A	393.325	399.65	391.83
Number of structural analyses		5330	4500	17,925	4620	4160	6960	10,460
Constraint tolerance (%)		None	None	None	None	None	None	None

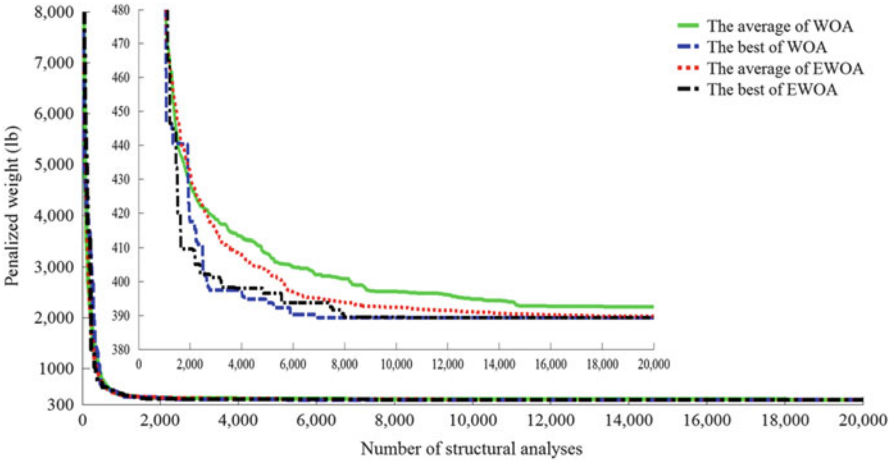


Fig. 4.3 Convergence curves obtained by EWOA and WOA in the 72-bar truss problem [1]

by WOA and EWOA. The best designs have been located at 6960 and 10,460 analyses for WOA and EWOA, respectively.

4.4.2 Spatial 582-Bar Tower Problem

The spatial 582-bar tower truss shown in Fig. 4.4 is optimized for minimum volume with the cross-sectional areas of the members being the design variables. The 582 members are divided into 32 groups, because of structural symmetry. Cross-sectional areas of elements (sizing variables) are selected from a discrete list of W-shaped standard steel sections based on area and radii of gyration properties. Cross-sectional areas of elements can vary between 616 and 215 in^2 (i.e., between 3974 and 138,709 cm^2). A single load case is considered: lateral loads of 112 kips (50 kN) applied in both x- and y-directions and a vertical load of -674 kips (-30 kN) applied in the z-direction at all nodes of the tower. Limitation on stress and stability of truss elements are imposed according to the provisions of AISC [8] as follows:

The allowable tensile stresses for tension members are calculated as

$$\sigma_i^+ = 0.6F_y \tag{4.15}$$

where F_y is the yield strength.

The allowable stress limits for compression members are calculated depending on two possible failure modes of the members known as elastic and inelastic buckling. Therefore,

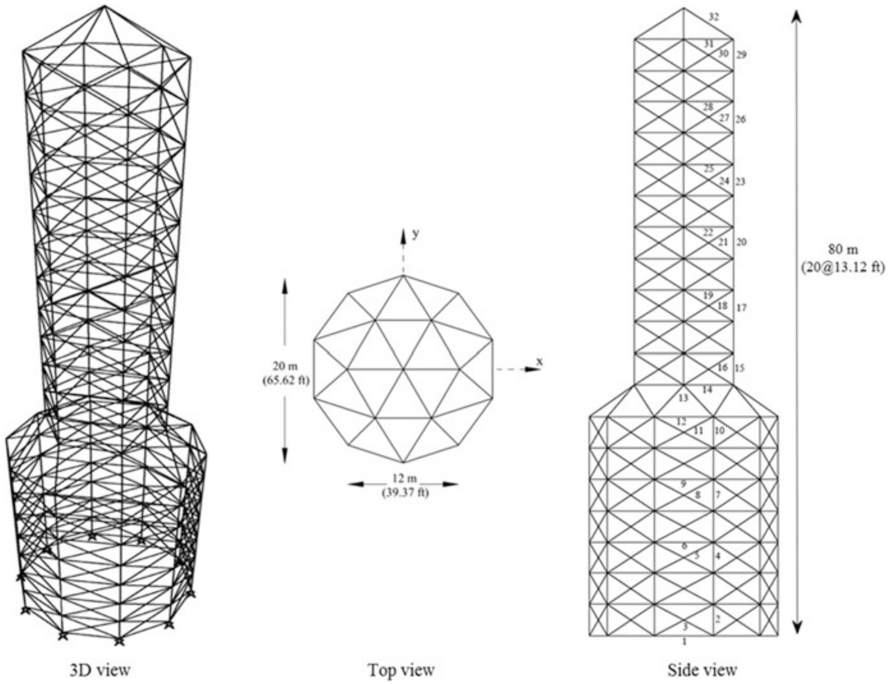


Fig. 4.4 Schematic of the spatial 582-bar tower

$$\sigma_i^- = \begin{cases} \left[\left(1 - \dfrac{\lambda_i^2}{2C_c^2}\right)F_y\right] \Big/ \left[\dfrac{5}{3} + \dfrac{3\lambda_i}{8C_c} - \dfrac{\lambda_i^3}{8C_c^3}\right] & \text{for } \lambda_i < C \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_c \end{cases} \tag{4.16}$$

where E is the modulus of elasticity; λ_i is the slenderness ratio $(\lambda_i = kl_i/r_i)$; C_c denotes the slenderness ratio dividing the elastic and inelastic buckling regions $C_c = \sqrt{\frac{2\pi^2 E}{F_y}}$; k is the effective length factor (k is set equal to 1 for all truss members); L_i is the member length; and r_i is the minimum radius of gyration.

The maximum slenderness ratio is limited to 300 for tension members, and it is recommended to be 200 for compression members. Moreover, nodal displacements in all coordinate directions must be less than ±315 in (i.e., ±8 cm) for this example.

Table 4.6 represents the results obtained by different optimization algorithms. The best design obtained by EWOA is better than other methods (1,294,929 in^3). The best volume found by PSO (particle swarm optimization) (Hasançebi et al. [9]), DHPSACO (Kaveh and Talatahari [4]), hybrid Big Bang–Big Crunch optimization (HBB–BC) (Kaveh and Talatahari [10]), CBO (Kaveh and Ilchi Ghazaan [11]), and WOA is 1,366,674 in^3, 1,346,227 in^3, 1,365,143 in^3, 1,334,994 in^3, and

Table 4.6 Optimized designs found by different algorithms in the 582-bar tower problem

Element group	Members	Cross-sectional areas (in^2)						
		DHPSACO (Kaveh and Talatahari [4])	ICA (Kaveh and Talatahari [5])	IRO (Kaveh et al. [6])	CBO (Kaveh and Ilchi Ghazaan [3])	aeDE (Ho-Huu et al. [7])	Kaveh and Ilchi Ghazaan [1]	
							WOA	EWOA
1	A_1–A_4	1.800	1.99	1.99	2.13	1.99	1.99	1.99
2	A_5–A_{12}	0.442	0.442	0.563	0.563	0.563	0.563	0.563
3	A_{13}–A_{16}	0.141	0.111	0.111	0.111	0.111	0.111	0.111
4	A_{17}–A_{18}	0.111	0.141	0.111	0.111	0.111	0.111	0.111
5	A_{19}–A_{22}	1.228	1.228	1.228	1.228	1.228	1.228	1.228
6	A_{23}–A_{30}	0.563	0.602	0.563	0.442	0.442	0.442	0.442
7	A_{31}–A_{34}	0.111	0.111	0.111	0.141	0.111	0.111	0.111
8	A_{35}–A_{36}	0.111	0.141	0.111	0.111	0.111	0.111	0.111
9	A_{37}–A_{40}	0.563	0.563	0.563	0.442	0.563	0.563	0.563
10	A_{41}–A_{48}	0.563	0.563	0.442	0.563	0.563	0.563	0.563
11	A_{49}–A_{52}	0.111	0.111	0.111	0.111	0.111	0.111	0.111
12	A_{53}–A_{54}	0.250	0.111	0.111	0.111	0.111	0.111	0.111
13	A_{55}–A_{58}	0.196	0.196	0.196	0.196	0.196	0.196	0.196
14	A_{59}–A_{66}	0.563	0.563	0.563	0.563	0.563	0.563	0.563
15	A_{67}–A_{70}	0.442	0.307	0.391	0.391	0.391	0.391	0.391
16	A_{71}–A_{72}	0.563	0.602	0.563	0.563	0.563	0.563	0.563
Weight (lb)		393.380	392.84	389.33	391.23	389.33	389.33	389.33
Average optimized weight (lb)		N/A	N/A	408.17	456.69	390.913	392.52	389.64
Worst optimized weight (lb)		N/A	N/A	N/A	N/A	393.325	399.65	391.83
Number of structural analyses		5330	4500	17,925	4620	4160	6960	10,460
Constraint tolerance (%)		None	None	None	None	None	None	None

1,302,038 in^3, respectively. EWOA was again the most robust optimizer, achieving the lowest average volume over the independent optimization runs. The stress ratios evaluated for the best design optimized by WOA and EWOA are shown in Fig. 4.5. The maximum stress ratio and the maximum nodal displacements obtained by WOA are 99.87 % and 31,499 in, respectively, while 99.90 % and 31,497 in are found by EWOA for maximum stress ratio and the maximum nodal displacements. Figure 4.6 illustrates the convergence curves found by the proposed methods. The best designs

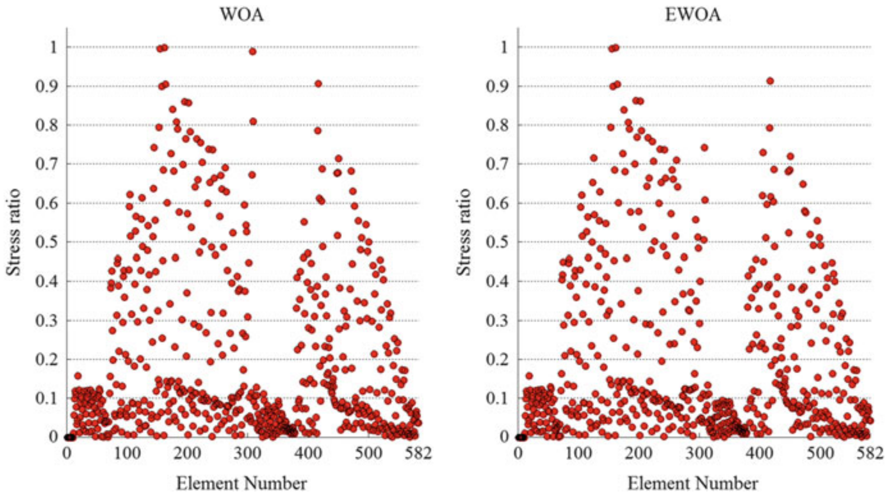


Fig. 4.5 Stress ratios evaluated at the optimized designs found by EWOA and WOA in the 582-bar tower problem

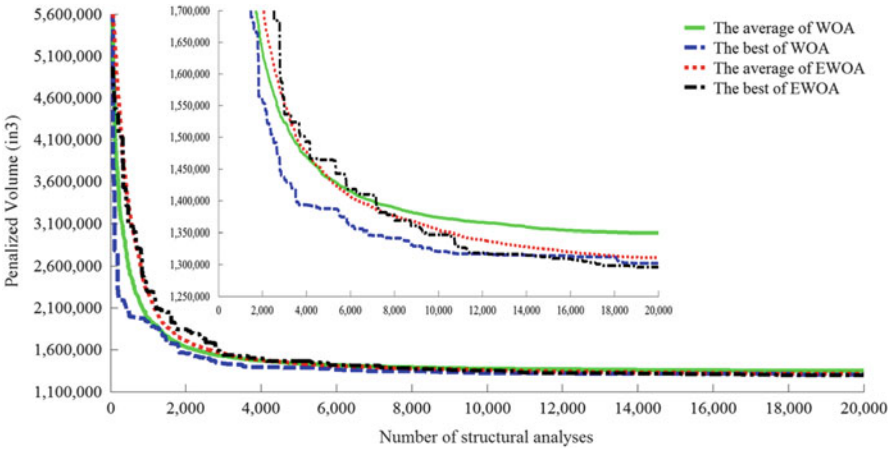


Fig. 4.6 Convergence curves obtained by EWOA and WOA in the 582-bar tower problem [1]

are achieved after 18,840 and 19,300 analyses in WOA and EWOA, respectively. However, EWOA required only about 14,000 analyses to find better intermediate designs than WOA and 17,100 analyses to find an intermediate design with volume 1,302,000 in^3, better than the WOA optimized volume (1,302,038 in^3). Furthermore, EWOA required only 11,740 analyses to find a volume of 1,330,000 in^3, better than the design optimized by CBO (1,334,994 in^3 within 17,700 analyses).

4.4.3 A 3-Bay 15-Story Frame Problem

Figure 4.7 represents the schematic of a 3-bay 15-story frame. The applied loads and the numbering of member groups are also shown in this figure. The modulus of elasticity is 29 Msi (200 GPa) and the yield stress is 36 ksi (2482 MPa). The effective length factors of the members are calculated as $k_x \geq 0$ for a sway-permitted frame, and the out-of-plane effective length factor is specified as $k_y = 10$. Each column is considered as non-braced along its length, and the non-braced length for each beam member is specified as one-fifth of the span length. Limitation on displacement and strength is imposed according to the provisions of AISC [12] as follows:

(a) Maximum lateral displacement

$$\frac{\Delta_T}{H} - R \leq 0 \tag{4.17}$$

where Δ_T is the maximum lateral displacement, H is the height of the frame structure, and R *is* the maximum drift index which is equal to 1/300.

(b) The inter-story displacements

$$\frac{d_i}{h_i} - R_I \leq 0, \quad i = 1, 2, \ldots, ns \tag{4.18}$$

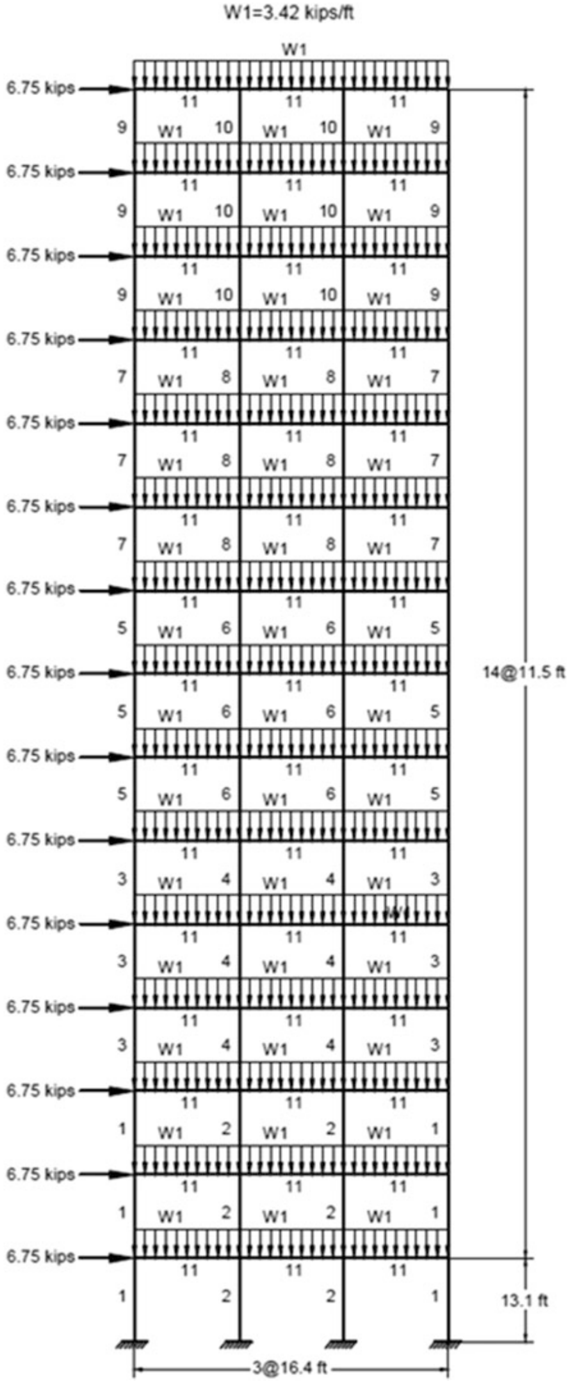
where d_i is the inter-story drift, h_i is the story height of the ith floor, ns is the total number of stories, and R_I is the inter-story drift index (1/300).

(c) Strength constraints

$$\begin{cases} \dfrac{P_{\mathrm{u}}}{2\varphi_c P_{\mathrm{n}}} + \dfrac{M_{\mathrm{u}}}{\varphi_b M_{\mathrm{n}}} - 1 \leq 0, & \text{for } \dfrac{P_{\mathrm{u}}}{\varphi_c P_{\mathrm{n}}} < 0.2 \\ \dfrac{P_{\mathrm{u}}}{\varphi_c P_{\mathrm{n}}} + \dfrac{8M_{\mathrm{u}}}{9\varphi_b M_{\mathrm{n}}} - 1 \leq 0, & \text{for } \dfrac{P_{\mathrm{u}}}{\varphi_c P_{\mathrm{n}}} \geq 0.2 \end{cases} \tag{4.19}$$

where P_{u} is the required strength (tension or compression), P_{n} is the nominal axial strength (tension or compression), φ_c is the resistance factor ($\varphi_c = 0.9$ for tension, $\varphi_c = 0.85$ for compression), M_{u} is the required flexural strengths, M_{n} is

Fig. 4.7 Schematic of the 3-bay 15-story frame



the nominal flexural strength, and φ_b denotes the flexural resistance reduction factor ($\varphi_b = 0.90$). The nominal tensile strength for yielding in the gross section is calculated by

$$P_{\mathrm{n}} = A_{\mathrm{g}} . F_{\mathrm{y}} \tag{4.20}$$

The nominal compressive strength of a member is computed as

$$P_{\mathrm{n}} = A_{\mathrm{g}} . F_{\mathrm{cr}} \tag{4.21}$$

where

$$\begin{cases} F_{\mathrm{cr}} = \left(0.658^{\lambda_c^2}\right) F_{\mathrm{y}}, & \text{for} \quad \lambda_c \leq 1.5 \\ F_{\mathrm{cr}} = \left(\dfrac{0.877}{\lambda_c^2}\right) F_{\mathrm{y}}, & \text{for} \quad \lambda_c > 1.5 \end{cases} \tag{4.22}$$

$$\lambda_c = \frac{kl}{r\pi} \sqrt{\frac{F_{\mathrm{y}}}{E}} \tag{4.23}$$

where A_{g} is the cross-sectional area of a member and k is the effective length factor that is calculated by (Dumonteil [13]):

$$k = \sqrt{\frac{1.6 G_A G_B + 4.0(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \tag{4.24}$$

where G_A and G_B are stiffness ratios of columns and girders at two end joints, A and B, of the column section being considered, respectively.
Also, the sway of the top story is limited to 925 in (235 cm) in this example.

The designs optimized by HPSACO (heuristic particle swarm ant colony optimization) (Kaveh and Talatahari [14]), HBB–BC (Kaveh and Talatahari [10]), ICA (Kaveh and Talatahari [5]), CSS (charged system search) (Kaveh and Talatahari [15]), CBO (Kaveh and Ilchi Ghazaan [3]), WOA, and EWOA are compared in Table 4.7. The EWOA algorithm obtained the lowest weight, which is 88,090 lb. EWOA was the most robust optimizer also in this test problem, obtaining the lowest average weight over the independent optimization runs. Stress ratios and inter-story drifts evaluated for the best designs of WOA and EWOA are shown in Figs. 4.8 and 4.9. Figure 4.10 compares the best and average convergence histories of EWOA and WOA. The best designs are achieved after 19,060 and 19,940 analyses in WOA and EWOA, respectively.

Table 4.7 Optimized designs found by different algorithms in the 3-bay 15-story frame problem

Element group	Optimal W-shaped sections						
	HPSACO (Kaveh and Talatahari [14])	HBB–BC (Kaveh and Talatahari [10])	ICA (Kaveh and Talatahari [5])	CSS (Kaveh and Talatahari [15])	CBO (Kaveh and Ilchi Ghazaan [3])	Kaveh and Ilchi Ghazaan [1]	
						WOA	EWOA
1	W21 × 111	W24 × 117	W24 × 117	W21 × 147	W24 × 104	W14 × 90	W14 × 99
2	W18 × 158	W21 × 132	W21 × 147	W18 × 143	W40 × 167	W30 × 173	W27 × 161
3	W10 × 88	W12 × 95	W27 × 84	W12 × 87	W27 × 84	W12 × 79	W27 × 84
4	W30 × 116	W18 × 119	W27 × 114	W30 × 108	W27 × 114	W27 × 114	W24 × 104
5	W21 × 83	W21 × 93	W14 × 74	W18 × 76	W21 × 68	W14 × 68	W21 × 68
6	W24 × 103	W18 × 97	W18 × 86	W24 × 103	W30 × 90	W30 × 90	W18 × 86
7	W21 × 55	W18 × 76	W12 × 96	W21 × 68	W8 × 48	W21 × 48	W21 × 48
8	W27 × 114	W18 × 65	W24 × 68	W14 × 61	W21 × 68	W14 × 68	W14 × 68
9	W10 × 33	W18 × 60	W10 × 39	W18 × 35	W14 × 34	W8 × 24	W8 × 31
10	W18 × 46	W10 × 39	W12 × 40	W10 × 33	W8 × 35	W14 × 48	W10 × 45
11	W21 × 44	W21 × 48	W21 × 44	W21 × 44	W21 × 50	W21 × 44	W21 × 44
Weight (lb)	95,850	97,689	93,846	92,723	93,795	88,651	88,090
Average optimized weight (lb)	N/A	N/A	N/A	N/A	98,738	92,903	90,784
Worst optimized weight (lb)	N/A	N/A	N/A	N/A	N/A	99,806	94,931
Number of structural analyses	6800	9900	6000	5000	9520	19,060	19,940
Constraint tolerance (%)	None	None	None	None	None	None	None

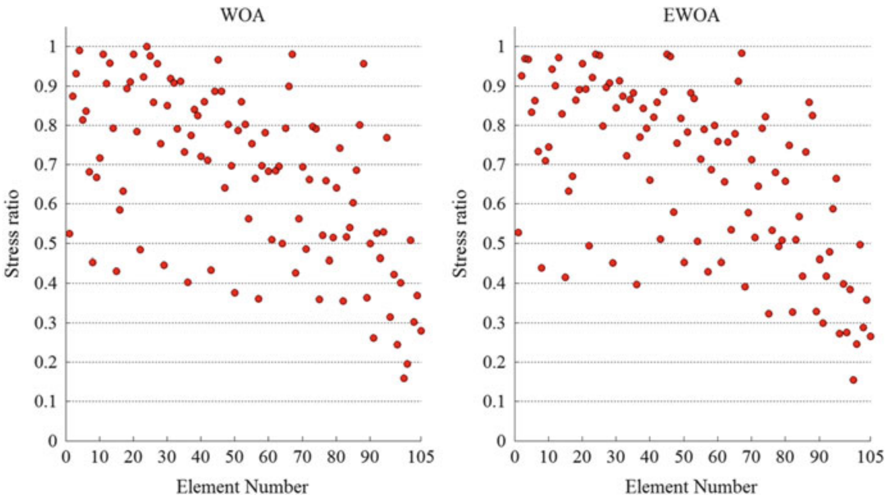


Fig. 4.8 Stress ratios evaluated at the optimized designs found by EWOA and WOA in the 3-bay 15-story frame problem

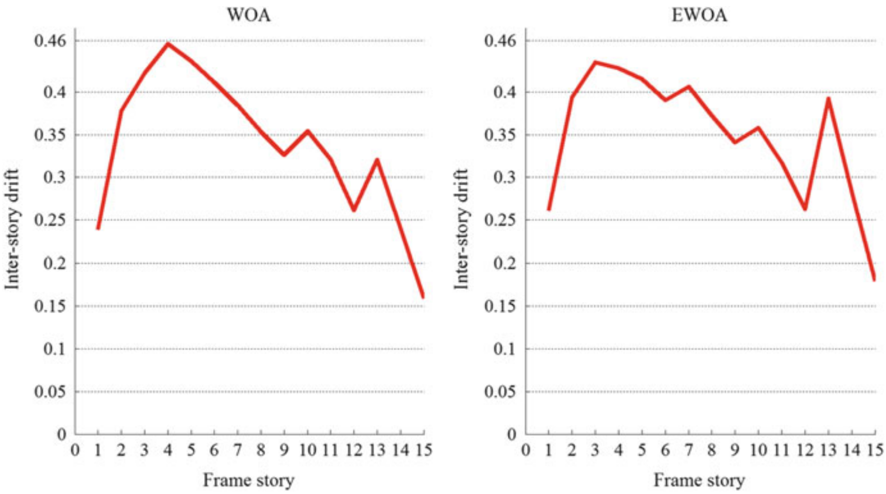


Fig. 4.9 Inter-story drifts evaluated at the optimized designs found by EWOA and WOA in the 3-bay 15-story frame problem [1]

4.4.4 A 3-Bay 24-Story Frame Problem

Figure 4.11 shows the schematic of a 3-bay 24-story frame. Frame members are collected in 20 groups (16 column groups and 4 beam groups). Each of the four beam element groups is chosen from all 267 W shapes, while the 16 column element groups are limited to W14 sections. The material has a modulus of

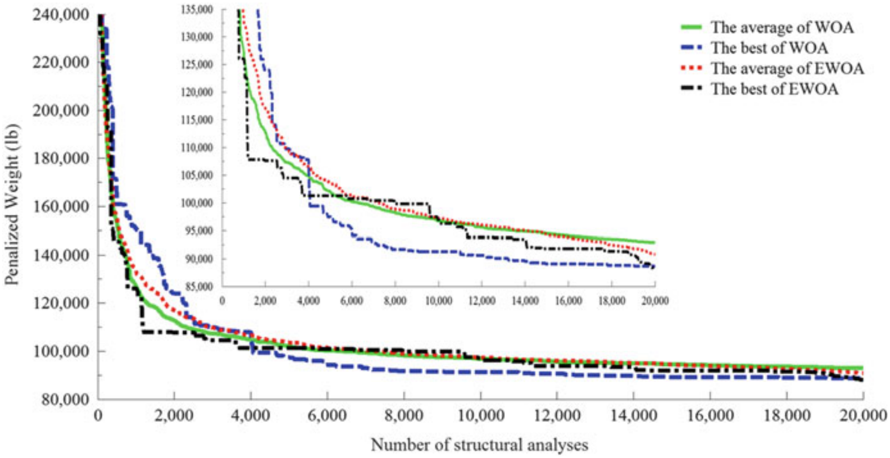


Fig. 4.10 Convergence curves obtained by EWOA and WOA in the 3-bay 15-story frame problem [1]

elasticity equal to $E = 29{,}732$ Msi (205 GPa) and a yield stress of $f_y = 334$ ksi (2303 MPa). The effective length factors of the members are calculated as $k_x \geq 0$ for a sway-permitted frame, and the out-of-plane effective length factor is specified as $k_y = 10$. All columns and beams are considered as non-braced along their lengths. Similar to the previous example, the frame is designed following the AISC-LRFD specifications and uses an inter-story drift displacement constraint (AISC [12]).

The optimized designs found by the different algorithms are compared in Table 4.8. The lightest design (i.e., 203,490 lb) is again obtained by EWOA. The best weights found by ACO (ant colony optimization) (Camp et al. [16]), HS (harmony search) (Degertekin [17]), ICA (Kaveh and Talatahari [5]), CSS (Kaveh and Talatahari [15]), CBO (Kaveh and Ilchi Ghazaan [3]), and WOA are, 220,465 lb, 214,860 lb, 212,640 lb, 212,364 lb, 215,874 lb, and 206,520 lb, respectively. The average optimized weight achieved by EWOA is better than those obtained by the other metaheuristic algorithms considered in this study. Figure 4.12 compares the convergence curves obtained by EWOA and WOA, which found the optimum weight after 18,820 and 19,640 structural analyses, respectively. It should be noted that EWOA required only 10,500 analyses to find an intermediate design weighing 210,000 lb, better than the designs optimized by ICA and CSS (212,640 and 212,364 lb, respectively), and only 13,500 analyses to find an intermediate design weighing 206,000 lighter than the WOA optimized design (206,520 lb).

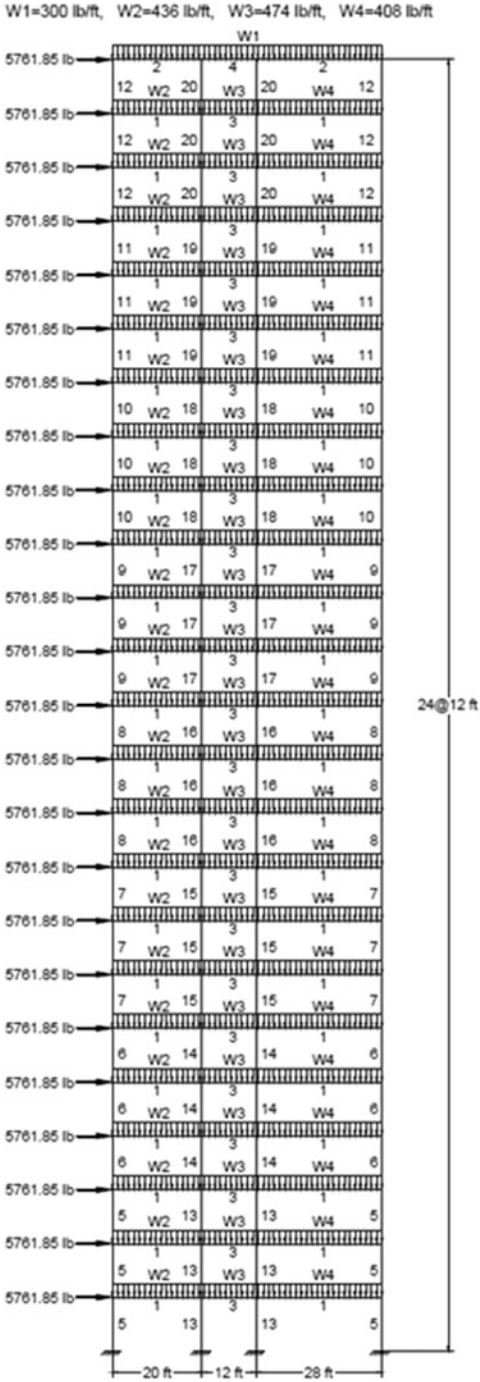


Fig. 4.11 Schematic of the 3-bay 24-story frame

Table 4.8 Optimized designs found by different algorithms in the 3-bay 24-story frame problem

	Optimal W-shaped sections						
						Kaveh and Ilchi Ghazaan [1]	
Element group	ACO (Camp et al. [16])	HS (Degertekin [17])	ICA (Kaveh and Talatahari [5])	CSS (Kaveh and Talatahari [15])	CBO (Kaveh and Ilchi Ghazaan [3])	WOA	EWOA
1	W30 × 90	W30 × 90	W30 × 90	W30 × 90	W27 × 102	W30 × 90	W30 × 90
2	W8 × 18	W10 × 22	W21 × 50	W21 × 50	W8 × 18	W10 × 17	W10 × 30
3	W24 × 55	W18 × 40	W24 × 55	W21 × 48	W24 × 55	W21 × 62	W24 × 55
4	W8 × 21	W12 × 16	W8 × 28	W12 × 19	W6 × 8.5	W14 × 26	W6 × 8.5
5	W14 × 145	W14 × 176	W14 × 109	W14 × 176	W14 × 132	W14 × 109	W14 × 159
6	W14 × 132	W14 × 176	W14 × 159	W14 × 145	W14 × 120	W14 × 145	W14 × 99
7	W14 × 132	W14 × 132	W14 × 120	W14 × 109	W14 × 145	W14 × 109	W14 × 120
8	W14 × 132	W14 × 109	W14 × 90	W14 × 90	W14 × 82	W14 × 99	W14 × 74
9	W14 × 68	W14 × 82	W14 × 74	W14 × 74	W14 × 61	W14 × 53	W14 × 74
10	W14 × 53	W14 × 74	W14 × 68	W14 × 61	W14 × 43	W14 × 43	W14 × 43
11	W14 × 43	W14 × 34	W14 × 30	W14 × 34	W14 × 38	W14 × 34	W14 × 30
12	W14 × 43	W14 × 22	W14 × 38	W14 × 34	W14 × 22	W14 × 22	W14 × 22
13	W14 × 145	W14 × 145	W14 × 159	W14 × 145	W14 × 99	W14 × 120	W14 × 90
14	W14 × 145	W14 × 132	W14 × 132	W14 × 132	W14 × 109	W14 × 99	W14 × 120
15	W14 × 120	W14 × 109	W14 × 99	W14 × 109	W14 × 82	W14 × 109	W14 × 90
16	W14 × 90	W14 × 82	W14 × 82	W14 × 82	W14 × 90	W14 × 82	W14 × 99
17	W14 × 90	W14 × 61	W14 × 68	W14 × 68	W14 × 74	W14 × 90	W14 × 68
18	W14 × 61	W14 × 48	W14 × 48	W14 × 43	W14 × 61	W14 × 61	W14 × 61
19	W14 × 30	W14 × 30	W14 × 34	W14 × 34	W14 × 30	W14 × 38	W14 × 43
20	W14 × 26	W14 × 22	W14 × 22	W14 × 22	W14 × 22	W14 × 22	W14 × 22
Weight (lb)	220,465	214,860	212,640	212,364	215,874	206,520	203,490
Average optimized weight (lb)	229,555	222,620	N/A	215,226	225,071	216,475	208,648

Worst optimized weight (lb)	N/A	N/A	N/A	N/A	N/A	243,143	226,019
Number of structural analyses	15,500	13,924	7500	5500	8280	19,640	18,820
Constraint tolerance (%)	None	None	None	None	None	None	None

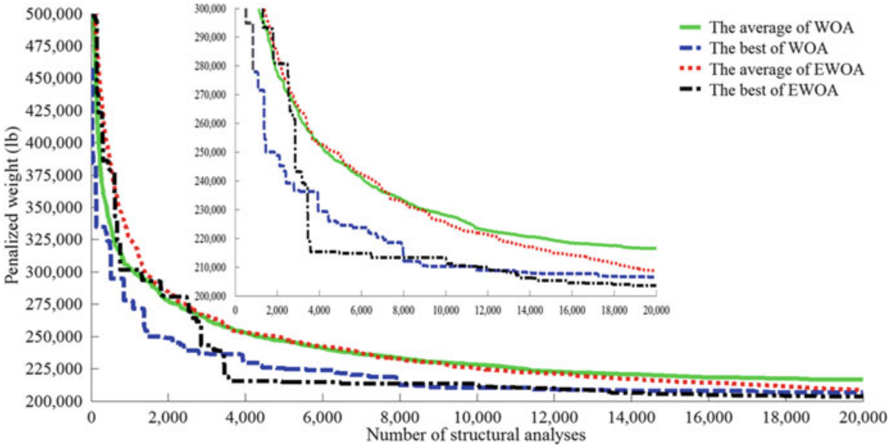


Fig 4.12 Convergence curves obtained by EWOA and WOA in the 3-bay 24-story frame problem [1]

4.5 Concluding Remarks

This chapter presented an improved formulation of the whale optimization algorithm which tries to maintain a proper balance between the diversification and the intensification inclinations. The EWOA algorithm was applied to weight minimization problems of skeletal structures. The simplicity of WOA is preserved in EWOA since no internal parameter is added. The suitability and efficiency of EWOA is illustrated through two truss and two frame optimization problems. EWOA converged to better designs in all test problems. Also, the average weight/volume found by EWOA in the independent optimization runs is lower in all benchmark examples indicating that the search reliability of the proposed method is superior to the compared methods. Besides, it can be seen from convergence history curves that the convergence rate of the EWOA algorithm is higher than that of the WOA.

References

1. Kaveh A, Ilchi Ghazaan M (2016) Enhanced whale optimization algorithm for sizing optimization of skeletal structures. Mech Based Des Struct Mach Int J (Published online: 21 July 2016)
2. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
3. Kaveh A, Ilchi Ghazaan M (2015) A comparative study of CBO and ECBO for optimal design of skeletal structures. Comput Struct 153:137–147
4. Kaveh A, Talatahari S (2009) A particle swarm ant colony optimization for truss structures with discrete variables. J Constr Steel Res 65:1558–1568

5. Kaveh A, Talatahari S (2010) Optimum design of skeletal structure using imperialist competitive algorithm. Comput Struct 88:1220–1229
6. Kaveh A, Ilchi Ghazaan M, Bakhshpoori T (2013) An improved ray optimization algorithm for design of truss structures. Periodica Polytech 57:1–15
7. Ho-Huu V, Nguyen-Thoi T, Vo-Duy T, Nguyen-Trang T (2016) An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. Comput Struct 165:59–75
8. American Institute of Steel Construction (AISC) (1989) Manual of steel construction: allowable stress design. American Institute of Steel Construction, Chicago, IL
9. Hasançebi O, Çarbas S, Dogan E, Erdal F, Saka MP (2009) Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Comput Struct 87(5–6):284–302
10. Kaveh A, Talatahari S (2010) A discrete Big Bang–Big Crunch algorithm for optimal design of skeletal structures. Asian J Civil Eng 11(1):103–122
11. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
12. American Institute of Steel Construction (AISC) (2001) Manual of steel construction: load and resistance factor design. American Institute of Steel Construction, Chicago, IL
13. Dumonteil P (1992) Simple equations for effective length factors. AISC Eng J 29(3):111–115
14. Kaveh A, Talatahari S (2009) Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization. Stud Comput Intell 239:159–198
15. Kaveh A, Talatahari S (2012) Charged system search for optimal design of frame structures. Appl Soft Comput 12:382–393
16. Camp CV, Bichon BJ, Stovall S (2005) Design of steel frames using ant colony optimization. J Struct Eng ASCE 131:369–379
17. Degertekin SO (2012) Improved harmony search algorithms for sizing optimization of truss structures. Comput Struct 92–93:229–241

Chapter 5
Size and Geometry Optimization of Double-Layer Grids Using the CBO and ECBO Algorithms

5.1 Introduction

Space structures have become popular not only because of their topological attractiveness and greater reserves of strength compared to conventional structures but also because of their easy and fast construction. Double-layer grids are ideally suited for covering exhibition pavilions, assembly halls, swimming pools, hangars, churches, bridge decks, and many types of industrial buildings in which large unobstructed areas are required. Double-layer grids have been built successfully at a lower cost than equivalent conventional systems, providing at the same time additional advantages, such as greater rigidity, erection simplicity, and possibility of covering larger areas.

These grids can be thought of as logical extensions of single-layer grid frameworks, consisting of two or more sets of parallel beams intersecting each other at right or oblique angles and loaded by forces perpendicular to the plane of the framework. Single-layer grids are used for clear spans up to 10 m. For larger spans, double-layer grids are more suitable and provide an economical solution for spans up to 100 m. Double-layer grids consist of two plane grids (which are not necessarily of identical layout) forming the top and bottom layers, parallel to each other, and interconnected by vertical or inclined "web" diagonal members. Single-layer grids are mainly under the action of flexural moments, whereas the component members of double-layer grids are almost exclusively under the action of axial forces. The elimination of bending moments leads to a full utilization of strength of all the elements. Double-layer grids have a greater number of structural elements and employing optimization techniques has a considerable impact on the economy and efficient design of such structures [1]. This study focuses on economical comparison of two commonly used double-layer grid configurations, namely, two-way on two-way grid and diagonal on diagonal grid and determining their optimum span-depth ratio. The span ranges of 15×15 m and 40×40 m with certain bays of equal length in two directions are considered as small and big size

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_5

grids, respectively. Bottom layer is simply supported at the corner nodes and as mid-edge at two parallel sides of the grid for the small and big span cases, respectively. The discrete values of depth are chosen from a certain interval with a 0.5 m increment for both cases to achieve the optimum value. For determining the grouping effects, various grouping patterns are applied in each case. Finally the 20×20 m square on larger square grid for the effect of support location on the weight of the double-layer grid is introduced. The discrete values of depth are selected from a certain interval with a 0.25 m increment in this case [2].

Colliding bodies optimization (CBO) is a new metaheuristic search algorithm that is developed by Kaveh and Mahdavi [3]. CBO is based on the governing laws of one-dimensional collision between two bodies from the physics where an object collides with another and they move toward the minimum energy level. The CBO is simple in concept, depends on no internal parameters, and does not use memory for saving the best-so-far solutions. The enhanced colliding bodies optimization (ECBO) is introduced by Kaveh and Ilchi Ghazaan [4], and it uses memory to save some historically best solutions to improve the CBO performance without increasing the computational cost. In this method, some components of agents are also changed to help the agents to escape from local minima. In this chapter, the ability of the CBO and ECBO on optimal design of double-layer grids is examined to carry out a precise comparison between different configurations. The design algorithm is supposed to obtain minimum weight grid through suitable selection of tube sections available in AISC-LRFD [5]. Strength constraints of AISC-LRFD specifications and displacement constraints are imposed on grids. Moreover, three other powerful advanced algorithms consisting of the HPSACO [6] (based on PSO, ACO, and HS algorithms), the HBB–BC [7] (based on BB–BC and PSO methods), and the CS [8] are applied to carry out a precise assessment and demonstrate the effectiveness and robustness of the CBO and ECBO algorithms in achieving better designs and estimating better depth for each type. Finally the effect of support location on the weight of different kinds of double-layer grids is investigated using ECBO algorithm.

The remainder of this chapter is organized as follows: In Sect. 5.2, the mathematical formulation of the structural optimization problems is presented and a brief explanation of the AISC-LRFD is provided. Section 5.3 includes an explanation of the CBO and ECBO algorithms. In Sect. 5.4 structural models are explained and three numerical examples are presented in Sect. 5.5. The last section concludes the chapter.

5.2 Optimal Design of Double-Layer Grids

The allowable cross sections are considered as 37 steel pipe sections as shown in Table 5.1, where the abbreviations ST, EST, and DEST stand for standard weight, extra strong, and double extra strong, respectively. These sections are taken from AISC-LRFD [5] and this code is also utilized for design.

Table 5.1 The allowable steel pipe sections taken from AISC-LRFD

	Type	Nominal diameter (in)	Weight per ft (lb)	Area (in^2)	*I* (in^4)	Gyration radius (in)	*J* (in^4)
1	ST	½	0.85	0.25	0.017	0.261	0.034
2	EST	½	1.09	0.32	0.02	0.25	0.04
3	ST	¾	1.13	0.333	0.037	0.334	0.074
4	EST	¾	1.47	0.433	0.045	0.321	0.09
5	ST	1	1.68	0.494	0.087	0.421	0.175
6	EST	1	2.17	0.639	0.106	0.407	0.211
7	ST	1¼	2.27	0.669	0.195	0.54	0.389
8	ST	1½	2.72	0.799	0.31	0.623	0.62
9	EST	1¼	3.00	0.881	0.242	0.524	0.484
10	ST	2	3.65	1.07	0.666	0.787	1.33
11	EST	1½	3.63	1.07	0.391	0.605	0.782
12	EST	2	5.02	1.48	0.868	0.766	1.74
13	ST	2½	5.79	1.7	1.53	0.947	3.06
14	ST	3	7.58	2.23	3.02	1.16	6.03
15	EST	2½	7.66	2.25	1.92	0.924	3.85
16	DEST	2	9.03	2.66	1.31	0.703	2.62
17	ST	3½	9.11	2.68	4.79	1.34	9.58
18	EST	3	10.25	3.02	3.89	1.14	8.13
19	ST	4	10.79	3.17	7.23	1.51	14.5
20	EST	3½	12.50	3.68	6.28	1.31	12.6
21	DEST	2½	13.69	4.03	2.87	0.844	5.74
22	ST	5	14.62	4.3	15.2	1.88	30.3
23	EST	4	14.98	4.41	9.61	1.48	19.2
24	DEST	3	18.58	5.47	5.99	1.05	12
25	ST	6	18.97	5.58	28.1	2.25	56.3
26	EST	5	20.78	6.11	20.7	1.84	41.3
27	DEST	4	27.54	8.1	15.3	1.37	30.6
28	ST	8	28.55	8.4	72.5	2.94	145
29	EST	6	28.57	8.4	40.5	2.19	81
30	DEST	5	38.59	11.3	33.6	1.72	67.3
31	ST	10	40.48	11.9	161	3.67	321
32	EST	8	43.39	12.8	106	2.88	211
33	ST	12	49.56	14.6	279	4.38	559
34	DEST	6	53.16	15.6	66.3	2.06	133
35	EST	10	54.74	16.1	212	3.63	424
36	EST	12	65.42	19.2	362	4.33	723
37	DEST	8	72.42	21.3	162	2.76	324

ST standard weight; *EST* extra strong; *DEST* double extra strong

The aim of weight minimization of a grid is to find a set of design variables leading to minimum weight satisfying certain constraints. This can be expressed as

$$\begin{aligned}&\text{Find } \{X\} = \left[x_1,\ x_2,\ x_3,\ \ldots,\ x_{ng}\right],\quad x_i \in D = \{d_1,\ d_2,\ d_3,\ \ldots,\ d_{37}\}\\ &\text{To minimize } W(\{X\}) = \sum_{i=1}^{ng} x_i \sum_{j=1}^{nm(i)} \rho_j . L_j\end{aligned} \tag{5.1}$$

where $\{X\}$ is the set of design variables, ng is the number of member groups in structure (number of design variables), D is the cross-sectional areas available for groups according to Table 5.1, $W(\{X\})$ presents weight of the grid, $nm(i)$ is the number of members for the ith group, and ρ_j and L_j denote the material density and the length for the jth member of the ith group, respectively.

The constraint conditions for grid structures are briefly explained in the following:

Displacement constraints:

$$\delta^i \leq \delta^{\max},\ i = 1, 2,\ \ldots, nn \tag{5.2}$$

Tension member constraints:

$$P_{\mathrm{u}} \leq P_{\mathrm{r}} :\ P_{\mathrm{r}} = \min\begin{cases} F_{\mathrm{y}} . A_{\mathrm{g}} . \phi_t & \phi_{\mathrm{t}}=0.9 \\ F_{\mathrm{u}} . A_{\mathrm{e}} . \phi_t & \phi_{\mathrm{t}}=0.75 \end{cases} \tag{5.3}$$

Compression member constraints:

$$\begin{aligned}&P_{\mathrm{u}} \leq P_{\mathrm{r}}, \qquad P_{\mathrm{r}} = \phi_c . F_{\mathrm{cr}} . A_{\mathrm{g}}; \qquad \phi_c = 0.85\\ &F_{\mathrm{cr}} = \min\begin{cases} \left(0.658 F_{\mathrm{y}}/F_{\mathrm{e}}\right) F_{\mathrm{y}}, & \dfrac{KL}{r} \leq 4.71\sqrt{E/F_{\mathrm{y}}} \\ 0.877 F_{\mathrm{e}} & \dfrac{KL}{r} > 4.71\sqrt{E/F_{\mathrm{y}}} \end{cases}, \quad F_{\mathrm{e}} = \pi^2 E/(KL/r)^2\end{aligned} \tag{5.4}$$

Slenderness ratio constraints:

$$\begin{aligned}&\lambda_c = KL/r \leq 200 \quad \text{for compression members}\\ &\lambda_t = KL/r \leq 300 \quad \text{for tension members}\end{aligned} \tag{5.5}$$

where δ^i and $\delta_i^{\max}$ are the displacement and allowable displacement for the ith node, nn is the number of nodes, nm is the total number of members and K is effective length factor taken as 1, P_{u} is the required strength (tension or compression), and A_{g} and A_{e} are the cross-sectional and effective net areas of a member, respectively.

In order to handle the constraints, a penalty approach is utilized. In this method, the aim of the optimization is redefined by introducing a cost function as

$$f_{\text{cost}}(\{X\}) = (1 + \epsilon_1 . v)^{\epsilon_2} \times W(\{X\}), \quad v = \sum_{i=1}^{nn} v_i^d + \sum_{i=1}^{nm} \left(v_i^\sigma + v_i^\lambda \right) \tag{5.6}$$

where v is the constraint violation function and v_i^d, v_i^σ, and v_i^λ are constraint violations for displacement, stress, and slenderness ratio, respectively. ϵ_1 and ϵ_2 are penalty function exponents which are selected considering the exploration and exploitation rate of the search space. Here ϵ_1 is set to unity; ϵ_2 is selected in a way that it decreases the penalties and reduces the cross-sectional areas. Thus, in the first steps of the search process, ϵ_2 is set to 1 and it linearly increases to 3.

5.3 CBO and ECBO Algorithms

Colliding bodies optimization (CBO) is a new population-based stochastic optimization algorithm based on the governing laws of one-dimensional collision between two bodies from the physics [3]. Each agent is modeled as a body with a specified mass and velocity. A collision occurs between pairs of objects to find the global or near-global solutions. Enhanced colliding bodies optimization (ECBO) uses a memory vector to save some best solutions and utilizes a mechanism to escape from local optima [4].

5.3.1 A Brief Explanation and Formulation of the CBO Algorithm

In CBO, each solution candidate X_i containing a number of variables (i.e., $X_i = \{X_{i,j}\}$) is considered as a colliding body (CB). The massed objects are composed of two main equal groups: stationary and moving objects, where the moving objects move to follow stationary objects and a collision occurs between pairs of objects (Fig. 5.1). This is done for two purposes: (i) to improve the positions of moving objects and (ii) to push stationary objects toward better positions. After the collision, new positions of the colliding bodies are updated based on new velocities using the collision laws governed by momentum and energy [3]. When a collision occurs in an isolated system, the total momentum of the system of objects is conserved. Provided that there are no net external forces acting upon the objects, the momentum of all objects before the collision equals the momentum of all objects after the collision.

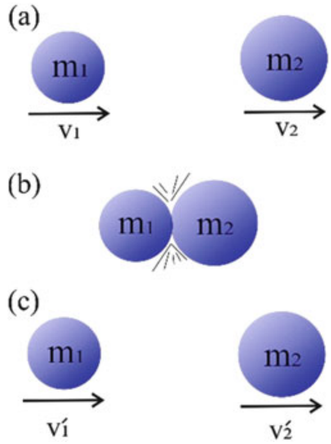


Fig. 5.1 Collision between two bodies: (**a**) before collision, (**b**) collision, and (**c**) after collision

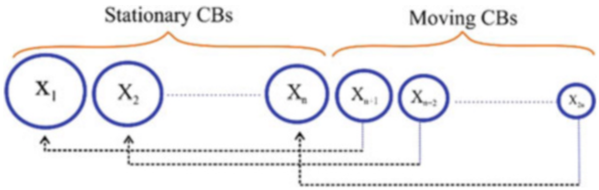


Fig. 5.2 The sorted CBs in an ascending order and the matching process for the collision

CBO starts with an initial population consisting of $2n$ parent individuals created by means of a random initialization. Then, CBs are sorted in ascending order based on the value of cost function as shown in Fig. 5.2.

The CBO procedure can briefly be outlined as follows.

As stated before each agent called CB has a specified mass that is defined as

$$m_k = \frac{\frac{1}{\mathrm{fit}(k)}}{\sum_{i=1}^{n} \frac{1}{\mathrm{fit}(i)}}, \ k = 1, 2, \ldots, n \tag{5.7}$$

where $\mathrm{fit}(i)$ represents the objective function value of the ith CB and n is the number of colliding bodies. After sorting colliding bodies according to their objective function values in an increasing order, two equal groups are created: (i) stationary group and (ii) moving group (Fig. 5.2). Moving objects collide with stationary objects to improve their positions and push stationary objects toward better positions. The velocities of the stationary and moving bodies before collision (v_i) are calculated by

$$v_i = 0, \ i = 1, \ldots, \frac{n}{2} \tag{5.8}$$

$$v_i = x_{i-\frac{n}{2}} - x_i,\ i = \frac{n}{2}+1, \frac{n}{2}+2\ldots, n \tag{5.9}$$

where x_i is the position vector of the ith CB. The velocity of stationary and moving CBs after the collision (v_i') is evaluated by

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} \qquad i = 1, 2,\ \ldots, \frac{n}{2} \tag{5.10}$$

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}} \qquad i = \frac{n}{2}+1, \frac{n}{2}+2,\ \ldots, n \tag{5.11}$$

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{5.12}$$

where ε is the coefficient of restitution (COR) and *iter* and $iter_{max}$ are the current iteration number and the total number of iterations for optimization process, respectively. New positions of each group are stated by the following formulas:

$$x_i^{\text{new}} = x_i + rand \circ v_i', \qquad i =\ 1, 2,\ \ldots, \frac{n}{2} \tag{5.13}$$

$$x_i^{\text{new}} = x_{i-\frac{n}{2}} + rand \circ v_i', \qquad i =\ \frac{n}{2}+1,\ \ldots, n \tag{5.14}$$

where x_i^{new}, x_i, and v_i' are the new position, previous position, and the velocity after the collision of the ith CB, respectively. *rand* is a random vector uniformly distributed in the range of [−1,1], and the sign "∘" denotes an element-by-element multiplication.

5.3.2 *Pseudo-Code of the ECBO Algorithm*

In the ECBO, a memory that saves a number of historically best CBs is utilized to improve the performance of the CBO and reduce the computational cost. Furthermore, ECBO changes some components of CBs randomly to prevent premature convergence [4]. In this section, in order to introduce the ECBO algorithm, the following steps should be taken.

5.3.2.1 Initialization

Step 1: The initial locations of CBs are created randomly in an m-dimensional search space.

$$x_i^0 = x_{\min} + random \circ (x_{\max} - x_{\min}), \qquad i = 1, 2, 3, \ldots, n \tag{5.15}$$

where x_i^0 is the initial solution vector of the ith CB, $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable limits vectors, and random is a random vector with each component being in the interval [0,1].

5.3.2.2 Search

Step 1: The value of the mass for each CB is calculated by Eq. (5.7).

Step 2: Colliding memory (CM) is considered to save some historically best CB vectors and their related mass and objective function values. The size of the CM is taken as $n/10$ (n is the population size) in this chapter. In each iteration, solution vectors that are saved in the CM are added to the population, and the same number of the current worst CBs are deleted.

Step 3: CBs are sorted according to their objective function values in an increasing order. To select the pairs of CBs for collision, they are divided into two equal groups: (i) stationary group and (ii) moving group.

Step 4: The velocities of stationary and moving bodies before collision are evaluated by Eqs. (5.8) and (5.9), respectively.

Step 5: The velocities of stationary and moving bodies after collision are calculated by Eqs. (5.10) and (5.11), respectively.

Step 6: The new location of each CB is evaluated by Eqs. (5.13) or (5.14).

Step 7: A parameter like ***Pro*** within (0, 1) is introduced which specifies whether a component of each CB must be changed or not. For each CB ***Pro*** is compared with rn_i ($i = 1,2,\ldots, n$) which is a random number uniformly distributed within (0, 1). If $rn_i < \boldsymbol{Pro}$, one dimension of ith CB is selected randomly and its value is regenerated by

$$x_{ij} = x_{j,\min} + random.\left(x_{j,\max} - x_{j,\min}\right) \tag{5.16}$$

where x_{ij} is the jth variable of the ith CB. $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the jth variable. In this chapter, the value of ***Pro*** is set to 0.3.

5.3.2.3 Terminating Condition Check

Step 1: After the predefined maximum evaluation number, the optimization process is terminated [9].

5.4 Structural Models

Two commonly used configurations for double-layer grids considered in this study are two-way on two-way and diagonal on diagonal square grids [10]. Two span values of 15×15 m and 40×40 m with certain bays of equal lengths in two directions are considered as small and big size spans. Simply supported condition is employed for bottom layer at the corner nodes and mid-edge at two parallel sides for the small and big span cases, respectively. The discrete values of depth are chosen from a certain interval with a 0.5 m increment for both cases to achieve the optimum value. At last the 20×20 m square on larger square grid for the effect of support locations on the weight of the double-layer grid is introduced. The discrete values of depth are selected from a certain interval with a 0.25 m increment in this case [2].

As mentioned before double-layer grids have a large number of structural elements, and in order to simplify the design, they should be divided into some groups. The element grouping of such design can be selected by designers in any scheme or patterns, but if the members with the same behavior are placed in the same group, the design becomes more efficient and economical (e.g., all members in one group have the same stress ratios, approximately). To address this issue, the SAP2000 toolbox for auto and fully stressed design could be used to select the element grouping pattern at the preliminary stage of design considering the stress ratios of the elements. However, the selection of such pattern can be based on experiences, engineering judgment, or administrative constraints. In this chapter three element grouping patterns, namely, GP1, GP2, and GP3, are introduced for the purpose of practical fabrication and determining the grouping effects on the different systems. Considering different sections of the top-layer, bottom-layer, and diagonal elements leads to the first grouping type which is only applied to the 15×15 m span case with three design variables. In the second one, the top-layer, bottom-layer, and diagonal elements are put into different groups in a diamond-like manner around central node. The GP3 grouping pattern is the same as the second one, but it is in a square form. The configuration, support locations, and element grouping patterns of double-layer grids are shown in Fig. 5.3. Due to symmetry, only a quarter of the 15×15 m span case is shown in this figure. The element grouping in the form of GP2 is depicted by dark and light hatching.

5.5 Numerical Examples

The double-layer grids are assumed as ball jointed, with top-layer joints being subjected to concentrated vertical loads corresponding to the uniformly distributed load of magnitude 200 kg/m^2. Stress and slenderness constraints [Eqs. (5.3), (5.4), and (5.5)] according to AISC-LRFD provisions and displacement limitations of span/600 are imposed on all the nodes in vertical direction. The modulus of

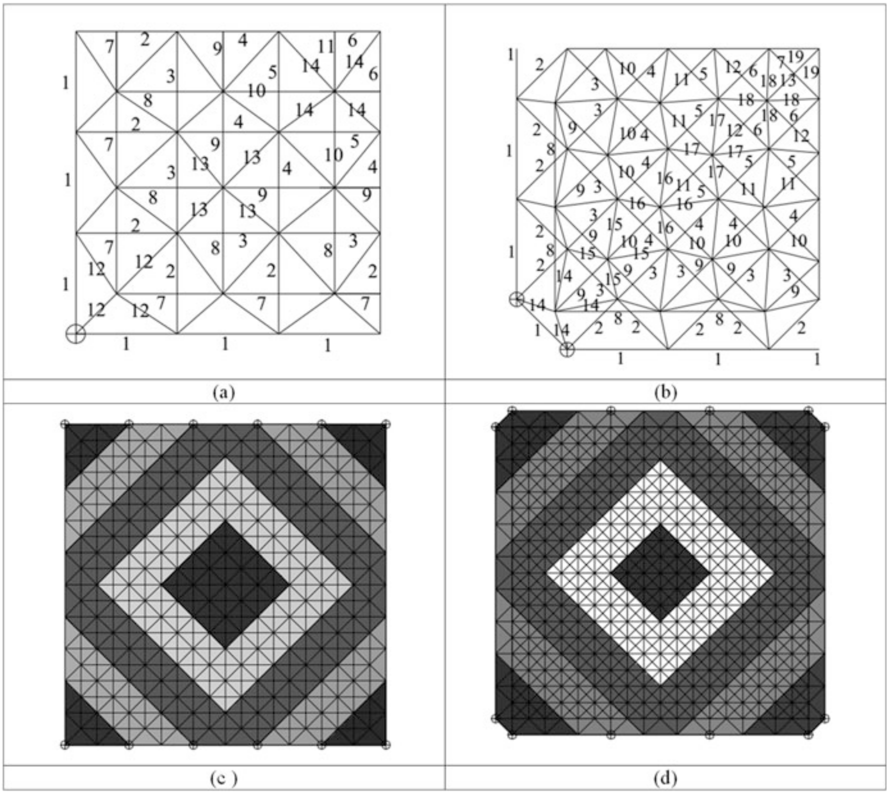


Fig. 5.3 Topology, element grouping, support locations for different cases; (**a**) 15 × 15 m two-way on two-way grid, (**b**) 15 × 15 m diagonal on diagonal grid, (**c**) 40 × 40 m two-way on two-way grid, (**d**) 40 × 40 m diagonal on diagonal grid [11]

elasticity is considered as 205 kN/mm^2, and the yield stress of steel is taken as 248.2 MPa.

In CBO and ECBO, a population of $n = 30$ CBs is utilized, and the size of colliding memory is considered as $n/10$ that is taken as 3 for ECBO. The predefined maximum evaluation number is considered as 15,000 analyses for all examples. Because of the stochastic nature of the algorithms, each example has been solved 5 times independently. In all problems, CBs are allowed to select discrete values from the permissible list of cross sections (real numbers are rounded to the nearest integer in each iteration). The algorithms are coded in MATLAB and the structures are analyzed using the direct stiffness method.

5.5.1 A 15 × 15 m Double-Layer Square Grid

A 15 × 15 m span structure is studied as a small sized double-layer grid. The first common type is the two-way on two-way grid which contains 85 nodes and 288 members, and the second one is the diagonal on diagonal grid with 145 nodes and 528 members. Each span contains 6 bays of equal length in both directions. Grouping patterns of GP1 and GP2 lead to 3 and 9 design variables for each type. The third grouping pattern yields 14 and 19 design variables for two-way on two-way and diagonal on diagonal grids, respectively. The range of discrete depths is considered as the interval [3, 6] with increments of 0.5 m for each type to achieve the optimum depth. The fundamental difference between diagonal and rectangular grids is that in the former beams are of varying length (L), and therefore, even if all the beams are of the same cross-sectional dimensions and have the same axial stiffness (EA), their relative stiffness (EA/L) varies considerably. The diagonal grid consists of beams forming an oblique angle with the walls. This type is often used for small span cases because its greater rigidity leads to a substantial reduction in the deflections and not considering the number and complexity of joints is often favored by engineers and architects because of its convenience and appealing features. Table 5.2 shows that diagonal on diagonal grid is a more suitable form for small span length compared to two-way on two-way grid even with larger number of members. It is apparent from the table that CBO has gained better results than other three methods (CS, HBB–BC, and HPSACO) except for some cases with slight differences. Furthermore, the ECBO has produced the lightest designs among other methods.

For graphical comparison of the algorithms, the convergence histories for the best result of five independent runs are shown in Fig. 5.4 for the diagonal on diagonal grid, GP3, and depth of 1.5 m.

Table 5.3 shows the best design vectors and the corresponding weights for two methods, and these are compared with those of engineering design found by SAP2000. The results of the CBO and ECBO are 22.6 % and 25 % lighter than engineering design, respectively.

Figure 5.5 shows the obtained optimum weights for various grouping patterns and depths of grids. As depicted, the optimum height for two-way on two-way and diagonal on diagonal grids are 2 m and 1.5 m, respectively. More importantly, the GP3 grouping type with more design variables results in heavier designs compared to those of GP2 grouping type for two-way on two-way grid. In the GP1 case in which the problem has only three design variables, all methods approximately yield the same design. In diagonal on diagonal grid, GP2 and GP3 grouping patterns yield the same results approximately. It is apparent that GP2 with fewer number of design variables is more economical for this type of grid.

Table 5.2 Performance comparison for the 15 × 15 m double-layer grids (kg)

Two-way on two-way grid									
	GP1			GP2			GP3		
	Height = 1 m	Height = 1.5 m	Height = 2 m	Height = 1 m	Height = 1.5 m	Height = 2 m	Height = 1 m	Height = 1.5 m	Height = 2 m
Kaveh et al. [12]									
CS	8931.492	6598.641	6768.999	7244.872	4127.063	4153.159	7006.406	4751.817	4920.276
HBB–BC	8931.492	6598.641	6768.999	7250.741	4371.717	4363.474	6999.783	5541.080	6550.501
HPSACO	8931.492	6598.641	6768.999	7235.120	4360.259	4533.831	7140.266	5056.159	5478.960
Present work [2]									
CBO	8931.492	6598.641	6768.999	7227.374	4127.063	3916.542	7129.515	4771.940	4904.675
ECBO	8931.492	6598.641	6768.999	7200.670	4127.063	3916.542	6941.492	4599.313	4641.717
Diagonal on diagonal grid									
	GP1			GP2			GP3		
	Height = 1 m	Height = 1.5 m	Height = 2 m	Height = 1 m	Height = 1.5 m	Height = 2 m	Height = 1 m	Height = 1.5 m	Height = 2 m
Kaveh et al. [12]									
CS	7471.287	5927.232	6002.888	6319.767	4757.806	4978.462	5402.767	4180.124	4654.215
HBB–BC	7471.287	5927.232	5590.446	6209.962	5104.293	6099.758	5643.173	5749.242	5931.222
HPSACO	7471.287	5927.232	5590.446	6203.467	4873.785	5097.121	6157.951	5270.756	6373.701
Present work [2]									
CBO	7471.287	5927.232	5590.446	6195.303	4532.091	4683.532	5576.106	3917.503	4661.475
ECBO	7471.287	5927.232	5590.446	6187.411	4444.858	4448.896	5381.764	3794.836	4263.224

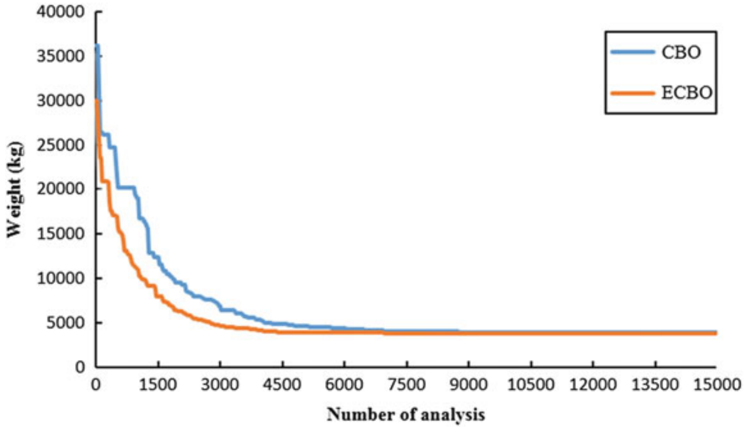


Fig. 5.4 The convergence history for the 15 × 15 m diagonal on diagonal grid (GP3 and layer thickness = 1.5 m)

Table 5.3 Optimum design of 15 × 15 m double-layer grids

		Optimum section (designations)	
Group number	Engineering design	CBO	ECBO
1	PIPST (2)	PIPST (1½)	PIPST (1½)
2	PIPST (2½)	PIPST (2)	PIPST (2)
3	PIPST (1½)	PIPST (1¼)	PIPST (1¼)
4	PIPST (1)	PIPST (1)	PIPST (1)
5	PIPST (½)	PIPST (½)	PIPST (½)
6	PIPST (½)	PIPST (½)	PIPST (½)
7	PIPST (½)	PIPST (½)	PIPEST (½)
8	PIPST (3)	PIPST (3)	PIPST (2½)
9	PIPST (1½)	PIPST (1)	PIPST (1¼)
10	PIPST (1½)	PIPST (1¼)	PIPST (1¼)
11	PIPST (2)	PIPEST (1½)	PIPST (1½)
12	PIPST (2)	PIPEST (1½)	PIPEST (1½)
13	PIPST (2)	PIPEST (1½)	PIPEST (1½)
14	PIPST (2½)	PIPST (2)	PIPST (2)
15	PIPST (1¼)	PIPST (1)	PIPST (1)
16	PIPEST (1)	PIPST (1)	PIPST (1)
17	PIPST (1)	PIPST (1)	PIPST (1)
18	PIPEST (¾)	PIPST (1)	PIPST (1)
19	PIPEST (¾)	PIPST (1)	PIPST (1)
Demand/Capacity ratio limit	0.999	–	–
Best weight (kg)	5061.1542	3917.5032	3794.8357

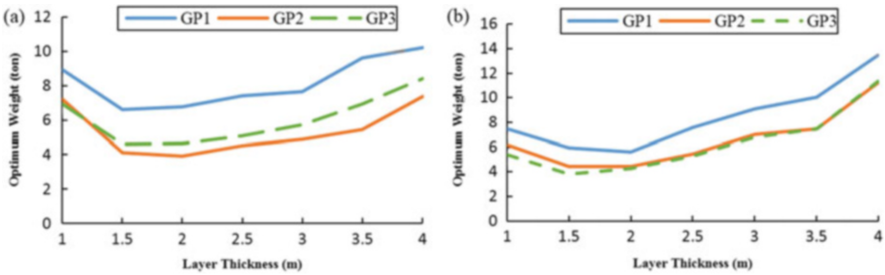


Fig. 5.5 Best results of ECBO for 15 × 15 m double-layer grids in each group type: (**a**) two-way on two-way grid and (**b**) diagonal on diagonal grid

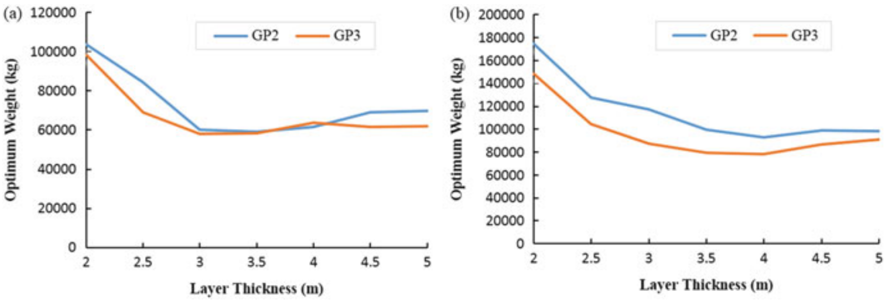


Fig. 5.6 Best results of ECBO for 40 × 40 m double-layer grids in each group type: (**a**) two-way on two-way grid and (**b**) diagonal on diagonal grid

5.5.2 A 40 × 40 m Double-Layer Square Grid

A 40 × 40 m span case is considered as a big size of double-layer grids. The first common type is two-way on two-way grid which contains 221 nodes and 800 members. The second one is diagonal on diagonal grid with 401 nodes and 1520 members. Each span contains 10 bays of equal length in both directions. The first grouping pattern is ignored in this case because of the size of the structure. Grouping pattern of GP2 leads to 15 design variables for both types. The third grouping pattern leads to 24 and 31 design variables for two-way on two-way and diagonal on diagonal grids, respectively. The range of discrete depths of [4, 7] is considered with a 0.5 m increment each type to achieve the optimum depth.

Figure 5.6 shows the obtained optimum weight for various grouping patterns and depth of grids. As depicted, the curves of different groups for two-way on two-way grid have approximately coincided with slight differences. It should be noted that the GP2 is a more suitable way of grouping for two-way on two-way grid, because of the fewer number of groups. It is shown that the optimum height of the first type is equal to 3.5 and 3 m for GP2 and GP3, respectively, while for the second type it equals 4 m for both grouping schemes. Table 5.4 in which the best obtained weight is hatched for each case presents the performance of algorithms. The obtained

Table 5.4 Performance comparison for the 40 × 40 m double-layer grids (kg)

Two-way on two-way grid						
	GP2			GP3		
	Height = 3 m	Height = 3.5 m	Height = 4 m	Height = 2.5 m	Height = 3 m	Height = 3.5 m
Kaveh et al. [12]						
CS	61,564.751	59,709.748	65,272.400	71,274.797	58,474.360	64,833.546
HBB–BC	80,776.255	70,748.113	92,783.126	82,342.363	79,576.315	90,213.388
HPSACO	82,866.081	85,118.951	102,055.248	88,623.380	79,390.971	96,137.848
Present work [2]						
CBO	68,065.983	69,954.260	73,687.941	75,250.497	67,247.606	67,630.344
ECBO	60,237.288	59,305.147	61,487.446	68,936.889	58,142.691	58,376.942
Diagonal on diagonal grid						
	GP2			GP3		
	Height = 3.5 m	Height = 4 m	Height = 4.5 m	Height = 3 m	Height = 3.5 m	Height = 4 m
Kaveh et al. [12]						
CS	97,965.173	95,661.984	99,775.377	89,729.390	86,883.682	93,751.030
HBB–BC	113,690.418	113,987.966	135,809.980	131,973.303	120,917.170	149,910.633
HPSACO	133,017.343	112,800.347	124,293.047	131,363.973	129,412.670	149,261.498
Present work [2]						
CBO	107,859.576	109,096.309	106,550.911	110,364.074	110,823.920	127,403.576
ECBO	99,741.612	92,861.835	99,073.395	87,533.094	79,132.887	78,337.836

optimum designs for the two-way on two-way grid in GP2 and GP3 grouping schemes are 36 % and 26 % lighter than those of the diagonal on diagonal cases, respectively. It can be realized that two-way on two-way grid is a more suitable form for big span cases with the same number of span divisions (without considering the number and complexity of joints). It is apparent from the table that CBO has obtained better results compared to HBB–BC and HPSACO in all cases. It could also be seen that the enhanced version (ECBO) is capable of finding the best results in all cases expect for one. The robustness of ECBO in size and geometry optimization of big span double-layer grids is also evident.

5.5.3 The Effect of Support Location on the Weight of Double-Layer Grids

In this case the 20 × 20 m square of large square double layer grid consisting of 136 nodes and 440 members is considered as the last example. Each span is divided into eight bays with equal lengths in each direction. There are some empty spaces in the middle of the grid created by removing some of the bottom-layer members ((usually in tension). The attached bracings of the square on square offset at a rectangular pattern lead to a construction lighter than the usual type (Fig. 5.7). Due to the addition of the openings, this system is more suitable when the architect intends to provide more natural lights inside the building (skylights can be placed within the openings). This system is usually selected for structures subjected to normal range of loads. A uniformly distributed load of 200 kg/m^2 is transmitted to the concentrated vertical loads which are assigned to the nodes of the top grid proportional to their load-bearing area. Double-layer grids can be supported by steel or concrete columns, load-bearing brickworks, or perimeter ring beams. The positions of

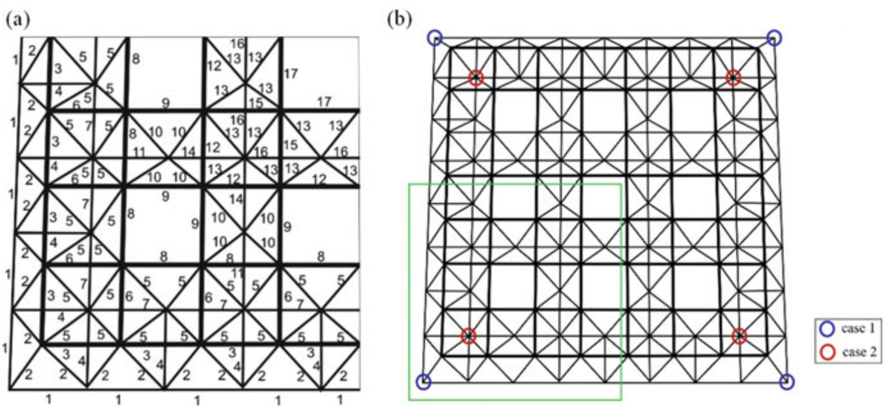


Fig. 5.7 (**a**) Element grouping for square on larger square double-layer grid. (**b**) Configuration and various types of support location

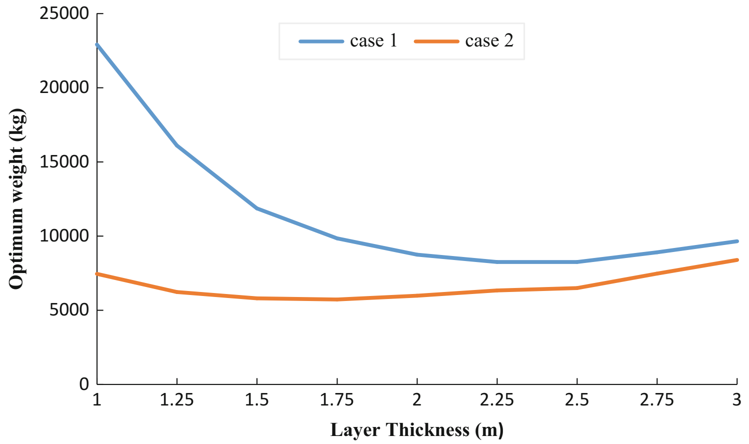


Fig. 5.8 Effect of support location on the weight of square on larger square double-layer grid and best results of ECBO algorithm

supports are important, as it influences the stress distribution. Often the locations of the supports are selected considering the functional requirements of the building. Sometimes architectural considerations may have a major effect on the location of the supports as well as in the shape of the supporting structure. For shape and size optimization, the ECBO is considered as optimization method. The grouping pattern leads to 17 design variables in a square-like manner and is introduced for the practical fabrication. Due to symmetry, only a quarter of this configuration is shown in Fig. 5.7a. The range of discrete depths of [3, 5] is considered with a 0.25 m increment to achieve the optimum depth. Figure 5.8 shows the obtained optimum weight, depth of grid, and comparison of the results between two cases of support locations. ECBO obtains an optimum height of 1.75 m for this type of grid. If possible, support at the extreme edges of the grid should be avoided as this will produce heavy forces in the directly loaded members. Support positions slightly in board are preferred. Often cantilevers can be provided by a proper support location; this leads to considerable reduction in forces and deflections. As a rule, cantilevers have little effect on shearing forces and hence on the size of the diagonals, but cantilevers of approximately 0.3 of the clear span will result in a structure that has less deflections, uses less material, and leads to a more uniform stress distribution. The forces in the lower layer are nearly twice as much as the upper layer; however, since these members are in tension, they are obviously not susceptible to buckling. Table 5.5 shows the optimum design variables and best weight that ECBO has produced as lightest design.

Table 5.5 Optimum design of 20 × 20 m double-layer grids

Group number	Optimum section (designations)
	ECBO
1	PIPST (1¼)
2	PIPST (1¼)
3	PIPST (2)
4	PIPST (1¼)
5	PIPST (2½)
6	PIPST (1¼)
7	PIPST (2)
8	PIPST (2½)
9	PIPST (1½)
10	PIPST (1¼)
11	PIPST (1)
12	PIPST (2)
13	PIPST (1¼)
14	PIPEST (1¼)
15	PIPEST (1½)
16	PIPEST (2)
17	PIPST (2)
Optimum height (m)	1.75
Best weight (kg)	5721.8492

5.6 Concluding Remarks

In this chapter, the CBO and ECBO algorithms are examined in the context of size and geometry optimization of double-layer grids designed for minimum weight. The CBO has simple structure and depends on no internal parameters and does not use memory for saving the best-so-far solutions. In order to improve the exploration capabilities of the CBO and to prevent premature convergence, a stochastic approach is employed in ECBO that changes some components of CBs randomly. Colliding memory is also utilized to save a number of the best-so-far solutions to reduce the computational cost. In order to indicate the similarities and differences between the characteristics of the CBO and ECBO algorithms, two types of double-layer grids with various span lengths are considered. Grids are designed in accordance with AISC-LRFD specifications and displacement constraints. In small span cases, diagonal on diagonal grid with more connections and members is a suitable form because of greater rigidity and other advantages like convenience and appealing features. For big span cases, two-way on two-way grids with fewer number of members are better than diagonal on diagonal ones. In this type of space structures, if the positions of supports are slightly in board, the weight of structure is decreased considerably due to reduction in forces and deflections as it influences the stress distribution and leads to using less material and results in lighter weight designs.

CBO has gained better results in small span case than three well-known algorithms (CS, HBB–BC, and HPSACO) with small differences and for big sizes has gained better design than HBB–BC and HPSACO. ECBO has better performance in all cases than other methods because of the reliability of search, solution accuracy, and speed of convergence. Generally, comparison of the results with other robustness and hybridized algorithms shows the suitability and efficiency of the proposed algorithms.

References

1. Makowski ZS (1990) Analysis, design and construction of double-layer grids, 1st edn. CRC Press, UK
2. Kaveh A, Moradveisi M (2016) Optimal design of double-layer barrel vaults using CBO and ECBO algorithms. Iran J Sci Technol Trans Civil Eng. doi:10.1007/s40996-016-0021-4
3. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel metaheuristic method. Comput Struct 139:18–27
4. Kaveh A, Ilchi Ghazaan M (2014) Computer codes for colliding bodies optimization and its enhanced version. Int J Optim Civil Eng 4:321–332
5. American Institute of Steel Construction (AISC) (1994) Manual of steel construction load resistance factor design, 2nd edn. AISC, Chicago, IL
6. Kaveh A, Talatahari S (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 87:267–283
7. Kaveh A, Talatahari S (2009) Size optimization of space trusses using Big Bang–Big Crunch algorithm. Comput Struct 87:1129–1140
8. Yang XS, Deb S (2010) Engineering optimization by Cuckoo search. Int J Math Model Numer Optim 1:330–343
9. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies algorithm for truss optimization with dynamic constraints. J Comput Civil Eng 10.1061/(ASCE)CP.1943-5487.0000445,04014104
10. Makowski ZS (1990) Analysis, design and construction of double-layer grids. Applied Science Publisher Ltd, London
11. Kaveh A, Servati H (2002) Neural networks for the approximate analysis and design of double layer grids. Int J Space Struct 17:77–89
12. Kaveh A, Bakhshpoori T, Afshari E (2011) An optimization-based comparative study of double layer grids with two different configurations using cuckoo search algorithm. Int J Optim Civil Eng 1:507–520

Chapter 6
Sizing and Geometry Optimization of Different Mechanical Systems of Domes via the ECBO Algorithm

6.1 Introduction

This chapter deals with the optimal design of double-layer Lamella domes, suspen-domes, and single-layer domes with relatively long spans including nonlinear structural behavior [1]. In recent years, much progress has been made in the optimal design of space structures by focusing on their linear behavior, neglecting non-linearities which can result in uneconomic designs. In this study, geometric nonlinearity optimization is taken into account for the abovementioned domes. There are two main steps involved in the optimization of structural problems: analysis and design. In this chapter, OPENSEES [2] is employed for analysis, and enhanced colliding bodies is utilized in the design phase. All of the required programs for the optimization phase are coded in MATLAB [3]. The design variables include cross-sectional areas of the structural elements, the height of dome, the initial strain of cables, and the cross sections of cables in the suspen-dome. In order to illustrate the efficiency of the proposed methodology, three numerical examples including optimization of a single-layer dome with rigid joints, a suspen-dome, and a double-layer dome with 12 rings subjected to dead and snow loading are presented. The main contribution of the chapter is to utilize an efficient metaheuristic algorithm for optimization of domes. Optimal design of structures is usually achieved by considering the design variables to find an objective function which is the minimum weight while all of the design constraints are satisfied.

The dome shape not only provides an elegant appearance but also offers one of the most efficient interior environments for human residence because air and energy circulation are managed without obstruction.

Suspen-Dome is a new style of prestressed space grid structure [4]. In recent years, this type of dome has been used in some large-scale engineering structures, such as Hikarigaoka Dome in Japan and Olympic Badminton Stadium of Beijing in China. The symmetrical configuration of the Lamella dome and its triangular configuration make it the topmost single-layer dome of the type. Figure 6.1

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_6

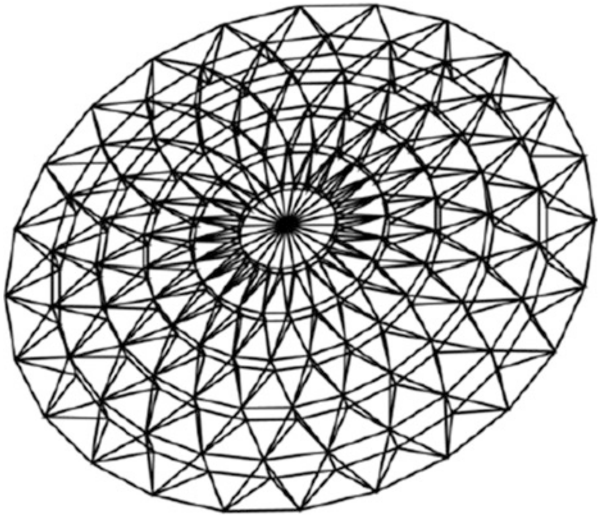


Fig. 6.1 Configuration of a double-layer/suspen-dome

presented by Kitipornchai et al. [5] shows a Lamella suspen-dome system. This study takes geometric imperfection, asymmetric loading, rise-to-span ratio, and connection rigidity of the dome into consideration.

The Colliding Bodies Optimization (CBO) was introduced for design of structures with continuous and discrete variables [6]. Design variables are cross-sectional areas selected from a discrete list of available values [7]. The design optimization of geometrically nonlinear geodesic domes was carried out, where the design algorithm developed determines the optimum height of the crown as well as the optimum tubular steel sections for the members [8]. In this chapter, optimum topology design of linear elastic geodesic domes was presented. The design algorithm determines the optimum number of rings, the optimum height of crown, and tubular sections for the geodesic domes. The optimum topology design algorithm based on the hybrid Big Bang–Big Crunch optimization method was presented for the Schwedler and Ribbed domes in Kaveh and Talatahari [9].

An investigation on the characteristics and feasibility of different tension schemes and also checking the accuracy of the numerical model and its calculated results was done for suspen-dome by Nie et al. [10]. Kamyab and Salajegheh [11] used an enhanced particle swarm optimization (EPSO) algorithm for size optimization of nonlinear scallop domes subjected to static loading. A comparative study was conducted for the optimal design of different types of single-layer domes by Kaveh and Rezaei [12]. In Kaveh and Rezaei [13], a sizing optimization was carried out for the optimum nonlinear design of suspen-domes having complex mechanical components. In this chapter, the optimum geometry and topology design for single-layer domes is carried out by utilizing the CBO.

The rest of this chapter is organized as follows. Section 6.2 consists of the formulation of the optimal design of dome structures according to LRFD design method. Section 6.3 summarizes the laws of collision between two bodies.

Table 6.1 The standard cable section according to BS 5896

Diameter (mm)	Tensile strength (MPa)	Mass (g/m)	Cross-sectional area (mm^2)	Yield stress at 0.1 % elongation
8	1860	296.8	38.0	60.8
9.3	1860	406.1	52.0	83.2
9.6	1960	429.6	55.0	87.7
11.3	1860	585.8	75.0	120.0
12.5	1860	726.3	93.0	149.0
12.9	1860	781.0	100.0	160.0
15.2	1770	1093.0	139.0	212.0
15.7	1770	1172.0	150.0	240.0

In Sect. 6.4, three metaheuristic algorithms are compared for optimization of domes. Comparative study is performed between optimal design of suspen-domes, single layer with pin- and rigid-jointed domes, and double-layer Lamella domes using ECBO algorithm in Sect. 6.5. Finally, Sect. 6.6 summarizes the main findings of this chapter.

6.2 Optimal Design Problem of Lamella Domes According to LRFD

The allowable and standard cables which should be used in the tensegrity system (hoop and radial cables) are shown in Table 6.1. The allowable cross sections of steel elements, used in the domes, are standard 37 steel pipe sections shown in Table 6.2. In this table, the abbreviations ST, EST, and DEST stand for standard weight, extra strong, and double-extra strong, respectively. These sections are taken from LRFD-AISC [14] which is also utilized as the code of practice. The process of the optimal design of the dome structures includes introducing variables and constraints and can be summarized as:

$$\begin{aligned}
&\text{Find } X = \left[x_1, x_2, .., x_{ng}\right], h \\
&x_i \in \left\{d_1, d_2, \ldots, d_{ng}\right\} \\
&h_i \in \left\{h_{\min}, h_{\min} + h^*, \ldots, h_{\max}\right\} \\
&\text{To minimize} \\
&V(x) = \sum_{i=1}^{nm} x_i . l_i
\end{aligned} \tag{6.1}$$

subjected to the following constraints:

Displacement constraint:

$$\delta_i \leq \delta_i^{\max} \quad i = 1, 2, \ldots, nn. \tag{6.2}$$

Interaction formula constraints:

$$\frac{P_{\mathrm{u}}}{2\phi_c P_{\mathrm{n}}} + \left(\frac{M_{\mathrm{u}x}}{\phi_b M_{\mathrm{n}x}} + \frac{M_{\mathrm{u}y}}{\phi_b M_{\mathrm{n}y}}\right) \leq 1 \text{ for} \frac{P_{\mathrm{u}}}{\phi_c P_{\mathrm{n}}} < 0.2 \tag{6.3}$$

$$\frac{P_{\mathrm{u}}}{\phi_c P_{\mathrm{n}}} + \frac{8}{9}\left(\frac{M_{\mathrm{u}x}}{\phi_b M_{\mathrm{n}x}} + \frac{M_{\mathrm{u}y}}{\phi_b M_{\mathrm{n}y}}\right) \leq 1 \text{ for} \frac{P_{\mathrm{u}}}{\phi_c P_{\mathrm{n}}} \geq 0.2 \tag{6.4}$$

where X is the vector containing the design variables of the elements; h is the crown height; d_j is the jth allowable discrete value for the design variables, $h_{\min}$, $h_{\max}$, and h^* are the permitted minimum, maximum, and increment values of the crown height which in this chapter are taken as $D/20$, $D/2$, and 0.25 m, respectively, in which D is the diameter of the dome; ng is the number of design variables or the number of groups; $V(x)$ is the volume of the structure; L_i is the length of member i; δ_i is the displacement of node i; $\delta_{i\max}$ is the permitted displacement for the ith node; nn is the total number of nodes; ϕ_c is the resistance factor ($\phi_c = 0.9$ for tension, $\phi_c = 0.85$ for compression); ϕ_b is the flexural resistance reduction factor ($\phi_b = 0.9$); $M_{\mathrm{u}x}$ and $M_{\mathrm{u}y}$ are the required flexural strengths in the x- and y-directions, respectively; $M_{\mathrm{n}x}$ and $M_{\mathrm{n}y}$ are the nominal flexural strengths in the x- and y-directions, respectively; P_{u} is the required strength; and P_{n} denotes the nominal axial strength which is computed as

$$P_{\mathrm{n}} = A_{\mathrm{g}} F_{\mathrm{cr}} \tag{6.5}$$

where A_{g} is the gross area of a member and F_{cr} is calculated as follows:

$$F_{\mathrm{cr}} = \left(0.658^{\lambda_c{}^2}\right).f_{\mathrm{y}} \text{ for } \lambda_{\mathrm{c}} \leq 1.5 \tag{6.6}$$

$$F_{\mathrm{cr}} = \left(\frac{0.877}{\lambda_c^2}\right).f_{\mathrm{y}} \text{ for } \lambda_c > 1.5 \tag{6.7}$$

Here, f_{y} is the specified yield stress and λ_c is obtained from

$$\lambda_c = \frac{kl}{\pi r}\sqrt{\frac{f_{\mathrm{y}}}{E}} \tag{6.8}$$

where k is the effective length factor taken as 1; l is the length of a dome member; r is governing radius of gyration about the axis of buckling; and E is the modulus of elasticity. In Eq. (6.9), V_{u} *is* the factored service load shear, V_{n} is the nominal strength in shear, and φ_v represents the resistance factor for shear ($\varphi_v = 0.9$).

$$V_{\mathrm{u}} \leq \varphi_v V_{\mathrm{n}} \tag{6.9}$$

Table 6.2 The allowable steel pipe sections taken from LRFD AISC

	Type	Nominal diameter (in)	Weight per ft. (lb)	Area (in^2)	I (in^4)	S (in^3)	J (in^4)	Z (in^3)
1	ST	½	0.85	0.250	0.017	0.041	0.082	0.059
2	EST	½	1.09	0.320	0.020	0.048	0.096	0.072
3	ST	¾	1.13	0.333	0.037	0.071	0.142	0.100
4	EST	¾	1.47	0.433	0.045	0.085	0.170	0.125
5	ST	1	1.68	0.494	0.087	0.133	0.266	0.187
6	EST	1	2.17	0.639	0.106	0.161	0.322	0.233
7	ST	1¼	2.27	0.669	0.195	0.235	0.470	0.324
8	ST	1½	2.72	0.799	0.310	0.326	0.652	0.448
9	EST	1¼	3.00	0.881	0.242	0.291	0.582	0.414
10	EST	1½	3.63	1.07	0.666	0.561	1.122	0.761
11	ST	2	3.65	1.07	0.391	0.412	0.824	0.581
12	EST	2	5.02	1.48	0.868	0.731	1.462	1.02
13	ST	2½	5.79	1.70	1.53	1.06	2.12	1.45
14	ST	3	7.58	2.23	3.02	1.72	3.44	2.33
15	EST	2½	7.66	2.25	1.92	1.34	2.68	1.87
16	DEST	2	9.03	2.66	1.31	1.10	2.2	1.67
17	ST	3½	9.11	2.68	4.79	2.39	4.78	3.22
18	EST	3	10.25	3.02	3.89	2.23	4.46	3.08
19	ST	4	10.79	3.17	7.23	3.21	6.42	4.31
20	EST	3½	12.50	3.68	6.28	3.14	6.28	4.32
21	DEST	2½	13.69	4.03	2.87	2.00	4.00	3.04
22	ST	5	14.62	4.30	15.2	5.45	10.9	7.27
23	EST	4	14.98	4.41	9.61	4.27	8.54	5.85
24	DEST	3	18.58	5.47	5.99	3.42	6.84	5.12
25	ST	6	18.97	5.58	28.1	8.50	17.0	11.2
26	EST	5	20.78	6.11	20.7	7.43	14.86	10.1
27	DEST	4	27.54	8.10	15.3	6.79	13.58	9.97
28	ST	8	28.55	8.40	72.5	16.8	33.6	22.2
29	EST	6	28.57	8.40	40.5	12.2	24.4	16.6
30	DEST	5	38.59	11.3	33.6	12.1	24.2	17.5
31	ST	10	40.48	11.9	161	29.9	59.8	39.4
32	EST	8	43.39	12.8	106	24.5	49.0	33.0
33	ST	12	49.56	14.6	279	43.8	87.6	57.4
34	DEST	6	53.16	15.6	66.3	20.0	40.0	28.9
35	EST	10	54.74	16.1	212	39.4	78.8	52.6
36	EST	12	65.42	19.2	362	56.7	113.4	75.1
37	DEST	8	72.42	21.3	162	37.6	75.2	52.8

6.2.1 Nominal Strengths

Based on LRFD-AISC [14] specifications, the nominal tensile strength of a member is equal to:

$$P_{\mathrm{n}} = F_{\mathrm{y}} A_{\mathrm{g}} \tag{6.10}$$

where A_{g} is the gross section of the member.

