Operations Research/Computer Science Interfaces Series

Jörn Schönberger

Model-Based Control of Logistics Processes in Volatile Environments

Decision Support for Operations Planning in Supply Consortia

Springer

Operations Research/Computer Science Interfaces Series

Volume 50

For further volumes:
http://www.springer.com/series/6375

Jörn Schönberger

Model-Based Control of Logistics Processes in Volatile Environments

Decision Support for Operations Planning in Supply Consortia

Springer

Jörn Schönberger
Chair of Logistics
University of Bremen
Wilhelm-Herbst-Str. 5
28359 Bremen
Germany
jsb@uni-bremen.de

ISSN 1387-666X
ISBN 978-1-4419-9681-7 e-ISBN 978-1-4419-9682-4
DOI 10.1007/978-1-4419-9682-4
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011928086

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The overarching goal of this study is to report on the development of new and innovative ideas to support the integration of decision making between a coordinator in a supply consortium (the principal) and a subordinate service providing partner (the subordinate agent). In particular, new approaches for the management of logistics processes in volatile and uncertain planning environments are developed and evaluated. These approaches improve the common decision making between a supply consortium coordinator and service proving partner(s), which finally leads to an increase of the quality of the generated value creation process provided by the supply consortium.

An installation and a setup of opportunities for the coordinator to intervene into the local resource deployment of a service providing partner is the core idea behind these new approaches. If necessary from the viewpoint of the complete supply consortium, the coordinator should be allowed (and obliged) to control temporarily the dispatching of resources partly or completely. The motivation behind this intervention is to protect and stabilize the performance of the overall supply consortium if a certain partner does not act in the sense of the consortium. To realize and to implement the coordinator interventions, we propose that the superior coordinator manipulates the formal deployment decision model of the subordinate service providing agent. Critical changes of the decision situation (demand peaks, resource unavailability, ...) are reflected back into the formalized representation (decision model) of the subordinate agent's decision task.

Answers to the following research questions are sought in this study.

1. What are the state-of-the-art techniques to integrate decision making of principals and service agents in a supply consortium?
2. In which situations can these techniques be applied successfully and in which situations do they fail?
3. Why do the so far discussed decision supporting approaches fail?
4. How can their deficiencies be remedied?
5. How can the improved decision support approaches contribute to handling more dynamically appearing disturbances?

6. Can the stability of once made decisions be increased?
7. How high is the "price" for the extension of the abilities of the integration techniques?

We have defined three milestones, each labeling a significant progress in the reported study. The first milestone is the identification of limits and barriers of repeated decision making (online decision making) in volatile and uncertain environments in presence of principal-agent-relationships. By means of an exemplarily analyzed interaction of a supply consortium coordinator (principal) and a transport service agent, the conceptual limitations of the use of model-based decision support for principal-agent interactions are explored (Part I). In Chapter 2, the determination problems and the subsequent update problems of freight transport processes in volatile and uncertain planning environments are addressed. The ability to cope with spontaneous and unexpectedly appearing demand peaks is identified as a core challenge. We introduce a corresponding version of the online vehicle routing problem with time windows in order to prepare a process management simulation. Potentially conflicting planning objectives of the supply consortium coordinator and of the transport organizing fleet manager are identified and analyzed. State-of-the-art concepts for integrating the planning objective and decision making of a superior coordinator and a transport fleet managing agent are presented in Chapter 3. Within computational experiments these concepts are evaluated and their limitations are revealed.

The second milestone comprises the deficiency analysis of recent integrated decision making techniques and the derivation of a conceptual framework for remedying the identified shortcomings (Part II). Chapter 4 addresses the extension of adaptive process control systems by installing coordinator intervention features. We propose a close-control-circuit to alter the agent's formal process decision model, e.g., a mathematical optimization model. The necessary modifications of the objective function and of the constraint set are feedback-driven. The feedback of a process is determined by comparing the current process quality and a given reference process quality. In Chapter 5, we derive three model-adjusting strategies for the aforementioned online vehicle routing problem with time windows. Within comprehensive computational simulation experiments, we identify the best possible parameterization of the strategies. Furthermore, we compare these three strategies among themselves and in addition with the state-of-the-art techniques describes in the third chapter. It turns out that the proposed techniques are able to remedy the deficiencies of the so far known techniques.