The nominal compressive strength of a member is the smallest value obtained from the limit states of flexural buckling, torsional buckling, and flexural–torsional buckling. For members with compact or non-compact elements, the nominal compressive strength of the member for the limit state of flexural buckling is as follows:

$$P_{\mathrm{n}} = F_{\mathrm{cr}} A_{\mathrm{g}} \tag{6.11}$$

where F_{cr} is the critical stress based on flexural buckling of the member, calculated using Eqs. (6.6) and (6.7).

In the above equations, l is the laterally unbraced length of the member, K is the effective length factor, r is the governing radius of gyration about the axis of buckling, and E is the modulus of elasticity.

6.3 Metaheuristic Algorithm

This section introduces the enhanced colliding bodies optimization (ECBO) algorithm. First, a brief description of standard CBO based on the work of Kaveh and Mahdavi [15] is provided, and then the ECBO is introduced [16].

6.3.1 Colliding Bodies Optimization

The collision is a natural occurrence, and the CBO algorithm was developed based on this phenomenon. In this method, one object collides with the other object, and they move toward a minimum energy level (Figure 6.2). The CBO is simple in concept, does not depend on any internal parameter, and does not use memory for saving the best-so-far solutions. CBO algorithm, like other multi-agent methods, is a population-based metaheuristic algorithm. Each solution candidate X_i containing a number of variables (i.e., $X_i = \{x_{i,j}\}$) is considered as a colliding body (CB). The massed objects are divided into two equal groups, namely stationary and moving objects, where moving objects collide with stationary objects to improve their positions and push stationary objects toward better positions. After the collision,

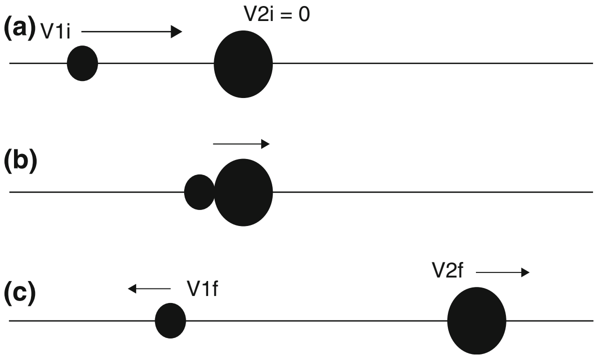


Fig. 6.2 Colliding of two bodies

the new position of colliding bodies is updated based on the new velocity by using the collision laws, and the lighter and heavier CBs move sharply and slowly, respectively.

6.3.2 Enhanced Colliding Bodies Optimization

A modified version of the CBO which is presented by Kaveh and Mahdavi [15] is ECBO, which improves the CBO to get faster and more reliable solutions [16]. The introduction of a memory increases the convergence speed of ECBO with respect to standard CBO. Furthermore, changing some components of colliding bodies will help ECBO to escape from local optima. The steps involved in ECBO are as follows:

Step 1: Initialization

The initial positions of all CBs are determined randomly in an *m*-dimensional search space according to

$$x_i^0 = x_{\min} + rand(x_{\max} - x_{\min}), \quad i = 1, 2, 3, \ldots, n \tag{6.12}$$

where x_i^0 is the initial solution vector of the *i*th CB. Here, $x_{\min}$ and $x_{\max}$ are the bounds of design variables, *rand* is a random vector for which each component is in the interval [0, 1], and *n* is the number of CBs.

Step 2: Defining mass

The value of mass for each CB is evaluated according to:

$$m_k = \frac{\frac{1}{fit(k)}}{\sum_{i=1}^{n} \frac{1}{fit(i)}}, \quad k = 1, 2, \ldots, n \tag{6.13}$$

Step 3: Saving

Considering a memory which saves some historically best CB vectors and their related mass and objective function values can make the algorithm performance better without increasing the computational cost [17]. Here a Colliding Memory (CM) is utilized to save a number of the best-so-far solutions. Therefore in this step, the solution vectors saved in CM are added to the population, and the same number of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: Creating groups

CBs are divided into two equal groups: (i) stationary group and (ii) moving group. The pairs of CBs are shown in Fig. 6.2.

Step 5: Criteria before the collision

The velocity of stationary bodies before collision is zero, i.e.,

$$v_i = 0,\ i = 1, \ldots, \frac{n}{2} \tag{6.14}$$

Moving objects move toward stationary objects, and their velocities before collision are calculated by

$$v_i = x_{i-\frac{n}{2}} - x_i,\ i = \frac{n}{2} + 1, \ldots, n \tag{6.15}$$

Step 6: Criteria after the collision

The velocities of stationary and moving bodies are evaluated using Eqs. (6.16) and (6.17), respectively.

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}} i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{6.16}$$

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i-\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} i = 1, 2, \ldots, \frac{n}{2} \tag{6.17}$$

Step 7: Updating CBs

The new position of each CB is calculated by the following equations:

$$\varepsilon = 1 - \frac{iter}{iter_{\max}} \tag{6.18}$$

$$x_i^{\text{new}} = x_{i-\frac{n}{2}} + rand\ 0\ v_i', \quad i = \frac{n}{2} + 1, \ldots, n \tag{6.19}$$

$$x_i^{\mathrm{new}} = x_i + rand\,0\,v_i', \quad i = 1, 2, \ldots, \frac{n}{2} \tag{6.20}$$

Step 8: Escape from local optima

Metaheuristic algorithms should have the ability to escape from the trap when agents get close to a local optimum. In ECBO, a parameter *Pro* within (0, 1) is introduced, which specifies whether a component of each CB must be changed or not. For each colliding body, *Pro* is compared with rn_i (i = 1, 2... *n*) which is a random number uniformly distributed within (0, 1). If $rn_i < Pro$, one dimension of the *i*th CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + random.\left(x_{j,\max} - x_{j,\min}\right) \tag{6.21}$$

where x_{ij} is the *j*th variable of the *i*th CB. $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the *j*th variable, respectively. In order to protect the structures of CBs, only one dimension is changed. This mechanism provides opportunities for the CBs to move all over the search space, thus providing better diversity.

Step 9: Terminating condition check

The optimization process is terminated after a fixed number of iterations. If this criterion is not satisfied, go to Step 2 for a new round of iteration.

For further details, the reader may refer to Kaveh and Mahdavi [18].

6.4 Configuration of Single-Layer Lamella Dome, Suspen-Dome, and Double-Layer Dome

6.4.1 Configuration of Single-Layer Lamella Dome with Rigid-Jointed Connections

Topology of a single-layer Lamella dome is shown in Fig. 6.3. For all domes, including the Lamella dome, it is possible to generate the geometric structural data if four parameters consisting of the diameter (*D*) of the dome, the total number of rings, the total number of joints, and the height of the crown (*h*) are known. When the geometry of a dome is formed according to mentioned parameters, the topology of domes can be obtained. The topology contains the total number of members, member incidences, and total number of joints of the domes. The distances between the rings in the dome on the meridian line are generally made to be equal. It can be easily seen from Fig. 6.4a and b that all the joints are located with equal distances on a particular ring in both domes. The top joint which is the dome's crown is numbered as first joint (joint number 1). The first joint on the first ring is numbered as joint 2 in each dome type. In Lamella dome, there is the same number of joints on

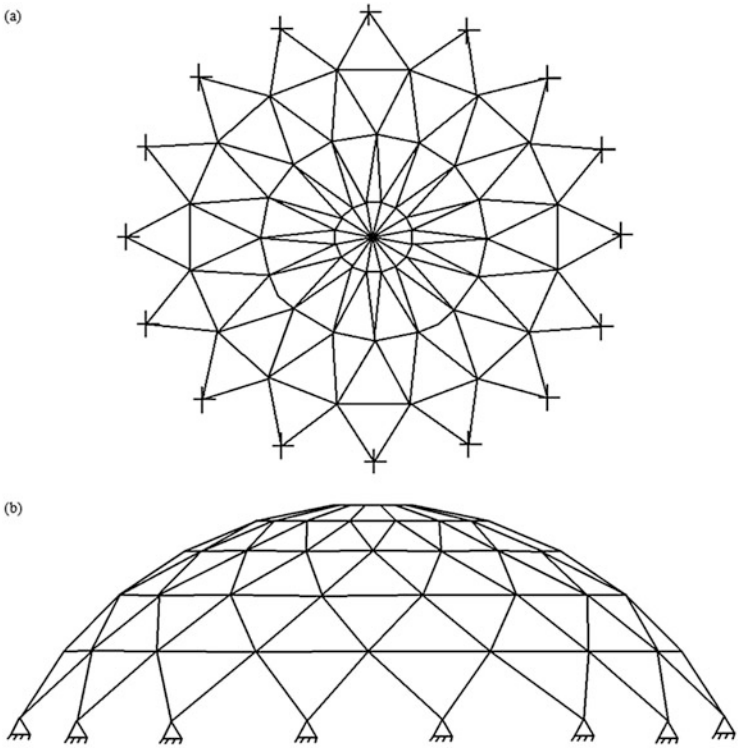


Fig. 6.3 Schematic of a Lamella dome. (**a**) Plan view and (**b**) side view

each ring. The joint numbers of all the other first joints of other rings are computed from the following equation:

$$J_{r1} + (r - 1) \times 10 \tag{6.22}$$

where r is the ring number and J_{r1} is the first joint number of the first ring, namely 2 for Lamella dome. It is worthwhile to mention that all of the first joints of the odd-numbered rings (ring 1 and ring 3) are located on the radius that makes angle of 16° with the x-axis and, similarly, the first joints of the even-numbered rings (ring 2) are located on the intersection points of that ring and the x-axis in Lamella dome. First member is taken as one and connects joint 1 to joint 2 which makes an angle of (360/Nn)° with x-axis in Lamella dome. For the first ring group, the start node for all elements is the joint number 1 and the end nodes are those on the first ring. The start and end nodes of ring elements can be obtained using Eqs. (6.24) and (6.25), and for other rings (2 and 3), this process is repeated and all the member incidences are similar.

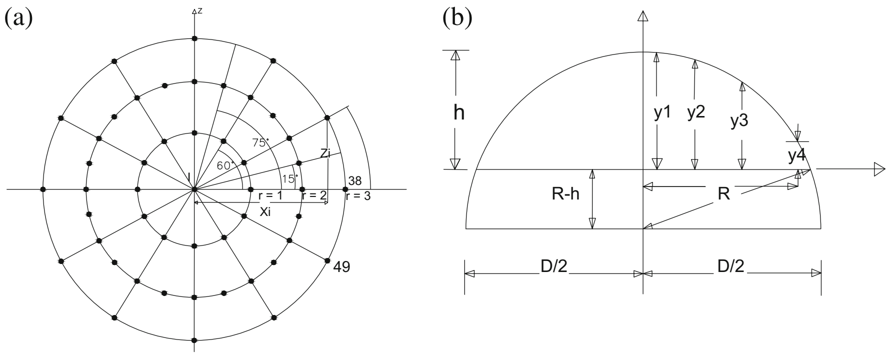


Fig. 6.4 (**a**) Joint coordinates of single-layer Lamella dome and (**b**) side view coordinate

$$\begin{cases} x_i = \dfrac{D}{2Nr}\cos\left(\dfrac{360}{4n_i}\left(i - \displaystyle\sum_{j=1}^{i-1}(4n_j - 1)\right)\right) \\ y_i = \dfrac{D}{2Nr}\sin\left(\dfrac{360}{4n_i}\left(i - \displaystyle\sum_{j=1}^{i-1}(4n_j - 1)\right)\right) \quad n_i = 1, 2, \ldots, Nr - 1;\;\; i: \text{Joint number} \\ z_i = \sqrt{\left(R^2 - \dfrac{{n_i}^2 D^2}{4Nr^2}\right)} - (R - h) \end{cases} \tag{6.23}$$

$$\begin{cases} I = 10 \times (n_i - 1) + j + 1 \\ J = 10 \times (n_i - 1) + j + 2 \end{cases} \quad (j = 1, 2, 3, \ldots, 9);\;\; n_i = 1, 2, \ldots, Nr - 1 \tag{6.24}$$

$$\begin{cases} I = 10 \times (n_i - 1) + 2 \\ J = 10 \times n_i + 1 \end{cases} \quad n_i = 1, 2, \ldots, Nr - 1 \tag{6.25}$$

Computation of x, y, and z coordinates of a joint on the domes requires the angle between the line that connects the considered joint to the joint placed at the crown of dome (joint number 1) and the x-axis as shown in Fig. 6.9. For Lamella dome, the mentioned angle can be computed by Eqs. (6.26) and (6.27) for the odd- and even-numbered rings, respectively:

$$a_i = \frac{360}{2Nn} \tag{6.26}$$

$$a_i = \frac{360}{2Nn}\left(i - j_{r,1}\right) \tag{6.27}$$

r is the ring number that joint i is placed on it and j is the first joint number on the ring number r which is on the x-axis. The members group which is used in Tables is mentioned in the following sentences. For Lamella domes, the ribbed members between the crown and the first ring are group 1, the diagonal members between fist ring and second ring are group 2, and the diagonal members between second ring and third ring are group 3. The members on the first ring are group 4, and the members on the second ring are group 5.

6.4.2 Configuration of Lamella Suspen-Domes

The lower tensegric system is detached from the upper single-layer dome as an independent system. In the lower tensegric system, the strands and the vertical struts are hinged in the joints. The tensegrity system is constructed of four rings of hoop steel cables, radial steel cables, and struts at the lower part of model. The cables are tension-only elements and the vertical struts are also compression elements.

The upper single-layer Lamella dome is arranged as a triangle circular truss. The struts which are the web members of suspen-dome and bending members that are the elements of single-layer Lamella dome are circular standard steel tubes for which the sections are listed in Table 6.1.

As it was mentioned before, the suspen-dome is constructed by combining tensegrity system (cable-struts) and a single-layer reticulated dome. The configuration of single-layer Lamella dome is explained in the previous part. As can be seen from Fig. 6.5, the tensegrity system is constructed of hoop cable, radial cable, and compression struts. The topology of tension-only cables, which are called radial and hoop cables, is the same as the upper single-layer reticulated dome. Therefore, the suitable configuration of tensegrity system depends on its upper single-layer dome.

The suspen-dome which is discussed in this study uses the configuration of a Lamella dome as the upper part. Therefore, the configuration of tensegrity system should be obtained using the configuration of the Lamella dome. The current tensegrity system is connected to the rings 3, 4, and 5 of a single-layer Lamella dome by vertical struts elements.

Computation of x and y coordinates of a joint on tensegrity system requires the angle between the line that connects the considered joint to the joint placed at the crown of dome (joint number 1) and the x-axis. For Lamella suspen-dome, the mentioned angle can be computed by Eqs. (6.26) and (6.27) for the odd- and even-numbered rings, respectively.

Computation of z coordinates of a joint on tensegrity system can be obtained using the following equation:

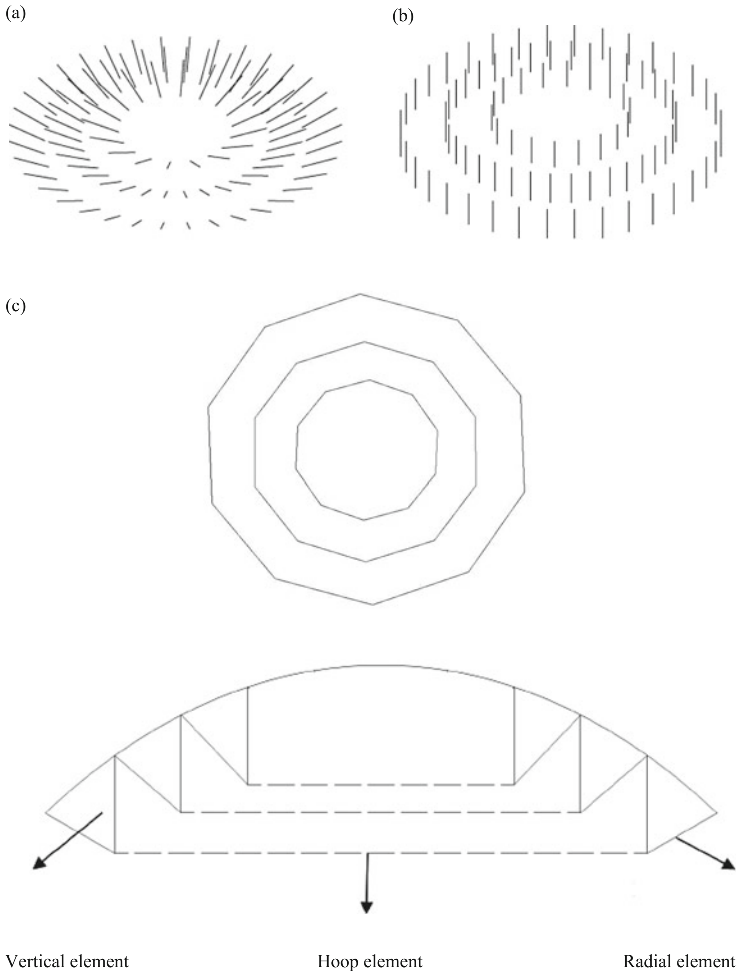


Fig. 6.5 Configuration of the double-layer dome or the tensegrity part of the suspen-dome. (**a**) Radial elements of the double-layer dome and suspen-dome. (**b**) Vertical elements of the double-layer dome and suspen-dome. (**c**) Hoop elements of the double-layer dome and suspen-dome

$$z_i = \sqrt{\left(R^2 - \frac{n_i^2 D^2}{4Nr^2}\right)} - (R - h) - H\text{hoop}(i) \tag{6.28}$$

where Hhoop is the distance between the upper single-layer Lamella dome. In other words, at the same time, it is the length of struts in the tensegrity structure.

For Lamella suspen-dome, the diagonal members between the crown and the first ring are group 1, the diagonal members between first ring and the second ring are group 2, the diagonal members between second ring and third ring are group 3, and the first ring, second ring, and third ring are groups 4, 5, and 6, respectively.

Then, after third ring each diagonal member and its related ring are numbered, respectively. For example, if group 7 is the diagonal member between the ring 3 and 4, then the group 8 is the fourth ring of the dome.

6.4.3 Configuration of Double-Layer Lamella Dome

The lower grid system is detached from the upper single-layer dome as an independent system. In the lower system, the steel elements and the vertical struts are hinged in the joints. The lower layer is constructed of four rings of hoop steel elements, radial steel elements, and vertical elements at the lower part of model where these can be subjected to tension and pressure, contrary to the cables\strands in suspen-dome.

In double-layer domes, the upper single-layer Lamella dome is arranged as a triangle circular truss. The vertical elements which are the web members of the double-layer dome and bending members which are the elements of single-layer Lamella dome are circular standard steel tubes.

As it mentioned, the double-layer dome is constructed by combining two layers of the grids which are lower grid (steel tube strut) and single-layer reticulated dome. The configuration of a single-layer Lamella dome is explained in previous part. As can be seen from Fig. 6.5, the configuration of a double-layer dome is chosen exactly the same as a suspen-dome.

6.5 Convergence Curves of the Metaheuristic Algorithms

6.5.1 Comparison of the Convergence Curves of PSO, CBO, and ECBO

To investigate the efficiency of different algorithms, the convergence curves of three popular algorithms for dome structures are obtained in this section. Figure 6.6 shows the convergence curves of the PSO, CBO, and ECBO algorithms which are useful metaheuristic methods for optimal design of various structures, and in this study, these are used for a single-layer dome with six variables. CBO and ECBO methods are parameter independent, but PSO depends on some parameters such as C_1, W, and C_2 which should be set before starting analysis. Also Fig. 6.6 shows that the design found by ECBO is lighter than those found by CBO and ECBO at the same number of analysis. As another observatory, it can be seen that the convergence rates of the ECBO and CBO algorithm are better than that of the PSO. Therefore, the results obtained for this example are the main reason for choosing ECBO in subsequent numerical models of this chapter. Therefore, optimization

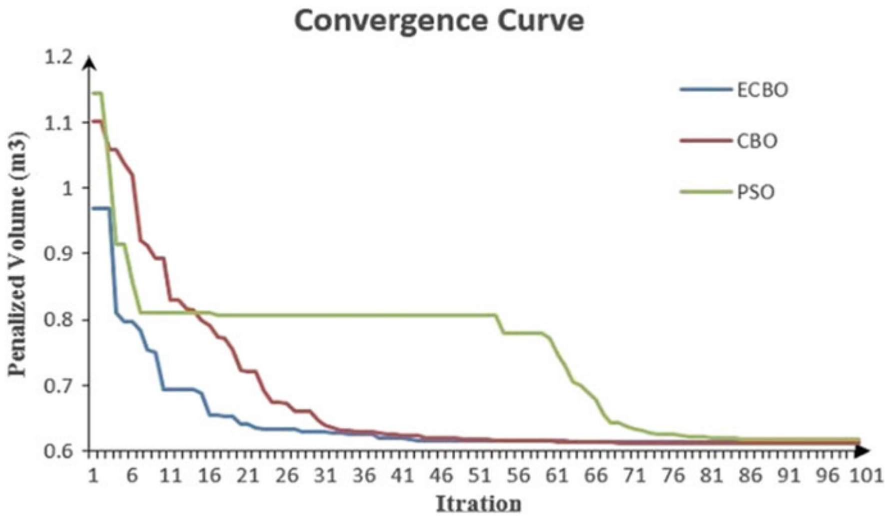


Fig. 6.6 Convergence curves for a single-layer dome

process is performed via ECBO algorithm to demonstrate the effectiveness and robustness of the ECBO in creating optimal design of different domes.

Nonlinear structural behavior can originate from geometrical or material nonlinearity. If a structure experiences large deformations, its changing geometric configuration can be the cause of nonlinearly. In this study, a finite elements model based on geometrical nonlinear analysis of different dome systems consisting of a double-layer dome, a suspen-dome, and a single-layer dome with rigid connections is presented by OPENSEES. In this model, a 3-D uniaxial co-rotational truss element is utilized.

A significant criterion governing the design of domes is the requirement of full triangulation of the geometry. Also this is one of the reasons for choosing Lamella dome. Since these types of structures have a high stiffness in all directions and are kinematically stable, triangulation must be used in the design of domes unless making rigid connection designs. Therefore, for pin-connected dome design, the latticed shell must be formed from the triangular units.

In this study, the different systems of the domes described in the previous sections are optimized utilizing the ECBO. The modulus of elasticity for the steel is taken as 205 kN/mm^2. The limitations imposed on the joint displacements are 28 mm in the *z*-direction and 33 mm in the *x*- and *y*-directions for the 1st, 2nd, and 3rd nodes, respectively (Table 6.3).

To investigate the real performance of these domes, they are subjected to dead and snow loads according to real load on roof of the dome. The design dead load is established on the basis of the actual loads like the weight of various accessories and cladding that may be expected to act on the dome structure. The dead, snow, and wind loads of 200 N/m^2, 800 N/m^2, and 200 N/m^2, respectively, are considered. The loads are converted into equivalent point loads for each joint for the sake of

Table 6.3 Displacement restrictions of single-layer domes

	Displacement limitations (mm)					
	X-direction		*Y*-direction		*Z*-direction	
Joint no	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound	Lower bound
1	–	–	–	–	28	−28
2	33	−33	33	−33	28	−28
3	33	−33	33	−33	28	−28

simplicity. For this conversion, distributed load is multiplied by surface area of dome.

The volume of the dome structures can be considered as a function of the average cross-sectional area of the elements ($\overline{A}$) and the sum of the element lengths, expressed as:

$$V(X) = \overline{A} \cdot \sum_{i=1}^{nm} L_i \tag{6.29}$$

6.6 Comparison of Different Mechanical Systems of Domes

As mentioned in the previous section, a finite elements model based on geometrical nonlinear analysis of different systems of domes which are double-layer dome, suspen-dome, and single-layer dome with rigid connections is presented.

Rigid connections are often employed in the construction of long span single-layer domes, since the load capacity of pin-connected single-layer domes is not sufficient. However, pin connections are often used in double-layer lattice domes or suspen-domes, because the additional layer can make a more stiff structure compared to single-layer latticed dome structures.

Also by using the tensegrity system, the suspen-dome structure performs like a double-layer dome structure. The tensegrity system is constructed of cables and struts stiffening the suspen-dome structure. The stiffness comes from the opposite force to the external gravity load. Therefore, it is logical to use pin-jointed connections in the construction of the suspen-dome system.

6.6.1 Optimal Design of Single-Layer Lamella Dome with Rigid Joints

In this section, a single-layer Lamella dome is optimized using the ECBO algorithm (Fig. 6.7). In this case, the dead and snow loads are considered for Lamella domes

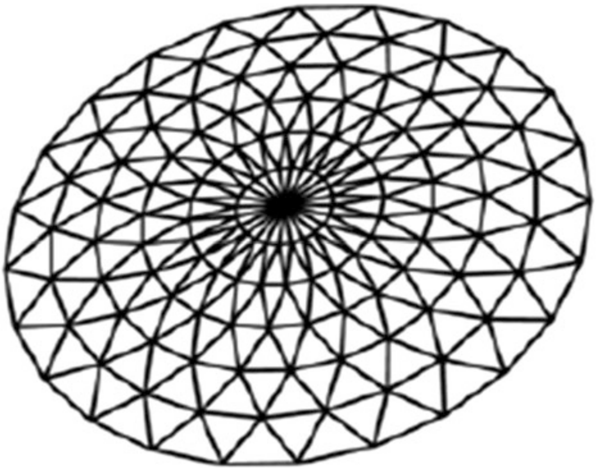


Fig. 6.7 Schematic of a single-layer Lamella dome

to investigate the real behavior and to obtain optimum geometry of dome under these loading conditions. The dome structure is subjected to 0.8 kN/m^2 of dead load, 0.2 kN/m^2 of live load, and 0.2 kN/m^2 of basic wind pressure.

The number of rings is considered as 6 under this loading condition. The results of the design are shown in Table 6.4. Due to the existence of a noticeable value of dead/snow loading on each joint, the cross sections are obtained close to each other. As can be seen, the optimal design of dome is found obtaining 5 m height for the single-layer dome. For the dome with lower number of rings and lower number of nodes, because of having the least number of joints and considerable amount of load value on each joint, higher volume for dome is obtained and higher height is chosen to provide higher stability. Because of this reason, the number of joints on each ring in this study is chosen equal to 12. Also when the number of joints is increased, the dead and snow loads are distributed among more joints.

6.6.2 Optimal Design of Lamella Suspen-Dome with Pin-Jointed and Rigid-Jointed Connections

The six-ring suspen-dome is employed as an example to illustrate this idea. The top part of the model is a single-layer lattice dome, which has 6 rings with 12 joints in each ring. The single-layer Lamella dome which is a popular type of latticed dome consists of steel tube beams that are fixed at both ends to suspen-dome with rigid-jointed topmost layer and steel tube trusses for suspen-dome with rigid-jointed topmost layer.

Its design tensile strength is 240 MPa. The computational model is a suspen-dome having a span of 40 m. The material of cables is made of high strength wire, the technical parameters of these are provided in Table 6.1. These dome structures

Table 6.4 Optimal design of single-layer Lamella dome with rigid-jointed connections for Lamella dome using ECBO algorithm

		ECBO algorithm
		Section
Number of rings		6
Optimum tubular	Group 1	PIPST (8)
Section designations	Group 2	PIPST (8)
	Group 3	PIPST (8)
	Group 4	PIPST (8)
	Group 5	PIPST (8)
	Group 6	PIPST (8)
	Group 7	PIPST (8)
	Group 8	PIPST (8)
	Group 9	PIPST (6)
	Group 10	PIPST (6)
	Group 11	PIPST (6)
	Group 12	PIPST (6)
Height of crown (m)		5.00
Maximum displacement (cm)		2.75
$\sum l_I$ (m)		982.52
$\bar{A}$ (cm^2)		49.34
Maximum strength ratio		27.01
Volume (m^3)		2.14

are subjected to 0.8 kN/m^2 of dead load, 0.2 kN/m^2 of live load, and 0.2 kN/m^2 of basic wind pressure.

In construction of the suspen-dome, the tensegrity system is constructed of three rings of hoop cables, radial cables, and struts at the lower part of model. The tensegrity system is connected to the rings 3, 4, and 5 of a single-layer Lamella dome by vertical struts elements. For example, the struts of group 1 are connected to the joints which are located in the third ring of the single-layer Lamella dome. The struts are compression elements and have hinged connections on both ends; their sections are circular steel tubes.

The tensegrity system is constructed of cables and struts stiffening the suspen-dome structure (Figure 6.8). This also helps the suspen-dome to work like a double-layer dome. Therefore, it is interesting to compare the optimum results of suspen-dome with double-layer dome which is discussed in this study.

It is worthwhile to mention that the applied optimum prestressed force of tensegrity system (radial and hoop cable) must be large enough to prevent cable slack, but not so large as to make the struts buckle or induce very large opposite moment compared to moment induced by external loads.

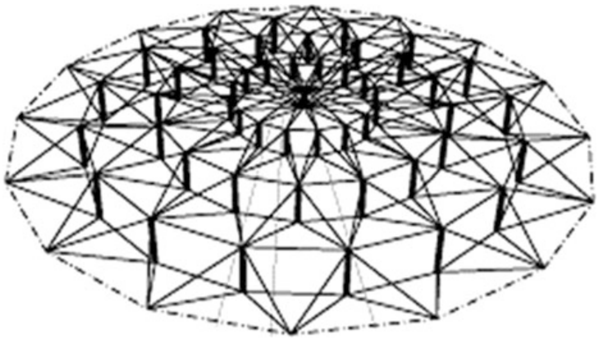


Fig. 6.8 Configuration of a suspen-dome

6.6.3 Optimal Design of Double-Layer Lamella Domes

As it was mentioned in the previous sections, the domes having rigid connections are often used in the construction of long span single-layer domes, because the load capacity and stiffness of pin-connected single-layer domes is very low. However, pin connections are often used in double-layer lattice domes, because the additional layer can make a more stiff structure compared to single-layer latticed dome structures. For this reason, double-layer dome is studied here to compare it with other systems of domes like single-layer and suspen-domes.

It is logical to use pin-jointed connections in the construction of the double-layer dome systems. The upper layer of dome is constructed as a single-layer Lamella dome. The lower system which is the second layer of dome is constructed utilizing radial and vertical elements which stiffen the single layer of the dome structure. The stiffness is provided by the second layer of dome. On the other hand, the moment that is induced by the external load is sustained by two layers of the dome. This also shows that the maximum bending moment of a double-layer dome which is balanced by two layers of dome is decreased. Therefore, using double-layer dome has two advantages consisting of reducing the element stresses and joint displacements of the structure.

The six-ring double-layer dome is employed as an example to compare the results with those of the previous example (Fig. 6.9). The computational model is a double-layer dome having a span of 40 m. The top part of the model is a single-layer Lamella dome, which has 6 rings and 12 joints in each ring. Both single-layer Lamella domes consist of steel tube beams that are hinged at both ends and constructed of steel tube trusses for double-layer dome which are made of pin-jointed connections.

For construction of the double-layer dome, the lower layer (second layer) consists of three rings of hoop elements, radial elements, and vertical elements at the lower part of model. The second layer of double-layer dome is connected to the rings 3, 4, and 5 of the single-layer Lamella dome by vertical elements. For example, the vertical elements of group 1 are connected to the joints which are

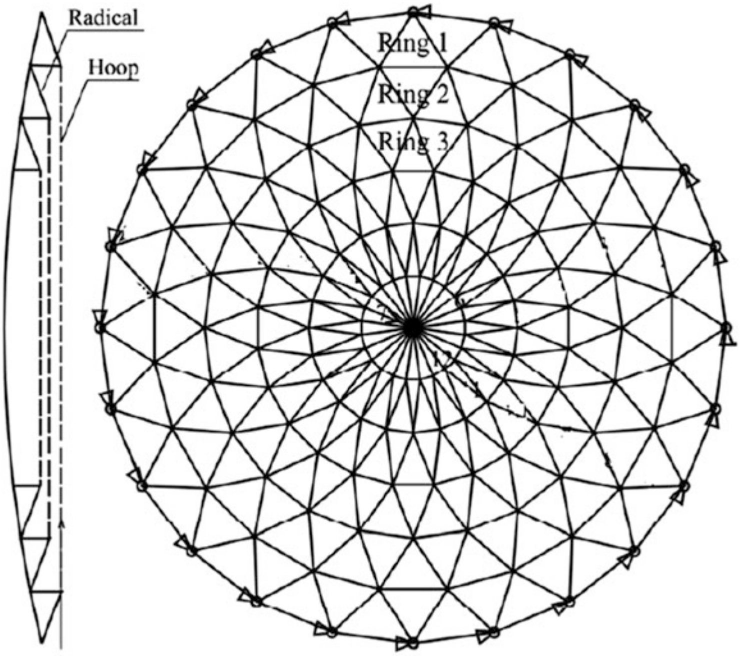


Fig. 6.9 Schematic of a double-layer Lamella dome

located in the third ring of the single-layer Lamella dome. The vertical elements are compression elements and are standard steel elements having hinged connections on both ends; its sections are circular steel tubes; also the hoop elements and radial elements are standard steel elements which induce the main difference between the standard double-layer dome and suspen-dome discussed in the previous section. Same as single-layer Lamella dome and suspen-dome, the double-layer dome structure is subjected to 0.8 kN/m^2 of dead load, 0.2 kN/m^2 of live load, and 0.2 kN/m^2 of basic wind pressure.

6.6.4 Results

In this case of loading, the wind, dead, and snow loads are applied on all the domes and the diameter is 20 m. It can be seen from Table 6.5 that the capacity of elements in the single-layer Lamella dome with rigid joints is approximately 27 % of the capacity of material which shows that material is overdesigned or on the other hand the stress ratio of material does not control the design and the displacements govern the design of single-layer Lamella dome with rigid joints. The maximum value of displacement for this dome is equal to 2.75 cm and nearly the same as 2.80 which is the maximum allowable displacement value of design. Therefore, displacement

Table 6.5 Optimal design of upper single-layer dome with pin-jointed connections for Lamella suspen-dome using the ECBO algorithm

		ECBO algorithm
		Pin-jointed
Number of rings		6
Optimum tubular	Group 1	PIPST (3)
Section designations	Group 2	PIPST (8)
	Group 3	PIPST (10)
	Group 4	PIPST (10)
	Group 5	PIPST (8)
	Group 6	PIPST (10)
	Group 7	PIPST (10)
	Group 8	PIPST (8)
	Group 9	PIPST (10)
	Group 10	PIPST (8)
	Group 11	PIPST (10)
	Group 12	PIPST (8)
Height of crown (m)		4.50
Maximum displacement (cm)		2.79
$\sum l_I$ (m)		979.37
$\bar{A}$ (cm^2)		54.64
Maximum strength ratio		44.16
Volume (m^3)		2.46

constraints are more active than the stress constraints for suspen-domes. The optimum volume of single-layer dome is obtained 2.14 m^3. The optimum height, the total length of elements, and average cross-sectional area of single-layer dome are obtained 5, 982.52, and 54.64, respectively.

It can be seen from Table 6.6 that using the capacity of elements in suspen-dome with rigid joints is approximately 27 % more than the suspen-dome with pin-jointed connections. Displacement constraints are more active than the stress constraints for suspen-domes. The length of the struts which are connected to rings number 3, 4, and 5 are obtained as 1.5, 1, and 0.5 for domes, respectively. Therefore, the least area sections are obtained for struts elements. When the tensegrity systems of suspen-domes are compared, according to their optimum geometry design, it can be seen that the cable system of the suspen-dome with rigid-jointed upper layer is more economical (Tables 6.6, 6.7, and 6.8).

When these suspen-domes are compared, it can be seen that the suspen-dome with rigid-jointed topmost layer provides a lighter design. For example, the optimum volumes of the topmost layers for the domes with pin-jointed and rigid-jointed connections are 2.46 m^3 and 1.86 m^3, respectively, which clearly shows that in suspen-domes, the topmost layer with pin-jointed connections is 24 % heavier than the topmost layer with rigid-jointed connections.

Table 6.6 Optimal design of tensegrity system of the suspen-dome with upper layer pin-jointed and rigid-jointed connections obtained using the ECBO algorithm

		ECBO algorithm
		Pin-jointed
Number of hoop cables		3
Cable and section	Hoop 1	Cable (11.3)
	Hoop 2	Cable (15.2)
	Hoop 3	Cable (9.6 + 15.7)
	Radial 1	Cable (8)
	Radial 2	Cable (11.3)
	Radial 3	Cable (15.2)
	Strut 1	PIPST (1/2)
	Strut 2	PIPST (1/2)
	Strut 3	PIPST (3/4)
Initial strain		0.00050
$\sum l_c$(m)		562.39
$\sum l_s$(m)		30
Hoop cable volume (m^3)		0.72
Radial cable volume (m^3)		0.031
Strut volume (m^3)		0.006

Table 6.7 Optimal design of upper single-layer dome with pin-jointed and rigid-jointed connections for Lamella suspen-dome using the ECBO algorithm

		ECBO algorithm
		Rigid-jointed
Number of rings		6
Optimum tubular	Group 1	PIPST (3)
Section designations	Group 2	PIPST (8)
	Group 3	PIPST (8)
	Group 4	PIPST (8)
	Group 5	PIPST (4)
	Group 6	PIPST (8)
	Group 7	PIPST (10)
	Group 8	PIPST (5)
	Group 9	PIPST (10)
	Group 10	PIPST (8)
	Group 11	PIPST (5)
	Group 12	PIPST (4)
Height of crown (m)		3.50
Maximum displacement (cm)		2.54
$\sum l_l$ (m)		976.54
$\bar{A}$ (cm^2)		39.28
Maximum strength ratio		75.07
Volume (m^3)		1.86

Table 6.8 Optimal design of tensegrity system of the suspen-dome with upper layer pin-jointed and rigid-jointed connections obtained using the ECBO algorithm

		ECBO algorithm
		Rigid-jointed
Number of hoop cables		3
Cable and section	Hoop 1	Cable (9.6)
	Hoop 2	Cable (15.2)
	Hoop 3	Cable (15.7 + 8)
	Radial 1	Cable (8)
	Radial 2	Cable (11.3)
	Radial 3	Cable (12.5)
	Strut 1	PIPST (1/2)
	Strut 2	PIPST (1/2)
	Strut 3	PIPST (3/4)
Initial strain		0.00041
$\sum l_c$(m)		562.39
$\sum l_s$(m)		30
Hoop cable volume (m^3)		0.6343
Radial cable volume (m^3)		0.027
Strut volume (m^3)		0.006

In geometry optimization of suspen-dome, the optimum height, the total length of the elements, average cross-sectional area, and maximum strength ratio of suspen-dome with rigid-jointed connections are obtained 3.50, 976.54, 39.28, and 75.07, respectively (Table 6.6). Also the optimum height, the total length of elements, average cross-sectional area, and maximum strength ratio of suspen-dome with pin connections are obtained 4.50, 979.37, 54.64, and 44.16, respectively.

It can be seen from Table 6.9 that the double-layer dome discussed in this study which has pin-jointed connections between elements performs as a truss structure. The optimum volume of double-layer dome is obtained 2.11 m^3. Also, the optimum height, the total length of elements, average cross-sectional area, and maximum strength ratio of the single-layer suspen-dome are obtained as 1577.04, 56.57, and 38.54, respectively. The length of the vertical elements in the double-layer dome which are connected to rings number 3, 4, and 5 are obtained as 1.5, 1, and 0.5 for domes, respectively. Also the least cross-sectional areas are obtained for vertical elements.

In conclusion, it can be seen from Tables 6.4, 6.5, 6.7, and 6.9 that the most optimum weight of the steel elements, between four discussed models in this study, is obtained for the suspen-dome with rigid-jointed connections. But it is essential to mention that considering the weight of strands which are in the tensegrity system of suspen-dome, the total weight of suspen-dome can be changeable. Apart from the weight of tensegrity system, the weight of suspen-dome with rigid-jointed connections is 25 %, 13.44 %, and 15 % lighter than suspen-dome with pin joints, single-layer Lamella dome, and double-layer dome, respectively.

Table 6.9 Optimal design of double-layer dome with pin-jointed connections for Lamella suspen-dome using ECBO algorithm

		ECBO algorithm
		Rigid-jointed
Number of rings		6
Optimum tubular	Group 1	PIPST (3)
Section designations	Group 2	PIPST (6)
	Group 3	PIPST (6)
	Group 4	PIPST (8)
	Group 5	PIPST (6)
	Group 6	PIPST (10)
	Group 7	PIPST (10)
	Group 8	PIPST (10)
	Group 9	PIPST (10)
	Group 10	PIPST (10)
	Group 11	PIPST (10)
	Group 12	PIPST (10)
	Group 13	PIPST (8)
	Group 14	PIPST (8)
	Group 15	PIPST (8)
	Group 16	PIPST (10)
	Group 17	PIPST (8)
	Group 18	PIPST (6)
	Group 19	PIPST (8)
	Group 20	PIPST (8)
	Group 21	PIPST (6)
Height of crown (m)		5.00
Maximum displacement (cm)		2.80
$\sum l_l$ (m)		1577.04
$\bar{A}$ (cm^2)		56.57
Maximum strength ratio		38.54
Volume (m^3)		2.11

As an another observatory, it can be seen that the double-layer dome has acceptable performance under gravity loading and with considering the weight of strands for rigid-jointed suspen-dome, double-layer dome can be comparable with suspen-dome and may obtain one of the most optimum weights among the compared systems. It should be mentioned that in double-layer domes because of having pin connections, the structure is similar to a truss and also is less stiff than other dome structures having rigid-jointed connections. Therefore, the displacement constraints control the design. On the contrary, the pin-jointed suspen-domes not only obtained the heaviest weight among others but also used the least amount of the capacity of the elements among other mechanical systems of domes.

6.7 Concluding Remarks

In this chapter, the ECBO is utilized for optimal design of different mechanical systems of domes with pin and rigid-jointed connections. The different mechanical systems of domes contain single-layer dome, double-layer dome and suspen-dome with pin and rigid-jointed connections. The height of the domes, the length of the strut/vertical elements, cables' initial strain, the cross-sectional areas of the cables, and the cross-sectional area of steel members are considered as design variables and the volume of the entire structure is taken as the objective function. The optimization method used in the chapter is based on the enhanced colliding bodies optimization algorithm. In this chapter, sizing and geometry of domes is presented. For sizing optimization, the optimum steel section designations for the members of domes are chosen from Table 6.2 and implemented in the design constraints from LRFD-AISC.

A simple approach is presented to calculate the joint coordinates and specify the elements to determine the configuration of single-layer Lamella domes and the corresponding suspen-domes which are spatial prestressed structures with complex mechanical characteristics. First, the joint coordinates are calculated, and then using some simple relationships, the steel elements, struts, and cables are constructed. This method considers not only the strength of steel components and cables for optimal design as constraints but also considers the stability of the steel members and controls the displacements of the overall structure.

An investigation on the efficiency of the ECBO method in optimal design of single-layer domes is performed. In the suspen-dome structure and double-layer dome, the tensegrity system and second layer significantly reduced the stresses and the displacements of dome structure, respectively. By using the tensegrity system, the suspen-dome structure performs like a double-layer dome structure. Therefore, it is logical to use pin-connected joints in the construction of the suspen-dome systems, and it is essential to compare them under the same conditions of loading as discussed in this chapter. However, it is seen that the suspen-dome with upper layer rigid joints offers a more economical design.

The ECBO method which is one of the recent additions to stochastic search techniques of numerical optimization, is used to obtain the solution of the numerical examples. It can be seen that the design examples of this study and the enhanced colliding body method can be used for finding the solution of geometry and sizing optimization of different mechanical system of domes such as double-layer domes, suspend-domes which has complex mechanical structure, and single-layer domes.

As the future work, the cost of joints can also be added to the optimization formulations. A comparative study can also be performed for other types of double-layer and suspen-domes that are not studied in this chapter. Also optimum dynamic analysis and design of different types of domes can be compared under seismic loads.

References

1. Kaveh A, Rezaei M (2016) Sizing and geometry optimization of different mechanical systems of domes via enhanced colliding body algorithm. Neural Comput Appl (submitted)
2. McKenna F, Fenves G (2001) The opensees command language manual. University of California, Berkeley (opensees.ce.berkeley.edu)
3. The Language of Technical Computing (2006) MATLAB. Math Works Inc., Natick, MA
4. Kawaguchi M, Abe M, Tatemichi I (1999) Design, tests and realization of suspen-dome system. J Int Assoc Shell Spatial Struct 40(3):179–192
5. Kitipornchai S, Wenjiang K, Lam H-F, Albermani F (2005) Factors affecting the design and construction of Lamella suspen-dome systems. J Constr Steel Res 61:764–785
6. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 39:18–27
7. Kaveh A, Talatahari S (2009) Size optimization of space trusses using Big Bang–Big Crunch algorithm. Comput Struct 87(17–18):1129–1140
8. Saka MP (2007) Optimum geometry design of geodesic domes using harmony search algorithm. Adv Struct Eng 10(6):595–606
9. Kaveh A, Talatahari S (2010) Optimal design of Schwedler and ribbed domes via hybrid Big Bang–Big Crunch algorithm. J Constr Steel Res 66:412–419
10. Nie GB, Fan F, Zhi XD (2013) Test on the suspended dome structure and joints of Dalian Gymnasium. Adv Struct Eng 16(3):467–486
11. Kamyab R, Salajegheh E (2013) Size optimization of nonlinear scallop domes by an enhanced particle swarm algorithm. Int J Civil Eng 11(2):77–89
12. Kaveh A, Rezaei M (2016) Topology and geometry optimization of single-layer domes utilizing CBO and ECBO. Sci Iranica 23(2):535–547
13. Kaveh A, Rezaei (2015) Topology and geometry optimization of different types of domes using ECBO. Adv Comput Des 1(1):1–25
14. American Institute of Steel Construction (AISC) (1989) Manual of steel construction allowable stress design, 9th edn. AISC, Chicago, IL
15. Kaveh A, Mahdavi VR (2015) Colliding bodies optimization; extensions and applications. Springer, Wien
16. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Struct Eng 77:66–75
17. Kaveh A, Ilchi Ghazaan M (2014) Computer codes for colliding bodies optimization and its enhanced version. Int J Optim Civil Eng 4(3):321–332
18. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization method for optimal design of truss structures with continuous variables. Adv Struct Eng 70:1–12

Chapter 7
Simultaneous Shape–Size Optimization of Single-Layer Barrel Vaults Using an Improved Magnetic Charged System Search Algorithm

7.1 Introduction

The use of braced barrel vaults as a lightweight space structures is very common and it is worthwhile to investigate their optimal design [1]. Metaheuristic algorithms explore the feasible region of the search space based on randomization and some specified rules through a group of search agents. Nature-inspired phenomena are commonly used as a basis for the rules employed by these agents [2].

In the field of size optimization of single-layer barrel vault frames, some studies are carried out. Kaveh and Eftekhar have presented optimal design of barrel vault frames using IBB–BC algorithm [3], in which a 173-bar single-layer barrel vault is optimized under both symmetrical and unsymmetrical load cases. In a study by the author and colleagues, size optimization of some single-layer barrel vault frames via IMCSS algorithm [4] has been presented.

In a study carried out by Parke [5], several different configurations of braced barrel vaults have been investigated using the stiffness method of analysis. Three different configurations have been analyzed, each with five different span/height ratios and under both cases of symmetrical and nonsymmetrical imposed nodal loads. The reported study which was a comparative investigation demonstrates that the most economical height-to-span ratio from weight point of view is approximately $0 \cdot 17$.

Some studies in the case of size optimization and a comparative study considering shape optimization are carried out for barrel vaults, but a more comprehensive study of the problem of simultaneous shape–size optimization of these structures is still needed. In this chapter, the latter problem is investigated using a new optimization approach. In this approach, a programming interface tool called OAPI is utilized, and an improved version of a recently proposed algorithm called IMCSS algorithm is used as the optimization tool.

Charged system search (CSS) is a relatively new metaheuristic optimization algorithm proposed by Kaveh and Talatahari [6]. This algorithm is based on the

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_7

Coulomb and Gauss laws from physics and the governing laws of motion from the Newtonian mechanics. The modified version of the CSS algorithm has also been proposed by Kaveh et al. [2, 7]. In MCSS algorithm, the magnetic laws are also considered in addition to electrical laws. In the present chapter, the IMCSS algorithm is utilized. In the IMCSS algorithm, the harmony search scheme is used to achieve better results. Some of the most effective parameters in the convergence rate of algorithm are also modified.

This chapter is organized as follows: in Sect. 7.2, the problem of simultaneous shape and size optimization for barrel vault frames is formulated. Section 7.3 presents the optimization approach. In Sect. 7.4, the static loading conditions acting on the structures are defined. Two illustrative numerical examples are presented in Sect. 7.5 to examine the efficiency of the proposed approach, and finally in Sect. 7.6, the concluding remarks are derived.

7.2 Statement of Optimization Problem for Barrel Vault Frames

The purpose of shape optimization of skeletal structures is to find a best state of nodal coordinates in order to minimize the weight of the structure W. On the other hand all of nodal coordinates of barrel vault structures are dependent to the height-to-span ratio. All of nodal coordinates, therefore, can be automatically calculated according to height in a constant span of barrel vault. In this process, the x and y coordinates of the joints will remain constant and the z coordinate of the nodes is calculated as follows:

$$z_i = \sqrt{R^2 - {x_i}^2 - \left(\sqrt{R - h}\right)} \tag{7.1}$$

where x_i is the x coordinate of the ith joint, h is the height of barrel vault, and R is the radius of semicircle which is expressed as

$$R = \frac{S^2 + 4h^2}{8h} \tag{7.2}$$

where S is the span of barrel vault.

The relation between nodal coordinates and height-to-span ratio for this type of space structures is depicted in Fig. 7.1.

The aim of size of optimization of skeletal structures is to minimize the weight of structure W through finding the optimal cross-sectional areas A_i of members. All constraints exerted on both problems of shape and size optimization must be satisfied, simultaneously.

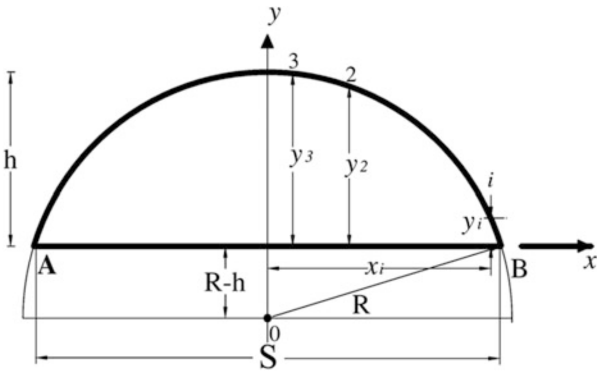


Fig. 7.1 The relation between nodal coordinates and height-to-span ratio (h/S) in the barrel vault

According to the mentioned considerations, the problem of simultaneous shape and size optimization of barrel vault frames can be formulated as follows:

$$
\begin{aligned}
&\text{Find} \quad && X = [x_1, x_2, x_3, \ldots, x_n],\ h \\
& && x_i \in \{d_1,\ d_2,\ \ldots,\ d_{37}\} : \text{Discrete Variables} \\
& && h_{\min} < h < h_{\max} \quad : \text{Continous Variable}
\end{aligned}
\tag{7.3}
$$

$$\text{to minimize} \quad \text{Mer}(X) = f_{\text{penalty}}(X) \times W(X)$$

Subjected to the following constraints

Displacement constraint:

$$v_i^{\mathrm{d}} = \left|\frac{\delta_i}{\bar{\delta}_i}\right| - 1 \le 0, \quad i = 1, 2, \ldots, nn \tag{7.4}$$

Shear constraint, for both major and minor axis (AISC-LRFD, Chapter G) [8]:

$$v_i^{\mathrm{s}} = \frac{V_{\mathrm{u}}}{\varphi_v V_{\mathrm{n}}} - 1 \le 0, \quad i = 1, 2, \ldots, nm \tag{7.5}$$

Constraints corresponding to interaction of bending moment and axial force (AISC-LRFD, Chapter H) [8]:

$$v_i^{\mathrm{I}} = \begin{cases} \dfrac{P_{\mathrm{u}}}{2\varphi_c P_{\mathrm{n}}} + \left(\dfrac{M_{\mathrm{u}x}}{\varphi_b M_{\mathrm{n}x}} + \dfrac{M_{\mathrm{u}y}}{\varphi_b M_{\mathrm{n}y}}\right) - 1 \le 0 \quad \text{for } \dfrac{P_{\mathrm{u}}}{\varphi_c P_{\mathrm{n}}} < 0.2 \\ \dfrac{P_{\mathrm{u}}}{\varphi_c P_{\mathrm{n}}} + \dfrac{8}{9}\left(\dfrac{M_{\mathrm{u}x}}{\varphi_b M_{\mathrm{n}x}} + \dfrac{M_{\mathrm{u}y}}{\varphi_b M_{\mathrm{n}y}}\right) - 1 \le 0 \quad \text{for } \dfrac{P_{\mathrm{u}}}{\varphi_c P_{\mathrm{n}}} \ge 0.2 \end{cases}, \quad i = 1, 2, \ldots, nm \tag{7.6}$$

where X is a vector which contains the design variables; for the discrete optimum design problem, the variables x_i are selected from an allowable set of discrete values; n is the number of member groups; h is the height of barrel vault which is known as the only shape variable; d_j is the jth allowable discrete value for the size design variables; $h_{\min}$ and $h_{\max}$ are the permitted minimum and maximum values of the height which are, respectively, taken as $S/20$ and $S/2$ in this chapter; S is the span of barrel vault; Mer(X) is the merit function; $W(X)$ is the cost function, which is taken as the weight of the structure; $f_{\text{penalty}}(X)$ is the penalty function which results from the violations of the constraints corresponding to the response of the structure; nn is the number of nodes; δ_i and $\overline{\delta}_i$ are the displacement of the joints and the allowable displacement, respectively; nm is the number of members; V_{u} is the required shear strength; V_{n} is the nominal shear strength which is defined by the equations in Chap. G of the LRFD specification [8]; φ_v is the shear resistance factor $\varphi_v = 0.9$; P_{u} is the required strength (tension or compression); P_{n} is the nominal axial strength (tension or compression); φ_c is the resistance factor ($\varphi_c = 0.9$ for tension, $\varphi_c = 0.85$ for compression); M_{u} is the required flexural strength, i.e., the moment due to the total factored load (subscript x or y denotes the axis about which bending occurs); M_{n} is the nominal flexural strength determined in accordance with the appropriate equations in Chap. F of the LRFD specification [8]; and φ_b is the flexural resistance reduction factor ($\varphi_b = 0.9$).

For the displacement limitations which must be considered to ensure the serviceability requirements, the BS 5950 [9] limits the vertical deflections δ_v due to unfactored loads to span/360, i.e., $\delta_V = S/360$ and horizontal displacements δ_H to height/300, i.e., $\delta_H = h/300$ [10].

The nominal axial strength P_{n} is defined as

$$P_{\text{n}} = A_{\text{g}} F_{\text{cr}} \tag{7.7}$$

where A_{g} is the gross area of member and F_{cr} is obtained as follows:

$$F_{\text{cr}} = \begin{cases} \left(\dfrac{0.658}{\lambda_c^2}\right).F_{\text{y}} & \text{for } \lambda_c \le 1.5 \\ \left(\dfrac{0.877}{\lambda_c^2}\right).F_{\text{y}} & \text{for } \lambda_c > 1.5 \end{cases} \tag{7.8}$$

where F_{y} is the specified minimum yield stress and the boundary between inelastic and elastic instability is $\lambda_c = 1.5$, where:

$$\lambda_c = \frac{KL}{r\pi}\sqrt{\frac{F_{\text{y}}}{E}} \tag{7.9}$$

where K is the effective length factor for the member ($K = 1.0$ for braced frames [8]), L is the unbraced length of member, r is the governing radius of gyration about plane of buckling, and E is the modulus of elasticity for the member of structure.

The cost function can be expressed as

$$W(X) = \sum_{i=1}^{nm} \gamma_i \cdot x_i \cdot L_i \tag{7.10}$$

where γ_i is the material density of member i; L_i is the length of member i; and x_i is the cross-sectional area of member i as the design variable.

The penalty function can be defined as

$$f_{\text{penalty}}(X) = \left(1 + \varepsilon_1 \cdot \sum_{j=1}^{np} \upsilon_{(j)}^{k}\right)^{\varepsilon_2}, \tag{7.11}$$

where np is the number of multiple loading conditions. In this chapter ε_1 is taken as unity and ε_2 is set to 1.5 in the first iterations of the search process, but gradually it is increased to 3 [11]. υ^k is the summation of penalties for all imposed constraints for kth charged particle which is mathematically expressed as

$$\upsilon^k = \sum_{i=1}^{nn} \max\left(\upsilon_i^{\mathrm{d}}, 0\right) + \sum_{i=1}^{nm} \left(\max\left(\upsilon_i^{\mathrm{I}}, 0\right) + \max\left(\upsilon_i^{\mathrm{s}}, 0\right)\right) \tag{7.12}$$

where υ_i^{d}, υ_i^{I}, υ_i^{s} are the summation of displacement, shear, and interaction formula penalties which are calculated by Eqs. (7.4) through (7.6), respectively.

7.3 The Optimization Approach

An approach which contains improved magnetic charged system search (IMCSS) and open application programming interface (OAPI) is presented for the problem of simultaneous shape and size optimization of barrel vaults. The IMCSS is used as the optimization algorithm, and the OAPI is utilized as an interface tool between analysis software and the programming language. In IMCSS algorithm, magnetic charged system search (MCSS) and an improved scheme of harmony search (IHS) are utilized, and two of the most effective parameters in the convergence rate of HS scheme are improved to achieve a good convergence rate and good solutions especially in final iterations [12].

The IMCSS algorithm and the OAPI tool are expressed in the following:

7.3.1 Improved Magnetic Charged System Search

Recently, the CSS algorithm and its modified version MCSS algorithm are, respectively, presented by Kaveh and Talathari [6] and Kaveh et al. [7] for optimization problems. The CSS algorithm takes its inspiration from the physical laws governing a group of charged particles (CPs). These charged particles are sources of the electric fields, and each CP can exert electric force on other CPs. The movement of each CP due to the electric force can be determined using the Newtonian mechanic laws. The MCSS algorithm considers the magnetic force in addition to electric force for movement of CPs.

In this chapter, an improved version of MCSS algorithm called IMCSS is presented. The IMCSS algorithm can be summarized as follows:

Level 1: Initialization

Step 1: Initialization. Initialize the algorithm parameters; the initial positions of CPs are determined randomly in the search space

$$x_{i,j}^{(0)} = x_{i,\min} + rand \cdot (x_{i.\max} - x_{i,\min}), \qquad i = 1, 2, \ldots, n. \tag{7.13}$$

where $x_{i,j}^{(0)}$ determines the initial value of the ith variable for the jth CP; $x_{i,\min}$ and $x_{i,\max}$ are the minimum and the maximum allowable values for the ith variable; *rand* is a random number in the interval [0,1]; and n is the number of variables. The initial velocities of charged particles are zero

$$v_{i,j}^{(0)} = 0, \;\; i = 1, 2, \ldots, n. \tag{7.14}$$

The magnitude of the charge is calculated as follows:

$$q_i = \frac{\text{fit}(i) - \text{fitworst}}{\text{fitbest} - \text{fitworst}}, \;\; i = 1, 2, \ldots, N. \tag{7.15}$$

where fitbest and fitworst are the best and the worst fitness of all particles; fit(i) represents the fitness of the agent i; and N is the total number of CPs. The separation distance r_{ij} between two charged particles is defined as

$$r_{ij} = \frac{\left\| X_i - X_j \right\|}{\left\| \left(X_i + X_j \right)/2 - X_{\text{best}} \right\| + \varepsilon}, \tag{7.16}$$

where X_i and X_j are the positions of the ith and jth CPs, X_{best} is the position of the best current CP, and ε is a small positive number to avoid singularities.

Step 2. CP ranking. Evaluate the values of Merit function for the CPs, compare with each other and sort them in an increasing order based on the corresponding value of merit function.

Step 3. Creation of charged memory (CM). Store CMS number of the first CPs in the CM.

Level 2: Search

Step 1: Force calculation. The probability of the attraction of the *i*th CP by the *j*th CP is expressed as

$$p_{ij} = \begin{cases} 1 & \dfrac{\text{fit}(i) - \text{fitbest}}{\text{fit}(j) - \text{fit}(i)} > rand \text{ or } \text{fit}(j) > \text{fit}(i), \\ 0 & \text{else.} \end{cases} \tag{7.17}$$

where *rand* is a random number which is uniformly distributed in the range of (0,1). The resultant electrical force $F_{E,j}$ acting on the *j*th CP can be calculated as follows:

$$F_{E,j} = q_j \cdot \sum_{i, i \neq j} \left(\frac{q_i}{a^3} r_{ij} \cdot i_1 + \frac{q_i}{r_{ij}{}^2} i_2 \right) \cdot p_{ij} \left(X_i - X_j \right), \quad \begin{cases} i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a, \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a, \\ j = 1, 2, \ldots, N. \end{cases} \tag{7.18}$$

The probability of the magnetic influence (attracting or repelling) of the *i*th wire (CP) on the *j*th CP is expressed as

$$pm_{ij} = \begin{cases} 1 & \text{fit}(j) > \text{fit}(i), \\ 0 & \text{else.} \end{cases} \tag{7.19}$$

where fit(i) and fit(j) are the objective values of the *i*th and *j*th CPs, respectively. Such a definition ensures that only a good CP can affect a bad CP by the magnetic force.

The resultant magnetic force $F_{B,j}$ acting on the *j*th CP due to the magnetic field of the *i*th virtual wire (*i*th CP) can be expressed as

$$F_{B,j} = q_j \cdot \sum_{i, i \neq j} \left(\frac{I_i}{R^2} r_{ij} \cdot z_1 + \frac{I_i}{r_{ij}} \cdot z_2 \right) \cdot pm_{ij} \left(X_i - X_j \right), \quad \begin{cases} z_1 = 1, z_2 = 0 \Leftrightarrow r_{ij} < R, \\ z_1 = 0, z_2 = 1 \Leftrightarrow r_{ij} \geq R, \\ j = 1, 2, \ldots, N. \end{cases} \tag{7.20}$$

where q_i is the charge of the ith CP, R is the radius of the virtual wires, I_i is the average electric current in each wire, and pm_{ij} is the probability of the magnetic influence (attracting or repelling) of the ith wire (CP) on the jth CP.

The average electric current in each wire I_i can be expressed as

$$\left(I_{\mathrm{avg}}\right)_{ik} = \mathrm{sign}\left(df_{i,k}\right) \times \frac{\left|df_{i,k}\right| - df_{\min,k}}{df_{\max,k} - df_{\min,k}}, \tag{7.21}$$

$$df_{i,k} = \mathrm{fit}_k(i) - \mathrm{fit}_{k-1}(i), \tag{7.22}$$

where $df_{i,k}$ is the variation of the objective function of the ith CP in the kth movement (iteration). Here, $\mathrm{fit}_k(i)$ and $\mathrm{fit}_{k-1}(i)$ are the values of the objective function of the ith CP at the start of the kth and $k-1$th iterations, respectively. Considering absolute values of $df_{i,k}$ for all of the current CPs, $df_{\max,k}$ and $df_{\min,k}$ would be the maximum and minimum values among these absolute values of df, respectively.

A modification can be considered to avoid trapping in part of search space (Local optima) because of attractive electrical force in CSS algorithm [7]:

$$F = p_r \times F_E + F_B, \tag{7.23}$$

where p_r is the probability that an electrical force is a repelling force which is defined as

$$p_r = \begin{cases} 1 & rand > 0.1 \cdot (1 - \mathrm{iter}/\mathrm{iter}_{\max}), \\ -1 & \text{else.} \end{cases} \tag{7.24}$$

where $rand$ is a random number uniformly distributed in the range of (0,1), iter is the current number of iterations, and $\mathrm{iter}_{\max}$ is the maximum number of iterations.

Step 2: Obtaining new solutions. Move each CP to the new position and calculate the new velocity as follows:

$$X_{j,\mathrm{new}} = rand_{j1} \cdot k_{\mathrm{a}} \cdot \frac{F_j}{m_j} \cdot \Delta t^2 + rand_{j2} \cdot k_{\mathrm{v}} \cdot V_{j,\mathrm{old}} \cdot \Delta t + X_{j,\mathrm{old}}, \tag{7.25}$$

$$V_{j,\mathrm{new}} = \frac{X_{j,\mathrm{new}} - X_{j,\mathrm{old}}}{\Delta t}, \tag{7.26}$$

where $rand_{j1}$ and $rand_{j2}$ are two random numbers uniformly distributed in the range of (0,1). Here, m_j is the mass of the jth CP which is equal to q_j. Δt is the time step and is set to unity. k_{a} is the acceleration coefficient; k_{v} is the velocity coefficient to control the influence of the previous velocity. k_{a} and k_{v} are considered as

$$k_{\mathrm{a}} = c_1 \cdot \left(1 + \mathrm{iter}\Big/\mathrm{iter_{max}}\right),\ k_{\mathrm{v}} = c_2 \cdot \left(1 - \mathrm{iter}\Big/\mathrm{iter_{max}}\right), \tag{7.27}$$

where c_1 and c_2 are two constants to control the exploitation and exploration of the algorithm, respectively.

Step 3. Position correction of CPs. If each CP violates the boundary, its position is corrected using an improved harmony search-based approach which is expressed as follows:

In the process of position correction of CPs using harmony search-based approach, the CMCR and PAR parameters help the algorithm to find globally and locally improved solutions, respectively. PAR and *bw* in HS scheme are very important parameters in fine-tuning of optimized solution vectors and can be potentially useful in adjusting convergence rate of algorithm to reach better solutions [13]. The standard version of CSS and MCSS algorithms use the traditional HS scheme with constant values for both PAR and *bw*. Small PAR values with large *bw* values can lead to poor performance of the algorithm and considerable increase in iterations needed to find optimum solution. Although small *bw* values in final iterations increase the fine-tuning of solution vectors, in the first iterations *bw* must take a bigger value to enforce the algorithm to increase the diversity of solution vectors. Furthermore, large PAR values with small *bw* values usually lead to improvement of the best solutions in final iterations and a better convergence to optimal solution vector. To improve the performance of the HS scheme and eliminate the drawbacks which lie with constant values of PAR and *bw*, the IMCSS algorithms use improved HS scheme with the variable values of PAR and *bw* in position correction step. PAR and *bw* change dynamically with iteration number as shown in Fig. 7.2 and are expressed as follows [13]:

$$\mathrm{PAR}(\mathrm{iter}) = \mathrm{PAR_{min}} + \frac{(\mathrm{PAR_{max}} - \mathrm{PAR_{min}})}{\mathrm{iter_{max}}} \cdot \mathrm{iter} \tag{7.28}$$

and

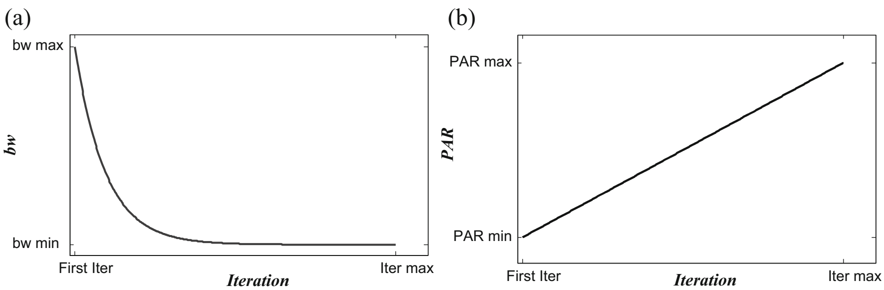


Fig. 7.2 Variation of (**a**) *bw* and (**b**) PAR versus iteration number in IMCSS algorithm [4]

$$bw(\text{iter}) = bw_{\max}\exp(c \cdot \text{iter}), \tag{7.29}$$

$$c = \frac{Ln\left({}^{bw_{\min}}\!/_{bw_{\max}}\right)}{\text{iter}_{\max}}, \tag{7.30}$$

where PAR(iter) and bw(iter) are the values of PAR and bandwidth for current iteration, respectively. $bw_{\min}$ and $bw_{\max}$ are the minimum and maximum bandwidth, respectively.

Step 4: CP ranking. Evaluate and compare the values of merit function for the new CPs, and sort them in an increasing order.

Step 5: CM updating. If some new CP vectors are better than the worst ones in the CM (in terms of corresponding merit function), include the better vectors in the CM and exclude the worst ones from the CM.

Level 3: Controlling the Terminating Criterion
Repeat the search level steps until a terminating criterion is satisfied. The terminating criterion is considered to be the number of iterations.

7.3.2 Discrete IMCSS Algorithm

The present algorithms can be also applied to optimal design problems with discrete variables. One way to solve discrete problems using a continuous algorithm is to utilize a rounding function which changes the magnitude of a result to the nearest discrete value as follows:

$$X_{j,\text{new}} = \text{Fix}\left(rand_{j1} \cdot k_{\text{a}} \cdot \frac{F_j}{m_j} \cdot \Delta t^2 + rand_{j2} \cdot k_{\text{v}} \cdot V_{j,\text{old}} \cdot \Delta t + X_{j,\text{old}}\right), \tag{7.31}$$

where Fix(X) is a function which rounds each element of vector X to the nearest permissible discrete value. Using this position updating formula, the agents will be permitted to select discrete values [13].

7.3.3 Open Application Programming Interface

Recently, Computers and Structures Inc. have introduced a powerful interface tool known as Open Application Programming Interface (OAPI). The OAPI can be utilized to automate and manage many of the processes required to build, analyze, and design models through a programming language [14].

The computer program SAP2000 is a software of proven ability in analysis and design of practical large-scale structures. The utilization of this software, therefore, could be useful for the problem of structural optimization. In this process, the OAPI can be utilized in order to connect SAP2000 with the programming language which provides a path for two-way exchange of SAP model information with the programming language. There are many programming languages that can be used to access SAP2000 through the OAPI such as MATLAB, Visual Basic, Visual C#, Intel Visual Fortran, Microsoft Visual C++, and Python.

In some studies carried out by the author and colleagues, size optimization of single-layer barrel vault frames [4] and double-layer barrel vaults [12] is already investigated using this interface tool and MATLAB. Furthermore, Kaveh et al. [13] have utilized this interfacing ability in the form of parallel computing within the MATLAB for practical optimum design of real-size 3D steel frames.

In this chapter, MATLAB is utilized in order to perform the process of optimization via presented approach (OAPI and IMCSS).

7.4 Static Loading Conditions

According to ANSI-A58.1 [15] and ASCE/SEI 7-10 [16] codes, there are some specific considerations for loading conditions of arched roofs such as barrel vault structures. In this chapter, three static loading conditions are considered for optimization of these structures which are expressed as follows:

7.4.1 Dead Load (DL)

A uniform dead load of 100 kg/m^2 is considered for estimated weight of sheeting, space frame, and nodes of barrel vault structure.

7.4.2 Snow Load (SL)

The snow load for arched roofs is calculated according to ANSI [15] and ASCE [16] codes. Snow loads acting on a sloping surface shall be assumed to act on the horizontal projection of that surface. The sloped roof (balanced) snow load, P_s, shall be obtained by multiplying the flat roof snow load, P_f, by the roof slope factor, C_s, as follows:

$$P_s = C_s.P_f \tag{7.32}$$

where C_s is

$$C_s = \begin{cases} 1.0 & \alpha \le 15^\circ \\ 1.0 - \dfrac{\alpha - 15}{60} & 15^\circ < \alpha < 60^\circ \\ 0.25 & \alpha > 60^\circ \end{cases} \tag{7.33}$$

The C_s distribution in arched roofs is shown in Fig. 7.3. In this chapter, the flat roof snow load P_f is set to 150 kg/m^2.

7.4.3 Wind Load (WL)

For wind load in arched roofs, different loads are applied in the windward quarter, center half, and leeward quarter of the roof which are computed based on ANSI [15] and ASCE [16] codes as

$$P = q G_h C_p \tag{7.34}$$

where q is the wind velocity pressure, G_h is gust-effect factor, and C_p is the external pressure coefficient. These parameters are calculated according to ANSI [15] and ASCE [16] codes.

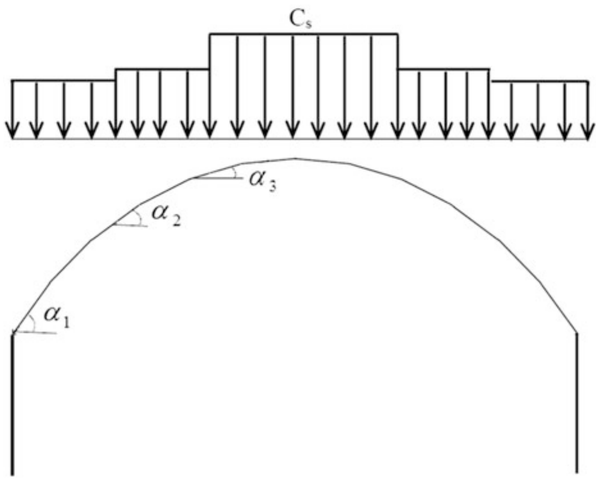


Fig. 7.3 C_s distribution in arched roofs

7.5 Numerical Examples

This study presents optimal shape and size design of two single-layer barrel vault frames which are first provided for size optimization by Kaveh et al. [4]. For all of examples, a population of 100 charged particles is used, and the value of CMCR is set to 0.95. The values of PAR_{min} and PAR_{max} in IMCSS algorithm are set to 0.35 and 0.9, respectively.

The two examples are discrete optimum design problems, and the variables are selected from an allowable set of steel pipe sections taken from AISC-LRFD code [17] shown in Table 7.1. For analysis of these structures, SAP2000 is used through OAPI tool, and the optimization process is performed in MATLAB.

In all examples, the material density is 0.2836 lb/in^3 (7850 kg/m^3) and the modulus of elasticity is 30,450 ksi (2.1 × 10^6 kg/cm^2). The yield stress F_y of steel is taken as 34,135.96 psi (2400 kg/cm^2) for both problems.

7.5.1 A 173-Bar Single-Layer Barrel Vault Frame

The 173-bar single-layer barrel vault frame with a 2-way grid pattern is shown in Fig. 7.4. This spatial structure consists of 108 joints and 173 members. There are 16 design variables in this problem which consist of size and shape variables. For the process of size optimization, all members of this structure are categorized into 15 groups, as shown in Fig. 7.4b. Furthermore, for the problem of shape optimization, the lower and upper bounds of height as the only shape variable are 1.5 m and 15 m, respectively. The nodal displacements are limited to ±1.05 in (26 mm) in x, y directions and ±1.64 in (41 mm) in z direction.

The configuration of the 173-bar single-layer barrel vault is as follows:

- Span (S) = 30 m (1181.1 in)
- Height (H) = 8 m (314.96 in)
- Length (L) = 30 m (1181.1 in)

According to ANSI/ASCE considerations mentioned in Sect. 7.4, this spatial structure is subjected to three loading conditions:

A uniform dead load of 100 kg/m^2 is applied on the roof. The applied snow and wind loads on this structure are shown in Fig. 7.5a and b, respectively.

The convergence history for optimization of this structure using CSS, MCSS, and IMCSS algorithms is shown in Fig. 7.6. Comparison of the optimal design results using presented algorithms is also provided in Table 7.2.

As seen in Table 7.2, the IMCSS algorithm finds its best solutions in 89 iterations (8900 analyses), but the CSS and MCSS algorithms have not found any better solutions in 10,000 analyses. The best weight of IMCSS is 39,778.21 lb (18,043.09 kg), while it is 41,589.25 lb and 42,957.98 lb for the MCSS and CSS

Table 7.1 The allowable steel pipe sections taken from AISC-LRFD code [17]

			Dimensions		Properties				
		Section name	Nominal diameter (in.)	Weight per ft (lb)	Area (in.^{2})	Moment of inertia (in.^{4})	Elastic section modulus (in.^{3})	Gyration radius (in)	Plastic section modulus (in.^{3})
1	Standard weight	P0.5	½	0.85	0.25	0.017	0.041	0.261	0.059
2		P0.75	¾	1.13	0.333	0.037	0.071	0.334	0.1
3		P1	1	1.68	0.494	0.087	0.133	0.421	0.187
4		P1.25	1¼	2.27	0.669	0.195	0.235	0.54	0.324
5		P1.5	1½	2.72	0.799	0.31	0.326	0.623	0.448
6		P10	2	3.65	1.07	0.666	0.561	0.787	0.761
7		P12	2½	5.79	1.7	1.53	1.06	0.947	1.45
8		P2	3	7.58	2.23	3.02	1.72	1.16	2.33
9		P2.5	3½	9.11	2.68	4.79	2.39	1.34	3.22
10		P3	4	10.79	3.17	7.23	3.21	1.51	4.31
11		P3.5	5	14.62	4.3	15.2	5.45	1.88	7.27
12		P4	6	18.97	5.58	28.1	8.5	2.25	11.2
13		P5	8	28.55	8.4	72.5	16.8	2.94	22.2
14		P6	10	40.48	11.9	161	29.9	3.67	39.4
15		P8	12	49.56	14.6	279	43.8	4.38	57.4
16	Extra strong	XP0.5	½	1.09	0.32	0.02	0.048	0.25	0.072
17		XP0.75	¾	1.47	0.433	0.045	0.085	0.321	0.125
18		XP1	1	2.17	0.639	0.106	0.161	0.407	0.233
19		XP1.25	1¼	3	0.881	0.242	0.291	0.524	0.414
20		XP1.5	1½	3.63	1.07	0.391	0.412	0.605	0.581
21		XP10	2	5.02	1.48	0.868	0.731	0.766	1.02
22		XP12	2½	7.66	2.25	1.92	1.34	0.924	1.87
23		XP2	3	10.25	3.02	3.89	2.23	1.14	3.08
24		XP2.5	3½	12.5	3.68	6.28	3.14	1.31	4.32

25		XP3	4	14.98	4.41	9.61	4.27	1.48	5.85
26		XP3.5	5	20.78	6.11	20.7	7.43	1.84	10.1
27		XP4	6	28.57	8.4	40.5	12.2	2.19	16.6
28		XP5	8	43.39	12.8	106	24.5	2.88	33
29		XP6	10	54.74	16.1	212	39.4	3.63	52.6
30		XP8	12	65.42	19.2	362	56.7	4.33	75.1
31	Double-extra strong	XXP2	2	9.03	2.66	1.31	1.1	0.703	1.67
32		XXP2.5	2½	13.69	4.03	2.87	2	0.844	3.04
33		XXP3	3	18.58	5.47	5.99	3.42	1.05	5.12
34		XXP4	4	27.54	8.1	15.3	6.79	1.37	9.97
35		XXP5	5	38.59	11.3	33.6	12.1	1.72	17.5
36		XXP6	6	53.16	15.6	66.3	20	2.06	28.9
37		XXP8	8	72.42	21.3	162	37.6	2.76	52.8

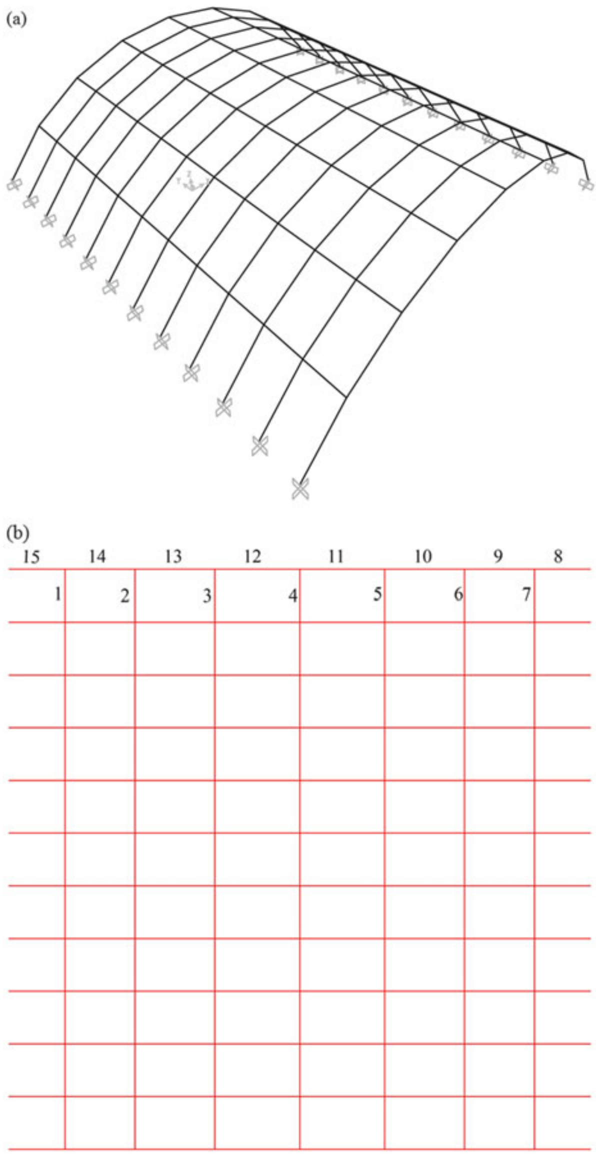


Fig. 7.4 The 173-bar single-layer barrel vault frame, (**a**) three-dimensional view, (**b**) member groups in top view [4]

algorithms, respectively. As it can be seen in the results, the IMCSS algorithm obtains a better weight in a lower number of analyses than previous algorithms.

Furthermore, the values of 131.03 in, 131.62 in, and 113.9 in are obtained for the height of barrel vault for the CSS, MCSS, and IMCSS algorithms, respectively. Hence, the best height-to-span ratios obtained from CSS, MCSS, and IMCSS are 0.11, 0.11, and 0.10, respectively. It can be seen that these values are approximately

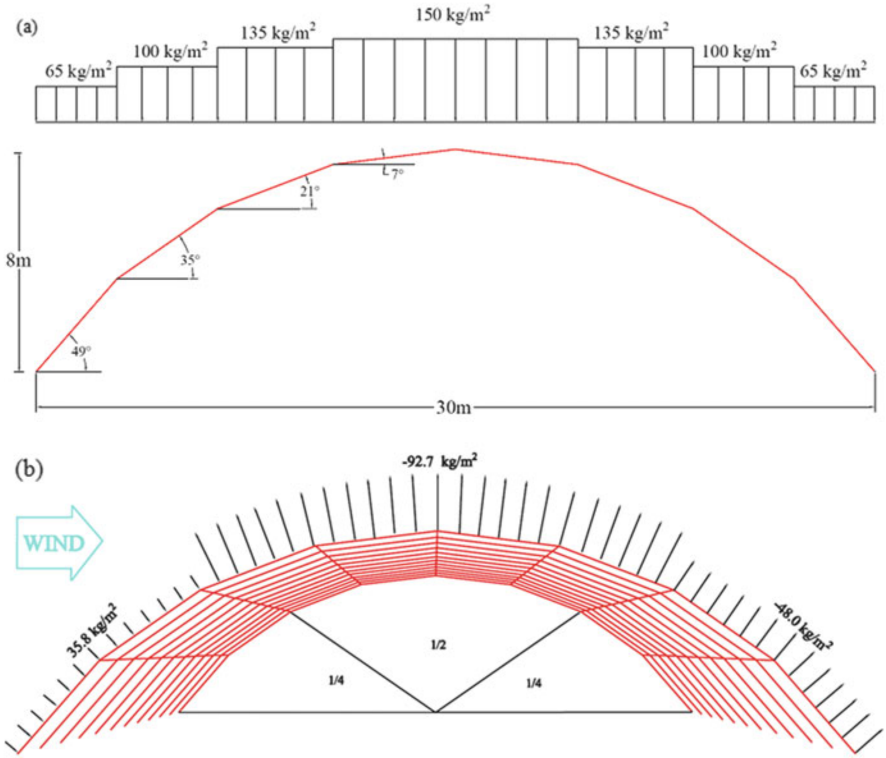


Fig. 7.5 The 173-bar single-layer barrel vault frame subjected to (**a**) snow and (**b**) wind loadings [4]

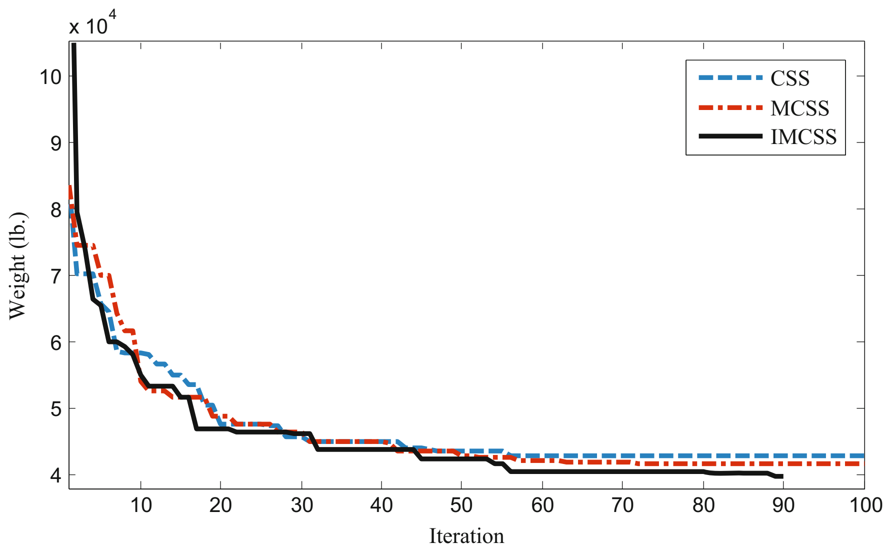


Fig. 7.6 Convergence curves for the 173-bar single-layer barrel vault frame using CSS, MCSS, and IMCSS algorithms

Table 7.2 Optimal solutions for simultaneous shape and size optimization of the 173-bar barrel vault (in^2)

Design variables		CSS		MCSS		IMCSS	
		Section name	Area (in.^{2})	Section name	Area (in.^{2})	Section name	Area (in.^{2})
1	A1	'XP1'	0.639	'XP1'	0.639	P1	0.494
2	A2	'XP1.5'	1.07	'XP1.25'	0.881	P2.5	1.7
3	A3	'XXP2'	2.66	'P2.5'	1.7	XP1.5	1.07
4	A4	'P1.5'	0.799	'XP2'	1.48	P3	2.23
5	A5	'P3.5'	2.68	'XP1.5'	1.07	XP1.5	1.07
6	A6	'XP1.25'	0.881	'P2.5'	1.7	P1.5	0.799
7	A7	'XP2'	1.48	'P1.5'	0.799	P1	0.494
8	A8	'P10'	11.9	'P10'	11.9	P10	11.9
9	A9	'XP6'	8.4	'XP6'	8.4	XP6	8.4
10	A10	'XP6'	8.4	'P10'	11.9	XP6	8.4
11	A11	'P10'	11.9	'XP6'	8.4	P10	11.9
12	A12	'XP6'	8.4	'P10'	11.9	P10	11.9
13	A13	'XP6'	8.4	'P6'	5.58	P6	5.58
14	A14	'P6'	5.58	'P6'	5.58	P6	5.58
15	A15	'P12'	14.6	'P10'	11.9	XP6	8.4
16	Height	131.0308 in (3.33 m)		132.6162 in (3.37 m)		113.9046 in (2.89 m)	
Weight. lb.		42,957.98		41,589.25		39,778.21	
Weight. kg.		19,485.41		18,864.57		18,043.09	
Max. displacement (in)		1.6118		1.4360		1.1277	
Max. strength ratio		0.9865		0.9604		0.9516	
No. of analyses		10,000		10,000		8900	

close to ratio of 0.17 from Parke's study. As seen in Table 7.2, the maximum strength ratio for CSS, MCSS, and IMCSS algorithms is 0.9865, 0.9604, and 0.9516, respectively, and the maximum displacement is 1.6118 in, 1.4360 in, and 1.1277 in for the CSS, MCSS, and IMCSS algorithms, respectively.

Figure 7.7a–c provides strength ratios for all elements of the 173-bar single-layer barrel vault frame for optimal results of CSS, MCSS, and IMCSS algorithms, respectively. The figures show that all strength ratios of elements are lower than 1; thus there is no violation of constraints in the optimal results of presented algorithms, and all strength constraints are satisfied. The maximum strength ratios for element groups of the 173-bar single-layer barrel vault frame are shown in Fig. 7.8a through c for optimal results of the presented algorithms.

Table 7.3 provides a comparison for the results of present work on simultaneous shape and size optimization with those of a previous study [15] on size optimization of the 173-bar barrel vault. Comparison of best weight for both problems is also shown in Table 7.4. As it can be seen in the results, the value of weight of structure has been reduced by 14.59 %, 17.23 %, and 18.8 % via CSS, MCSS, and IMCSS algorithms, respectively.

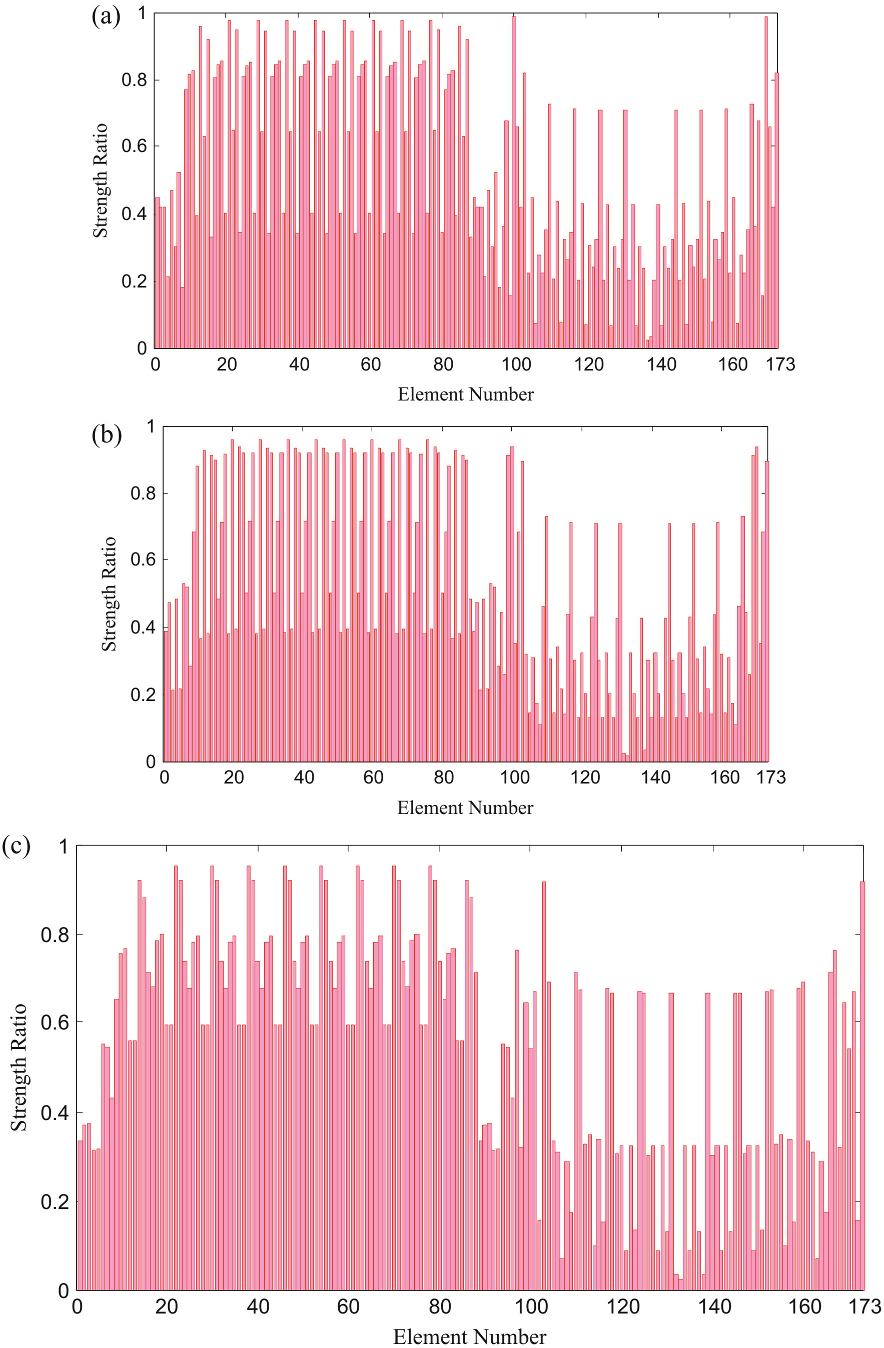


Fig. 7.7 Strength ratios for the elements of the 173-bar single-layer barrel vault frame for optimal results of (**a**) CSS, (**b**) MCSS, and (**c**) IMCSS algorithms

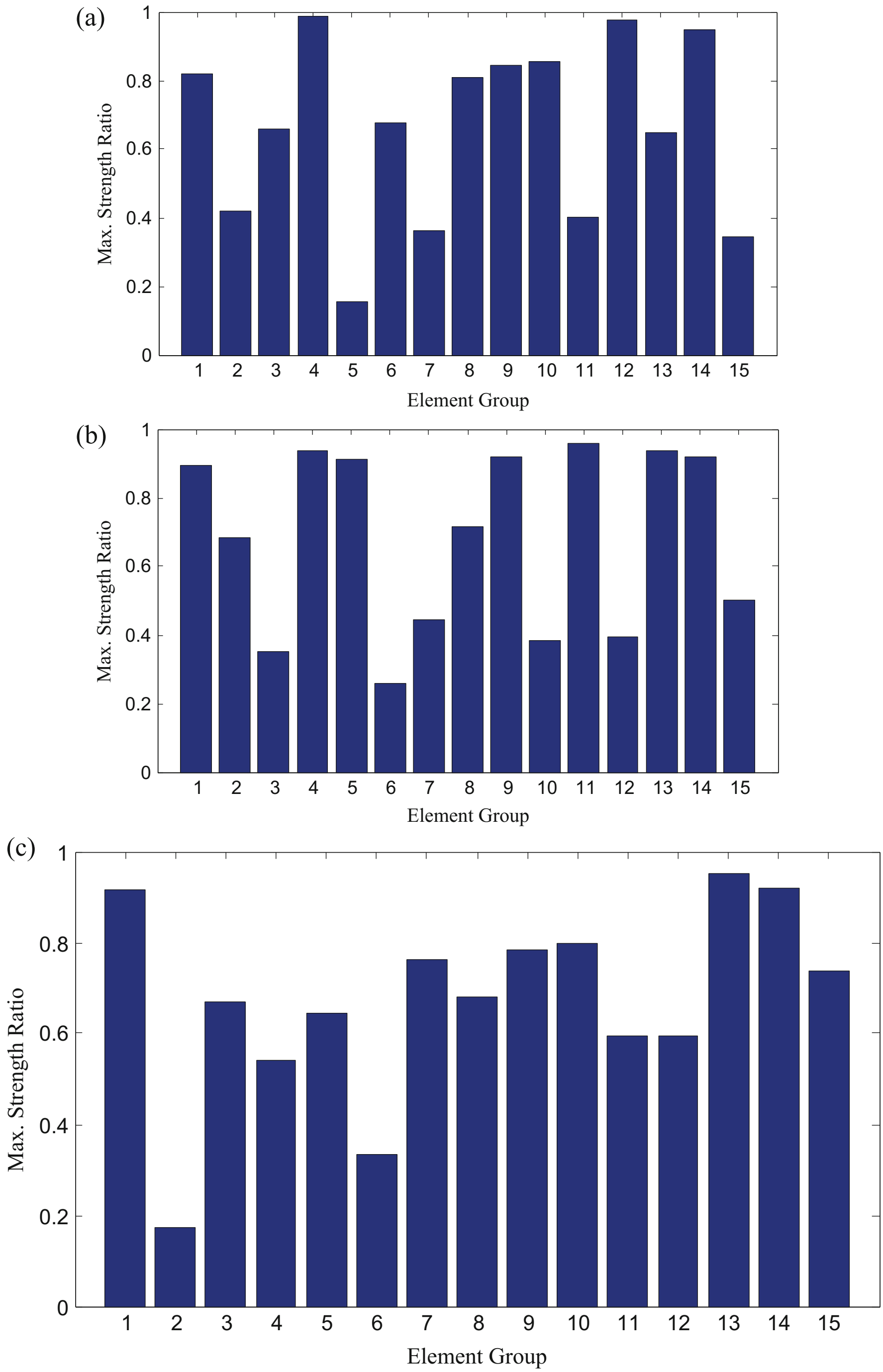


Fig. 7.8 Maximum strength ratios for element groups of the 173-bar single-layer barrel vault frame for optimal results of (**a**) CSS, (**b**) MCSS, and (**c**) IMCSS algorithms

Table 7.3 Comparison of the optimal solutions for the 173-bar single-layer barrel vault frame

Design variables		Kaveh et al. [4]			Present work		
		Size optimization			Simultaneous shape and size optimization		
		CSS	MCSS	IMCSS	CSS	MCSS	IMCSS
1	A1	0.494	0.639	0.25	0.639	0.639	0.494
2	A2	0.494	0.433	0.25	1.07	0.881	1.7
3	A3	1.07	0.494	0.25	2.66	1.7	1.07
4	A4	0.333	0.333	0.25	0.799	1.48	2.23
5	A5	0.32	0.639	0.32	2.68	1.07	1.07
6	A6	0.881	1.07	0.32	0.881	1.7	0.799
7	A7	0.799	0.639	0.25	1.48	0.799	0.494
8	A8	11.9	11.9	14.6	11.9	11.9	11.9
9	A9	11.9	11.9	8.4	8.4	8.4	8.4
10	A10	11.9	11.9	11.9	8.4	11.9	8.4
11	A11	11.9	11.9	11.9	11.9	8.4	11.9
12	A12	11.9	11.9	11.9	8.4	11.9	11.9
13	A13	5.58	5.58	5.58	8.4	5.58	5.58
14	A14	5.58	5.58	5.58	5.58	5.58	5.58
15	A15	11.9	11.9	11.9	14.6	11.9	8.4
16	Height (in)	Invariable	Invariable	Invariable	131.03	131.62	113.90
Weight (lb.)		50,295.90	50,247.66	48,985.05	42,957.98	41,589.25	39,778.21
Max. strength ratio		0.8724	0.8689	0.8751	0.9865	0.9604	0.9516
No. of analyses		20,000	20,000	19,800	10,000	10,000	8900

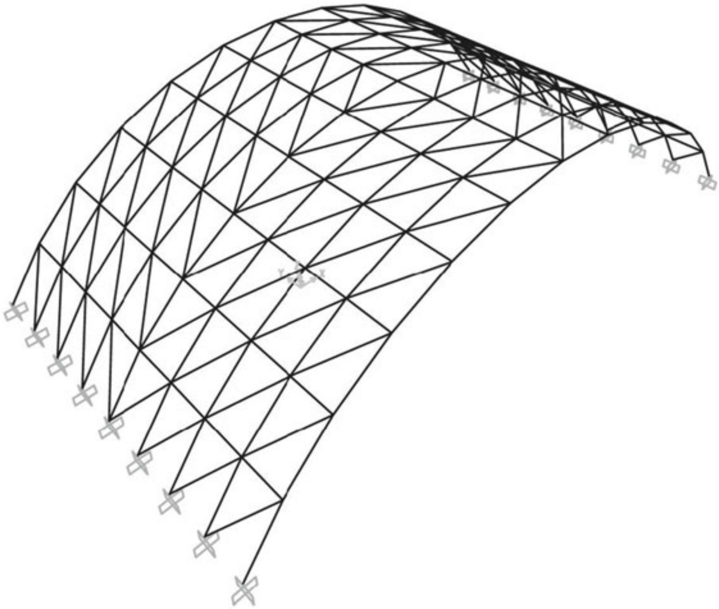
Table 7.4 Comparison of the best weights for the 173-bar single-layer barrel vault frame

optimization problem	Best weight (lb.)		
	CSS	MCSS	IMCSS
Size optimization [4]	50,295.90	50,247.66	48,985.05
Simultaneous shape and size optimization	42,957.98	41,589.25	39,778.21
Percent of reduction in best weights	14.59 %	17.23 %	18.80 %

7.5.2 A 292-Bar Single-Layer Barrel Vault

This spatial structure which is shown in Fig. 7.9 has a three-way pattern [4]. The structure consists of 117 joints and 292 members. The problem has 31 design variables and consists of size and shape variables. In the problem of size optimization, considering the symmetry of the geometry and loading conditions, all members are grouped into 30 independent size variables as shown in Fig. 7.9b. For the problem of shape optimization, the lower and upper bounds of height as the only

(a)



(b)

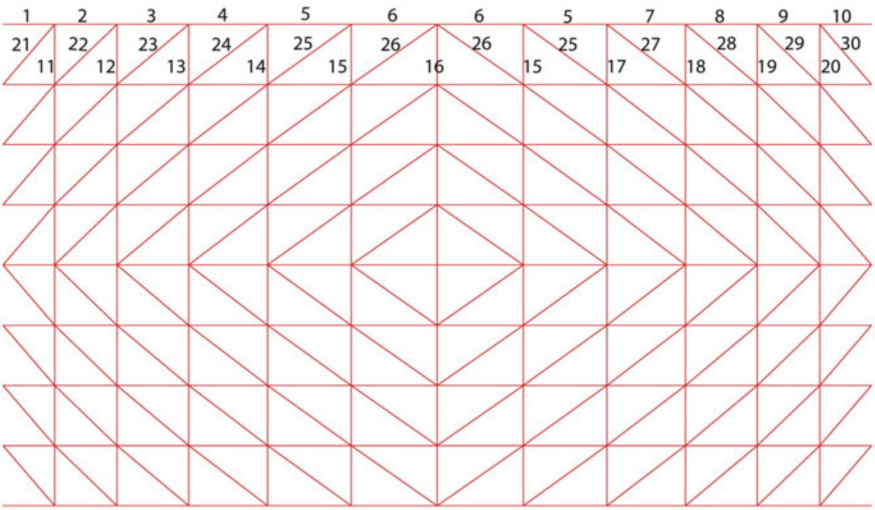


Fig. 7.9 The 292-bar single-layer barrel vault frame: (**a**) three-dimensional view, (**b**) member groups in top view [4]

shape variable are 1.8 m and 18 m, respectively. The nodes are subjected to the displacement limits of ±1.31 in (33 mm) in x, y directions and ±1.97 in (50 mm) in z directions.

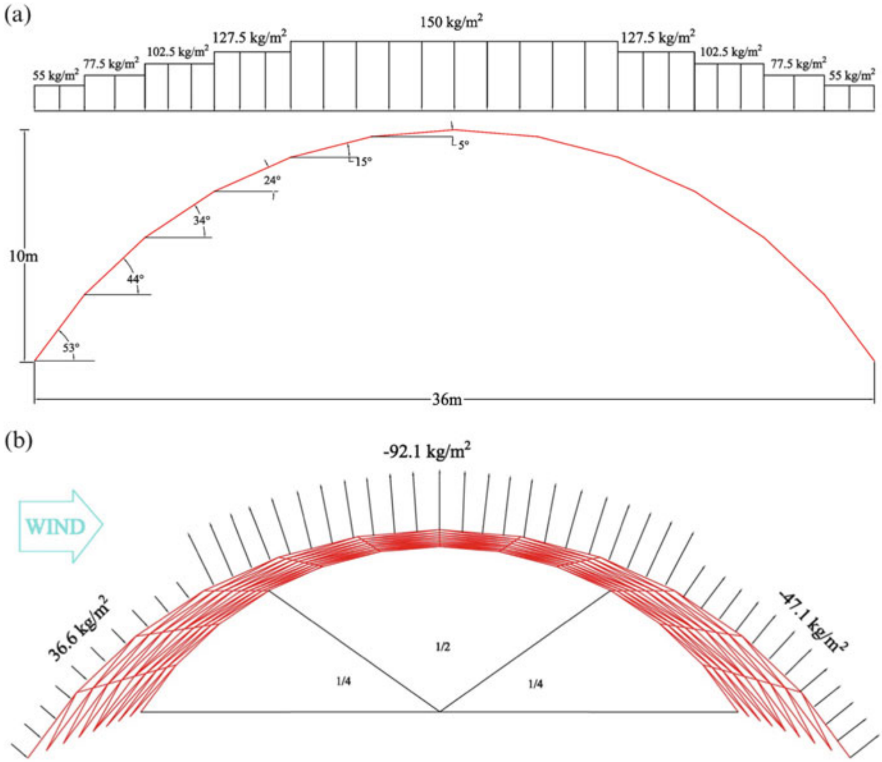


Fig. 7.10 The 292-bar single-layer barrel vault frame subjected to (**a**) snow and (**b**) wind loadings [4]

The configuration of this structure is as follows:

- Span (S) = 36 m (1417.3 in)
- Height (H) = 8 m (393.7 in)
- Length (L) = 20 m (787.4 in)

According to the loading consideration in Sect. 7.4, three loading conditions are applied to this barrel vault as follows:

A uniform dead load of 100 kg/m^2 is applied on the roof. The applied snow load and wind load acting on this barrel vault are shown in Fig. 7.10a and b.

Table 7.5 is provided for comparison of the results of the CSS, MCSS, and IMCSS algorithms for this structure. The convergence history of all algorithms is shown in Fig. 7.11.

As shown in Table 7.5, the best weight of IMCSS algorithm is 51,856.76 lb (23,521.83 kg), while it is 57,119.63 and 52,773.58 lb for the CSS and MCSS algorithms. Although the CSS and MCSS algorithms find their best solutions in 13,200 and 12,500 analyses, the IMCSS algorithm obtains better solutions in 122 iterations (12,200 analyses).

Table 7.5 Optimal solutions for simultaneous shape and size optimization of the 292-bar barrel vault (in^2)

Design variables		CSS		MCSS		IMCSS	
		Section name	Area (in.^{2})	Section Name	Area (in.^{2})	Section name	Area (in.^{2})
1	A1	‘P12’	14.6	‘P10’	11.9	P10	11.9
2	A2	‘XP6’	8.4	‘XP6’	8.4	P10	11.9
3	A3	‘XP10’	16.1	‘XP8’	12.8	XXP5	11.3
4	A4	‘XXP5’	11.3	‘P10’	11.9	XP6	8.4
5	A5	‘XP6’	8.4	‘XP5’	6.11	XP6	8.4
6	A6	‘XP6’	8.4	‘XP6’	8.4	XP6	8.4
7	A7	‘XP6’	8.4	‘P10’	11.9	P10	11.9
8	A8	‘XXP5’	11.3	‘XP6’	8.4	P10	11.9
9	A9	‘XP6’	8.4	‘XXP5’	11.3	P10	11.9
10	A10	‘XP12’	19.2	‘P12’	14.6	P12	14.6
11	A11	‘XP2.5’	2.25	‘P1.25’	0.669	XP3	3.02
12	A12	‘XP3.5’	3.68	‘P2.5’	1.7	P1	0.494
13	A13	‘P2.5’	1.7	‘XXP3’	5.47	XP1.5	1.07
14	A14	‘P2.5’	1.7	‘P1.25’	0.669	P1	0.494
15	A15	‘XP2.5’	2.25	‘XP2.5’	2.25	XP2.5	2.25
16	A16	‘P2.5’	1.7	‘P2.5’	1.7	XP3.5	3.68
17	A17	‘P2.5’	1.7	‘XP5’	6.11	P2.5	1.7
18	A18	‘XP1.25’	0.881	‘P6’	5.58	P1.5	0.799
19	A19	‘XP3.5’	3.68	‘P2.5’	1.7	P2.5	1.7
20	A20	‘P0.75’	0.333	‘XP0.5’	0.32	XP3	3.02
21	A21	‘XP3’	3.02	‘P3’	2.23	XP2	1.48
22	A22	‘P4’	3.17	‘XP4’	4.41	XP1.5	1.07
23	A23	‘P2.5’	1.7	‘P2.5’	1.7	XP1.5	1.07
24	A24	‘P3’	2.23	‘P3’	2.23	XP3	3.02
25	A25	‘P2.5’	1.7	‘XP2’	1.48	P3	2.23
26	A26	‘P3’	2.23	‘XP2’	1.48	P3	2.23
27	A27	‘XP2.5’	2.25	‘XP4’	4.41	XP3.5	3.68
28	A28	‘P2.5’	1.7	‘XP3’	3.02	P2.5	1.7
29	A29	‘XP6’	8.4	‘XP2’	1.48	P1.25	0.669
30	A30	‘XP2.5’	2.25	‘XP2.5’	2.25	XP1.25	0.881
31	Height	204.8791 in (5.20 m)		163.0436 in (4.14 m)		173.0666 in (4.40 m)	
Weight. lb.		57,119.63		52,773.58		51,856.76	
Weight. Kg.		25,909.03		23,937.69		23,521.83	
Max. displacement (in)		1.5802		1.5008		1.4424	
Max. strength ratio		0.9413		0.9303		0.9746	
No. of analyses		13,200		12,500		12,200	

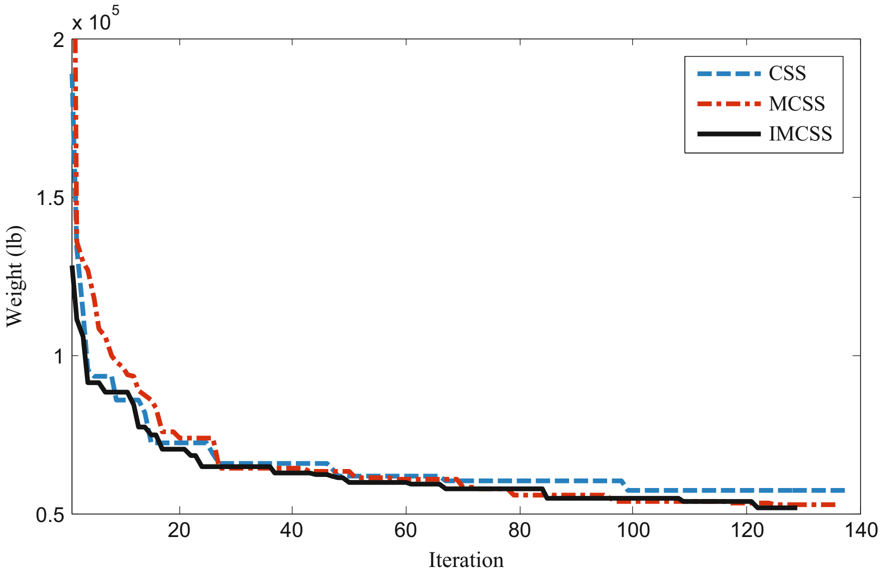


Fig. 7.11 Convergence history for the 292-bar single-layer barrel vault frame using CSS, MCSS, and IMCSS algorithms

The best value for height of this barrel vault from CSS, MCSS, and IMCSS algorithms is 204.88 in, 163.04 in, and 173.07 in, respectively. The best height-to-span ratios, therefore, obtained from CSS, MCSS, and IMCSS algorithms are 0.15, 0.12, and 0.12, respectively, which are approximately close to value of 0.17 from Parke's study.

Table 7.5 also shows the maximum displacement and strength ratios for all algorithms. The values of maximum strength ratio for CSS, MCSS, and IMCSS algorithms are 0.9413, 0.9303, and 0.9746, respectively, and the values of maximum displacement are 1.5802 in. 1.5008 in. and 1.4424 in. respectively. The strength ratios for all elements of the 292-bar single-layer barrel vault are depicted in Fig. 7.12a through c, and the maximum strength ratios for element groups of this structure are presented in Fig. 7.13a through c for optimal results of CSS, MCSS, and IMCSS algorithms, respectively.

As shown in Fig. 7.12a–c, all of the strength ratios of elements are lower than 1; therefore, all of the presented algorithms have no violation of constraints in their best solutions, and the constraints are satisfied.

Table 7.6 draws a comparison between the results of present work on simultaneous shape and size optimization and those of a previous study on size optimization [4] for this structure. On comparison of the best weights for presented algorithms shown in Table 7.7, the value of weight of structure has decreased by 16.4 %, 17.23 %, and 17.65 % via CSS, MCSS, and IMCSS algorithms, respectively.

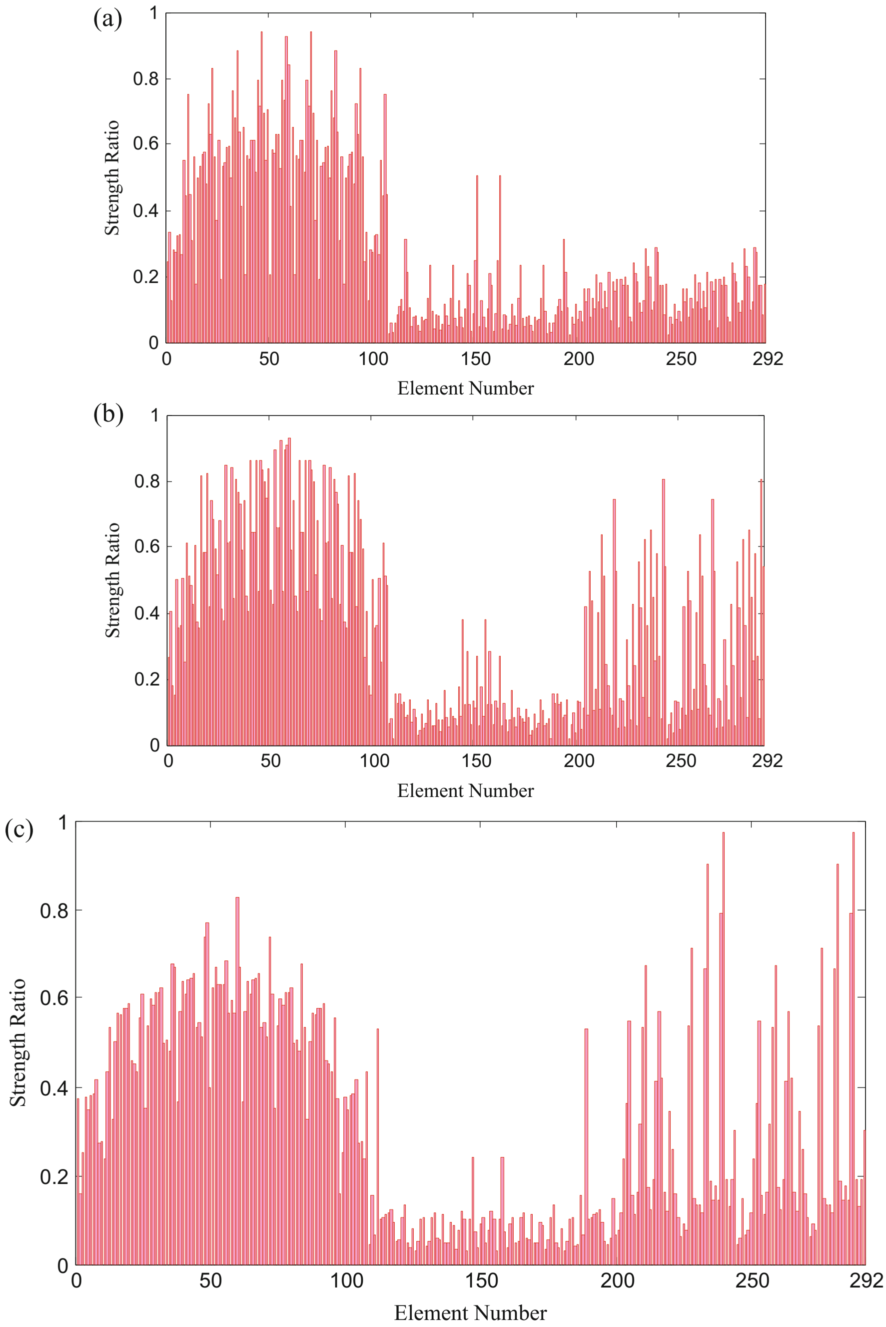


Fig. 7.12 Strength ratios for the elements of the 292-bar single-layer barrel vault frame for optimal results of (**a**) CSS, (**b**) MCSS, and (**c**) IMCSS algorithms

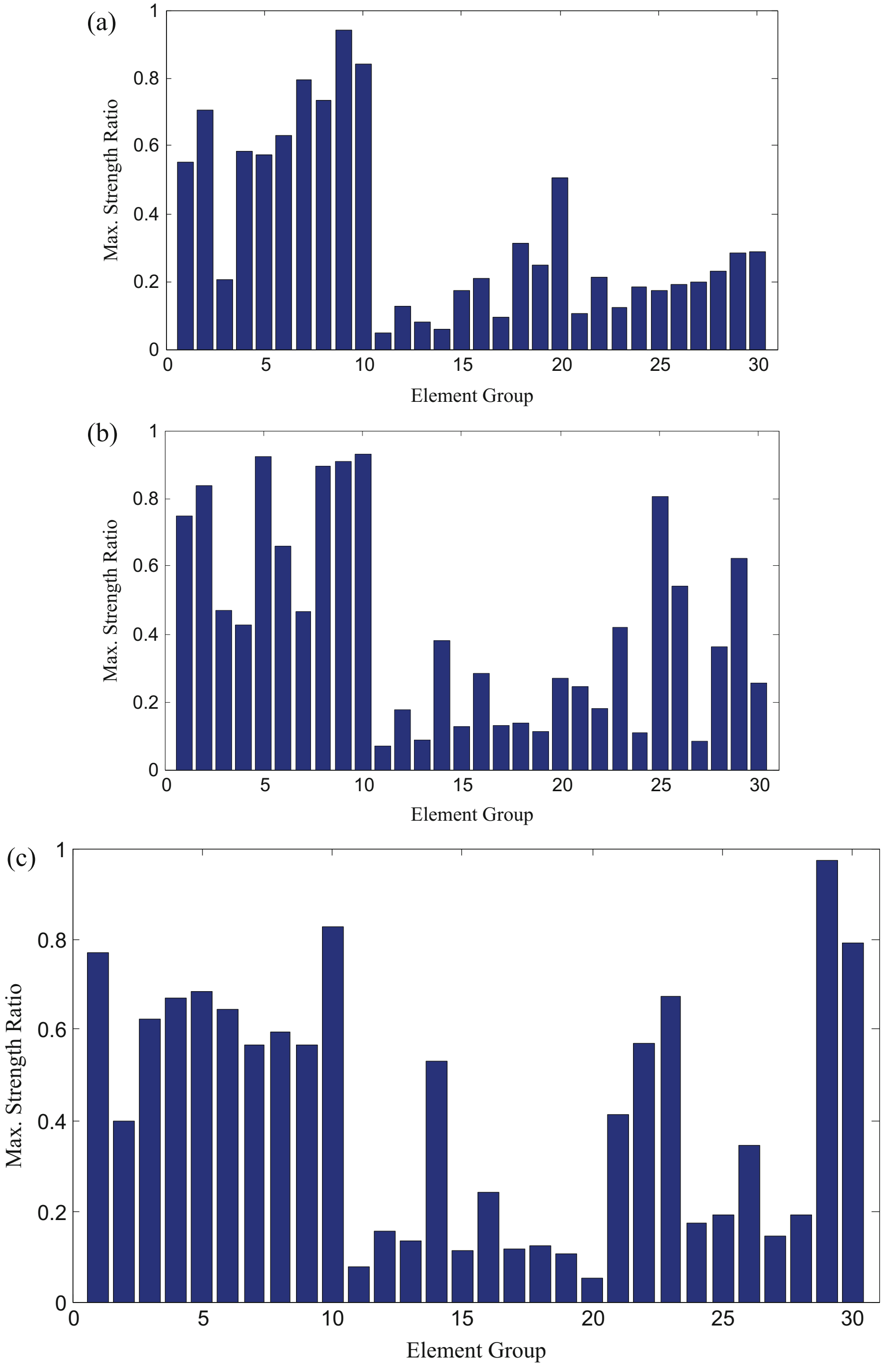


Fig. 7.13 Maximum strength ratios for element groups of the 292-bar single-layer barrel vault frame for optimal results of (**a**) CSS, (**b**) MCSS, and (**c**) IMCSS algorithms

Table 7.6 Comparison of the optimal solutions for the 292-bar single-layer barrel vault frame

Design variables		Kaveh et al. [4]			Present work		
		Size optimization			Simultaneous shape and size optimization		
		CSS	MCSS	IMCSS	CSS	MCSS	IMCSS
1	A1	14.6	14.6	14.6	14.6	11.9	11.9
2	A2	11.9	8.4	8.4	8.4	8.4	11.9
3	A3	12.8	12.8	11.9	16.1	12.8	11.3
4	A4	5.58	14.6	8.4	11.3	11.9	8.4
5	A5	12.8	11.9	11.9	8.4	6.11	8.4
6	A6	11.9	11.9	11.9	8.4	8.4	8.4
7	A7	11.9	14.6	11.9	8.4	11.9	11.9
8	A8	14.6	16.1	14.6	11.3	8.4	11.9
9	A9	11.9	11.9	11.9	8.4	11.3	11.9
10	A10	19.2	19.2	14.6	19.2	14.6	14.6
11	A11	2.25	0.25	1.48	2.25	0.669	3.02
12	A12	0.669	0.433	0.799	3.68	1.7	0.494
13	A13	6.11	1.7	0.669	1.7	5.47	1.07
14	A14	3.68	0.639	0.799	1.7	0.669	0.494
15	A15	1.7	0.669	0.494	2.25	2.25	2.25
16	A16	3.17	1.07	0.799	1.7	1.7	3.68
17	A17	1.48	2.68	2.25	1.7	6.11	1.7
18	A18	1.48	1.07	0.669	0.881	5.58	0.799
19	A19	5.47	0.639	0.639	3.68	1.7	1.7
20	A20	4.3	2.23	1.48	0.333	0.32	3.02
21	A21	2.66	1.48	0.799	3.02	2.23	1.48
22	A22	2.25	1.07	1.07	3.17	4.41	1.07
23	A23	0.639	2.23	0.799	1.7	1.7	1.07
24	A24	1.48	1.7	1.07	2.23	2.23	3.02
25	A25	0.799	0.669	0.669	1.7	1.48	2.23
26	A26	1.07	0.669	0.881	2.23	1.48	2.23
27	A27	0.799	1.7	0.799	2.25	4.41	3.68
28	A28	1.48	2.23	0.799	1.7	3.02	1.7
29	A29	1.07	0.799	1.48	8.4	1.48	0.669
30	A30	2.68	0.799	12.8	2.25	2.25	0.881
31	Height (in)	Invariable	Invariable	Invariable	204.88	163.04	173.07
Weight (lb.)		68,324.57	65,892.33	62,968.19	57,119.63	52,773.58	51,856.76
Max. strength ratio		0.9527	0.8883	0.9939	0.9413	0.9303	0.9746
No. of analyses		20,000	20,000	17,500	13,200	12,500	12,200

Table 7.7 Comparison of the best weights for the 292-bar single-layer barrel vault frame

Optimization problem	Best weight (lb.)		
	CSS	MCSS	IMCSS
Size optimization [4]	68,324.57	65,892.33	62,968.19
Simultaneous shape and size optimization	57,119.63	52,773.58	51,856.76
Percent of reduction in best weights	16.40 %	19.91 %	17.65 %

7.6 Concluding Remarks

This chapter has applied an optimization approach which contains improved magnetic charged system search (IMCSS) and open application programming interface (OAPI) for simultaneous shape and size optimization of barrel vault frames. In this approach, OAPI is utilized as a programming interface tool through programming language to manage the process of structural analysis during the optimization process, and the IMCSS which is an improved version of MCSS algorithm is used for achieving better solutions for the optimization problem.

Two single-layer barrel vault frames with different patterns are optimized via the presented approach. In the process of optimization, contrary to size variables, shape is a continuous variable. In the case of shape optimization of this type of space structures, since all of the nodal coordinates of the shape variables are dependent on the height-to-span ratio of the barrel vault, height is considered as the only shape variable in a constant span of barrel vault.

In comparison, the best height-to-span ratios of barrel vaults under static loading conditions obtained from CSS, MCSS, and IMCSS algorithms are approximately close to value of 0.17 from comparative study carried out by Parke. Furthermore, as seen in the results, different patterns of barrel vaults have different effects on the value of best height-to-span ratio. Moreover, in comparison to CSS and MCSS algorithms, IMCSS has found more optimal values for the weight of structures in a lower number of analyses.

Since SAP2000 is a powerful software in modeling, analyzing, and designing of large-scale spatial structures, OAPI would be a profit interface tool between this software and MATLAB in the process of structural optimization.

References

1. Kaveh A, Mirzaei B, Jafarvand A (2014) Shape-size optimization of single-layer barrel vaults using improved magnetic charged system search. Int J Civil Eng 12(4):447–465
2. Kaveh A (2017) Advances in metaheuristic algorithms for optimal design of structures, 2nd edn. Springer, Switzerland
3. Kaveh A, Eftekhar B (2012) Improved big bang big crunch to optimize barrel vault frames. In: Proceeding of the 9th International Congress on civil engineering, Isfahan University of Technology (IUT), Isfahan, Iran

4. Kaveh A, Mirzaei B, Jafarvand A (2013) Optimal design of single-layer barrel vault frames using improved magnetic charged system search. Int J Optim Civil Eng 3(4):575–600
5. Parke GAR (2006) Comparison of the structural behaviour of various types of braced barrel vaults. In: Makowski ZS (ed) Analysis, design and construction of braced barrel vaults. Taylor & Francis e-Library, Hoboken, NJ, pp 113–144
6. Kaveh A, Talatahari S (2010) Optimal design of truss structures via the charged system search algorithm. Struct Multidisip Optim 37(6):893–911
7. Kaveh A, Motie Share MA, Moslehi M (2013) Magnetic charged system search: a new metaheuristic algorithm for optimization. Acta Mech 224(1):85–107
8. American Institute of Steel Construction (AISC) (2005) Construction manual, 13th edn. AISC, Chicago, IL
9. British Standards Institution (BSI) (2000) Structural use of steelwork in building Part 1: code of practice for design-rolled and welded section (BS5950-1). British Standards Institution, London
10. Kameshki ES, Saka MP (2007) Optimum geometry design of nonlinear braced domes using genetic algorithm. Comput Struct 85:71–79
11. Kaveh A, Farahmand Azar B, Talatahari S (2008) Ant colony optimization for design of space trusses. Int J Space Struct 23(3):167–181
12. Kaveh A, Mirzaei B, Jafarvand A (2014) Optimal design of double layer barrel vaults using improved magnetic charged system search. Asian J Civil Eng (BHRC) 15(1):135–154
13. Kaveh A, Bakhshpoori T, Ashoory M (2012) An efficient optimization procedure based on cuckoo search algorithm for practical design of steel structures. Int J Optim Civil Eng 2(1):1–14
14. Computers and Structures Inc. (CSI) (2011) SAP2000 OAPI documentation. University of California, Berkeley, CA
15. American National Standards Institute (ANSI) (1980) Minimum design loads for buildings and other structures (ANSI A58.1)
16. American Society of Civil Engineers (ASCE) (2010) Minimum design loads for buildings and other structures (ASCE-SEI 7-10)
17. American Institute of Steel Construction (AISC) (1994) Manual of steel construction—load and resistance factor design (AISC-LRFD), 2nd edn. AISC, Chicago, IL

Chapter 8
Optimal Design of Double-Layer Barrel Vaults Using CBO and ECBO Algorithms

8.1 Introduction

Barrel vault is one of the oldest architectural forms, used since antiquity. The brick architecture of the Orient or the masonry construction of the Romans provides numerous examples of the structural use of barrel vaults. The industrial and technological developments which have taken place during the last three decades have had a far-reaching effect upon contemporary architecture and modern engineering. New building techniques, new constructional materials, and new structural forms have been introduced all over the world. The architectural search for new structural forms has resulted in the widespread use of three-dimensional structures. The evolution of effective computer techniques of analysis is undoubtedly one of the reasons for the truly phenomenal acceptance of space structures. During recent years, architects and engineers have rediscovered the advantages of barrel vaults as viable and often highly suitable forms for covering not only low-cost industrial buildings, warehouses, large-span hangars, and indoor sports stadiums but also large cultural and leisure centers. The impact of industrialization on prefabricated barrel vaults has proved to be the most significant factor leading to lower costs for these structures. A barrel vault consists of one or more layers of elements that are arched in one direction [1]. Barrel vaults are given different names depending on the way their surface is formed. The earlier types of barrel vaults were constructed as single-layer structures [2–4]. Nowadays, with increase of the spans, double-layer systems are often preferred. Whereas the single-layer barrel vaults are mainly under the action of flexural moments, the component members of double-layer barrel vaults are almost exclusively under the action of axial forces; the elimination of bending moments leads to a full utilization of strength of all the elements. Formex algebra is a mathematical system that provides a convenient medium for configuration processing. The concepts are general and can be used in many fields. In particular, the ideas may be employed for generation of information about various aspects of structural systems such as element connectivity, nodal coordinates, details of

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_8

loadings, joint numbers, and support arrangements. The information generated may be used for various purposes, such as graphic visualization or input data for structural analysis. Double-layer barrel vaults have great number of structural elements, and utilizing optimization techniques has considerable influence on the economy.

Methods of optimization can be divided into two general categories of gradient-based methods and metaheuristic algorithms. Many of gradient-based optimization algorithms have difficulties when dealing with large-scale optimization problems. To overcome these difficulties, utilizing metaheuristic algorithms is inevitable. The formulation of metaheuristic algorithms is often inspired by either natural phenomena or physical laws. A metaheuristic algorithm consists of two phases: exploration of the search space and exploitation of the best solutions found. One of the main problems in developing a good metaheuristic algorithm is to maintain a reasonable balance between the exploration and exploitation abilities. In the past decades, structural optimization has been studied by using different metaheuristic algorithms [5]. Colliding bodies optimization (CBO) is a new metaheuristic search algorithm that is developed by Kaveh and Mahdavi [6]. CBO is based on the governing laws of one-dimensional collision between two bodies in the physics that one object collides with the other object and they move toward a minimum energy level. CBO is simple in concept, depends on no internal parameters, and does not use memory for saving the best-so-far solutions. The enhanced colliding bodies optimization (ECBO) is introduced by Kaveh and Ilchi Ghazaan [7], and it uses memory to save some historically best solutions to improve the CBO performance without increasing the computational cost. In this method, some components of agents are also changed to jump out from local minima. In this chapter, the performance of the CBO and ECBO on optimal design of double-layer barrel vaults is examined. The design algorithm is supposed to obtain minimum weight grid through suitable selection of tube sections available in AISC-LRFD [8]. The strength and stability requirements of steel members are imposed according to AISC-ASD [9].

The remainder of this chapter is organized as follows: In Sect. 8.2, the mathematical formulation of the structural optimization problems is presented and a brief explanation of the AISC-ASD is provided. Section 8.3 includes an explanation of the CBO and ECBO algorithms. In Sect. 8.4 structural models are explained and three numerical examples are presented. The last section concludes the chapter.

8.2 Optimum Design of Double-Layer Barrel Vaults

The allowable cross sections are considered as 37 steel pipe sections shown in Table 8.1, where the abbreviations ST, EST, and DEST stand for standard weight, extra strong, and double extra strong, respectively. These sections are taken from AISC-LRFD [8] which is also utilized as the code of design.

The aim of optimizing the truss structures is to find a set of design variables that has the minimum weight satisfying certain constraints. This can be expressed as

Table 8.1 The allowable steel pipe sections taken from AISC-LRFD

	Type	Nominal diameter (in)	Weight per ft (lb)	Area (in^2)	I (in^4)	Gyration radius (in)	J (in^4)
1	ST	½	0.85	0.25	0.017	0.261	0.034
2	EST	½	1.09	0.32	0.02	0.250	0.040
3	ST	¾	1.13	0.333	0.037	0.334	0.074
4	EST	¾	1.47	0.433	0.045	0.321	0.090
5	ST	1	1.68	0.494	0.087	0.421	0.175
6	EST	1	2.17	0.639	0.106	0.407	0.211
7	ST	1¼	2.27	0.669	0.195	0.54	0.389
8	ST	1½	2.72	0.799	0.31	0.623	0.620
9	EST	1¼	3.00	0.881	0.242	0.524	0.484
10	ST	2	3.65	1.07	0.666	0.787	1.330
11	EST	1½	3.63	1.07	0.391	0.605	0.782
12	EST	2	5.02	1.48	0.868	0.766	1.740
13	ST	2½	5.79	1.7	1.53	0.947	3.060
14	ST	3	7.58	2.23	3.02	1.16	6.030
15	EST	2½	7.66	2.25	1.92	0.924	3.850
16	DEST	2	9.03	2.66	1.31	0.703	2.620
17	ST	3½	9.11	2.68	4.79	1.34	9.580
18	EST	3	10.25	3.02	3.89	1.14	8.130
19	ST	4	10.79	3.17	7.23	1.51	14.50
20	EST	3½	12.50	3.68	6.28	1.31	12.60
21	DEST	2½	13.69	4.03	2.87	0.844	5.740
22	ST	5	14.62	4.3	15.2	1.88	30.30
23	EST	4	14.98	4.41	9.61	1.48	19.20
24	DEST	3	18.58	5.47	5.99	1.05	12.00
25	ST	6	18.97	5.58	28.1	2.25	56.3
26	EST	5	20.78	6.11	20.7	1.84	41.3
27	DEST	4	27.54	8.1	15.3	1.37	30.6
28	ST	8	28.55	8.4	72.5	2.94	145
29	EST	6	28.57	8.4	40.5	2.19	81
30	DEST	5	38.59	11.3	33.6	1.72	67.3
31	ST	10	40.48	11.9	161	3.67	321
32	EST	8	43.39	12.8	106	2.88	211
33	ST	12	49.56	14.6	279	4.38	559
34	DEST	6	53.16	15.6	66.3	2.06	133
35	EST	10	54.74	16.1	212	3.63	424
36	EST	12	65.42	19.2	362	4.33	723
37	DEST	8	72.42	21.3	162	2.76	324

ST Standard weight; *EST* Extra strong; *DEST* Double extra strong

$$\text{Find } \{X\} = [x_1, x_2, x_3, \ldots, x_{ng}], \quad x_i \in D = \{d_1, d_2, d_3, \ldots, d_{37}\}$$
$$\text{To minimize } W(\{X\}) = \sum_{i=1}^{ng} x_i \sum_{j=1}^{nm(i)} \rho_j . L_j \tag{8.1}$$

The constraint conditions are briefly explained in the following:

$$\begin{gathered} \delta_{\min} < \delta_i < \delta_{\max}, \; i = 1, 2, \ldots, nn \\ \sigma_{\min} < \sigma_i < \sigma_{\max}, \; i = 1, 2, \ldots, nm \\ \sigma_i^b < \sigma_i < 0, \; i = 1, 2, \ldots, ns \end{gathered} \tag{8.2}$$

where $\{X\}$ is the set of design variables, ng is the number of member groups in structure (number of design variables), D is the list of cross-sectional areas available for groups according to Table 8.1, $W(\{X\})$ presents weight of the structure, $nm(i)$ is the number of members for the ith group, nn and ns are the number of nodes and number of compression elements, respectively, σ_i is the element stress and δ_i is the nodal displacement, and ρ_j and L_j denote the material density and the length for the jth member of the ith group, respectively. σ_i^{b} is the allowable buckling stress in member i when it is in compression. *min* and *max* mean the lower and upper bounds of constraints, respectively.

The penalty function can be defined as

$$f_{\text{cost}}(\{X\}) = (1 + \epsilon_1 . v)^{\epsilon_2} \times W(\{X\}), \; v = \sum_{i=1}^{nn} v_i^d + \sum_{i=1}^{nm} \left(v_i^\sigma + v_i^\lambda \right) \tag{8.3}$$

where v is the constraint violations function, v_i^d, v_i^σ, and v_i^λ are constraint violations for displacement, stress, and slenderness ratio, respectively, ϵ_1 and ϵ_2 are penalty function exponents which were selected considering the exploration and exploitation rate of the search space. Here, ϵ_1 is set to unity; ϵ_2 is selected in a way that it decreases the penalties and reduces the cross-sectional areas. Thus, in the first steps of the search process, ϵ_2 is set to 1.5 and it linearly increases to 3 [10].

The allowable tensile and compressive stresses are used according to the AISC-ASD code [9], as follows:

$$\begin{cases} \sigma_i^+ = 0.6 F_{\mathrm{y}} & for\ \sigma_i \geq 0 \\ \sigma_i^- & for\ \sigma_i < 0 \end{cases} \tag{8.4}$$

where σ_i^- is calculated according to the slenderness ratio:

$$\sigma_i^- = \begin{cases} \left[\left(1 - \frac{\lambda_i^2}{2C_c^2}\right)F_y\right] \Big/ \left(\frac{5}{3} + \frac{3\lambda_i}{8C_c} - \frac{\lambda_i^3}{8C_c^3}\right) & \text{for } \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_c \end{cases} \tag{8.5}$$

where E is the modulus of elasticity, F_y is the yield stress of steel, C_c is the slenderness ratio (λ_i) dividing the elastic and inelastic buckling regions $\left(C_c = \sqrt{\frac{2\pi^2 E}{F_y}}\right)$, λ_i is the slenderness ratio ($\lambda_i = kL_i/r_i$), k is the effective length factor, L_i is the member length, and r_i is the radius of gyration.

According to AISC-ASD, the allowable slenderness ratio can be formulated as follows:

$$\begin{aligned} &\lambda_c = KL_i/r_i \leq 200 \quad \text{for compression members} \\ &\lambda_t = KL_i/r_i \leq 300 \quad \text{for tension members} \end{aligned} \tag{8.6}$$

where K is the effective length factor for the members and equal to 1 for all truss members. L_i and r_i are the length and minimum radius of gyration for the member i, respectively.

8.3 CBO and ECBO Algorithms

Colliding Bodies Optimization (CBO) is a new population-based stochastic optimization algorithm based on the governing laws of one-dimensional collision between two bodies in physics [6]. Each agent is modeled as a body with a specified mass and velocity. A collision occurs between pairs of objects to find the global or near-global solutions. Enhanced colliding bodies optimization (ECBO) uses memory to save some best solutions and utilizes a mechanism to escape from local optima [11].

8.3.1 A Brief Explanation and Formulation of the CBO Algorithm

In CBO, each solution candidate X_i containing a number of variables (i.e., $X_i = \{X_{i,j}\}$) is considered as a colliding body (CB). The massed objects are composed of two main equal groups: stationary and moving objects, where the moving objects move to follow stationary objects and a collision occurs between pairs of objects (Fig. 8.1). This is done for two purposes: (i) to improve the positions of moving objects and (ii) to push stationary objects toward better positions. After the collision, new positions of colliding bodies are updated based on new velocity by using the collision laws governed by the laws of momentum and energy [6]. When a collision

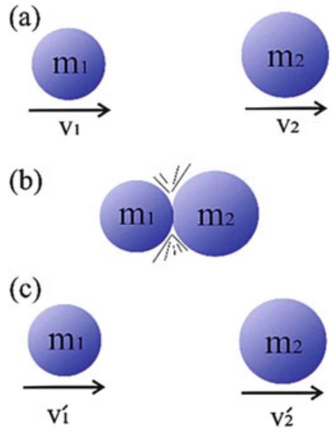


Fig. 8.1 Collision between two bodies: (**a**) before collision, (**b**) during collision, and (**c**) after collision

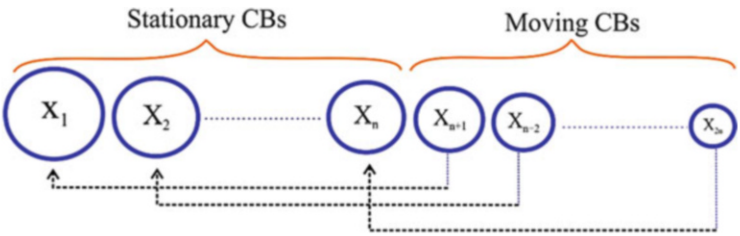


Fig. 8.2 The sorted CBs in an ascending order and the mating process for the collision

occurs in an isolated system, the total momentum of the system of objects is conserved. Provided that there are no net external forces acting upon the objects, the momentum of all objects before the collision equals the momentum of all objects after the collision.

CBO starts with an initial population consisting of $2n$ parent individuals created by means of a random initialization. Then, CBs are sorted in ascending order based on the value of cost function as shown in Fig. 8.2.

The CBO procedure can briefly be outlined as follows:

As stated before each agent called CB has a specified mass, which is defined as

$$m_k = \frac{\frac{1}{\mathrm{fit}(k)}}{\sum_{i=1}^{n} \frac{1}{\mathrm{fit}(i)}}, \quad k = 1, 2, \ldots, n \tag{8.7}$$

where $\mathrm{fit}(i)$ represents the objective function value of the ith CB and n is the number of colliding bodies. After sorting colliding bodies according to their objective function values in an increasing order, two equal groups are created: (i) stationary group and (ii) moving group (Fig. 8.2). Moving objects collide with stationary objects to improve their positions and push stationary objects toward better positions. The velocities of the stationary and moving bodies before collision (v_i) are calculated by

$$v_i = 0,\ i = 1, \ldots, \frac{n}{2} \tag{8.8}$$

$$v_i = x_{i-\frac{n}{2}} - x_i,\ \ i = \frac{n}{2} + 1, \frac{n}{2} + 2 \ldots, n \tag{8.9}$$

where x_i is the position vector of the ith CB. The velocity of stationary and moving CBs after the collision (v_i') is evaluated by

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}}\ \ i = 1, 2, \ldots, \frac{n}{2} \tag{8.10}$$

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}}\ \ i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{8.11}$$

$$\varepsilon = 1 - \frac{\text{iter}}{\text{iter}_{\max}} \tag{8.12}$$

where ε is the coefficient of restitution (COR) and iter and $\text{iter}_{\max}$ are the current iteration number and the total number of iterations for optimization process, respectively. New positions of each group are stated by the following formulas:

$$x_i^{\text{new}} = x_i + rand \circ v_i',\ \ i = \ 1, 2, \ldots, \frac{n}{2} \tag{8.13}$$

$$x_i^{\text{new}} = x_{i-\frac{n}{2}} + rand \circ v_i',\ \ i = \ \frac{n}{2} + 1, \ldots, n \tag{8.14}$$

where x_i^{new}, x_i and v_i' are the new position, previous position, and the velocity after the collision of the ith CB, respectively. *rand* is a random vector uniformly distributed in the range of [−1,1] and the sign "∘" denotes an element-by-element multiplication.

8.3.2 Pseudo-Code of the ECBO Algorithm

In the Enhanced Colliding Bodies Optimization (ECBO), a memory that saves a number of historically best CBs is utilized to improve the performance of the CBO and reduce the computational cost. Furthermore, ECBO changes some components of CBs randomly to prevent premature convergence [12]. In this section, in order to introduce the ECBO algorithm, the following steps should be taken.

8.3.2.1 Initialization

Step 1: The initial locations of CBs are created randomly in an m-dimensional search space.

$$x_i^0 = x_{\min} + random \circ (x_{\max} - x_{\min}), \quad i = 1, 2, 3, \ldots, n \tag{8.15}$$

where x_i^0 is the initial solution vector of the ith CB. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable limits vectors, respectively, and random is a random vector with each component being in the interval [0,1].

8.3.2.2 Search

Step 1: The value of the mass for each CB is calculated by Eq. (8.7).
Step 2: Colliding Memory (CM) is considered to save some historically best CB vectors and their related mass and objective function values. The size of the CM is taken as $n/10$ (n is the population size) in this study. At each iteration, solution vectors that are saved in the CM are added to the population and the same number of the current worst CBs are deleted.
Step 3: CBs are sorted according to their objective function values in an increasing order. To select the pairs of CBs for collision, they are divided into two equal groups: (i) stationary group and (ii) moving group.
Step 4: The velocities of stationary and moving bodies before collision are evaluated by Eqs. (8.8) and (8.9), respectively.
Step 5: The velocities of stationary and moving bodies after collision are calculated by Eqs. (8.10) and (8.11), respectively.
Step 6: The new location of each CB is evaluated by Eqs. (8.13) or (8.14).
Step 7: A parameter like ***Pro*** within (0, 1) is introduced which specifies whether a component of each CB must be changed or not. For each CB ***Pro*** is compared with rn_i ($i = 1, 2, \ldots, n$) which is a random number uniformly distributed within (0, 1). If $rn_i <$ ***Pro***, one dimension of ith CB is selected randomly and its value is regenerated by

$$x_{ij} = x_{j,\min} + random.\left(x_{j,\max} - x_{j,\min}\right) \tag{8.16}$$

where x_{ij} is the jth variable of the ith CB. $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the jth variable. In this chapter, the value of ***Pro*** is set to 0.3.

8.3.2.3 Terminating Condition Check

Step 1: After the predefined maximum evaluation number, the optimization process is terminated [11].

8.4 Numerical Examples

In this section, two kinds of double-layer barrel vaults are optimized by CBO and ECBO algorithms and the results are compared with the engineering design which was found by SAP2000 to show the efficiency of these algorithms. SAP2000 software has a toolbox for the auto and fully stressed design according to the related provisions. Auto select section lists are lists of previously defined steel sections (including cold-formed steel). When an auto select section list is assigned to a frame member, the program can automatically select the most economical, adequate section from the auto select section list when designing the member. The first example is a 384-bar double-layer barrel vault, which was optimized by Kaveh et al. [13] using continuous variables under two types of loadings. The second one is a 910-bar double-layer braced barrel vault introduced as a new type. Two problems are solved utilizing discrete variables for the purpose of practical design. All connections are assumed as ball-jointed, and top-layer joints are subjected to concentrated vertical loads. Stress and slenderness constraints [Eqs. (8.4), (8.5), and (8.6)] are according to AISC-ASD provisions, and displacement limitations of ± 0.1969 in (5 mm) are imposed on all nodes in x-, y-, and z-directions. The modulus of elasticity is considered as 30,450 ksi (210,000 MPa), and the yield stress of steel is taken as 58 ksi (400 MPa).

In CBO and ECBO, the population of $n = 30$ CBs is utilized, and the size of colliding memory is considered as $n/10$ that is taken as 3 for ECBO. The predefined maximum evaluation number is considered as 30,000 analyses for all examples. Because of the stochastic nature of the algorithms, each example is solved 5 times independently. In all problems, CBs are allowed to select discrete values from the permissible list of cross sections (real numbers are rounded to the nearest integer in each iteration). The algorithms are coded in MATLAB, and the structures are analyzed using the direct stiffness method. The computational time is measured in terms of CPU time of a PC with the processor of Intel® Core™ i7-3612 QM @ 2.1 GHz equipped with 6 GBs of RAM.

8.4.1 A 384-Bar Double-Layer Barrel Vault

Similar to the flat double-layer grids, double-layer barrel vaults consist of a top and bottom layer connected to each other by bracing members. The top/bottom layers are also called the "chord members." All the flat double-layer configurations can also be used for doublelayer braced barrel vaults. The 384-bar double-layer barrel vault is the first example; this structure consists of two rectangular nets, and for making it stable, angles of the bottom nets are put into the center of one of the above nets, and these are connected through diametrical elements as shown in Fig. 8.3a. This example is subjected to two types of loadings. Case 1 is a symmetric loading condition where the vertical concentrated loads of −20 kips (−88.964 kN) are

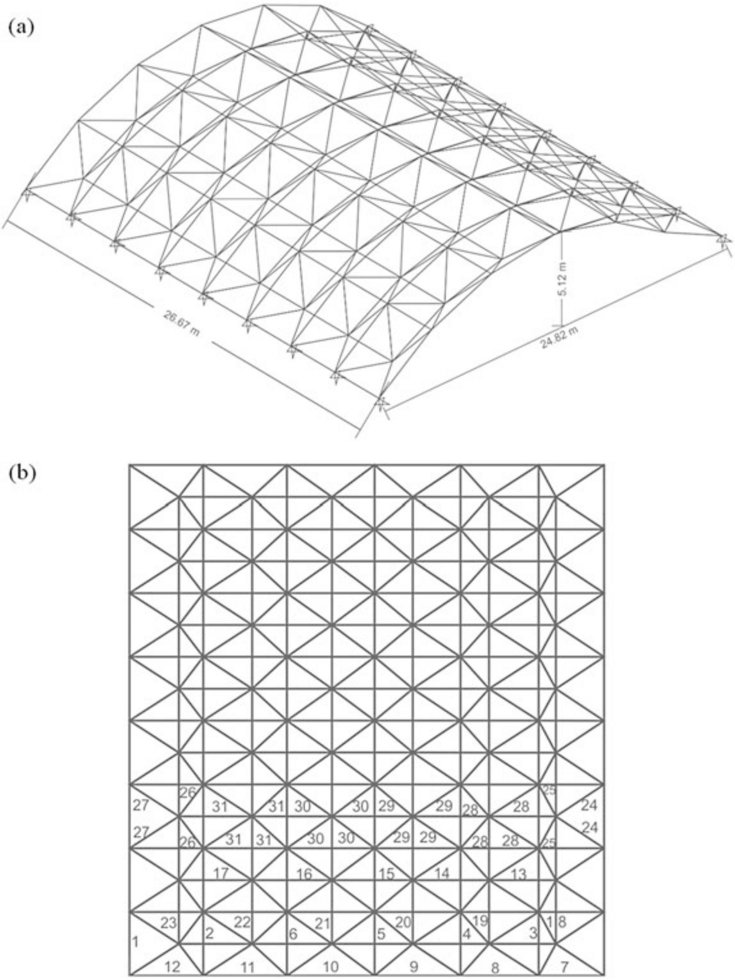


Fig. 8.3 Schematic of the 384-bar double-layer barrel vault: (**a**) 3D view and (**b**) element grouping (plan view)

applied on free joints (nonsupport joints) of top layer. In Case 2, which is asymmetric, the concentrated loads of −10 kips (−44.482 kN) are applied at the right-hand half and at the left-hand half of the structure the loads of −6 kips (−26.689 kN) are applied on nonsupport top layer joints, respectively. All members of this double-layer barrel vault are categorized into 31 groups, as shown in Fig. 8.3b, and the supports are considered at the two external edges of the top layer of the barrel vault.

Tables 8.2 and 8.3 show the best design vectors and the corresponding weights for the two methods, for the Case 1 and Case 2 loading conditions, respectively. In Case 1 (Symmetric loading condition), ECBO could find the weight which is 2.2 %

Table 8.2 Optimal design of the 384-bar double-layer barrel vault for Case 1

		Optimum section (designations)	
Group number	Engineering design	CBO	ECBO
1	ST 1¼	ST 1¼	ST 1¼
2	EST 2	EST 2	ST 2½
3	EST 2	EST 3	EST 2
4	ST 1¼	ST 1¼	ST 1¼
5	EST 4	DEST 2	DEST 2½
6	DEST 8	EST 1½	ST 1¼
7	ST 12	EST 10	EST 12
8	EST 8	DEST 5	DEST 5
9	ST 10	DEST 6	ST 10
10	EST 10	EST 10	ST 12
11	ST 8	DEST 5	ST 8
12	EST 8	ST 12	ST 12
13	EST 5	EST 5	DEST 4
14	ST 8	ST 5	ST 6
15	ST 3½	ST 3½	ST 3
16	ST 6	DEST 2½	DEST 2½
17	ST 8	EST 5	EST 5
18	EST 1½	EST 2	ST 2
19	ST 1¼	ST 1¼	ST 1¼
20	EST 2	ST 1¼	ST 2
21	EST 2	EST 2	EST 1¼
22	EST 2	ST 1¼	ST 1¼
23	EST 2	EST 2	ST 2
24	ST 4	EST 3	ST 4
25	ST 2½	EST 2	EST 2½
26	ST 3	EST 2	ST 2½
27	DEST 2½	DEST 2	ST 3½
28	ST 2½	EST 2	EST 2
29	ST 2½	ST 2½	ST 2
30	EST 2	EST 2	ST 2½
31	EST 2	EST 2	EST 2
Demand/capacity ratio limit	0.999	–	–
Max stress ratio	0.559	0.7649	0.8773
Max displacement ratio	0.9997	0.9994	0.9999
Best weight (kg)	32,259.90	29,057.93	28,415.20
Mean weight (kg)	–	33,465.09	29,900.15
Computation time (s)	–	296	291

lighter than CBO and 11.9 % lighter than Engineering design which was found by SAP2000. In Case 2 (Asymmetric loading condition), this percentage was equal to 10.7 % and 19.7 % better than CBO and Engineering design, respectively. It is also

Table 8.3 Optimal design of the 384-bar double-layer barrel vault for Case 2

Group number	Engineering design	Optimum section (designations)	
		CBO	ECBO
1	ST 1¼	ST 1¼	ST 1¼
2	EST 2	EST 2	ST 2
3	ST 1¼	ST 1¼	ST 1¼
4	DEST 2	ST 3½	ST 3½
5	EST 2	EST 2	EST 2
6	EST 2	EST 2	ST 2
7	EST 5	EST 5	EST 6
8	EST 5	ST 8	DEST 4
9	EST 5	DEST 3	EST 3
10	ST 5	ST 3½	ST 3
11	ST 5	ST 3½	ST 3
12	ST 5	EST 3½	ST 4
13	DEST 2	EST 1½	EST 1½
14	ST 2½	ST 3½	ST 2½
15	EST 3	ST 4	ST 4
16	DEST 2½	EST 3½	ST 5
17	ST 5	ST 4	EST 3½
18	EST 1½	ST 1½	ST 1½
19	EST 2	EST 2	ST 2
20	EST 2	EST 2	ST 2
21	EST 2	EST 2	ST 2
22	ST 2½	EST 2	ST 2
23	ST 1½	EST 2	ST 1¼
24	EST 1¼	EST 1	ST 1
25	EST 2	EST 2	ST 1½
26	EST 1½	ST 1½	ST 1½
27	EST 2	EST 2½	ST 2
28	EST 2	EST 2	ST 2
29	EST 2½	EST 2	ST 2
30	DEST 2	ST 2½	ST 3
31	ST 2½	EST 2	ST 2
Demand/capacity ratio limit	0.999	–	–
Max stress ratio	0.888	0.7176	0.9372
Max displacement ratio	0.9962	0.9991	0.9996
Best weight (kg)	16,617.81	14,940.13	13,345.92
Mean weight (kg)	–	18,602.01	15,856.61
Computation time (s)	–	301	299

worthwhile to mention that CBO results were 9.9 % and 10.1 % better than Engineering design for Case 1 and Case 2 loading condition, respectively. It can be observed that ECBO has better performance than CBO without increasing the

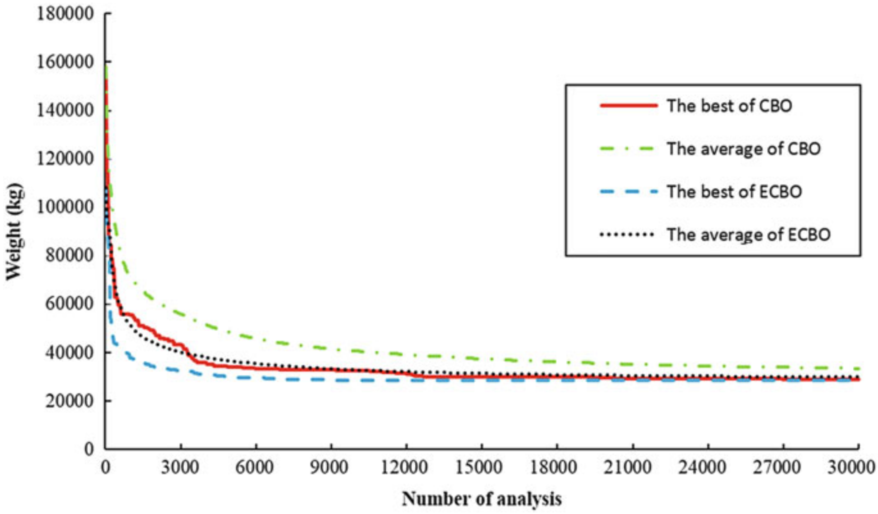


Fig 8.4 Convergence curves for the 384-bar double-layer barrel vault (Case 1)

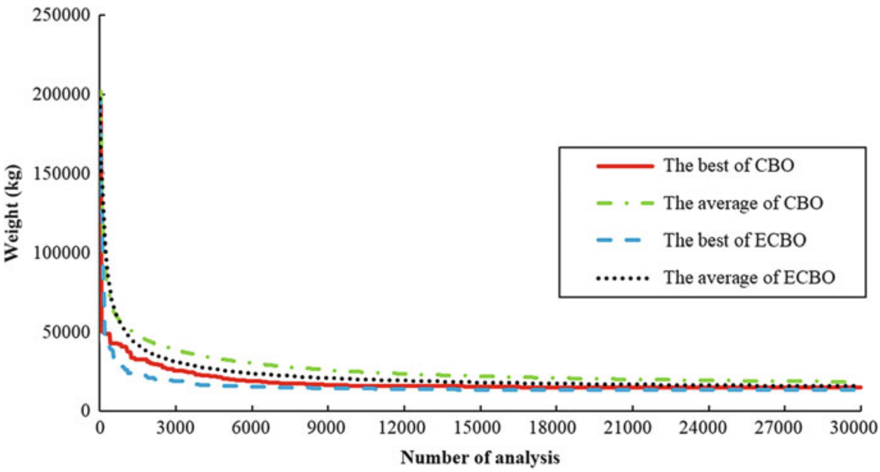


Fig 8.5 Convergence curves for the 384-bar double-layer barrel vault (Case 2)

computational cost. For graphical comparison of the algorithms, Figs. 8.4 and 8.5 illustrate the convergence curves for the Case 1 and Case 2 loading conditions by the proposed methods, respectively.

8.4.2 A 910-Bar Double-Layer Braced Barrel Vault

Braced barrel vaults consist of developable surfaces generated by the repetitive use of a curve known as "directrix" over a generator straight line. The directrix may be a circular arc, an ellipse, a catenary, a parabola, or a cycloid. Most of braced barrel vaults built in practice are part of a right circular cylinder, which may be either supported by columns or simply springing from the ground surface. The semicircular barrel vaults have the clear advantage of facilitating water drainage and providing strong architectural form recognition. Under loads, braced barrel vaults may behave in two different modes: arch and beam, depending mainly on the location of supports. The braced barrel vault behaves as an arch when supported along the sides. The braced barrel vault behaves in the beam mode when it is supported at its ends. In this case, the longitudinal compression forces occur near the crown and longitudinal tensile forces toward the free edge. If the braced barrel vault is supported at the four corners, it behaves as a combined beam and arch under loads. In this case, it acts as a series of arches in cross-section direction and as a beam longitudinally.

In this section, one type of braced barrel vault which contains 266 nodes and 910 members is introduced as the last example. The structural members are divided into 30 groups as shown in Fig. 8.6a, and the other related details are shown in Fig. 8.6b and c.

The uniformity of the distribution of stiffness in the vicinity of the structure is an important issue for large-scale structures. If part of the structure has elements of low axial forces and small displacements (low cross sections), and another part contains elements of high cross sections, then the uniformity of the distribution of the stiffness will not be achieved. For this reason, the element grouping is selected according to two symmetry lines of the configuration leading to uniform distribution of stiffness for the entire structure. The loading conditions consist of the following:

1. At the nodes of central arc, a downward concentrated load of −15 kips (−66.72 kN).
2. At the nodes of the arcs adjacent to the central arc, a downward concentrated load of −10 kips (44.48 kN).
3. At the nodes of arcs adjacent to the external arcs, a downward concentrated load of −5 kips (−22.24 kN).
4. At the nodes of external arcs, a downward concentrated load of −2 kips (−8.90 kN).

All external and internal side nodes are simply supported, and for this reason, this double-layer braced barrel vault behaves as an arch. Table 8.4 lists the optimal values of 30 variables obtained by ECBO and CBO. The result of ECBO method is lighter than the result found by CBO. The optimum design for CBO and ECBO has the weights of 18,636 kg and 18,615 kg, respectively, and all optimum designs found by the algorithms satisfy the design constraints. The CBO and ECBO weights are 1258.77 kg (6.3 %) and 1279.12 kg (6.4 %) lighter than Engineering design,

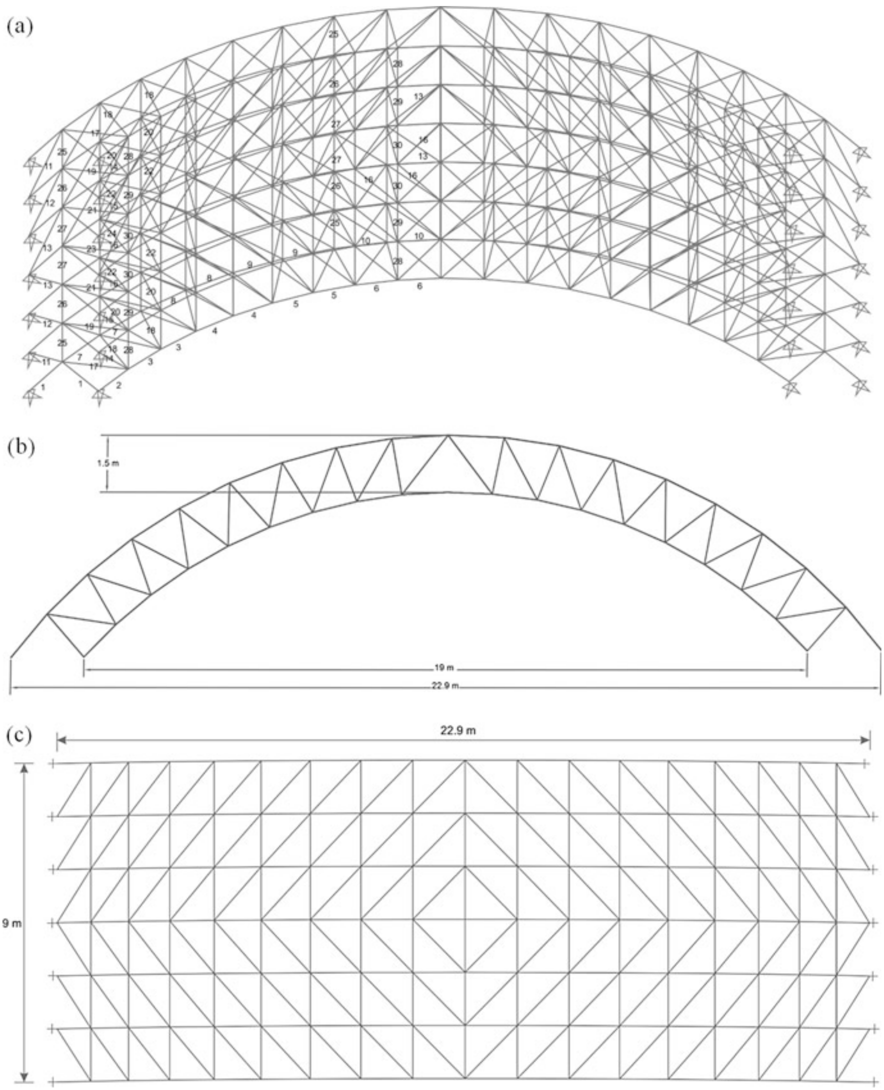


Fig 8.6 Schematic of the 910-bar double-layer braced barrel vault: (**a**) element grouping in 3D view, (**b**) front view, and (**c**) plan view

respectively. Convergence history of the present algorithms for the best optimum designs is depicted in Fig. 8.7.

Table 8.4 Optimal design of the 910-bar double-layer braced barrel vault

		Optimum section (designations)	
Group number	Engineering design	CBO	ECBO
1	DEST 2	EST 2	ST 3
2	DEST 5	ST 10	DEST 6
3	ST 8	ST 10	ST 10
4	ST 8	DEST 2½	ST 8
5	DEST 2	ST 3½	ST 2
6	EST ¾	EST ¾	ST 1½
7	ST 4	EST 3	EST 1½
8	ST 8	DEST 3	EST 3
9	ST 10	DEST 5	DEST 4
10	ST 12	DEST 6	EST 12
11	ST 1	ST 1¼	ST 1
12	ST 1	ST 1¼	ST 1
13	EST 1	ST 1¼	ST 1¼
14	ST 1	ST 1	ST 1
15	EST 2	ST 1	ST 1
16	ST 1	ST 3	EST 2
17	EST 1½	EST 1½	EST 2
18	EST 1½	EST 1	EST 1½
19	ST 1¼	EST 2	EST 3
20	EST 2	ST 2½	EST 2
21	EST 3½	ST 2½	EST 2
22	DEST 2½	ST 2½	EST 2
23	ST 5	EST 4	DEST 3
24	EST 5	ST 8	EST 5
25	ST ¾	ST ¾	ST ¾
26	ST ¾	ST ¾	ST ¾
27	ST ¾	ST½	ST ¾
28	ST ¾	ST ¾	EST ¾
29	EST 2	ST ¾	ST ¾
30	EST 1	EST 2	EST 1½
Demand/capacity ratio limit	0.999	–	–
Max stress ratio	0.95	0.9767	0.9818
Max displacement ratio	0.9993	0.9990	0.9978
Best weight (kg)	19,894.44	18,635.67	18,615.32
Mean weight (kg)	–	23,806.75	22,442.64
Computation time (s)	–	975	926

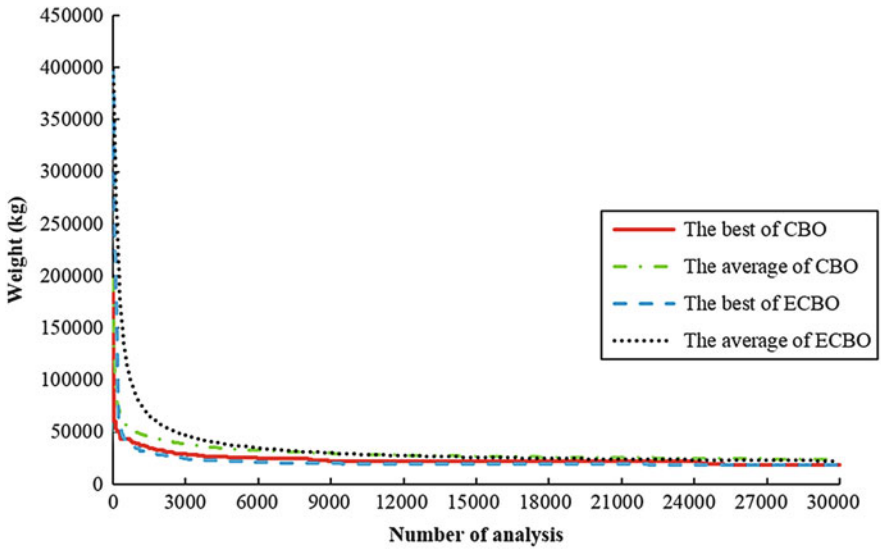


Fig. 8.7 Convergence curves for the 910-bar double-layer braced barrel vault

8.5 Concluding Remarks

This chapter utilizes two newly developed, simple, and efficient metaheuristic algorithms for discrete optimization of double-layer barrel vaults. The CBO has simple structure and depends on no internal parameter and does not use memory for saving the best-so-far solutions. In order to improve the exploration capabilities of the CBO and to prevent premature convergence, a stochastic approach is employed in ECBO that changes some components of CBs randomly. Colliding Memory is also utilized to save a number of the so-far-best solutions to reduce the computational cost. In order to indicate the similarities and differences between the characteristics of the CBO and ECBO algorithms, two types of double-layer barrel vaults are examined. Structures are designed in accordance with AISC-ASD specifications and displacement constraints. In both examples, the discrete variables are assigned to each group for the purpose of practical design and selected from available steel pipe section table. ECBO has better performance in all cases than CBO because of the reliability of search, solution accuracy, and speed of convergence. It can be also stated that both CBO and ECBO have better efficiency in finding results than SAP2000 in all cases. It is also worthwhile to mention that all designs are governed by displacements because of large vertical displacements at the apex of these structures. Furthermore, the results show that CBO and ECBO are robust optimization tools for optimum practical design of large-scale structures like double-layer barrel vaults.

References

1. Kaveh A, Moradveisi M (2016) Optimal design of double-layer barrel vaults using CBO and ECBO algorithms. Iran J Sci Technol Trans Civil Eng. doi:10.1007/s40996-016-0021-4
2. Makowski ZS (2006) Analysis, design and construction of braced barrel vaults. Taylor & Francis e-Library, Hoboken, NJ
3. Kaveh A, Mirzaei B, Jafarvand A (2013) Optimal design of single-layer barrel vault frames using improved magnetic charged system search. Int J Optim Civil Eng 3(3):575–600
4. Kaveh A, Mirzaei B, Jafarvand A (2014) Shape-size optimization of single-layer barrel vaults using improved magnetic charged system search. Int J Civil Eng IUST 12(4):447–465
5. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer, Switzerland
6. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel metaheuristic method. Comput Struct 139:18–27
7. Kaveh A, Ilchi Ghazaan M (2014) Computer codes for colliding bodies optimization and its enhanced version. Int J Optim Civil Eng 4:321–332
8. American Institute of Steel Construction (AISC) (1994) Manual of steel construction load resistance factor design, 2nd edn. AISC, Chicago, IL
9. American Institute of Steel Construction (AISC) (1989) Manual of steel construction allowable stress design (ASD-AISC), 9th edn. AISC, Chicago, IL
10. Kaveh A, Farahmand Azar B, Talatahari S (2008) Ant colony optimization for design of space trusses. Int J Space Struct 23:167–181
11. Kaveh A, Ilchi Ghazaan M (2015) A comparative study of CBO and ECBO for optimal design of skeletal structures. Comput Struct 153:137–147
12. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies algorithm for truss optimization with dynamic constraints. J Comput Civil Eng. doi:10.1061/(ASCE)CP.1943-5487.0000445, 04014104
13. Kaveh A, Mirzaei B, Jafarvand A (2014) Optimal design of double layer barrel vaults using improved magnetic charged system search. Asian J Civil Eng 15(1):135–154

Chapter 9
Optimum Design of Steel Floor Systems Using ECBO

9.1 Introduction

Decks, interior beams, edge beams, and girders are parts of a steel floor system. If the deck is optimized without considering beam optimization, finding the best result is simple. However, a deck with a higher cost may increase the composite action of the beams and decrease the beam cost, thus reducing the total expense. Also, a different number of floor divisions can improve the total floor cost. Increasing beam capacity by using castellated beams is another efficient cost-saving method. In this study, floor optimization is performed and these three issues are discussed. Floor division number and deck sections are some of the variables. Also, for each beam, profile section of the beam, beam-cutting depth, cutting angle, spacing between holes, and number of filled holes at the ends of castellated beams are other variables. Constraints include the application of stress, stability, deflection, and vibration limitations according to the load and resistance factor (LRFD) design. The objective function is the total cost of the floor consisting of the steel profile, cutting and welding, concrete, steel deck, shear stud, and construction costs. Optimization is performed by enhanced colliding bodies optimization (ECBO). Results show that using castellated beams, selecting a deck with a higher price and considering the different number of floor divisions can decrease the total cost of the floor (Kaveh and Ghafari [1]).

Many researchers have tried to optimize simple, composite, and castellated beams. Morton and Webber [2] used a relatively straightforward exhaustive search method to optimize composite beams. Klanšek and Kravanja [3] utilized the nonlinear programming (NLP) approach to optimize composite beams according to Euro-code 4 and conditions of both ultimate and serviceability limit states. Senouci and Al-Ansari [4] optimized composite beams by genetic algorithms according to AISC-LRFD. They also tried to find the effect of span and loading on the optimum result by a parametric study.

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_9

Cost optimization of floor systems is studied first by Adeli and Kim [5]. They utilized neural networks and mixed integer nonlinear programming according to the LRFD criteria. They also employed floating-point genetic algorithms to find the best results. Platt [6] used the evolver (genetic algorithm solving program) to parametric optimization of the floor. She considers the combination of configuration, size, topology, and spacing of truss girders and beams. Kaveh and Abadi [7] used an improved harmony search (HS) algorithm. They optimized a composite floor system consisting of reinforced concrete slab and steel I-beams according to AISC-LRFD rules. Poitras et al. [8] considered a complete floor system and utilized particle swarm optimization (PSO) for optimization. They found that composite action can be as economical as non-composite action depending on some conditions, and they used formed steel deck instead of normal concrete deck. Kaveh and Ahangaran [9] employed the social harmony search and found this new variant of HS to be better than other variants of it. Kaveh and Massoudi [10] optimized floors by ant colony optimization (ACO).

The main objective of the present chapter is to optimize the cost of the steel floor elements and to find the effect of the number of floor divisions, concrete thickness, and using castellated beams. This chapter is organized as follows: In Sect. 9.2, the design of structural elements of floor is introduced. Section 9.3 defines the optimization problem and identifies the variables, the constraints, and the objective function. The optimization algorithm is discussed in Sect. 9.4. Some numerical examples are introduced in Sect. 9.5. Finally, conclusions are extracted in Sect. 9.6.

9.2 Structural Floor Design

Structural elements are designed according to AISC-LRFD 10. Thus, the load combination W for stress and stability check is (ASCE [11])

$$W = 1.2\,\mathrm{DL} + 1.6\,\mathrm{LL}$$

where DL is the dead load and LL is the live load, and the load combination for serviceability criteria (deflection and vibration) is

$$W_{\mathrm{def}} = \mathrm{DL} + \mathrm{LL}$$

where W_{def} is the total loading for deflection calculation.

A composite castellated beam and a steel deck section (perpendicular to each other) are presented in Fig. 9.1. The deck should be designed independent of the beam as follows:

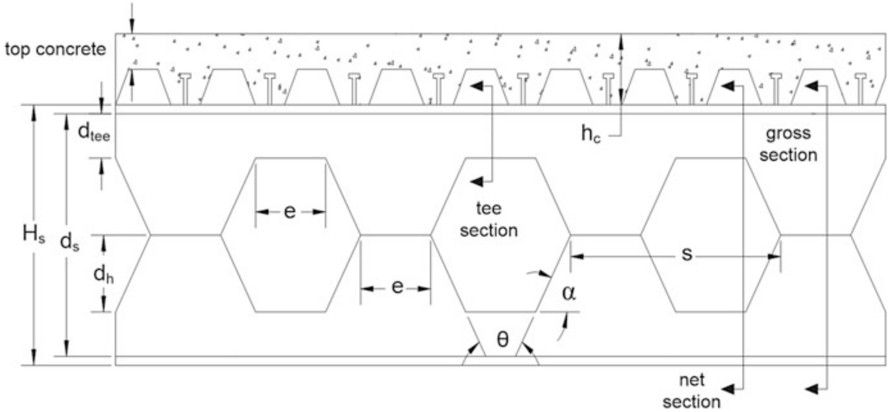


Fig. 9.1 Details of a composite castellated beam and steel deck

9.2.1 Deck Design

Deck span is the distance between two beams (B), and deck width is taken as 1 meter for the design. In this study, composite steel deck is used so that its section shape can guarantee the composite action roll formed steel decks and concrete. Also the shrinkage and temperature effects of the concrete are controlled by rebar. Due to the complex effect of roll formed steel decks, the partial composite action, and the wide variety of the produced sections, the specifications provided by the manufacturers should be used for determining their capacity.

9.2.2 Castellated Composite Beam Design

Castellated beams are produced by cutting rolled profile beam in special shape and welding them together in order to increase moment of inertia and moment capacity. Hexagonal cutting shape is one of the most popular cutting methods. But it is necessary to avoid keen corners because of stress concentration effects. Web openings of these beams produce some secondary effects, which can be controlled by filling end holes.

Composite beams are produced by composite interaction between concrete and steel. This composite action can help to increase the moment capacity of the beams. For designing this type of beams, first the effective width of the concrete slab should be calculated for interior beams, edge beams, and girders according to span and beam spacing (AISC [12]). Second, for the composite section, the center line must be calculated. For interior and edge beams, deck ribs are perpendicular to the beam axis, and top concrete (Fig. 9.1) must be considered only. However, for girders, the deck ribs are parallel to the beam axis and the entire concrete can be considered (AISC [12]). In this study, the center line, the moment of inertia, and the moment capacity of the composite section are determined by the superposition of the elastic stresses. For some stresses, stability, deflection, and vibration criteria must be checked as follows.

9.2.2.1 Stress Criteria

In this study, the unbraced length ratio of all beams is considered as zero. This is because the top flange of the beam is controlled by concrete slab.

The ultimate moment calculated for load combinations must be smaller than the nominal moment (AISC [12]):

$$\begin{aligned} M_{\mathrm{u}} < \varphi_{\mathrm{b}} M_{\mathrm{n}} &= \varphi_{\mathrm{b}} \times \min\left(M_{\text{n-con}}, M_{\text{n-st}}\right) \\ &= \varphi_{\mathrm{b}} \times \min\left(0.7F_{\mathrm{c}} Z_{\text{net-com-top}}, F_{\mathrm{y}} Z_{\text{net-com-bot}}\right) \end{aligned} \tag{9.1}$$

where M_{n} is the nominal moment capacity of the beam, $M_{\text{n-con}}$ is the nominal moment capacity (concrete limit), $M_{\text{n-st}}$ is the nominal moment capacity (steel limit), $Z_{\text{net-com-bot}}$ is the plastic modulus at the bottom of composite net section, $Z_{\text{net-com-top}}$ is the plastic modulus at the top of composite net section, φ_{b} is the bending reduction factor, F_{c} is the compressive strength of the concrete, and F_{y} is the yield strength of the steel.

Also the Vierendeel effect at unfilled holes produces secondary moment, and these two moments must satisfy the following equations:

$$m_{\mathrm{u}} = \frac{V_{\mathrm{u}} \times e}{4} \tag{9.2}$$

$$\frac{M_{\mathrm{u}}}{Z_{\text{net-com-bot}}} + \frac{m_{\mathrm{u}}}{Z_{\text{tee}}} < \varphi_{\mathrm{b}} F_{\mathrm{y}} \tag{9.3}$$

where m_{u} is the secondary shear ultimate moment, V_{u} is the ultimate shear force, e is the web post length, M_{u} is the ultimate moment, $Z_{\text{net-st}}$ is the plastic modulus of steel net section, and Z_{tee} is the plastic modulus of steel tee section. φ_{b} for concrete and steel are considered to be 0.9 (AISC [12]).

For a composite section, steel beams must resist shear forces alone (AISC [12]) as described in the following:

$$A_{\mathrm{W}} = d_{\mathrm{s}} \times t_{\mathrm{w}} \tag{9.4}$$

$$V_{\mathrm{u}} < \varphi_{\mathrm{v}} V_{\mathrm{n-w}} = \varphi_{\mathrm{v}} \times 0.6 F_{\mathrm{y}} A_{\mathrm{W}} C_{\mathrm{v}} \tag{9.5}$$

where A_{w} is the area of the net section web, t_{w} is the thickness of the web, d_{s} is the internal castellated beam height, V_{u} is the ultimate shear force, $V_{\mathrm{n-w}}$ is the nominal web shear capacity of net section, φ_{v} is the shear reduction factor, and C_{v} is the web shear coefficient.

Also the vertical shear capacity of the tee beams must be controlled by (AISC [12]):

$$A_{\mathrm{tee}} = d_{\mathrm{tee}} \times t_{\mathrm{w}} \tag{9.6}$$

$$\frac{V_{\mathrm{u}}}{2} < \varphi_{\mathrm{v}} V_{\mathrm{n-tee}} = \varphi_{\mathrm{v}} \times 0.6 F_{\mathrm{y}} A_{\mathrm{tee}} C_{\mathrm{v}} \tag{9.7}$$

where A_{tee} is the area of each tee section and $V_{\mathrm{n-tee}}$ is the nominal web shear capacity of the tee section.