The third and final milestone consists in the analysis of the impacts of applying the new techniques to dynamic transport process planning problems in volatile environments (Part III). Flexibility of transport systems controlled by the integrated strategies is investigated in Chapter 6. We interpret flexibility as the property to handle unexpected disturbances of processes so that an update of existing processes is possible. In Chapter 7, we try to protect once made decisions from further revisions in order to reduce the process nervousness. It is shown that the application of an integrated decision making between a coordinator and a transport service providing agent contributes to prevent further decision revisions, so that once accepted decisions exhibit an increased stability. Finally, we investigate the tradeoff between process quality increase (as a result of the application of the integrated planning approaches) and the additional expenditures resulting from overruling the typically cost-minimal deployment decisions of a service agent (Chapter 8).

We conclude this study with a summary of the main findings (Chapter 9).

This book reports scientific results from the research project *Autonomous Adaptation of Vehicle Schedules*. The reported research was carried out by the group of the Chair of Logistics, University of Bremen.

Prof. Dr.-Ing. Herbert Kopfer, holder of the Chair of Logistics and principal investigator in this project, gave me the opportunity to design and to conduct my project-related research as freely as possible. He supported me continuously during the six years of project work.

The present research stands to benefit from continuous scientific discussions. In this context I would like to express my gratitude to former and current colleagues at the Chair of Logistics, to the associates from the Collaborative Research Center 637 at the University of Bremen and to the colleagues who provided my with new ideas when we met at conferences and workshops. In particular, I have to thank Prof. Dr. Hans-Dietrich Haasis (Institute of Shipping Economics and Logistics), Prof. Dr. Christian Bierwirth (Martin-Luther-Universität Halle-Wittenberg), Prof. Dr. Martin G. Möhrle (University of Bremen) and Prof. Dr. Thorsten Poddig (University of Bremen).

This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Center 637 "Autonomous Cooperating Logistic Processes" (Subproject B7).

Bremen, *Jörn Schönberger*
February 2011

Contents

Chapter 1
Process Planning in Supply Consortia

Nowadays, value creation is mainly based on a strict division of labor. Specialists providing particular services act on the market instead of companies offering complete value creation chains (Stadtler, 2002). This is mainly caused by (i) a drastic reduction of the width of market opportunity windows (Stock et al., 2000); (ii) the pressure to continuously re-improve products and services (Ballou et al., 2000) and (iii) the requirement to gain increasing profit (Lambert and Cooper, 2000). The specialization of a company makes it possible to capture the synergy of intra- and inter-company integration and management (Lambert and Cooper, 2000), and especially to realize economies of scale (Corsten, 2007). Thus, the fulfillment of customer demand with full service at competitive prices calls for a temporal collaboration among the specialists in order to organize the required value creation processes. Typically, the collaborating partners maintain their legal and economic autonomy. Contracts are agreed between the partners in which their contributions to the current value creation project are described. Thommen (1991) calls a group of independent partners a *consortium* if - and only if - the collaboration is established to facilitate the realization of a clearly defined project. In the literature, consortia for managing a value creation chain are called supply networks (Sundermeyer, 2001) or supply chain networks (Wathne and Heide, 2004). In order to avoid misunderstanding with respect to the term "network", we use the term *supply consortium* as synonym for a formation of partners running a value creation project. Basic decision tasks for coordinating the value creation in supply consortia and for the necessary demand fulfillment are surveyed in Section 1.1.

Compared to more traditional value creation systems, the coordination and planning of value creation activities in supply consortia come along with several additional challenges. A major driver of the need for enhanced decision support is the fact that the involved partners want to maintain their own responsibilities, i.e. they want to decide autonomously about the activities that form their contribution to the coalition (Bloos et al., 2009; Villa, 2002). However, each partner has to take into account the wishes, desires and requirements of other supply consortium members as well. We review different paradigms for the coordination of decision making in supply consortia in Section 1.2.

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*,
Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_1,

Each consortium partner contributes its knowledge and experience to the management of the value creation but also provides capacities of its resources. Typically, plants, warehouses, transshipment or vending facilities of different partners are involved in the physical flow of a specific product through the value creation stages. In order to bridge the spatial distances between subsequent locations in the process of guiding products physically through the supply stages, extensive transportation of raw-material, semi-finished and finished goods is performed (Fleischmann, 2002). Specific challenges of transportation planning in supply consortia are presented in Section 1.3.