Horizontal shear between holes in castellated beams must be checked as follows:

$$A_{\mathrm{he}} = e \times t_{\mathrm{w}} \tag{9.8}$$

$$V_{\mathrm{h}} = \frac{V_{\mathrm{u}} \times Q_{\mathrm{com}}}{I_{\mathrm{com}}} \times s < \varphi_{\mathrm{v}} V_{\mathrm{n-p}} = \varphi_{\mathrm{v}} \times 0.6 F_{\mathrm{y}} A_{\mathrm{he}} C_{\mathrm{v}} \tag{9.9}$$

where V_{h} is the horizontal shear at web post; Q_{com} and I_{com} are the first and second moments of inertia of the composite section, respectively; s is the spacing between the holes (Fig. 9.1); $V_{\mathrm{n-p}}$ is the nominal shear capacity of the web post; and φ_{v} and C_{v} are equal to 1 (AISC [12]).

When steel deck is used in a perpendicular position, Q_{com} and I_{com} must be considered for two conditions, because each choice may produce a greater shear force and a more critical condition:

(a) Considering the whole thickness of the concrete
(b) Considering the top thickness of the concrete

9.2.2.2 Stability Criteria

Horizontal shear may cause web plate buckling in the castellated beam (Kerdal and Nethercot [13]). According to the Structural Stability Research Council (SSRC), in-plane stress at the unfilled web must satisfy the following equations:

$$\begin{aligned}
&L_{\mathrm{b}} = 2d_{\mathrm{h}}\\
&r_{\mathrm{T}} = \frac{t_{\mathrm{w}}}{\sqrt{12}}\\
&C_{\mathrm{b}} = 1.75 + 1.05\frac{M_1}{M_2} + 0.3\left(\frac{M_1}{M_2}\right)^2 < 2.3\\
&C_{\mathrm{c}} = \frac{2\pi^2 E_{\mathrm{s}}}{F_{\mathrm{y}}}\\
&f_{\mathrm{rb}} = \frac{3}{4}\frac{V_{\mathrm{h}}\tan\theta}{t_{\mathrm{w}}\theta^2 e} < \varphi_{\mathrm{b}} F_{\mathrm{rb}} = \left[1 - \frac{\left(\frac{L_{\mathrm{b}}}{r_{\mathrm{T}}}\right)^2}{2{C_{\mathrm{c}}}^2 C_{\mathrm{b}}}\right]\varphi_{\mathrm{b}} F_{\mathrm{y}}
\end{aligned} \tag{9.10}$$

where θ, e, and d_h are the cutting angle, hole pure distance, and cutting depth of castellated beam, respectively (Fig. 9.1), t_w is the thickness of the web, M_1 and M_2 are the moments at each beam end, E_s is the modulus of elasticity of the steel, and φ_b is equal to 0.9 similar to the moment equation.

9.2.2.3 Deflection Criteria

Beam deflection can be calculated by means of the standard equations of structural analysis. For interior and edge beams, bending deflection ($\mathrm{def_b}$) can be calculated as

$$\mathrm{def_b} = \frac{5W_{d1}{L_\mathrm{T}}^4}{384E_\mathrm{s}I_\mathrm{n}} + \frac{5W_{d2}{L_\mathrm{T}}^4}{384E_\mathrm{s}I_\mathrm{def}} \tag{9.11}$$

where W_{d1} and W_{d2} are the pre-composite and post-composite loads, respectively, L_T is the total beam length, and I_def and I_n are the effective moment of inertia for deflection of composite beam and steel net section moment of inertia, respectively.

Concrete weight must be resisted by steel section only (pre-composite level), and other dead and live loads must be sustained by composite section (post-composite level).

Deflection of the girders is related to the number floor divisions (beam spacing) and the number of interior beams.

Unlike the standard composite beam, the shear deflection of the composite beam with web opening is significant. Thus researchers have developed experimental-based equation for calculating the shear deflection ($\mathrm{def_s}$) as follows (Benitez et al. [14]):

$$\mathrm{def_s} = \mathrm{def_b} \times \left(1 + \frac{1}{5}\left(\frac{\mathrm{HEW}}{L_\mathrm{T}}\right)\left(\frac{I_\mathrm{com}}{I_\mathrm{com-g}} - 1\right)\left(3 \times \left(\frac{\mathrm{HEW}}{L_\mathrm{T}}\right)^3 - 4 \times \left(\frac{\mathrm{HEW}}{L_\mathrm{T}}\right)^2 - 6 \times \left(\frac{\mathrm{HEW}}{L_\mathrm{T}}\right) + 12\right)\right) \tag{9.12}$$

where I_com and $I_\mathrm{com-g}$ are the net and gross composite section moments of inertia, respectively. This equation is based on rectangular shape holes, and the hexagonal shapes must be considered as rectangular shapes with effective width:

$$\mathrm{HEW} = e + d_\mathrm{h} \times \cot(\alpha) \tag{9.13}$$

and $\mathrm{def_s}$ identifies the effect of one hole. For web opening with a width to height ratio lower than 2, maximum deflection of the beam is independent of the location of the holes. Thus, the total shear deflection can be obtained from the number of unfilled holes (N_uh) times the $\mathrm{def_s}$, and the total beam deflection is calculated as follows:

$$\mathrm{def} = \mathrm{def_b} + \mathrm{def_s} \times N_{\mathrm{uh}}$$

Also for considering the effect of differential shrinkage and creep on a composite steel–concrete structure, the effective width (or concrete modulus of elasticity) can be divided by 3 (Roll [15]).

Also, the allowable deflection ($\mathrm{def_{all}}$) under the live and dead loads is specified by AISC [12] as

$$\mathrm{def} < \mathrm{def_{all}} = \frac{L_{\mathrm{T}}}{240} \tag{9.14}$$

9.2.2.4 Vibration Criteria

A portion of the live load (between 10 and 25 %) that is used for calculating deflection is utilized for calculating vibration ($\mathrm{def_{vib}}$) (Murray et al. [16]). Combining the effect of the interior beam deflection ($\mathrm{def_{int}}$), the girder beam deflection ($\mathrm{def_{gir}}$), and column deflection ($\mathrm{def_{col}}$) for calculating frequency is considered as follows (Naeim [17]):

$$\mathrm{def_{vib}} = \frac{\mathrm{def_{int}} + \mathrm{def_{gir}}}{1.3} + \mathrm{def_{col}} \tag{9.15}$$

In order to take into account the difference between the frequency of a simply supported beam with distributed mass and concentrated mass at mid-span, the deflection is divided by 1.3 $\left(\frac{4}{\pi}\right)$ (Murray et al. [16]).

Because of the small compression deflection of the column, $\mathrm{def_{col}}$ is considered as zero. Also, 0.2 times of the live load is used in calculating the deflection.

For considering greater stiffness of concrete on the metal deck under dynamic loading compared to the static loading, it is assumed that the modulus of elasticity for the concrete is 1.35 times that of the normal concrete. The effect of differential shrinkage and creep on a composite steel–concrete structure is not considered for vibration calculations.

$$f = \frac{1}{2\pi}\sqrt{\frac{\text{Stiffness}}{\text{Mass}}} = \frac{1}{2\pi}\sqrt{\frac{\frac{W}{\mathrm{def_{vib}}}}{\frac{W}{g}}}\sqrt{\frac{g}{\mathrm{def_{vib}}}} = \frac{1}{2\pi}\sqrt{\frac{g}{\mathrm{def_{vib}}}} \tag{9.16}$$

where W and g are the load and gravity acceleration, respectively.

In order to consider the effect of frequency of all parts of the floor, the total frequency of the floor (f_{t}) is determined by

$$\frac{1}{f_{\mathrm{t}}} = \frac{1}{f_{\mathrm{int}}} + \frac{1}{f_{\mathrm{gir}}} + \frac{1}{f_{\mathrm{col}}} \tag{9.17}$$

where f_{int}, f_{int}, and f_{int} are the interior, girder, and column frequencies, respectively.

Due to the large axial stiffness of the column in comparison to the bending stiffness of beams, column frequency is considered infinity.

The maximum initial amplitude (inch) of the beam (A_{o}) is determined as (Naeim [17])

$$A_{\mathrm{ot}} = (\mathrm{DLF})_{\max} \times \left(\frac{0.6(L_{\mathrm{T}} \times 0.393)^3}{48\left(E_{\mathrm{s}} \times 14.22 \times 10^{-3}\right)\left(I_{\mathrm{def}} \times 0.393^4\right)} \right) \tag{9.18}$$

$$h_{\mathrm{c-eff}} = \frac{\text{Actual Slab Weight}}{\text{Concrete Weight}} \tag{9.19}$$

$$N_{\mathrm{eff}} = 2.97 - 0.0578 \times \left(\frac{S_{\mathrm{b}}}{h_{\mathrm{c-eff}}}\right) + 2.56 \times 10^{-8} \times \left(\frac{{L_{\mathrm{T}}}^4}{I_{\mathrm{def}}}\right) + 0.0001\left(\frac{L_{\mathrm{T}}}{S_{\mathrm{b}}}\right)^3 \tag{9.20}$$

$$A_0 = \frac{A_{\mathrm{ot}}}{N_{\mathrm{eff}}} \tag{9.21}$$

where S_{b} is the beam spacing. $(\mathrm{DLF})_{\max}$ values for various natural frequencies are presented in design practice to prevent floor vibrations (Naeim [17]). Effective concrete height ($h_{\mathrm{c-eff}}$) is not equal to the concrete height in the steel deck floor. Required damping ratio (D_{req}) for specified amplitude and frequency must be lower than the allowable damping ratio (D_{all}), and it is determined as (Naeim [17])

$$D_{\mathrm{req}} = 35A_{\mathrm{o}}f + 2.5 < D_{\mathrm{all}} = 0.035 \tag{9.22}$$

9.2.3 Shear Stud Design

For a desired composite action between steel and concrete, shear studs are required. The shear capacity of these elements must be larger than the maximum shear forces that composite beam will experience. Steel-headed stud anchor is considered in this chapter. Its diameter is considered as 19 mm and 1, 2, or 3 studs can be installed at each rib.

$$Q_{\mathrm{u}} = \min\left(0.85F_{\mathrm{c}}b_{\mathrm{e}}h_{\mathrm{c}}, A_{\mathrm{s}}F_{\mathrm{y}}\right) < N_{\mathrm{c}}\varphi_{\mathrm{v}}Q_{\mathrm{n}} = N_{\mathrm{c}} \times \varphi_{\mathrm{v}} \times 0.5A_{\mathrm{sa}}\sqrt{F_{\mathrm{c}}E_{\mathrm{c}}} \\ \leq R_{\mathrm{g}}R_{\mathrm{p}}A_{\mathrm{sa}}F_{\mathrm{u-ss}} \tag{9.23}$$

where F_c and E_c are the compression strength and modulus of elasticity of concrete, respectively; b_e and h_c are the effective width and height of concrete, respectively; A_s and A_{sa} are the steel section area and steel-headed shear stud area, respectively; F_{u-ss} is the ultimate stress of shear stud; and R_g and R_p are the group and position effect factor for shear stud, respectively. Considering linear shear diagram, N_c is half of the total number of shear stud and φ_v is equal to 0.75 (AISC [12]).

9.3 Problem Definition

9.3.1 Cost Function

The cost for each beam is considered as the sum of the profile steel beam cost, welding procedure cost, cutting procedure cost, and shear stud cost. The cost for steel deck is the sum of the steel deck concrete cost, steel deck steel plate cost, and steel deck application cost. Initial cost is the sum of the beam costs and steel deck cost.

Each sub-cost is determined by multiplying the corresponding weight, length, volume, or area by appropriate coefficients. Cost of filling end holes by plates is considered by the cost of the added weights, cutting, and welding to the total cost.

9.3.2 Variables

In this chapter, five variables are used for optimal design of each beam, consisting of the profile section, cutting depth (d_h), cutting angle (α), hole spacings (s), and number of filled end holes of the castellated beams. The number of beams at floor width and concrete thickness are two other variables that are changed. The minimum and maximum magnitudes of the variables must be known for avoiding unacceptable results and for fast convergence to the global optimum. Profile section is the sequence number of the hot rolled steel profiles. Cutting angle is limited between 40° and 64°. Other limits on the variables are presented as the constraints.

9.3.3 Constraints

Castellated beam application constraints (g_1 to g_5) and steel beam design constraints (g_6 to g_{14}) are considered as follows:

$$g_1 = d_\mathrm{h} - \frac{3}{8}(H_\mathrm{s} - 2t_\mathrm{f}) \tag{9.24}$$

$$g_2 = (H_\mathrm{s} - 2t_\mathrm{f}) - 10(d_\mathrm{t} - t_\mathrm{f}) \tag{9.25}$$

$$g_3 = \frac{2}{3}d_\mathrm{h}\cot(\alpha) - e \tag{9.26}$$

$$g_4 = e - 2d_\mathrm{h}\cot(\alpha) \tag{9.27}$$

$$g_5 = 2d_\mathrm{h}\cot(\alpha) + e - 2d_\mathrm{h} \tag{9.28}$$

$$g_6 = M_\mathrm{u} - \varphi_\mathrm{b}M_\mathrm{n} \tag{9.29}$$

$$g_7 = \frac{M_\mathrm{u}}{Z_\text{net-com-bot}} + \frac{m_\mathrm{u}}{Z_\text{tee-com}} - \varphi_\mathrm{b}F_\mathrm{y} \tag{9.30}$$

$$g_8 = V_\mathrm{u} - \varphi_\mathrm{v}V_{\mathrm{n-w}} \tag{9.31}$$

$$g_9 = \frac{V_\mathrm{u}}{2} - \varphi_\mathrm{v}V_{\mathrm{n-tee}} \tag{9.32}$$

$$g_{10} = V_\mathrm{h} - \varphi_\mathrm{v}V_{\mathrm{n-p}} \tag{9.33}$$

$$g_{11} = f_\mathrm{rb} - \varphi_\mathrm{b}F_\mathrm{rb} \tag{9.34}$$

$$g_{12} = \mathrm{def} - \mathrm{def}_\mathrm{all} \tag{9.35}$$

$$g_{13} = D_\mathrm{req} - D_\mathrm{all} \tag{9.36}$$

Some design constraints for the steel decks are as follows:

$$g_{14} = B - L_{\mathrm{sd-max}} \tag{9.37}$$

where $L_{\mathrm{sd-max}}$ is the maximum length of the unshored steel deck.

For comparison and for comparing the sum of constraints with each other, these are normalized.

9.3.4 Penalty Function

Optimization algorithms are designed for unconstraint problems, and an external procedure should be defined for avoiding unacceptable regions. Penalty functions increase the objective function cost, and the optimization algorithm automatically avoids infeasible areas. In this study, penalty function is expressed as the function of positive (unacceptable) values of the constraint functions:

$$\mathrm{NAC} = \mathrm{sum}(g_i > 0) \tag{9.38}$$

$$\mathrm{PF} = 10^{\mathrm{NAC}} \tag{9.39}$$

$$\mathrm{Cost}_{\mathrm{fin}} = \mathrm{Cost}_{\mathrm{ini}} \times \mathrm{PF} \tag{9.40}$$

where $\mathrm{Cost}_{\mathrm{fin}}$ and $\mathrm{Cost}_{\mathrm{ini}}$ are the final and initial costs, respectively. The value of 10 is chosen by the experience for the current problem and it can be changed for other problems.

9.4 Optimization Algorithm

Interior beam optimization, edge beam optimization, girder optimization, and deck optimization are four suboptimizations of this problem. Each of the first three problems has five variables according to the explanation given in the previous section. Deck optimization has one variable and the number of floor division is another variable. Thus, there are 17 optimization variables for this problem. Optimizing these variables simultaneously decreases the convergence rate. In order to solve this problem, and to observe the conditions around the optimum result, the following approach is adopted.

9.4.1 Suboptimization Approach

If the deck is optimized without considering beam optimization, finding the best result is simple. But other decks with higher costs can increase composite action of the beams and decrease the beam cost, hence reducing the total cost. Thus, after finding the best deck independently (by sorting deck choices from lower to highest cost and selecting the first acceptable choice), some other near acceptable results are considered, and optimum result of other parts of the floor is calculated for the entire system.

The range for the number of divisions of the floor is limited for different examples. To observe the impact of increasing the number of division, different values are considered and the results of optimization are obtained.

In order to optimize each beam, the following metaheuristic algorithm is used:

9.4.2 Metaheuristic Optimization Algorithm

Metaheuristic algorithms try to find the best solution to a problem in an iterative manner. They have an initial population and evaluate the objective function values of them. The algorithm produces the next generation from the initial population in order to increase the chance of find the best result. So increasing the number of population and iteration number can increase the chance of finding the optimum result.

Colliding bodies optimization is one of the recently developed metaheuristic algorithms. The efficiency of this algorithm for structural optimization is validated by researchers (Kaveh and Mahdavi [18]). The CBO is simple in concept and depends on no internal parameter.

In this technique, one object collides with other objects and they move toward a minimum energy level. Each colliding body (CB) has a specified mass (m_k) related to the fitness function as

$$m_k = \frac{\frac{1}{\mathrm{fit}(k)}}{\sum_{i=1}^{n} \frac{1}{\mathrm{fit}(i)}, \quad k = 1, 2, \ldots, n} \tag{9.41}$$

where fit and n are the fitness function and the number of CBs, respectively. In order to select pairs of objects for the collision, CBs are sorted according to the magnitudes of their mass in a decreasing order, and are divided into two equal groups: a stationary group and a moving group. Moving objects collide with stationary objects to improve their positions and push stationary objects toward better positions by changing their velocity. Initial velocity of the moving objects (v_1) is defined as a distance between their positions and destination of the stationary object. Initial velocity of stationary objects is considered as zero. Next, velocity of stationary (v_{sta}) and moving (v_{mov}) groups is calculated as follows:

$$v_{\mathrm{mov}} = \frac{(m_1 - \varepsilon m_2)v_1}{m_1 + m_2} \tag{9.42}$$

$$v_{\mathrm{sta}} = \frac{(m_1 + \varepsilon m_2)v_1}{m_1 + m_2} \tag{9.43}$$

where m_1, m_2, v_1, and v_2 are the mass and velocity of each pair of moving and stationary objects. Also, ε is defined as follows:

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{9.44}$$

where $iter$ and $iter_{max}$ are the current iteration number and maximum iteration number, respectively. Next, position of each CB is its last position plus a random ratio of velocity.

In order to improve the CBO to get faster and more reliable solutions, enhanced colliding bodies optimization (ECBO) has been developed which uses a memory to save a number of historically best CBs and also utilizes a mechanism to escape from local optima (Kaveh and Ilchi Ghazaan [19]). Utilizing this improvement requires to identify the colliding memory size (CMS) and the random parameter (RP).

Flowchart of the analysis and optimization of floor system is shown in Fig. 9.2.

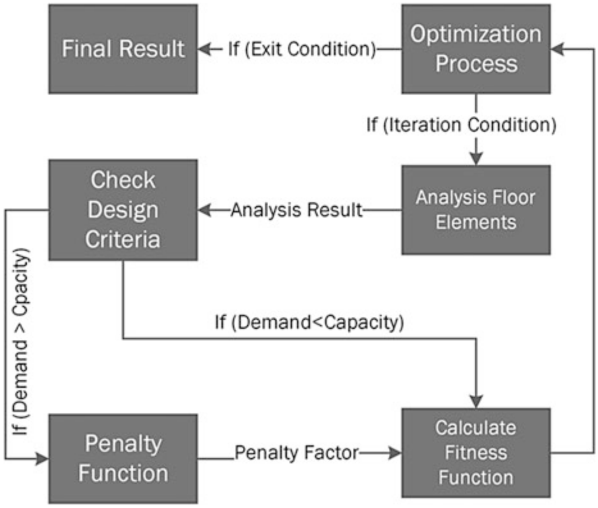


Fig. 9.2 Flowchart of the process of optimum design of steel floor system

9.5 Numerical Examples

In order to study the effect of parameters on the optimum cost of the floor, two examples are studied. MATLAB software is used for modeling the optimization process. This software is also used for the analysis and checking design criteria. The design results are also double-checked with ETABS software.

In both examples, floor systems with two girders, two edge beams, and some interior beams are considered as shown in Fig. 9.3, and all connections are assumed pinned connections.

For algorithm adjustments, the population size and the iteration number are 40 and 60, respectively. Also, CMS and RP are considered to be 4 and 0.3, respectively.

9.5.1 Example 1: Floor System (Span 10 m and Width 8 m)

At the first example, the span and width of the floor system are 10 m and 8 m, respectively. Interior beams are affected by live and dead area loads. Edge and girder beams are affected by live and dead uniformly distributed loads (in order to take the influence of adjacent bay and wall load into account). Girder beam is also affected by end reaction of interior beam as a point load.

Full composite action is considered, since partially composite action is very sensitive to construction and installation conditions of shear studs and it has a large amount of uncertainty.

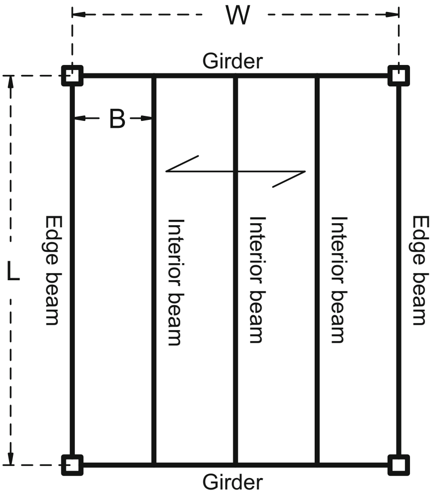


Fig. 9.3 Floor system configuration for the floor division number is equal to 4

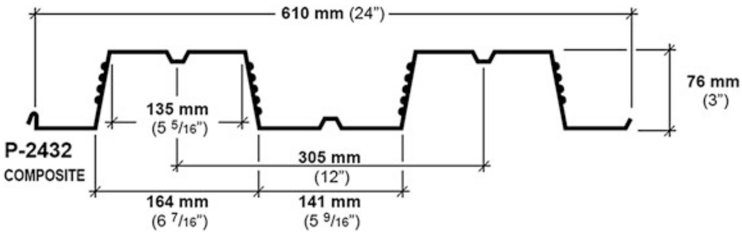


Fig. 9.4 Details of a steel deck from Canam® steel catalogue

In order to have a comparison with other reference examples (Poitras et al. [8]), the steel deck choices were taken from the Canam® steel catalogue as presented in Fig. 9.4. According to P-2434 (composite type of this catalogue), deck thickness values are considered as 0.76, 0.91, and 1.21 mm. Slab thickness values are taken as 125, 140, 150, 165, 190, and 200 mm. Maximum span for each combination of deck and steel thickness is determined and the load resistance for each span is calculated. It is assumed that each span has adjacent span in the start and end (triple span condition). Shoring decks are not considered.

The profile sections are chosen by the *Canadian Handbook of the Steel Construction*, starting from W410 × 39 and ending with W690 × 289. The steel yielding stress, steel modulus of elasticity, and concrete compression capacity are 3550 kg/cm^2, 2,050,000 kg/cm^2, and 200 kg/cm^2, respectively.

The values of the cost coefficients are determined by other researchers (Poitras et al. [8]) and engineering experiences. Cost coefficients are given in Table 9.1.

Table 9.1 Cost coefficients

Component	Price ($)	Unit
Steel profile	2.86	$ per each kg
Welding beams	1	$ per each m
Cutting beams	0.8	$ per each m
Shear studs	2.4	$ per each kg
Concrete	131	$ per each m^3
Steel deck	2.25	$ per each kg
Application	10.8	$ per m^2

Table 9.2 Problem-type description and costs (Example 1)

Type	Description	Cost ($)	%
1	Poitras et al. [8] best results	14,832	0.96
2	Checking Poitras et al. [8] results	15,523	1.00
3	Optimizing composite beams	14,097	0.91
4	Optimizing composite castellated beams	12,796	0.82

Table 9.3 Critical constraints (Example 1)[a]

Type	Girders		Edge beams			Interior beams			
1	–	–	–	–	–	–	–	–	–
2	FM	CC	VB			FM			
3	FM	CC	FM	De	VB	FM	De	VB	
4	FM	BU	HS	De	VB	HS	FM	De	VB

[a] *HS* horizontal shear, *RM* radial moment, *DE* deflection, *FM* flexural moment, *VB* vibration, *BU* buckling web, *CC* concrete compression

Poitras et al. [8] did not consider the effect of shrinkage and temperature as discussed below. In order to compare the results of this study with their results, shrinkage and temperature effects are not considered in Example 1. They also used the S16 standard requirements (CSA [20]). Penalty factors in their work were considered constant and this assumption decreased the convergence rate.

For comparing results with other researchers and presenting effect of castellated beams, four problem types are assumed and they are defined in Table 9.2. Also final costs of each type are presented in this table.

Critical constraints (over 80 % demand capacity ratio) are shown in Table 9.3. Also, detailed results include the section profile of each beam as presented in Table 9.4.

The results of Example 1 are shown for comparison, and 4 % difference is observed between the results of Poitras et al. [8] and the checked values. It should be mentioned that they considered 75 % for composite action and our study considers full composite action. Thus, the number of shear studs is lower than our study.

Table 9.4 Results (Example 1)

Type	Girders	Edge beams	Interior beams		Concrete floor	
	Section	Section	Section	Number	Steel thickness (mm)	Depth (mm)
1	W530 × 82	W460 × 60	W460 × 60	2	0.76	140.00
2	W530 × 82	W460 × 60	W460 × 60	2	0.76	140.00
3	W460 × 60	W410 × 39	W410 × 46	3	0.91	125.00
4	W460 × 52	W410 × 39	W410 × 39	3	0.76	125.00

Table 9.5 Effect of the floor division number and deck section on the total cost (Example 1)

Floor division number	Deck price					
	Composite beams			Composite castellated beams		
	Low	Medium	High	Low	Medium	High
2	20,197	17,851	17,207	19,762	16,350	17,484
3	14,892	14,170	14,323	14,422	13,969	14,446
4	14,399	14,097	14,639	12,796	13,312	13,842
5	14,208	14,274	14,444	13,781	15,440	14,057

By changing floor division numbers and deck sections, a parametric study is performed for composite and composite castellated beams and it is presented in Table 9.5.

9.5.2 Example 2: Floor System (Span 6 m and Width 7 m)

This example is similar to Example 1. Span and width are 6 m and 7 m, respectively. The profile sections are chosen from the IPE steel sections, starting from IPE140 and ending with IPE600. The steel yielding stress, steel modulus of elasticity, and concrete compression capacity are 2400 kg/cm^2, 2,039,000 kg/cm^2, and 250 kg/cm^2, respectively. The effects of shrinkage and temperature are considered. There is no uniform distributed load on edge beams and girders. In order to simulate adjacent bay conditions, they also resist two times of typical load of the exiting bay. Because the same loading was used on the interior and edge beams, their results are presented together. Other parameters of Example 2 are similar to those of Example 1.

Critical constraints (over 80 %), detailed results, and costs of the choices are shown in Table 9.6, Table 9.7, and Table 9.8, respectively. Also, hole spacing for cutting depths is extracted from detailed results for beams, and the average of these ratios is calculated and presented in Table 9.9.

Table 9.6 Critical constraints (Example 2)[a]

Floor division number	Deck price	Girders			Interior and edge beams			
3	Low	HS	FM	RM	HS	FM	RM	VS
	Medium	HS			HS	FM	RM	VS
	High	HS	FM	RM	HS	FM	RM	VS
4	Low	HS	FM	RM	HS	FM	RM	
	Medium	HS			HS	FM	VS	DE
	High	FM			HS	FM	RM	
5	Low	HS	RM		HS	RM	FM	VS
	Medium	HS	FM	RM	HS	FM	VS	DE
	High	HS	FM	VS	HS	FM	VS	

[a]*HS* horizontal shear, *RM* radial moment, *DE* deflection, *FM* flexural moment, *VS* vertical shear

9.6 Concluding Remarks

Optimization and parametric studies of steel floor systems with composite and castellated beams and steel decks are performed in this study. The objective function is the floor cost where 17 variables and parameters are considered. The stress, stability, deflection, and vibration criteria are all discussed. Results indicate that:

1. Using the high-price decks in order to amplify the composite action can improve the results and decrease the cost between 5 and 10 % in composite beams and composite castellated beams. It seems that choosing the most expensive deck does not guarantee the best result. So considering the first three acceptable decks is a good assumption.
2. Considering different number of divisions can decrease the total cost between 10 and 20 %.
3. Using composite castellated beams improves the results by about 14 % compared to the composite beams.
4. The optimum degree of castellated cutting angle is about 63°.
5. Average ratio of hole spacing to cutting depth is between 2 and 3. This ratio is 3 for commercial castellated beams

The results show that the utilized optimization algorithm, ECBO, performs quite well, and it has reliable and accurate solution. The fast-converging feature of the standard CBO is generally preserved in ECBO, whereas the modifications of the latter algorithm improve the exploration capabilities of the CBO. One can conclude that ECBO algorithm is competitive with the other available optimization methods. For an extensive comparative study of ECBO, when applied to different structural optimization problems, one can refer to Kaveh [21].

Table 9.7 Results (Example 2)

Floor division number	Deck price	Girder					Edge and interior beam					Deck result	
		Section	Cut depth (cm)	Cut angle (d)	Hole spacing (cm)	Filled hole	Section	Cut depth (cm)	Cut angle (d)	Hole spacing (cm)	Filled hole	Steel thickness (cm)	Concrete height (cm)
3	Low	IPE500	13.93	62.48	37.19	0	IPE240	7.95	63.8	23.07	0	12.5	0.91
	Medium	IPE500	13.93	62.48	37.19	0	IPE240	7.95	63.8	23.07	0	12.5	0.91
	High	IPE450	9.93	63.67	29.49	0	IPE240	5.88	63.7	17.04	0	14	0.91
4	Low	IPE360	10.96	63.98	31.05	0	IPE220	8.07	63.8	17.69	0	12.5	0.76
	Medium	IPE400	8.16	62.69	21.79	0	IPE240	11.7	63.2	23.96	0	14	0.76
	High	IPE300	17.46	63.78	50.54	3	IPE220	6.82	63.5	15.79	0	12.5	0.91
5	Low	IPE330	9.81	63.80	26.81	0	IPE200	8.66	59.9	19.38	0	12.5	0.76
	Medium	IPE300	8.62	62.88	25.51	0	IPE180	6.37	64	13.66	0	14	0.76
	High	IPE330	12.61	63.16	30.47	0	IPE200	4.32	59.4	11.1	0	12.5	0.91

Table 9.8 Effect of floor division number and deck section on the total cost (Example 2)

Floor division number	Total cost ($)		
	Low	Medium	High
3	7514.7	8532.2	7711.6
4	6871.3	7275.5	6732.8
5	6810.4	6207.9	6795.3

Table 9.9 Average ratio of hole spacing to cutting depth (Example 2)

Section	Average s/d_h
IPE180	2.14
IPE200	2.40
IPE220	2.25
IPE240	2.69
IPE300	2.93
IPE330	2.57
IPE360	2.83
IPE400	2.67
IPE450	2.97
IPE500	2.67

References

1. Kaveh A, Ghafari MH (2016) Optimum design of steel floor system: effect of floor division number, deck thickness and castellated beams. Struct Eng Mech 59(5):933–950
2. Morton S, Webber J (1994) Optimal design of a composite I-beam. Compos Struct 28 (2):149–168
3. Klanšek U, Kravanja S (2007) Cost optimization of composite I beam floor system. Am J Appl Sci 5(1):7–17
4. Senouci AB, Al-Ansari MS (2009) Cost optimization of composite beams using genetic algorithms. Adv Eng Softw 40(11):1112–1118
5. Adeli H, Kim H (2001) Cost optimization of composite floors using neural dynamics model. Commun Numer Methods Eng 17(11):771–787
6. Platt BS, Mtenga PV (2007) Parametric optimization of steel floor system cost using evolver. WIT Trans Built Environ 91:119–128
7. Kaveh A, Abadi ASM (2010) Cost optimization of a composite floor system using an improved harmony search algorithm. J Constr Steel Res 66(5):664–669
8. Poitras G, Lefrançois G, Cormier G (2011) Optimization of steel floor systems using particle swarm optimization. J Constr Steel Res 67(8):1225–1231
9. Kaveh A, Ahangaran M (2012) Discrete cost optimization of composite floor system using social harmony search model. Appl Soft Comput 12(1):372–381
10. Kaveh A, Massoudi M (2012) Cost optimization of a composite floor system using ant colony system. Iran J Sci Technol Trans Civil Eng 36(C2):139–148
11. ASCE (1994) Minimum design loads for buildings and other structures, vol 7. American Society of Civil Engineers, Chicago, IL
12. AISC (2010) Specification for structural steel buildings (ANSI/AISC 360-10). American Institute of Steel Construction, Chicago, IL
13. Kerdal D, Nethercot D (1984) Failure modes for castellated beams. J Constr Steel Res 4 (4):295–315
14. Benitez MA, Darwin D, Donahey RC (1998) Deflections of composite beams with web openings. J Struct Eng ASCE 124(10):1139–1147

15. Roll F (1971) Effects of differential shrinkage and creep on a composite steel-concrete structure. ACI Spec Publ 27
16. Murray TM, Allen DE, Ungar EE (2003) Floor vibrations due to human activity. American Institute of Steel Construction, Chicago, IL
17. Naeim F (1991) Design practice to prevent floor vibrations. In: Steel Tips, Structural Steel Educational Council, Technical Information and Product Service, Steel Committee of California
18. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
19. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
20. Csa C (2009) CSA-S16-09: design of steel structures. Canadian Standards Association, Mississauga, ON
21. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer, Switzerland

Chapter 10
Optimal Design of the Monopole Structures Using the CBO and ECBO Algorithms

10.1 Introduction

Tubular steel monopole structures are widely used for supporting antennas in telecommunication industries. This chapter utilizes two recently developed metaheuristic algorithms, so-called colliding bodies optimization (CBO) and its enhanced version (ECBO), for size optimization of monopole steel structures. The optimal design procedure aims to obtain minimum weight of monopole structures subjected to the TIA-EIA222F specifications. Two numerical examples are examined to verify the suitability of the design procedure and to demonstrate the effectiveness and robustness of the CBO and ECBO in creating optimal design for this problem. The outcomes of the ECBO are also compared to those of the standard CBO to illustrate the importance of the enhancement of the CBO algorithm [1].

Over the last decade, there has been an increasing use of cellular telephones, including new smartphones, for voice and data communication, and wireless Internet access, which has increased the demand for wireless data transmission bandwidth. As a result, there has been a large increase in the number of monopoles installed around populated areas to support antennas. Monopoles have become an important part of our communications infrastructure [2–4]. Therefore, optimal design of the monopole structures can be an interesting and challenging issue in the structural engineering research.

The monopole structures can be categorized based on cross-sectional variations along height into two types: the tapered type and stepped type. In tapered type the cross section is continuously decreasing from bottom to top of monopole, and in stepped type the structure is divided into some parts with abrupt changes between sections [2]. The sections of stepped monopoles can be circular and polygonal in shape [5]. Figure 10.1 shows the schematic shape of a treble-part-monopole with circular sections. The main objective of this chapter is to find the optimum size of sections of the steel circular stepped monopoles. Here, the CBO and ECBO algorithms are utilized for optimization, where the weight of the monopole is

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_10

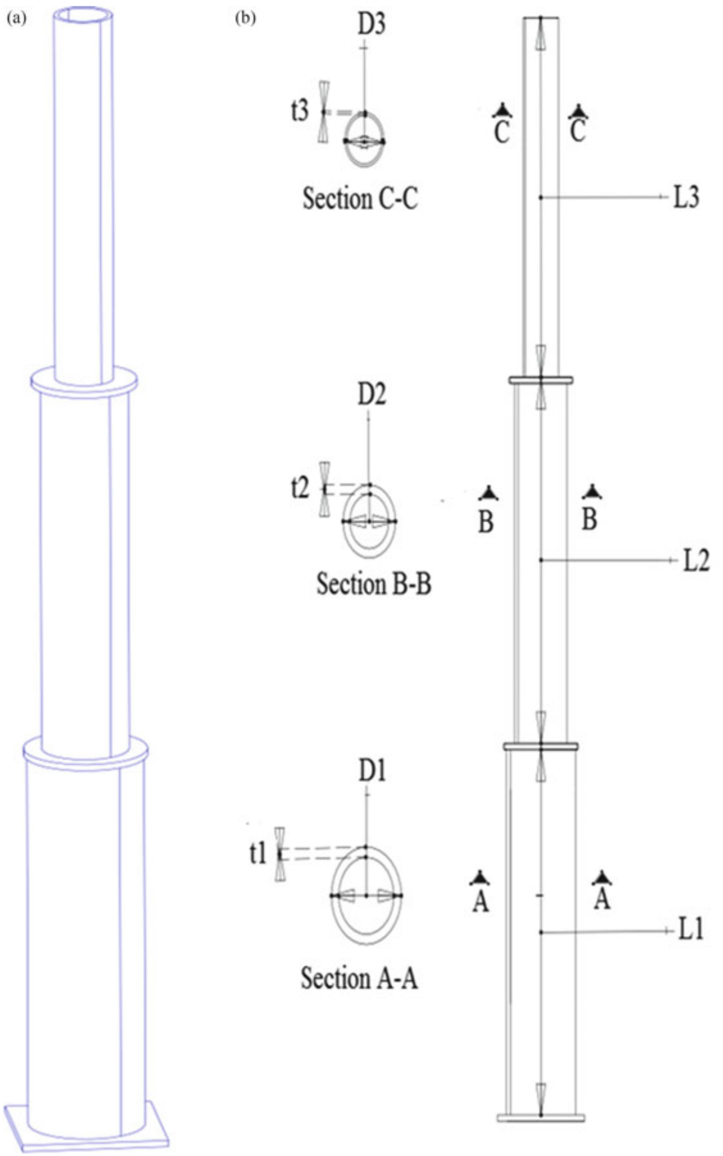


Fig. 10.1 The circular treble-part-monopole: (**a**) Three-dimensional view, (**b**) front view

considered as the objective function. The design method used in this chapter is also consistent with TIA-EIA222F specifications [6].

Optimization algorithms can be divided into two categories: (1) local optimizers and (2) global optimizers. Local optimizer algorithms which often utilize the gradient information or iterative methods to search the solution space near an initial starting point by local changes, are hard to apply and time-consuming in these optimization problems. Therefore, global optimizers such as metaheuristic algorithms are proposed for solving difficult optimization problems by performing global search [7, 8]. In recent years, many metaheuristics have been developed based on or inspired by natural phenomena from a variety of scientific fields (see, e.g., [9–12]). CBO belongs to a family of metaheuristic algorithms which are recently developed by the author and colleagues [8, 13]. This algorithm can be considered as a multi-agent method, where each agent is a colliding body (CB). Simple formulation and no internal parameter tuning are advantages of this algorithm. The ECBO was introduced by Kaveh and Ilchi Ghazaan [14], and it uses memory to save some historically best solutions to improve the CBO performance without increasing the computational cost. ECBO also changes some components of agents randomly to help them leave the local minima.

In this chapter, two design examples are considered to be optimized by CBO and ECBO algorithms. Comparison of the optimal solutions of the ECBO algorithm with those of the CBO method demonstrates the capability of CBO in solving the present type of design problems. It is also observed that optimization results obtained by the ECBO algorithm for two design examples have less weight in comparison to the results of the standard CBO algorithm. From the results obtained in this chapter, it can be concluded that the optimum structures obtained by metaheuristic algorithms require smaller amount of steel material.

The remainder of this chapter is organized as follows: In Sect. 10.2, firstly, the mathematical formulations of the structural optimization of monopole structure problems are presented, and a brief explanation of the TIA-EIA222F [6] is provided. In Sect. 10.3, after an explanation of the CBO, the ECBO algorithm is presented. Section 10.4 includes two standard examples. The last sections provide a discussion on the results of the examples and conclude the chapter.

10.2 Monopole Structure Optimization Problem

The optimization problem can formally be stated as follows:

$$\begin{array}{ll} \text{Find} & X = [x_1, x_2, x_3, \ldots, x_n] \\ \text{to minimizes} & \mathrm{Mer}(X) = f(X) \times f_{\text{penalty}}(X) \\ \text{subjected to} & g_i(X) \leq 0, \; i = 1, 2, \ldots, m \\ & x_{i\min} \leq x_i \leq x_{i\max} \end{array} \tag{10.1}$$

where X is the vector of design variables with n unknowns, g_i is the ith constraint from m inequality constraints, Mer(X) is the merit function, $f(X)$ is the cost function, $f_{\text{penalty}}(X)$ is the penalty function which results from the violations of the constraints corresponding to the response of the monopole structures, and also $x_{i\min}$ and $x_{i\max}$ are the lower and upper bounds of the design variable vector, respectively.

Exterior penalty function method is employed to transform the constrained optimization problem into an unconstrained one as follows:

$$f_{\text{penalty}}(X) = 1 + \gamma_{\text{p}} \sum_{i=1}^{m} \max\left(0, g_j(x)\right) \tag{10.2}$$

where γ_{p} is the penalty multiplier.

10.2.1 Design Variables

The most effective parameters for creating the monopole structure geometry are shown in Fig. 10.1. These parameters can be adopted as design variables:

$$X = \{ D_1 \quad D_2 \quad \cdots \quad D_n \quad t_1 \quad t_2 \quad \cdots \quad t_n \} \tag{10.3}$$

where X, the vector of design variables, contains $2n$ shape parameters of monopole structures, n is the number of monopole parts, and D_i and t_i are the diameter and thickness of pipe cross section of ith part.

10.2.2 Design Constraints

Design constraints are divided into some groups including the operational, stress, and stability constraints. The operational constraint is the restricted rotation at the top of pole structure that is limited to 1.5°. The stress constraint is considered according to ASICE-LRFD [15] manual. The constraint on the local stability of the cross-section is achieved as follows:

$$\frac{D_i}{t_i} \leq 0.11 \frac{E}{F_{\text{y}}} \Rightarrow \frac{D_i}{t_i} \leq 96.25 \tag{10.4}$$

where E and F_{y} are the modulus of elasticity and minimum yield stress of the material, respectively. Here, it is assumed that the material type is st-37 ($E = 210$ GPa, $F_{\text{y}} = 240$ MPa, and $\rho = 7928.5$ kg/m^3).

10.2.3 Cost Function

The cost function is the weight of the monopole structure, which may be expressed as

$$f(X) = \sum_{i=1}^{n} \rho V_i = \sum_{i=1}^{n} \rho A_i l_i = \sum_{i=1}^{n} \rho (2\pi r_i t_i) l_i \tag{10.5}$$

where ρ is the weight per volume of monopole material and V_i, A_i, and l_i are the volume, cross-sectional area, and length of ith part of monopole structure, respectively.

10.2.4 The Applied Loads

In this study, TIA-EIA222F [6] specifications are used for considering the wind and ice loading and their influence on structures. The applied loads on the monopole structures consist of the vertical and horizontal loads, which are described in the following subsections.

10.2.4.1 The Vertical Loads

The most effective vertical loads, which should be considered in analysis process, consist of the self-weight of structure, the weight of ice, and the weight of appurtenance (i.e., dish, light rod, and cable). For considering the load of ice weight, it is assumed that the type of ice is solid and its density (ρ_{ice}) is equal to 897.043 kg/m^3 and thickness of attached ice on structure (t_{ice}) is 0.0127 m (0.5 in). Thus, the weight of ice on unit length of ith part of pole structure (W_i^{ice}) is calculated as

$$W_i^{\text{ice}} = \rho_{\text{ice}} S_i t_{\text{ice}} = 897.043 * (\pi D_i) * 0.0127 = 35.790 D_i \tag{10.6}$$

where S_i and D_i are the circumference and diameter of cross section of the ith part. The (W_i^{ice}) load is a uniform load which is vertically assigned to the ith part.

In the load case of attached appurtenance weight at the top of pole structure, the weight of feedle cable of monopole is assumed as 2721.6 kg. The weight of dish and light rod with and without ice weight are also assumed as in Table 10.1. It should be noted that these concentrated loads are assigned to the top point of the pole structure.

10.2.4.2 The Horizontal Loads

The wind load is considered as lateral load applied to the pole structure. The applied distributed wind load to unit length of the ith part (ω_i^{wind}) is calculated as

Table 10.1 Weight of the appurtenance loading with and without the influence of ice

Description	Weight (kg)
The light rod	16
The dish	1235
Sum of the weights	1251
Sum of the weights with considering the ice	1625

$$\omega_i^{\text{wind}} = F_i Z_i \tag{10.7}$$

where Z_i is the elevation of the center of the ith part and F_i is related to the coefficient of wind force of the ith part which is calculated as

$$F_i = G_{\text{h}} Qz_i Ae_i CF \tag{10.8}$$

where G_{h} is the gust response factor for the fastest mile basic wind speed and it is assumed as 1.69 for pole structures. The structure force coefficient CF is determined as 0.59 based on Table 1 of TIA/EIA-222-F. Qz is the velocity pressure and determined as

$$Qz_i = 0.613 Kz_i V^2 \tag{10.9}$$

where V is the basic wind speed of the location of the structure that is assumed as 36.1 m/s (130 km/h) and Kz is the exposure coefficient:

$$Kz_i = (Z_i/10)^{0.285} \geq 1 \tag{10.10}$$

Also, Ae_i is the effective projected area of the ith part cross section in one face:

$$Ae_i = 1.03 Ag_i = 1.03 L_i D_i \tag{10.11}$$

where Ag_i, L_i, and D_i are the projected area, length, and diameter of the ith part.

Moreover, the ice effect is ignored in above equation. If we consider the ice thickness (i.e., 0.0254 m or 1 in. on the diameter of pole structure, Ae_i is modified as

$$Ae_i^{\text{ice}} = 1.03 Ag_i^{\text{ice}} = 1.03 L_i (D_i + 0.0254) \tag{10.12}$$

The wind load applied to the appurtenance at the top of the pole structure is similarly calculated. In this case, the coefficient of wind force (F) is calculated as

$$F = G_{\text{h}} Qz Aa Ca \tag{10.13}$$

where Aa and Ca are the projected area and force coefficients of appurtenance, respectively. The appurtenance force coefficient (Ca) is assumed as 1.20 based on Table 3 of TIA/EIA-222-F. The Aa is assumed as 1.45 and 1.50 m^2 with and without the effect of ice thickness on the appurtenance, respectively.

10.2.5 Loading Combinations

In this chapter, two loading combinations have been considered based on existence of the ice load effect. Then, two loading combinations are defined:

The load combination 1 (without consideration of the ice load effect): dead load (consisting of the self-weight of structure and weight of the appurtenance) + wind load (consisting of the applied wind load to the face of the pole structure and appurtenance without the ice thickness)

The load combination 2 (with consideration of the ice load effect): dead load (consisting of the self-weight of structure, weight of the appurtenance, and ice thickness) + wind load (consisting of the applied wind load to the face of pole structure and appurtenance with consideration of the ice thickness)

10.3 Enhanced Colliding Bodies Optimization Algorithm

Optimization of monopole structures is a complex problem because of a large search space, multiple local optima, and corresponding constraints. In this chapter we apply a simple and efficient metaheuristic algorithm, the so-called enhanced colliding bodies optimization (ECBO), to solve this problem. For comparative study and showing the complexity of the problem, the standard CBO is also utilized. In the following, both standard CBO and ECBO algorithms are briefly introduced.

10.3.1 Colliding Bodies Optimization Algorithm

The CBO is based on momentum and energy conservation law for one-dimensional collision [13]. This algorithm contains a number of colliding bodies (CBs) where each one is treated as an object with specified mass and velocity which collides with others. After collision, each CB moves to a new position with new velocity with respect to old velocities, masses, and coefficient of restitution. CBO starts with a set of agents determined with random initialization of a population of individuals in the search space. Then, CBs are sorted in an ascending order based on the values of cost function (see Fig. 10.2a). The sorted CBs are divided equally into two groups. The first group is the stationary group, which consists of good agents for which the velocities before collision are zero. The second group consists of moving agents which move toward the first group. Then, the better and worse CBs, i.e., agents with upper fitness value, of each group collide together to improve the positions of moving CBs and to push stationary CBs toward better positions (see Fig. 10.2b). The change of the colliding bodies positions represent the velocities of the CBs before collision as

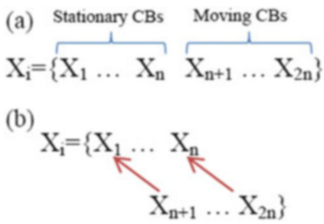


Fig. 10.2 (**a**) The sorted CBs in an increasing order. (**b**) The pairs of objects for the collision

$$v_i = \begin{cases} 0, & i = 1, \ldots, n \\ x_i - x_{i-n}, & i = n+1, \ldots, 2n \end{cases} \tag{10.14}$$

where v_i and x_i are the velocity vector and position vector of the ith CB, respectively, and $2n$ is the population size.

After the collision, the velocity of bodies in each group is evaluated using momentum and energy conservation law and the velocities before collision. The velocity of the CBs after the collision is

$$v_i' = \begin{cases} \dfrac{(m_{i+n} + \varepsilon m_{i+n})v_{i+n}}{m_i + m_{i+n}}, & i = 1, \ldots, n \\ \dfrac{(m_i - \varepsilon m_{i-n})v_i}{m_i + m_{i-n}}, & i = n+1, \ldots, 2n \end{cases} \tag{10.15}$$

where v_i and v_i' are the velocities of the ith CB before and after the collision, respectively, and m_i is the mass of the ith CB defined as

$$m_k = \frac{\frac{1}{fit(k)}}{\sum_{i=1}^{n} \frac{1}{fit(i)}}, \quad k = 1, 2, \ldots, 2n \tag{10.16}$$

where $fit(i)$ represents the objective function value of the ith agent. Obviously, a CB with good values exerts a larger mass and fewer moves than the bad ones. Also, for maximizing the objective function, the term $\frac{1}{fit(i)}$ is replaced by $fit(i)$. ε is the coefficient of restitution (COR) and is defined as the ratio of the separation velocity of the two agents after collision to the approaching velocity of the two agents before collision. In this algorithm, this index is defined to control the exploration and exploitation rates. For this purpose, the COR decreases linearly from unit value to zero. Here, ε is defined as

$$\varepsilon = 1 - (iter/iter_{max}) \tag{10.17}$$

where $iter$ is the actual iteration number and $iter_{max}$ is the maximum number of iterations. Here, COR values equal to unity and zero correspond to the global and local search phases, respectively. In this way a good balance between the global and local search is achieved as the iteration number increases.

The new positions of CBs are evaluated using the generated velocities after the collision in the position of stationary CBs:

$$x_i^{\text{new}} = \begin{cases} x_i + rand \circ v_i', & i = 1, \ldots, n \\ x_{i-n} + rand \circ v_i', & i = n+1, \ldots, 2n \end{cases} \tag{10.18}$$

where x_i^{new} and v_i' are the new position and the velocity after the collision of the ith CB, respectively.

10.3.2 Enhanced Colliding Bodies Algorithm

In order to improve the CBO to obtain faster and more reliable solutions, ECBO was developed which uses a memory to save a number of historically best CBs and also utilizes a mechanism to escape from local optima [14]. The steps of this technique are given as follows:

Level 1: Initialization

Step 1: The initial positions of all the CBs are determined randomly in the search space.

Level 2: Search

Step 2: The value of mass for each CB is evaluated according to Eq. (10.16).

Step 3: Colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values. Solution vectors which are saved in CM are added to the population and the same number of current worst CBs are removed. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: CBs are divided into two equal groups: (i) stationary group and (ii) moving group (Fig. 10.2).

Step 5: The velocities of stationary and moving bodies before collision are evaluated by Eq. (10.14).

Step 6: The velocities of stationary and moving bodies after the collision are evaluated using Eq. (10.15).

Step 7: The new position of each CB is calculated by Eq. (10.18).

Step 8: A parameter like **Pro** within (0, 1) is introduced, which specifies whether a component of each CB must be changed or not. For each colliding body, **Pro** is compared with rn_i ($i = 1, 2, \ldots, n$) which is a random number uniformly distributed within (0, 1). If $rn <$ **Pro**, one dimension of the ith CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + random.\left(x_{j,\max} - x_{j,\min}\right) \tag{10.19}$$

where x_{ij} is the jth variable of the ith CB and $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the jth variable, respectively. In order to protect the structures of CBs, only one dimension is changed.

Level 3: Termination Condition Check

Step 9: After a predefined maximum evaluation number, the optimization process is terminated.

10.4 Design Examples

In this section, two recently developed optimization algorithms consisting of the CBO and ECBO are utilized for optimization of two monopole structures. The number of design variables for the first and the second examples are 10 and 12, respectively. Similarly, the number of colliding bodies (CBs) or agents for these examples is considered as 30. For both examples, the maximum number of iterations is considered as 200. For the sake of simplicity, the penalty approach is used for constraint handling. The optimization algorithms and the analysis and design of monopole structures are coded in MATLAB and SAP200 software, respectively.

10.4.1 A 30 m High Monopole Structure

As the first example, a monopole structure with a height of 30 m is considered. The height of the structure is divided into five equal parts. For this test example, the weight of structure is the objective function. The monopole structure is modeled by ten shape design variables as

$$X = \{D_1 \quad D_2 \quad D_3 \quad D_4 \quad D_5 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5\} \tag{10.20}$$

Design variables can be selected from a discrete list of available values set $D = \{20, 21, 22, \ldots, 89, 90\}$ cm and $t = \{0.4, 0.45, 0.5, 0.6, 0.8, 0.9, 1\}$ cm, which have 78 discrete values.

Table 10.2 compares the results obtained by both algorithms with engineering design values, for which the appropriate values are determined by the author using trial–error method [16]. The constraint values are also shown in Table 10.2; it can be seen that all constraints of the results of both algorithms are satisfied. Moreover,

Table 10.2 Optimum design variables (cm) for the 30 m high monopole using different methods

Design variables	Engineering design	CBO	ECBO
D_5	40	38	38
D_4	47	50	55
D_3	60	57	59
D_2	70	73	69
D_1	80	75	76
t_5	0.45	0.6	0.4
t_4	0.5	0.6	0.6
t_3	0.8	0.6	0.8
t_2	0.8	0.8	0.8
t_1	1	1	0.8
Weight (kg)	3329.4	3253.4	3123.1
Rotation	1.3454	1.3469	1.3499
Maximum stress ratio	0.4194	0.4416	0.4574
Maximum (D/t)	94.00	95.00	95.00

Rotation: rotation at top pole structure (degree)

the evolution process of best fitness values obtained by both algorithms are shown in Fig. 10.3.

10.4.2 A 36 m High Monopole Structure

We now consider a monopole structure with a height of 36 m. The height of the structure is divided into six equal parts. Similarly, for this test example, the weight of the structure is the objective function. All assumptions and definitions are the same to the first example. The monopole structure is modeled by 12 shape design variables as

$$X = \{ D_1 \quad D_2 \quad D_3 \quad D_4 \quad D_5 \quad D_6 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \} \tag{10.21}$$

Table 10.3 compares the results obtained by both algorithms with engineering design values. All of the constraints for the designs obtained by both algorithms are satisfied as the first example. Moreover, the evolution process of best fitness values obtained by both algorithms are shown in Fig. 10.4.

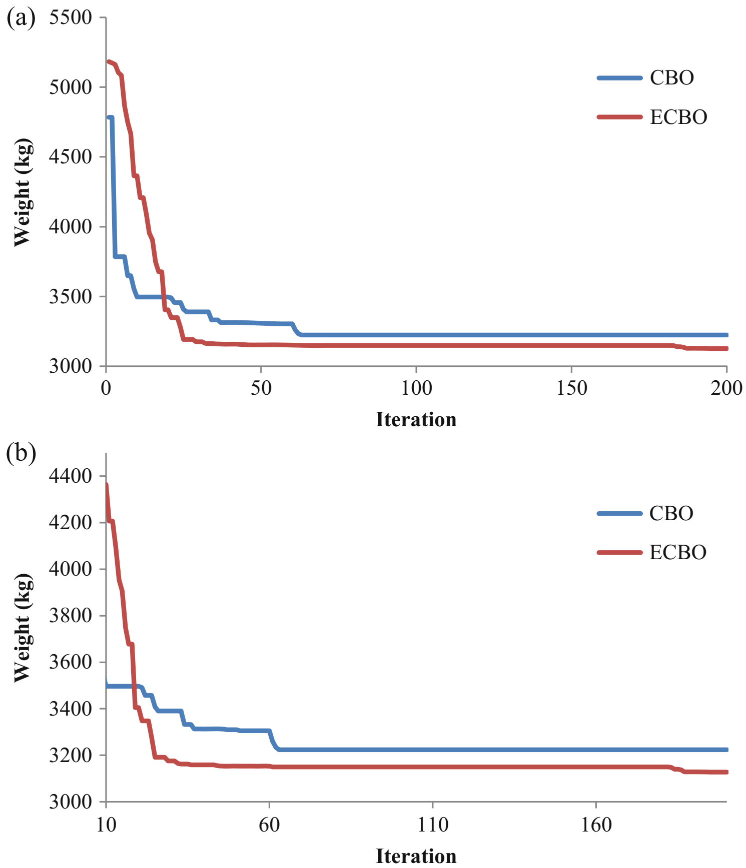


Fig. 10.3 Comparison of the convergence rates between the two algorithms for the first example. (**a**) All iterations, (**b**) 10–200 iterations [1]

10.5 Discussion on the Results of the Examples

In this section, the results obtained in the examples will be discussed. Firstly, it should be noted that optimization of monopole structures is a non-convex and nonlinear optimization problem, because the stiffness and applied loads [consisting of the self-weight, ice, and wind load as described in Eqs. (10.6–10.13)] simultaneously increase with increasing the cross-sectional diameters of parts.

Tables 10.2 and 10.3 compare the results obtained using the CBO and ECBO algorithms with the engineering design ones for both examples, respectively. As discussed before and shown in these tables, the constraints of the final designs of both algorithms are satisfied, and therefore these results could be compared with the engineering design. As anticipated the results obtained using both algorithms are

Table 10.3 Optimum design variables (cm) for the 36 m high monopole using different methods

Design variables	Engineering design	CBO	ECBO
D_6	43	40	39
D_5	57	56	56
D_4	66	65	64
D_3	73	74	74
D_2	75	76	76
D_1	85	86	86
t_6	0.5	0.45	0.45
t_5	0.6	0.60	0.60
t_4	0.8	0.80	0.80
t_3	0.8	0.80	0.80
t_2	0.8	0.80	0.80
t_1	1	1	0.90
Weight (kg)	4608.55	4557.59	4430.80
Rotation	1.4115	1.4449	1.4951
Maximum stress ratio	0.6247	0.6060	0.6041
Maximum (D/t)	95.00	95.00	95.00

Rotation: rotation at top pole structure (degree)

better than the engineering design for both examples. Moreover, the results obtained by the ECBO algorithm are better than those of CBO using the same number of objective function evaluations.

It can be seen from Figs. 10.3 and 10.4, though the CBO algorithm is considerably faster in the early optimization iterations, the ECBO algorithm has converged to a significantly better design in the latter optimization iterations without being trapped in local optima.

10.6 Concluding Remarks

An efficient optimization method is proposed for optimal design of the steel circular stepped monopole structures, based on CBO and ECBO algorithms. The CBO mimics the laws of collision between objects. The simple implementation and parameter independency are definite strength points of CBO. In the ECBO, some strategies have been utilized to promote the exploitation ability of the CBO. In order to find the optimal cross-sectional sizes of monopole structure, the weight of monopole and cross-sectional sizes are respectively defined as objective function and variables in the optimization process. Then, the cross-sectional sizes are selected based on optimization algorithms from available discrete variables.

The validity and efficiency of the proposed method are shown through two test problems. The results of the proposed algorithms are compared to those of the engineering design values. The results indicate that both algorithms could decrease

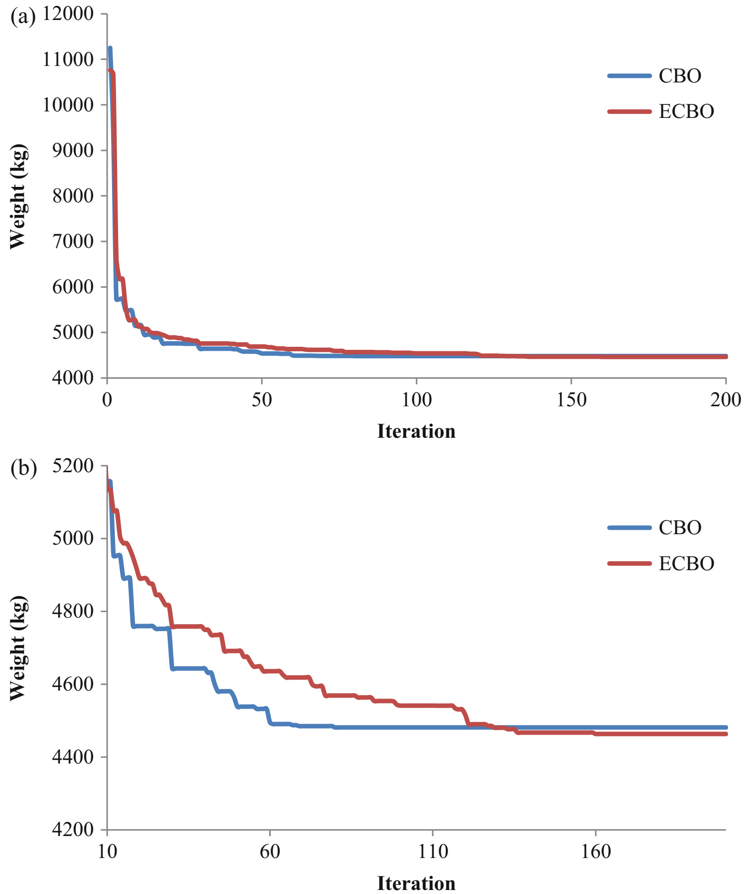


Fig. 10.4 Comparison of the convergence rates between the two algorithms for the second example. (**a**) All iterations, (**b**) 10–200 iterations [1]

the weight of engineering design monopole structures without causing any violations. Moreover, the ECBO algorithm clearly outperforms the CBO algorithm with the same computational time. This indicates the importance of selecting the effective optimization algorithm in this problem. Future researches can investigate problems such as optimization of other types of monopole structures using recently developed metaheuristic optimization algorithms.

References

Kaveh A, Mahdavi VR, Kamalinejad M (2016) Optimal design of the monopole structures using CBO and ECBO algorithms. Periodica Polytech Civil Eng (Published online)

2. Dicleli M (1997) Computer-aided optimum design of steel tubular telescopic pole structures. Comput Struct 62(6):961–973
3. Hawkins DW (2010) Discussion of current issues related to steel telecommunications monopole structures. In: Structures congress 2010 (ASCE)
4. Gonen T (1988) Electric power transmission system engineering-analysis and design. Wiley, New York, NY
5. Hu J (2011) A study on cover plate design and monopole strengthening application. Thin Wall Struct 49:1098–1107
6. TIA Telecommunication Industry Association (1996) TIA/EIA-222-F(TIA-Rev. F): structural standards for steel antenna towers and antenna supporting structures. Arlington, VA
7. Kaveh A (2014) Advance in metaheuristic algorithms for optimal design of structures. Springer, Wien, New York, NY
8. Kaveh A, Mahdavi VR (2015) Colliding bodies optimization; extensions and applications. Springer International Publishing, Switzerland
9. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan
10. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26(1):29–41
11. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289
12. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
13. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
14. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
15. American Institute of Steel Construction (AISC) (2005) Steel construction manual (AISC specification), 13th edn. AISC, Chicago, IL
16. Kamalinejad M (2005) Constructing of cell towers for the IRANCELL operator. PARTO SANAT VIRA company, registration no. 315839, Limited liability company

Chapter 11
Damage Detection in Skeletal Structures Based on CSS Optimization Using Incomplete Modal Data

11.1 Introduction

It is well known that damaged structural members may alter the behavior of the structures considerably. Careful observation of these changes has often been viewed as a means to identify and assess the location and severity of damages in structures. Among the responses of a structure, natural frequencies and natural modes are both relatively easy to obtain and independent from external excitation and, therefore, can be used as a measure of the structural behavior before and after an extreme event which might have led to damage in the structure. This chapter applies charged system search algorithm to the problem of damage detection using vibration data. The objective is to identify the location and extent of multi-damage in a structure. Both natural frequencies and mode shapes are used to form the required objective function. To moderate the effect of noise on measured data, a penalty approach is applied. A variety of numerical examples including beams, frames, and trusses are examined. The results show that the present methodology can reliably identify damage scenarios using noisy measurements and incomplete data [1].

During the past two decades, structural damage identification has gained increasing attention from the scientific and engineering communities, since damage that is not detected and not repaired may lead to catastrophic structural failure. Former methods of damage identification either visual or localized experimental methods require that the vicinity of the damage is known and accessible. Hence, the vibration-based damage identification method as a global damage identification technique is developed to overcome these difficulties. The basic idea of vibration-based damage methods is that modal parameters (notably frequencies, mode shapes, and modal damping) are functions of the physical properties of the structure (mass, damping, and stiffness). Therefore, changes in the physical properties will cause changes in the modal properties [2].

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_11

The usual model-based damage detection methods minimize an objective function, which is defined in terms of the discrepancies between the mathematical model and real structural system. There are two general methods to optimize the objective function, namely, mathematical programming and metaheuristic methods. Unlike the mathematical methods, one of the important characteristics of metaheuristic methods is their effectiveness and robustness in coping with uncertainty, insufficient information, and noise. Many successful applications of damage detection using the metaheuristic algorithms have been reported in the literature. Perera and Torres [3] proposed a method based on mode shapes and frequencies using genetic algorithm on beams. Laier and Morales [4] improved the genetic algorithm to solve damage detection problem for two-dimensional truss-type structures. They used natural frequencies and mode shapes to form objective function. Miguel et al. [5] combined time-domain modal identification technique (SSI) with evolutionary harmony search (HS) algorithm to detect damages under ambient vibration; they studied three cantilever beams under different damage scenarios. Kang et al. [6] proposed an immunity-enhanced particle swarm optimization (IEPSO) for damage detection of structures; they tested this method on a simple beam and a truss. Majumdar et al. [7] presented a method to identify structural damages in truss structures from changes in natural frequencies by using ant colony optimization.

Natural frequencies and mode shapes are the most popular parameters used in the damage identification. These gain their popularity because the modal properties have their physical meanings and are thus easier to be interpreted or interrogated than those abstract mathematical features extracted from the time or frequency domain [8].

Metaheuristic optimization methods are the recent generation of optimization methods. These methods are inspired from natural phenomena. Particle swarm optimization proposed by Eberhart and Kennedy [9] and ant colony optimization proposed by Dorigo et al. [10] simulate social behavior of animals. Harmony search presented by Geem et al. [11], Big Bang–Big Crunch algorithm proposed by Erol and Eskin [12], charged system search proposed by Kaveh and Talatahari [13], magnetic charged system search (MCSS) proposed by Kaveh et al. [14], ray optimization of Kaveh and Khayatazad [15], and dolphin echolocation optimization of Kaveh and Farhoudi [16] are other metaheuristic algorithms which have sources in nature.

In this chapter an objective function based on natural frequencies and mode shapes is used to solve damage detection problem. Charged system search algorithm and enhanced charged system search are utilized to search for global optimum of the proposed objective function. The damage detection methodology is applied to four different types of structures.

11.2 Damage Identification Methodology

The proposed damage detection method consists of performing an optimization problem through an objective function based on vibration data. Here, damage is considered as a reduction in the elastic modulus.

11.2.1 Objective Function

The objective function is based on natural frequencies and mode shapes and is given by Eq. (11.1). Due to measurement noise, tendency will always be to find damage at most of the elements [17]. Thus, a penalty is introduced to weigh against an increased number of damage sites:

$$\text{cost} = E(1 + \beta \times \text{penalty}), \quad E = E_\phi + E_\omega \tag{11.1}$$

$$E_\phi = \sum_{j=1}^{r} \frac{\phi_j^m - \phi_j^a}{\phi_j^m + \phi_j^a} \tag{11.2}$$

$$E_\omega = \sum_{j=1}^{r} \left(\frac{\left(\omega_j^m - \omega_j^a\right)^2}{\left(\omega_j^m\right)^2} \right) \tag{11.3}$$

where ω_j^m and ω_j^a are the jth measured and analytical natural frequencies of the damaged structure, respectively; ϕ_j^m and ϕ_j^a are the measured and analytical values of the jth mode shapes, respectively; r is the number of measured modes; and β is a penalty factor which is related to the type of structure and the closeness of the measured data and the exact data. Here, penalty is the number of damaged elements in the analytical model.

11.3 Optimization Algorithm

11.3.1 Standard Charged Search System

Charged system search is a population-based metaheuristic algorithm proposed by Kaveh and Talatahari [12]. This algorithm is based on laws from electrostatics of physics and Newtonian mechanics. The pseudo-code of the CSS algorithm is presented as follows [18]:

Level 1: Initialization

Step 1: Initialization. Initialize the parameters of the CSS algorithm. Initialize an array of charged particles (CPs) with random positions. The initial velocities of the CPs are taken as zero. Each CP has a charge of magnitude (q) defined considering the quality of its solution as

$$q_i = \frac{\text{fit}(i) - \text{fit}_{\text{worst}}}{\text{fit}_{\text{best}} - \text{fit}_{\text{worst}}} \tag{11.4}$$

where fit_{best} and $\text{fit}_{\text{worst}}$ are the best and the worst fitness of all the particles respectively, and $\text{fit}(i)$ represents the fitness of agent i. The separation distance r_{ij} between two charged particles is defined as

$$r_{ij} = \frac{X_i - X_j}{\frac{(X_i + X_j)}{2} - X_{\text{best}} + \varepsilon} \tag{11.5}$$

where X_i and X_j are the positions of the ith and jth CPs, respectively, X_{best} is the position of the best current CP, and ε is a small positive value to avoid singularities.

Step 2: CP ranking. Evaluate the magnitudes of the fitness function for the CPs, compare with each other, and sort them in increasing order.

Step 3: CM creation. Store the number of the first CPs equal to charged memory size (CMS) and their related values of the fitness functions in the charged memory (CM).

Level 2: Search

Step 1: Attracting force determination. Determine the probability of moving each CP toward the others considering the following probability function:

$$pm_{ji} = \begin{cases} 1 \Leftrightarrow & \text{fit}(i) > \text{fit}(j) \vee 0.02\left(1 - \left(\frac{\text{iter}}{\text{iter}_{\max}}\right)\right) > rand \\ 0 \Leftrightarrow & \text{else}, \end{cases} \tag{11.6}$$

and calculate the attracting force vector for each CP as follows:

$$F_j = q_j \sum_{i, i \neq j} \left(\frac{q_i}{a^3} r_{ij} \cdot i_1 + \frac{q_i}{r_{ij}^2} \cdot i_2 \right) p_{ij} \left(X_i - X_j \right), \begin{cases} i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a, \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a, \end{cases} \tag{11.7}$$

where F_j is the resultant force affecting the jth CP.

Step 2: Solution construction. Move each CP to the new position and find its velocity using the following equations:

$$X_{j,\mathrm{new}} = rand_{j1} \cdot k_{\mathrm{a}} \cdot \frac{F_j}{m_j} \cdot \Delta t^2 + rand_{j,2} \cdot k_{\mathrm{v}} \cdot V_{j,\mathrm{old}} \cdot \Delta t + X_{j,\mathrm{old}} \tag{11.8}$$

$$V_{j,\mathrm{new}} = \frac{X_{j,\mathrm{new}} - X_{j,\mathrm{old}}}{\Delta t} \tag{11.9}$$

where $rand_{j2}$ and $rand_{j2}$ are two random numbers uniformly distributed in the range (1,0); *mj* is the mass of the CPs, which is set to unity in this chapter; Δt is the time step, which is set to 1; k_{a} is the acceleration coefficient; and k_{v} is the velocity coefficient to control the influence of the previous velocity. In this chapter k_{v} and k_{a} are taken as

$$k_{\mathrm{a}} = c_{\mathrm{a}}\left(1 + \frac{\mathrm{iter}}{\mathrm{iter}_{\max}}\right) \tag{11.10}$$

$$k_{\mathrm{v}} = c_{\mathrm{v}}\left(1 - \frac{\mathrm{iter}}{\mathrm{iter}_{\max}}\right) \tag{11.11}$$

where c_{a} and c_{v} are two constants to control the exploitation and exploration of the algorithm, iter is the iteration number, and $\mathrm{iter}_{\max}$ is the maximum number of iterations.

Step 3: CP position correction. If each CP exits from the allowable search space, correct its position.

Step 4: CP ranking. Evaluate and compare the values of the fitness function for the new CPs, and sort them in an increasing order.

Step 5: CM updating. If some new CP vectors are better than the worst ones in the CM, in terms of their objective function values, include the better vectors in the CM and exclude the worst ones from the CM.

Level 3: Controlling the Terminating Criterion

Repeat the search level steps until a terminating criterion is satisfied.

11.3.2 Enhanced Charged Search System

As mentioned before, CSS is a population-based algorithm. For multi-agent methods, the updating process is performed after all agents have created their solutions. Similarly, for the CSS algorithm, when the calculations of the amount of forces are completed for all CPs and the new locations of agents are determined, the CM updating is performed. In the present case, it is assumed that after creating each solution, all updating processes are performed. In this way, the new position of each agent can affect on the moving of the subsequent CPs, while in the standard CSS unless an iteration is completed, the new positions cannot be utilized. Due to using the information obtained by the CPs immediately after creation, this modification enhances the intensification of the algorithm [19].

11.4 Numerical Examples

In this section, the efficiency and effectiveness of the proposed methods are evaluated through some numerically simulated damage identification tests using incomplete modal data. A continuous beam, a three-story and three-span plane frame, and a two- and three-dimensional truss are considered with two different damage scenarios for each of them. Due to the stochastic nature of the metaheuristic algorithms for each scenario, the algorithm is run ten times and the solution with the lowest cost is selected as the ultimate damage scenario. The mode shapes are measured with less accuracy than the natural frequencies. In order to simulate the conditions of a real test, the measured parameters are numerically perturbed by 1 % for natural frequencies and 3 % for mode shapes to consider the presence of the noise.

11.4.1 A Continuous Beam

For the first example, a continuous beam depicted in Fig. 11.1 is considered. Beam length is equally divided into 26 elements with a uniform section (IPE240). The area of cross section and moment of inertia of the simulated beam are 39.1 cm^2 and 3892 cm^4, respectively. The modulus of elasticity and the material density are 200 GPa and 7780 kg/m^3, respectively. The first six natural frequencies and mode shapes of the structure are used to form the objective function. Figures 11.2 and 11.3 represent the damage states found by both optimization algorithms with the actual damage states in different scenarios.

11.4.2 A Planar Frame

The frame with three spans and three stories depicted in Fig. 11.4 is considered as the second example. The sections used for the beams and columns are IPE240 and IPE300, respectively. The modulus of elasticity and material density are identical to those of the previous model. The first six natural frequencies and six mode shapes of the structure are utilized to form the objective function. Figures 11.5 and 11.6 represent the damage states found by both optimization algorithms with the actual damage states in different scenarios.

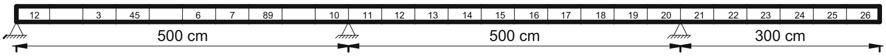


Fig. 11.1 Schematic of a beam modeled with 26 finite elements

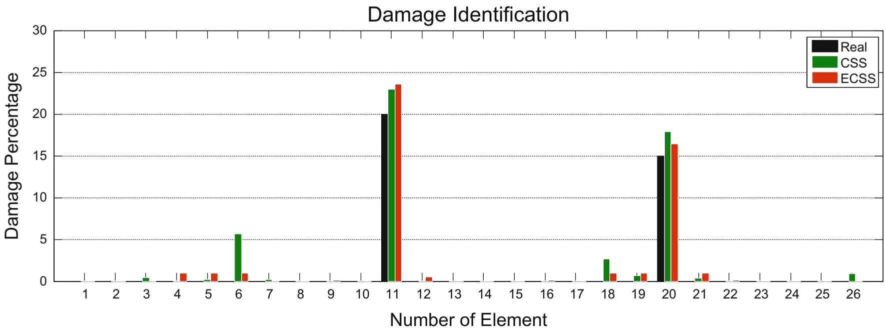


Fig. 11.2 Damage detection results of the algorithms for the beam (scenario I)

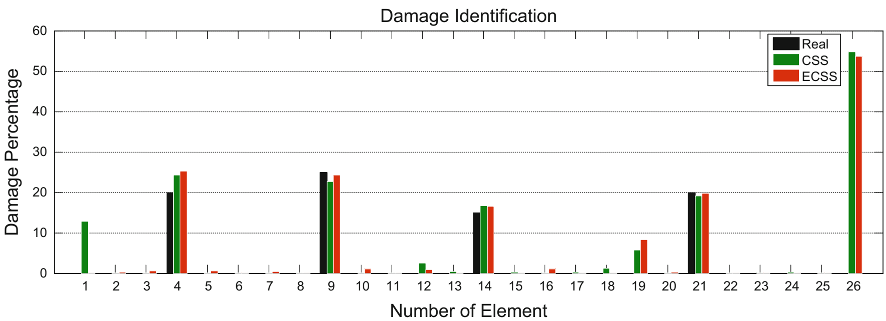


Fig. 11.3 Damage detection results of the algorithms for the beam (scenario II)

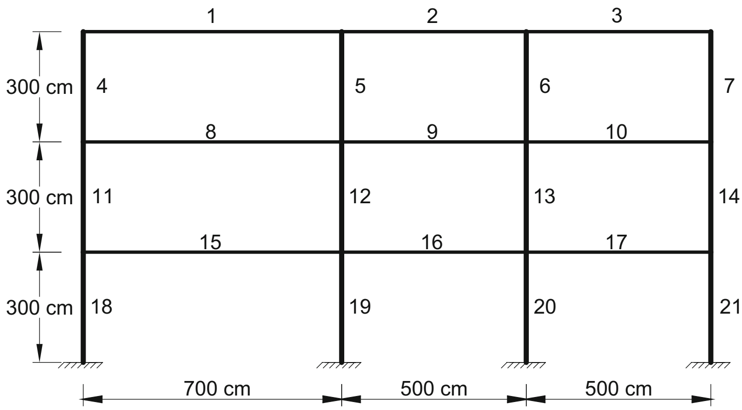


Fig. 11.4 Schematic of a three-span two-story frame

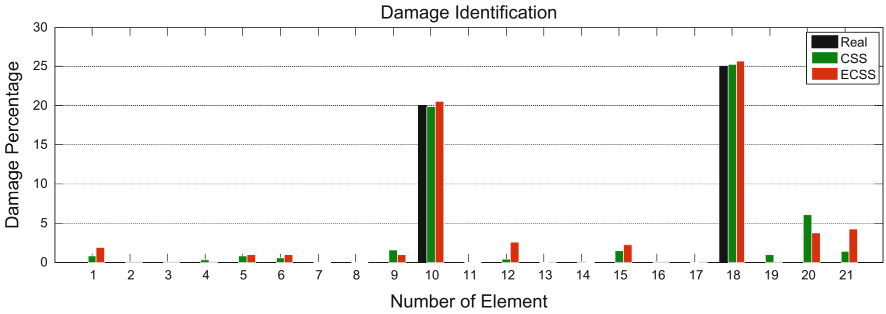


Fig. 11.5 Damage detection results of the algorithms for the three-span two-story frame (scenario I)

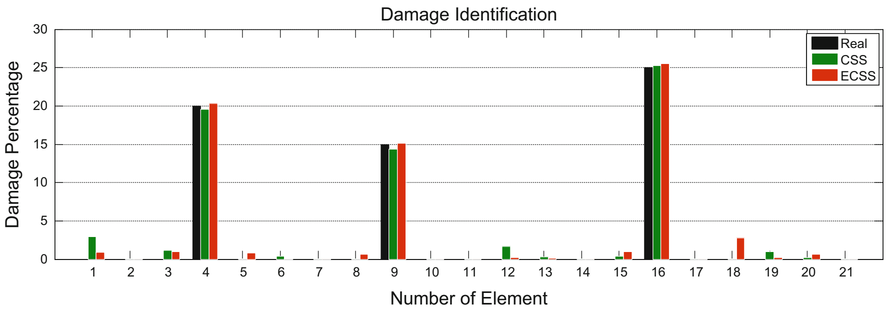


Fig. 11.6 Damage detection results of the algorithms for the three-span two-story frame (scenario II)

11.4.3 A Planar Truss

As the third example, a statically indeterminate truss bridge shown in Fig. 11.7 is considered. The area of cross section for all elements is taken as 10 cm^2. The modulus of elasticity and material density are the same as the previous model. The first five natural frequencies and mode shapes of the structure are used to form the objective function. Figures 11.8 and 11.9 represent the damage states found by both optimization algorithms with the actual damage states in different scenarios.

11.4.4 A Space Truss

A space truss is considered as the last example. The geometry, element numbering, and material properties are shown in Fig. 11.10. The first six natural frequencies and mode shapes of the structure are utilized to form the objective function. Figures 11.11 and 11.12 represent the damage states found by both optimization algorithms with the actual damage states in different scenarios.

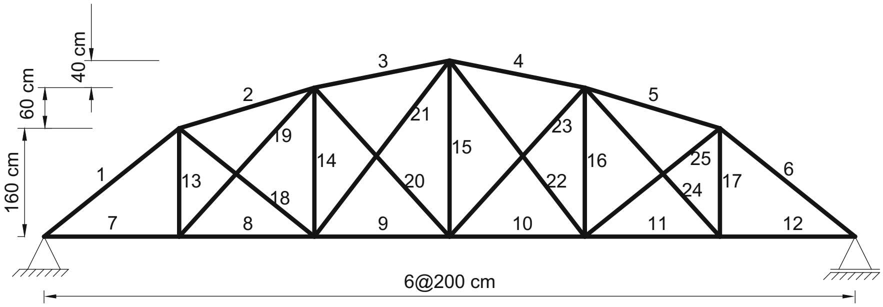


Fig. 11.7 Schematic of a truss with 25 elements

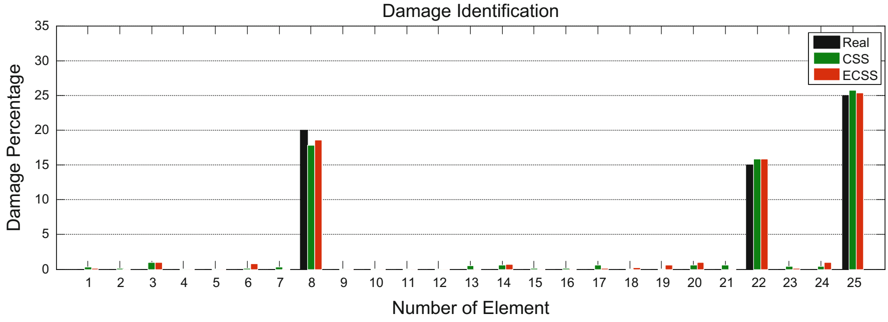


Fig. 11.8 Damage detection results of the algorithms for the planar truss (scenario I)

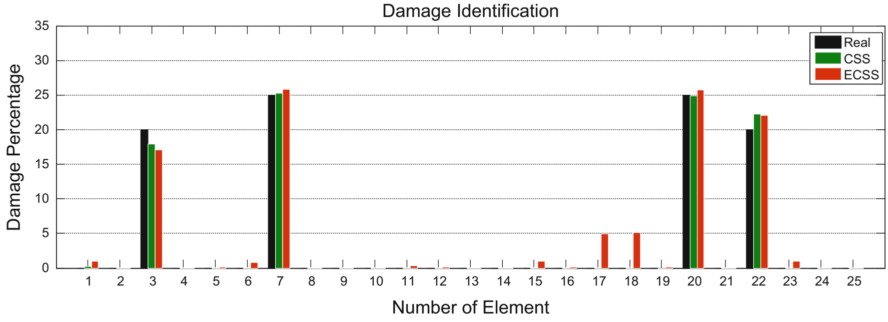


Fig. 11.9 Damage detection results of the algorithms for the planar truss (scenario II)

11.5 Concluding Remarks

A method for damage detection in skeletal structures based on natural frequencies and mode shapes is studied in this chapter. A penalty approach is applied to moderate the effect of noise on modal data. Two versions of the CSS are utilized for searching the correct damage scenarios. Damage detection is conducted on a

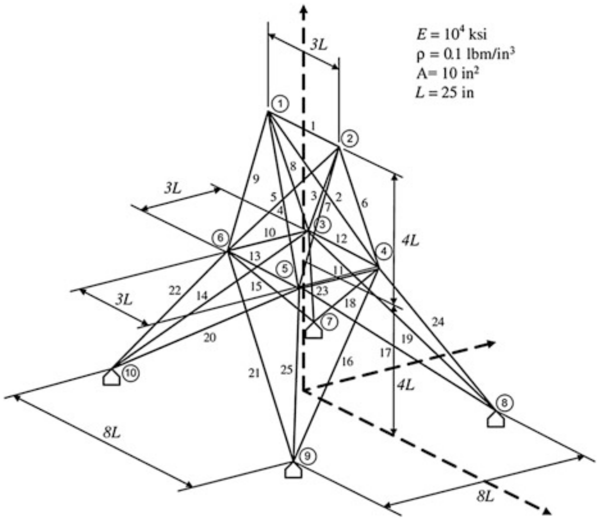


Fig. 11.10 Schematic of a space truss with 25 elements

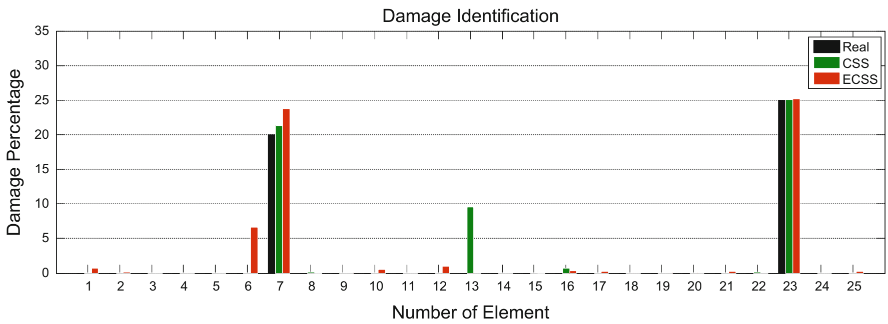


Fig. 11.11 Damage detection results of the algorithms for the space truss (scenario I)

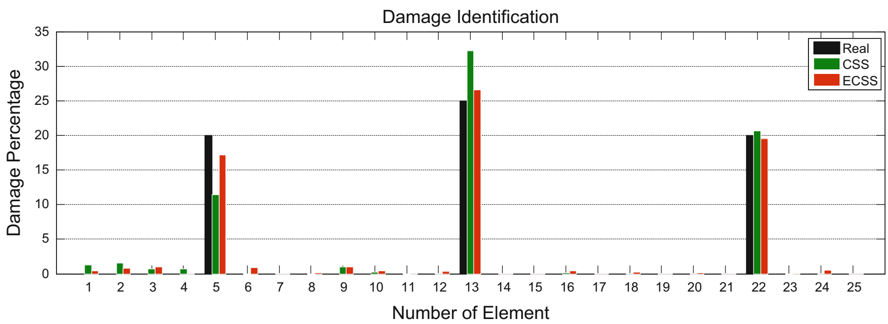


Fig. 11.12 Damage detection results of the algorithms for the space truss (scenario II)

variety of numerical problems with different scenarios to verify the performance of the proposed methodologies. In most of the cases, the results show that the algorithm successfully finds the location and the severity of the damages. In the continuous beam, the cantilever part is adversely affected by the noise which causes a misidentification in the second scenario for both algorithms. Generally, it can be concluded that the both proposed algorithms are quite efficient and robust for damage detection problems, and they can identify the locations and severities of damages using incomplete modal data which is contaminated by random noise.

References

1. Kaveh A, Maniat M (2014) Damage detection in skeletal structures based on charged system search optimization using incomplete modal data. Int J Civil Eng IUST 12(2):291–298
2. Doebling Scott W, Farrar CR, Prime MB (1998) A summary review of vibration-based damage identification methods. Shock Vib Dig 30:91–105
3. Perera R, Torres R (2006) Structural damage detection via modal data with genetic algorithms. J Struct Eng 132:1491–1501
4. Laier JE, Morales JD (2009) Improved genetic algorithm for structural damage detection. Comput Struct Eng 91:833–839
5. Miguel L, Miguel L, Kaminski J Jr, Riera J (2012) Damage detection under ambient vibration by harmony search algorithm. Expert Syst Appl 39(10):9704–9714
6. Kang F, Li JJ, Xu Q (2012) Damage detection based on improved particle swarm optimization using vibration data. Appl Soft Comput 12(8):2329–2335
7. Majumdar A, Kumar Maiti D, Maity D (2012) Damage assessment of truss structures from changes in natural frequencies using ant colony optimization. Appl Math Comput 218:9759–9772
8. Fan W, Qiao P (2011) Vibration-based damage identification methods: a review and comparative study. Struct Health Monit 10:83–111
9. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan
10. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B26:29–41
11. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony Search. Simulation 76:60–68
12. Erol OK, Eksin I (2006) New optimization method: Big Bang–Big Crunch. Adv Eng Softw 37:106–111
13. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289
14. Kaveh A, Motie Share MA, Moslehi M (2013) A new metaheuristic algorithm for optimization: magnetic charged system search. Acta Mech 224:85–107
15. Kaveh A, Khayatazad M (2012) A new metaheuristic method: ray optimization. Comput Struct 112–113:283–294
16. Kaveh A, Farhoudi N (2013) A new optimization method: dolphin echolocation. Adv Eng Softw 59:53–70
17. Friswell MI, Penny JET, Garvey SD (1998) A combined genetic and Eigen-sensitivity algorithm for the location of damage in structures. Comput Struct 69:547–556
18. Kaveh A, Talatahari S (2011) Optimization of large-scale truss structures using modified charged system search. Int J Optim Civil Eng 1:15–28
19. Kaveh A, Talatahari S (2011) An enhanced charged system search for configuration optimization using the concept of fields of forces. Struct Multidiscip Optim 43:339–351

Chapter 12
Modification of Ground Motions Using Enhanced Colliding Bodies Optimization Algorithm

12.1 Introduction

In this chapter a simple and robust approach is presented for spectral matching of ground motions utilizing the wavelet transform and an improved metaheuristic optimization technique. For this purpose, wavelet transform is used to decompose the original ground motions to several levels, where each level covers a special range of frequency, and then each level is multiplied by a variable. Subsequently, the enhanced colliding bodies optimization (ECBO) technique is employed to calculate the variables such that the error between the response and target spectra is minimized. The application of the proposed method is illustrated through modifying 12 sets of ground motions [1].

Recent aseismic code regulations recommend the use of linear or nonlinear dynamic time history analyses for design of irregular, high rise, and important structures due to the increased capabilities of the commercial software to account for the potential inelastic behavior of structural systems under seismic time histories. These acceleration time histories can be achieved either by using a set of real recorded earthquake accelerograms associated with historical seismic events, or utilizing an ensemble of numerically simulated earthquake signals. In the latter approach, one can make pure artificial records and filter them according to the site characteristics or to reconstruct the real record so that its spectrum fits the target standard [2]. Obviously finding suitable methods for reconstructing or modifying realistic ground motions is an important and challenging problem.