1.1 Value Creation in Supply Consortia

Stadtler (2005, 2002) proposes the so-called "house of supply chain management" that integrates the necessary building blocks for a successful supply consortium formation and for an efficient coordination of corresponding activities. According to this framework, the integration of the supply consortium partners starts with the selection of adequate partners. Next, a suitable organization of the consortium must be found. Finally, the leadership of the partnership is to be defined. A successful coordination of the activities of the partners requires the usage of latest information and communication technology (ICT). It is necessary that each partner is informed quickly and in a reliable fashion about the current and expected demand, the current inventory situation and about the availability of supply materials. Furthermore, a process orientation of the consortium members must be achieved in order to realize maximal cooperation gains. Finally, the application of advanced planning is mandatory in order to enable an integrated planning of the entire supply consortium where the most appropriate decision alternatives are identified. In addition, the desires and wishes of each single partner have to be respected to the greatest possible extent (Fleischmann et al., 2002).

Table 1.1 Acting members in a supply consortium scenario

protagonist	received information	generated information
customer	—	specified demand
coordinator	specified demand	supply consortium orders
service providing agent	specific supply consortium orders	resource requests
resource agent	specific resource requests	—

The participants in supply consortium-based value creation scenario are presented in Table 1.1. Several customers communicate their demand for products offered by the supply consortium to a consortium coordinator. The coordinator is a leading entity in the consortium and sets up an instance of an adequate customer demand fulfillment process. Such a process determines how activities interact in order to contribute to the fulfillment of demand. As parts of a process, sequences of

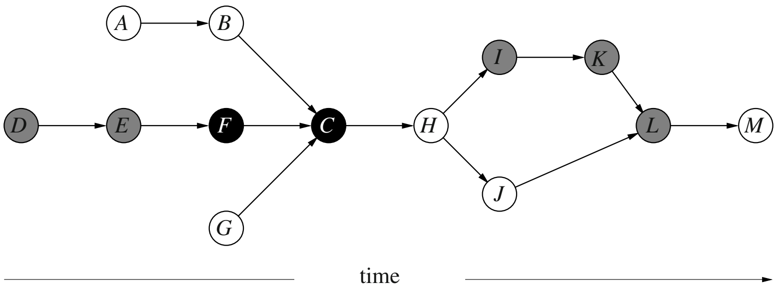

Fig. 1.1 Network capacity disposition

concatenated and interdependent activities must be fulfilled. Fig. 1.1 illustrates an example of a process with 13 activities A, B, ..., M. The arcs in the network structure represent logical precedence constraints. An activity can only be started once its predecessor(s) is (are) finished. Some activities (represented by the nodes in the network structure) can be started simultaneously (A, D and G) while others require the completion of several upstream activities (C) or trigger more than one downstream activity (H).

Several decisions must be made in order to enable the execution of a process instance by the supply consortium. Each of the necessary decisions falls into one of the three following categories of decision tasks.

Network Capacity Disposition: The customer demand is split into several supply consortium *orders*. Such an order comprises the execution of one or more activities in the selected process. The supply consortium orders are then distributed among the consortium partners, the so-called service agencies, and each agency is responsible for the correct and timely fulfillment of the received orders. In the example process depicted in Fig. 1.1 the activities are distributed to supply consortium partners according to the color of the activities. All white activities are assigned to a certain consortium member, all gray activities to another service providing agency, and all black activities to a third agency.

Service Agency Resource Dispatching: A service providing agency receives supply consortium orders for fulfillment. In the example introduced in Fig. 1.1, the "black" agency is responsible for the fulfillment of the supply consortium orders C and F. Each order comprises several indivisible tasks to be fulfilled in order to contribute to the fulfillment of the corresponding supply consortium order. In our example, the order C comprises the three tasks $T_{C,1}$, $T_{C,2}$ and $T_{C,3}$. The order F consists of the three tasks $T_{F,1}$, $T_{F,2}$ and $T_{F,3}$ as shown in Fig. 1.2. Tasks from different orders are re-grouped and compiled in *requests*. Such a request describes a set of tasks that can be fulfilled by a certain resource belonging to the service agency. Four requests $r_1 := \{T_{C,1}\}$, $r_2 := \{T_{C,2}, T_{F,1}\}$, $r_3 := \{T_{C,3}, T_{F,2}\}$ and $r_4 := \{T_{F,3}\}$ are derived from the supply consortium orders C and F and have to be completed by the corresponding agency. For each of the generated requests an appropriate resource belonging to the service providing agency is selected and the request is then forwarded to the