The main objective of the reconstruction/modification of ground motions is to modify a given set of ground motions such that these response spectrums become compatible with a specified design spectrum. For this purpose, various time or frequency-domain methods are used. The time-domain methods manipulate only the amplitude of the recorded ground motions, while the frequency-domain approaches operate the frequency contents and phasing of actual ground motions in order to match with the design spectrum. During the last two decades, a number

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_12

of researches are performed on this problem employing the frequency-domain methods. Gupta and Joshi [3] and Shrikhande and Gupta [4] used the phase characteristics of recorded accelerograms. Conte and Peng [5] directly modeled the evolutionary power spectral density function of the ground motion process. Recently, many researches focused on modifying the recorded ground motions using wavelet (e.g., Refs. [6–10]). For examples, Hancock et al. [6] utilized wavelet and Mukherjee and Gupta [7] developed an iterative wavelet-based method for spectral matching. Cecini and Palmeri [8] also proposed an iterative procedure based on the harmonic wavelet transform to match the target spectrum through deterministic corrections to a recorded accelerogram. As will be mentioned in the coming sections, these works achieved an iterative approach to obtain the sought spectrum-compatible accelerograms. These approaches do not guarantee the requirements of the code regulations.

In this chapter an approach is utilized to modify the real ground motions such that these response spectrums become compatible with the elastic spectrum of the European Code (CEN [11]) regulation. For this purpose, wavelet transform is used to decompose the ground motions to several levels each covering a special range of frequency. Then each level is multiplied by a variable. Subsequently, an optimization algorithm is employed to calculate the variables to minimize the error between response and target spectrums, while the requirements of the code regulations are considered as constrains of the optimization process [1].

Optimization algorithms can be divided into two categories: (1) deterministic and (2) stochastic. Deterministic algorithms are mostly gradient-based methods, and the stochastic algorithms consist of heuristic and metaheuristic methods. These optimization techniques which mimic stochastic natural phenomena have emerged as robust and reliable computational tools compared to the conventional gradient-based methods in solving complex problems. The stochastic nature of such algorithms allows exploration of a larger fraction of the search space compared to the case of gradient-based methods. Since the objective function of this work (the difference between design spectrum and average response spectrum of modified ground motion) is non-smooth and non-convex, the gradient-based optimization methods can be trapped in local optima. Thus, a recently developed metaheuristic algorithm is utilized to optimize this objective function. Some algorithms based on natural evolution phenomenon are developed by Eberhart and Kennedy [12], Dorigo et al. [13], Eroland and Eksin [14], Kaveh and Talatahari [15], Sadollah et al. [16], and Kaveh and Mahdavi [17]. ECBO is an improved version of the recently developed metaheuristic algorithm so-called colliding bodies optimization (CBO) [18]. Simple formulation and the need for no parameter tuning are the main characteristics of this algorithm.

12.2 Spectral Matching Problem According to Eurocode-8

12.2.1 Standard Design Spectrum in Eurocode-8

The elastic acceleration response spectrum, $S_a(T)$, for oscillators with 5 % ratio of critical damping and natural period, T, is defined by the European seismic code provisions (CEN [11]) as

$$S_a(T) = \begin{cases} \alpha_g S\left(1 + \dfrac{1.5T}{T_B}\right) & 0 \le T \le T_B \\ 2.5\alpha_g S & T_B \le T \le T_c \\ 2.5\alpha_g S\left(\dfrac{T_C}{T}\right) & T_C \le T \le T_D \\ 2.5\alpha_g S\left(\dfrac{T_C T_D}{T^2}\right) & T_D \le T \le 4s \end{cases} \tag{12.1}$$

where S is the soil factor, T_B and T_C are the limiting periods of the constant spectral acceleration branch, T_D defines the beginning of the constant displacement response range of the spectrum, and a_g is the design ground acceleration on type A ground, which is defined according to the seismic hazard. In this study, a_g is chosen as 0.35 g.

The values of the periods T_B, T_C, and T_D and the soil factor S describing the shape of the elastic response spectrum depend on the ground type. In Table 12.1, the specific values that determine the spectral shapes for Type 1 spectra are listed, and the resulting spectra is normalized by a_g and plotted in Fig. 12.1.

12.2.2 Spectra Matching Requirements Based on Eurocode-8

According to Eurocode-8, seismic ground motions can be classified depending on the nature of the application and on the information actually available by natural, artificial, or simulated accelerograms. These seismic ground motions should reflect some important seismological parameters in local seismic scenarios and should match the following criteria: (1) a minimum of 3 accelerograms should be used; (2) mean of the zero period spectral response acceleration values should not be

Table 12.1 Values of the parameters describing the recommended Type 1 elastic response spectra

Ground type	S	T_B (s)	T_C (s)	T_D (s)
A	1.0	0.15	0.4	2.0
B	1.2	0.15	0.5	2.0
C	1.15	0.2	0.6	2.0
D	1.35	0.2	0.8	2.0
E	1.4	0.15	0.5	2.0

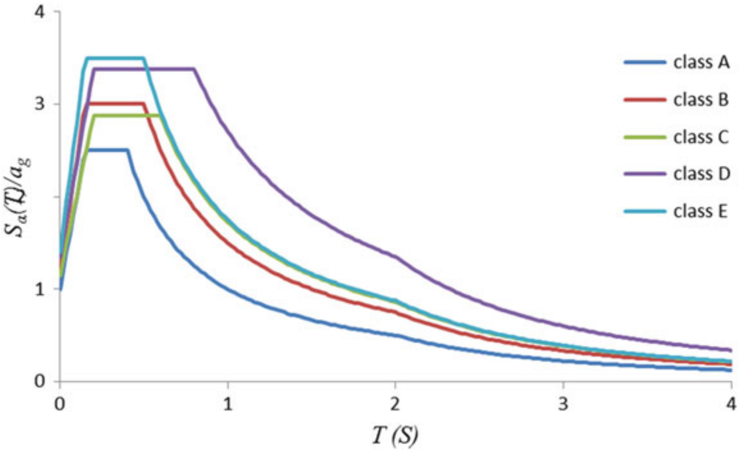


Fig. 12.1 Elastic response spectra for different site soil classes, based on the EC8

smaller than the value of a_gS for the site in question; and (3) in the range of periods between $0.2T_n$ and $2T_n$, where T_n is the fundamental period of the structure in the direction where the accelerogram is applied, no value of the mean 5 % damping elastic spectrum calculated from all time histories should be $< 90\,\%$ of the corresponding value of the 5 % damping elastic response spectrum.

Moreover, the code allows the consideration of the mean effect on the structure, rather than the maximum effect if at least seven nonlinear time history analyses are performed.

12.3 Wavelet Transform

Wavelet transform provides a powerful tool to characterize local features of a signal. Unlike Fourier transform, where the function used as the basis of decomposition is always a sinusoidal wave, other basis functions can be selected for wavelet shape according to the features of the signal. The wavelet transform uses a series of high-pass filters to analyze high frequencies of a signal, and a series of low-pass filters to analyze low frequencies of a signal. In the first level of wavelet transform process, the signal $f(t)$, which is a finite energy function, is filtered into high- and low-pass frequency signals indicating a detailed and approximate version of the original signal, respectively. The low-pass filtered signal (i.e., approximate signal) is sent to next level, and it filters into high- and low-pass frequency signals once again. The decomposition levels continue until the desired level is attained, as shown in Fig. 12.2.

By decomposing a signal $f(t)$ of length T into n signals, the detailed signal at level j ($D_j(t)$) is defined as

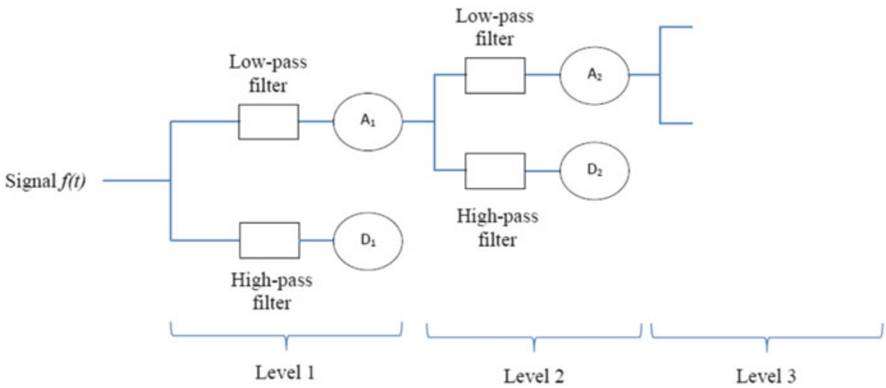


Fig. 12.2 Signal decomposition in wavelet transform

$$D_j(t) = \sum_{k=-\infty}^{\infty} cD_j(k)\psi_{j,k}\mathrm{d}k \tag{12.2}$$

where ψ_j is the wavelet function, k is the translation parameter, and $cD_j(k)$ is the wavelet coefficient at level j which is defined as

$$cD_j(k) = \int_{-\infty}^{\infty} f(t)\psi_{j,k}\mathrm{d}t \tag{12.3}$$

The approximate signal at level j is defined as

$$A_j(t) = \sum_{k=-\infty}^{\infty} cA_j(k)\varphi_{j,k}\mathrm{d}k \tag{12.4}$$

where φ_j is the scaling function and $cA_j(k)$ is the scaling coefficient at level j which is defined as

$$cA_j(k) = \int_{-\infty}^{\infty} f(t)\varphi_{j,k}\,\mathrm{d}t \tag{12.5}$$

In this chapter for decomposing the signals, Daubechies wavelet and scaling function of order 10 (db-10) are used [19]. Finally, the signal $f(t)$ can be represented by

$$f(t) = A_n(t) + \sum_{j\leq n} D_j(t) \tag{12.6}$$

In wavelet transformation, scaling and wavelet functions are used. These are related to low-pass and high-pass filters, respectively. A wavelet function can also be represented as

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t - 2^j k}{2^j}\right) \tag{12.7}$$

The scaling function can also be expressed as

$$\varphi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\varphi\left(\frac{t - 2^j k}{2^j}\right) \tag{12.8}$$

In wavelet transform, each $D_j(t)$ has nonzero components only in an exclusive range of frequency which is denoted by

$$\text{Frequency range of level}\, j = [f1, f2] = \left[\frac{1}{2^{j+1}\Delta t}, \frac{1}{2^j \Delta t}\right] \tag{12.9}$$

$$\text{Period range of level}\, j = [T1, T2] = \left[2^j \Delta t, 2^{j+1}\Delta t\right] \tag{12.10}$$

where Δt is the time step of the signal $f(t)$ (Refs. [20, 21]).

12.4 The Proposed Methodology

An iterative method is used for solving spectral matching problem that is based on the work of Mukherjee and Gupta [7]. In this method, first an ordinary ground motion is decomposed using wavelet transform, and detailed signals are determined. Then, the ground motion is modified by scaling each of the detailed signals (D_j) up/down based on the amplification/reduction required to reach target spectral ordinates in the period band corresponding to that time history. Thus, in the ith iteration, the detailed signals $\left(D_j^i\right)$ are modified for level j to the modified detailed signal $\left(D_j^{i+1}\right)$ such that

$$D_j^{i+1} = D_j^i \frac{\int_{T1}^{T2} [S_a(T)]_{\text{Target}} \mathrm{d}T}{\int_{T1}^{T2} [PSA(T)]_{\text{calculated}} \mathrm{d}T} \tag{12.11}$$

where $T1$ and $T2$ are the period bounds on the range of level j [Eq. (12.10)]. Finally, a modified ground motion is constructed using Eq. (12.6). The disadvantages of this method can be mentioned as (i) it modifies only one ground motion, (ii) it cannot handle the manual requirements, and (iii) it needs a non-overlapping wavelet transform for decomposing ground motion.

Here, we propose a new method based on a constrained metaheuristic algorithm, where its variables are scaling factors of Eq. (12.11), and wavelet transform modifies the recorded accelerograms until the response spectrum gets close to a specified design spectrum. Further, the response spectrum obtained from modified accelerograms should also satisfy the requirements of the Eurocode-8 mentioned in Sect. 12.2.

The proposed method is briefly outlined as follows:

Step 1. *Selection of ground motions*: A set of ground motions is selected. According to Eurocode-8, the minimum number of records for this selection is 3. In this chapter, three horizontal ground motion components with identical soil conditions are selected from the well-known PEER strong motion database [22].

Step 2. *Decomposition of the ground motions*: In this step the ground motions are decomposed with wavelet to levels $j = n$, and the detailed and approximate signals (D_j and A_j) at each level are specified based on Eqs. (12.2) and (12.4), respectively. The number of decomposition levels (n) depends on the studied period range. In this chapter, the studied period range and the time step of ground motions are taken as 0–5 s and 0.01 s, respectively. Given Eq. (12.10), the ground motions are decomposed into eight levels using wavelet with the detailed coefficients covering the period range of [0–5.12] s.

Step 3. *Reconstruction of the modified ground motions*: After specifying the detailed and approximate signals of the original ground motions in each level (in the previous step), the modified ground motions ($f_m(t)$) can be expressed by the following equation:

$$f_m(t) = \sum_{j=1}^{n} \left(\alpha_j D_j\right) + \alpha_{n+1} A_n \tag{12.12}$$

where D_j and A_n are the detailed and approximate signals at levels j and n, respectively, and α_j is the jth modified value. In fact, this value is a variable in the optimization process. The number of optimization variables is equal to $n + 1$ multiplied by the number of ground motions, and in the present chapter, this is equal to $9 * 3 = 27$.

Step 4. *Creation of the response spectrum*: In this step, the response pseudo-acceleration spectrums of the modified ground motions is determined. As mentioned before based on Eurocode-8, when a set of 3 through 6 ground motions is used, the structural engineer should use the maximum response value instead of the mean response value. Hence, the response spectrum of ground motions should be calculated as

$$PSA(T) = \max(PSA_i(T)) \quad i = 1, 2, 3 \tag{12.13}$$

where $PSA_i(T)$ is the pseudo-acceleration spectrum of the ith modified ground acceleration in period T which is calculated as

$$PSA(\omega, \xi) = \omega^2 \max_t (|x(t)|), \quad \xi = 5\,\%, \quad \omega = \frac{2\pi}{T} \tag{12.14}$$

$$\ddot{x}(t) + 2\xi\omega\dot{x}(t) + \omega^2 x(t) = -f_m(t) \tag{12.15}$$

where ω, ζ, and $f_m(t)$ are the fundamental frequency, the damping coefficient of the single degree of freedom system, and the earthquake ground acceleration, respectively.

Step 5. *Determination of the penalty function*: In this chapter penalty method is utilized to satisfy the code requirements:

$$\text{Penalty} = P_1 + P_2 + P_3 \tag{12.16}$$

$$P_1 = \max\bigl(0, \max_i (0.9^* S_a(T_i) - PSA(T_i)), \quad 0.2T_n \le T_i \le 2T_n \tag{12.17}$$

$$P_2 = \max(0, S_a(T_1) - PSA(T_1)), \quad T_1 = 0 \tag{12.18}$$

$$P_3 = \max\left(0, -\max_i (\alpha_i)\right), \quad i = 1, 2, \ldots, 27 \tag{12.19}$$

Here, P_1 and P_2 are considered in order to prevent the maximum response spectrum from falling below the target spectrum within the code-specific period range and zero period, respectively; P_3 keeps the values of scale factors in the range of >0. S_a and T_n are the target spectrum and fundamental period of structure, respectively.

Step 6. *Computation of the objective function*. In this step the objective function of the optimization process is computed as

$$F(X) = \text{Err}(X)^*(1 + \lambda^*\text{penalty}(X)) \tag{12.20}$$

where X is the vector of the optimization variables [i.e., the modified values in Eq. (12.12)], λ is a large number which is selected to magnify the penalty effects, and Err is calculated using Eq. (12.21) as the response spectrum becomes close to the target spectrum:

$$\text{Err}(X) = 100^* \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\log(S_a(T_i)) - \log(PSA(T_i)))^2} \tag{12.21}$$

where N is the number of specified periods. Here, 500 period points are considered in the range [0–5] s with period steps of 0.01 s.

Step 7. *Termination criterion*: The optimization process is repeated starting with Step 3 until the maximum number of iterations as a termination criterion is attained.

Step 8. *Correction of baseline*: The velocity and displacement time history of reconstructed ground accelerations do not become unrealistic due to systematic low-frequency errors. Hence, the baseline correction of the modified accelerograms is needed for this purpose.

The flowchart of this method is shown in Fig. 12.3.

12.5 Enhanced Colliding Bodies Optimization Algorithm

The ground motion modification problem is a complex problem because of having a large search space, multiple local optima, and corresponding constraints. In this chapter we apply a simple and efficient metaheuristic algorithm, so-called ECBO, to solve this problem. For comparative study and showing the complexity of the problem, the standard CBO is also utilized. In the following, both standard CBO and ECBO algorithms are briefly introduced.

12.5.1 Colliding Bodies Optimization Algorithm

The CBO is based on momentum and energy conservation law for one-dimensional collision (Kaveh and Mahdavi [23]). This algorithm contains a number of colliding bodies (CB) where each one is treated as an object with specified mass and velocity which collides with others. After collision, each CB moves to a new position with new velocity with respect to previous velocities, masses, and coefficient of restitution. CBO starts with a set of agents determined with random initialization of a population of individuals in the search space. Then, CBs are sorted in an ascending order based on the values of their cost functions (see Fig. 12.4a). The sorted CBs are divided equally into two groups. The first group is stationary and consists of good agents. This set of CBs is stationary and their velocity before collision is zero. The second group consists of moving agents which move toward the first group. Then, the better and worse CBs, i.e., agents with upper fitness values of each group, collide together to improve the positions of the moving CBs and to push stationary CBs toward better positions (see Fig. 12.4b). The change of the body position represents the velocity of the CBs before collision as

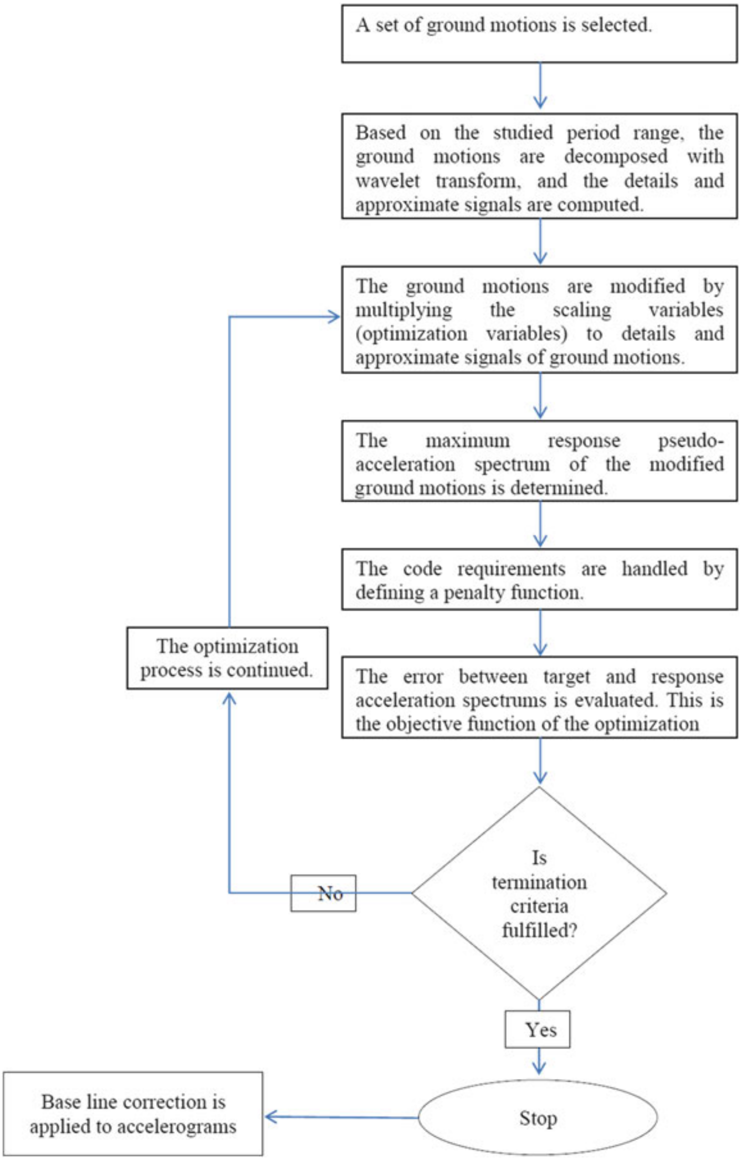


Fig. 12.3 Flowchart of the proposed method

$$v_i = \begin{cases} 0, & i = 1, \ldots, n \\ x_i - x_{i-n}, & i = n+1, \ldots, 2n \end{cases} \tag{12.22}$$

where, v_i and x_i are the velocity vector and position vector of the ith CB, respectively. $2n$ is the number of population size.

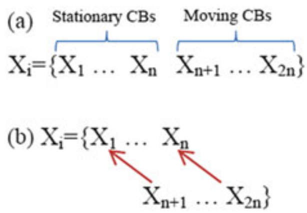


Fig. 12.4 (**a**) The sorted CBs in an increasing order, (**b**) the mating process for the collision

After the collision, the velocity of bodies in each group is evaluated using momentum and energy conservation law and the velocities before collision [Eq. (12.22)]. The velocity of the CBs after the collision becomes

$$v_i' = \begin{cases} \dfrac{(m_{i+n} + \varepsilon m_{i+n})v_{i+n}}{m_i + m_{i+n}}, & i = 1, \ldots, n \\ \dfrac{(m_i - \varepsilon m_{i-n})v_i}{m_i + m_{i-n}}, & i = n+1, \ldots, 2n \end{cases} \tag{12.23}$$

where v_i and v_i' are the velocities of the ith CB before and after the collision, respectively; m_i is the mass of the ith CB defined as

$$m_k = \frac{\frac{1}{fit(k)}}{\sum_{i=1}^{n} \frac{1}{fit(i)}}, \quad k = 1, 2, \ldots, 2n \tag{12.24}$$

where $fit(i)$ represents the objective function value of the ith agent. Obviously a CBs with better objective function values will be assigned with larger mass values. Also, for maximizing the objective function, the term $1/fit(i)$ is replaced by $fit(i)$. ε is the coefficient of restitution (COR) and is defined as the ratio of the separation velocity of the two agents after collision to approach velocity of two agents before collision. In this algorithm, this index is defined to control the exploration and exploitation rates. For this purpose, the COR decreases linearly from unit value to zero. Here, ε is defined as

$$\varepsilon = 1 - \frac{iter}{iter_{\max}} \tag{12.25}$$

where $iter$ is the actual iteration number and $iter_{\max}$ is the maximum number of iterations. Here, COR is equal to unity and zero representing the global and local search, respectively. In this way a good balance between the global and local search is achieved by increasing the iteration number.

The new positions of CBs are evaluated using the generated velocities after the collision:

$$x_i^{new} = \begin{cases} x_i + rand \circ v_i', & i = 1, \ldots, n \\ x_{i-n} + rand \circ v_i', & i = n+1, \ldots, 2n \end{cases} \tag{12.26}$$

where x_i^{new} and x_i are the new position and the velocity after the collision of the ith CB, respectively.

12.5.2 Enhanced Colliding Bodies Optimization Algorithm

In order to improve the CBO to obtain faster and more reliable solutions, ECBO is developed which uses memory to save a number of historically best CBs and also utilizes a mechanism to escape from local optima (Kaveh and Ilchi Ghazaan [18]). The steps of this technique are as follows:

Level 1: Initialization

Step 1: The initial positions of all the CBs are determined randomly in the search space.

Level 2: Search

Step 1: The value of mass for each CB is evaluated according to Eq. (12.24).
Step 2: Colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values. Solution vectors which are saved in CM are added to the population, and the same number of current worst CBs are removed. Finally, CBs are sorted according to their masses in a decreasing order.
Step 3: CBs are divided into two equal groups: (i) stationary group and (ii) moving group (Fig. 12.4).
Step 4: The velocities of stationary and moving bodies before collision are evaluated by Eq. (12.22).
Step 5: The velocities of stationary and moving bodies after the collision are evaluated using Eq. (12.23).
Step 6: The new position of each CB is calculated by Eq. (12.26).
Step 7: A parameter like **Pro** within (0, 1) is introduced, which specifies whether a component of each CB must be changed or not. For each colliding body, **Pro** is compared with rn_i $(i = 1, 2, \ldots, n)$ which is a random number uniformly distributed within (0, 1). If $rn <$ **Pro**, one dimension of the ith CB is selected randomly, and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + random.\left(x_{j,\max} - x_{j,\min}\right) \tag{12.27}$$

where x_{ij} is the jth variable of the ith CB and $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the jth variable, respectively. In order to protect the structures of CBs, only one dimension is changed.

Level 3: Termination Condition Check

Step 1: After a predefined maximum evaluation number, the optimization process is terminated.

12.6 Numerical Examples

The proposed method is applied to a sample with 12 recorded earthquake accelerograms to obtain the modified accelerogram sets compatible with Eurocode-8 design spectrum of soil classes A and B. The earthquake accelerograms are categorized into two classes according to these soil conditions in order to be consistent with soil classes of target spectrums. Moreover, in each soil class, two sets of accelerograms are selected to illustrate the independency of the proposed method to the selection of the accelerograms. Therefore, the number of ground motions selected for a ground motion set is set to 4, as shown in Table 12.2. All of the records are discretized at 0.01 s with different durations for the strong ground motions. After considering records, three fundamental periods of 0.45, 0.9, and 1.8 s, which represent typical short period, medium period, and long period, respectively, are selected for controlling the requirements of Eurocode-8 in the range of the considered periods [24].

Table 12.2 The sets of earthquake components for spectral matching

Site soil class	Set No.	Name of station	Record ID
Class A	Set 1-A	Anza (Horse Cany)	ANZA/PFT135
		Kocaeli, Turkey	KOCAELI/GBZ000
		Loma Prieta	LOMAP/G01090
	Set 2-A	Whittier Narrows	WHITTIER/A-GRN180
		Northridge	NORTHR/WON185
		San Fernando	SFERN/L09021
Class B	Set 1-B	Cape Mendocino	CAPEMEND/EUR090
		Coyote Lake	COYOTELK/G06320
		Duzce, Turkey	DUZCE/1061-E
	Set 2-B	Friuli, Italy	FRIULI/B-FOC270
		Kern County	KERN/TAF111
		Morgan Hill	MORGAN/G06090

In the optimization process of all the cases, the CBO and ECBO algorithms are used to provide a comparison between these two algorithms. In these cases, the number of agents is set as 30. The maximum number of iterations is also considered as 300. As mentioned before, the well-known penalty approach is used for satisfying the code requirements. Comparisons are made through the error between the target spectrum and modified maximum response spectrums [Eq. (12.21)]. The algorithms are also coded in MATLAB.

Figures 12.5 and 12.6 display the original and modified acceleration and the displacement time histories of the SetA-1, respectively. From these figures it can be seen that the frequency contents of the modified acceleration time histories are different with those of original ones. In this case, comparing the actual and modified accelerograms, it can be seen that the modified acceleration and displacement time histories of the Anza and Kocaeli earthquakes are modified more than the Loma Prieta earthquake. The modified displacement time histories are also realistic due to the use of the baseline correction in the last step of proposed method.

The maximum response spectrums of the SetA-1 original and modified ground motions obtained by both algorithms for three fundamental periods and target spectrum are shown in Fig. 12.7. The 90 % design spectrum (the red dashed lines) and the period ranges of interest (the vertical blue dashed lines) are also displayed as these are the spectral amplitude limits specified by the Eurocode-8. It can be seen the maximum response spectrum of the original accelerograms is far away from the target spectrum, and it falls below the 90 % design spectrum within the period limits as well. While the maximum response spectrums of the modified accelerograms have approached to the target spectrum with modification of these original ground motions using the presented method. Also, the maximum response spectrum does not fall below the 90 % target spectrum within the code-specific period range and zero periods.

Figures 12.7, 12.8, 12.9, and 12.10 show the maximum response spectrums of the modified ground motions obtained by the proposed method for the SetA-2, SetB-1, and SetB-2 as well as three fundamental periods, respectively. Similar results and comparisons can be obtained from these figures. Table 12.3 shows the optimized error obtained by CBO and ECBO for all cases. As shown in this table and Figs. 12.7, 12.8, 12.9, and 12.10, the resulted lower error leads to the response spectrum that is close to the target spectrum. This indicates that more suitable modification of the recorded accelerograms can be achieved using more efficient optimization algorithms. It can be seen that the errors obtained by ECBO are better than those obtained for the CBO algorithm, which indicates the importance of the enhancement of the algorithm in this problem. The errors are also decreased with increase of the fundamental period (Tn); therefore, the recorded accelerograms can easily be modified in high fundamental periods using the proposed method.

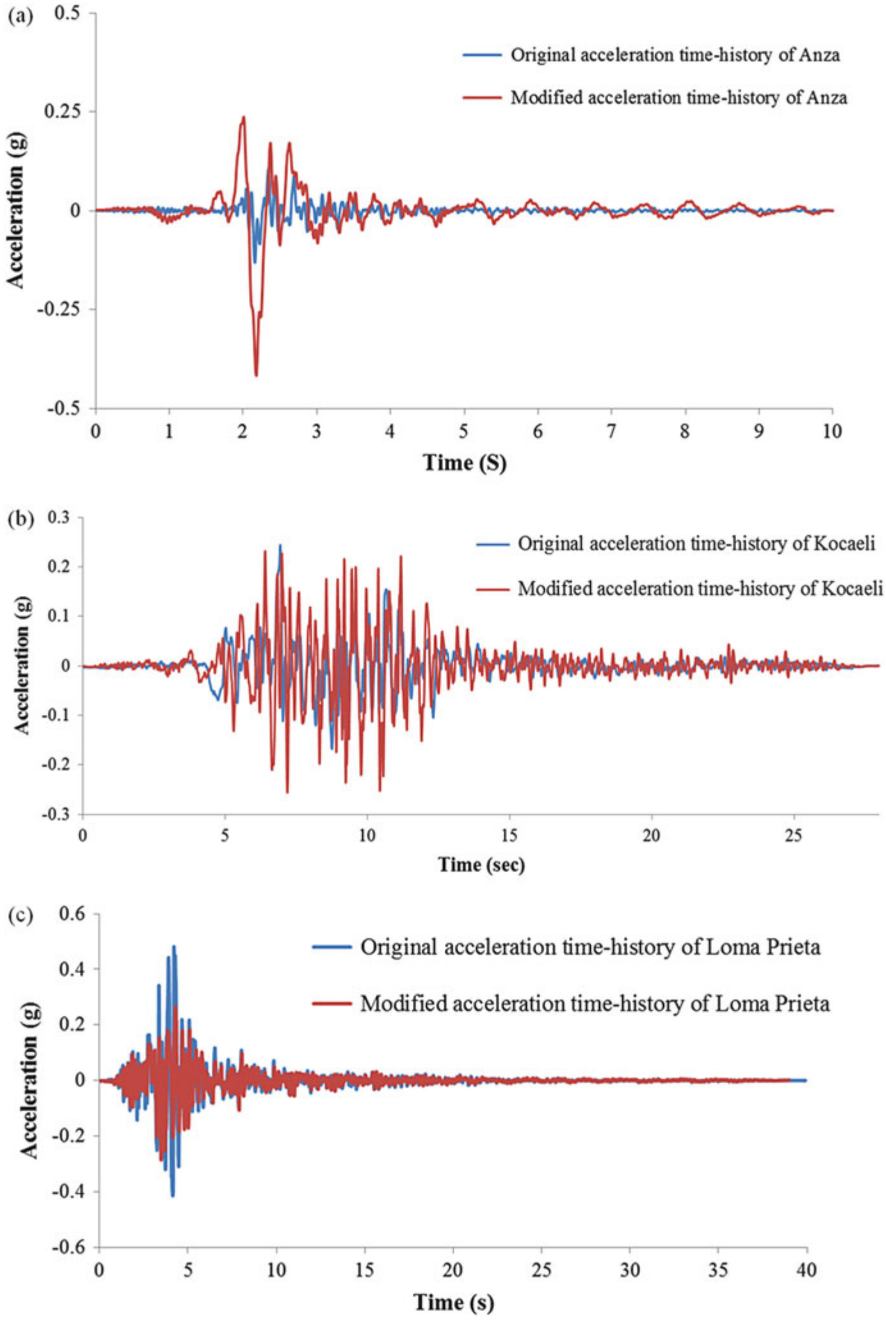


Fig. 12.5 Original and modified acceleration time histories of (**a**) Anza, (**b**) Kocaeli, and (**c**) Loma Prieta

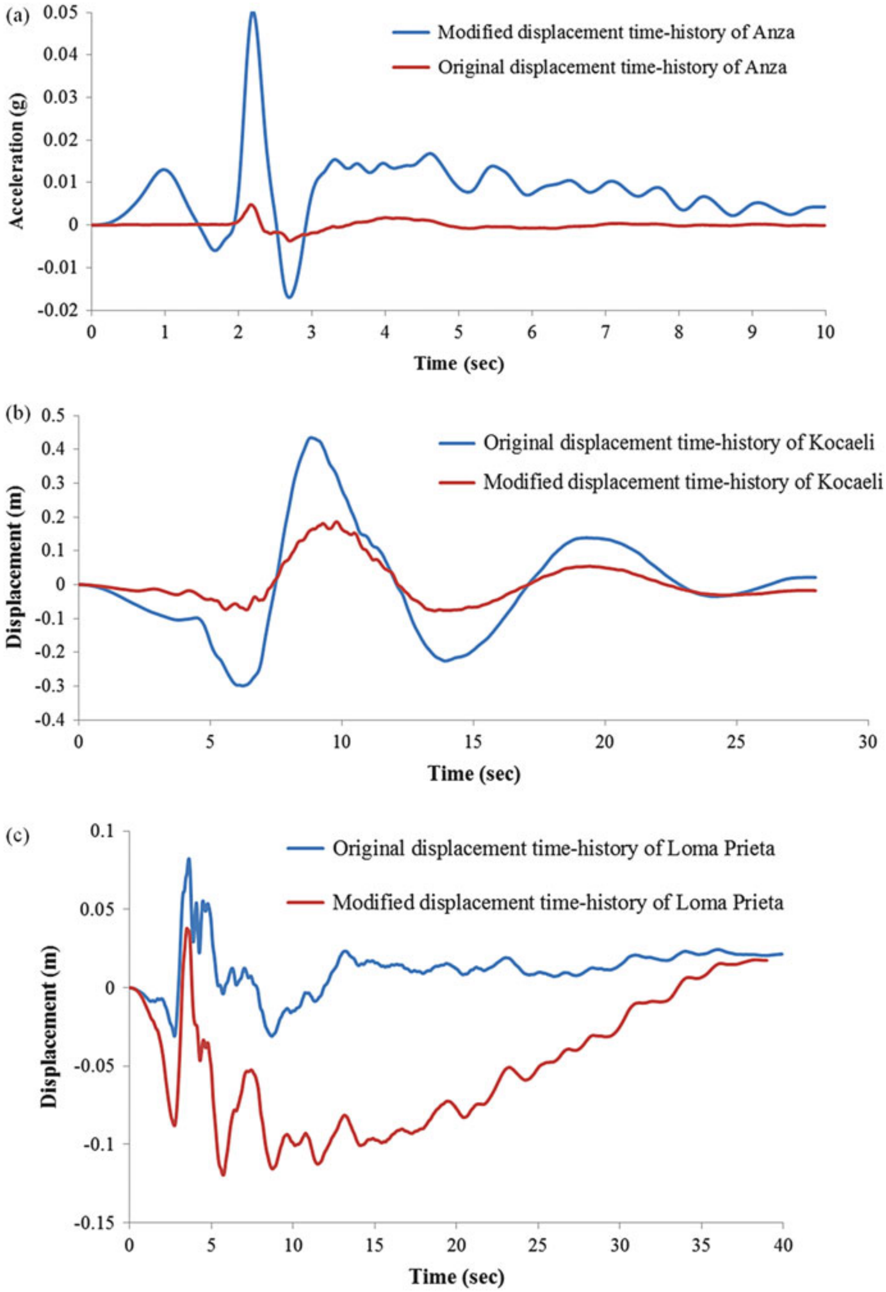


Fig. 12.6 Original and modified displacement time histories of (**a**) Anza, (**b**) Kocaeli, and (**c**) Loma Prieta

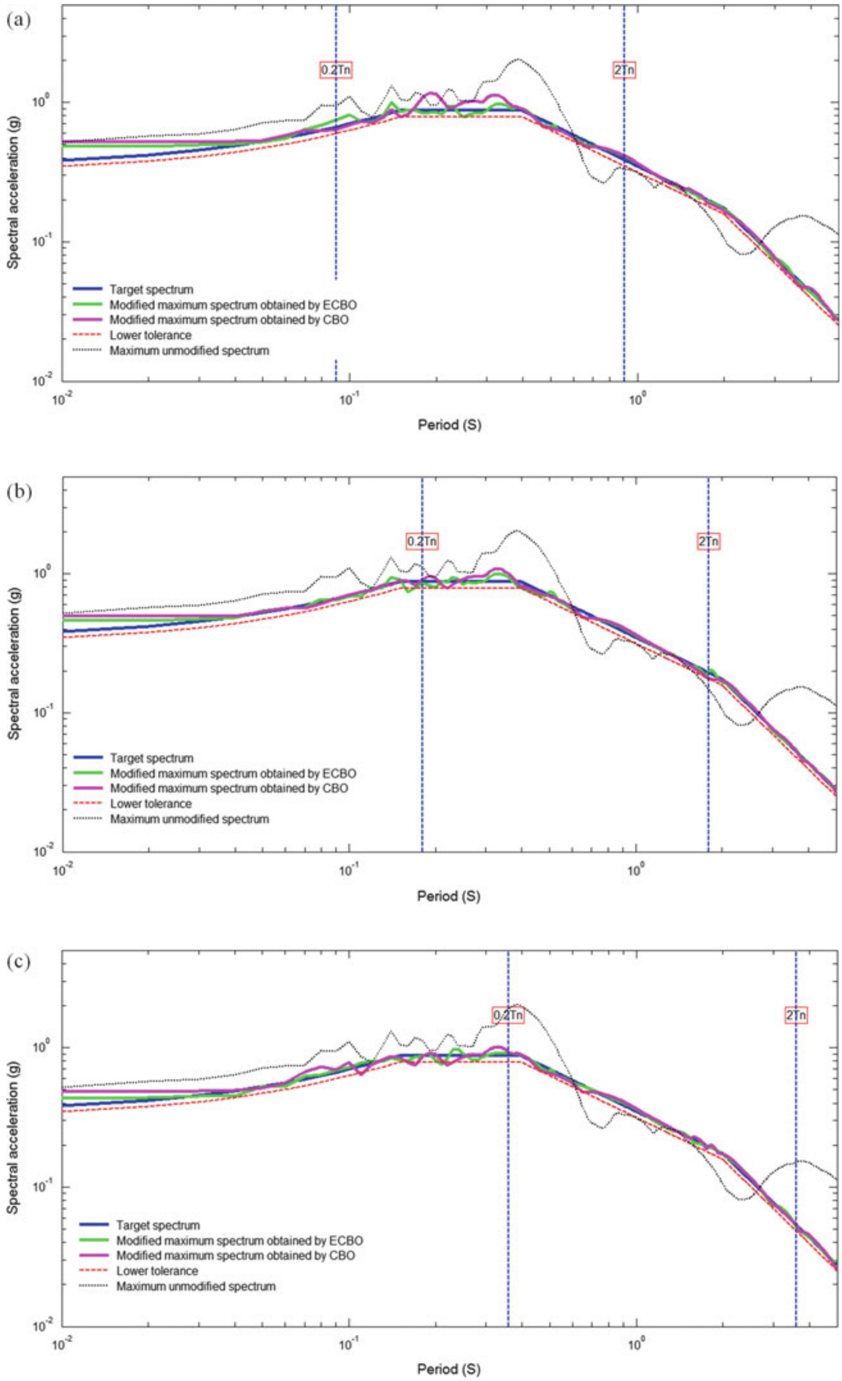


Fig. 12.7 Comparison of various maximum response spectrums of SetA-1 matched with the target spectrum of soil class A for fundamental periods: (**a**) $T_n = 0.45$, (**b**) $T_n = 0.9$, (**c**) $T_n = 1.8$

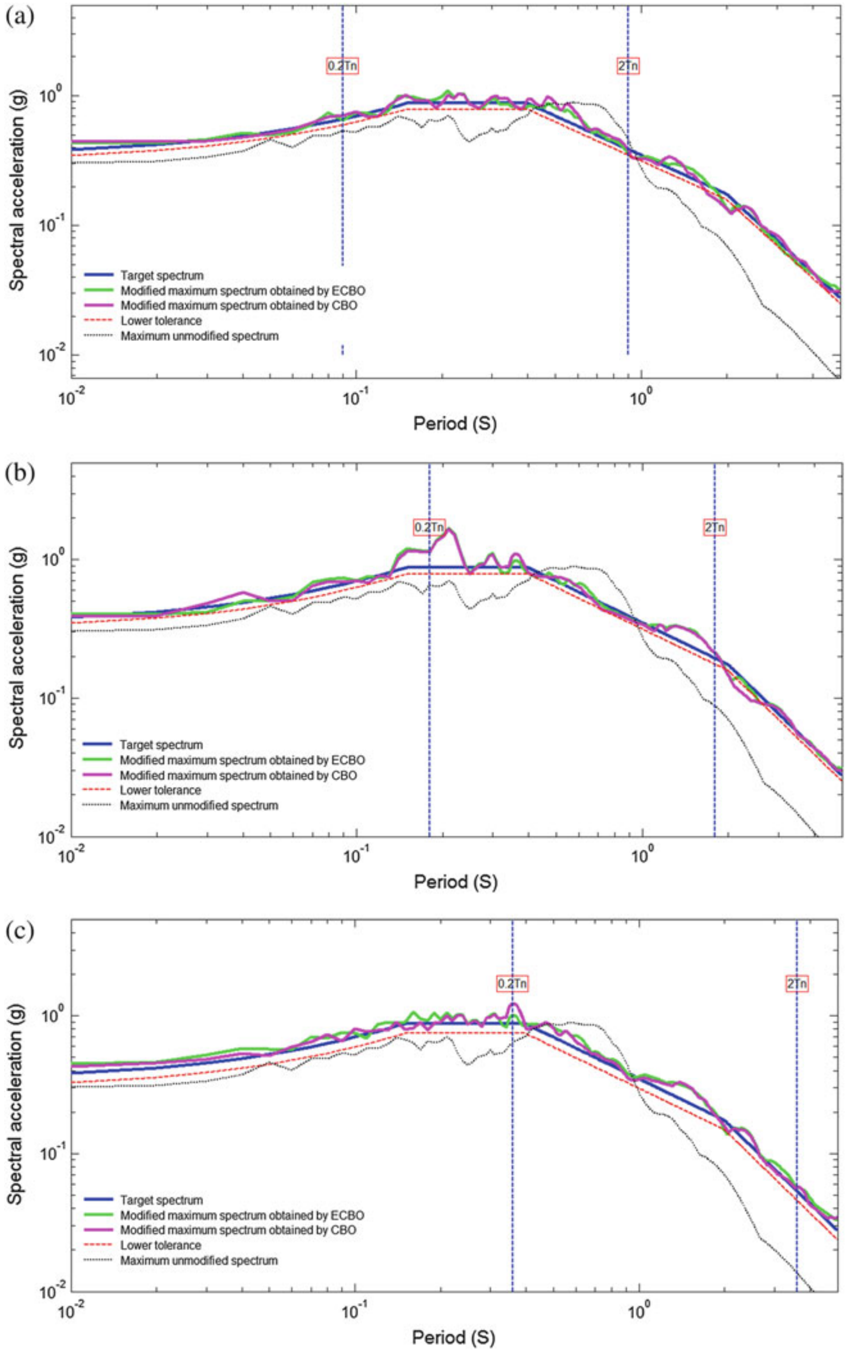


Fig. 12.8 Comparison of various maximum response spectrums of SetA-2 matched with the target spectrum of soil class A for fundamental periods: (**a**) $T_n = 0.45$, (**b**) $T_n = 0.9$, (**c**) $T_n = 1.8$

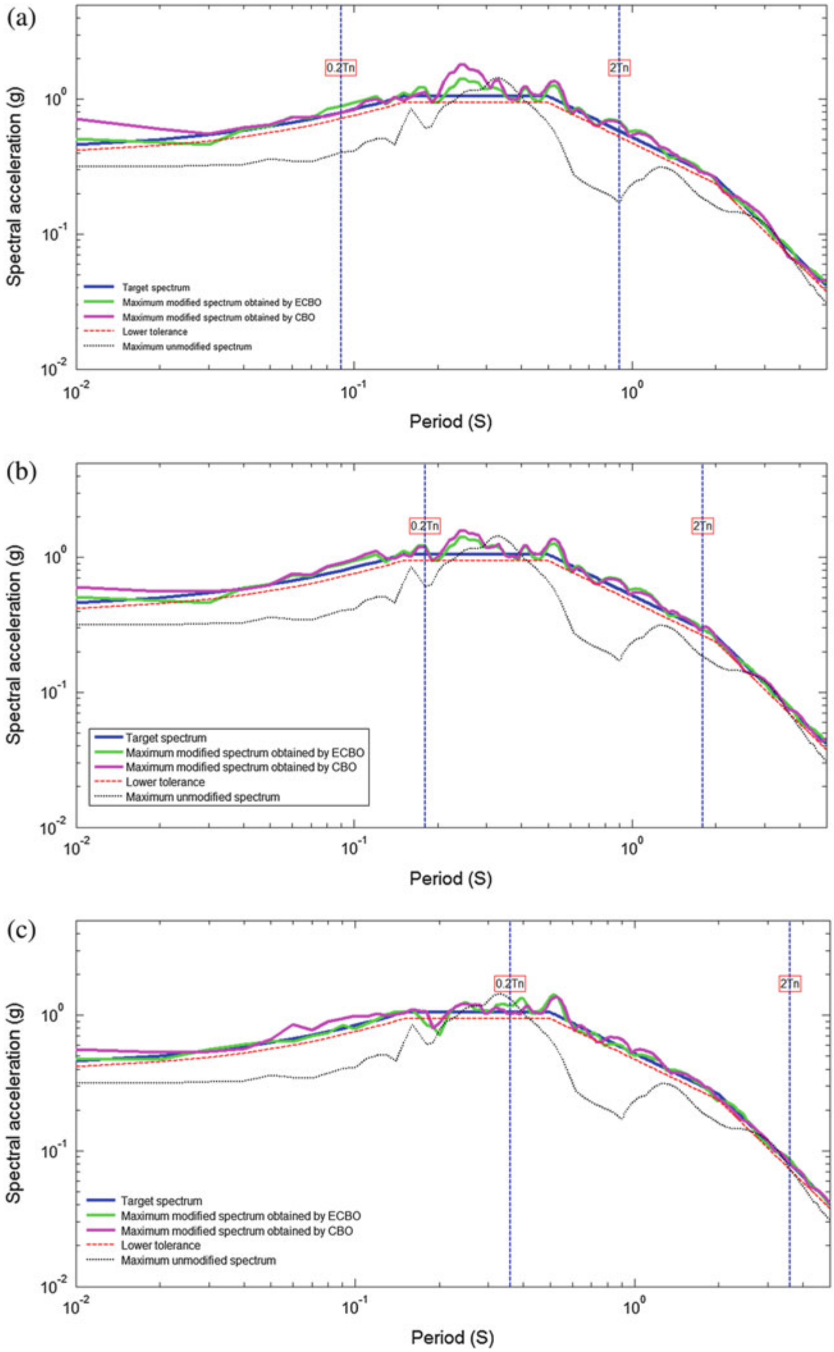


Fig. 12.9 Comparison of various maximum response spectrums of SetB-1 matched with the target spectrum of soil class B for fundamental periods: (**a**) $T_n = 0.45$, (**b**) $T_n = 0.9$, (**c**) $T_n = 1.8$

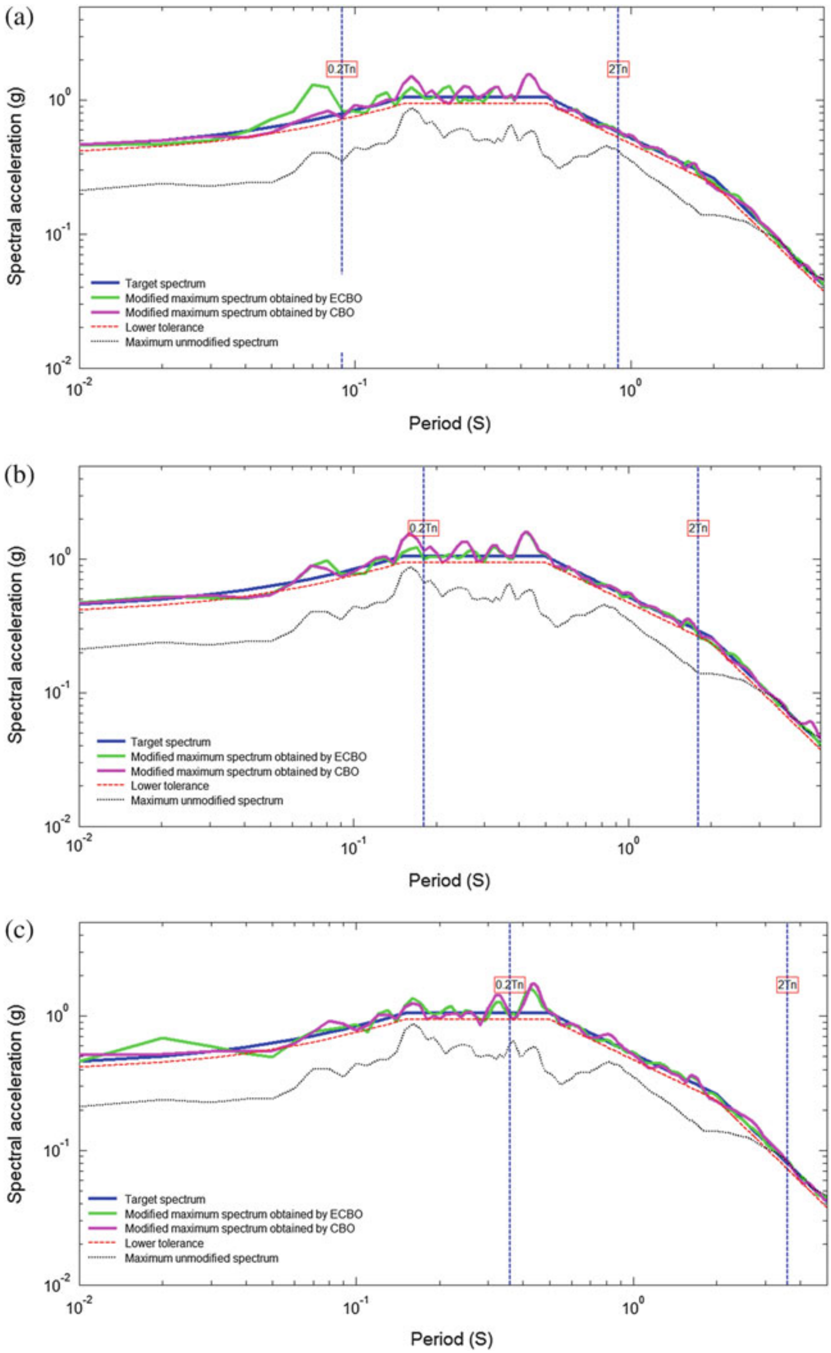


Fig. 12.10 Comparison of various maximum response spectrums of SetB-2 matched with the target spectrum of soil class B for fundamental periods: (**a**) $T_n = 0.45$, (**b**) $T_n = 0.9$, (**c**) $T_n = 1.8$

Table 12.3 The errors obtained for all cases using both algorithms

	Error (%)					
	$T_n = 0.45$ s		$T_n = 0.9$ s		$T_n = 1.8$ s	
Set No.	CBO	ECBO	CBO	ECBO	CBO	ECBO
Set 1-A	5.84	3.43	4.22	3.27	4.42	2.97
Set 2-A	10.32	9.06	12.96	11.27	9.32	8.57
Set 1-B	10.31	8.92	8.12	7.08	7.66	6.45
Set 2-B	7.36	7.23	8.94	6.31	8.78	6.41

12.7 Concluding Remarks

In the present chapter, a new method is proposed for modification/reconstruction of ground motions utilizing a metaheuristic algorithm and wavelet transformation. From the results obtained, the following conclusions can be derived:

(i) The accelerograms are modified in time and frequency domain using the wavelet transformation such that the response spectrums get closer to the target spectrum.
(ii) A common method for solving spectral matching problem is iterative wavelet-based approach, and this procedure has some disadvantages. However, in the proposed method, this problem is formulated as a constrained optimization problem leading to some improvements such as modification of a set of ground motions and handling the manual requirements.
(iii) The Eurocode-8 is utilized for spectra matching requirements and definition of target spectra. In the proposed method, the penalty function is employed to satisfy the corresponding requirements.
(iv) The problem is non-convex and has some local optima because of using the overlapping frequency domain in wavelet transformation having some constraints. Hence the selection of an efficient optimization algorithm is an important issue for handling this problem.
(v) An improved version of the recently developed metaheuristic algorithm called ECBO is used to reduce the error between the response and target spectra. A comparative study of ECBO and CBO algorithms on modifying four sets of accelerograms clearly indicates that the response modified spectrums obtained by ECBO are closer to the target spectrum than those obtained by the CBO.
(vi) It should be noted that the purpose of this chapter has been the introduction of a new method for spectra matching of accelerograms. This goal can also be achieved by considering different target spectrums, manual requirements, optimization algorithms, and transformation functions such as wavelet packet and S transform.

References

1. Kaveh A, Mahdavi VR (2016) A new method for modification of ground motions using wavelet transform and enhanced colliding bodies optimization. Appl Soft Comput 47:357–369
2. Bommer JJ, Acevedo AB (2004) The use of real earthquake accelerograms as input to dynamic analysis. J Earthq Eng 8:43–91
3. Gupta ID, Joshi RG (1993) On synthesizing response spectrum compatible accelerograms. Eur J Earthq Eng 7:25–33
4. Shrikhande M, Gupta VK (1996) On generating ensemble of design spectrum-compatible accelerograms. J Earthq Eng 10:49–56
5. Conte JP, Peng BF (1997) Fully nonstationary analytical earthquake ground motion model. J Eng Mech ASCE 123:15–24
6. Hancock J, Watson-Lamprey J, Abrahamson NA, Bommer JJ, Markatis A, McCoy E, Mendis E (2006) An improved method of matching response spectra of recorded earthquake ground motion using wavelets. J Earthq Eng 10:67–89
7. Mukherjee S, Gupta VK (2002) Wavelet-based generation of spectrum-compatible time-histories. Soil Dyn Earthq Eng 22:799–804
8. Cecini D, Palmeri A (2015) Spectrum-compatible accelerograms with harmonic wavelets. Comput Struct 147:26–35
9. Gao Y, Wu Y, Li D, Zhang N, Zhang F (2014) An improved method for the generating of spectrum-compatible time series using wavelets. Earthq Spectra 30:1467–1485
10. Ghodrati Amiri G, Abdolahi Rad A, Khanmohamadi Hazaveh N (2014) Wavelet-based method for generating nonstationary artificial pulse-like near-fault ground motions. Comput Aided Civil Infrastruct Eng 29:758–770
11. CEN. Eurocode-8 (2003) Design provisions for earthquake resistance of structures. Part 1: general rules, seismic actions and rules for buildings. Brussels
12. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan
13. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26:29–41
14. Eroland OK, Eksin I (2006) New optimization method: Big Bang–Big Crunch. Adv Eng Softw 37:106–111
15. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289
16. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(2013):2592–2612
17. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
18. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
19. Daubechies I (1992) Ten lectures on wavelets. In: CBMS-NSF Conference series in applied mathematics, Montpelier, VT
20. Ghodrati Amiri G, Bagheri A, Seyed Razaghi SA (2009) Generation of multiple earthquake accelerograms compatible with spectrum via the wavelet packet transform and stochastic neural networks. J Earthq Eng 13:899–915
21. Qina X, Fangb B, Tianb S, Tonga X, Wangc Z, Sua L (2014) The existence and uniqueness of solution to wavelet collocation. Appl Math Comput 231:63–72
22. PEER NGA (2014) Strong motion database. http://peer.berkeley.edu/nga/
23. Kaveh A, Mahdavi VR (2015) A hybrid CBO–PSO algorithm for optimal design of truss structures with dynamic constraints. Appl Soft Comput 34:260–273
24. Ye Y, Chen Z, Zhu H (2014) Proposed strategy for the application of the modified harmony search algorithm to code-based selection and scaling of ground motions. J Comput Civil Eng ASCE 28:04014019

Chapter 13
Bandwidth, Profile, and Wavefront Optimization Using CBO, ECBO, and TWO Algorithms

13.1 Introduction

In this chapter three recently developed metaheuristic optimization algorithms, known as colliding bodies optimization (CBO), enhanced colliding bodies optimization (ECBO), and tug of war optimization (TWO), are utilized for optimum nodal ordering to reduce bandwidth, profile, and wavefront of sparse matrices. The bandwidth, profile, and wavefront of some graph matrices, which have equivalent patterns to structural matrices, are minimized using these methods. Comparison of the achieved results with those of some existing approaches shows the robustness of these three new metaheuristic algorithms for bandwidth, profile, and wavefront optimization [1].

The solution of sparse systems of simultaneous equations is required by the analysis of many problems in structural engineering. Such non-singular systems of linear algebraic equations are in the form $Ax = b$ arising from finite element method. These types of equations commonly involve a positive definite, symmetric, and sparse coefficient matrix A. For large structures a great deal of the computational effort and memory are dedicated to the solution of these equations. Hence, some suitable specified patterns for the coefficients of the corresponding equations have been provided, like banded form, profile form, and partitioned form. These patterns are often achieved by nodal ordering of the corresponding models.

In finite element model (FEM) analysis, for the case of one degree of freedom per node, performing nodal ordering is equivalent to reordering the equations. In a more general problem with m degrees of freedom per node, there are m-coupled equations produced for each node. In this case re-sequencing is usually performed on the nodal numbering of the graph models, to reduce the bandwidth, profile, or wavefront, because the size of these problems is m fold smaller than those for m degree of freedom numbering. In this chapter, the mathematical model of a FEM is considered as an element clique graph, and nodal ordering is carried out to

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_13

decrease the bandwidth, profile, or wavefront of the corresponding matrices (Kaveh [2–4]).

There is an important rule for nodal numbering in the solution of sparse systems. It can be attained by permuting the rows and columns of a matrix by proper renumbering of the nodes of the associated graph. Three important subjects in nodal ordering are bandwidth, profile, and wavefront optimization. In fact, for sparse matrices the size can be measured by the bandwidth or profile or wavefront of such matrices. These problems have created considerable interest during recent years because it has practical relevance to a significant range of global optimization applications. Since the nature of the problem of nodal numbering is NP complete (Papademetrious [5]), many approximate approaches and heuristics are proposed, examples of which can be found in Gibbs et al. [6], Cuthill and McKee [7], Kaveh [2], Bernardes and Oliveira [8], and King [9].

Metaheuristic techniques are the recent generation of the optimization methods to solve complex problems. These approaches explore the feasible region based on randomization and some specified rules through a group of search agents. The rules are usually inspired from laws of natural phenomena (Kaveh [10]).

As a newly developed type of metaheuristic algorithm, CBO is introduced and employed to structural problems by Kaveh and Mahdavi [11]. The CBO is a multi-agent approach which is inspired by a collision between two objects in one dimension. Each agent is considered as a body with a specified mass and velocity. A collision occurs between pairs of bodies, and the new positions of the colliding bodies are updated based on the collision laws. The enhanced colliding bodies optimization is introduced by Kaveh and Ilchi Ghazaan [12], and it employs memory to save some best-so-far positions to improve the CBO performance without increasing the computer execution time. This algorithm uses a mechanism to escape from local optima.

TWO is a multi-agent metaheuristic approach, which is introduced by Kaveh and Zolghadr [13]. This method models each candidate solution as a team engaged in a series of tug of war competitions. The weight of the teams is defined based on the quality of the corresponding solutions, and the amount of pulling force which a team can exert on the rope is assumed to be proportional to its weight. Naturally, the opposite team will have to maintain at least the same amount of force in order to sustain its grip of the rope. The lighter team accelerates toward the heavier team and this forms the convergence operator of TWO algorithm. The approach improves the quality of the solutions iteratively by maintaining a proper exploration/exploitation balance using the described convergence operator.

The rest of this chapter is organized as follows: In Sect. 13.2 some definitions from graph theory, bandwidth, profile, and wavefront are stated. The CBO, ECBO, and TWO algorithms are briefly presented in Sect. 13.3. In order to show the performance of these methods on bandwidth, profile, and wavefront reduction, Sects. 13.4 and 13.5 contain the results of four examples and the corresponding discussions, respectively. The final section concludes the chapter.

13.2 Problem Definition

13.2.1 Definitions from Graph Theory

Let $G(N,M)$ be a graph with members set $M(|M| = m)$ and nodes set $N(|N| = n)$ with a relation of incidence. The *degree* of a node is the number of members incident with the node, and the *1-weighted degree* of a node is defined as the sum of the degrees of its adjacent nodes. A *spanning tree* is a tree containing all the nodes of S. A *shortest route tree* (SRT_{n0}) rooted from a specified node (starting node) n_0 is a spanning tree for which the distance between every node n_j of S and n_0 is minimum, where the *distance* between two nodes is defined as the number of members in the shortest path between these nodes. A *contour* $C_k^{n_0}$ contains all the nodes with equidistance k from node n_0. The number of contours of an SRT_{n0} is known as its *depth*, denoted by $d(\mathrm{SRT}_{n0})$, and the highest number of nodes in a contour specifies the *width* of the SRT_{n0}. A labeling As of G assigns the set of integers $\{1, 2, 3, \ldots, n\}$ to the nodes of graph G. $As(i)$ is the label or the integer assigned to node i and each node has a different label. The bandwidth of node i for this assignment, $bw(i)$, is the maximum difference of $As(i)$ and $As(j)$, where $As(j)$ is the label of nodes adjacent to node i or the number assigned to its adjacent nodes (Kaveh [3]). That is

$$bw_{As}(i) = \max\{|As(i) - As(j)| \ : j \in N(i)\} \tag{13.1}$$

where $N(i)$ is the set of adjacent nodes of node i. The bandwidth of the graph G with respect to the assignment $As(i)$ is then

$$BW_{As}(G) = \max\{bw(i) \ : i \in G\} \tag{13.2}$$

The minimum value of BW over all possible assignments is the bandwidth of the graph:

$$BW(G) = \min\{BW_{As}(G) \ : \forall As(i)\} \tag{13.3}$$

The profile of the $N \times N$ matrix related to graph G, for the assignment $As(i)$, is defined as

$$P_{As} = \sum_{i=1}^{N} b_i \tag{13.4}$$

where the row bandwidth, b_i, for row i is defined as the number of inclusive entries from the first nonzero element in the row to the $(i+1)$th entry for this assignment. The efficiency of any given ordering for the profile solution scheme is related to the number of active equations during each step of the factorization process. Formally, row j is defined to be active during elimination of column i if $j \geq i$, and there exists $a_{ik} = 0$ with $k \leq i$. Hence, at the ith stage of the factorization, the number of active equations is the number of rows of profile that intersect column *i*, which is ignored if those rows already eliminated. Let f_i denote the number of equations that are active during the elimination of the variable x_i. It follows from the symmetric structures of the matrix that

$$P_{As} = \sum_{i=1}^{N} f_i = \sum_{i=1}^{N} b_i \tag{13.5}$$

where f_i is commonly known as the frontwidth or wavefront. Assuming that N and the average value of f_i are significantly large, it can be shown that a complete profile or front factorization needs approximately $O(NF_{\mathrm{rms}}^2)$ operations, where F_{rms} is the root–mean square frontwidth, defined as

$$F_{\mathrm{rms}} = \left[\frac{1}{N}\sum_{i=1}^{N} f_i^2\right]^{0.5} \tag{13.6}$$

In the bandwidth reduction problem, one searches an assignment $As(i)$ which minimizes $BW(G)$. Such an assignment moves all the nonzero elements of the matrix onto a band, which is as close as possible to the main diagonal (Kaveh and Sharafi [14,15] and Kaveh and Bijari [16]). In this chapter, for bandwidth minimization, one should find a suitable assignment for nodal ordering of a graph to reduce the bandwidth of the associated matrix employing PSO, CBO, ECBO, and TWO algorithms. The algorithms for bandwidth reduction are based on reordering or assigning new integers to the nodes of the graph to achieve an optimal bandwidth.

Each permutation of columns and rows of an $N \times N$ sparse matrix associated to graph G leads to a new reordering called the assigning set. If the primary ordering of the graph is $\{1, 2, 3, \ldots, n\}$, each permutation of this list will be a new assigned set. The aim is to find the optimal assigning list to obtain the best bandwidth.

For the purpose of finding an optimal nodal ordering in the profile and frontwidth reduction problems, it is tried to assign the set of integers $\{1, 2, 3, \ldots, n\}$ to the nodes of G using a priority function, and the coefficients of the priority function are found employing PSO, CBO, ECBO, and TWO algorithms.

13.2.2 An Algorithm Based on Priority Queue for Profile and Wavefront Minimization

The nodal numbering in a priority queue is carried out through the assignment of status, based on the numbering approach of King [9]. King's method was generalized by Sloan [17], by introducing a priority queue which controls the order to be followed in the numbering of the nodes. This algorithm comprises of two phases:

Phase 1: Selecting a pair of pseudo-peripheral nodes
Phase 2: Nodal numbering

Phase 1 selects a pair of nodes as starting and ending nodes according to the following steps:

Step 1: Choose an arbitrary node s of minimum degree.
Step 2: Generate an $\mathrm{SRT}_s = \{C_1^s, C_2^s, \ldots, C_d^s\}$ rooted from s. Let S be the list of the nodes of C_d^s, which is stored in the order of increasing degree.
Step 3: Decompose S into subsets S_j of cardinality $|S_j|$, $j = 1, 2, \ldots, \Delta$ where Δ is the maximum degree of any node of S, such that all nodes of S_j have degree j. Generate an SRT from each node y of S, for the first $1 \le m_j \le \Delta$. If $d(\mathrm{SRT}_y) > d(\mathrm{SRT}_s)$, then set $s = y$ and go to Step 2.
Step 4: Let e be the root of the longest SRT that has the smallest width. When the algorithm terminates, s and e are the end points of a pseudo-diameter.

Phase 2 reorders the nodes of an element clique graph and ensures that the position of a node in this reordering phase follows a priority rule according to the following steps:

Step 1: Find the status of all nodes. A node can be in the following four states as shown in Fig. 13.1. A node which has been assigned a new label is defined as *post-active*. Nodes which are adjacent to a post-active node, but do not have a post-active status, are said to be *active*. Each node which is adjacent to an active node, but is not post-active or active, is said *pre-active*. The nodes which are not post-active, active, or pre-active are said to be *inactive*.
Step 2: Prepare the list of the candidate nodes for labeling in the next step, which consists of active and pre-active nodes.
Step 3: Calculate the priority number for all the candidate nodes. For node i the number is obtained from the following relationship:

$$P_i = W_1 \times \delta_i - W_2 \times D_i \tag{13.7}$$

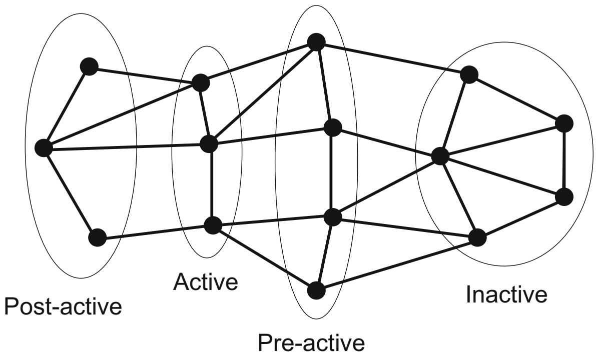


Fig. 13.1 Different status of nodes

where W_1 and W_2 are integer weights (suggested as $W_1 = 1$ and $W_2 = 2$ in the original Sloan's algorithm), δ_i is the distance between each node i from the end node, and D_i is the incremental degree of node i which is defined as

$$D_i = d_i - c_i + k_i \tag{13.8}$$

where d_i is the degree of node i, c_i is the number of active and post-active nodes adjacent to node i, and k_i is zero if the node i is active or post-active and unity otherwise.

Step 4: Select the node with the highest priority among the candidate nodes and label it.

Step 5: Repeat Steps 1–4 until all the nodes are labeled.

In Eq. (13.7) if $W_1 = 0$ and $W_2 = 1$, the node-labeling algorithm will become similar to the one proposed by King.

13.2.3 The Priority Function with New Integer Weights

As can be seen from Eq. (13.7), Sloan's algorithm employs a linear priority function of two graph parameters and the weights determine the importance of each parameter. In Sloan's algorithm the pair $W_1 = 1$ and $W_2 = 2$ has been recommended for the weights. However, some research results (Kaveh and Roosta [18], Rahimi Bondarabadi and Kaveh [19]) show that for some problems, there are advantages in using other values.

In general, the priority can be determined by a general linear function of vectors of graph parameters and their coefficients as

$$P_i = \sum_{i=1}^{L} W_i \times C_i \tag{13.9}$$

where $C_i\ (i = 1, 2, \ldots, L)$ are the normalized Ritz vectors indicating the graph parameters, and $W_i\ (i = 1, 2, \ldots, L)$ are the coefficients of the Ritz vectors (Ritz coordinates) that are unknowns. That is, one can employ L characteristics of a graph to define the priority function and find the coefficients which can guide the algorithm to select an optimal profile and wavefront.

Sloan's algorithm employs $L = 2$ characteristics of the graph model. Here, we find the best sets of coefficients for the priority function with $L = 2$ and 5. These sets of coefficients (integer weights) are found by optimizing the results utilizing PSO, CBO, ECBO, and TWO algorithms.

In the first case, $L = 2$ method is presented. The vectors of graph properties are taken similar to those of Sloan's algorithm. In the second case, $L = 5$ method is presented using five vectors $C_i\ (i = 1, 2, \ldots, 5)$ as follows:

C_1 Degrees of the nodes
C_2 Node distances from the end node
C_3 Node distances from the starting node
C_4 The 1-weighted degree
C_5 The width of an SRT rooted from the starting node

Once the graph parameter vectors are formed, their coefficients can be obtained using PSO, CBO, ECBO, and TWO algorithms.

13.3 Metaheuristic Algorithms

This section includes the colliding bodies optimization algorithm, its enhanced version, and tug of war optimization algorithm. First, a brief description of standard CBO is provided. The ECBO is presented (Kaveh and Ilchi Ghazaan [12]), and then a new algorithm called TWO is stated.

13.3.1 Colliding Bodies Optimization

Collision is a natural phenomenon, and the colliding bodies optimization algorithm was developed based on this occurrence by Kaveh and Mahdavi [11]. In this method, one object collides with another and they move toward a minimum energy level. The CBO utilizes simple formulation, does not require any internal parameters, and does not use memory for saving the best solutions so far.

This technique is a population-based metaheuristic algorithm. Each solution candidate X_i is considered as a colliding body (CB), and it has a specified mass defined as

$$m_k = \frac{\frac{1}{fit(k)}}{\frac{1}{\sum_{i=1}^{n} \frac{1}{fit(i)}}} \qquad k = 1,\ 2,\ \ldots,\ n \tag{13.10}$$

where $fit(i)$ represents the objective function value of the ith CB and n is the number of colliding bodies. In order to select pairs of objects for collision, CBs are sorted according to their mass in a decreasing order and they are divided into two equal groups: (i) stationary group and (ii) moving group. Moving objects collide to stationary objects to improve their positions and push stationary objects toward better positions (see Fig. 13.2).

The velocity of the stationary bodies before collision is zero, so

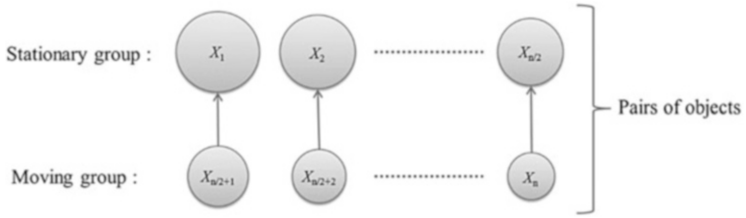


Fig. 13.2 The pairs of CBs for collision

$$v_i = 0, \qquad i = 1\ ,\ 2\ ,\ \ldots\ ,\ \frac{n}{2} \tag{13.11}$$

The velocity of each moving body before collision is

$$v_i = x_{i-\frac{n}{2}} - x_i\ , \qquad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{13.12}$$

The velocity of each stationary CB after the collision (v_i') is specified by

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}}, \qquad i = 1\ ,\ \ldots\ ,\ \frac{n}{2} \tag{13.13}$$

The velocity of each moving CB after the collision (v_i') is defined by

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}}, \qquad i = \frac{n}{2} + 1\ ,\ \frac{n}{2} + 2\ ,\ \ldots\ ,\ n \tag{13.14}$$

Here ε is the coefficient of restitution (COR) that decreases linearly from unit to zero.

Thus, it is stated as

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{13.15}$$

where *iter* is the current iteration number and $iter_{max}$ is the total number of iterations for optimization process.

New positions of CBs are updated according to their velocities after the collision and the positions of stationary CBs. Therefore, the new position of each stationary CB is

$$x_i^{\text{new}} = x_i + rand \circ v_i', \quad i = 1, \ \ldots, \frac{n}{2} \tag{13.16}$$

where x_i^{new}, x_i, and v_i' are the new position, previous position, and the velocity after the collision of the ith CB, respectively, *rand* is a random vector uniformly distributed in the range of [−1, 1], and the sign "∘" denotes an element-by-element multiplication. The new position of each moving CB is calculated by

$$x_i^{\text{new}} = x_{i-\frac{n}{2}} + rand \circ v_i', \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ \ldots, n \tag{13.17}$$

The process of optimization is terminated if the maximum number of analyses has been evaluated. For further details, the reader may refer to Kaveh and Mahdavi [11].

13.3.2 Enhanced Colliding Bodies Optimization

A modified version of the CBO is enhanced colliding bodies optimization, which improves the CBO to get more reliable solutions. The introduction of memory can increase the convergence speed of ECBO compared to standard CBO. Furthermore, changing some components of colliding bodies will help ECBO to escape from local optima. The steps of ECBO are as follows:

Step 1: Initialization

The initial positions of all CBs are determined randomly in an m-dimensional search space.

$$x_i^0 = x_{\min} + random \circ (x_{\max} - x_{\min}), \quad i = 1, 2, \ \ldots, n \tag{13.18}$$

where x_i^0 is the initial solution vector of the ith CB. Here, $x_{\min}$ and $x_{\max}$ are the bounds of design variables, *random* is a random vector in which each component is in the interval [0, 1], and n is the number of CBs.

Step 2: Defining mass

The value of mass for each CB is evaluated according to Eq. (13.10).

Step 3: Saving

Considering a memory which saves some historically best CB vectors and their related mass and objective function values can make the algorithm performance better without increasing the computational cost (Kaveh and Ilchi Ghazaan [12]). Here, a colliding memory (CM) is utilized to save a number of the best-so-far solutions. Therefore, in this step, the solution vectors saved in CM are added to the population, and the same numbers of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: Creating groups
CBs are divided into two equal groups: (i) stationary group and (ii) moving group. The pairs of CBs are defined according to Fig. 13.2.

Step 5: Criteria before the collision
The velocity of stationary bodies before collision is zero [Eq. (13.11)]. Moving objects move toward stationary objects and their velocities before collision are calculated by Eq. (13.12).

Step 6: Criteria after the collision
The velocities of stationary and moving bodies are evaluated using Eqs. (13.13) and (13.14), respectively.

Step 7: Updating CBs
The new position of each CB is calculated by Eqs. (13.16) and (13.17).

Step 8: Escape from local optima
Metaheuristic algorithms should have the ability to escape from the trap when agents get close to a local optimum. In ECBO, a parameter like *Pro* within (0, 1) is introduced which specifies whether a component of each CB must be changed or not. For each colliding body, *Pro* is compared with *rn* (i = 1, 2, . . ., n) which is a random number uniformly distributed within (0, 1). If $rni < Pro$, one dimension of the *i*th CB is selected randomly, and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + random.\left(x_{j,\max} - x_{j,\min}\right) \tag{13.19}$$

where x_{ij} is the *j*th variable of the *i*th CB and $x_{j,\min}$ and $x_{j,\max}$, respectively, are the lower and upper bounds of the *j*th variable. In order to protect the structures of CBs, only one dimension is changed. This mechanism provides opportunities for the CBs to move all over the search space, thus providing better diversity.

Step 9: Terminating condition check

The optimization process is terminated after a fixed number of iterations. If this criterion is not satisfied, go to Step 2 for a new round of iteration (Kaveh and Ilchi Ghazaan [12]).

13.3.3 Tug of War Optimization Algorithm

TWO is a multi-agent metaheuristic approach, which is introduced by Kaveh and Zolghadr [13]. This method models each candidate solution $X_i = \left\{x_{i,j}\right\}$ as a team engaged in a series of tug of war competitions. The weight of the teams is defined based on the quality of the corresponding solutions, and the amount of pulling force which a team can exert on the rope is assumed to be proportional to its weight. Naturally, the opposite team will have to maintain at least the same amount of force in order to sustain its grip of the rope. The lighter team accelerates toward the heavier team and this forms the convergence operator of TWO algorithm. The approach improves the quality of the solutions iteratively by maintaining a proper exploration/exploitation balance using the described convergence operator. The steps of TWO can be stated as follows:

Step 1: Initialization

A population of N initial solutions is generated randomly:

$$x_{ij}^0 = x_{j,\min} + rand\left(x_{j,\max} - x_{j,\min}\right) \quad j = 1, 2, \ldots, n \tag{13.20}$$

where x_{ij}^0 is the initial value of the jth variable of the ith candidate solution; $x_{j,\max}$ and $x_{j,\min}$ are the maximum and minimum permissible values for the jth variable, respectively; rand is a random number from a uniform distribution in the interval [0, 1]; and n is the number of variables.

Step 2: Evaluation and weight assignment

The objective function values for the candidate solutions are evaluated and sorted. The best solution so far and its objective function value are saved. Each solution is considered as a team with the following weight:

$$W_i = 0.9\left(\frac{fit(i) - fit_{worst}}{fit_{best} - fit_{worst}}\right) + 0.1 \qquad i = 1, 2, \ldots, N \tag{13.21}$$

where $fit(i)$ is the fitness value for the ith particle. The fitness value can be considered as the penalized objective function value for constrained problems; fit_{best} and fit_{worst} are the fitness values for the best and worst candidate solutions of the current iteration. According to Eq. (13.21), the weights of the teams range between 0.1 and 1.

Step 3: Competition and displacement

In TWO each team competes against all the others one at a time to move to its new position in every iteration. The pulling force exerted by a team is assumed to be equal to its static friction force ($W\mu_s$). Hence, the pulling force between teams i and j ($F_{p,ij}$) can be determined as $max\{W_i\mu_s, W_j\mu_s\}$. Such a definition keeps the position of the heavier team unaltered.

The resultant force affecting team i due to its interaction with heavier team j in the kth iteration can then be calculated as follows:

$$F_{r,ij}^k = F_{p,ij}^k - W_i^k\mu_k \tag{13.22}$$

where $F_{p,ij}^k$ is the pulling force between teams i and j in the kth iteration and μ_k is the coefficient of kinematic friction.

$$a_{ij}^k = \left(\frac{F_{r,ij}^k}{W_i^k\mu_k}\right)g_{ij}^k \tag{13.23}$$

in which a_{ij}^k is the acceleration of team i toward team j in the kth iteration and g_{ij}^k is the gravitational acceleration constant which is defined as

$$g_{ij}^k = X_j^k - X_i^k \tag{13.24}$$

where X_j^k and X_i^k are the position vectors for candidate solutions j and i in the kth iteration. Finally, the displacement of team i after competing with team j can be derived as

$$\Delta X_{ij}^k = \frac{1}{2} a_{ij}^k \Delta t^2 + \alpha^k (X_{\max} - X_{\min}) \circ (-0.5 + rand(1, n)) \tag{13.25}$$

The second term of Eq. (13.25) induces randomness into the algorithm. This term can be interpreted as the random portion of the search space travel by team i before it stops after the applied force is removed. Here, α is a constant chosen from the interval [0,1]; $X_{\max}$ and $X_{\min}$ are the vectors containing the upper and lower bounds of the permissible ranges of the design variables, respectively; $\circ$ denotes element-by-element multiplication; and $rand(1, n)$ is a vector of uniformly distributed random numbers.

It should be noted that when team j is lighter than team i, the corresponding displacement of team i will be equal to zero (i.e., ΔX_{ij}^k). Finally, the total displacement of team i in iteration k is equal to

$$\Delta X_i^k = \sum_{j=1}^{N} \Delta X_{ij}^k \tag{13.26}$$

The new position of team i at the end of kth iteration is then calculated as

$$X_i^{k+1} = X_i^k + \Delta X_i^k \tag{13.27}$$

Step 4: Side constraint handling

It is possible for the candidate solutions to leave the search space, and it is important to deal with such solutions properly. This is especially the case for the solutions corresponding to lighter teams for which the values of ΔX are usually bigger. Different strategies might be used in order to solve this problem. For example, such candidate solutions can be simply brought back to their previous permissible position (flyback strategy) or they can be regenerated randomly. In this chapter a new strategy is introduced and incorporated using the global best solution. If the jth variable of any candidate solution, X_i violates the side constraints in the kth iteration, the new value is defined as

$$x_{ij}^k = GB_j + \left(\frac{randn}{k}\right)\left(GB_j - x_{ij}^{k-1}\right) \tag{13.28}$$

where GB_j is the jth variable of the global best solution (i.e., the best solution so far) and $randn$ is the random number drawn from a standard normal distribution.

There is a very slight possibility for the newly generated variable being still outside the search space. In such cases a flyback strategy is used.

Step 5: Termination

Steps 2 through 5 are repeated until a termination criterion is satisfied (Kaveh and Zolghadr [13]).

13.4 Numerical Examples

In this section, four finite element meshes (FEMs) are considered. Element clique graph is a type of graph model that is employed for transferring topological properties of finite element models into connectivity properties of graphs (Kaveh [4]). This graph model has the same nodes as those of corresponding finite element model, and the nodes of each element are cliqued, avoiding the multiple members for the whole graph. The first example is a Z-shaped finite element model for shear wall. An element clique graph of a rectangular FEM with four openings is considered in the second example. The third example is the grid model of a fan with one-dimensional beam elements, and an H-shaped finite element grid is presented in the fourth example. The well-known standard PSO algorithm; two new algorithms, namely, the colliding bodies optimization and enhanced colliding bodies optimization; and a recently developed method called tug of war optimization are applied for all of three bandwidth, profile, and wavefront minimizing problems. The results in bandwidth reduction problem are then compared to those of the four-step algorithm of Kaveh [2] and those of Kaveh and Sharafi [14,15] in Table 13.1. The results obtained in profile and wavefront minimizing problems with $L = 2$ and 5 methods are compared to those of the Sloan and King's algorithms in Tables 13.2 and 13.3, respectively.

13.4.1 Example 1: The FEM of a Shear Wall

The FEM of a shear wall with 550 nodes is considered. The element clique graph of this model is shown in Fig. 13.3. The performance of the abovementioned

Table 13.1 Comparison of the results of different algorithms for bandwidth reduction

						Kaveh and Sharafi [14,15]		
	4-step algorithm	PSO	CBO	ECBO	TWO	4-step	ACO	CSS
Example 1	28	28	28	28	28	29	29	–
Example 2	29	29	29	29	29	–	–	–
Example 3	18	18	18	18	18	23	23	21
Example 4	57	57	57	57	57	66	60	58

Table 13.2 Comparison of the results of different algorithms for profile reduction

Example	Algorithm		W_1	W_2	W_3	W_4	W_5	Profile
Example 1	Sloan		1	2				10,530
	King		0	1				10,974
	PSO	$L=2$	0.0007	0.4852				10,501
		$L=5$	−0.0863	0.4638	−0.3677	0.0034	0.9191	9280
	CBO	$L=2$	0.0043	0.4001				10,501
		$L=5$	0.2191	0.9551	−0.6962	−0.0390	−0.3207	9242
	ECBO	$L=2$	0.0001	0.9881				10,501
		$L=5$	−0.0255	0.8883	−0.6256	−0.0110	−0.9183	9237
	TWO	$L=2$	0.0129	0.7645				10,501
		$L=5$	−0.0404	0.8801	−0.5940	−0.0063	0.7426	9237
Example 2	Sloan		1	2				18,719
	King		0	1				18,839
	PSO	$L=2$	0.6645	0.9066				18,690
		$L=5$	0.1056	0.8858	−0.4835	−0.0152	0.0524	17,136
	CBO	$L=2$	0.1899	0.2566				18,689
		$L=5$	−0.3332	−0.7097	0.9412	0.0507	0.9747	17,122
	ECBO	$L=2$	0.6665	0.9228				18,581
		$L=5$	−0.0458	0.7835	−0.6332	0.0060	−0.2254	17,039
	TWO	$L=2$	0.7136	0.9633				18,581
		$L=5$	−0.0178	−0.4092	0.9291	0.0024	−0.5575	17,039
Example 3	Sloan		1	2				28,703
	King		0	1				28,853
	PSO	$L=2$	0.2588	0.6068				28,629
		$L=5$	−0.2965	0.5407	−0.6326	−0.0214	−0.1835	29,674
	CBO	$L=2$	0.2129	0.4426				28,608
		$L=5$	−0.5700	0.8618	−0.5831	0.0777	0.3363	27,992
	ECBO	$L=2$	0.1765	0.9272				28,587
		$L=5$	−0.4186	0.9776	−0.7792	0.1007	−0.0557	27,982
	TWO	$L=2$	0.1613	0.8465				28,579
		$L=5$	−0.3306	0.8144	−0.5654	0.0668	0.6810	27,977
Example 4	Sloan		1	2				157,457
	King		0	1				157,103
	PSO	$L=2$	0.0449	0.6963				157,095
		$L=5$	0.1106	0.9323	−0.0624	−0.0284	−0.3706	160,705
	CBO	$L=2$	0.0229	0.9146				157,095
		$L=5$	−0.9709	−0.9856	0.1853	0.2102	−0.3565	159,681
	ECBO	$L=2$	0.0620	0.9365				157,095
		$L=5$	−0.5805	−0.7778	0.2437	0.1277	−0.1065	159,676
	TWO	$L=2$	0.0721	0.9800				157,095
		$L=5$	−0.8206	−0.8691	0.5953	0.4801	0.2781	159,675

Table 13.3 Comparison of the results of different algorithms for wavefront reduction

Example	Algorithm		W_1	W_2	W_3	W_4	W_5	F_{rms}
Example 1	Sloan		1	2				20.1739
	King		0	1				21.0798
	PSO	$L=2$	0.3069	0.2202				20.1401
		$L=5$	−0.3188	0.9852	−0.7009	0.0489	0.118	17.2693
	CBO	$L=2$	0.8701	0.6144				20.1401
		$L=5$	−0.1888	0.9093	−0.8122	0.0210	0.8648	17.1544
	ECBO	$L=2$	0.8858	0.6517				20.1401
		$L=5$	−0.1891	−0.8720	0.9777	0.0199	−0.4653	17.2239
	TWO	$L=2$	0.8711	0.6352				20.1401
		$L=5$	0.0101	0.9086	−0.8164	−0.0034	−0.1742	17.1492
Example 2	Sloan		1	2				25.9092
	King		0	1				26.6508
	PSO	$L=2$	0.2053	0.3469				25.8411
		$L=5$	−0.0879	−0.3811	0.8933	0.012	−0.1249	23.5891
	CBO	$L=2$	0.3181	0.5433				25.8438
		$L=5$	−0.0713	−0.6558	0.9556	0.0095	−0.6861	23.4955
	ECBO	$L=2$	0.1855	0.3155				25.8411
		$L=5$	−0.1056	0.8982	−0.7698	0.0121	−0.7255	23.4743
	TWO	$L=2$	0.5752	0.9624				25.8438
		$L=5$	−0.0919	0.9571	−0.6803	0.0130	−0.0568	23.5090
Example 3	Sloan		1	2				18.3958
	King		0	1				18.4698
	PSO	$L=2$	0.0605	0.3289				18.3126
		$L=5$	−0.0709	0.6589	−0.9003	−0.0677	−0.4860	18.9964
	CBO	$L=2$	0.1382	0.8659				18.3235
		$L=5$	−0.2174	−0.4203	0.4590	0.0462	−0.2162	19.2531
	ECBO	$L=2$	0.1688	0.8892				18.3232
		$L=5$	0.1883	0.7370	−0.5928	−0.0607	−0.0945	18.6574
	TWO	$L=2$	0.0357	0.1982				18.3240
		$L=5$	0.0441	0.9101	−0.7696	0.0139	−0.6153	18.0211
Example 4	Sloan		1	2				32.3665
	King		0	1				32.2875
	PSO	$L=2$	0.0445	0.6963				32.2869
		$L=5$	0.2036	−0.9340	0.0601	0.0151	−0.5008	32.9486
	CBO	$L=2$	0.0361	0.8204				32.2869
		$L=5$	−0.0469	−0.9805	0.0890	0.0109	−0.0406	32.7939
	ECBO	$L=2$	0.0145	0.6215				32.2869
		$L=5$	−0.0665	−0.9779	0.2937	0.0204	0.0726	32.8298
	TWO	$L=2$	0.0772	0.9898				32.2869
		$L=5$	−0.3335	−0.6943	0.6808	0.1610	−0.0604	32.8845

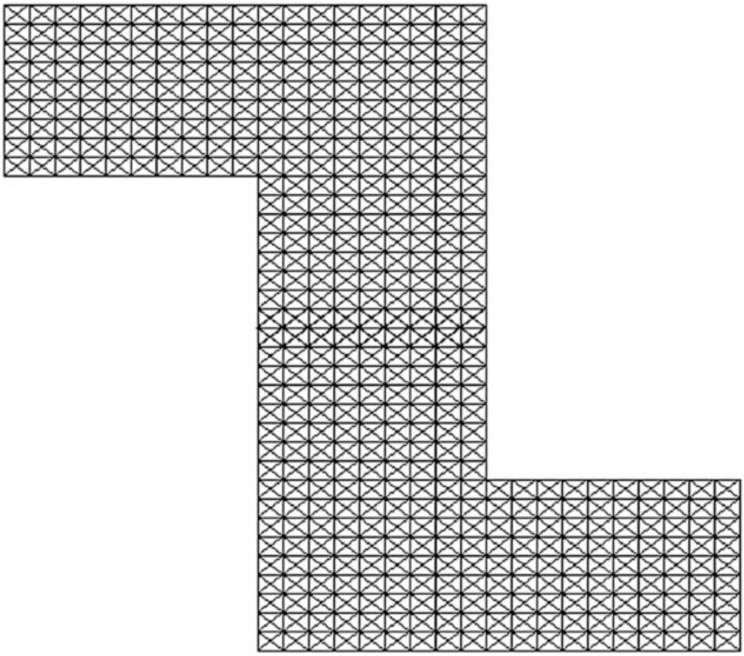


Fig. 13.3 A FEM of a shear wall

algorithms is tested on this model for bandwidth, profile, and wavefront optimization problems. The results for these problems are given in Tables 13.1, 13.2, and 13.3, respectively. Quality of the results is provisioned in these tables.