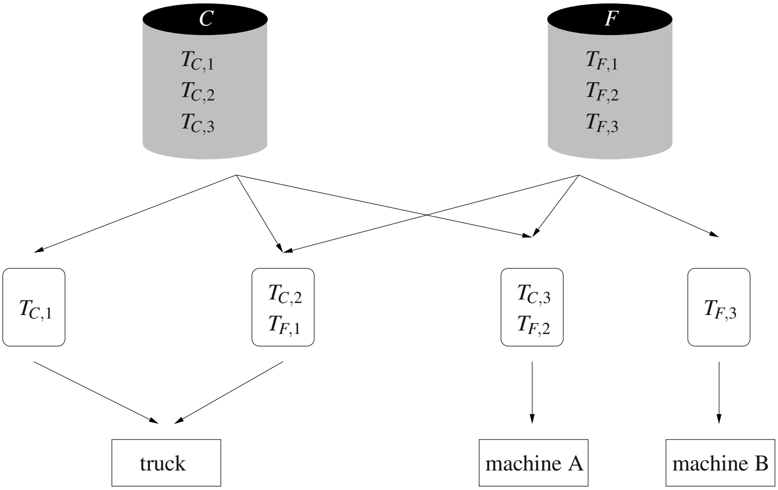

Fig. 1.2 Network Agency Resource Dispatching

selected resource. This resource is used to fulfill the forwarded request(s). The two requests r_1 and r_2 are transportation requests and are both assigned to the truck, which is a transport resource. Both remaining requests contain production tasks, so they are assigned to the two machines, which are production resources.

Resource Deployment: Using availability information about the service agency resources, it is decided how the available resources are deployed in order to fulfill the received requests. In transportation logistics, routes have to be compiled for the trucks and vehicles (Schönberger, 2005) but in production logistics, machine schedules are compiled that determine the sequence and duration of jobs on a particular machine or on a shop floor (Pinedo and Chao, 1999).

The provision of resource capacity by the service agencies in response to a coordinator's call for resources is regulated in the contracts agreed for a fixed period between the partners of the consortium. Incentives to be given to the service agencies for covering the service agencies expenditures are also fixed in these contracts.

Conflicts and mistrust between the consortium partners occur from information asymmetry (Ballou et al., 2000) and endanger the efficiency of the supply network's business operations. The principal-agent-theory (Elschen, 1991) attempts to explain the impacts of these disturbances in the interaction between superior consortium partners (the principals) and subordinate partners (the agents). Kaluza et al. (2003) apply the principal-agent-theory to coordination and adjustment problems in supply consortium scenarios. They point out two major sources of information asymmetry:

- A principal does not know how the agents will react after they have been instructed to fulfill a certain order (*hidden action*).
- A principal is not informed about all objectives of the agents (*hidden intention*).

We have two principal-agent interfaces in a supply consortium. At first, a coordinator acts as a principal towards the service agencies who act as subordinate agents ensuring the fulfillment of the supply consortium orders. Here, the information asymmetry is caused by incomplete knowledge about the customer demand. Secondly, a service agency plays the role of the principal towards the agents representing its resources. In both principal-agent relations, mistrust and conflicts prevent the identification and realization of common decisions that lead to so-called Pareto-optima, representing those disposition or dispatching decisions that provide non-dominated solutions to the benefit of both the principal and the agents and, hence, of the supply consortium. Thus, conflicts caused by hidden action and hidden intention must be recognized and even accomplished in order to preserve a well-organized and efficiently acting supply consortium.

1.2 Supply Consortia Resource Deployment Paradigms

As a consequence of the initially mentioned market-related challenges, firms have established coalitions with trusted partners in order to maintain and even increase their competitiveness. To reach this goal it is necessary to achieve a successful integration of the partners as well as to coordinate the acting of the partners. A clear distribution of responsibilities among the coalition partners is required in order to make clear process decisions in all three areas (network capacity disposition, service agency resource dispatching and resource deployment).

The number of participants in a supply consortium has been increased continuously since the end of the 1990s up to more than 1000 network partners (Sundermeyer, 2001). However, a phase of consolidation is currently observed in major industrial sectors like the automotive sector (Wallentowitz et al., 2009) or the aviation industries (Rudzki et al., 2005). A drastic reduction of procurement and distribution partners takes place. At the end of the consolidation, only few but highly integrated and interrelated partners will form a typical supply consortium. In order to meet the service quality requirements of customers, more detailed information specifying their desires has to be considered during the deployment of the coalition resources. On the other hand, the available resources are typically scarce, so that bottleneck situations must be prevented by a careful resource deployment. As a result, the complexity of supply consortium operations planning is increasing continuously. Computer-support for the preparation of the fulfillment of customer demand is mandatory and necessary.

Although computational planning systems become more elaborated due to the increasing processing speed, there are some specific and as yet unsolved challenges associated with deployment planning in consortia. Each automatic planning system has to cope with (Sundermeyer, 2001) the effects arising from this type of value creating system. Effects related to decisions occur with longer time lags (a-synchronism of decision and effect). Effects of local decisions occur at places far away from the point of decision making and are not visible at other decision points

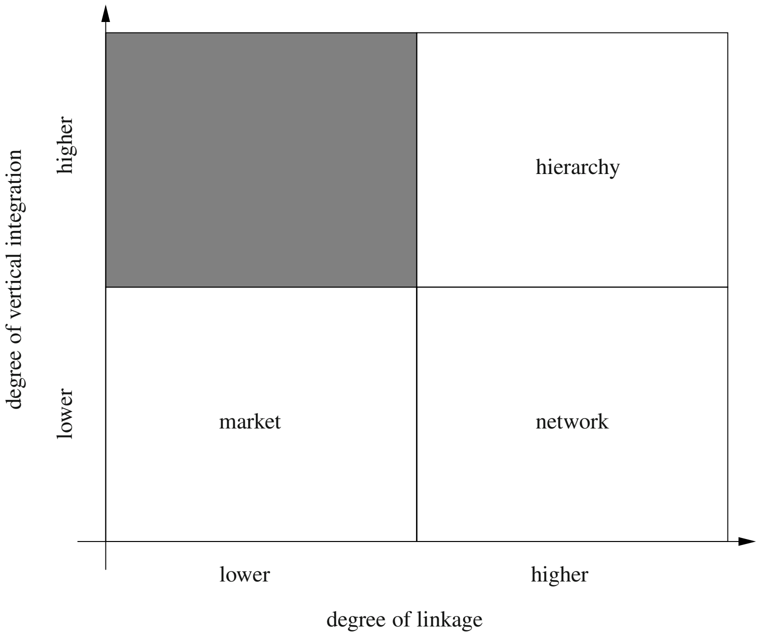

Fig. 1.3 Configuration of supply consortium governance (Stock et al., 2000)

(non-locality). The interaction effect of the many parameters in the supply consortium on common objectives is not predictable (non-linearity). (Sadler, 2007) remarks that the integration of the different supply consortium partners requires special efforts because conflicting planning objectives must be integrated. Furthermore, multi-facetted planning requirements have to be considered (non-homogeneity). Finally, (Gupta and Maranas, 2003) point out that planning data are often uncertain or even unknown, so that decisions are made which often require a later revision (uncertainty).

The organization of a supply consortium plays an important role in the control of the deployment of the resources provided by the coalition partners. Stock et al. (2000) list three generic concepts for the definition of a leadership and control strategy of a supply consortium. These three approaches are distinguished by the instantiation of the two parameters *degree of vertical integration* and *degree of linkage*. The degree of vertical integration is defined as the percentage to which decisions of two or more coalition partners are exclusively made centrally. If this degree is zero, then all partners are allowed to make decisions about the deployment of their resources independently. In the event that the degree approximates 1, no partner has the right to decide about the deployment of its own resources. The degree of linkage expresses the availability of inter-organizational information systems among supply consortium partners. If the degree of linkage is close to zero then nearly no common information infrastructure is available, which is often evidence for a fragile and non-

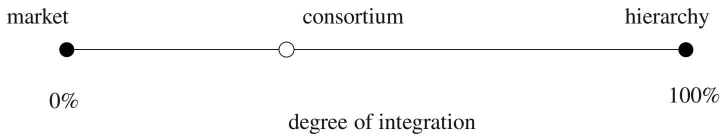

Fig. 1.4 The continuum of supply consortium governance types

resilient partnership. If the degree of linkage is quite high then all partners share the same data basis and inform themselves about process-related events and data modifications. This is interpreted as the existence of trust among the consortium members who want to act together to reach common goals.

The three aforementioned government configurations can now be defined using the extent of vertical integration and linkage degree (Fig. 1.3). *Hierarchical governance* exhibits a high degree of vertical integration and a high degree of linkage. In contrast, *market governance* comes along with low degrees of linkage and vertical integration. As a compromise between hierarchy and market governance, *network governance* is proposed. Here, a strong degree of linkage is preserved but the vertical integration is low, e.g., independent partners are intensively linked by an encompassing information system.