13.4.2 Example 2: A Rectangular FEM with Four Openings

This is the element clique graph of a rectangular FEM with four openings, as shown in Fig. 13.4, having 760 nodes. The performance of the PSO, CBO, ECBO, and TWO algorithms is tested on this model for bandwidth, profile, and wavefront minimizing problems. The results for these problems are provided in Tables 13.1, 13.2, and 13.3, respectively.

13.4.3 Example 3: The Model of a Fan

The graph model of a fan containing 1575 nodes is considered, as shown in Fig. 13.5. Similar to the previous examples, the results of the algorithms for

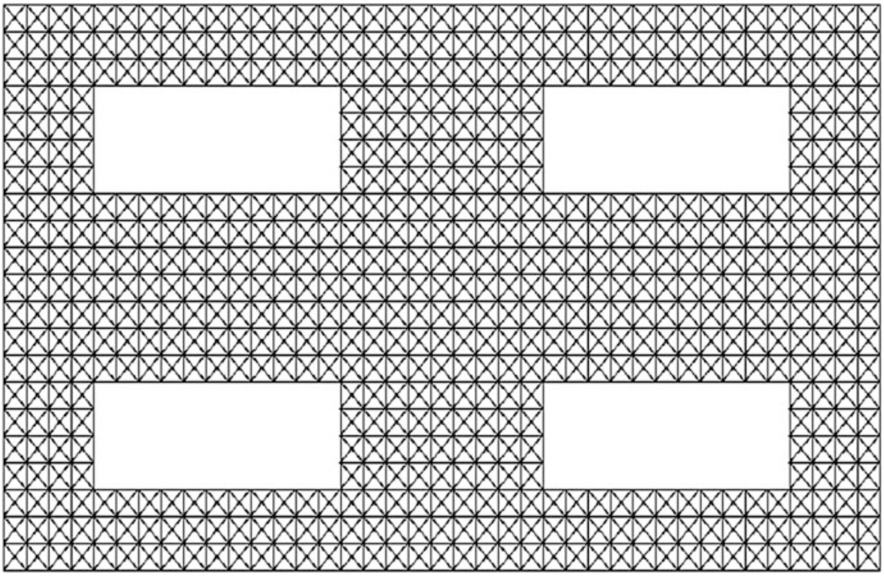


Fig. 13.4 The element clique graph of a rectangular FEM with four openings

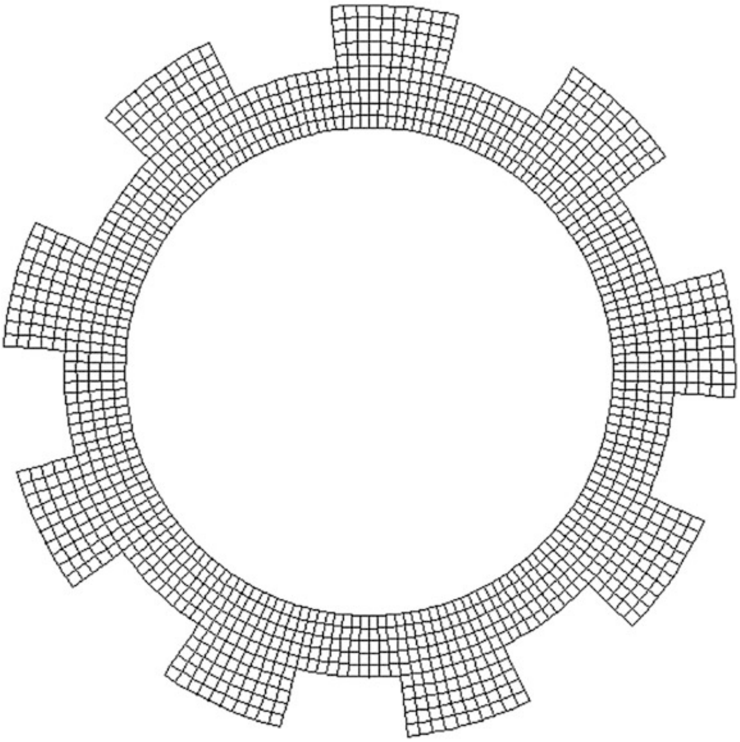


Fig. 13.5 The graph model of a fan

bandwidth, profile, and wavefront reduction problems are represented in Tables 13.1, 13.2, and 13.3 respectively, where the results can easily be compared.

13.4.4 Example 4: An H-Shaped Shear Wall

The FEM of an H-shaped shear wall with 4949 nodes is considered, as shown in Fig. 13.6. The performance of the abovementioned algorithms is tested on this model, and the results for bandwidth, profile, and wavefront minimizing problems are given in Tables 13.1, 13.2, and 13.3, respectively.

13.5 Discussion

For Example 2, comparison of the results is shown in Figs. 13.7 and 13.8. The convergence curves of the CBO, ECBO, PSO, and TWO algorithms are illustrated in Figs. 13.9, 13.10, 13.11, and 13.12. The convergence histories show that these four algorithms act in relatively the same way. To indicate the difference of the

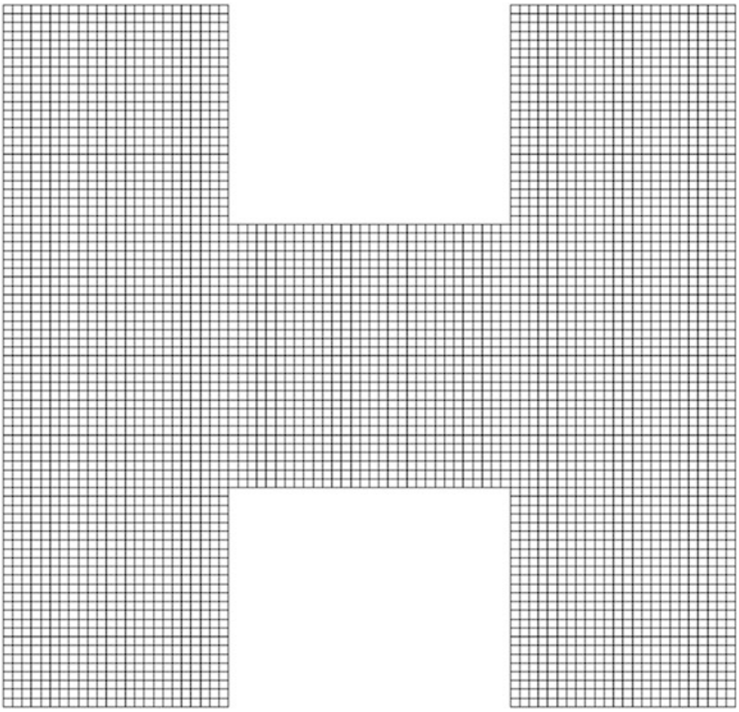


Fig. 13.6 The finite element grid model of a shear wall

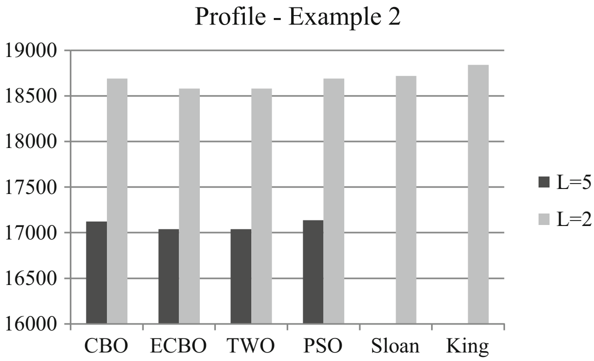


Fig. 13.7 Comparison of the profile results for Example 2 [1]

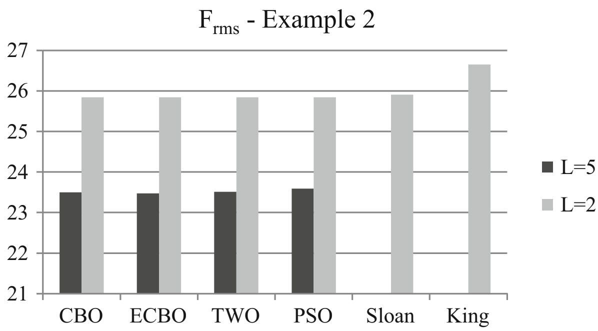


Fig. 13.8 Comparison of the F_{rms} results for Example 2 [1]

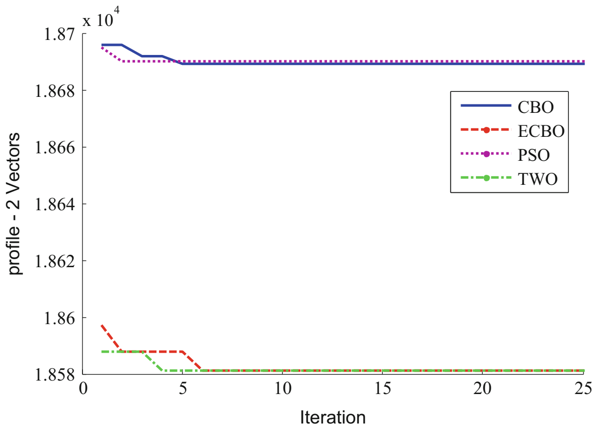


Fig. 13.9 Convergence curves of Example 2 for the CBO, ECBO, PSO, and TWO algorithms [1]

convergence curves better, only 25 iterations have been shown. As can be seen from Figs. 13.9, 13.10, 13.11, and 13.12, the CBO, ECBO, and TWO algorithms have better convergence, search better the space of the problem, and obtain better results compared to the PSO method.

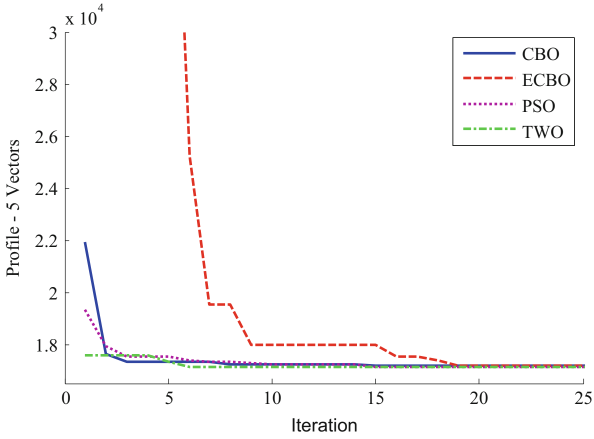


Fig. 13.10 The convergence history of Example 2 for the CBO, ECBO, PSO, and TWO algorithms [1]

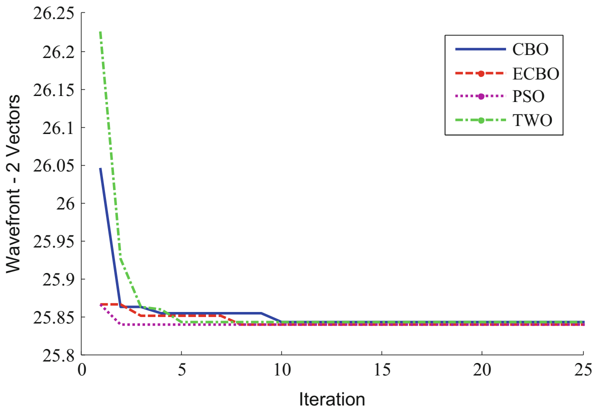


Fig. 13.11 Convergence curves of Example 2 for the CBO, ECBO, PSO, and TWO algorithms [1]

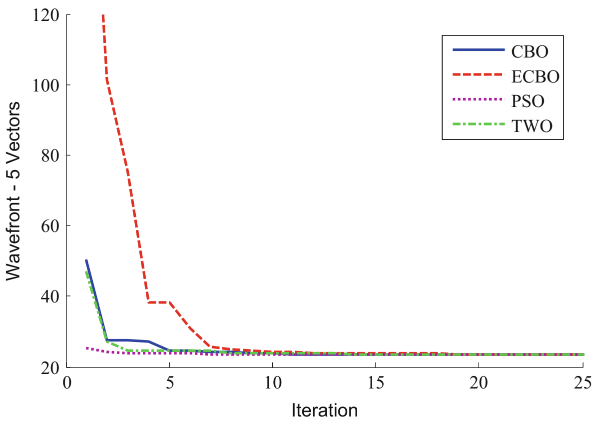


Fig. 13.12 Convergence curves of Example 2 for the CBO, ECBO, PSO, and TWO algorithms [1]

13.6 Concluding Remarks

The main purpose of this chapter was to show the performance and robustness of the CBO, ECBO, and TWO for bandwidth, profile, and wavefront reduction of matrices. From Table 13.1, it can be observed that the attained results from these three algorithms are quite satisfactory compared to the well-known graph theoretical method, four-step algorithm. CBO, its enhanced version, and TWO improve the bandwidth values previously obtained by CSS and ACO algorithms, and these values are the best results so far.

In profile and wavefront minimizing problems with $L = 2$ and 5 methods, the aim was to show the applicability of using different priority functions employing CBO, ECBO, and TWO algorithms. Optimal coefficients for these functions are obtained by optimization process, for decreasing the profile and wavefront of the stiffness matrices of finite element models. From Tables 13.2 and 13.3, it can be observed that Sloan and King's methods can be improved in most cases using some new parameters and coefficients. The weights achieved for different examples show that in the two-parameter approach ($L = 2$), more suitable profile and wavefront values can be obtained than those of the Sloan and King's algorithms. In the five-parameter method ($L = 5$), smaller profile and wavefront values can be attained than two-parameter approach and Sloan and King's algorithms except Example 4 that profile and wavefront values of Sloan and King's methods are smaller than those of the five-parameter approach. It should be noted that in the $L = 5$ algorithm, the active degrees are not updated as in Sloan's method. Therefore, one should not always expect a better result when five adjusted parameters are utilized in place of two free parameters. The value of profile and wavefront reduction in $L = 5$ method proportion to Sloan and King's algorithm is more than that in $L = 2$ method because of utilizing more graph properties. For example, comparison of profile and wavefront results for Example 2 is represented in Figs. 13.7 and 13.8, respectively. Among five parameters, the importance of parameter C_2 is the highest and parameter C_4 has the smallest effect.

A recently developed metaheuristic algorithm, tug of war optimization, is employed, and from Tables 13.2 and 13.3, it can be seen that this algorithm obtains good results like CBO and ECBO and in some cases achieves better values and the best results so far.

Though the methods of this chapter are used for nodal ordering in the stiffness method, however, the application of the methods can easily be extended to cycle or generalized cycles ordering to optimize the bandwidth of the flexibility matrices [3,4].

References

1. Kaveh A, Bijari Sh (2016) Bandwidth, profile and wavefront reduction using PSO, CBO, ECBO and TWO algorithms. Iran J Sci Technol (published online)
2. Kaveh A (1974) Applications of topology and matroid theory to the analysis of structures. Ph. D. thesis, Imperial College of Science and Technology, London University, UK

3. Kaveh A (2004) Structural mechanics: graph and matrix methods, 3rd edn. Research Studies Press, Somerset
4. Kaveh A (2006) Optimal structural analysis, 2nd edn. Wiley, Chichester
5. Papademetrious CH (1976) The NP-completeness of bandwidth minimization problem. Comput J 16:177–192
6. Gibbs NE, Poole WG, Stockmeyer PK (1976) An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM J Numer Anal 12:236–250
7. Cuthill E, McKee J (1969) Reducing the bandwidth of sparse symmetric matrices. In: Proceedings of the 24th national conference ACM, Bradon System Press, New Jersey, pp 157–172
8. Bernardes JAB, Oliveira SLGD (2015) A systematic review of heuristics for profile reduction of symmetric matrices. Proc Comput Sci 51:221–230
9. King IP (1970) An automatic reordering scheme for simultaneous equations derived from network systems. Int J Numer Methods Eng 2:523–533
10. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer International Publishing, Switzerland
11. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
12. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
13. Kaveh A, Zolghadr A (2016) A novel meta-heuristic algorithm: tug of war optimization. Int J Optim Civil Eng 6:469–493
14. Kaveh A, Sharafi P (2009) Nodal ordering for bandwidth reduction using ant system algorithm. Eng Comput 26(3):313–323
15. Kaveh A, Sharafi P (2012) Ordering for bandwidth and profile minimization problems via Charged System Search algorithm. Iran J Sci Technol Trans Civil Eng 36(C1):39–52
16. Kaveh A, Bijari S (2015) Bandwidth optimization using CBO and ECBO. Asian J Civil Eng 16 (4):535–545
17. Sloan SW (1986) An algorithm for profile and wavefront reduction of sparse matrices. Int J Numer Methods Eng 23:1693–1704
18. Kaveh A, Roosta GR (1997) Graph-theoretical methods for profile reduction. In: Mouchel Centenary conference on innovation in civil and structural engineering, Cambridge, UK
19. Rahimi Bondarabady HA, Kaveh A (2004) Nodal ordering using graph theory and a genetic algorithm. Finite Elem Anal Des 40(9–10):1271–1280

Chapter 14
Optimal Analysis and Design of Large-Scale Domes with Frequency Constraints

14.1 Introduction

Structural optimization involves a large number of structural analyses. When optimizing large structures, these analyses require a considerable amount of computational time and effort. However, there are specific types of structure for which the results of the analysis can be achieved in a much simpler and quicker way due to their special repetitive patterns. In this chapter, frequency constraint optimization of cyclically repeated space trusses is considered. An efficient technique is used to decompose the large initial eigenproblem into several smaller ones and thus to decrease the required computational time significantly (Kaveh and Zolghadr [1]).

In low-frequency vibration problems, the response of the structure primarily depends on its fundamental frequencies and mode shapes (Grandhi [2]). Therefore, the dynamic behavior of a structure can be controlled by constraining its fundamental frequencies. Mass minimization of a structure for which some natural frequencies should be upper and/or lower bounded is known as a structural optimization problem with frequency constraints.

History of structural optimization with frequency constraints dates back to 1960s and since then has always received considerable attention by optimization experts utilizing a wide variety of algorithms (Taylor [3], Armand [4], Cardou and Warner [5], Elwany and Barr [6], Lin et al. [7], Konzelman [8], Grandhi and Venkayya [9], Sedaghati et al. [10], Lingyun et al. [11], Gomes [12], Kaveh and Zolghadr [13, 14]).

In a frequency constraint structural optimization problem, large generalized eigenproblems should be solved in order to find the natural frequencies of the structure. The size of the structure affects the dimensions of the matrices involved and thus the required computational time and effort. On the other hand, as the number of optimization variables increases, more and more structural analyses are needed to be performed in order to obtain a near-optimal solution. There are numerous algebraic methods for eigensolution of large structural systems, some

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_14

of them utilizing such properties as sparsity and symmetry of the associated matrices. For general structures, utilization of general time-consuming algebraic methods seems to be inevitable. However, fast and efficient techniques could be used for several types of structures, which benefit specific characteristics such as symmetry. These methods utilize the characteristics of special categories of matrices whose eigenvalues and eigenvectors can be more easily obtained by using block diagonalization techniques. Several applications of these techniques could be found in the literature. Kaveh and Rahami [15, 16] utilized block diagonalization techniques for different types of canonical forms for applications in structural mechanics. Kaveh [17] employed special canonical forms for the efficient eigensolutions of Laplacian and adjacency matrices of special graphs and free vibration and buckling load analysis of cyclically repeated space truss structures (Koohestani and Kaveh [18]).

Many different types of complex structural systems can be considered as the cyclic repetition of a simple substructure around a revolution axis. These structures, which are usually called cyclically symmetric, exhibit some special patterns in their structural matrices. Structures like domes and cooling towers fall into this category. These special patterns and the benefits they bring about in the analysis of such structures have been studied in the works of Courant [19], Leung [20], Williams [21], Vakakis [22], Karpov et al. [23], Liu and Yang [24], El-Raheb [25], Zingoni [26], Tran [27], and Kaveh [17] among many others.

The aim of this chapter is to incorporate previously existing efficient methods of analysis for cyclically repeated truss structures into the well-known frequency constraint optimization problem in order to achieve considerable computational savings. An efficient method for free vibration analysis of these structures, introduced by Koohestani and Kaveh [18], is utilized to decompose the initial generalized eigenproblem to several smaller ones and to reduce the required computational time consequently. Other swift and efficient methods for the analysis of different types of symmetric, regular, and near-regular structures could be found in Kaveh [17].

The remainder of this chapter is organized as follows: In Sect. 14.2, the mathematical statement of the minimum weight optimization problem for a truss structure subject to frequency constraints is summarized. In Sect. 14.3, basic formulation of free vibration analysis of a truss structure and the corresponding stiffness matrix are presented concisely. The efficient eigensolution of cyclically repeated dome trusses is then discussed in Sect. 14.4 followed by three numerical examples, examined in Sect. 14.5, in order to show the efficiency of the proposed method. Finally, some concluding remarks are presented in Sect. 14.6.

14.2 Formulation of the Optimization Problem

Size optimization of a truss structure subject to frequency constraints where the objective is to minimize the weight of the structure can be mathematically stated as follows:

$$\begin{aligned}
&\text{Find } X = [x_1, x_2, x_3, \ldots, x_n] \\
&\text{to minimize } P(X) = f(X) \times f_{\text{penalty}}(X) \\
&\text{subject to} \\
&\omega_j \le \omega_j^* \quad \text{for some natural frequencies } j \\
&\omega_k \ge \omega_k^* \quad \text{for some natural frequencies } k \\
&x_{i\min} \le x_i \le x_{i\max}
\end{aligned} \tag{14.1}$$

where X is the vector of the design variables, i.e., cross-sectional areas; n is the number of optimization variables which depends on the element grouping scheme; $f(X)$ is the cost function, which is taken as the weight of the structure in a weight optimization problem; and $f_{\text{penalty}}(X)$ is the penalty function, which is used to make the problem unconstrained. When some constraints are violated in a particular solution, the penalty function magnifies the weight of the solution by taking values bigger than one; $P(X)$ is the penalized cost function or the objective function to be minimized; ω_j is the jth natural frequency of the structure with the corresponding upper bound ω_j^*, while ω_k is the kth natural frequency of the structure with the corresponding lower bound ω_k^*; and $x_{i\min}$ and $x_{i\max}$ are the lower and upper bounds for the design variable x_i, respectively.

The cost function can be expressed as

$$f(X) = \sum_{i=1}^{nm} \rho_i L_i A_i \tag{14.2}$$

where nm is the number of structural members and ρ_i, L_i, and A_i are the material density, length, and cross-sectional area of the ith element.

The penalty function is defined as

$$f_{\text{penalty}}(X) = (1 + \varepsilon_1 . v)^{\varepsilon_2}, \quad v = \sum_{i=1}^{q} v_i \tag{14.3}$$

where q is the number of frequency constraints. The values for v_i can be considered as

$$v_i = \begin{cases} 0 & \text{if the } i\text{th constraint is satisfied} \\ \left| 1 - \dfrac{\omega_i}{\omega^*_i} \right| & \text{else} \end{cases} \tag{14.4}$$

The parameters ε_1 and ε_2 determine the degree to which a violated solution should be penalized. In this study ε_1 is taken as unity, and ε_2 starts from 1.5 and then linearly increases to 6 for all test problems. Such a scheme penalizes the infeasible solutions more severely as the optimization process proceeds. As a result, in the early stages, the agents are free to explore the search space, but at the end they tend to choose solutions without violation.

14.3 Free Vibration Analysis of Structures

14.3.1 Basic Formulation

Abovementioned frequency constraint structural optimization involves a large number of free vibration analyses of the structural system under consideration. The mathematical formulation of the free vibration of a structure leads to a generalized eigenproblem of the following form:

$$\boldsymbol{K}\boldsymbol{\phi}_i = \boldsymbol{\gamma}_i \boldsymbol{M}\boldsymbol{\phi}_i \tag{14.5}$$

in which $\boldsymbol{K}$ is the elastic stiffness matrix and $\boldsymbol{M}$ is the mass matrix of the structure, $\boldsymbol{\phi}_i$ is the ith eigenvector (mode shape) corresponding to the ith eigenvalue $\boldsymbol{\gamma}_i$, and the ith period (T_i) and circular frequency ($\boldsymbol{\omega}_i$) are related to the ith eigenvalue by

$$\boldsymbol{\gamma}_i = \boldsymbol{\omega}_i^{\,2} = (2\boldsymbol{\pi}/T_i)^2 \tag{14.6}$$

General methods to solve the generalized eigenproblem of Eq. (14.4) require manipulation of large matrices resulting in high computational costs. This is particularly the case when performing structural optimization, where the analysis part should be carried out thousands of times. Specifically, when the number of degrees of freedom of the structure is relatively large, the required computational time becomes significant. In the next subsection, a formulation is presented based on the works of Kaveh [17] and Koohestani and Kaveh [18], which helps to obtain special patterns in the matrices involved in Eq. (14.4). Such a formulation allows the initial eigenproblem to be decomposed into several smaller ones and results in a much faster solution to the problem at hand.

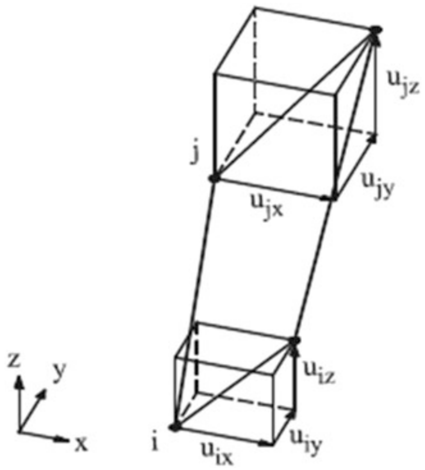


Fig. 14.1 A three-dimensional truss element in the global Cartesian coordinate system

14.3.2 Elastic Stiffness Matrix of a Three-Dimensional Truss Element

Figure 14.1 shows a three-dimensional (3D) truss element in global Cartesian coordinate system together with the corresponding components of displacement. The elastic stiffness matrix of such an element is as follows:

$$K_{ij}^{xyz} = \frac{EA_{ij}}{L_{ij}} \begin{bmatrix} d_{ii} & -d_{ii} \\ -d_{ii} & d_{ii} \end{bmatrix}, \quad d_{ii} = \begin{bmatrix} l_{ij}^2 & l_{ij}m_{ij} & l_{ij}n_{ij} \\ m_{ij}l_{ij} & m_{ij}^2 & m_{ij}n_{ij} \\ n_{ij}l_{ij} & n_{ij}m_{ij} & n_{ij}^2 \end{bmatrix} \tag{14.7}$$

where E is the modulus of elasticity and A_{ij} and L_{ij} are the cross-sectional area and the length of the element, respectively. In the submatrix d_{ij}, l_{ij}, m_{ij}, and n_{ij} are the direction cosines of the element with respect to x-, y-, and z-axes, respectively:

$$l_{ij} = \frac{x_i - x_i}{L_{ij}}, \quad m_{ij} = \frac{y_i - y_i}{L_{ij}}, \quad n_{ij} = \frac{z_i - z_i}{L_{ij}} \tag{14.8}$$

It is apparent from Eq. (14.6) that the element stiffness matrix in Cartesian coordinates is not invariant under rotation about any axis. Therefore, the global stiffness matrix of a cyclically repetitive structure does not generally exhibit any favorable pattern in Cartesian coordinates.

In order to use the desirable patterns of the global stiffness matrices of cyclically symmetric structures, the element global stiffness matrix should be developed in a cylindrical coordinate system. In such a coordinate system, the element stiffness matrix is invariant under rotation about an axis of revolution. Thus, the global stiffness matrix of a cyclically repeated structure exhibits a special pattern which is highly desired for efficient eigensolutions. A three-dimensional truss element

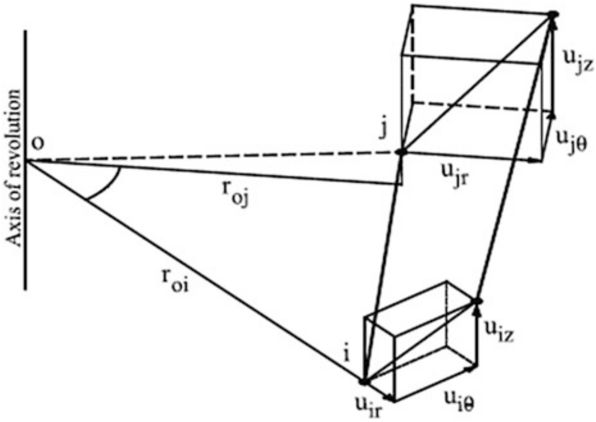


Fig. 14.2 Schematic of the three-dimensional truss element in the global cylindrical coordinate system

together with its displacement components in cylindrical coordinate system is shown in Fig. 14.2.

The element stiffness matrix in Cartesian coordinate system can be transformed into the cylindrical coordinate system by the following transformation:

$$K_{ij}^{rz\theta} = R^t K_{ij}^{xyz} R \tag{14.9}$$

where R is a transformation matrix:

$$R = \begin{bmatrix} R_{oi} & 0 \\ 0 & R_{oj} \end{bmatrix} \tag{14.10}$$

in which the submatrices R_{oi} and R_{oj} can be defined as

$$R_{oi} = \begin{bmatrix} l_{oi} & -m_{oi} & 0 \\ m_{oi} & l_{oi} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{oi} = \begin{bmatrix} l_{oj} & -m_{oj} & 0 \\ m_{oj} & l_{oj} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

in which we have

$$l_{oi} = \frac{x_i - x_0}{r_{oi}}, \quad m_{oi} = \frac{y_i - y_0}{r_{oi}}, \quad l_{oj} = \frac{x_j - x_0}{r_{oj}}, \quad m_{oj} = \frac{y_j - y_0}{r_{oj}} \tag{14.12}$$

where

$$r_{oi} = \sqrt{x_i^2 + y_i^2}, \quad r_{oj} = \sqrt{x_j^2 + y_j^2} \tag{14.13}$$

The expanded form of the element global stiffness matrix in cylindrical coordinates can then be derived as

$$K_{ij}^{rz\theta} = \frac{EA_{ij}}{L_{ij}} \begin{bmatrix} s_1^2 & -s_1 s_2 & s_1 n_{ij} & -s_1 s_3 & s_1 s_4 & -s_1 n_{ij} \\ & s_2^2 & -s_2 n_{ij} & s_2 s_3 & -s_2 s_4 & s_2 n_{ij} \\ & & n_{ij}^2 & -s_3 n_{ij} & s_4 n_{ij} & -n_{ij}^2 \\ & & & s_3^2 & -s_3 s_4 & s_3 n_{ij} \\ & sym & & & s_4^2 & -s_4 n_{ij} \\ & & & & & n_{ij}^2 \end{bmatrix} \tag{14.14}$$

where

$$\begin{aligned} s_1 &= l_{ij} l_{oi} + m_{ij} m_{oi} \\ s_2 &= l_{ij} m_{oi} + m_{ij} l_{oi} \\ s_3 &= l_{ij} l_{oj} + m_{ij} m_{oj} \\ s_4 &= l_{ij} m_{oj} + m_{ij} l_{oj} \end{aligned} \tag{14.15}$$

As it can be seen, this form of element stiffness matrix is invariant under rotation about the axis of revolution. Therefore, all similar substructures have the same stiffness matrix regardless of their rotational positions. Hence, the global stiffness matrix of the structure embodies some interesting patterns, which can be used for efficient eigensolution of the structure.

In relation to mass matrix, it should be noted that both lumped and consistent mass matrices are invariant under rotation and therefore no transformation is needed. Since additional lumped masses are added to the free nodes, the difference between consistent and lumped mass matrices is negligible. A lumped mass matrix, which lumps the masses of the elements in their end nodes, is utilized in this chapter. Therefore, the mass matrix is a diagonal one.

14.4 Efficient Eigensolution

Matrices related to a three-dimensional truss element in cylindrical coordinate system are invariant under rotation about axis of revolution. Therefore, if the nodes of all similar substructures are labeled in a similar manner, the matrices corresponding to these substructures would be the same, and the global mass and stiffness matrices of a cyclically repeated structure exhibit the canonical form shown in Eq. (14.15). This canonical form is called block tri-diagonal matrix with corner blocks (BTMCB).

$$\begin{bmatrix} A & B & & & & B^t \\ B^t & A & B & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & B^t & A & B \\ B & & & & B^t & A \end{bmatrix} \tag{14.16}$$

For a three-dimensional truss structure which is formed of n cyclically repeated substructures each having m nodes, both mass and stiffness matrices are $3\,nm \times 3\,nm$. Submatrices A, B, and B^t are square matrices with dimension 3 m. Although applying the support conditions will change these dimensions, the canonical form of Eq. (14.15) will be preserved if the boundary conditions are also cyclically symmetric. Hence, the structural matrices could be decomposed using Kronecker products as

$$K_{(3nm\times 3nm)} = I_{n\times n} \otimes A_{K(3m\times 3m)} + H_{(n\times n)} \otimes B_{K(3m\times 3m)} + H^t_{(n\times n)} \otimes B^t_{K(3m\times 3m)} \tag{14.17}$$

$$M_{(3nm\times 3nm)} = I_{n\times n} \otimes A_{M(3m\times 3m)} + H_{(n\times n)} \otimes B_{M(3m\times 3m)} + H^t_{(n\times n)} \otimes B^t_{M(3m\times 3m)} \tag{14.18}$$

where subscripts K and M for A, B, and B^t refer to stiffness and mass matrices, respectively, I is an $n \times n$ identity matrix, and H is an $n \times n$ unsymmetric permutation matrix as

$$H = \begin{bmatrix} 0 & 1 & & & & 0 \\ 0 & 0 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & 0 & 0 & 1 \\ 1 & & & & 0 & 0 \end{bmatrix} \tag{14.19}$$

Kronecker product of two matrices $A_{m\times n}$ and $B_{p\times q}$, denoted by $A \otimes B$, is an $mp \times nq$ block matrix and could be defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{14.20}$$

Block diagonalization of a BTMCB matrix is studied in Kaveh [17] and Koohestani and Kaveh [18] and could be summarized as follows. Equation (14.4) has a nontrivial solution if and only if

$$\det(\Omega_i) = \det(K - \gamma_i M) = 0 \tag{14.21}$$

where "det" stands for determinant. Here, the goal is to block diagonalize Ω_i and hence to decompose the main problem into some simpler subproblems. Let us consider the following definitions:

$$\begin{aligned} A &= A_K - \gamma_i A_M \\ B &= B_K - \gamma_i B_M \\ B^t &= B_K^t - \gamma_i B_M^t \end{aligned} \tag{14.22}$$

Combining Eqs. (14.17 and 14.18) with the above equations, Ω_i can be written as

$$\Omega_i = I \otimes A + H \otimes B + H^t \otimes B^t \tag{14.23}$$

This form of Ω_i can now be block diagonalized, and its jth block is as follows:

$$\Omega_i^j = A + \lambda_j B + \overline{\lambda_j} B^t \tag{14.24}$$

where λ_j is the jth eigenvalue of matrix H and the bar sign means conjugation of a general complex number. Thus, the following equation holds

$$\det(\Omega_i) = \prod_{j=1}^{n} \det\left(\Omega_i^j\right) \tag{14.25}$$

The determinant of the jth block of Ω_i is in turn a new generalized eigenproblem. Therefore, the original eigenproblem is decomposed into n highly smaller and simpler subproblems as

$$K_j x_i = \gamma_i M_j x_i, \quad j = 1, 2, 3, \ldots, n \tag{14.26}$$

in which

$$\begin{aligned} K_j &= A_K + \lambda_j B_K + \overline{\lambda_j} B_K^t \\ M_j &= A_M + \lambda_j B_M + \overline{\lambda_j} B_M^t \end{aligned} \tag{14.27}$$

where x_i could be converted to the required eigenvector corresponding to $\boldsymbol{\gamma}_{\mathrm{i}}$ (Kaveh [17]).

14.5 Numerical Examples

In this section three numerical examples are studied in order to examine the viability and efficiency of the proposed method. Democratic particle swarm optimization (DPSO) as introduced by Kaveh and Zolghadr [14] is utilized as the optimization algorithm. However, any other metaheuristic algorithm could be used. The algorithm and the finite element analysis were implemented by MATLAB R2009a on a laptop computer with an Intel (R) Core(TM)2 Duo 2.50 GHz processor and 4.00 GB RAM under the Microsoft Windows Vista™ Home Basic operating system. MATLAB internal eigenvalue function was used equally for the initial eigenproblem and the decomposed ones. The overall computational times required for different optimization runs utilizing the standard method and the proposed one are compared. The results show that the proposed efficient method is significantly faster.

14.5.1 A 600-Bar Single-Layer Dome

The first test problem is the 600-bar single-layer dome structure shown in Fig. 14.3. The entire structure is composed of 216 nodes and 600 elements generated by cyclic repetition of a substructure having 9 nodes and 25 elements. The angle of cyclic symmetry between similar substructures is 15°. A nonstructural mass of 100 kg is attached to all free nodes. Table 14.1 summarizes the material properties, variable bounds, and frequency constraints for this example. Figure 14.4 shows a substructure in more detail for nodal numbering and coordinates. Each of the elements of this substructure is considered as a design variable. Thus, this is a size optimization problem with 25 variables.

Using the classical method, it takes 2.6150 s to perform a typical analysis for this structure, while the efficient method needs 0.0198 s, i.e., the efficient method is about 132 times faster on a single analysis. Two different optimization cases are performed on this example as well as the other two. In Case 1, the initial eigenproblem is solved directly using MATLAB internal eigenvalue function;

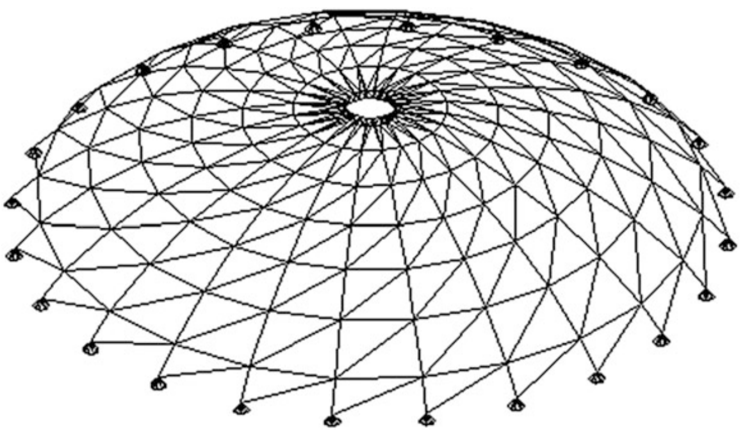


Fig. 14.3 Schematic of the 600-bar single-layer dome

Table 14.1 Material properties, variable bounds, and frequency constraints for the 600-bar single-layer dome

Property/unit	Value
E (modulus of elasticity)/N/m^2	2×10^{11}
ρ (material density)/kg/m^3	7850
Added mass/kg	100
Design variable lower bound/m^2	1×10^{-4}
Design variable upper bound/m^2	100×10^{-4}
Constraints on the first three frequencies/Hz	$\omega_1 \geq 5$, $\omega_3 \geq 7$

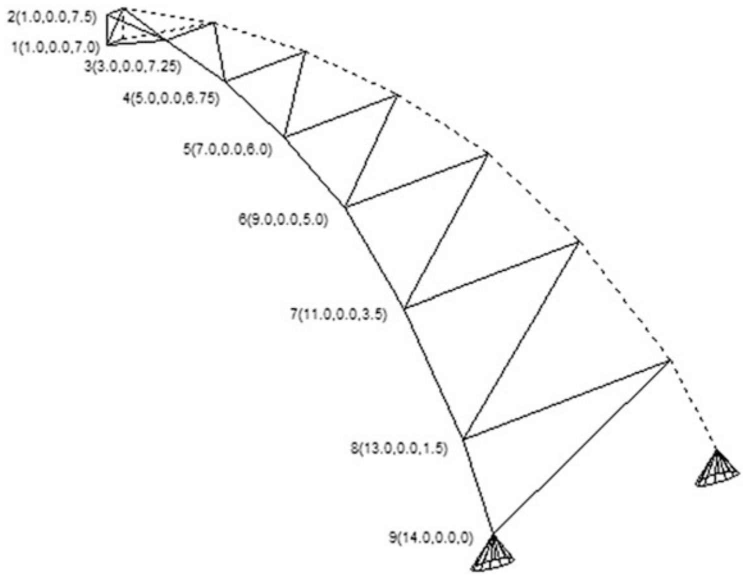


Fig. 14.4 Details of a substructure of the 600-bar single-layer dome

this is called the classical method. In Case 2, the abovementioned efficient method is used for the analysis part, i.e., the initial eigenproblem is decomposed into several smaller ones, and then each of the subproblems is solved using the same MATLAB function. In this example, 30 particles and 300 iterations (9000 analyses) are used for both cases. The required computational time to complete a single optimization run for Cases 1 and 2 is 27,326.25 s and 190.77 s, respectively. This means that the optimization procedure could be performed about 143 times faster using the efficient analysis method under the same circumstances. This example was solved 10 times using the efficient analysis method and the best result is presented in Table 14.2.

The total computational time to perform ten optimization runs using the efficient method is 1906.68 s (less than an hour), while it would have taken approximately

Table 14.2 Optimized design for the 600-bar dome truss problem (added masses are not included)

Element no. (nodes)	Cross-sectional area (cm^2)	Element no. (nodes)	Cross-sectional area (cm^2)
1 (1–2)	1.365	14 (5–13)	5.529
2 (1–3)	1.391	15 (5–14)	7.007
3 (1–10)	5.686	16 (6–7)	5.462
4 (1–11)	1.511	17 (6–14)	3.853
5 (2–3)	17.711	18 (6–15)	7.432
6 (2–11)	36.266	19 (7–8)	4.261
7 (3–4)	13.263	20 (7–15)	2.253
8 (3–11)	16.919	21 (7–16)	4.337
9 (3–12)	13.333	22 (8–9)	4.028
10 (4–5)	9.534	23 (8–16)	1.954
11 (4–12)	9.884	24 (8–17)	4.709
12 (4–13)	9.547	25 (9–17)	1.410
13 (5–6)	7.866	Weight (kg)	6344.55

Table 14.3 Natural frequencies (Hz) evaluated at the optimized design for the 600-bar dome truss problem

Frequency number	Frequency value
1	5.000
2	5.000
3	7.000
4	7.000
5	7.000

273,112.84 s (more than 3 days) to perform the same runs using the classical method. Table 14.3 presents the first five natural frequencies of the optimized structure. It can be seen that the constraints are fully satisfied. These frequencies are in full agreement with the results of the classical analysis method up to ten significant digits. The mean weight of the structures found in ten runs is 6674.71 kg with a standard deviation of 473.21 kg. Figure 14.5 shows the convergence curve of the best result for the 600-bar dome truss using the efficient method.

14.5.2 A 1180-Bar Dome Truss

The second test problem solved in this study was the weight minimization of the 1180-bar dome truss structure shown in Fig. 14.6. The entire structure is composed of 400 nodes and 1180 elements generated by cyclic repetition of a substructure with 20 nodes and 59 elements. The angle of cyclic symmetry between similar substructures is 18°. A nonstructural mass of 100 kg is attached to all free nodes. Table 14.4 summarizes the material properties, variable bounds, and frequency constraints for this example. Figure 14.7 shows a substructure in more detail for

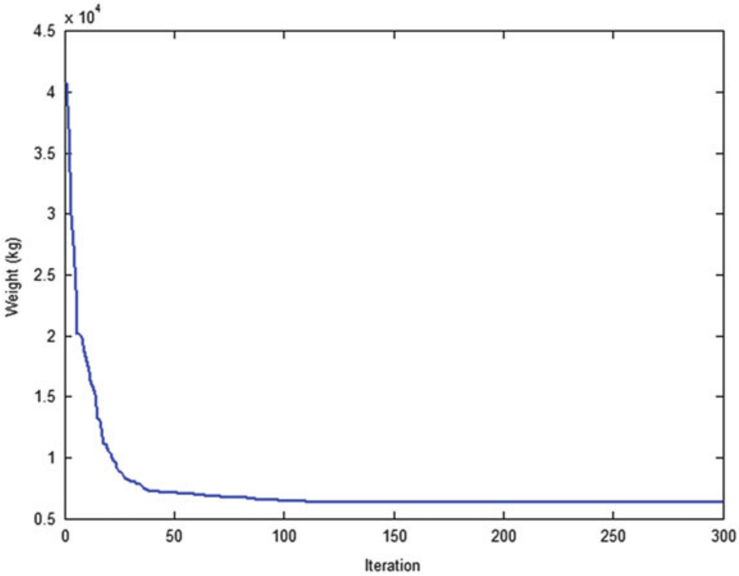


Fig. 14.5 Convergence curve of the best result for the 600-bar dome truss using the efficient method [1]

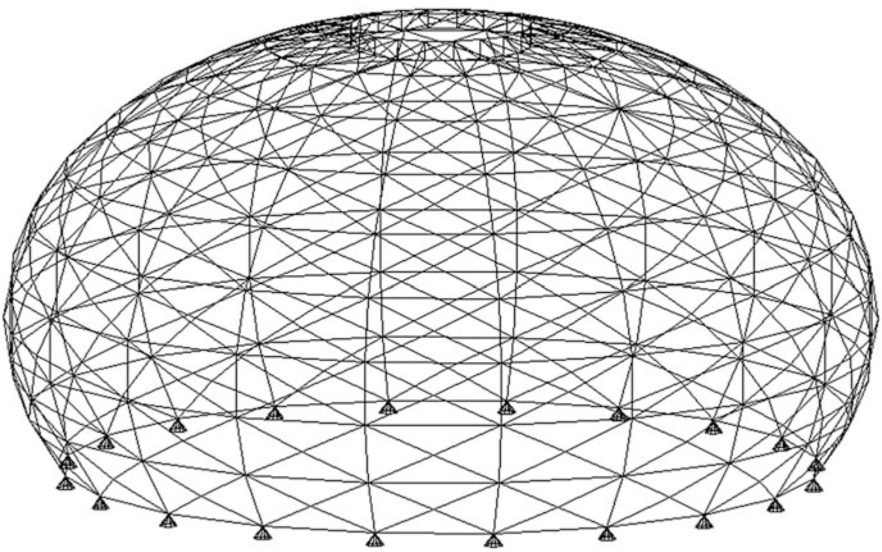


Fig. 14.6 Schematic of the 1180-bar dome truss

Table 14.4 Material properties, variable bounds, and frequency constraints for the 1180-bar dome truss

Property/unit	Value
E (modulus of elasticity)/N/m^2	2×10^{11}
ρ (material density)/kg/m^3	7850
Added mass/kg	100
Design variable lower bound/m^2	1×10^{-4}
Design variable upper bound/m^2	100×10^{-4}
Constraints on the first three frequencies/Hz	$\omega_1 \geq 7$, $\omega_3 \geq 9$

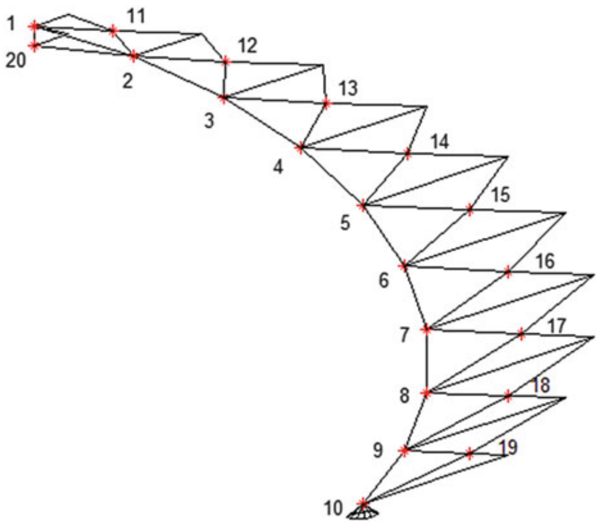


Fig. 14.7 Details of a substructure of the 1180-bar dome truss

nodal numbering. Table 14.5 summarizes the coordinates of the nodes in Cartesian coordinate system. Each of the elements of this substructure is considered as a design variable. Thus, this is a size optimization problem with 59 variables.

A single analysis takes up to 11.3575 s of computational time using the classical method. The required computational time for a similar analysis using the efficient method is only 0.0720 s. This means that the efficient method is about 157 times faster for a single analysis. About 100 particles and 500 iterations (50,000 analyses) are used for optimization of this test problem. The required computational time to complete a single run for Case 2 is 7095.56 s. Figure 14.8 shows the variation of the computational time with the number of analyses for Case 1. According to the figure, it is estimated that it would take 800,160 s to perform the same optimization run for Case 1 (50,000 analyses). Therefore, the optimization procedure could be performed about 113 times faster under the same circumstances using the efficient analysis method. Again, this example was solved ten times using the efficient analysis method and the best result is presented in Table 14.6.

Table 14.5 Coordinates of the nodes of the main structure (the 1180-bar dome truss)

Node no.	Coordinates (*x*, *y*, *z*)	Node no.	Coordinates (*x*, *y*, *z*)
1	(3.1181, 0.0, 14.6723)	11	(4.5788, 0.7252, 14.2657)
2	(6.1013, 0.0, 13.7031)	12	(7.4077, 1.1733, 12.9904)
3	(8.8166, 0.0, 12.1354)	13	(9.9130, 1.5701, 11.1476)
4	(11.1476, 0.0, 10.0365)	14	(11.9860, 1.8984, 8.8165)
5	(12.9904, 0.0, 7.5000)	15	(13.5344, 2.1436, 6.1013)
6	(14.2657, 0.0, 4.6358)	16	(14.4917, 2.2953, 3.1180)
7	(14.9179, 0.0, 1.5676)	17	(14.8153, 2.3465, 0.0)
8	(14.9179, 0.0, −1.5677)	18	(14.4917, 2.2953, −3.1181)
9	(14.2656, 0.0, −4.6359)	19	(13.5343, 2.1436, −6.1014)
10	(12.9903, 0.0, −7.5001)	20	(3.1181, 0.0, 13.7031)

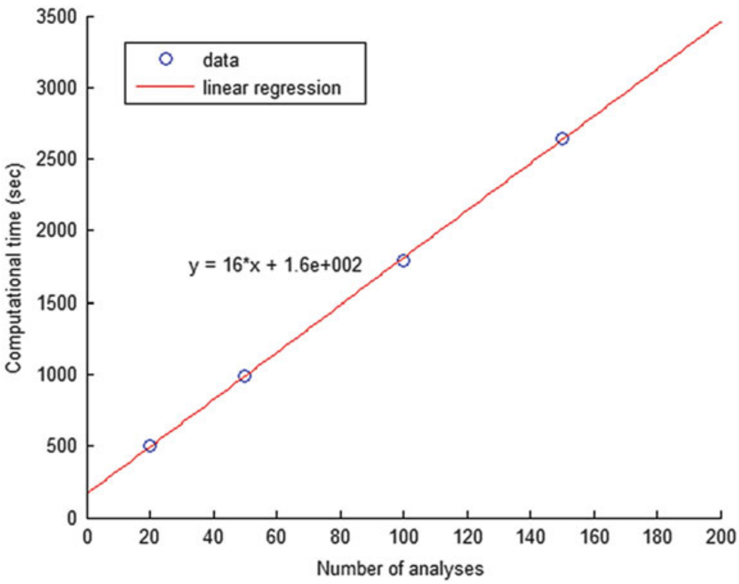


Fig. 14.8 Variation of the computational time with a number of analyses for Case 1 (1180-bar dome truss)

It takes 68,933.06 s to perform ten optimization runs using the efficient method for this example, while it would have taken approximately 7,773,580 s (about 90 days) to perform the same runs using the classical method. Table 14.7 presents the first five natural frequencies of the optimized structure for this example. The mean weight of the structures found in ten runs is 38,294.45 kg with a standard deviation of 550.5 kg. Figure 14.9 shows the convergence curve of the best result for the 1180-bar dome truss using the efficient method.

Table 14.6 Optimized design for the 1180-bar dome truss problem (added masses are not included)

Element no. (nodes)	Cross-sectional area (cm^2)	Element no. (nodes)	Cross-sectional area (cm^2)
1 (1–2)	7.926	31 (8–9)	34.642
2 (1–11)	10.426	32 (8–17)	19.860
3 (1–20)	2.115	33 (8–18)	25.079
4 (1–21)	14.287	34 (8–28)	18.965
5 (1–40)	3.846	35 (9–10)	47.514
6 (2–3)	5.921	36 (9–18)	28.133
7 (2–11)	7.955	37 (9–19)	33.023
8 (2–12)	6.697	38 (9–29)	32.263
9 (2–20)	1.889	39 (10–19)	33.401
10 (2–22)	11.881	40 (10–30)	1.344
11 (3–4)	7.121	41 (11–21)	9.327
12 (3–12)	6.080	42 (11–22)	7.202
13 (3–13)	6.599	43 (12–22)	6.792
14 (3–23)	7.772	44 (12–23)	6.228
15 (4–5)	9.358	45 (13–23)	6.601
16 (4–13)	6.213	46 (13–24)	6.584
17 (4–14)	8.200	47 (14–24)	8.320
18 (4–24)	7.799	48 (14–25)	8.844
19 (5–6)	11.752	49 (15–25)	11.254
20 (5–14)	7.494	50 (15–26)	12.162
21 (5–15)	9.696	51 (16–26)	13.854
22 (5–25)	9.177	52 (16–27)	13.844
23 (6–7)	17.326	53 (17–27)	17.536
24 (6–15)	11.797	54 (17–28)	20.551
25 (6–16)	14.002	55 (18–28)	24.072
26 (6–26)	11.562	56 (18–29)	27.287
27 (7–8)	23.981	57 (19–29)	32.965
28 (7–16)	12.996	58 (19–30)	36.940
29 (7–17)	16.591	59 (20–40)	3.837
30 (7–27)	15.910	Weight (kg)	37,779.81

Table 14.7 Natural frequencies (Hz) evaluated at the optimized design for the 1180-bar dome truss problem

Frequency number	Frequency value
1	7.000
2	7.000
3	9.000
4	9.000
5	9.005

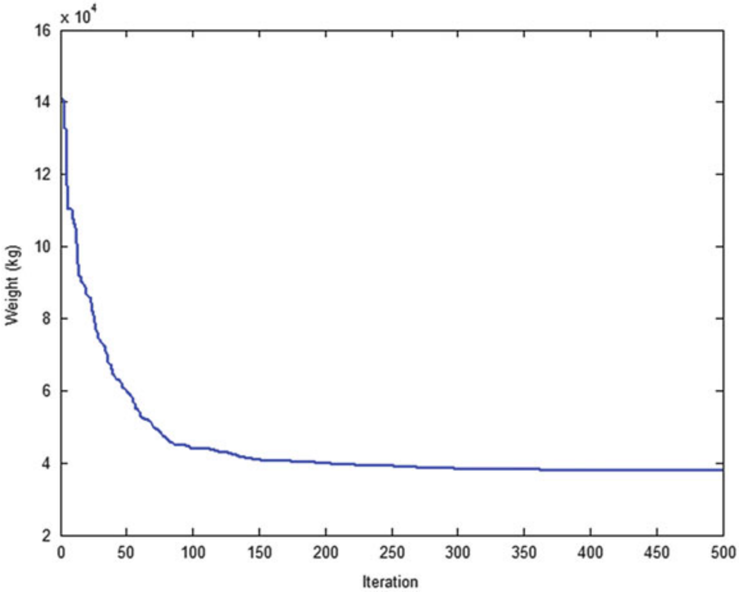


Fig. 14.9 Convergence curve of the best result for the 1180-bar dome truss using the efficient method [1]

14.5.3 A 1410-Bar Double-Layer Dome Truss

The third test problem solved in this chapter was the weight minimization of the 1410-bar double-layer dome truss as shown in Fig. 14.10. The entire structure is composed of 390 nodes and 1410 elements generated by cyclic repetition of a substructure with 13 nodes and 47 elements. The angle of cyclic symmetry between similar substructures is 12°. A nonstructural mass of 100 kg is attached to all free nodes. Table 14.8 summarizes the material properties, variable bounds, and frequency constraints for this example. Figure 14.11 shows a substructure in more detail for nodal numbering. Table 14.9 presents the coordinates of the nodes in Cartesian coordinate system. Each of the elements of this substructure is considered as a design variable. Thus, this is a size optimization problem with 47 variables.

Required computational times for classical and efficient methods are 11.7101 and 0.0140 s, respectively. Like the previous example, 100 particles and 500 iterations (50,000 analyses) are used for optimization of this test problem. The required computational time to complete a single run for Case 2 is 3871.62 s. Figure 14.12 shows the variation of the computational time with the number of analyses for Case 1. According to the figure, it is estimated that it would take 950,240 s to perform the same optimization run for Case 1 (50,000 analyses). Therefore, the optimization procedure could be performed about 245 times faster under the same circumstances using the efficient analysis method. This example was solved ten times using the efficient analysis method and the best result is presented in Table 14.10.

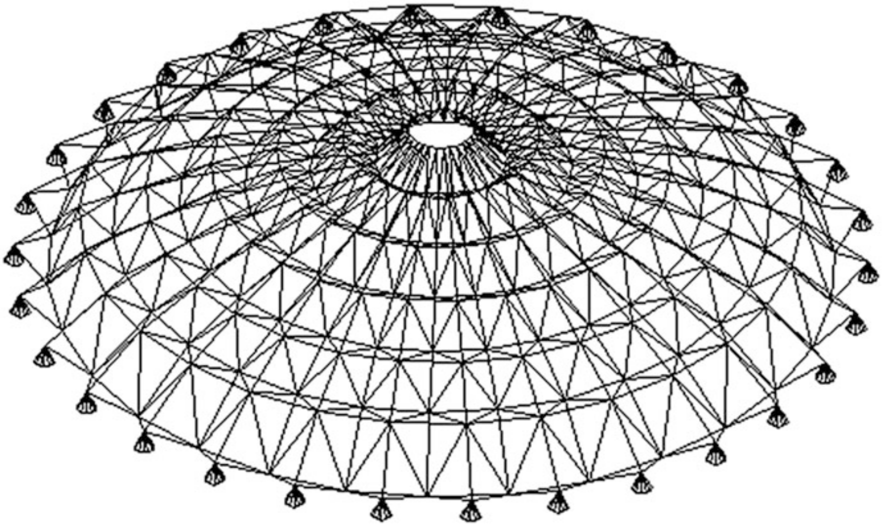


Fig. 14.10 Schematic of the 1410-bar dome truss

Table 14.8 Material properties, variable bounds, and frequency constraints for the 1410-bar dome truss

Property/unit	Value
E (modulus of elasticity)/N/m^2	2×10^{11}
ρ (material density)/kg/m^3	7850
Added mass/kg	100
Design variable lower bound/m^2	1×10^{-4}
Design variable upper bound/m^2	100×10^{-4}
Constraints on the first three frequencies/Hz	$\omega_1 \geq 7$, $\omega_3 \geq 9$

Fig. 14.11 Details of a substructure of the 1410-bar dome truss

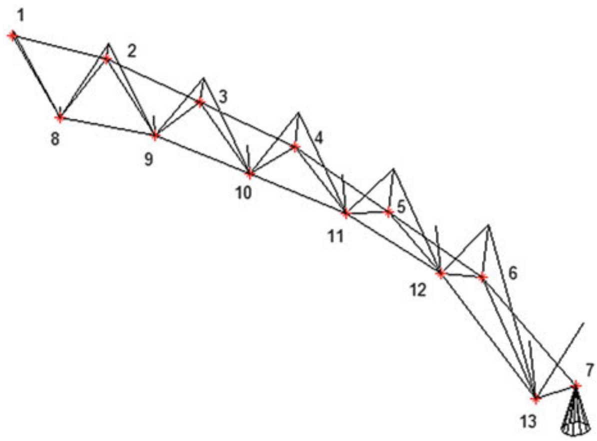


Table 14.9 Coordinates of the nodes of the main substructure (the 1410-bar dome truss)

Node no.	Coordinates (x, y, z)	Node no.	Coordinates (x, y, z)
1	(1.0, 0.0, 4.0)	8	(1.989, 0.209, 3.0)
2	(3.0, 0.0, 3.75)	9	(3.978, 0.418, 2.75)
3	(5.0, 0.0, 3.25)	10	(5.967, 0.627, 2.25)
4	(7.0, 0.0, 2.75)	11	(7.956, 0.836, 1.75)
5	(9.0, 0.0, 2.0)	12	(9.945, 1.0453, 1.0)
6	(11.0, 0.0, 1.25)	13	(11.934, 1.2543, −0.5)
7	(13.0, 0.0, 0.0)		

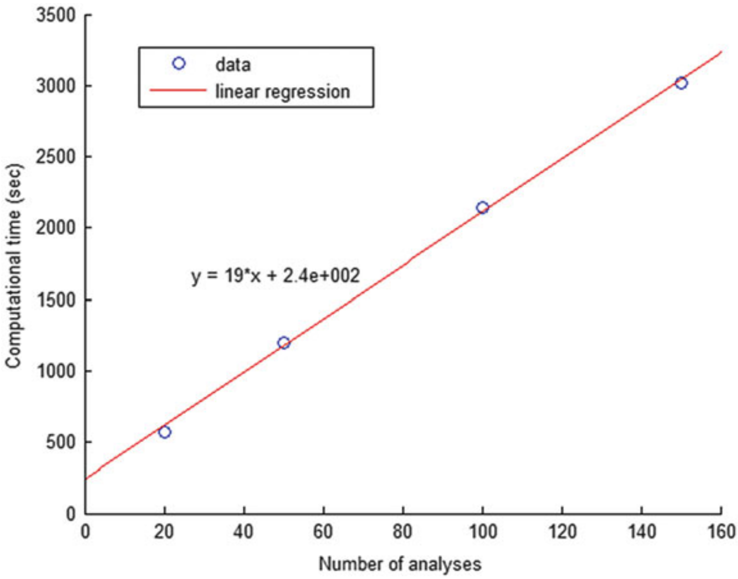


Fig. 14.12 Variation of the computational time with a number of analyses for Case 1 (1410-bar dome truss)

It takes 38,310.43 s to perform ten optimization runs using the efficient method for this example, while it would have taken approximately 9,386,055 s (about 108 days) to perform the same runs using the classical method. Table 14.11 presents the first five natural frequencies of the optimized structure for this example. The mean weight of the structures found in ten runs is 38,294.45 kg with a standard deviation of 550.5 kg. Figure 14.13 shows the convergence curve of the best result for the 1410-bar dome truss using the efficient method.

Table 14.10 Optimized design for the 1410-bar dome truss problem (added masses are not included)

Element no. (nodes)	Cross-sectional area (cm^2)	Element no. (nodes)	Cross-sectional area (cm^2)
1 (1–2)	7.209	25 (8–9)	2.115
2 (1–8)	5.006	26 (8–14)	4.923
3 (1–14)	38.446	27 (8–15)	4.047
4 (2–3)	9.438	28 (8–21)	5.906
5 (2–8)	4.313	29 (9–10)	3.392
6 (2–9)	1.494	30 (9–15)	1.902
7 (2–15)	8.455	31 (9–16)	4.381
8 (3–4)	9.488	32 (9–22)	8.442
9 (3–9)	3.480	33 (10–11)	5.011
10 (3–10)	3.495	34 (10–16)	3.577
11 (3–16)	16.037	35 (10–17)	2.805
12 (4–5)	9.796	36 (10–23)	2.024
13 (4–10)	2.413	37 (11–12)	6.709
14 (4–11)	5.681	38 (11–17)	5.054
15 (4–17)	15.806	39 (11–18)	3.259
16 (5–6)	8.078	40 (11–24)	1.063
17 (5–11)	3.931	41 (12–13)	5.934
18 (5–12)	6.099	42 (12–18)	7.057
19 (5–18)	10.771	43 (12–19)	5.745
20 (6–7)	13.775	44 (12–25)	1.185
21 (6–12)	4.231	45 (13–19)	7.274
22 (6–13)	6.995	46 (13–20)	4.798
23 (6–19)	1.837	47 (13–26)	1.515
24 (7–13)	4.397	Weight (kg)	10,453.84

Table 14.11 Natural frequencies (Hz) evaluated at the optimized design for the 1410-bar dome truss problem

Frequency number	Frequency value
1	7.001
2	7.001
3	9.003
4	9.005
5	9.005

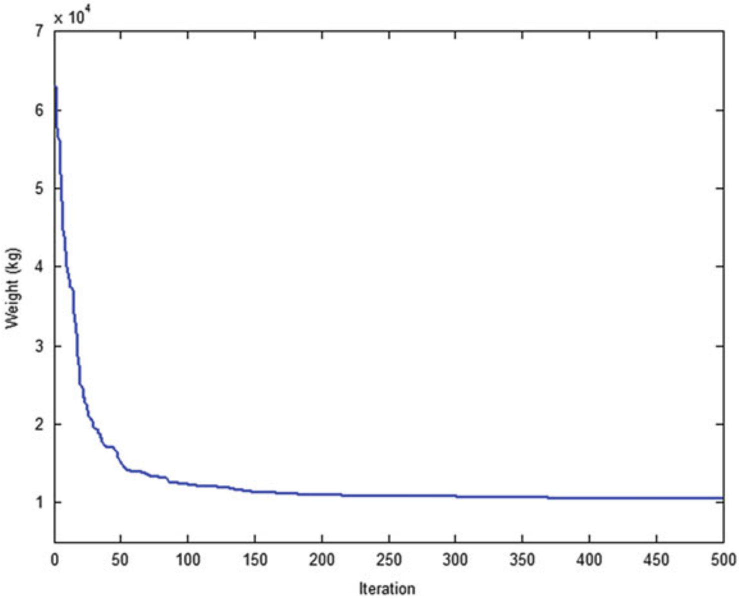


Fig. 14.13 Convergence curve of the best result for the 1410-bar dome truss using the efficient method [1]

14.6 Concluding Remarks

Structural optimization using metaheuristic algorithms involves a large number of structural analyses, which requires a great amount of computational time, especially when optimizing large structural systems. In this chapter simultaneous optimal analysis and design of cyclically repetitive dome trusses with frequency constraints are considered. These types of structures exhibit some favorable patterns in their structural matrices, which makes it possible to utilize some efficient analysis methods. These methods decompose the original eigenproblem into several smaller ones, which are simpler to solve and require less computational time. Democratic particle swarm optimization (DPSO) introduced by Kaveh and Zolghadr [14] is utilized as the optimization algorithm.

Three different dome trusses are considered as numerical examples to show the efficiency of the proposed method. It can be seen that using the efficient method for analysis, the optimization procedure can be performed significantly faster. While all the runs are taken in <2 days using the efficient methods, it would have taken more than 200 days to do the same thing using classical methods. Such a substantial saving in computational time is due to the regular nature of the structures under consideration. Other types of efficient methods could also be used in order to deal with near-regular structures (Kaveh [17]).

The presented concepts can be generalized to optimization of other types of symmetric or regular structures as well as structural optimization with static constraints.

References

1. Kaveh A, Zolghadr A (2016) Optimal analysis and design of large-scale domes with frequency constraints. Smart Struct Syst 18(5):733–754
2. Grandhi RV (1993) Structural optimization with frequency constraints—a review. AIAA J 31 (12):2296–2303
3. Taylor JE (1967) Minimum mass bar for axial vibration at specified natural frequency. AIAA J 5(10):1911–1913
4. Armand JL (1971) Minimum mass design of a plate like structure for specified fundamental frequency. AIAA J 9(9):1739–1745
5. Cardou A, Warner WH (1974) Minimum mass design of sandwich structures with frequency and section constraints. J Optim Theory Appl 14(6):633–647
6. Elwany MHS, Barr ADS (1979) Minimum weight design of beams in torsional vibration with several frequency constraints. J Sound Vib 62(3):411–425
7. Lin JH, Chen WY, Yu YS (1982) Structural optimization on geometrical configuration and element sizing with static and dynamic constraints. Comput Struct 15(5):507–515
8. Konzelman CJ (1986) Dual methods and approximation concepts for structural optimization. MSc thesis, Department of Mechanical Engineering, University of Toronto, Canada
9. Grandhi RV, Venkayya VB (1988) Structural optimization with frequency constraints. AIAA J 26(7):858–866
10. Sedaghati R, Suleman A, Tabarrok B (2002) Structural optimization with frequency constraints using finite element force method. AIAA J 40(2):382–388
11. Lingyun W, Mei Z, Guangming W, Guang M (2005) Truss optimization on shape and sizing with frequency constraints based on genetic algorithm. J Comput Mech 35(5):361–368
12. Gomes MH (2011) Truss optimization with dynamic constraints using a particle swarm algorithm. Expert Syst Appl 38(1):957–968
13. Kaveh A, Zolghadr A (2014) Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints. Adv Eng Softw 76:9–30
14. Kaveh A, Zolghadr A (2014) Democratic PSO for truss layout and size optimization with frequency constraints. Comput Struct 130:10–21
15. Kaveh A, Rahami H (2006) Block diagonalization of adjacency and Laplacian matrices for graph product; applications in structural mechanics. Int J Numer Methods Eng 68(1):33–63
16. Kaveh A, Rahami H (2007) Compound matrix block diagonalization for efficient solution of eigenproblems in structural mechanics. Acta Mech 188(3–4):155–166
17. Kaveh A (2013) Optimal analysis of structures by concepts of symmetry and regularity. Springer, Wien, New York, NY
18. Koohestani K, Kaveh A (2010) Efficient buckling and free vibration analysis of cyclically repeated space truss structures. Finite Elem Anal Des 46(10):943–948
19. Courant R (1943) Variational methods for the solution of problems of equilibrium and vibrations. Bull Am Math Soc 49:1–23
20. Leung AYT (1980) Dynamic analysis of periodic structures. J Sound Vib 72(4):451–467
21. Williams FW (1986) An algorithm for exact eigenvalue calculations for rotationally periodic structures. Int J Numer Methods Eng 23(4):609–622
22. Vakakis AF (1992) Dynamics of a nonlinear periodic structure with cyclic symmetry. Acta Mech 95(1–4):197–226

23. Karpov EG, Stephen NG, Dorofeev DL (2002) On static analysis of finite repetitive structures by discrete Fourier transform. Int J Solids Struct 39(16):4291–4310
24. Liu L, Yang H (2007) A paralleled element-free Galerkin analysis for structures with cyclic symmetry. Eng Comput 23(2):137–144
25. El-Raheb M (2011) Modal properties of a cyclic symmetric hexagon lattice. Comput Struct 89:2249–2260
26. Zingoni A (2014) Group-theoretic insights on the vibration of symmetric structures in engineering. Philos Trans R Soc A 372:20120037
27. Tran D-M (2014) Reduced models of multi-stage cyclic structures using cyclic symmetry reduction and component mode synthesis. J Sound Vib 333:5443–5463

Chapter 15
Optimum Design of Large-Scale Truss Towers Using Cascade Optimization

15.1 Introduction

High number of design variables, large size of the search space, and control of a great number of design constraints are major preventive factors in performing optimum design of real-world structures in a reasonable time. This chapter presents an accurate and efficient technique for optimal design of truss towers with large number of design variables to illustrate its applicability to optimum design of practical structures [1].

Cascade sizing optimization utilizing a series of design variable configurations (DVCs) is used in this study. Several DVCs are constructed in order to utilize a different configuration at each cascade optimization stage. Each new cascade stage is coupled with the previous one by initializing the new stage using the finally attained optimum design of the previous stage. The first stages of the cascade procedure are executed with the coarsest DVCs, and the final cascade stages utilize the finest DVCs in order to handle large numbers of design variables. In all stages of the procedure, enhanced colliding bodies optimization (ECBO) is employed. The multi-DVC cascade optimization performs better than non-cascade procedure in all the considered examples. High solution accuracy and convergence speed of the proposed method are shown through three test examples [1].

In the last decades, a number of optimization techniques have been developed and used for structural optimization problems. The aim of the optimization is to minimize an objective function that is often considered as the cost of the structure or a quantity directly proportional to the cost under certain constraints. These constraints may consist of any engineering demand parameter like stresses, displacements, maximum inter-story drift, etc. Recent years have witnessed an increasing interest in the development and application of metaheuristic algorithms that are effective and robust techniques for optimization problems. These algorithms are often population based, and they search for the global optimum of the problem through sharing information to cooperate and/or compete among the

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_15

individuals. Many of the recently developed metaheuristic algorithms for optimal design of structures can be found in Kaveh [2].

Structural optimization has grown from a narrow academic discipline, where researchers focused on optimum design of small idealized structural components and systems, to form the basis for modern design of complex structural systems [3]. On the other hand, optimal design of large-scale structures is a rather difficult task and the computational efficiency of the currently available methods needs to be improved. In this chapter, optimal design of three truss towers with 582, 942, and 2386 elements is studied in the framework of cascade evolutionary structural sizing optimization for presenting the efficiency of this technique in solving large-scale truss tower problems. In this method, several DVCs are constructed, in order to utilize a different configuration at each cascade optimization stage. Each new cascade stage is coupled with the previous one by initializing the new stage using the finally attained optimum design of the previous stage. The early optimization stages of the cascade procedure make use of the coarsest configurations with small numbers of design variables and serve the purpose of basic design space exploration. The later stages exploit finer configurations with larger numbers of design variables and aim at fine-tuning of the achieved optimal solution [4].

In general, the optimization algorithm utilized at each stage of a cascade process may or may not be the same. In this chapter, ECBO (Kaveh and Ilchi Ghazaan [5]) is utilized in all stages of a cascade process. In this technique, one object collides with another and they move toward a minimum energy level. The CBO developed by Kaveh and Mahdavi [6] has a simple theoretical structure, usually converges quickly, and depends on no internal parameters. By using memory to save a number of historically best solutions and also utilizing a mechanism to escape from local optima, ECBO usually performs better than CBO (Kaveh and Ilchi Ghazaan [7]).

The rest of this chapter is organized as follows: In Sect. 15.2, description of the cascade sizing optimization method that employs a series of configurations is presented. In Sect. 15.3, the ECBO algorithm is presented in detail. Then Sect. 15.4 uses numerical examples to confirm the validity of the proposed approach. Finally, concluding remarks are provided in Sect. 15.5.

15.2 Cascade Sizing Optimization Utilizing Series of Design Variable Configurations

In this section, the multi-DVC cascade optimization is presented after a brief introduction to the concept of the cascade optimization.

15.2.1 Concept of Cascade Optimization

No single optimizer can successfully solve all the structural design problems. Cascade optimization strategy was proposed to alleviate this deficiency which utilizes several optimizers, one followed by another in a specified sequence, to solve a problem [8]. In the first stage of the cascade procedure, the first optimizer starts from a user-specified design, known as the "cold-start." The intermediate optimal solution reached in the first cascade stage, which may be perturbed using a pseudo-random technique, is called a "hot-start" and is used to initiate the second optimization stage. Accordingly, each optimization stage of the cascade procedure starts from the optimum design achieved in the previous stage (possibly perturbed). Thus, each cascade stage initiates from a hot-start and produces a new hot-start for the next stage. This way the autonomous computations of successive optimization stages are coupled. In general, the optimization algorithm implemented at each stage of a cascade process may or may not be the same. Cascade optimization has been implemented using different deterministic and/or probabilistic optimizers in the cascade stages (Charmpis et al. [3]).

15.2.2 Multi-DVC Cascade Optimization

A series of appropriate DVCs for the sizing optimization problem under consideration is formed, in order to utilize a different configuration at each cascade optimization stage. Each new cascade stage is coupled with the previous one by initializing the new stage using the finally attained optimum design of the previous one (Charmpis et al. [4]). The first stages of the cascade procedure are executed with the coarsest DVCs aiming at a basic non-detailed search of the full design space. This search is facilitated by the manageable DVCs handled to avoid confusing the optimizer with huge design spaces. Thus, the areas of appropriate design variable values are identified by detecting near optimum solutions among the relatively limited design options provided. As the numbers of design variables processed in the cascade stages become larger, more detailed representation of the full design space is offered and the optimizer is given the opportunity to improve the quality of the optimal solution reached. In the final cascade stages utilizing the finest DVCs, relatively small adjustments to an already good-quality design occur, in an effort to identify (or at least approach) the globally optimum design. Hence, the early optimization stages of the cascade procedure serve the purpose of basic design space exploration, while the later stages aim at fine-tuning of the achieved optimal solution (Charmpis et al. [4]).

This multi-DVC cascade computational procedure can be implemented using an arbitrary optimization algorithm. In this study, ECBO (Kaveh and Ilchi Ghazaan [7]) that is presented in the next section is utilized in all stages. Flowchart of the Multi-DVC cascade optimization procedure is shown in Fig. 15.1.

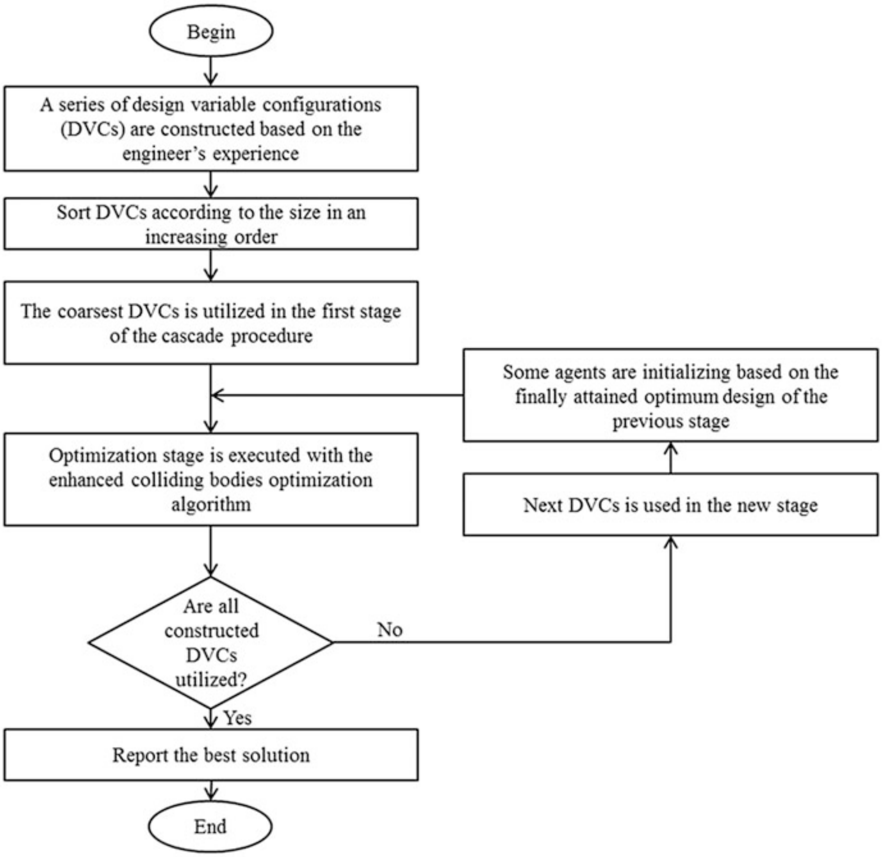


Fig. 15.1 Flowchart of the multi-DVC cascade optimization procedure

15.3 Enhanced Colliding Bodies Optimization

Colliding bodies optimization (CBO) is a population-based metaheuristic algorithm introduced by Kaveh and Mahdavi [6]. This method originates from one-dimensional collisions between two bodies in which one object collides with the another and they move toward minimum energy level. The movement process of the objects is based on the governing laws of collision in physics. ECBO was proposed by Kaveh and Ilchi Ghazaan [7] that utilizes a memory to store a certain number of best designs obtained so far to improve convergence behavior of CBO. Furthermore, some components of agents are randomly changed to allow them to escape from local minima and prevent premature convergence. This algorithm consists of the following steps:

Step 1: Initialization

Each solution candidate x_i is considered as a colliding body (CB) and the initial positions of all CBs are determined randomly in an m-dimensional search space.

$$x_i^0 = x_{\min} + rand \circ (x_{\max} - x_{\min}), \qquad i = 1, 2, \ldots, n \tag{15.1}$$

where x_i^0 is the initial solution vector of the ith CB. Here, $x_{\min}$ and $x_{\max}$ are the bounds of design variables; *rand* is a random vector in which each component is in the interval [0, 1]; the sign "∘" denotes an element-by-element multiplication; n is the number of CBs.

Step 2: Defining mass

Each CB has a specified mass defined as

$$m_k = \frac{\frac{1}{\mathrm{fit}(k)}}{\frac{1}{\sum_{i=1}^{n} \frac{1}{\mathrm{fit}(i)}}}, \qquad k = 1,\ 2,\ \ldots,\ n \tag{15.2}$$

where fit(i) represents the objective function value of the ith CB.

Step 3: Saving

Colliding memory (CM) is utilized to save a number of the best-so-far solutions. In this study, the size of the CM is taken as $n/10$. At each iteration, solution vectors saved in CM are added to the population, and the same number of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: Creating groups

In order to select pairs of objects for collision, CBs are divided into two equal groups: (i) stationary group and (ii) moving group. Moving objects collide with stationary objects to improve their positions and push stationary objects toward better positions.

Step 5: Criteria before the collision

The velocity of the stationary bodies before collision is zero so

$$v_i = 0, \qquad i = 1, 2, \ldots, \frac{n}{2} \tag{15.3}$$

The velocity of each moving body before collision is

$$v_i = x_{i-\frac{n}{2}} - x_i, \qquad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{15.4}$$

Step 6: Criteria after the collision

The velocity of each stationary CB after the collision (v_i') is specified by

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} \qquad i = 1, 2, \ \ldots \ , \ \frac{n}{2} \tag{15.5}$$

The velocity of each moving CB after the collision (v_i') is defined by

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}} \qquad i = \frac{n}{2} + 1 \ , \ \frac{n}{2} + 2 \ , \ \ldots \ , \ n \tag{15.6}$$

ε is the coefficient of restitution (COR) that decreases linearly from unit to zero

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{15.7}$$

where *iter* is the current iteration number and $iter_{max}$ is the total number of iterations for optimization process.

Step 7: Updating CBs

New positions of CBs are updated according to their velocities after the collision and the positions of stationary CBs. Therefore, the new position of each stationary CB is

$$x_i^{\text{new}} = x_i + rand \circ v_i', \qquad i = 1, 2, \ldots, \frac{n}{2} \tag{15.8}$$

New position of each moving CB is calculated by

$$x_i^{\text{new}} = x_{i-\frac{n}{2}} + rand \circ v_i', \qquad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{15.9}$$

Step 8: Escape from local optima

A parameter like ***Pro*** within (0, 1) is introduced which specifies whether a component of each CB must be changed or not. For each colliding body, ***Pro*** is compared with rn_i ($i = 1, 2, \ldots, n$) which is a random number uniformly distributed within (0, 1). If $rn_i <$ ***pro***, one dimension of the ith CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + rnd.\left(x_{j,\max} - x_{j,\min}\right) \tag{15.10}$$

where x_{ij} is the jth variable of the ith CB. $x_{j,\min}$ and $x_{j,\max}$, respectively, are the lower and upper bounds of the jth variable. *rnd* is a random number in the interval [0,1]. In this study, ***pro*** is set to 0.25.

Step 9: Termination condition check

After the predefined maximum evaluation number, the optimization process is terminated.

15.4 Design Examples

Three large-scale truss structures are optimized for minimum volume with the cross-sectional areas of the members being the design variables to verify the efficiency of the multi-DVC cascade optimization. A population of 20 CBs is used for the first and second examples and 30 CBs are utilized for the last problem. The optimization process in each stage except the last one is terminated after a fixed number of iterations without any improvement. This value is considered as the minimum of the number of design variables in the stage as 30. When the total number of iterations is equal to 1000, the process is terminated. In all problems, the CBs are allowed to select discrete values from the permissible list of cross sections (real numbers are rounded to the nearest integer in the each iteration). The well-known penalty approach is employed to handle the constraints (Kaveh and Ilchi Ghazaan [7]). The algorithms are coded in MATLAB, and the structures are analyzed using the direct stiffness method.

15.4.1 A Spatial 582-Bar Tower

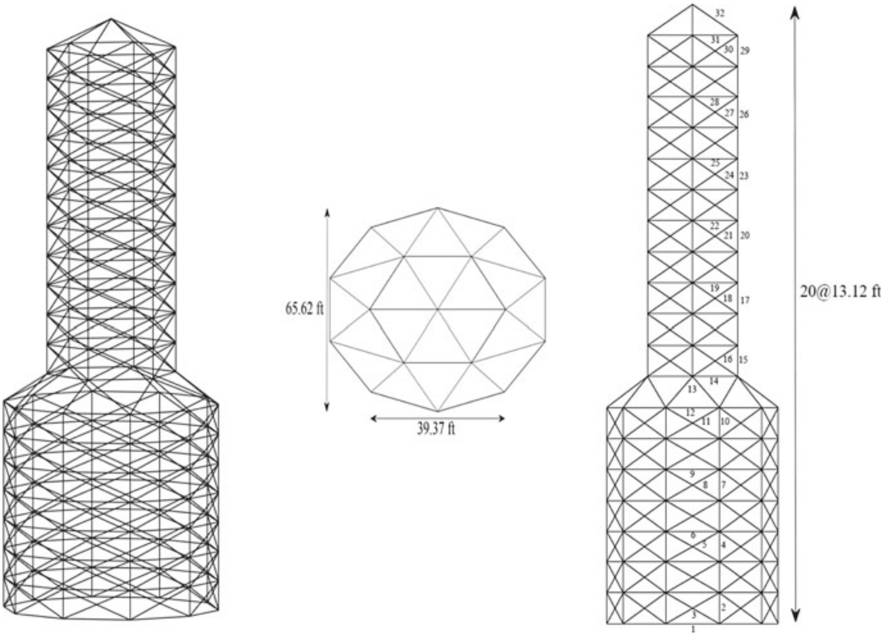
The schematic of a 582-bar tower truss is shown in Fig. 15.2 as a well-known benchmark problem. The symmetry of the tower about x-axis and y-axis is considered to group the 582 members into 32 independent sizing variables. A single load case is considered consisting of the lateral loads of 1.12 kips (5.0 kN) applied in both x- and y-directions and a vertical load of −6.74 kips (−30 kN) applied in the z-direction at all nodes of the tower. A discrete set of standard steel sections selected from W-shape profile list based on area and radii of gyration properties is used as sizing variables. Cross-sectional areas of elements can vary between 6.16 and 215 in^2 (i.e., between 39.74 and 1387.09 cm^2). Limitations on the stress and stability of truss elements are imposed according to the provisions of ASD-AISC [9] as follows.

The allowable tensile stresses for tension members are calculated by

$$\sigma_i^+ = 0.6F_y \tag{15.11}$$

where F_y stands for the yield strength.

The allowable stress limits for compression members are calculated depending on two possible failure modes of the members known as elastic and inelastic buckling. Thus,



3D view Top view Side view

Fig. 15.2 Schematic of the spatial 582-bar tower

$$\sigma_i^- = \begin{cases} \left[\left(1 - \dfrac{\lambda_i^2}{2C_\mathrm{c}^2}\right)F_\mathrm{y}\right] \Big/ \left[\dfrac{5}{3} + \dfrac{3\lambda_i}{8C_\mathrm{c}} - \dfrac{\lambda_i^3}{8C_\mathrm{c}^3}\right] & \text{for } \lambda_\mathrm{i} < C \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_\mathrm{i} \geq C_\mathrm{c} \end{cases} \tag{15.12}$$

where E is the modulus of elasticity, λ_i is the slenderness ratio $(\lambda_i = kl_i/r_i)$, C_c denotes the slenderness ratio dividing the elastic and inelastic buckling regions $\left(c_\mathrm{c} = \sqrt{2\pi^2 E/F_\mathrm{y}}\right)$, k is the effective length factor (k is set 1 for all truss members), L_i is the member length, and r_i is the minimum radius of gyration.

In this design code provisions, the maximum slenderness ratio is limited to 300 for tension members, and it is recommended to be 200 for compression members.

Nodal displacements in all coordinate directions must be less than ± 3.15 in (i.e., ± 8 cm).

This problem is optimized in 3 stages. The number of design variable in stages 1, 2, and 3 are 8, 15, and 32, respectively. Table 15.1 presents the DVCs. The multi-DVC cascade optimization procedure achieves 1,295,779 m^3 after 18,700 analyses. This problem was previously solved by ECBO, and it obtained 1,296,776 m^3 after 19,700 analyses (Kaveh and Ilchi Ghazaan [5]). The required number of analyses to achieve 0.5 % heavier designs than the optimal design for non-cascade and cascade

Table 15.1 Design variable configurations utilized for the 582-bar tower problem

	Number of design variables in stages	Design variables in the group (design variable configurations)
Stage 1	8	[1 6 9]; [2 4 7 10]; [3 5 8 11]; [12 13 14]; [19 22 25 28 31]; [32]; [15 17 20 23 26 29]; [16 18 21 24 27 30]
Stage 2	15	[1 6 9]; [2 4]; [7 10]; [3 5]; [8 11]; [12]; [13]; [14]; [19 22 25]; [28 31]; [32]; [15 17 20]; [23 26 29]; [16 18 21]; [24 27 30]

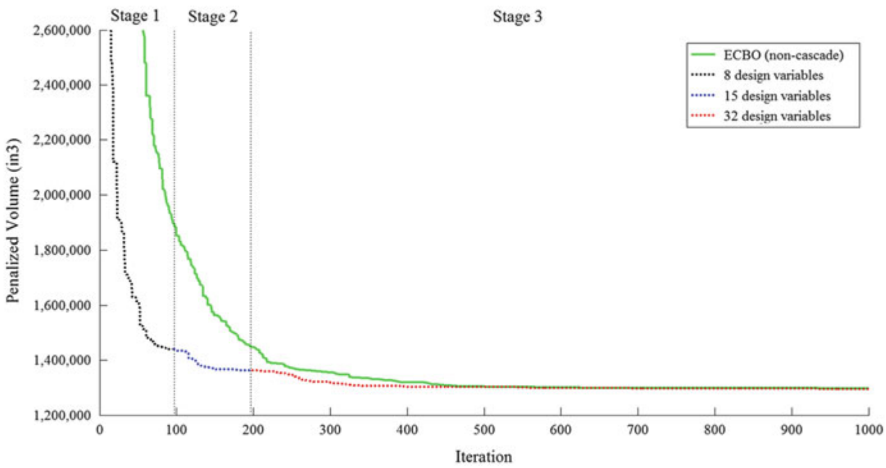


Fig. 15.3 Convergence curves of non-cascade (*solid lines*) and cascade optimization procedures (*dotted lines*) obtained in the 582-bar tower problem [1]

optimization procedures are 8980 and 6620 analyses, respectively. It means that the algorithm manages to find a near optimal solution in the early iterations while it continues searching the search space until the last iterations. Convergence curves are depicted in Fig. 15.3. The final volumes achieved in stage 1 (containing 8 design variables) and stage 2 (containing 15 design variables) are 1,438,697 m^3 and 1,363,348 m^3, respectively. These stages are terminated in 95th and 197th iterations. It can be seen that the convergence rate of the cascade optimization procedures is higher than the non-cascade procedure.

15.4.2 A Spatial 942-Bar Tower

Figure 15.4 shows the schematic of a 942-bar tower truss. This example has been analyzed by many researchers considering 59 design variables (Hasançebi [10]). In this study, the design variables are increased to 76 and the performance constraints, material properties, and other conditions are the same as those of the first example.

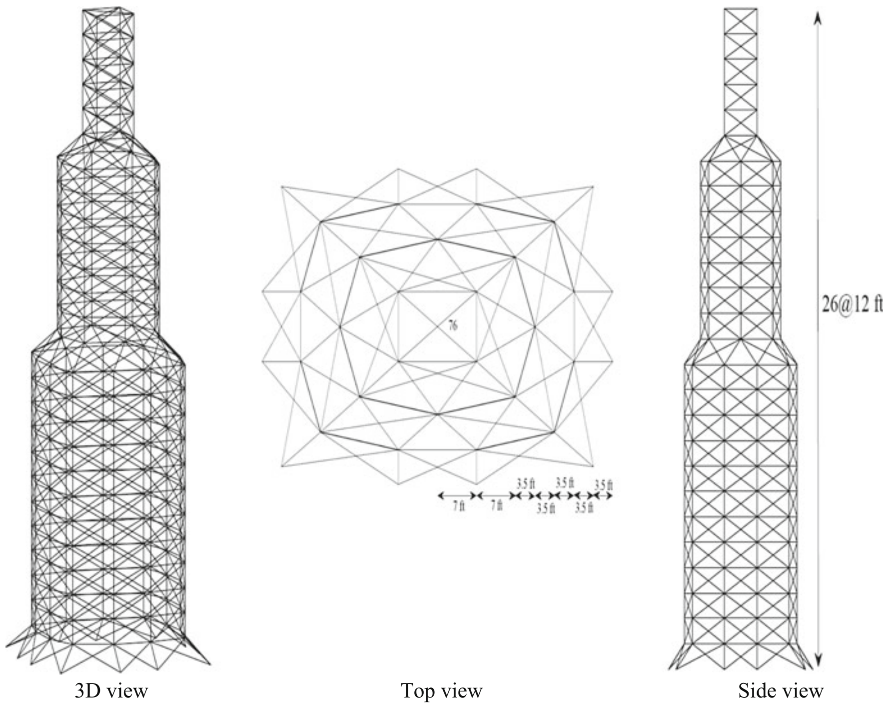


Fig. 15.4 Schematic of the spatial 942-bar tower

Figure 15.5 shows the member groups. Three stages with 16, 28, and 76 design variables are considered to solve this problem. The DVCs are shown in Table 15.2.

The design obtained by cascade optimization procedures is 3,323,028 m^3, and the best design attained without cascading is 3,376,968 m^3. These values are found after 18,320 and 19,960 analyses, respectively. The proposed method can reach the best design of non-cascade procedure after about 11,060 analyses. Convergence history diagrams are depicted in Fig. 15.6. The final volume found in stage 1 (containing 16 design variables) in 112th iteration is 4,467,989 m^3. Stage 2 (containing 28 design variables) terminated in 287th iteration and its corresponding value is 3,809,870 m^3. It can be seen that the curve of the multi-DVC cascade optimization lies below those of the non-cascading procedure.

15.4.3 A Spatial 2386-Bar Tower

The schematic of a 2386-bar tower truss is shown in Fig. 15.7 as the last design example. This example is studied here for the first time. The Performance constraints, material properties, and other conditions are the same as those of the first

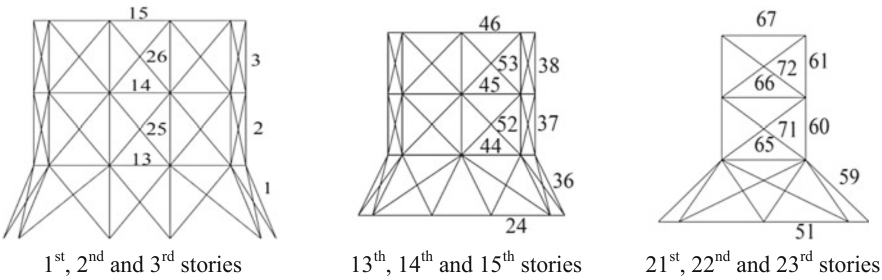


Fig. 15.5 Member groups of spatial 942-bar tower

Table 15.2 Design variable configurations utilized for the 942-bar tower problem

	Number of design variables in stages	Design variables in the group (design variable configurations)
Stage 1	16	[1]; [2–6]; [7–12]; [13–18]; [19–24]; [25–29]; [30–35]; [36]; [37–43]; [44–51]; [52–58]; [59]; [60–64]; [65–70]; [71–75]; [76]
Stage 2	28	[1]; [2 3]; [4–6]; [7–9]; [10–12]; [13–15]; [16–18]; [19–21]; [22–24]; [25 26]; [27–29]; [30–32]; [33–35]; [36]; [37–39]; [40–43]; [44–47]; [48–51]; [52–54]; [55–58]; [59]; [60 61]; [62–64]; [65–67]; [68–70]; [71 72]; [73–75]; [76]

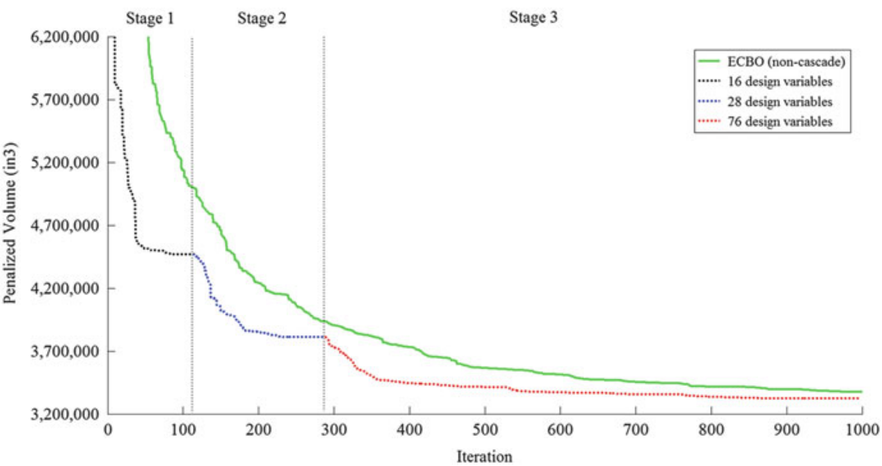


Fig. 15.6 Convergence curves of non-cascade (*solid lines*) and cascade optimization procedures (*dotted lines*) obtained in the 942-bar tower problem [1]

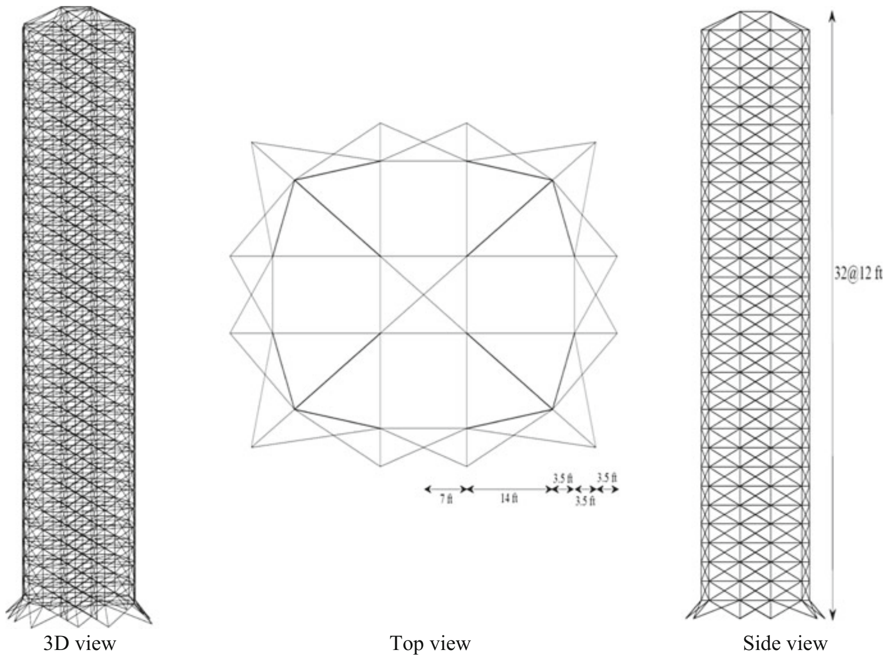


Fig. 15.7 Schematic of the spatial 2386-bar tower

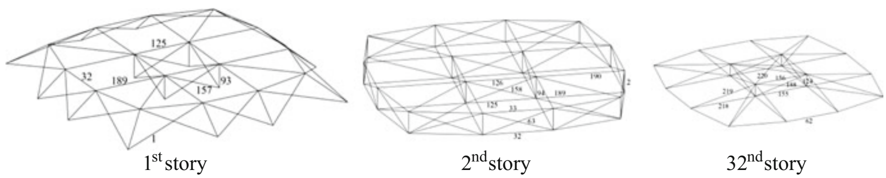


Fig. 15.8 Member groups of spatial 2386-bar tower

example. The elements are divided into 220 groups and member groups are presented in Fig. 15.8. Four stages are considered to optimize this example. The number of design variables in stages 1, 2, 3, and 4 are 21, 42, 84, and 220, respectively. Table 15.3 lists the DVCs.

The proposed method obtained 12,535,919 m^3 after 29,010 analyses which is better than 14,086,857 m^3 found by the non-cascade procedure after 29,670 analyses. The best design of non-cascade procedure can be achieved by multi-DVC cascade optimization after only 6900 analyses. Convergence curves are compared in Fig. 15.9. The final volumes achieved in stage 1 (containing 21 design variables), stage 2 (containing 42 design variables), and stage 3 (containing 84 design variables) are 14,504,868 m^3, 13,416,104 m^3, and 12,862,132 m^3, respectively. These stages are terminated in 225th, 501th, and 761th iterations. It can be seen from the

Table 15.3 Design variable configurations utilized for the 2386-bar tower problem

	Number of design variables in stages	Design variables in the group (design variable configurations)
Stage 1	21	[1–10]; [11–20]; [21–31]; [32–41]; [42–51]; [52–62]; [63–72]; [73–82]; [83–92] [93–103]; [104–113]; [114–124]; [125–135]; [136–146]; [147–156]; [157–167] [168–178]; [179–188]; [189–199]; [200–210]; [211–220]
Stage 2	42	[1:4]; [5:10]; [11:15]; [16:20]; [21:25]; [26:31]; [32:36]; [37:41]; [42:46]; [47:51]; [52:56]; [57:62]; [63:67]; [68:72]; [73:77]; [78:82]; [83:87]; [88:92]; [93:97]; [98:103]; [104:108]; [109:113]; [114:118]; [119:124]; [125:129]; [130:135]; [136:140]; [141:146]; [147:151]; [152:156]; [157:161]; [162:167]; [168:172]; [173:178]; [179:183]; [184:188]; [189:193]; [194:199]; [200:204]; [205:210]; [211:215]; [216:220]
Stage 3	84	[1]; [2–4]; [5–7]; [8–10]; [11 12]; [13–15]; [16 17]; [18–20]; [21 22]; [23–25]; [26–28]; [29–31]; [32 33]; [34–36]; [37 38]; [39–41]; [42 43]; [44–46]; [47 48]; [49–51]; [52 53]; [54–56]; [57–59]; [60–62]; [63 64]; [65–67]; [68 69]; [70–72]; [73 74]; [75–77]; [78 79]; [80–82]; [83 84]; [85–87]; [88 89]; [90–92]; [93 94]; [95–97]; [98–100]; [101–103]; [104 105]; [106–108]; [109 110]; [111–113]; [114 115]; [116–118]; [119–121]; [122–124]; [125 126]; [127–129]; [130–132]; [133–135]; [136 137]; [138–140]; [141–143]; [144–146]; [147 148]; [149–151]; [152 153]; [154–156]; [157 158]; [159–161]; [162–164]; [165–167]; [168 169]; [170–172]; [173–175]; [176–178]; [179 180]; [181–183]; [184 185]; [186–188]; [189 190]; [191–193]; [194–196]; [197–199]; [200 201]; [202–204]; [205–207]; [208–210]; [211 212]; [213–215]; [216 217]; [218–220]

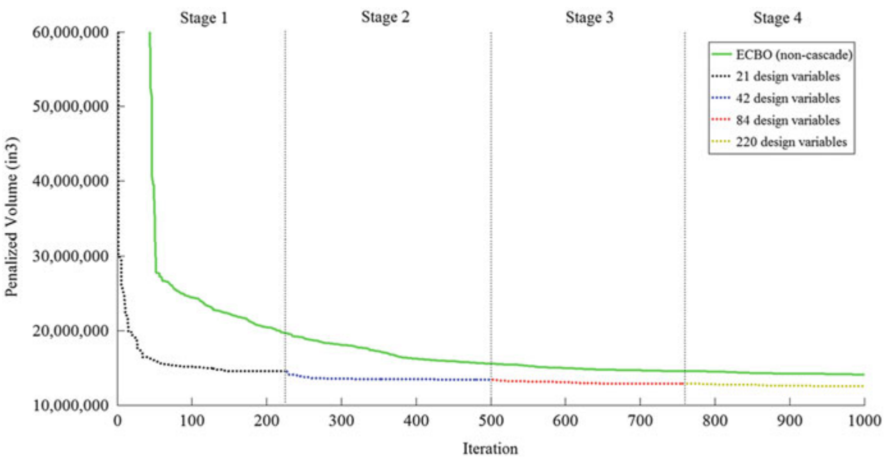


Fig. 15.9 Convergence curves of non-cascade (*solid lines*) and cascade optimization procedures (*dotted lines*) obtained in the 2386-bar tower problem [1]

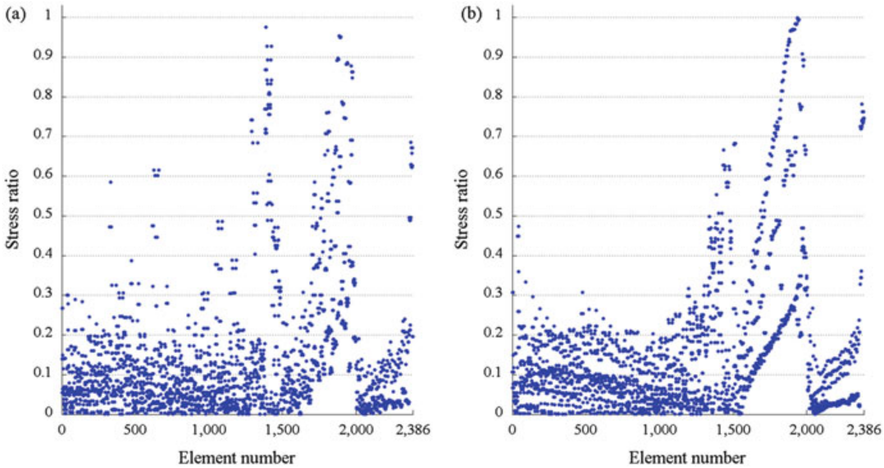


Fig. 15.10 Element stress ratio obtained in the 2386-bar tower problem: (**a**) non-cascade optimization procedures and (**b**) cascade optimization procedures

plots that the intermediate designs found by proposed method are always better than those found by non-cascade procedure. The stress ratios for all the members are shown in Fig. 15.10. The maximum values of the stress ratio for non-cascade and cascade procedures are 97.57 % and 99.96 %, respectively.

15.5 Concluding Remarks

Three numerical examples chosen from size optimum design of truss towers with large number of design variables are studied to test and verify efficiency of the multi-DVC cascade optimization that utilizes a different DVC in each stage of the cascade optimization procedure, as well as to illustrate its applicability for optimum design of practical structures. In the 32-variable design example, the best volumes obtained by non-cascade and cascade optimization procedures were approximately the same, but cascade optimization procedure had a better convergence rate. The optimum volume found by cascade optimization procedure in the 76-variable design example was about 2 % lighter than that obtained by non-cascade procedure. Also, the required number of iterations for achieving the best design of non-cascade procedure was also decreased 50 % by the proposed method. In the 220-variable design example, the design obtained by the cascade optimization procedures is about 11 % lighter than the best design attained without cascading. The required number of iterations for achieving the best design of non-cascade procedure was also decreased 50 % by the proposed method. It can be concluded that by increasing the size of the search space, the differences between the accuracy of the cascading and non-cascading procedures considerably increase. To sum up, multi-DVC cascade optimization can be considered as a fast and reliable method in handling large number of design variables and corresponding design spaces in the context of size optimization problems.

References

1. Kaveh A, Ilchi Ghazaan M (2016) Optimum design of large-scale truss towers using cascade optimization. Acta Mech 227(9):2645–2656
2. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer, Switzerland
3. Charmpis DC, Lagaros ND, Papadrakakis M (2005) Multi-database exploration of large design spaces in the framework of cascade evolutionary structural sizing optimization. Comput Methods Appl Mech Eng 194:3315–3330
4. Charmpis DC, Lagaros ND, Papadrakakis M (2015) Cascade structural sizing optimization with large numbers of design variables. Adv Eng Softw (submitted)
5. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
6. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
7. Kaveh A, Ilchi Ghazaan M (2015) A comparative study of CBO and ECBO for optimal design of skeletal structures. Comput Struct 153:137–147
8. Patnaik SN, Coroneos RM, Hopkins DA (1997) A cascade optimization strategy for solution of difficult design problems. Int J Numer Methods Eng 40:2257–2266
9. American Institute of Steel Construction (AISC) (1989) Manual of steel construction—allowable stress design, 9th edn. AISC, Chicago, IL
10. Hasançebi O (2008) Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. Comput Struct 86:119–132

Chapter 16
Vibrating Particles System Algorithm for Truss Optimization with Frequency Constraints

16.1 Introduction

In this chapter a recently developed physics-inspired non-gradient algorithm is employed for structural optimization with frequency constraints. The algorithm being called vibrating particles system (VPS) mimics the free vibration of single degree of freedom systems with viscous damping. Truss optimization with frequency constraints is believed to represent nonlinear and non-convex search spaces with several local optima and therefore is suitable for examining the capabilities of the new algorithms. A set of five truss design problems are considered for evaluating the VPS in this article. The numerical results demonstrate the efficiency and robustness of the new method (Kaveh and Ilchi Ghazaan [1]).

Fundamental frequencies of a structure are important, easily obtained characteristics which allow the designer to keep out from the dangerous resonance phenomenon. When dynamic excitations are critical, these characteristics cannot be neglected. Frequency responses are highly implicit, non-convex, and nonlinear with respect to the cross-sectional area of bar elements, so the search spaces normally contain multiple local minima [2] and call for a competent optimization algorithm in order to be appropriately addressed.

Structural optimization considering natural frequency constraints has been studied since the 1980s [3] using mathematical programming and metaheuristic algorithms. Lin et al. [4] studied the minimum weight design of structures under simultaneous static and dynamic constraints proposing a bi-factor algorithm based on the Kuhn–Tucker criteria. Konzelman [5] considered the problem using some dual methods and approximation concepts for structural optimization. Grandhi and Venkayya [6] utilized an optimality criterion based on uniform Lagrangian density for resizing and scaling procedure to locate the constraint boundary. Wang et al. [7] proposed an optimality criterion algorithm for combined sizing–layout optimization of three-dimensional truss structures. In this method, the sensitivity analysis helps to determine the search direction, and the

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_16

optimal solution is achieved gradually from an infeasible starting point with a minimum weight increment, and the structural weight is indirectly minimized. Sedaghati [8] utilized a new approach using combined mathematical programming based on the sequential quadratic programming (SQP) technique and a finite element solver based on the integrated force method. Lingyun et al. [9] combined the simplex search method and the niche genetic hybrid algorithm (NGHA) for mass minimization of structures with frequency constraints. Gomes [10] used the particle swarm optimization (PSO) algorithm to study simultaneous layout and sizing optimization of truss structures with multiple frequency constraints. Kaveh and Zolghadr [11] combined charged system search and Big Bang-Big Crunch with trap recognition capability (CSS–BBBC) to solve layout and sizing optimization problems of truss structures with natural frequency constraints. Miguel and Fadel Miguel [12] employed harmony search (HS) and firefly algorithm (FA) to study simultaneous layout and sizing optimization of truss structures with multiple frequency constraints. A hybrid optimality criterion (OC) and genetic algorithm (GA) method was used by Zuo et al. [13] for truss optimization with frequency constraints. Kaveh and Javadi [14] utilized hybridization of harmony search, ray optimizer, and particle swarm optimization (PSO) algorithm for weight minimization of trusses under multiple natural frequency constraints. Kaveh and Ilchi Ghazaan [15] employed particle swarm optimization with an aging leader and challengers (ALC–PSO) and HALC–PSO that transplants harmony search-based mechanism to ALC–PSO as a variable constraint-handling approach to optimize truss structures with frequency constraints. Hosseinzadeh et al. [16] used hybrid electromagnetism-like mechanism algorithm and migration strategy (EM–MS) for layout and size optimization of truss structures with multiple frequency constraints.

This chapter proposes the application of the VPS for optimum design of truss structures with frequency constraints. In this method, the solution candidates are considered as particles that gradually approach to their equilibrium positions. Equilibrium positions are achieved from current population and historically best position in order to have a proper balance between exploration and exploitation [17]. In order to evaluate the performance of the VPS, five truss structures are optimized for minimum weight that the design variables are considered to be the cross-sectional areas of the members and/or the coordinates of some nodes. The truss examples have 10, 37, 72, 120, and 600 members. The numerical results indicate that the proposed algorithm is quite competitive with other state-of-the-art metaheuristic methods.

The remainder of this chapter is organized as follows: In Sect. 16.2, the mathematical formulation of the structural optimization with frequency constraints is stated. The optimization algorithm is proposed after a brief overview of the free vibration of single degree of freedom systems with viscous damping in Sect. 16.3. Five structural design examples are studied in Sect. 16.4 and some concluding remarks are finally provided in Sect. 16.5.

16.2 Statement of the Optimization Problem

In this chapter, the objective is to minimize the weight of the structure while satisfying some constraints on natural frequencies. Each variable should be chosen within a permissible range. The mathematical formulation of these problems can be expressed as follows:

$$\begin{aligned}
&\text{Find} && \{X\} = \left[x_1, x_2, .., x_{ng}\right] \\
&\text{to minimize} && W(\{X\}) = \sum_{i=1}^{nm} \rho_i A_i L_i \\
&\text{subjected to :} && \begin{cases} \omega_j \le \omega_j^* \\ \omega_k \ge \omega_k^* \\ x_{\mathrm{i\ min}} \le x_{\mathrm{i}} \le x_{i\mathrm{max}} \end{cases}
\end{aligned} \tag{16.1}$$

where $\{X\}$ is the vector containing the design variables; ng is the number of design variables; $W(\{X\})$ presents the weight of the structure; nm is the number of elements of the structure; ρ_i, A_i, and L_i denote the material density, the cross-sectional area, and the length of the ith member, respectively; ω_j is the jth natural frequency of the structure and ω_j^* is its upper bound; ω_k is the kth natural frequency of the structure and ω_k^* is its lower bound; $x_{i\ \min}$ and $x_{i\ \max}$ are the lower and upper bounds of the design variable x_i, respectively.

To handle the constraints, the well-known penalty approach is employed. Thus, the objective function is redefined as follows:

$$f_{\text{cost}}(\{X\}) = (1 + \varepsilon_1 . \upsilon)^{\varepsilon_2} \times W(\{X\}), \qquad \upsilon = \sum_{j=1}^{nc} \max\left[0, g_j(\{X\})\right] \tag{16.2}$$

where υ denotes the sum of the violations of the design constraints and nc is the number of the constraints. Here, ε_1 is set to unity and ε_2 is calculated by

$$\varepsilon_2 = 1.5 + 1.5 \times \frac{iter}{iter_{max}} \tag{16.3}$$

Thus, in the first steps of the search process, ε_2 is set to 1.5 and ultimately increased to 3. Such a scheme penalizes the infeasible solutions more severely as the optimization process proceeds. As a result, in the early stages, the agents are free to explore the search space, but at the end they tend to choose solutions with no violation.

16.3 The Vibrating Particles System Algorithm

This section describes the VPS algorithm. First, a brief overview of the free vibration of single degree of freedom systems with viscous damping is provided, and then the proposed method is presented.

16.3.1 The Physical Background of the VPS Algorithm

There are two general types of vibrations, namely, free vibration and forced vibration. In free vibration, the motion is only maintained by the restoring forces, and in the forced vibration, a time-dependent force is applied to the system. The effects of friction in a vibrating system can be neglected resulting in an undamped vibration. However, all vibrations are actually damped to some degree by friction forces. These forces can be caused by dry friction, or Coulomb friction, between rigid bodies, by fluid friction when a rigid body moves in a fluid, or by internal friction between the molecules of a seemingly elastic body. In this section, the free vibration of single degree of freedom systems with viscous damping is studied. The viscous damping is caused by fluid friction at low and moderate speeds. Viscous damping is characterized by the fact that the friction force is directly proportional and opposite to the velocity of the moving body [18].

Figure 16.1 shows the vibrating motion of a body or system of mass m having viscous damping. A spring of constant k and a dashpot are connected to the block. The effect of damping is provided by the dashpot, and the magnitude of the friction force exerted on the plunger by the surrounding fluid is equal to $c\dot{x}$ (c is the coefficient of viscous damping, and its value depends on the physical properties of the fluid and the construction of the dashpot). When the block is displaced a distance x from its position of stable equilibrium, the equation of motion can be expressed as

$$m\ddot{x} + c\dot{x} + kx = 0 \tag{16.4}$$

Before presenting the solutions for this differential equation, we define the critical damping coefficient c_c as

$$c_c = 2m\omega_n \tag{16.5}$$

$$\omega_n = \sqrt{\frac{k}{m}} \tag{16.6}$$

where ω_n is the natural circular frequency of the vibration.

Depending on the value of the coefficient of viscous damping, three different cases of damping can be distinguished: (1) over-damped system ($c > c_c$), (2) critically damped system ($c = c_c$), and (3) under-damped system ($c < c_c$). The solutions

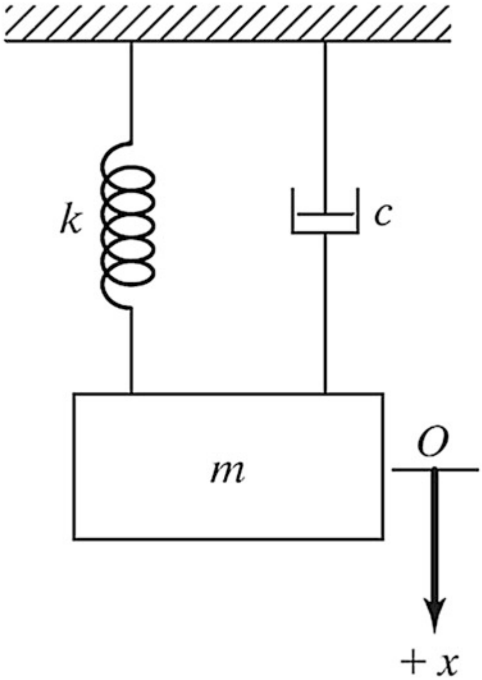


Fig. 16.1 Free vibration of a system with damping

of over-damped and critically damped systems correspond to a nonvibratory motion. Therefore, the system only oscillates and returns to its equilibrium position when $c < c_c$.

The solution of Eq. (16.4) for under-damped system is as follows:

$$x(t) = \rho e^{-\xi \omega_n t} \sin(\omega_D t + \varphi) \tag{16.7}$$

$$\omega_D = \omega_n \sqrt{1 - \xi^2} \tag{16.8}$$

$$\xi = \frac{c}{2m\omega_n} \tag{16.9}$$

where ρ and φ are constants generally determined from the initial conditions of the problem. ω_D and ξ are damped natural frequency and damping ratio, respectively. Equation (16.7) is shown in Fig. 16.2 and the effect of damping ratio on vibratory motion is illustrated in Fig. 16.3.

16.3.2 The VPS Algorithm

The VPS is a population-based algorithm which simulates a free vibration of single degree of freedom systems with viscous damping [17]. Similar to other multi-agent

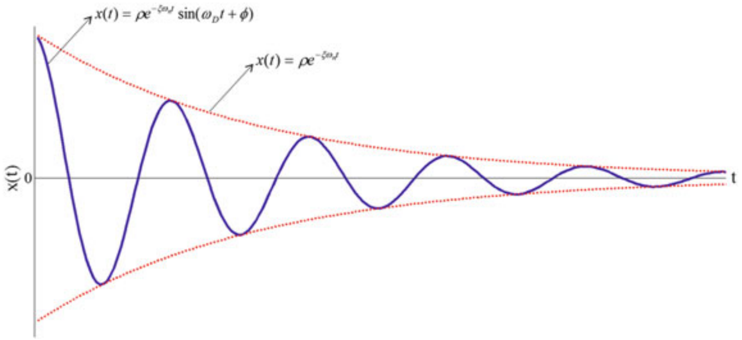


Fig. 16.2 Vibrating motion of under-damped system

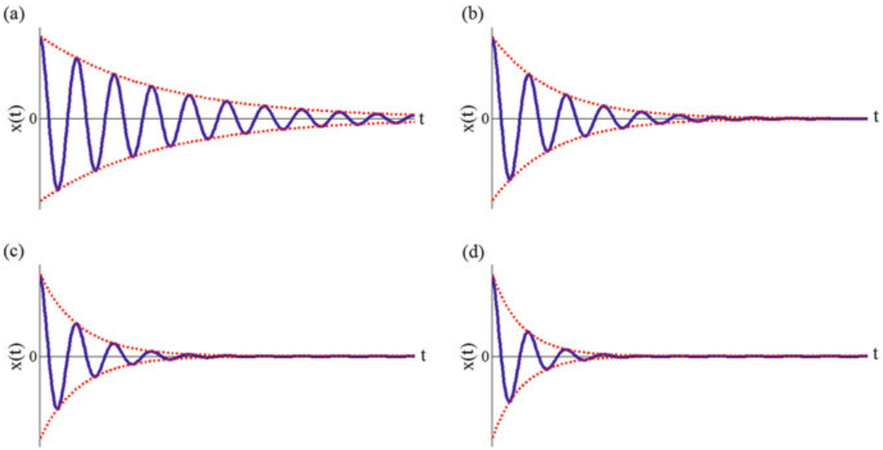


Fig. 16.3 Free vibration of systems with four levels of damping: (**a**) $\xi = 5\%$, (**b**) $\xi = 10\%$, (**c**) $\xi = 15\%$, and (**d**) $\xi = 20\%$

methods, VPS has a number of individuals (or particles) consisting of the variables of the problem. The solution candidates gradually approach to their equilibrium positions that are achieved from current population and historically best position in order to have a proper balance between diversification and intensification. In VPS, the initial locations of particles are created randomly in an *n*-dimensional search space.

$$x_i{}^j = x_{\min} + rand.(x_{\max} - x_{\min}) \tag{16.10}$$

where x_i^j is the *j*th variable of the particle *i*, $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable variable bound vectors, and *rand* is a random number uniformly distributed in the range of [0, 1].

For each particle, three equilibrium positions with different weights are defined, and during each generation, the particle position is updated by learning from them: (1) the historically best position of the entire population (*HB*), (2) a good particle (*GP*), and (3) a bad particle (*BP*). In order to select the *GP* and *BP* for each candidate solution, the current population is sorted according to their objective function values in an increasing order, and then *GP* and *BP* are chosen randomly from the first and second half, respectively.

A descending function based on the number of iterations is proposed in VPS to model the effect of damping level in the vibration that is depicted in Fig. 16.3.

$$D = \left(\frac{iter}{iter_{max}}\right)^{\alpha} \tag{16.11}$$

where *iter* is the current iteration number and $iter_{max}$ is the total number of iterations for optimization process. α is a constant.

According to the above concepts, the update rules in the VPS are given by

$$\begin{aligned} x_i^j = {} & w_1.\left[D.\, A.\, rand1 + HB^j\right] + w_2.\left[D.\, A.\, rand2 + GP^j\right] \\ & + w_3.\left[D.\, A.\, rand3 + BP^j\right] \end{aligned} \tag{16.12}$$

$$A = \left[w_1.\left(HB^j - x_i^j\right)\right] + \left[w_2.\left(GP^j - x_i^j\right)\right] + \left[w_3.\left(BP^j - x_i^j\right)\right] \tag{16.13}$$

$$w_1 + w_2 + w_3 = 1 \tag{16.14}$$

where x_i^j is the *j*th variable of the particle *i*; w_1, w_2, and w_3 are three parameters to measure the relative importance of *HB*, *GP,* and *BP*, respectively; and *rand*1, *rand*2, and *rand*3 are random numbers uniformly distributed in the range of [0,1]. The effects of *A* and *D* parameters in Eq. (16.12) are similar to that of ρ and $e^{-\xi\omega_n t}$ in Eq. (16.7). Also, the value of $\sin\left(\omega_D t + \varphi\right)$ is considered unity in Eq. (16.12) ($x(t) = \rho e^{-\xi\omega_n t}$ are shown in Fig. 16.2 by red lines).

In order to have a fast convergence in the VPS, the effect of *BP* is sometimes ignored in updating the position formula. Therefore, for each particle, a parameter like p within (0, 1) is defined, and it is compared with *rand* (a random number uniformly distributed in the range of [0,1]), and if $p < rand$, then $w_3 = 0$ and $w_2 = 1 - w_1$.

There is a possibility of boundary violation when a particle moves to its new position. In the proposed algorithm, for handling boundary constraints, a harmony search-based approach is used [19]. In this technique, there is a possibility like harmony memory considering rate (*HMCR*) that specifies whether the violating component must be changed with the corresponding component of the historically best position of a random particle or it should be determined randomly in the search space. Moreover, if the component of a historically best position is selected, there is a possibility like pitch adjusting rate (*PAR*) that specifies whether this value should be changed with a neighboring value or not.

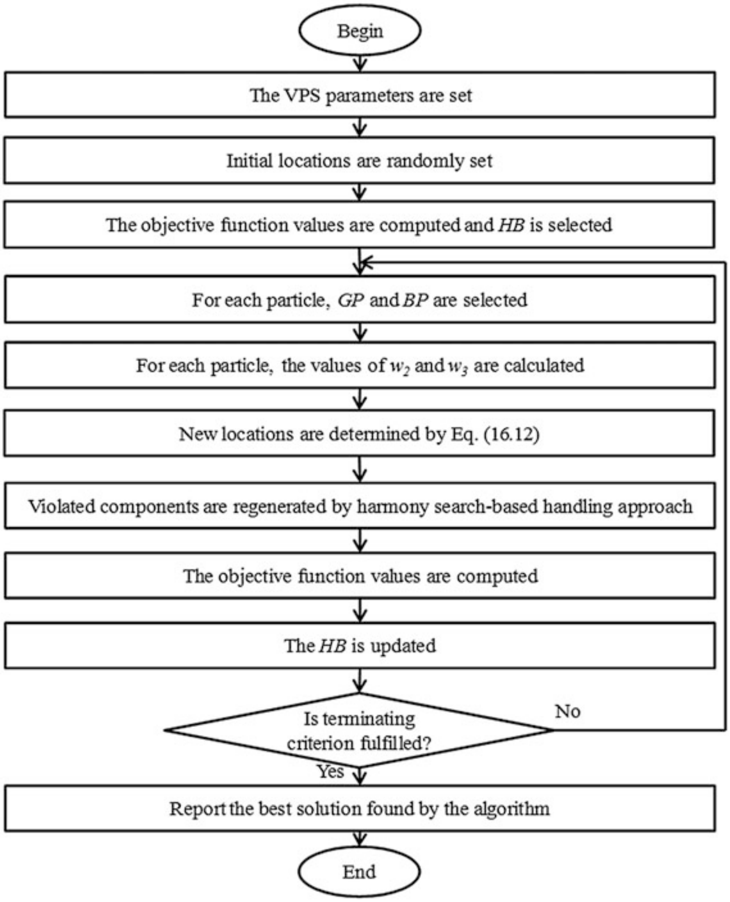


Fig. 16.4 Flowchart of the VPS algorithm

In this chapter, after the predefined maximum evaluation number, the optimization process is terminated. However, any terminating condition can be used. Flowchart of the VPS is illustrated in Fig. 16.4.

16.4 Test Problems and Optimization Results

This section discusses the computational examples used to investigate the performance of the proposed algorithm. The values of the population size, the total number of iteration, α, p, w_1, and w_2 are set to 20, 1500, 0.05, 70 %, 0.3, and 0.3 for all examples, respectively. Sensitivity analyses of the VPS on these parameters are investigated in [17]. Twenty independent optimization runs are carried out for the first four considered examples, and the last example has been solved five times independently. The algorithm is coded in MATLAB, and the structures are analyzed using the direct stiffness method by our own codes.

16.4.1 A 10-Bar Plane Truss

The 10-bar plane truss is a well-known benchmark problem, and Fig. 16.5 shows the topology and nodal and element numbering of this truss. The cross-sectional area of each of the members is considered to be an independent variable. The material density is 2767.99 kg/m^3 and the modulus of elasticity is 68.95 GPa for all elements. At each free node (1–4), a nonstructural mass of 453.6 kg is attached. The range of cross-sectional area of all members is from 0.645 to 50 cm^2. The first three natural frequencies of the structure must satisfy the following limitations ($f_1 \geq 7$ Hz, $f_2 \geq 15$ Hz, and $f_3 \geq 20$ Hz).

Table 16.1 provides a comparison between some optimal design reported in the literature and the present work. It can be seen that the lightest design (i.e., 530.77 kg) and the best standard deviation on average (i.e., 2.55 kg) are obtained by the VPS. The firefly algorithm (FA) [12] achieved the best average optimized weight (i.e., 535.07 kg), and after that the VPS obtained 535.64 kg. Table 16.2 reports the natural frequencies of the optimized structures, and it is clear that none of the frequency constraints are violated. The VPS converges to the optimum solution after 4620 analyses. The methods utilized by Lingyun et al. [9], and Gomes [10] and Miguel and Fadel Miguel [12] give the best result in 8000, 2000, and 50,000 analyses. However, the VPS achieve the best design of PSO [10] after 940 analyses.

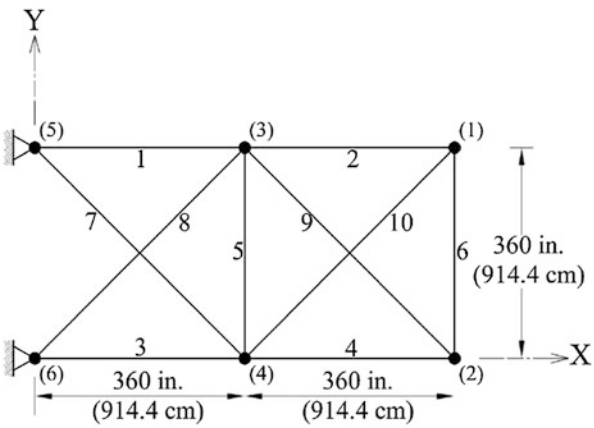


Fig. 16.5 Schematic of the 10-bar plane truss

Table 16.1 Comparison of optimized designs found for the 10-bar plane truss problem

Design variable	Areas (cm^2)				
	Wang et al. [7]	Lingyun et al. [9]	Gomes [10]	Miguel and Fadel Miguel [12]	Present work [1]
1	32.456	42.234	37.712	36.198	35.1471
2	16.577	18.555	9.959	14.030	14.6668
3	32.456	38.851	40.265	34.754	35.6889
4	16.577	11.222	16.788	14.900	15.0929
5	2.115	4.783	11.576	0.654	0.6450
6	4.467	4.451	3.955	4.672	4.6221
7	22.810	21.049	25.308	23.467	23.5552
8	22.810	20.949	21.613	25.508	24.4680
9	17.490	10.257	11.576	12.707	12.7198
10	17.490	14.342	11.186	12.351	12.6845
Weight (kg)	553.8	542.75	537.98	531.28	530.77
Average optimized weight (kg)	N/A	552.447	540.89	535.07	535.64
Standard deviation on average weight (kg)	N/A	4.864	6.84	3.64	2.55

Table 16.2 Natural frequencies (Hz) evaluated at the optimum designs of the 10-bar plane truss problem

Frequency number	Natural frequencies (Hz)				
	Wang et al. [7]	Lingyun et al. [9]	Gomes [10]	Miguel and Fadel Miguel [12]	Present work [1]
1	7.011	7.008	7.000	7.0002	7.0000
2	17.302	18.148	17.786	16.1640	16.1599
3	20.001	20.000	20.000	20.0029	20.0000
4	20.100	20.508	20.063	20.0221	20.0001
5	30.869	27.797	27.776	28.5428	28.6008
6	32.666	31.281	30.939	28.9220	29.0628
7	48.282	48.304	47.297	48.3538	48.4904
8	52.306	53.306	52.286	50.8004	51.0476

16.4.2 A Simply Supported 37-Bar Plane Truss

The 37-bar plane truss with initial configuration is shown in Fig. 16.6. Nodal coordinates in the upper chord and member areas are regarded as design variables. In the optimization process, nodes of the upper chord can be shifted vertically. In addition, nodal coordinates and member areas are linked to maintain the structural symmetry. Thus, only five layout variables and fourteen sizing variables will be considered for the optimization. All members on the lower chord (numbers 28–37) are modeled as bar elements with constant rectangular cross-sectional areas of 4×10^{-3} m^2, and the others are modeled as bar elements with initial cross-sectional

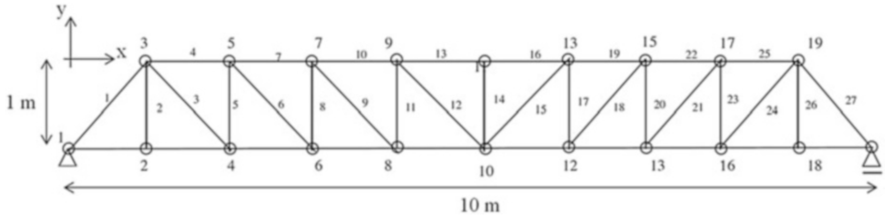


Fig. 16.6 Schematic of the simply supported 37-bar plane truss

areas of 1×10^{-4} m^2. The material density is 7800 kg/m^3 and the modulus of elasticity is 210 GPa for all elements. Nonstructural mass of 10 kg is attached to each of the free nodes on the lower chord which remain fixed during the design process. The first three natural frequencies of the structure must satisfy the following limitations: $f_1 \geq 20$ Hz, $f_2 \geq 40$ Hz, and $f_3 \geq 60$ Hz.

This truss structure was previously optimized by Wang et al. [7] utilizing an evolutionary node shift method, Lingyun et al. [9] using niche hybrid genetic algorithm, Gomes [10] employing particle swarm optimization algorithm, Miguel and Fadel Miguel [12] using firefly algorithm, and Kaveh and Ilchi Ghazaan [15] utilizing particle swarm optimization with an aging leader and challengers and harmony search-based side constraint-handling approach. Table 16.3 presents a comparison between the results of the optimal designs reported in the literature and the present work. The best weight, average optimized weight, and standard deviation on average weight obtained by VPS and HALC–PSO [15] are approximately identical although their designs are different. Table 16.4 shows the optimized structural frequencies (Hz) for various methods. None of the frequency constraints are violated. The proposed method requires 7940 structural analyses to find the optimum solution, while NHGA [9], PSO [10], FA [12], and HALC–PSO [15] require 8000, 12,500, 50,000, and 10,000 structural analyses, respectively.

16.4.3 A 72-Bar Space Truss

The 72-bar space truss is shown in Fig. 16.7 as the third design example. The elements are divided into 16 groups, because of symmetry. The material density is 2767.99 kg/m^3 and the elastic modulus is 68.95 GPa for all members. Four nonstructural masses of 2268 kg are attached to the nodes 1 through 4. The allowable minimum cross-sectional area of all elements is set to 0.645 cm^2. This example has two frequency constraints. The first frequency is required to be $f_1 = 4$ Hz and the third frequency is required to be $f_3 \geq 6$ Hz.

Optimal structures found by Konzelman [5], Gomes [10], Kaveh and Zolghadr [11], Miguel and Fadel Miguel [12], and Kaveh and Ilchi Ghazaan [15] and the proposed method are summarized in Table 16.5. The CSS–BBBC (hybridization of charged system search and Big Bang with trap recognition capability) [11] obtained

Table 16.3 Comparison of optimized designs found for the 37-bar truss problem

Design variable	Y coordinates (m) and areas (cm^2)					
	Wang et al. [7]	Lingyun et al. [9]	Gomes [10]	Miguel and Fadel Miguel [12]	Kaveh and Ilchi Ghazaan [15]	Present work [1]
Y3, Y19 (m)	1.2086	1.1998	0.9637	0.9392	0.9750	0.9042
Y5, Y17 (m)	1.5788	1.6553	1.3978	1.3270	1.3577	1.2850
Y7, Y15 (m)	1.6719	1.9652	1.5929	1.5063	1.5520	1.5017
Y9, Y13 (m)	1.7703	2.0737	1.8812	1.6086	1.6920	1.6509
Y11 (m)	1.8502	2.3050	2.0856	1.6679	1.7688	1.7277
A1, A27 (cm^2)	3.2508	2.8932	2.6797	2.9838	2.9652	3.1306
A2, A26 (cm^2)	1.2364	1.1201	1.1568	1.1098	1.0114	1.0023
A3, A24 (cm^2)	1.0000	1.0000	2.3476	1.0091	1.0090	1.0001
A4, A25 (cm^2)	2.5386	1.8655	1.7182	2.5955	2.4601	2.5883
A5, A23 (cm^2)	1.3714	1.5962	1.2751	1.2610	1.2300	1.1119
A6, A21 (cm^2)	1.3681	1.2642	1.4819	1.1975	1.2064	1.2599
A7, A22 (cm^2)	2.4290	1.8254	4.6850	2.4264	2.4245	2.6743
A8, A20 (cm^2)	1.6522	2.0009	1.1246	1.3588	1.4618	1.3961
A9, A18 (cm^2)	1.8257	1.9526	2.1214	1.4771	1.4328	1.5036
A10, A19 (cm^2)	2.3022	1.9705	3.8600	2.5648	2.5000	2.4441
A11, A17 (cm^2)	1.3103	1.8294	2.9817	1.1295	1.2319	1.2977
A12, A15 (cm^2)	1.4067	1.2358	1.2021	1.3199	1.3669	1.3619
A13, A16 (cm^2)	2.1896	1.4049	1.2563	2.9217	2.2801	2.3500
A14 (cm^2)	1.0000	1.0000	3.3276	1.0004	1.0011	1.0000
Weight (kg)	366.5	368.84	377.20	360.05	359.93	359.94
Average optimized weight (kg)	N/A	378.8259	381.2	360.37	360.23	360.23
Standard deviation on average weight (kg)	N/A	9.0325	4.26	0.26	0.24	0.22

Table 16.4 Natural frequencies (Hz) evaluated at the optimum designs of the 37-bar truss problem

Frequency number	Natural frequencies (Hz)					
	Wang et al. [7]	Lingyun et al. [9]	Gomes [10]	Miguel and Fadel Miguel [12]	Kaveh and Ilchi Ghazaan [15]	Present work [1]
1	20.0850	20.0013	20.0001	20.0024	20.0216	20.0002
2	42.0743	40.0305	40.0003	40.0019	40.0098	40.0005
3	62.9383	60.0000	60.0000	60.0043	60.0017	60.0000
4	74.4539	73.0444	73.0440	77.2153	76.7857	77.2124
5	90.0576	89.8244	89.8240	96.9900	96.3543	97.3173

the lightest design; however, the best designs of all methods are approximately identical. The average optimized weight and the standard deviation on average weight of the VPS are less than those of all other methods. Frequency constraints

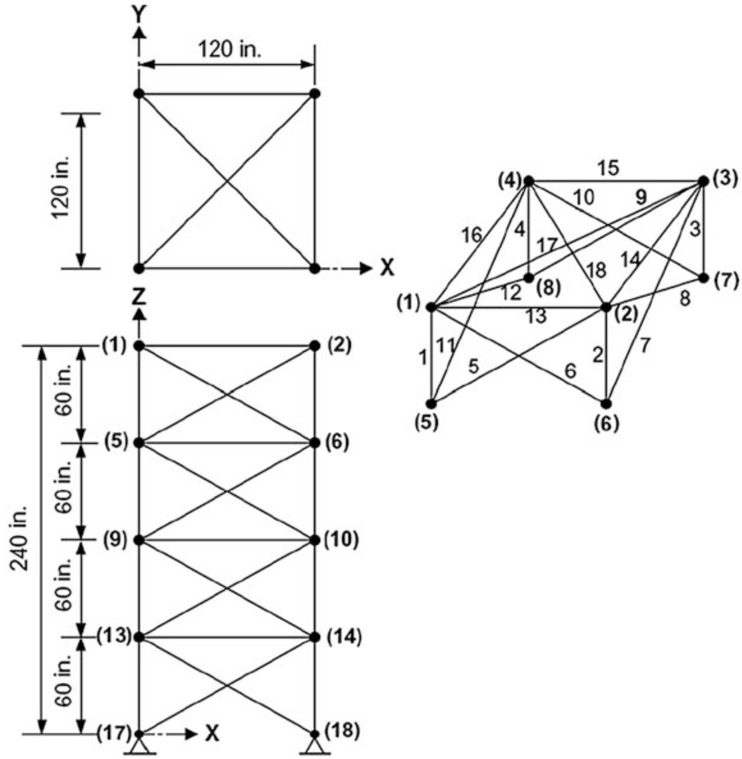


Fig. 16.7 Schematic of the spatial 72-bar truss

are satisfied by all methods (see Table 16.6). Figure 16.8 compares the best and average runs of convergence histories for the proposed method. The VPS requires 4720 structural analyses to find the optimum solution, while PSO [10], FA [12], and HALC–PSO [15] require 42,840, 100,000, and 8000 structural analyses, respectively.

16.4.4 A 120-Bar Dome Truss

Figure 16.9 shows the 120-bar dome truss. The members are categorized into seven groups because of symmetry. The material density is 7971.810 kg/m^3, and the modulus of elasticity is 210 GPa for all elements. Nonstructural masses are attached to all free nodes as follows: 3000 kg at node one, 500 kg at nodes 2–13, and 100 kg at the remaining nodes. Element cross-sectional areas can vary between 1 cm^2 and 129.3 cm^2. The frequency constraints are as follows: $f_1 \geq 9$ Hz and $f_2 \geq 11$ Hz.

Table 16.5 Comparison of optimized designs obtained for the 72-bar truss problem

		Areas (cm^2)					
Design variable	Members in the group	Konzelman [5]	Gomes [10]	Kaveh and Zolghadr [11]	Miguel and Fadel Miguel [12]	Kaveh and Ilchi Ghazaan [15]	Present work [1]
1	1–4	3.499	2.987	2.854	3.3411	3.3437	3.5017
2	5–12	7.932	7.849	8.301	7.7587	7.8688	7.9340
3	13–16	0.645	0.645	0.645	0.6450	0.6450	0.6450
4	17–18	0.645	0.645	0.645	0.6450	0.6450	0.6450
5	19–22	8.056	8.765	8.202	9.0202	8.1626	8.0215
6	23–30	8.011	8.153	7.043	8.2567	7.9502	7.9826
7	31–34	0.645	0.645	0.645	0.6450	0.6452	0.6450
8	35–36	0.645	0.645	0.645	0.6450	0.6450	0.6450
9	37–40	12.812	13.450	16.328	12.0450	12.2668	12.8175
10	41–48	8.061	8.073	8.299	8.0401	8.1845	8.1129
11	49–52	0.645	0.645	0.645	0.6450	0.6451	0.6450
12	53–54	0.645	0.645	0.645	0.6450	0.6451	0.6450
13	55–58	17.279	16.684	15.048	17.3800	17.9632	17.3362
14	59–66	8.088	8.159	8.268	8.0561	8.1292	8.1010
15	67–70	0.645	0.645	0.645	0.6450	0.6450	0.6450
16	71–72	0.645	0.645	0.645	0.6450	0.6450	0.6450
Weight (kg)		327.605	328.823	327.507	327.691	327.77	327.649
Average optimized weight (kg)		N/A	332.24	N/A	329.89	327.99	327.670
Standard deviation on average weight (kg)		N/A	4.23	N/A	2.59	0.19	0.018

Table 16.6 Natural frequencies (Hz) evaluated at the optimum designs of the 72-bar truss problem

Frequency number	Natural frequencies (Hz)					
	Konzelman [5]	Gomes [10]	Kaveh and Zolghadr [11]	Miguel and Fadel Miguel [12]	Kaveh and Ilchi Ghazaan [15]	Present work [1]
1	4.000	4.000	4.000	4.0000	4.000	4.0000
2	4.000	4.000	4.000	4.0000	4.000	4.0002
3	6.000	6.000	6.004	6.0000	6.000	6.0000
4	6.247	6.219	6.2491	6.2468	6.230	6.2428
5	9.074	8.976	8.9726	9.0380	9.041	9.0698

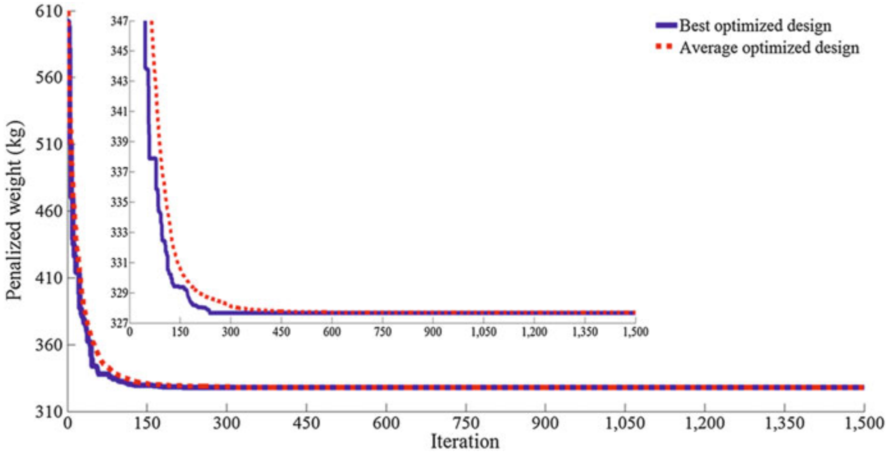


Fig. 16.8 Convergence curves obtained for the 72-bar truss

The comparison of the results of the VPS algorithm with the outcomes of other algorithms is shown in Table 16.7. The present algorithm yields the least weight. The best weight of the VPS algorithm is 8888.74 kg, while it is 9046.34 kg for CSS–BBBC [11] and 8889.96 kg for the HALC–PSO [15]. Moreover, it can be seen that the lightest average optimized weight and the standard deviation on average weight are found by the proposed method. Table 16.8 reports the natural frequencies of the optimized structures, and it is clear that none of the frequency constraints are violated. Figure 16.10 compares the convergence curves of the best and the average results obtained by the proposed method. The HALC–PSO [15] and VPS algorithms get the optimal solution after 17,000 and 6860 analyses, respectively.

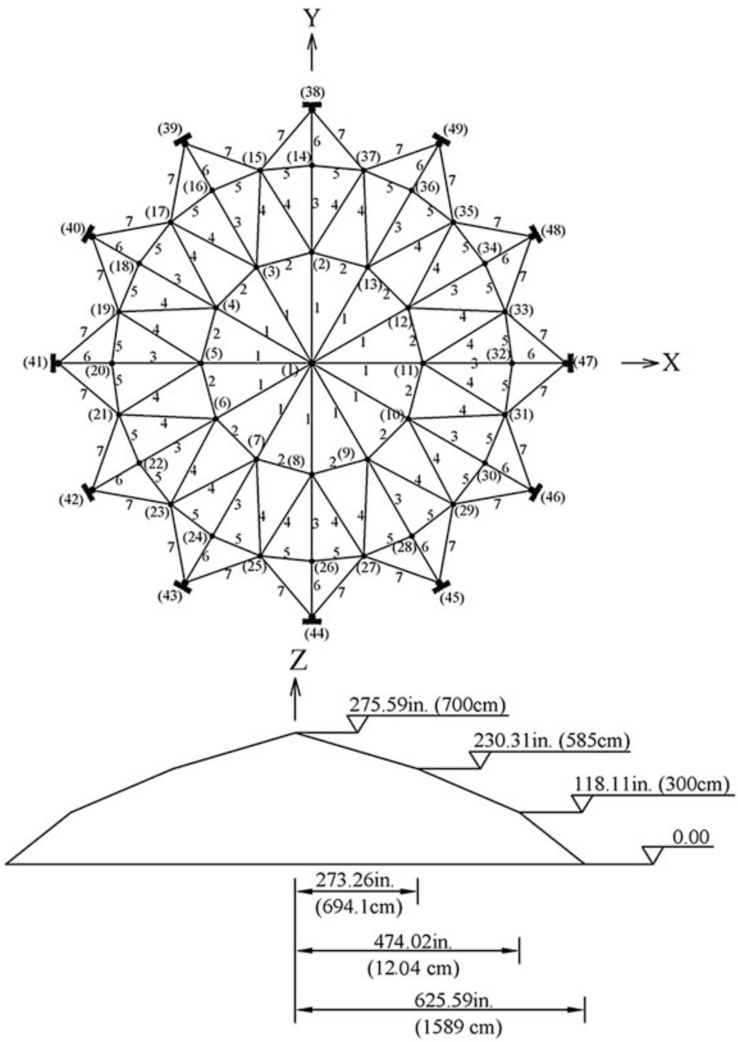


Fig. 16.9 Schematic of the spatial 120-bar dome truss

16.4.5 A 600-Bar Single-Layer Dome Truss

The 600-bar single-layer dome structure shown in Fig. 16.11 is considered as the last example. The entire structure is composed of 216 nodes and 600 elements. A more detailed substructure is depicted in Fig. 16.12 to show the nodal numbering and coordinates. Each of the elements of this substructure is considered as a design variable. Thus, this is a size optimization problem with 25 variables. The material density is 7850 kg/m^3 and the elastic modulus is 200 GPa for all members. A nonstructural mass of 100 kg is attached to all free nodes. The minimum

Table 16.7 Comparison of optimized designs obtained for the 120-bar dome problem

Design variable	Areas (cm^2)		
	Kaveh and Zolghadr [11]	Kaveh and Ilchi Ghazaan [15]	Present work [1]
1	17.478	19.8905	19.6836
2	49.076	40.4045	40.9581
3	12.365	11.2057	11.3325
4	21.979	21.3768	21.5387
5	11.190	9.8669	9.8867
6	12.590	12.7200	12.7116
7	13.585	15.2236	14.9330
Weight (kg)	9046.34	8889.96	8888.74
Average optimized weight (kg)	N/A	8900.39	8896.04
Standard deviation on average weight (kg)	N/A	6.38	6.65

Table 16.8 Natural frequencies (Hz) evaluated at the optimum designs of the 120-bar dome problem

Frequency number	Natural frequencies (Hz)		
	Kaveh and Zolghadr [11]	Kaveh and Ilchi Ghazaan [15]	Present work [1]
1	9.000	9.000	9.0000
2	11.007	11.000	11.0000
3	11.018	11.000	11.0000
4	11.026	11.010	11.0096
5	11.048	11.050	11.0491

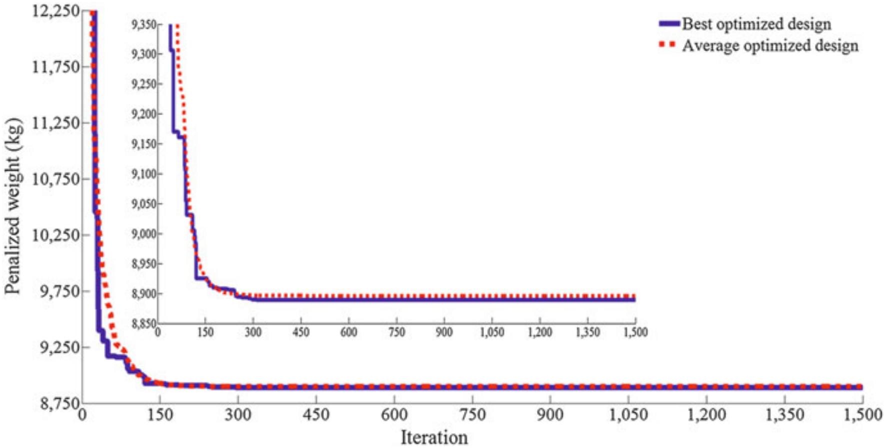


Fig. 16.10 Convergence curves obtained for the 120-bar dome truss

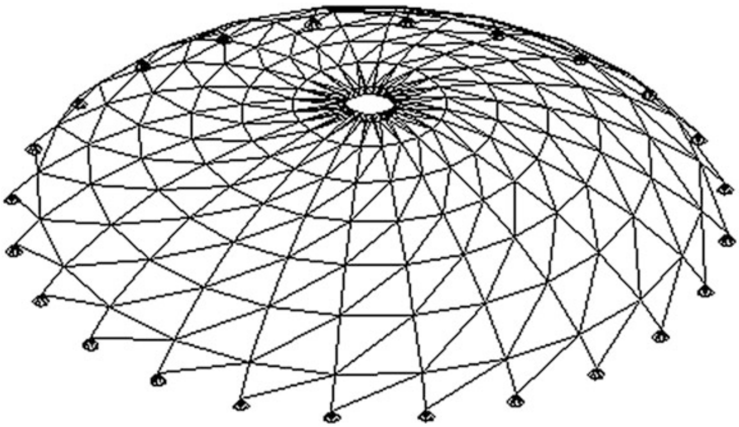


Fig. 16.11 Schematic of the 600-bar single-layer dome truss

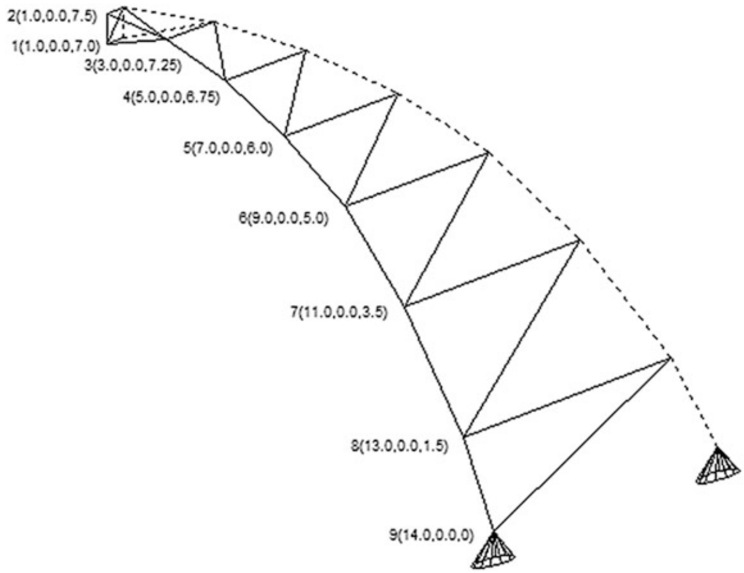


Fig. 16.12 Details of a substructure of the 600-bar single-layer dome truss

cross-sectional area of all members is 1×10^{-4}, and the maximum cross-sectional area is taken as 100×10^{-4} m^2. The frequency constraints are as follows: $\omega_1 \geq 5$ Hz and $\omega_3 \geq 7$ Hz.

The optimized designs found by the ECBO [20] and VPS are compared in Table 16.9. It can be seen that the lightest design (i.e., 6133.02 kg) is obtained by the VPS, and this method performs better than ECBO in terms of average optimized weight and standard deviation on average weight. Table 16.10 reports the natural

Table 16.9 Comparison of optimized designs obtained for the 600-bar single-layer dome truss problem

Design variable (nodes)	Areas (cm^2)	
	Kaveh and Ilchi Ghazaan [20]	Present work [1]
1 (1–2)	1.4305	1.3030
2 (1–3)	1.3941	1.3998
3 (1–10)	5.5293	5.1072
4 (1–11)	1.0469	1.3882
5 (2–3)	16.9642	16.9217
6 (2–11)	35.1892	38.1432
7 (3–4)	12.2171	11.8319
8 (3–11)	16.7152	16.6149
9 (3–12)	12.5999	11.3403
10 (4–5)	9.5118	9.3865
11 (4–12)	8.9977	8.7692
12 (4–13)	9.4397	9.6682
13 (5–6)	6.8864	6.9826
14 (5–13)	4.2057	5.4445
15 (5–14)	7.2651	6.3247
16 (6–7)	6.1693	5.1349
17 (6–14)	3.9768	3.3991
18 (6–15)	8.3127	7.7911
19 (7–8)	4.1451	4.4147
20 (7–15)	2.4042	2.2755
21 (7–16)	4.3038	4.9974
22 (8–9)	3.2539	4.0145
23 (8–16)	1.8273	1.8388
24 (8–17)	4.8805	4.7965
25 (9–17)	1.5276	1.5551
Weight (kg)	6171.51	6133.02
Average optimized weight (kg)	6191.50	6142.03
Standard deviation on average weight (kg)	39.08	12.54

Table 16.10 Natural frequencies (Hz) evaluated at the optimum designs of the 600-bar single-layer dome truss problem

Frequency number	Natural frequencies (Hz)	
	Kaveh and Ilchi Ghazaan [20]	Present work [1]
1	5.002	5.0000
2	5.003	5.0003
3	7.001	7.0000
4	7.001	7.0001
5	7.002	7.0002

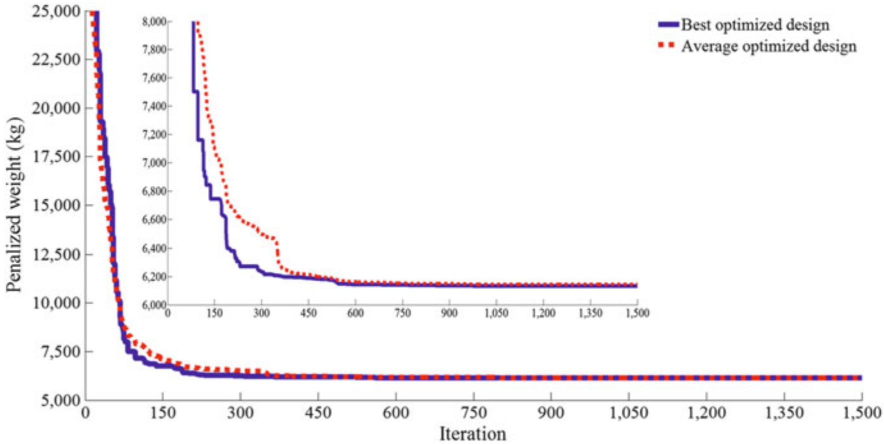


Fig. 16.13 Convergence curves obtained for the 600-bar single-layer dome truss

frequencies of the optimized structures, and it is clear that none of the frequency constraints are violated. The convergence rates of the best and average results found by the proposed method are provided in Fig. 16.13. The ECBO and VPS algorithms get the optimal solution after 19,020 and 19,740 analyses, respectively.

16.5 Concluding Remarks

Structural optimization with multiple natural frequency constraints is a challenging class of optimization problems characterized by highly nonlinear and non-convex search spaces with numerous local optima. This chapter presents VPS for finding the optimum design of this kind of problems. The VPS has a simple theoretical structure, and self-adaptation, cooperation, and competition concepts are considered in its updating formula. The solution candidates gradually approach to *HB*, and any particle has the chance to have an influence on the new position of the other one; therefore, the self-adaptation and cooperation between the particles are provided. Moreover, since the influence of *GP* is more than that of *BP* in position updating, the competition is supplied. Five planar and spatial trusses are studied in this work to verify the proposed method. The numerical results of the investigated design examples indicate the advantages of the proposed method in terms of speed of convergence, stability, and optimality of the final solutions.

References

1. Kaveh A, Ilchi Ghazaan M (2016) Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. Acta Mech. doi:10.1007 /s00707-016-1725-z. First Online: 20 Sept 2016

2. Tong WH, Jiang JS, Liu GR (2000) Solution existence of the optimization problem of truss structures with frequency constraints. Int J Solids Struct 37:4043–4060
3. Bellagamba L, Yang TY (1981) Minimum mass truss structures with constraints on fundamental natural frequency. AIAA J 19:1452–1458
4. Lin JH, Chen WY, Yu YS (1982) Structural optimization on geometrical configuration and element sizing with static and dynamic constraints. Comput Struct 15:507–515
5. Konzelman CJ (1986) Dual methods and approximation concepts for structural optimization. MSc Thesis, Department of Mechanical Engineering, University of Toronto, Canada
6. Grandhi RV, Venkayya VB (1988) Structural optimization with frequency constraints. AIAA J 26:858–866
7. Wang D, Zha W, Jiang J (2004) Truss optimization on shape and sizing with frequency constraints. AIAA J 42:622–630
8. Sedaghati R (2005) Benchmark case studies in structural design optimization using the force method. Int J Solids Struct 42:5848–5871
9. Lingyun W, Mei Z, Guangming W, Guang M (2005) Truss optimization on shape and sizing with frequency constraints based on genetic algorithm. Comput Mech 35:361–368
10. Gomes HM (2011) Truss optimization with dynamic constraints using a particle swarm algorithm. Expert Syst Appl 38:957–968
11. Kaveh A, Zolghadr A (2012) Truss optimization with natural frequency constraints using a hybridized CSS–BBBC algorithm with trap recognition capability. Comput Struct 102–103:14–27
12. Miguel LFF, Fadel Miguel LF (2012) Shape and size optimization of truss structures considering dynamic constraints through modern meta-heuristic algorithms. Expert Syst Appl 39:9458–9467
13. Zuo W, Bai J, Li B (2014) A hybrid OC–GA approach for fast and global truss optimization with frequency constraints. Appl Soft Comput 14:528–535
14. Kaveh A, Javadi SM (2014) Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm. Acta Mech 225:1595–1605
15. Kaveh A, Ilchi Ghazaan M (2015) Hybridized optimization algorithms for design of trusses with multiple natural frequency constraints. Adv Eng Softw 79:137–147
16. Hosseinzadeh Y, Taghizadieh N, Jalili S (2016) Hybridizing electromagnetism-like mechanism algorithm with migration strategy for layout and size optimization of truss structures with frequency constraints. Neural Comput Appl 27(4):953–971
17. Kaveh A, Ilchi Ghazaan M (2016) A new meta-heuristic algorithm: vibrating particles system. Scientia Iranica (Published online)
18. Beer FP, Johnston ER Jr, Mazurek DF, Cornwell PJ, Self BP (2013) Vector mechanics for engineers. McGraw-Hill Companies, New York, NY
19. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–286
20. Kaveh A, Ilchi Ghazaan M (2016) Optimal design of dome truss structures with dynamic frequency constraints. Struct Multidiscip Optim 53(3):605–621

Chapter 17
Cost and CO_2 Emission Optimization of Reinforced Concrete Frames Using Enhanced Colliding Bodies Optimization Algorithm

17.1 Introduction

This chapter investigates discrete design optimization of reinforcement concrete frames using the recently developed metaheuristic called Enhanced Colliding Bodies Optimization (ECBO) and the Non-dominated Sorting Enhanced Colliding Bodies Optimization (NSECBO) algorithm. The objective function of algorithms consists of construction material costs of reinforced concrete structural elements and carbon dioxide (CO_2) emissions through different phases of a building life cycle that meets the standards and requirements of the American Concrete Institute's Building Code. The proposed method uses predetermined section database (DB) for design variables that are taken as the area of steel and the geometry of cross sections of beams and columns. A number of benchmark test problems are optimized to verify the good performance of this methodology. The use of ECBO algorithm for designing reinforced concrete frames indicates an improvement in the computational efficiency over the designs performed by Big Bang–Big Crunch (BB–BC) algorithm. The analysis also reveals that the two objective functions are quite relevant and designs focused on mitigating CO_2 emissions could be achieved at an acceptable cost increment in practice. Pareto results of the NSECBO algorithm indicate that both objectives yield similar solutions [1].

The growing global climate change with the progress of human activity and rapid industrialization has created a need to appraise the impact of the products used in construction process and has challenged many contractors and companies to come up with more environmentally friendly ways of construction. Most of global warming has being caused by increasing concentration of greenhouse gases in the earth's atmosphere during the past ten decades [2]. The Intergovernmental Panel on Climate Change [3] reported that carbon dioxide makes up approximately 77 % of greenhouse gases in which construction industry has a remarkable contribution.

Concrete as the most popular manufactured product with sustainability benefits, including considerable compressive strength and durability, excellent thermal

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_17

mass, and long service life, contributes 5 % of annual anthropogenic global CO_2 production. Main contributor for it to happen is chemical conversion process used in the production of Portland clinker and cement production by fossil fuel combustion. With annual consumption approaching 20,000 million metric tons of concrete, the manufacturing process releases 0.9 tons of CO_2 per ton of clinker [2]. In addition to the 1.6 billion tons of cement used worldwide, the concrete industry is consuming 12.6 billion tons of raw materials each year. Thus, besides cement's role in CO_2 emission, mining, processing, and transporting of raw materials consume energy in large quantities and adversely affect theology of the planet [4]. Reducing atmospheric concentration of CO_2 caused by construction industry can be reached through innovative architecture, sustainable structural design, and reducing the cement of concrete mixture [2].

The purpose of this chapter is to present an optimal design technique in order to achieve more sustainable, environmentally friendly, and economically feasible structural design. The methods of structural optimization can be divided into two categories: exact methods and approximate methods. The exact methods are based on mathematical programming techniques such as the Lagrangian multipliers method, convex programming, linear programming, and sequential unconstrained minimization for which the required computational cost for finding an optimal solution grow polynomially with problem size, hence the applications of the exact methods are limited to simple and deterministic polynomial problem instances. To overcome these problems, metaheuristic methods are developed. These methods provide the practical possibility to improve the design process without the need for complex analysis; however, they require a great computational effort because of a large number of iterations needed for the evaluation of objective functions and structural constraints.

Some recent research studies are focused on cost optimization of reinforced concrete structures using evolutionary optimization methods. Rajeev and Krishnamoorthy [5] applied a simple genetic algorithm to perform optimal design of planar reinforced concrete frames, Camp et al. [6] used genetic algorithm for flexural design of RC frames, Lee and Ahn [7] applied genetic algorithm to optimum design of two-dimensional frames, Paya-Zaforteza et al. [8] conducted a multi-objective comparison for RC building frames using simulated annealing, Kwak and Kim [9] studied an optimum design of RC plane frames using integrated genetic algorithm complemented with direct search, Kaveh and Sabzi [10] conducted a comparative study of heuristic big bang–big crunch, heuristic particle swarm, and ant colony optimization for optimum design of RC frames, and Akin and Saka [11] used harmony search algorithm for optimum detailed design of RC plane frames.

Recently, attention to the preservation of environment and reducing CO_2 emissions has been the focus of studies in optimum design of RC structures. Paya-Zaforteza et al. [12] used simulated annealing for CO_2 optimization of reinforced concrete frames; Camp and Huq [13] applied the Big Bang–Big Crunch algorithm for CO_2 and cost optimization of RC frames. The objective of this chapter is optimal design of cost and CO_2 emissions in terms of cross-section dimensions and reinforcement details applying the American Concrete Institute's Building Code [14] of practice. The optimization is carried out using enhanced colliding bodies optimization (CBO) algorithm developed by Kaveh and Ilchi Ghazaan [15] based on the improvement of CBO performance originally developed by Kaveh and Mahdavi [16] using memory to preserve some historically best solutions.

The rest of this chapter is structured as follows: Sect. 17.2 describes the formulation of optimization problem, Sect. 17.3 contains the explanations of utilized metaheuristic algorithm, and in Sect. 17.4 the results obtained for three benchmark frames are detailed and discussed. Finally, in Sect. 17.5, the concluding remarks are presented.

17.2 Formulation of the RC Frame Optimization Problem

17.2.1 Design Variables and Section Databases

The assessment of the objective functions requires the definition of the structure in terms of the design variables including cross-sectional dimensions of elements, area and type of steel bars, and resisting capacity. Due to the discreteness of member dimensions and reinforcement sizes, large number of sections, and different patterns of reinforcements, two section databases for beams and columns are created to reduce the elaboration of the problem. The identification numbers of the sections are related with all design variables. It is worth pointing out that the capacity of members is defined by applying ultimate strength design method. Two section databases are created based on ACI building code criteria and specified assumptions, which are followed for both beams and columns sections.

17.2.1.1 Beams

For beams, the sections are considered as rectangular and singly reinforced; therefore, the compression reinforcement at support and the tension reinforcement near mid-span are checked separately. This approach leads to a conservative and simple analysis. The area of steel varies from one #3 bar to a maximum of four #11 bars. The depth to width ratio varies between 1 and 2.5.

The last distance measured from the surface of the concrete member to the surface of the embedded reinforcing steel is taken as 380 mm. The assumed ranges and increment steps for cross-sectional dimensions are different in each design example. Figure 17.1a defines the geometry of a general rectangular singly reinforced concrete beam.

To evaluate flexural response of the beam elements, their capacity is defined using the ACI code. In order to ensure ductile failure, these must be designed as

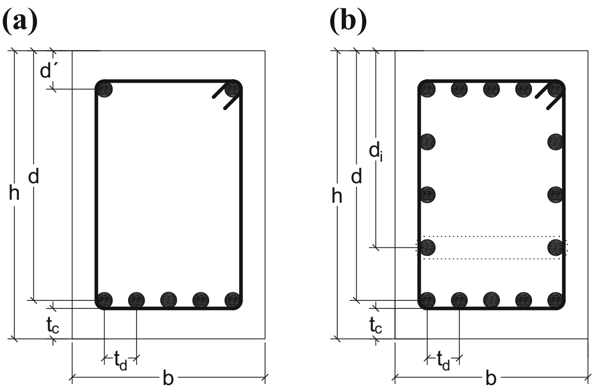


Fig. 17.1 General rectangular reinforced concrete beam and column

under reinforced beams. The nominal resisting moment capacity of a singly reinforced concrete beam section is

$$M_n = A_s f_y \left(d - \frac{a}{2}\right) \tag{17.1}$$

where A_s is the total area of tensile reinforcement, f_y is the yield strength of reinforcement, d is the distance from extreme compression fibers of the concrete to the centroid of tension reinforcement, and a refers to the depth of equivalent rectangular compression block given as

$$a = \frac{A_s f_y}{0.85 f'_c b} \tag{17.2}$$

where f'_c denotes the specified compressive strength of concrete and b is the width of section.

Taking the abovementioned rules into account, DB sections for beams containing the width, the height, the number of reinforcing bar, the steel ratio, the moments of inertia, and the ultimate bending moment capacity can be created. Finally, the sections are arranged in the order of increasing moment resisting capacities.

17.2.1.2 Columns

For columns, the sections are considered as rectangular tied and short, so the applied moment will not be magnified. The area of steel varies from four #3 bars to a maximum of twelve #11 bars. For the rebar topologies, an even number of bars with the same size are distributed along all four faces so that the column is symmetric about the axis of bending. Table 17.1 represents the prespecified

Table 17.1 Column reinforcement combinations [13]

	Reinforcement combination	
Index no.	Width side	Height side
1	2	2
2	3	2
3	2	3
4	3	3
5	4	3
6	4	4

reinforcement patterns for columns. The depth to width ratio is considered between 1 and 2.5. Figure 17.1b defines the geometry of a rectangular tied column.

Column sections are subjected to bending moment in combination with axial forces; therefore, the equilibrium of internal forces changes resulting in different behavioral modes depending on the level of accompanying eccentricity. The sustainability and serviceability of column sections can be evaluated in a variety of combinations of bending moment and axial force derived by varying the applied axial strain. To find points corresponding to a specific value of strain distribution within the cross section, a rectangular stress block in the concrete must be determined. The same method is used to specify the stress distribution in reinforcement. Plotting values of load and moment capacities corresponding to different assumed values for the neutral axis depth (resulting in different strain distributions) via an iterative calculation results in some contour charts called interaction diagrams. Figure 17.2 shows a curve plot of controlling key points connected by linear relationships for a typical column section. The nominal axial load capacity for a given strain distribution defined by ACI Code is found by

$$P_{\mathrm{n}} = C_{\mathrm{c}} + \sum\nolimits_{i=1}^{n} F_{si} \tag{17.3}$$

where n is the number of reinforcement layers and C_{c} is the compressive force of concrete given as

$$C_{\mathrm{c}} = 0.85 f'_{\mathrm{c}} ab \tag{17.4}$$

and F_{si} is the force in each layer of the reinforcement given as

$$F_{si} = f_{si} A_{si} \text{ if } a \leq d_i \tag{17.5a}$$

$$F_{si} = \left(f_{si} - 0.85 f'_{\mathrm{c}}\right) A_{si} \text{ if } a > d_i \tag{17.5b}$$

where f_{si} is the yield strength of reinforcement given as

$$f_{si} = \varepsilon_{si} E_{\mathrm{s}} \quad -f_{\mathrm{Y}} < f_{si} < f_{\mathrm{Y}} \tag{17.6}$$

where E_{s} is the elastic modulus of reinforcement and ε_{si} is the strain of the ith layer of steel given as

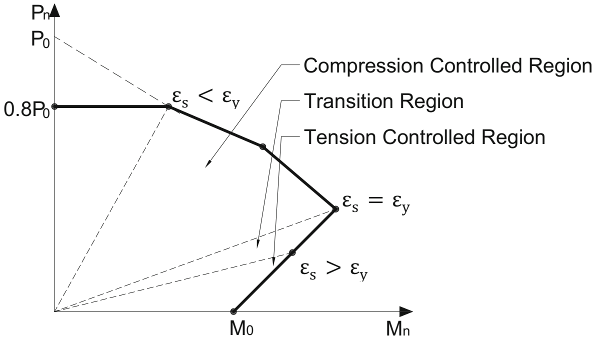


Fig. 17.2 Column load–moment interaction diagram

$$\varepsilon_{si} = 0.003\left(\frac{c - d_i}{c}\right) \tag{17.7}$$

where c is:

$$c = \left(\frac{0.003}{0.003 - \varepsilon_y}\right) d_i \tag{17.8}$$

The nominal moment capacity for the specified strain distribution defined by ACI Code is found by

$$M_n = C_c\left(\frac{h}{2} - \frac{a}{2}\right) + \sum_{i=1}^{n} F_{si}\left(\frac{h}{2} - d_i\right) \tag{17.9}$$

where a is:

$$a = \beta_1 c \tag{17.10}$$

and β is:

$$\beta_1 = 0.85 - 0.05\frac{(f_c' - 30)}{7} \geq 0.65 \text{ if } f_c' > 30\,\text{MPa} \tag{17.11a}$$

$$\beta_1 = 0.85 \text{ if } 30\,\text{MPa} < f_c' < 50\,\text{MPa} \tag{17.11b}$$

Considering the above information, DB sections for columns containing the width, the height, the number of reinforcing bars, the steel ratio, the moments of inertia, and the combination of bending moment and axial force capacities can be created. Finally, the sections are arranged in increasing order of normalized areas for the P–M interaction diagram.

17.2.2 Structural Constraints

Structural constraints are a series of restrictions in terms of the limitations and specifications provided by the ACI code. A structure should comply with these limitations in order to guarantee the feasibility of the solutions generated during iterative procedure. Making the solutions stand inside the feasible region is often a challenging effort, and it is one of the complexities for handling the constrained problems. The most common method to overcome this issue is reducing the fitness value of the merit functions by a product of eventual constraint and the objective function which converts the constrained problem into an unconstrained problem. The use of exponential penalty function allows us to enforce the constraint on the objective function. To compute the capacity constraints violation, the internal forces by the action of the vertical and horizontal loads upon the RC element are required. In this study, the first-order elastic analysis via matrix method is used to obtain the stress envelopes. By summing over the different constraints either in terms of capacity or geometry, the total penalty of each design can be expressed as

$$f_{\mathrm{p}}(x) = \left(1 + \sum\nolimits_{i=1}^{n} \max(0, C_i(x))\right)^k \tag{17.12}$$

where x is the vector of design variables that are taken as the area of steel and the geometry of cross sections of beams and columns, C_i is the normalized degree of violation of the ith constraint, n is the number of constraints, and $k > 0$ is a penalty exponent required for tuning the penalty function. Since k reflects the solution quality, imposing a large k results in severe penalty, which is reflected in rapid convergence to local optima (exploitation). Conversely, a small k reduces the severity of penalty; therefore, a comprehensive search through the search space with slow convergence will be used to explore the solutions (exploration). Depending on the case study, penalty exponent can be obtained through trial and error.

17.2.2.1 Beam Constraints

Structural capacity of reinforced concrete beams must be greater than the ultimate bending moment derived from the applied loading. The moment capacity penalty can be expressed in normalized form as below:

$$C_1 = \frac{|M_{\mathrm{u}}| - \varnothing M_{\mathrm{n}}}{\varnothing M_{\mathrm{n}}} \tag{17.13}$$

where M_{u} is the ultimate applied moment and $\varnothing$ is the strength reduction factor. For compression-controlled sections having a net tensile strain in the extreme tension steel equal to or smaller than 0.002 while the extreme fibers of compression face in concrete reach its crushing strain of 0.003, ∅ is taken as 0.65, and for tension-controlled sections having the strain values in tension reinforcement farthest from the compression face of a member >0.005 while concrete reaches its crushing strain of 0.003, ∅ is taken as 0.9. Sections between these two extremes are called transition sections, and the strength reduction factor is calculated by linear interpolation.

In order to prevent the possibility of sudden failure and improve the cracking behavior, the lower bound of reinforcement ratio is limited to

$$\rho_{\min} = \frac{\sqrt{f'_c}}{4f_y} \geq \frac{1.4}{f_y} \tag{17.14}$$

The minimum reinforcement ratio penalty is

$$C_2 = \rho_{\min} - \rho \tag{17.15}$$

To ensure the ductile behavior and the requirements for placing the reinforcing bars, the upper bound on the reinforcement ratio is limited to

$$\rho_{\max} = 0.85\beta_1 \frac{f'_c}{f_y} \frac{600}{600 + f_y} \tag{17.16}$$

The maximum reinforcement ratio penalty is

$$C_3 = \rho - \rho_{\max} \tag{17.17}$$

For controlling the deflection, the minimum thickness is limited depending on the manner in which beams are supported. In this study, the beams are considered as non-prestressed at both continuous ends with allowable thickness of

$$h_{\min} = \frac{l}{21} \tag{17.18}$$

where l is the span of the beam. The penalty for the thickness of the beam can be expressed as

$$C_4 = \frac{h_{\min} - h}{h_{\min}} \tag{17.19}$$

If the rectangular compression-block depth is greater than the effective depth, the penalty is applied as

$$C_5 = \frac{a - d}{d} \tag{17.20}$$

In order to place and compact concrete between bars satisfactorily and provide proportionate bond, the minimum clear spacing $s_{\min}$ should be d_b but not <1 in. Here d_b is the diameter of reinforcement bars. The bar spacing penalty is

$$C_6 = \frac{s_{\min} - s}{s_{\min}} \tag{17.21}$$

Since the section capacities are evaluated separately, the reinforcement topology including bar spacing and steel ratio could be different in both sections at the support and mid-span while the dimensions are the same. For this reason, the same procedure for determining constraints related to reinforcement topology must be performed for the section under negative bending moment.

17.2.2.2 Column Constraints

A column section is acceptable when the design action effects defined by combination of M_n and P_n fall within the load–moment interaction diagram. The load–moment interaction penalty can be expressed as

$$C_7 = \frac{r - r_0}{r_0} \tag{17.22}$$

where r is the radial distance between the origin of the interaction diagram and the corresponding pair under the applied loading and r_0 is the radial distance between the origin of the interaction diagram and the intersection of vector r with the load–moment curve.

For compression members, the minimum longitudinal reinforcement $\rho_{\min}$ is limited to 0.01. The minimum reinforcement penalty is

$$C_8 = \rho_{\min} - \rho \tag{17.23}$$

For compression members, the maximum longitudinal reinforcement $\rho_{\max}$ is limited to 0.08. The maximum reinforcement penalty is

$$C_9 = \rho - \rho_{\max} \tag{17.24}$$

The clear distance between longitudinal bars should be 1.5 d_b but not <1.5 in. The longitudinal bar spacing penalty is

$$C_{10} = \frac{s_{\min} - s}{s_{\min}} \tag{17.25}$$

Since the bars are distributed along all four faces, the longitudinal bar spacing constraint must be checked in both width side and height side of the section.

17.3 Formulation of the Optimization Problem

17.3.1 Objective Functions

The optimal design criterion for reinforced concrete frames involves two different objective functions: The first objective function is based on the most economical solution that accounts for the cost of materials in terms of the concrete, the steel, and the labor cost in construction process. The second objective function quantifies the embedded CO_2 resulting from the use of materials, which involve emissions at different stages of the production and the placement of concrete and steel in structure. The unit costs and CO_2 emissions were obtained from the 2007 database of the Institute of Construction Technology of Catalonia [17]. It is important to note that the calculation of GHG or CO_2 emissions of buildings does not contain transport emissions including transportation for building materials, construction equipment, and workers, since transport distance from cradle to site is highly dependent on the case study. The general form of the objective function for current study can be expressed as

$$\begin{aligned} \min : f(x) &= \sum_{i=1}^{n} u_i m_i (x_1, x_2, \ldots, x_r) \\ \text{s.t. } & C_i(x_1, x_2, \ldots, x_r) \le 0 \end{aligned} \tag{17.26}$$

where u_i represents the unit prices or unit CO_2 emissions of material and construction components, m_i is the measurements of the construction units, x_i are the design variables, n is the number of construction members, r is the number of design variables, and C_i $(i = 1, 2, \ldots, n)$ are the design constraints.

17.3.2 Proposed Metaheuristic Algorithm

Metaheuristic algorithms are often based on the simulation of natural evolution and the principle of preservation or the survival of the fittest, which is a hypothetical population-based optimization procedure. In other words, a metaheuristic algorithm is an iterative process, which applies a set of agents to move through the design space and seek near-optimal solutions of the complex problems in a reasonably practical timescale. Although these optimization algorithms are usually nondeterministic, they make a reasonable trade-off between randomization and local search, this is why they can be used to find good feasible solutions in an acceptable time especially in case of intractable real-world problems. This chapter presents the application of a novel population-based stochastic algorithm so-called CBO which simulates a fundamental law of physics, namely collision between two bodies.

17.3.2.1 Enhanced Colliding Bodies Optimization Method

Collision is a short-term interaction between two bodies in which they are pushed away from each other and tend to form the most stable configuration and achieve the lowest energy state. According to the law of energy and momentum conservation, in all collisions the total amount of momentum possessed by the two objects does not change, i.e., the amount of momentum gained by one object is equal to the amount of momentum lost by the other object while the total kinetic energy after the collision may not be equal to the total kinetic energy before the collision and it changes to some other form of energy. What distinguishes different types of collisions is whether they conserve kinetic energy. When the total kinetic energy of system is lost, a perfectly inelastic collision occurs in which the two bodies stick together after the impact. Contrariwise if the total kinetic energy of system is conserved, a perfectly elastic collision occurs. The plot for this configuration is shown in Fig. 17.3.

In terms of this conception, the search ability of the CBO algorithm can be framed based on the interaction between colliding bodies (CBs) that are moving through predefined amplitude, starting with random initial positions to find near-optimal solutions. Each colliding body, as a solution candidate, contains a number of decision variables and is characterized by its position and velocity. The laws of

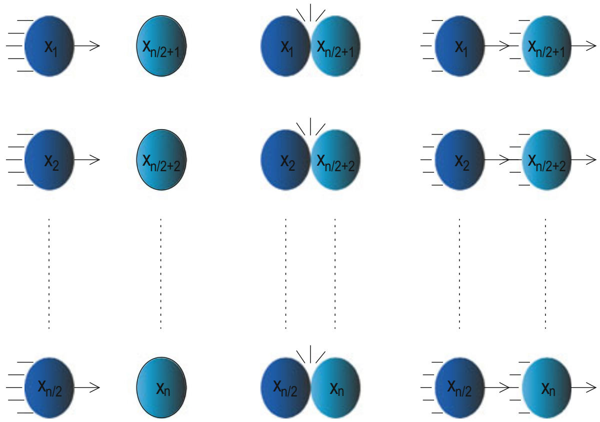


Fig. 17.3 The collision between the sorted pairs of CBs

energy conservation as well as linear momentum conservation allow us to adjust the changes of these attributes in two-body collisions.

The magnitude of the body mass for each CB is defined in association with the respective fitness value given as

$$m_i = \frac{\frac{1}{\mathrm{fit}(i)}}{\sum_{j=1}^{n} \frac{1}{\mathrm{fit}(j)}} \qquad i = 1,\ 2,\ \ldots,\ n \tag{17.27}$$

where fit is the objective function value of the CBs and n is an even number of colliding bodies. In order to select pairs of objects for collision, CBs are sorted according to the value of their objective function in an increasing order and divided into two equal groups. Agents with upper fitness values (moving objects) and finite speed push the corresponding agents with lower fitness values (stationary objects), which are at rest before the collision, toward better positions. The velocity of moving bodies before the collision is given as

$$v_i = x_i - x_{i-\frac{n}{2}} \qquad i = \frac{n}{2} + 1, \ldots, n \tag{17.28}$$

where x_i is the position vector of the ith CB in moving group and $x_{i-\frac{n}{2}}$ is the corresponding position vector in the stationary group.

After the collision, the attributes of each moving object are updated as follows:

$$v'_i = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}} \qquad i = \frac{n}{2} + 1\ , \ \ldots\ ,\ n \tag{17.29}$$

$$x'_i = x_{i-\frac{n}{2}} + r v'_i \tag{17.30}$$

where m_i is the mass of the ith moving CB, v_i is the velocity of the ith moving CB before the collision, $m_{i-\frac{n}{2}}$ is the mass of the ith stationary CB, $x_{i-\frac{n}{2}}$ is the previous position of the ith stationary CB, r is a random vector uniformly distributed in the range of (−1,1), and ε represents the coefficient of restitution defined as

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{17.31}$$

where *iter* is the number of iterations. Adjustment of this indicator changes the rate of intensification and diversification in the system and generally ranges between zero and one.

In addition, the attributes of each stationary object after the collision, which now has a velocity in the same direction of the moving object, are updated as follows:

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}}, \qquad i = 1\,,\ \ldots\,,\ \frac{n}{2} \tag{17.32}$$

$$x_i' = x_i + r v_i' \tag{17.33}$$

where m_i is the mass of the ith stationary CB, $m_{i+\frac{n}{2}}$ is the mass of the ith moving CB, $v_{i+\frac{n}{2}}$ is the velocity of the ith moving CB before the collision, and x_i is the previous position of the ith stationary CB.

Historical best solutions are saved by employing the colliding memory (CM) which stores some best solutions of each iteration found in previous population and substitutes them with some current worst CB vectors. Introducing new best bodies into the population prevents the population from moving only to neighboring states and speeds up the convergence rate without increasing the computational cost.

In order to break one or more members of the population out of local minima and produce a more efficient search, one component of the ith CB is regenerated in a random manner in any given generation. The probability of choosing the component is expressed as Pro, which ranges between (0, 1).

In accord with the given definition, enhanced colliding bodies algorithm is a continuous variable-based method improved by saving the best solutions and regenerating random members of population occasionally to produce a more efficient and reliable solution. The steps of this algorithm can briefly be outlined as follows:

Step 1: Randomly initialize the vector of CBs with n variables and evaluate their associated fitness function.

Step 2: Store some best solutions of each iteration in the colliding memory and replace them with the current worst CB vectors.

Step 3: Calculate the mass value for each CB using Eq. (17.27).

Step 4: Sort the fitness value of the objective function for each CB in an increasing order, and then determine the pairs of CBs for collision.

Step 5: Evaluate the velocity of moving bodies before the collision using Eq. (17.28).

Step 6: Update the velocities of stationary and moving bodies after the collision using Eqs. (17.32) and (17.29), respectively.

Step 7: Update the positions of stationary and moving bodies using the generated velocities after the collision in Step 6 and Eqs. (17.33) and (17.30), respectively. If some bodies' new positions violate the boundaries, correct their position and return to the specified domain.

Step 8: Compare *Pro* with a random number, rn_i $(i = 1, 2 \ldots n)$, which is distributed uniformly between (0, 1), if $rn_i < pro$, randomly select a CB from both moving and stationary group and regenerate one related component accidentally.

Step 9: Return to Step 2 until a terminating criterion is satisfied.

17.3.2.2 Non-dominated Sorting Enhanced Colliding Bodies Optimization

The proposed multi-objective algorithm is based on an improved version of the non-dominated sorting genetic algorithm, called NSGA-II, which is proposed by Deb et al. [18]. The non-dominated sorting genetic algorithm (NSGA) is based on some modifications to the ranking procedure of the individuals, originally proposed by Goldberg.

The basic design concept of NSGA-II is to find a set of non-dominated and evenly distributed solutions using two ranking techniques called non-dominated sorting and crowding approach. Each individual in population is assigned a rank on the basis of non-domination before selection. All non-dominated solutions are ranked 1. In other words, these individuals are assigned the highest rank. Then, this group of classified individuals is removed from the population and another set of non-dominated individuals from the remaining population are ranked. This group of classified individuals is also removed. This process continues until all individuals in the objective function space are classified. In order to provide a diversity and uniform distribution across the Pareto front, individuals at the same non-domination front are compared with a crowding distance. This helps the algorithm to explore the search space. After sorting procedure, the evolutionary operations are adopted to create new pool of offspring, and then the parents and offspring are combined.

Considering the basic concept of NSGA-II, in order to select pairs of objects for collision, CB vector of each iteration is sorted by non-dominated sorting and crowding approach. Since agents in the first front have the maximum fitness value, they push the corresponding agents with the lower fitness value (stationary objects). The ranking techniques are also adopted to store some best CB vectors into the colliding memory.

17.4 Design Examples

In order to demonstrate the efficiency and performance of the proposed algorithms, three symmetric multistory and multi-bay benchmark problems of reinforced concrete frames are adapted and solved: the first example is a two-bay six-story frame originally designed by Rajeev and Krisnamoorthy [5] and redesigned by Camp et al. The remaining examples are a two-bay four-story frame and a two-bay six-story frame presented by Paya-Zaforteza et al. [8] and redesigned by Camp and Huq [13]. In order to compare the results with those of the previous researches, the same assumptions are followed. It is important to note that the assessment of the frames originally designed by Paya-Zaforteza et al. [8] follows the Spanish Code of structural concrete [19].

17.4.1 Two-Bay Six-Story Frame

Figure 17.4 illustrates the two-bay six-story frame originally designed by Rajeev and Krisnamoorthy [5] using standard GA algorithm and redesigned by Camp at al. [6, 13] using GA and BB–BC algorithm. The height of each story is 4 m and the span of the left and right bay is 6 m and 4 m, respectively. The optimal dimension of width for beam and column sections is considered between (200, 460) mm and (150, 560) mm, respectively. The step of increment for both beam and column sections is 30 mm. As shown in Fig. 17.4, the frame consists of 12 beams and

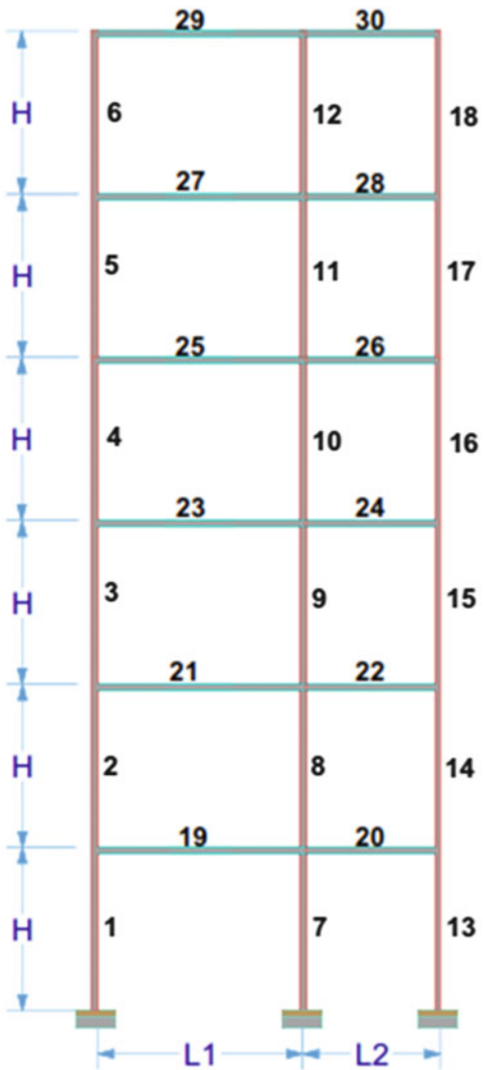


Fig. 17.4 Two-bay six-story RC plane frame

18 columns arranged in 4 beam groups and 3 column groups according to case 1 of Table 17.2. A factored uniformly distributed dead load of 30 kN/m is applied on each beam, and the lateral equivalent static load of 10 kN is applied as joint load at each story level. Concrete has the compressive strength of 20 MPa, and the unit weight of 2323 kg/m^3. Reinforcement has the yield strength of 414 MPa, and the unit weight of 7849 kg/m^3. The number of DB sections created for beams and columns are 7128 and 9450, respectively, which results in a design space of 2.17e27. The frame has a total of 36 design variables, which define the geometry of the cross sections, the reinforcement bar size, and the number of reinforcing bars. Due to the number of design variables and the size of the design space, a small population of 12 with a typical stopping criterion of 3000 was required. In all cases the algorithm is executed 50 times to obtain the best statistical data of the results. Based on the examinations, the suitable values for the parameter Pro and CM are taken as 0.35 and $np/2$, respectively. Where np is the number of CBs. The objective function is implemented to minimize the structural cost defined as

$$f_k = \sum_{i=1}^{n_b+n_c} \{C_c b_i h_i + C_s A_{si} + 2C_f(b_i + h_i)\} l_i \tag{17.34}$$

where C_c is the unit cost of concrete, C_s is the unit cost of steel reinforcement, A_{si} is the area of reinforcing bars, C_f is the unit cost of formwork, n_b is the number of

Table 17.2 Different type of grouping for two-bay six-story frame

Member type	Group no.	Grouping type					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Beam	1	29	29	19-20	19-21	19-20	19-20
	2	30	30	21-22	20-22	21-22	21-22
	3	19-21-23-25-27	19-21-23-25-27	23-24	23-25	23-24	23-24
	4	20-22-24-26-28	20-22-24-26-28	25-26	24-26	25-26	25-26
	5	–	–	27-28	27-29	27-28	27-28
	6	–	–	29-30	28-30	29-30	29-30
Column	1	1-2-3-4-5-6	1-7-13-2-8-14	1-13	1-13	1-7	1
	2	7-8-9-10-11-12	3-9-15-4-10-16	2-14	2-14	2-8	2
	3	13-14-15-16-17-18	5-11-17-6-12-18	3-15	3-15	3-9	3
	4	–	–	4-16	4-16	4-10	4
	5	–	–	5-17	5-17	5-11	5
	6	–	–	6-18	6-18	6-12	6
	7	–	–	7	7	13	7-13
	8	–	–	8	8	14	8-14
	9	–	–	9	9	15	9-15
	10	–	–	10	10	16	10-16
	11	–	–	11	11	17	11-17
	12	–	–	12	12	18	12-18

beams, and n_c is the number of columns. The unit costs of concrete, steel, and formwork are estimated as $735/m^3, $7.1 /kg, and $54/m^2, respectively.

Table 17.3 compares the results obtained by the proposed algorithm with the previous solutions.

The best solution reported by the ECBO is 23,081.57$. The best ECBO design is 2.46 % less than the best solution given by BB–BC.

Five more types of grouping are considered for the design of frame listed in Table 17.2. The comparison of the solutions (Fig. 17.5) shows a maximum of 4.39 % decrease in cost for case 2 of grouping. Since the members in the same group have the same design variables, the capacity violations must be relatively close. More precisely, the internal force distributions in each group, which is highly related to the load pattern, should have insignificant difference as much as possible. Hence, the pattern of grouping should match closer to the internal force distributions while the number of groups should compromise between the economic design and computing time. The information pertaining to compare the strength ratio between different cases of grouping has been quantified in Fig. 17.6.

One of the best approaches to handle the constraints is evaluating the fitness function in the feasible search space. This approach is called death penalty. The feasible region is achieved by rejection of infeasible individuals. Some of the geometric constraints can be applied during the process of creating DB sections. Therefore, no further calculations are necessary to enforce these constraints on the objective function. This technique is limited to problems in which the constraints are not dependent on the geometric information related to the structure. The remaining constraints to be checked in each iteration are the capacity (C_1, C_7) and the allowable thickness (C_4) restrictions. Taking the abovementioned procedure into account, the size of the search space is declined to 7.28e24 (Table 17.4). The algorithm could attain the similar best solution in a significant short iteration number of 800 and computational time of 0.46 s which is 6.93 times faster than case 1. With the stopping criterion of 3000, it could decrease the solution by 2.73 % with the computational time of 2.56 s, which is 1.24 times faster than case 1. As shown in Fig. 17.7, the speed of convergence to the optimum value has had a considerable increase.

17.4.2 Two-Bay Four-Story Frame

Figure 17.8 illustrates the two-bay four-story frame originally designed by Paya-Zaforteza et al. [8] using SA algorithm and redesigned by Camp et al. [13] using BB–BC algorithm. The height of each story is 3 m, and the span of each bay is 5 m. The optimal dimension of width for beam and column sections is considered between (150, 1200) mm and (250, 1200) mm, respectively. The step of increment for beam sections is 10 mm and for column sections is 50 mm. As shown in Fig. 17.8, the frame is consisted of 8 beams and 12 columns arranged in 4 beam

Table 17.3 Best design for two-bay six-story frame

Member type	Group no.	GA [6]			BB–BC [13]			Kaveh and Ardalani [1]		
		Width (mm)	Depth (mm)	Bars	Width (mm)	Depth (mm)	Bars	Width (mm)	Depth (mm)	Bars
Beam	1	280	560	2#6+2#8	360	480	3#5+1#10	230	530	2#6+1#8
	2	330	480	1#5+2#7	330	430	1#9+1#10	200	370	1#6+1#8
	3	230	560	4#4+1#11	200	480	2#6+2#9	200	490	1#8+1#11
	4	200	480	1#6+2#5	230	330	2#5+2#6	200	430	3#4+2#7
Column	1	180	200	4#5	180	280	4#5	180	270	4#4
	2	180	460	4#7	280	250	8#5	210	330	4#5
	3	180	280	4#4	150	200	6#3	210	360	6#4
Best Cost ($)		24,959			23,664			23,081.57		
Average ($)		–			26,520.55			27,028.98		
Std deviation ($)		–			1069.91			2695.02		

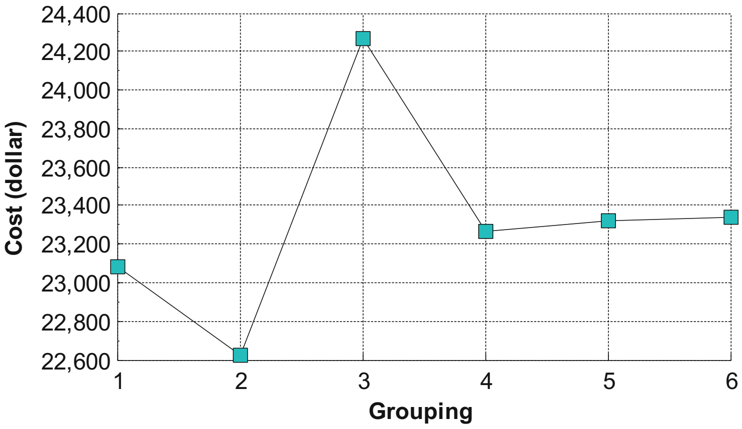


Fig. 17.5 Best cost design for different cases of grouping

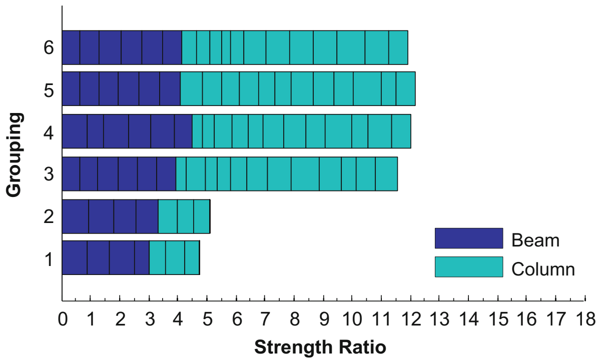


Fig. 17.6 Strength ratio in the groups for different cases of grouping

Table 17.4 Best cost design in different size of the search space and number of iteration

Description	Case 1	Case 2	Case 3
Database of beam	7128	3330	3330
Database of column	9450	3898	3898
Search space	2.17e27	7.28e24	7.28e24
Iteration	3000	3000	800
Best cost ($)	23,081.57	22,450.4	23,008.15
Computation time (s)	3.19	2.56	0.46

groups and 8 column groups. The spacing considered between adjacent parallel frames is 5.00 m, and the thickness of the slab for all stories is 290 mm. Twelve load combinations that include counteracting effects of dead, live, and wind loads are taken into account to determine the required strength of the members as listed below:

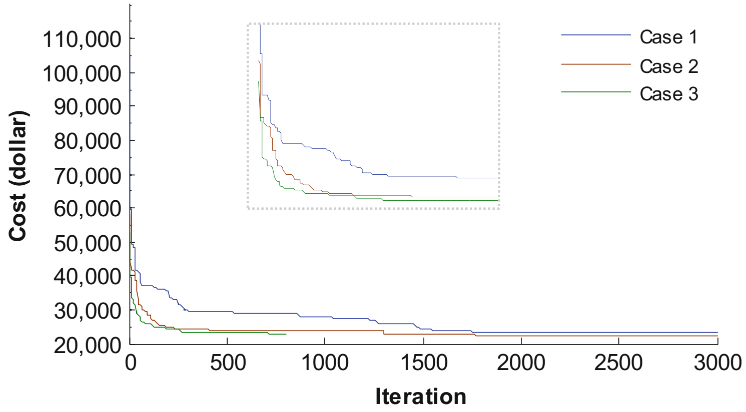


Fig. 17.7 Convergence rate in different size of the search space and number of iteration [1]

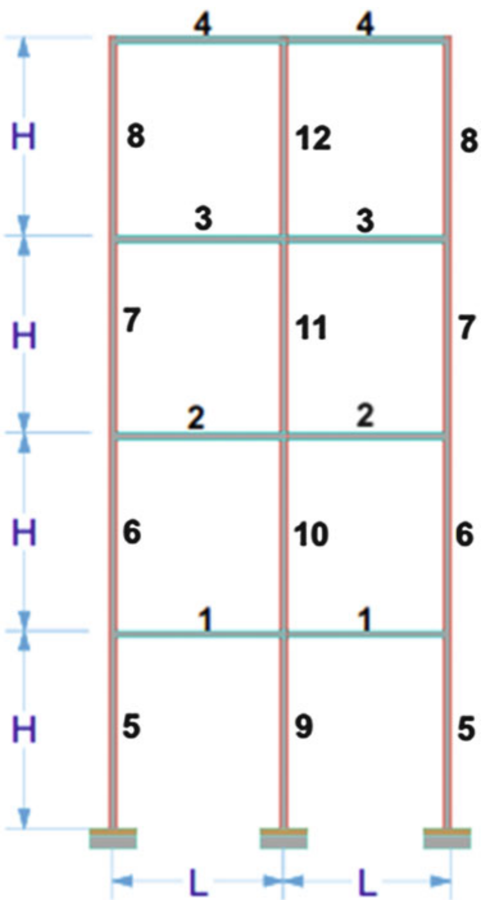


Fig. 17.8 Two-bay four-story RC plane frame

Table 17.5 The applied loads on the frame

Action	Value
DL in story 1–3 (kN/m^2)	4
DL in story 4 (kN/m^2)	6
LL in story 1–3 (kN/m^2)	3
LL in story 4 (kN/m^2)	1
WL in story 1 (kN)	8.83
WL in story 2 (kN)	9.86
WL in story 3 (kN)	10.74
WL in story 4 (kN)	5.81

$$U = 1.5D \tag{17.35a}$$

$$U = 1.5D + 1.6L1 \tag{17.35b}$$

$$U = 1.5D + 1.6L2 \tag{17.35c}$$

$$U = 1.5D + 1.6LT \tag{17.35d}$$

$$U = 1.5D + 1.6W1 \tag{17.35e}$$

$$U = 1.5D + 1.6W2 \tag{17.35f}$$

$$U = 1.5D + 1.44L1 + 1.44W1 \tag{17.35g}$$

$$U = 1.5D + 1.44L2 + 1.44W1 \tag{17.35h}$$

$$U = 1.5D + 1.44LT + 1.44W1 \tag{17.35i}$$

$$U = 1.5D + 1.44L1 + 1.44W2 \tag{17.35j}$$

$$U = 1.5D + 1.44L2 + 1.44W2 \tag{17.35k}$$

$$U = 1.5D + 1.44LT + 1.44W2 \tag{17.35l}$$

where D is the uniform dead load applied to each beam, $L1$ stands for the live load applied to only one beam in each story while the bays change alternatively, $L2$ is the uniform live load applied in a pattern opposite of $L1$, $W1$ is the wind load applied to the left side of the frame, and $W2$ is the wind load applied to the right side of the frame. Table 17.5 lists the values of the uniform loads and wind loads at each story. Compressive strength of concrete varies in each story from 25 to 50 MPa with the increment step of 5 MPa. The unit weight of concrete is $2323\,\mathrm{kg/m^3}$. Reinforcement has the yield strength of 500 MPa, and the unit weight of $7849\,\mathrm{kg/m^3}$. The number of DB sections created for beams and columns are 98,424 and 7584, respectively, which results in a design space of 2.23e60. The frame has a total of 60 design variables. Hence, the population of 16 CBs with a typical stopping criterion of 4000 was required. In this example, two objective functions are implemented to minimize cost and CO_2 emissions in terms of the materials and construction process. The general form of the cost function is defined as

$$f_k = \sum_{i=1}^{n_b+n_c} \{C_c b_i h_i + C_s A_{si}\} l_i + \sum_{i=1}^{n_b} \{C_f(b_i + 2(h_i - t_i)) + C_t b_i\} l_i + \sum_{i=1}^{n_c} \{2C_f(b_i + h_i)\} l_i \quad (17.36)$$

where C_t is the unit rate of scaffolding and t_i is the thickness of the slab. The CO_2 emission function has the same form of the cost function; however, the unit values are different and also the scaffolding term is not considered. The unit rates for cost and CO_2 emissions are listed in Table 17.6.

The results for single objective of cost function obtained by the proposed algorithm and the previous research works are compared in Table 17.7. The best solution reported by the ECBO is 3429.92€ with 3587.88 kg of CO_2 emissions. The best ECBO cost design is 3.13 % less than the best solution given by BB–BC. Concrete represents 18.22 % of the total cost, while reinforcing steel about 25.55 % of the total cost. Table 17.8 compares the results for single objective of CO_2 emission functions. The best solution reported by the ECBO is 3238.25 kg with a cost of 3525.27€. The best ECBO CO_2 design is 2.67 % less than the best solution given by BB–BC. The percentage comparison of the solutions indicates that the best CO_2 emission design decreased the CO_2 emissions by 9.74 % with a slight increase in cost of 2.77 %. Since more environmentally friendly solutions are recommended by IPCC, on the other hand, the low-CO_2 emission design could decrease the CO_2 emissions considerably at an acceptable cost increment in practice; it seems that designing the RC structures based on the CO_2 emissions is more logistical (Table 17.9).

Figure 17.9 compares the strength ratio in element groups for both cost and CO_2 objective functions. As can be seen, in beam groups the use of section capacity in low-cost design is lower than low-CO_2 emission design, while in column groups the use of section capacity is higher. This finding shows that there is a relationship between the geometry of frame and the objective functions. Table 17.2 indicates the

Table 17.6 Unit prices and CO_2 emissions

	Cost (€)		CO_2 (kg)	
Description	Beam	Column	Beam	Column
Steel B-500 (kg)	1.3	1.3	3.01	3.01
Concrete HA-25 (m^3)	78.40	77.80	132.88	132.88
Concrete HA-30 (m^3)	82.79	82.34	143.48	143.48
Concrete HA-35 (m^3)	98.47	98.03	143.77	143.77
Concrete HA-40 (m^3)	105.93	105.17	143.77	143.77
Concrete HA-45 (m^3)	112.13	111.72	143.77	143.77
Concrete HA-50 (m^3)	118.60	118.26	143.77	143.77
Form work (m^2)	25.05	22.75	3.13	8.90
Scaffolding (m^2)	38.89	–	4.86	–

Table 17.7 Design results for cost objective for two-bay four-story frame

Member type	Group no.	BB–BC [13] Concrete strength (MPa)	BB–BC [13] Width (mm)	BB–BC [13] Depth (mm)	BB–BC [13] Bars	Kaveh and Ardalani [1] Concrete strength (MPa)	Kaveh and Ardalani [1] Width (mm)	Kaveh and Ardalani [1] Depth (mm)	Kaveh and Ardalani [1] Bars
Beam	1	40	180	430	1#8+2#8	30	220	430	2#7+3#7
	2	40	180	450	1#10+2#8	30	250	450	2#7+4#5
	3	30	190	460	1#8+1#11	30	220	440	3#6+3#6
	4	25	220	530	4#4+1#10	25	220	430	1#9+3#7
Column	1	40	250	550	6#5	30	300	500	8#3
	2	40	250	300	4#5	30	300	400	6#4
	3	30	250	300	4#6	30	250	350	8#3
	4	25	250	300	6#6	25	250	350	12#4
	5	40	250	300	4#5	30	300	450	6[a]#4
	6	40	250	250	8#5	30	250	250	4#4
	7	30	250	250	6#4	30	250	300	4#3
	8	25	250	250	4#3	25	250	250	4#3
Best cost (€)		3540.88				3429.92			
Average (€)		3790.25				3682.09			
Std deviation (€)		139.28				156.51			
CO_2 emission (kg)		3778.24				3587.88			

Table 17.8 Design results for CO_2 objective for two-bay four-story frame

		BB–BC [13]				Kaveh and Ardalani [1]			
Member type	Group no.	Concrete strength (MPa)	Width (mm)	Depth (mm)	Bars	Concrete strength (MPa)	Width (mm)	Depth (mm)	Bars
Beam	1	50	210	510	2#5+3#6	40	230	420	1#8+3#7
	2	30	220	530	2#5+1#10	40	240	510	4#4+3#6
	3	25	210	520	2#5+2#7	25	250	550	4#4+3#5
	4	25	240	590	4#4+1#9	25	260	560	2#6+3#5
Column	1	50	250	400	6#3	40	250	450	6[a]#3
	2	30	250	400	6[a]#3	40	250	400	6[a]#3
	3	25	250	400	10#3	25	250	450	8#4
	4	25	250	400	4#6	25	250	250	6#4
	5	50	250	450	10#3	40	250	350	4#4
	6	30	250	400	4#3	40	250	350	4#4
	7	25	250	300	12#3	25	250	300	6#3
	8	25	250	250	4#3	25	250	250	4#3
Best CO_2 emission (kg)		3327.29				3238.25			
Average (kg)		3650.33				3554.30			
Std deviation (kg)		127.81				216.83			
Cost (€)		3617.06				3525.27			

Table 17.9 Ratio between Cost and CO_2-optimized design variables

Group no.	Frame characteristics	
	Concrete strength	Area of elements
1	0.75	0.97
2	0.75	0.91
3	1.2	0.70
4	1	0.64
5	0.75	1.33
6	0.75	1.20
7	1.2	0.77
8	1	1.40
9	0.75	1.54
10	0.75	0.71
11	1.2	1
12	1	1

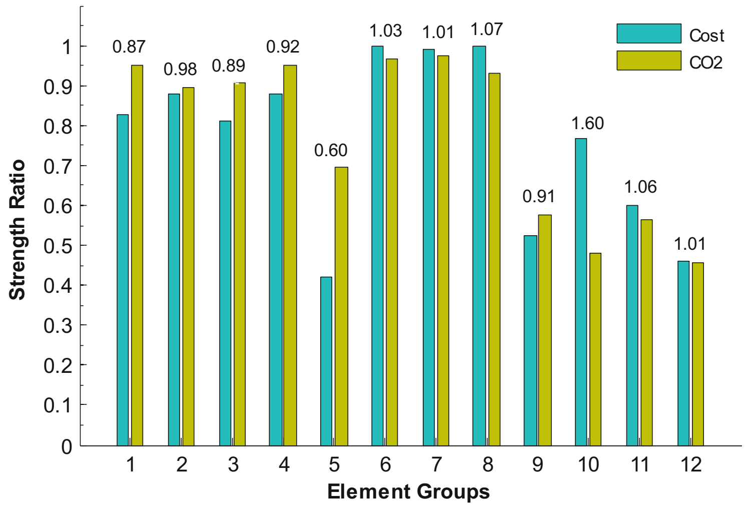


Fig. 17.9 Strength ratios in element groups for both cost and CO_2 objective functions

ratio between cost and CO_2-optimized design variables. The dimension of beams are bigger over the low-CO_2 emission design than over the low-cost design.

In Table 17.10, the percentage of cost and CO_2 emissions is quantified for materials and construction components. Concrete, reinforcing steel, formwork, and scaffolding represent approximately 26, 18, 46, and 10 % of the total cost and 50, 35, and 15 % of the total emissions, respectively.

Table 17.11 summarizes the results of the ECBO single-objective and multi-objective designs. The best NSECBO design with lower cost is 3490€ with 3475 kg of CO_2 emissions which are 1.78 % and 7.31 % higher compared to single-objective designs of cost and CO_2 emissions, respectively. Alternatively, the best NSECBO

Table 17.10 Percentage of total cost and CO_2 emissions

	Cost (%)			CO_2 (%)		
Description	Beam	Column	Total	Beam	Column	Total
Steel	70	30	26	70	30	50
Concrete	52	48	18	60	40	35
Form work	33	67	46	18	82	15
Scaffolding	10	–	10	–	–	–
Total			100			100

Table 17.11 Results of the ECBO single-objective and multi-objective designs

Objective	Cost (€)	CO_2 (%)
ECBO-Cost	3429	3587
NSECBO-Cost	3490	3475
ECBO-CO_2	3525	3238
NSECBO-CO_2	3520	3318

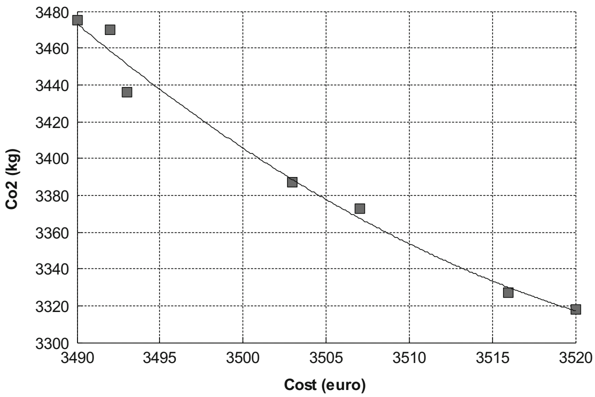


Fig. 17.10 NSECBO Pareto front

design with lower emissions is 3318 kg with a cost of 3520€, which are 2.47 % and 2.65 % higher, respectively. Both objectives are closely related and result in similar solutions. All these lead to a tentative conclusion that the CO_2 and cost objectives should be considered together in RC structural designs. The Pareto front is presented in Fig. 17.10.

17.4.3 Two-Bay Six-Story Frame with Unequal Bays

Figure 17.11 illustrates the two-bay six-story frame originally designed by Paya-Zaforteza et al. [8] using SA algorithm and redesigned by Camp et al. [13] using BB–BC algorithm. The story height and bay span of the frame and the search space specifications are the same as defined for the two-bay four-story frame in Example

2. As shown in Fig. 17.11, the frame consists of 12 beams and 18 columns, which are arranged in 6 beam groups and 12 column groups. The type of grouping, spacing considered between adjacent parallel frames, the thickness of the slab, the strength and the unit weight of concrete and steel, the load patterns, and the magnitude of

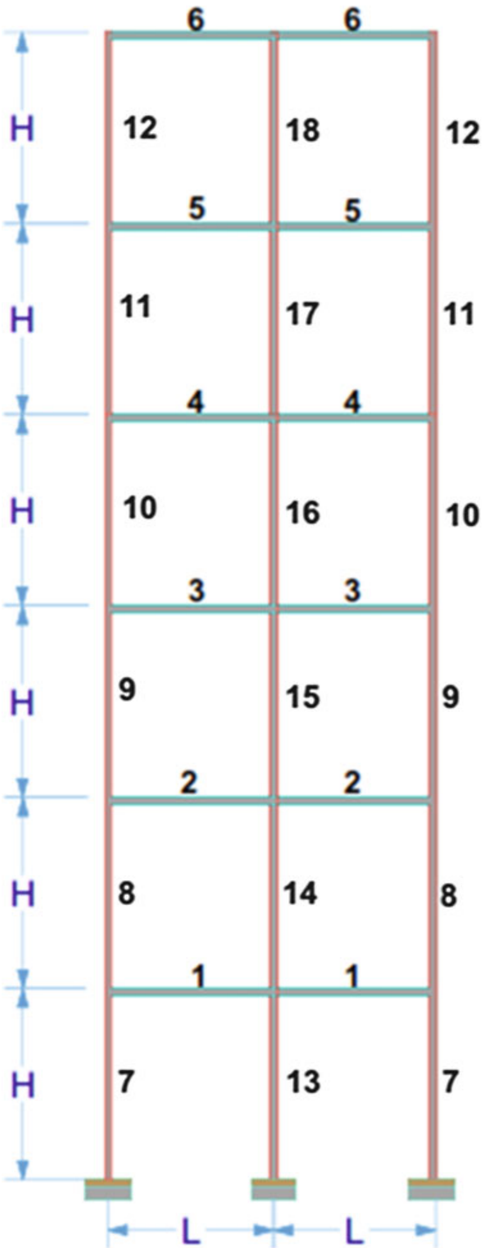


Fig. 17.11 Two-bay six-story RC plane frame

Table 17.12 Wind loads for two-bay six-story frame

Action	Value
WL in story 1	8.83
WL in story 2	9.86
WL in story 3	10.74
WL in story 4	11.62
WL in story 5	12.36
WL in story 6	6.62

loads except the wind loads are the same as in Example 2. Table 17.12 lists the values of the wind loads at each story. The frame has a total of 90 design variables and the design space of 3.34e90. The general form of the objective functions is given in Eq. (17.36).

Table 17.13 compares the results for single objective of cost function obtained by the proposed algorithm with those of the previous researches. The best solution reported by the ECBO is 5697.98€ with 5834.72 kg of CO_2 emissions. The best ECBO cost design is 2.29 % less than the best solution given by BB–BC. Table 17.14 compares the results for single objective of CO_2 emission functions. The best solution reported by the ECBO is 5682.82 kg with a cost of 5913.02€. The best ECBO CO_2 design is 2.16 % less than the best solution given by BB–BC. The percentage comparison of the solutions confirms the previous findings.

17.5 Concluding Remarks

This chapter aimed to evaluate the usefulness of the ECBO and NSECBO through the optimization of three multistory-multi-bay frames based on the ACI Code including architectural and reinforcement detailing. The algorithm is applied to two objective functions: the cost of material and the embedded CO_2 emissions during the construction process. Based on the present work, the following conclusions can be derived:

1. The ECBO design improved the results from both objective functions in a reasonably practical time over the designs developed by the BB–BC algorithm. Moreover, in comparison with other evolutionary approaches, the ECBO algorithm is simple to implement and it requires a few parameters to be set. These findings proved that ECBO-based methodology could be applied as an effective and powerful algorithm to arrive at a realistic design solution for real complex problems.
2. Conclusive solution of the algorithm is improved through selecting more rational groups of the elements. This implies that grouping in which the members in the same group are similar in the internal force distribution results in more economical solutions.

Table 17.13 Design results for cost objective for two-bay six-story frame

		BB–BC [13]				Kaveh and Ardalani [1]			
Member type	Group no.	Concrete strength (MPa)	Width (mm)	Depth (mm)	Bars	Concrete strength (MPa)	Width (mm)	Depth (mm)	Bars
Beam	1	45	180	420	3#5+2#9	50	220	410	1#10 +2#9
	2	30	210	500	1#8+3#7	40	270	470	3#7+4#6
	3	30	200	500	2#6+3#7	35	220	440	2#6+3#7
	4	30	200	470	1#8+3#7	35	270	480	3#5+3#6
	5	25	210	520	1#10+3#6	30	260	480	3#7+3#6
	6	25	230	570	2#6+2#7	30	250	460	2#7+3#6
Column	1	45	250	650	6[a]#3	50	250	300	6#3
	2	30	250	500	6#3	40	250	450	8#4
	3	30	250	400	4#4	35	250	500	10#3
	4	30	250	400	4#6	35	300	300	8#3
	5	25	250	300	6#6	30	250	450	4#6
	6	25	250	250	6#6	30	250	300	6[a]#5
	7	45	250	400	12#4	50	350	650	6[a]#5
	8	30	250	400	12#5	40	250	250	6[a]#3
	9	30	250	350	10#6	35	300	500	6#4
	10	30	250	300	8#6	35	250	450	8#3
	11	25	250	300	6#4	30	250	250	4#4
	12	25	250	250	6#4	30	250	250	4#3
Best cost (€)		5831.70				5697.98			
Average (€)		6416.73				6236.61			
Std deviation (€)		219.05				369.43			
CO_2 emission (kg)		6306.40				5834.72			

Table 17.14 Design results for CO_2 objective for two-bay six-story frame

Member type	Group no.	BB–BC [13] Concrete strength (MPa)	BB–BC [13] Width (mm)	BB–BC [13] Depth (mm)	BB–BC [13] Bars	Kaveh and Ardalani [1] Concrete strength (MPa)	Kaveh and Ardalani [1] Width (mm)	Kaveh and Ardalani [1] Depth (mm)	Kaveh and Ardalani [1] Bars
Beam	1	35	230	560	1#7+1#11	50	300	500	1#9+2#8
	2	30	220	550	3#4+2#8	50	290	530	3#6+3#7
	3	25	250	620	4#4+4#5	45	230	450	3#7+3#7
	4	25	230	550	1#8+3#6	45	250	430	2#6+3#7
	5	25	230	550	1#8+1#10	40	250	490	1#9+3#6
	6	25	230	550	1#8+3#6	30	260	510	3#7+3#5
Column	1	35	250	500	6[a]#3	50	250	350	6#4
	2	30	250	450	4#6	50	250	250	4#3
	3	25	250	450	4#5	45	250	500	6#4
	4	25	250	400	6#5	45	350	350	10#3
	5	25	250	300	6#5	40	250	300	6#3
	6	25	250	250	4#7	30	250	450	4#7
	7	35	700	250	4#3	50	250	300	4#3
	8	30	700	250	8#3	50	250	500	8#3
	9	25	700	250	6[a]#3	45	250	500	8#3
	10	25	500	250	10#3	45	300	500	8#3
	11	25	250	250	4#7	40	250	350	4#4
	12	25	250	250	4#3	30	250	250	4#3
Best CO_2emission (kg)		5808.70				5682.82			
Average (kg)		6392.72				6134.52			
Std deviation (kg)		279.59				403.39			
Cost (€)		5948.81				5913.02			

3. Considerable reduction of the size of the search space by rejection of infeasible individuals during the process of creating DB sections and eliminating the related terms of violation from the penalty function can reduce the calculation time and give a very rapid convergence in the early iterations toward the feasible solution. Moreover, with the same number of iterations and qualifications, the best solution decreases significantly.
4. Investigating the relationship between the two objective functions of cost and CO_2 emissions indicates that although the CO_2 emission function causes a relative increase in the cost, it decreases the CO_2 emissions by up to 9.74 %. Due to the growing efforts and the IPCC recommendation to reduce the atmospheric concentration of CO_2 caused by construction industry, it appears that optimal design of RC structures with respect to the CO_2 emissions as the key control point of the low carbon economy and a sustainable environment is more rational.
5. Comparison between the cost and CO_2-optimized design variables indicates that the geometry and physical dimension of elements are different in a way that the beam areas are bigger over the low-CO_2 emission design than over the low-cost design.
6. The results of the ECBO single-objective and multi-objective designs reveal that both objective functions yield similar solutions and economical solutions also perform well in terms of CO_2 emissions.

References

1. Kaveh A, Ardalani S (2016) Cost and CO_2 emission optimization of reinforced concrete frames using ECBO algorithm. Asian J Civil Eng 17(6):831–858
2. Mehta PK (2002) Greening of the concrete industry for sustainable development. Concr Int 24:23–28
3. IPCC (2007) The fourth assessment report of the Intergovernmental Panel on Climate Change. IPCC, Geneva
4. Worrell E, Price L, Martin N, Hendriks C, Meida LO (2001) Carbon dioxide emissions from the global cement industry. Annu Rev Energy Environ 26:303–329
5. Rajeev S, Krisnamoorthy CS (1998) Genetic algorithm-based methodology for design optimization of reinforced concrete frames. Comput Aided Civil Infrastruct Eng 13:63–74
6. Camp CV, Pezeshk S, Hansson H (2003) Flexural design of reinforced concrete frames using a genetic algorithm. Struct Eng ASCE 129:105–115
7. Lee C, Ahn J (2003) Flexural design reinforced concrete frames by genetic algorithm. Struct Eng 129:762–774
8. Paya-Zaforteza I, Yepes V, Gonzalez-Vidosa F, Hospitaler A (2008) Multiobjective optimization of concrete frames by simulated annealing. Comput Aided Civil Infrastruct Eng 23:596–610
9. Kwak HG, Kim J (2009) An integrated genetic algorithm complemented with direct search for optimum design of RC frames. Comput Aided Des 41:490–500
10. Kaveh A, Sabzi O (2011) A comparative study of two meta-heuristic algorithms for optimum design of reinforced concrete frames. Struct Eng 9:193–206
11. Akin A, Saka MP (2015) Harmony search algorithm based optimum detailed design of reinforced concrete plane frames subject to ACI 318-05 provisions. Comput Struct 147:79–95

12. Paya-Zaforteza I, Yepes V, Hospitaler A, Gonzalez-Vidosa F (2009) CO_2 optimization of reinforced concrete frames by simulated annealing. Eng Struct 31:1501–1508
13. Camp CV, Huq F (2013) CO_2 and cost optimization of reinforced concrete frames using a Big Bang–Big Crunch algorithm. Eng Struct 48:363–372
14. American Concrete Institute (2008) Building code requirements for structural concrete and commentary. ACI 318-08
15. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
16. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
17. Database BEDEC PR/PCT (2007) Institute of Construction Technology of Catalonia, Barcelona (in Spanish)
18. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
19. Fomento M (1998) Code of structural concrete. EHE-08, Madrid (in Spanish)

Chapter 18
Construction Site Layout Planning Using Colliding Bodies Optimization and Enhanced Colliding Bodies Optimization

18.1 Introduction

In this chapter, two recently developed metaheuristic algorithms, so-called CBO and ECBO, are employed for construction site layout planning. Results show that both of these algorithms have the capability of solving this kind of problem. Two case studies are presented to show the applicability and performance of the utilized methods [1].

Suitable facility layout is believed to be the heart of efficient production (Wong et al. [2]) that should be considered early in the planning phase [3]. An appropriate construction site layout boosts the effectiveness and efficiency of the works. However, the arrangement of site facilities is hindered by many constraints such as limitations on the site area, adjacent buildings, access, the location and orientation of the building to be constructed, Ref. [4].

The objective of the construction site layout is to arrange the temporary facilities such as job office, labor residence, warehouse, and batch plants (Adrian et al. [3]), so that all design requirements are fulfilled and maximum design quality is achieved in terms of design preferences such as minimizing the total cost associated with the interactions among these facilities [5]. Based on studies carried out in the manufacturing industry, the material handling costs can be reduced by 20–60 % if appropriate facility layout is adopted [4].

Since site layout planning is an intricate task, the construction managers often implement this task using previous experiences, ad hoc rules, and first-come-first-serve approach which leads to inefficiency (Adrian et al. [3], and Said and El-Rayes [6]). Therefore, an effective construction site layout planning (CSLP) is of utmost importance for the success of a construction project [7].

Construction site layout problem can be modeled as a quadratic assignment problem (QAP) when the costs associated with flow between departments are assumed to be linear with respect to distance traveled and quantity of the flow [8]. The QAP is one of the classical combinatorial optimization problems, which is

A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, DOI 10.1007/978-3-319-48012-1_18

known for its diverse applications and is widely regarded as the most difficult problem in classical combinatorial optimization [9]. QAP problems are known as non-polynomial hard problems (NP-hard) and because of the combinatorial complexity, these cannot be solved exhaustively for reasonably sized layout problems [5]. As an example, for n facilities, the number of possible alternatives, that is the number of feasible configurations, is $n!$ with larger growth than e^n. This is a huge number, even for a small n. For 10 facilities, the number of possible alternatives is already well over 3,628,000. For 15 facilities, we are already in the 12-digit numbers. In real problems, a project with $n = 15$ can be considered as a small project [10].

Due to the complexity of the site layout problems, numerous techniques have been proposed to find solutions to these problems; however, it is very difficult to obtain an optimal one suitable for hand calculations. Thus, optimization techniques seem to be suitable means to search for solutions of the site layout problems. The problem can be solved using two classes of techniques: exact algorithms and approximate algorithms. Exact algorithms such as mathematical optimization procedures were designed to find optimum solutions. But these methods could not be adopted for large-scale projects because of the need for huge calculations and computational efforts [11]. Therefore, they have been only successful for a single or very limited number of facilities, as reported by Tommelein et al. [12]. Approximate algorithms are categorized into two groups, heuristic and metaheuristic algorithms, and they are developed to get the near-optimal solution in a short and reasonable time for handling complex real-life projects. When the number of facilities is <15, these two types of methods are able to reach an optimal solution. However, when the number of facilities is more than 15, the problem becomes NP-complete. For definition of NP-complete problems, the reader may refer to Garey and Johnson [13]. As the number of facilities increases, the computational time increases exponentially by 2^n.

Since the optimal solution is not easy to obtain for large projects, researchers have tackled the construction site layout problem (CSLP) utilizing metaheuristic algorithms. There are many metaheuristics that can be used to address the problem of construction site layouts (Adrian et al. [3]).

The use of artificial neural networks was investigated by Yeh [10] to improve a predetermined site layout planning. The model minimizes a total cost function that includes the cost of constructing a facility at the assigned location on site and the cost of interacting with other facilities.

The Genetic Algorithm (GA) mimics the process of natural evolution and is routinely utilized to generate useful solutions to optimization and search problems. GA generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Numerous applications of GA are suggested for the facility site layout problems (Adrian et al. [3], Cheung et al. [14], Li and Love [15, 16], Zouein et al. [17], Mawadesley et al. [18], and Mavadesley and Al-Jibouri [19]). Li and Love [16] presented an investigation applying the Genetic Algorithm to attain the optimal solution for single-objective CSLP problem to accommodate facilities of unequal area in predetermined locations. Osman et al. [20] proposed a hybrid CAD-based algorithm using genetic algorithm in order to optimize the assignment of unequal area facilities to any unoccupied space at a construction site.

Particle swarm optimization (PSO) is another metaheuristic approach that simulates the social behavior of bird flocking to a desired place [21]. Zang and Wang [22] proposed a PSO-based methodology. They modeled the CSLP problem to optimize static layout under single-objective function to accommodate facilities of unequal area in predetermined locations. Another study relevant to the PSO was conducted by Xu and Li [23]. Their approach used a multi-objective particle swarm optimization (MOPSO) algorithm. This approach was also applied for solution of the multi-objective dynamic CSLP problem. Lien and Cheng [24] proposed a hybrid swarm intelligence-based particle-bee algorithm for construction site layout optimization with single-objective function to locate facilities in predetermined locations.

The ant colony optimization (ACO) is a biologically inspired metaheuristic that simulates the behavior of ants searching for food [25]. The ACO was employed to solve facility layout problem in a hypothetical medium-sized construction site [26]. Gharaie et al. [27] and Lam et al. [26] employed ACO to solve a static site layout problem for a construction project. Ning et al. [7] used Max–Min Ant System (MMAS), which is one of the standard versions of ACO algorithms to solve a dynamic CSLP. Though the CSLP problem has been tackled by some researchers; however, the application of new metaheuristics is always beneficial and can improve the solutions.

In this chapter, two recently developed metaheuristic algorithms, known as colliding bodies optimization (CBO) and Enhanced colliding bodies optimization (ECBO), are applied to the solution of construction site layout problems. Colliding Bodies Optimization is developed by Kaveh and Mahdavi [28], and Enhanced Colliding Bodies Optimization is presented by Kaveh and Ilchi Ghazaan [29]. CBO and ECBO are employed for solution of the CSLP problem and results are compared with those of some previous algorithms. Two case studies are conducted to evaluate the performance and applicability of the utilized algorithms. The structure of the chapter is as follows: in Sect. 18.2, the construction site layout problem is described briefly and the mathematical model is presented. In Sect. 18.3, the CBO and ECBO are described in detail. Section 18.4 shows the computational results, and finally the concluding remarks are provided in Sect. 18.6.

18.2 Construction Site Layout Planning Problem

Construction site layout planning problems can be modeled as a QAP in which costs associated with the flow between facilities are linear with respect to the distance traveled and quantity of the flow [8]. The objective of construction site layout planning is to assign a number of predetermined facilities (n) uniquely into a number of predetermined locations (m) where the number of locations should be equal or greater than number of facilities. If the number of predetermined locations (m) is greater than the number of predetermined facilities (n), then $m - n$ dummy facilities can be added to make both numbers equal. By assigning both the distance and frequency as 0, the "dummy" facilities will not affect the layout results [16].

If each of the predetermined places is capable of accommodating any of the facilities, then the facility layout problem can be modeled as an equal-area facility layout problem. If some of the predetermined places are only able to accommodate some of the facilities, then the problem becomes an unequal-area facility layout problem, where predetermined places have differing areas. Generally, unequal-area layout problems are more difficult to solve than equal-area layout problems, primarily because unequal-area layout problems introduce additional constraints into the problem formulation [16].

18.2.1 Objective Function

The objective function of several models given in Table 18.1 takes the general form (Osman et al. [20]):

$$\text{Minimize } F = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \times d_{ij} \tag{18.1}$$

where F is the objective function and n is the number of facilities and locations. Coefficient W_{ij} represents either the actual transportation cost per unit distance between facilities i and j (taking into consideration the number of trips made) or a relative proximity weight that reflects the required closeness between facilities i and j, and d_{ij} is the distances between facilities i and j.

Table 18.1 Different kind of objective functions in the previous researches (Osman et al. [20])

No.	Pseudo model of the objective function	Study reference
1	To minimize the frequency of trips made by construction personnel	Li and Love [15, 16]
2	To minimize the total transportation costs of resources between facilities	Cheung et al. [14] and Tam et al. [30]
3	To minimize the cost of facility construction and the interactive cost between facilities	Yeh [10]
4	To minimize the total transportation costs of resources between facilities (presented through a system of proximity weights associated with an exponential scale)	Hegazy and Elbeltag [31]
5	To minimize the total transportation costs of resources between facilities and the total relocation costs (presented through a system of proximity weights and relocation weights)	Zouein and Tommelein [32]

Table 18.2 An example of the permutation matrix representation for CSLP

Number of facilities	Number of locations									
	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
F1	0	1	0	0	0	0	0	0	0	0
F2	0	0	0	0	0	0	0	0	1	0
F3	1	0	0	0	0	0	0	0	0	0
F4	0	0	0	0	1	0	0	0	0	0
F5	0	0	1	0	0	0	0	0	0	0
F6	0	0	0	0	0	0	1	0	0	0
F7	0	0	0	0	0	0	0	0	0	1
F8	0	0	0	1	0	0	0	0	0	0
F9	0	0	0	0	0	1	0	0	0	0
F10	0	0	0	0	0	0	0	1	0	0

Table 18.3 An example of the sequence-based representation for CSLP

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
2	9	1	5	3	7	10	4	6	8

18.2.2 Layout Representation

Each layout alternative can be represented by a $n \times n$ permutation matrix (n is the number of facilities or locations), whose rows and columns represent facilities and locations, respectively. The permutation matrix allows a single entry of one in each row and each column, with all remaining entries being zero. Table 18.2 shows an example of a permutation matrix with 10 facilities and 10 locations.

A specific solution to the site layout problem as shown by Table 18.2 is a very sparse matrix and would therefore consume unnecessary computing resources if it is used for large and practical problems. Therefore, because of the property of one to one correspondences between facilities and locations, a sequence of integers can be used as a more efficient alternative, like that in Table 18.3. Each position or entry in the sequence represents a facility; the integer number in the entry represents the location to place the corresponding facility. However, the sequence-based representation may lead to infeasible solutions where multiple entries in the sequence have the same integer number, i.e., the situation of overlay, when adopting the metaheuristic methods. Therefore, some modifications should be made to overcome this difficulty (Li and Love [15], Mavadesley and Al-Jibouri [19], and Zhang and Wang [22]).

18.3 Metaheuristic Algorithms

In this chapter, two new metaheuristic algorithms consisting of the colliding bodies optimization and enhanced colliding bodies optimization are used for construction site layout problems (CSLP). These algorithms, which are powerful and effective in finding the best solution for NP-hard problems, are utilized for CSLP problem.

18.3.1 Colliding Bodies Optimization

Colliding Bodies Optimization (CBO) is an efficient metaheuristic optimization algorithm that is based on one-dimensional collisions between bodies [28]. All of the following explanations about this method, including definitions and formulas, are extracted from Kaveh and Mahdavi [28] and Kaveh [33].

In this method, one object collides with the other object, and they move toward a minimum energy level. Collisions between these objects are governed by two laws of physics: momentum law and energy law.

In CBO, each solution candidate X_i containing a number of variables $\left(\text{i.e.}, X_i = \left\{X_{ij}\right\}\right)$ is considered as a colliding body (CB). The objects have assigned masses and are divided into two equal groups, i.e., stationary and moving objects (Fig. 18.1), where the moving objects move to follow stationary objects and a collision occurs between pairs of objects. This takes place for two purposes: (i) to improve the locations of moving objects and (ii) to push stationary objects toward better locations. After the collision, new locations of colliding bodies are updated based on the new velocities using collision laws.

The CBO algorithm is briefly presented in the following:

Step 1: Initialization

The algorithm starts with a random initial population of agents (CBs) in an *m*-dimensional search space by the following formula:

$$x_i{}^0 = x_{\min} + random \circ (x_{\max} - x_{\min}), \qquad i = 1, 2, \ldots, n \tag{18.2}$$

where x_i^0 determines the initial value vector of the *i*th CB. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable value vectors of variables, *rand* is a random number in the interval [0, 1], and *n* is the number of CBs.

Step 2: Defining mass

Each colliding body (CB), X_i, has a specified mass defined as

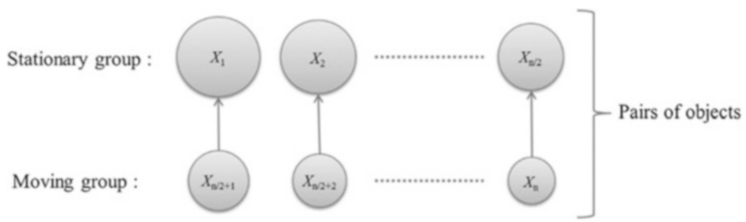


Fig. 18.1 The pairs of CBs for collision

$$m_k = \frac{\frac{1}{fit(k)}}{\frac{1}{\sum_{i=1}^{n} \frac{1}{fit(i)}}}, \quad k = 1, 2, \ldots, n \tag{18.3}$$

where $fit(i)$ represents the objective function value of the ith CB and n is the number of colliding bodies. It should be noted that larger mass values are assigned to CBs with better objective function values.

Step 3: Creating groups

Then CB's objective function values are arranged in an ascending order. The sorted CBs are divided into two equal groups:

- The lower half of the CBs are stationary CBs that have lower objective function values. These CBs are considered as good agents.
- The CBs of the upper half are moving ones. These CBs move toward the lower ones and then the agents with upper value of each group collide together.

Step 4: Criteria before the collision

The initial velocities of stationary CBs are equal to:

$$v_i = 0, \quad i = 1, 2, \ldots, \frac{n}{2} \tag{18.4}$$

The velocities of moving CBs before collision are equal to:

$$v_i = x_{i-\frac{n}{2}} - x_i, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{18.5}$$

where v_i and x_i are the velocity and location vector of the ith CB in this group, respectively, and $x_{i-\frac{n}{2}}$ is the ith CB pair location of x_i in the previous group.

Step 5: Criteria after the collision

After the collision, the velocity of stationary CBs ($v_i^{'}$) are specified by

$$v_i^{'} = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} \quad i = 1, 2, \ldots, \frac{n}{2} \tag{18.6}$$

Also, the velocities of moving CBs ($v_i^{'}$) after the collision are

$$v_i^{'} = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}} \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{18.7}$$

where ε is the coefficient of restitution (COR) that decreases linearly from unity to zero. Thus, it is expressed as

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{18.8}$$

where $iter$ and $iter_{max}$ are the current iteration number and the total number of iterations for optimization process, respectively.

Step 6: Updating CBs

New locations of the CBs are evaluated using their velocities after the collision. The new locations of stationary CBs are

$$x_i^{\mathrm{new}} = x_i + rand^{\circ} v_i' \quad i = 1, 2, \ldots, \frac{n}{2}, \tag{18.9}$$

and the new locations of moving CBs are

$$x_i^{\mathrm{new}} = x_{i-\frac{n}{2}} + rand^{\circ} v_i', \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n \tag{18.10}$$

where x_i^{new}, x_i, and v_i' are the new location, previous location, and the velocity after the collision of the ith CB, respectively. $rand$ is a random vector uniformly distributed in the range of [−1,1] and the sign "°" denotes an element-by-element multiplication.

Step 7: Termination criterion check

The process of CBO algorithm is repeated from Step 2 until a termination criterion, such as maximum iteration number, is satisfied.

Flowchart of the CBO algorithm is depicted in Fig. 18.2.

18.3.2 Enhanced Colliding Bodies Optimization

Enhanced Colliding Bodies Optimization (ECBO) is a new version of the CBO which improves the CBO to get faster and to obtain more reliable solutions. This method is developed recently by Kaveh and Ilchi Ghazaan [29]. Unlike CBO, the main feature of the ECBO is that it uses a memory to save some best solutions that cause an increase in the convergence speed of ECBO with respect to standard CBO. In order to improve the exploration capabilities of the CBO and to prevent premature convergence, ECBO utilizes a mechanism to escape from local optimal.

All of the following explanations about this method, including definitions and formulas, are extracted from Kaveh and Ilchi Ghazaan [29]. In order to introduce the ECBO, the following steps are developed:

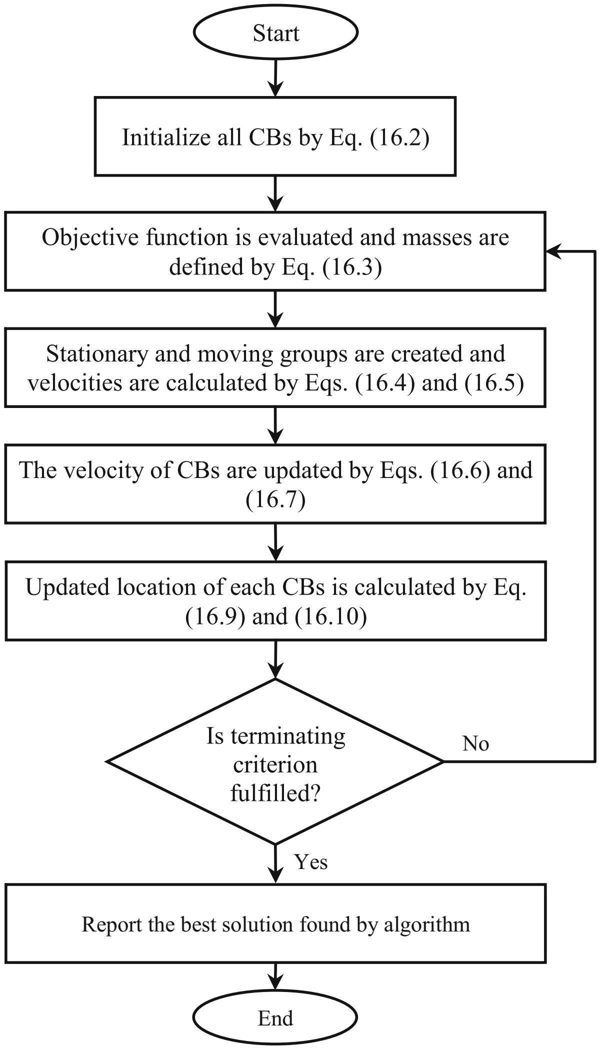


Fig. 18.2 Flowchart of the CBO algorithm [28]

Step 1: Initialization

The algorithm starts with a random initial population of agents (CBs) in an m-dimensional search space by the following formula:

$$x_i{}^0 = x_{\min} + random \circ (x_{\max} - x_{\min}), \qquad i = 1, 2, \ldots, n \tag{18.11}$$

where x_i^0 determines the initial value vector of the ith CB. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable values vectors of variables, *rand* is a random number in the interval [0, 1], and n is the number of CBs.

Step 2: Defining mass

The value of mass for each CB is evaluated according to Eq. (18.3).

Step 3: Saving

In this step, colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values with the aim of improving the algorithm's performance. At each iteration, solution vectors that are saved in the CM are added to the population and the same number of the current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: Creating groups

The CBs are divided into two equal groups according to their objective function values: stationary and moving group.

Step 5: Criteria before the collision

The velocities of the stationary and moving bodies before collision are evaluated by Eqs. (18.4) and (18.5), respectively.

Step 6: Criteria after the collision

The velocities of the stationary and moving bodies after collision are evaluated by Eqs. (18.6) and (18.7), respectively.

Step 7: Updating CBs

The new location of each CB is evaluated by Eqs. (18.8 or 18.9).

Step 8: Escape from local optimal

In order to escape from local optimal, a parameter like *Pro* within (0, 1) is introduced, which specifies whether a component of each CB must be changed or not. For each colliding body *Pro* is compared with $rn_i(i = 1, 2, \ldots, n)$ which is a random number uniformly distributed within (0, 1). If $rn_i < pro$, one dimension of the ith CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + random \cdot \left(x_{j,\max} - x_{j,\min}\right) \tag{18.12}$$

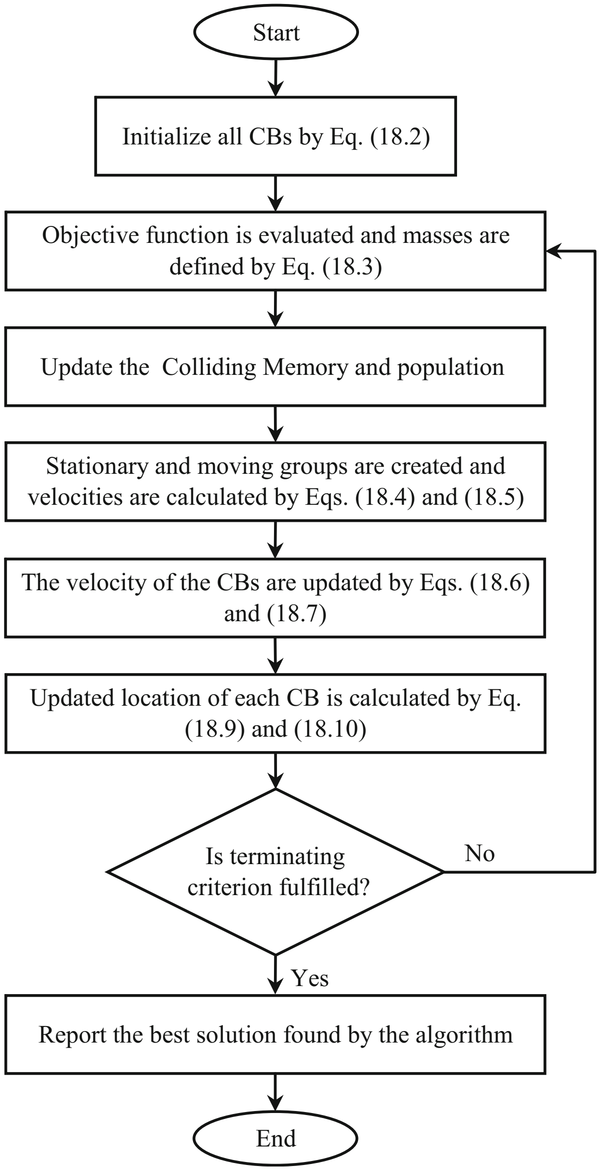
where x_{ij} is the jth variable of the ith CB. $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the jth variable, respectively. In order to protect the structure of CBs, only one dimension is changed.

Step 9: Termination criterion check

After the predefined maximum iteration number, the optimization process is terminated. If this criterion is not satisfied, go to Step 2 for a new round of iteration.

Flowchart of the ECBO algorithm is illustrated in Fig. 18.3.

Fig. 18.3 Flowchart of the ECBO algorithm [29]



18.4 Model Application and Discussion of the Results

In CBO and ECBO algorithms, each solution candidate X_i containing a number of variables $\left(\text{i.e.}, X_i = \left\{X_{ij}\right\}\right)$ is considered as a colliding body (CB). In the CSLP problems, each CB is considered as a sequence of variables that represents a layout solution and different sequences mean different layout solutions. Each variable in the sequence represents a facility, and the value of variable indicates the location that is assigned to the corresponding facility. Since every location is capable of receiving only one facility, the CBs should not have duplicated values; violation from this point generates infeasible solutions. However, all the variables of a CB in CBO and ECBO are independent of each other; thus, updating the velocity and position of a CB is performed independently. Therefore, more than one variable in an updated CB may have the same value. Thus, some modifications in updating mechanism should be performed to overcome this infeasibility. The updating mechanism of the CBs is explained in the following:

CB's Updating Mechanism

Partially mapped crossover (PMX) in GA is a mechanism to overcome infeasibility in permutation problems. In the PMX method, at each of the selective (i.e., randomly) gene, two values in the two parents' chromosome are exchanged. Then, the repeated value at another gene in one parent is replaced by the mapped value at the specified selective gene in the second parent (the former value of selective gene in first parent), and then, the same action is performed with second parent [34].

In this chapter, inspired by the concept of the PMX in generating feasible layout, an updating mechanism for generating feasible layout in the CBO and ECBO algorithms is used. In this mechanism, the velocity of stationary and moving CBs after collision is computed and considered as a criterion to decide which variable of a CB should be updated earlier. A larger velocity means there is larger gap between that variable and its goal, and it has higher tendency to be updated earlier. Therefore, the absolute value of the velocity is used herein to represent the order of variables that should be updated [34].

Every variable of a CB is selected as the current variable (CV) according to the sorted velocity of variables. The value of the current variable is updated according to its reference and using to Eqs. (18.6–18.10). The reference of moving CB is its corresponding stationary CB, and the reference of a stationary CB is itself. Then, the repeated value of another variable in this CB is substituted by the former value of the current variable. In this step, if the value of the variable that is obtained in this step has been selected before (for any of the previous current variables), updating the CV is ignored and the next variable is selected for updating until the last variable is updated. Flowchart for the updating mechanism of a CB is presented in Fig. 18.4.

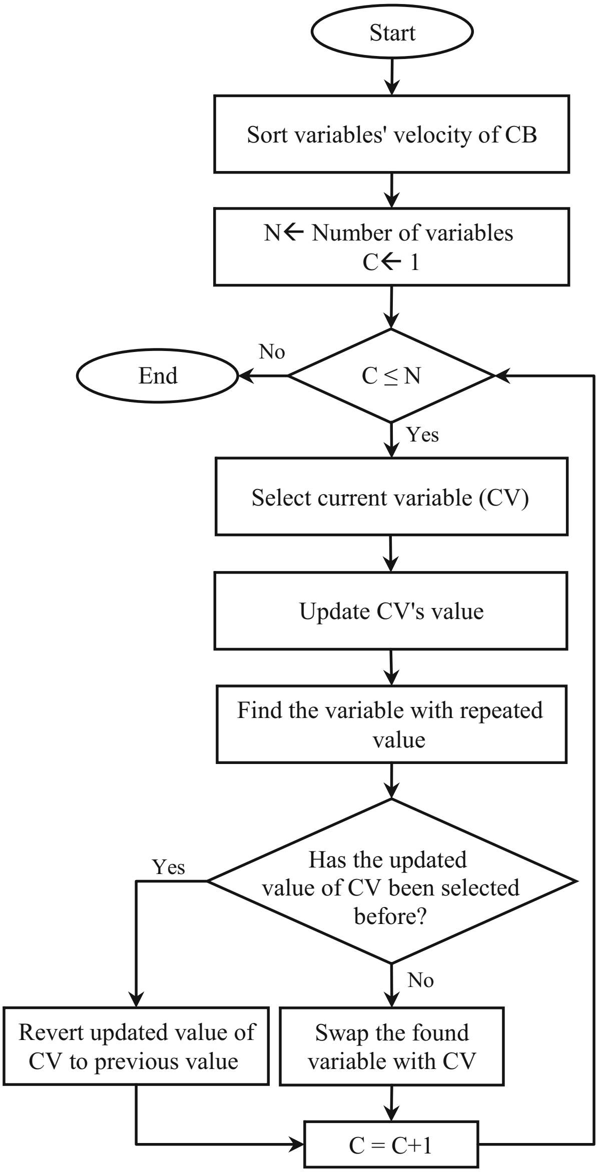


Fig. 18.4 Flowchart for the updating mechanism of a CB

18.5 Case Studies of Construction Site Layout Planning

Two case studies are selected to show the applicability and performance of the CBO and ECBO algorithms for construction site layout optimization and their results are compared to those of the PSO. Parameter values used in these case studies are shown in Table 18.4. The algorithms are coded in *MATLAB R2011a*, and the

Table 18.4 Parameter values used in case studies

PSO		CBO		ECBO	
Population size	50	Population size	50	Population size	50
Inertia weight	0.4-0.9			CM size	5
C_1=C_2	2			*pro*	0.3

experiments are performed on a personal computer with Intel®Core™ i7 processor (1.73 GHz) and 4 GB RAM under the windows 10 Home 64-bit operating system. The detailed case studies and the results are as follows:

18.5.1 Case Study 1

This case study is a medium-sized project and is taken from Li and Love [15]. The purpose of this problem is to find the most appropriate arrangement for placing 11 facilities into 11 predetermined locations on the site. Table 18.5 shows the 11 facilities and their corresponding index numbers.

In this case study, for the construction site layout selection, two assumptions are made:

1. Each of the predetermined locations is capable of accommodating any of the facilities.
2. The main gate and side gate are treated as special facilities, which have been fixed on the predetermined locations.

18.5.1.1 Objective Function

The objective of this case is to minimizing the total traveling distance of site personnel between facilities. The total travel distance is based on the formulation of Li and Love [15] as

$$\begin{aligned} &\text{Minimize} \qquad \mathrm{TD} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{k=1}^{n} x_{ik} \times x_{jl} \times f_{ij} \times d_{kl} \\ &\text{Subjected to} \quad \sum_{i=1}^{n} x_{ij} = 1 \ , \quad \sum_{j=1}^{n} x_{ij} = 1 \end{aligned} \tag{18.13}$$

where n = number of facilities. $x_{ik} = 1$ when the facility i is assigned to the location k; otherwise it is equal to 0; x_{jl} is similarly defined. Coefficient f_{ij} is the frequency of trips made by construction personnel between facilities i and j per day. Coefficient d_{kl} is the distances between the locations k and l. Therefore, TD provides the total traveling distance made by construction personnel per day.

Table 18.5 Facilities and their corresponding index numbers for case study 1

Index number	Site facilities	Note
1	Site office	Not fixed
2	False work workshop	Not fixed
3	Labor residence	Not fixed
4	Storeroom 1	Not fixed
5	Storeroom 2	Not fixed
6	Carpentry workshop	Not fixed
7	Reinforcement steel workshop	Not fixed
8	Side gate	Fixed to 1
9	Electrical, water, and other utilities control room	Not fixed
10	Concrete batch workshop	Not fixed
11	Main gate	Fixed to 10

Table 18.6 Travel distances between the predetermined locations

		Location										
Distance		1	2	3	4	5	6	7	8	9	10	11
Location	1	0	15	25	33	40	42	47	55	35	30	20
	2	15	0	10	18	25	27	32	42	50	45	35
	3	25	10	0	8	15	17	22	32	52	55	45
	4	33	18	8	0	7	9	14	24	44	49	53
	5	40	25	15	7	0	2	7	17	37	42	52
	6	42	27	17	9	2	0	5	15	35	40	50
	7	47	32	22	14	7	5	0	10	30	35	40
	8	55	42	32	24	17	15	10	0	20	25	35
	9	35	50	52	42	37	35	30	20	0	5	15
	10	30	45	55	49	42	40	35	25	5	0	10
	11	20	35	45	53	52	50	40	35	15	10	0

18.5.1.2 Travel Distances Between Site Locations

The travel distances between predetermined locations are provided in Table 18.6 (Li and Love [15]).

18.5.1.3 Trip Frequencies Between Facilities

Trip frequencies between facilities influence the site layout planning and proximity between predetermined site facilities. Therefore, the frequencies of the trips made between facilities on a single day are presented in Table 18.7 (Li and Love [15]).

Table 18.7 Trip frequencies between the facilities

Trip frequency		Facility										
		1	2	3	4	5	6	7	8	9	10	11
Facility	1	0	5	2	2	1	1	4	1	2	9	1
	2	5	0	2	5	1	2	7	8	2	3	8
	3	2	2	0	7	4	4	9	4	5	6	5
	4	2	5	7	0	8	7	8	1	8	5	1
	5	1	1	4	8	0	3	4	1	3	3	6
	6	1	2	4	7	3	0	5	8	4	7	5
	7	4	7	9	8	4	5	0	7	6	3	2
	8	1	8	4	1	1	8	7	0	9	4	8
	9	2	2	5	8	3	4	6	9	0	5	3
	10	9	3	6	5	3	7	3	4	5	0	5
	11	1	8	5	1	6	5	2	8	3	5	0

Table 18.8 Comparison of the results of 50 independent runs for the first case example

Algorithm	Best	Average	Worst	Difference best–average solution%	Difference best–worst solution%	STD
PSO	12,546	12,560	12,756	0.112	1.647	47.39
CBO	12,546	12,558	12,768	0.096	1.769	45.51
ECBO	12,546	12,555	12,746	0.072	1.594	32.11

18.5.1.4 Result and Discussion

This example is solved by carrying out 50 independent optimization runs through 200 iterations to obtain statistically significant results by PSO, CBO, and ECBO. Statistical results of 50 independent runs are provided in Table 18.8 for comparison. As it can be seen from this table, the average, worst, and standard deviation for ECBO are 12,555, 12,746, and 32.11, respectively, which are better than those of CBO and PSO. This indicates that ECBO not only finds a better best solution, but also it is more stable. The convergence curves for the ECBO, CBO, and PSO in terms of the number of iterations are shown in Fig. 18.5. A comparison of the results of the present algorithms and those of the previously reported researches for Case 1 is shown in Table 18.9. The results show that for this case study, the best result is 12,546 which is better than that of the GA and it is the same as that of the ACO.

18.5.2 Case Study 2

In the optimization of construction-site precast yard layout, the efficiency of a site precast yard is very much affected by positioning of the various facilities [2]. The hypothetical site precast yard in this section is taken from Cheung et al. [14]. There

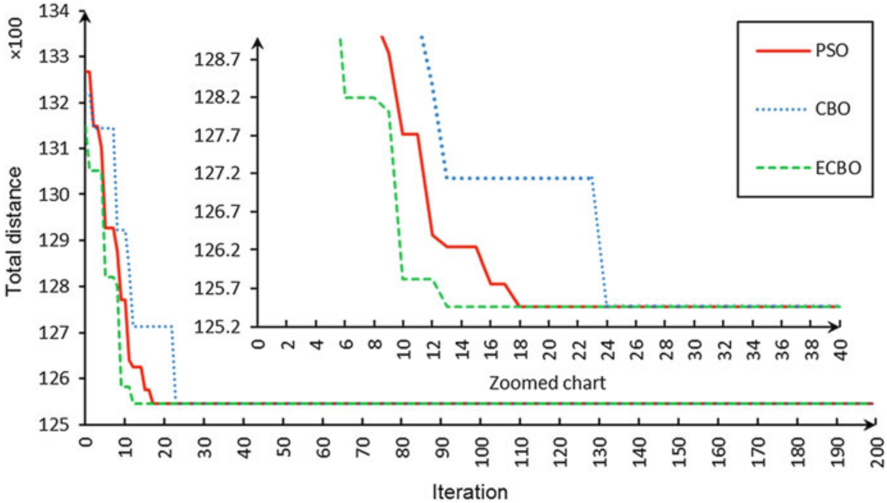


Fig. 18.5 Convergence curves of the utilized metaheuristics

Table 18.9 A comparison between the final solution of the present work and those of the previously reported researches

Algorithms	Total distance	Best layout										
		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}
PSO[a]	12,546	9	11	5	6	7	4	3	1	2	8	10
CBO[a]	12,546	9	11	6	5	7	4	3	1	2	8	10
ECBO[a]	12,546	9	11	4	5	7	6	3	1	2	8	10
GA (Li and Love [15])	15,090	11	5	8	7	2	9	3	1	6	4	10
ACO (Gharaie et al. [27])	12,546	9	11	6	5	7	2	4	1	3	8	10

[a]Current study

are 11 facilities that should be assigned to 11 predetermined locations on the yard. The facilities and their corresponding index numbers are listed in Table 18.10. Four types of resources and transport costs per unit distance are also presented in Table 18.11.

18.5.2.1 Objective Function

The objective function is considered as the total cost per day for transporting all resources necessary to achieve the anticipated output. The objective function based on Cheung et al. [14] is calculated as follows:

Table 18.10 Facilities and the corresponding index numbers

Index number	Site facilities
1	Main gate
2	Side gate
3	Batching plant
4	Steel storage yard
5	Formwork storage yard
6	Bending yard
7	Cement and sand and aggregate storage yard
8	Curing yard
9	Refuse dumping area
10	Casting yard
11	Lifting yard

Table 18.11 Four types of resources and transport costs per unit distance

Mk	Resources	Cost per Unit
1	Aggregate, sand, and cement/concrete	5
2	Reinforcement bars	4
3	Formwork	8
4	Completed precast units	8.5

$$\begin{aligned} &\text{Minimize} \qquad \text{TC} = \sum_{k=1}^{n}\sum_{i=1}^{q}\sum_{j=1}^{q} \text{TC}L_{Mk,i,j} \\ &\text{TC}L_{Mk,i,j} = M_{LMij} \times C_{Mk} \\ &M_{LMij} = FL_{Mkij} \times D_{ij} \end{aligned} \tag{18.14}$$

where

D_{ij} = rectangular distance between the location i and location j.
C_{Mk} = cost per unit distance for resource Mk flow.
$\text{TC}L_{Mk,i,j}$ = total cost of resource Mk flow between the locations i and j.
$ML_{Mki,j}$ = distance traveled of resource Mk flow per unit time between locations i and location j.
$FL_{Mk,i,j}$ = frequency of resource Mk flow between the locations i and j per unit time.

18.5.2.2 Travel Distance Between Site Precast Yard Locations

The rectangular distance between locations is measured and presented in Table 18.12.

Table 18.12 Distance between locations in case study 2

Distance	Location											
		1	2	3	4	5	6	7	8	9	10	11
Location	1	0	12	17	30	35	33	55	53	38	30	19
	2	12	0	9	22	27	21	47	45	40	18	31
	3	17	9	0	13	22	30	38	36	31	27	22
	4	30	22	13	0	15	23	25	23	38	20	29
	5	35	27	22	15	0	8	20	38	53	25	44
	6	33	21	30	23	8	0	28	46	61	17	52
	7	55	47	38	25	20	28	0	18	33	45	40
	8	53	45	36	23	38	46	18	0	15	43	38
	9	38	40	31	38	53	61	33	15	0	58	23
	10	30	18	27	20	25	17	45	43	58	0	49
	11	19	31	22	29	44	52	40	38	23	49	0

18.5.2.3 Frequency of Resources Flow Between Facilities

The flow frequency of the four types of resources between the facilities is presented in Table 18.13.

18.5.2.4 Result and Discussion

This example was solved by carrying out 30 independent optimization runs through 1000 iterations to obtain statistically significant results by PSO, CBO, and ECBO. Statistical results of 30 independent runs are compared in Table 18.14. As it can be seen from Table 18.14, the average, worst, and standard deviation for ECBO are, respectively, 92,758, 102,920, and 2733.5, which are better than those of CBO and PSO. This indicates that ECBO not only finds a better best solution but also is more stable. The convergence curves for the ECBO, CBO, and PSO in terms of the number of iterations are shown in Fig. 18.6, indicating that ECBO has better convergence rate than others. Table 18.15 summarizes the results obtained by the present work and those of the previously reported researches. In this case study, the best result is 92,758 which is better than that of GA, multi-searching TS, and MIP, and it is the same as that of the Harmony search.

18.6 Concluding Remarks

In this chapter, the application of two recently developed metaheuristic algorithms, CBO and ECBO, is introduced to solve construction site layout problem. The governing laws of physics initiate the base of the CBO and ECBO algorithms, where these laws determine the movement process of the objects. CBO utilizes

Table 18.13 The flow frequency of the four types of resources between the facilities

Flow frequency		Facility 1	2	3	4	5	6	7	8	9	10	11
1. Aggregate, sand, and cement												
Facility	1							20				
	2							15				
	3							35			35	
	4											
	5											
	6											
	7	20	15	35								
	8											
	9											
	10			35								
	11											
2. Reinforcement												
Facility	1				30							
	2				20							
	3											
	4	30	20				50					
	5											
	6				50						50	
	7											
	8											
	9											
	10						50					
	11											
3. Formwork												
Facility	1											
	2											
	3											
	4											
	5										48	
	6											
	7											
	8											
	9											
	10					48						
	11											
4. Completed precast units												
	1											28
	2											20
	3											
	4											

(continued)

Table 18.13 (continued)

Flow frequency		Facility 1	2	3	4	5	6	7	8	9	10	11
	5											
	6											
	7											
	8										48	48
	9											
	10								48			
	11	28	20						48			

Table 18.14 Comparing of the results of 30 independent runs for second case example

Algorithm	Best	Average	Worst	Difference best–average solution%	Difference best–worst solution%	STD
PSO	92,758	97,667	106,630	5.292	14.955	3363.1
CBO	92,758	97,504	103,038	5.117	11.083	3149
ECBO	92,758	96,670	102,920	4.217	10.955	2733.5

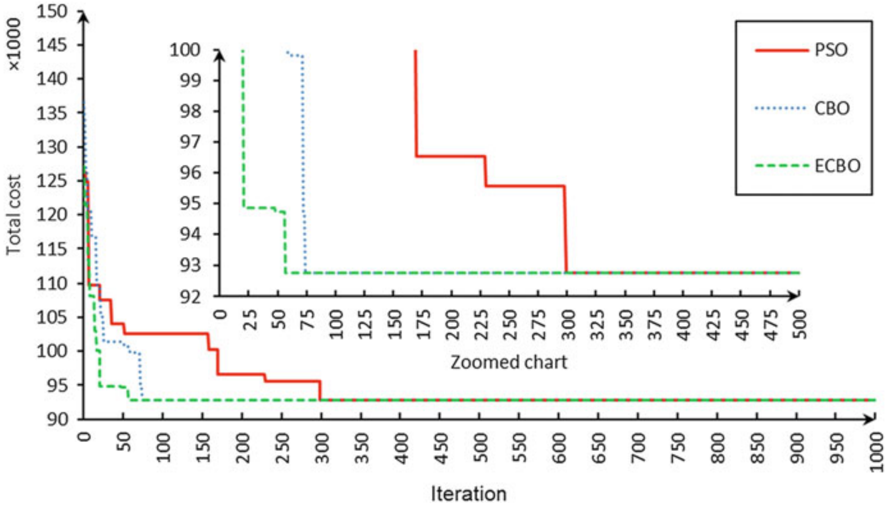


Fig. 18.6 Convergence curves of the employed metaheuristics

simple formulation to find minimum of objective functions and does not depend on any internal parameter. In order to improve the exploration capabilities of the CBO and to prevent a premature convergence, ECBO uses a mechanism to escape from local optimal. The latter also uses a Colliding Memory to save a number of the so far best solutions to reduce the computational cost. To validate the models, two case studies are considered. The results verify that the proposed approach performs very

Table 18.15 A comparison between the final solution of the present work and those of the previously reported researches

Algorithms	Total cost	Best layout										
		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}
PSO[a]	92,758	5	7	9	6	1	10	8	3	11	2	4
CBO[a]	92,758	5	7	9	6	1	10	8	3	11	2	4
ECBO[a]	92,758	5	7	9	6	1	10	8	3	11	2	4
GA (Cheung et al. [14])	99,788	1	10	9	6	8	5	11	3	7	4	2
Multi-searching TS (Liang and Chao [35])	94,858	5	7	10	8	1	9	6	3	11	2	4
Harmony search (Kaveh [33])	92,758	5	7	9	6	1	10	8	3	11	2	4
MIP (Wong et al. [2])	98,424	1	10	8	6	7	5	9	3	11	4	2

[a]Current study

well both in finding better results and using lower number of evaluations to find the optimum. Comparison of the results with some other well-known metaheuristics shows the suitability and efficiency of the utilized algorithms in CSLP. The proposed algorithms are highly competitive with other metaheuristic algorithms in quality of solutions and convergence speed.

References

1. Kaveh A, Khanzadi M, Alipour M, Rastegar Moghaddam M (2016) Application of two new meta-heuristic algorithms for construction site layout planning problem. Iran J Sci Technol 40(4):263–275
2. Wong CK, Fung IWH, Tam CM (2010) Comparison of using mixed-integer programming and genetic algorithms for construction site facility layout planning. J Constr Eng Manag 136 (10):1116–1128
3. Adrian AM, Utamima A, Wang KJ (2014) A comparative study of GA, PSO and ACO for solving construction site layout optimization. KSCE J Civil Eng 19(3):520–527
4. Lam KC, Tang CM, Lee WC (2005) Application of the entropy technique and genetic algorithms to construction site layout planning of medium-size projects. Constr Manag Econ 23(2):127–145
5. Yeh IC (2006) Architectural layout optimization using annealed neural network. Autom Constr 15(4):531–539
6. Said H, El-Rayes K (2013) Performance of global optimization models for dynamic site layout planning of construction projects. Autom Constr 36:71–78
7. Ning X, Lam KC, Lam MCK (2010) Dynamic construction site layout planning using max–min ant system. Autom Constr 19(1):55–65
8. Tate DM, Smith AE (1995) Unequal-area facility layout by genetic search. IIE Trans 27 (4):465–472
9. Azarbonyad H, Babazadeh R (2012) A genetic algorithm for solving quadratic assignment problem (QAP). In Proceeding of 5th International Conference of Iranian Operations Research Society (ICIORS), Tabriz, Iran, pp 2–5

10. Yeh IC (1995) Construction-site layout using annealed neural network. J Comput Civil Eng 9 (July):201–208
11. Sanad HM, Ammar MA, Ibrahim ME (2008) Optimal construction site layout considering safety and environmental aspects. J Constr Eng Manag 134(7):536–544
12. Tommelein ID, Levitt RE, Hayes-Roth B (1992) Site-layout modeling: how can artificial intelligence help? J Constr Eng Manag ASCE 118(3):594–611
13. Garey MR, Johnson DS (1979) A guide to the theory of NP-completeness. A series of books in the mathematical sciences. W. H. Freeman, New York
14. Cheung SO, Tong TKL, Tam CM (2002) Site pre-cast yard layout arrangement through genetic algorithms. Autom Constr 11:35–46
15. Li H, Love PED (1998) Site-level facilities layout using genetic algorithms. J Comput Civil Eng 12(October):227–231
16. Li H, Love EDP (2000) Genetic search for solving construction site-level unequal-area facility layout problems. Autom Constr 9(2):217–226
17. Zouein PP, Harmanani H, Hajar A (2002) Genetic algorithm for solving site layout problem with unequal-size and constrained facilities. J Comput Civil Eng 16(2):143
18. Mavadesley MJ, Al-Jibouri SH, Yang H (2002) Genetic algorithms for construction site layout in project planning. J Constr Eng Manag 128(October):418–426
19. Mavadesley MJ, Al-Jibouri SH (2003) Proposed genetic algorithms for construction site layout. Eng Appl Artif Intell 16(5–6):501–509
20. Osman HM, Georgy ME, Ibrahim ME (2003) A hybrid CAD-based construction site layout planning system using genetic algorithms. Autom Constr 12(6):749–764
21. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan
22. Zhang H, Wang JY (2008) Particle swarm optimization for construction site unequal-area layout. J Constr Eng Manag 134(9):739–748
23. Xu J, Li Z (2012) Multi-objective dynamic construction site layout planning in fuzzy random environment. Autom Constr 27:155–169
24. Lien LC, Cheng MY (2012) A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. Expert Syst Appl 39(10):9642–9650
25. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26:29–41
26. Lam K, Ning X, Ng T (2007) The application of the ant colony optimization algorithm to the construction site layout planning problem. Constr Manag Econ 25(4):359–374
27. Gharaie E, Afshar A, Jalali M (2006) Site layout optimization with ACO algorithm. In: Proceedings of the 5th WSEAS international conference on artificial intelligence, pp 90–94
28. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27
29. Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw 77:66–75
30. Tam CM, Tong KL, Chan Wilson KW (2001) Genetic algorithm for optimizing supply locations around tower crane. Constr Eng Manag 127(4):315–321
31. Hegazy T, Elbeltagl E (1999) EvoSite: evolution-based model for site layout planning. J Comput Civil Eng 13(3):198–206
32. Zouein PP, Tommelein ID (1999) Dynamic layout planning using a hybrid incremental solution method. J Constr Eng Manag 5(January):1–16
33. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer International Publishing, Switzerland
34. Zhang H, Li H, Tam CM (2006) Permutation-based particle swarm optimization for resource-constrained project scheduling. Delay 24(1):83–92
35. Liang LY, Chao WC (2008) The strategies of tabu search technique for facility layout optimization. Autom Constr 17(6):657–669