For two reasons, the governance type definitions of Stock et al. (2000) are misleading. At first, the cluster-oriented categorization of the three governance types suggests that the membership is based on explicit values of linkage and vertical integration degrees. Secondly, the term "network" is already used in a quite different context. In order to remedy these two inaccuracies, we propose the continuum representation of governance types shown in Fig. 1.4). We consolidate the two linkage degrees by defining the *integration degree* as half of the degree of vertical integration plus half of the degree of linkage. Only two explicit configuration types are emphasized. A market-type governance exhibits a minimal degree of vertical integration (=0%) and a hierarchy-type governance comes along with the maximal possible integration degree among the consortium members (=100%). In order to avoid confusion, we call all other governance types "consortium".

1.2.1 Centralized Planning

A hierarchically governed supply consortium (focal supply consortium) is typically characterized by the presence of one single leading partner (focal partner), who is selected as the leader due to its financial power or exceptional knowledge of products and processes (Stadtler, 2005). Often, this driving element in the supply consortium is the enterprise that provides the brand of the generated processes, e.g., in die automotive industry the original equipment manufacturer (OEM) leads a supply consortium (Sundermeyer, 2001).

In a hierarchically organized supply consortium, the leader is responsible for all three planning steps: Network Capacity Disposition, Service Agency Resource Dispatching and Resource Deployment. This kind of process configuration is referred to as centralized planning (Pibernik and Sucky, 2006; Lin and Shaw, 1998).

The application of centralized planning requires the willingness and acceptance of all supply consortium partners to subordinate themselves and to transfer all decision privilege to the focal partner. Furthermore, it is necessary that the supply consortium is transparent, so that all information and data relevant for the process configuration are available to the leading entity. This is the reason why the degree of linkage is quite high in hierarchically governed supply consortiums.

1.2.2 Autonomous Control

In a supply consortium that is organized as a market there is no distinguished leading entity. Therefore, each consortium partner itself performs as a coalition representative and collects demand from customers. Consequently, each member becomes responsible for fulfilling a specific customer demand. It derives supply consortium orders. In order to determine the necessary process instance(s) for fulfilling the demand of the customers, each partner tries to hire other consortium members that are willing to execute one or several orders. Payments are transferred for the execution of orders and/or orders are interchanged among consortium partners. In addition, a similar negotiation can happen among the resources in the assignment of requests. Similar to a traditional market trading, the responsibilities for the fulfillment of orders and requests are traded on a virtual market where the reward is fixed pair wise between coalition partners or even between resource representatives. This paradigm for deployment is referred to as autonomous cooperation and control (Hülsmann and Windt, 2007).

A necessary prerequisite for the forming of an autonomous cooperating and controlled supply consortium is the willingness of all members to interact with each other member of the consortium. Furthermore, a common communication platform is required to which a partner can connect in order to exchange messages with other consortium members. However, this platform is used only for exchanging information. An integration of data sources is not intended. For this reason, the degree of linkage is low. In addition, the vertical integration of the coalition members is also quite low due to the autonomy of the coalition partners.

1.2.3 Hierarchical Planning for Consortium-Type Governance

If a market-type decision framework is not realizable and if a strict centralized decision making in a hierarchy layout is also not desired then a compromise between the pure central control and the completely distributed decision making is required.

The consortium members agree that there are some leading units in the consortium that are selected to supervise and instruct other coalition members. In addition, a hierarchy among the leaders is set up. A higher ranked leader specifies instructions towards lower ranked leaders, which are free to decide how to follow the instructions. Such a partnership represents the type of consortium-governed supply system.

The previously described hierarchy in the supply consortium induces a step-wise decision sequence for the deployment of the consortium resources. There is again one consortium coordinator (global coordinator) for the complete supply consortium. This leader receives the customer demand, specifies the supply consortium orders and assigns these orders to consortium members. Here, the global coordinator represents a principal compared to the other coalition members. However, in contrast to the strictly centralized planning paradigm, the global coordinator does not decide how the resources of the coalition partners are used to fulfill the supply consortium orders. Figuratively speaking, the global coordinator selects only the colors of the activities in the selected process instance (Fig. 1.1). The derivation of requests from the tasks of an order assigned to a certain partner is made by a dispatcher belonging to this partner. This so-called service-agency dispatcher compiles the requests from the received orders and assigns these requests to the resources provided by the considered coalition partner. This stepwise decision making is referred to as hierarchical planning (Schneeweiß, 1994; Hax and Meal, 1975).

A semi-hierarchical structure of a supply consortium is induced by the nomination of the consortium dispatchers and the sharing of competencies among them. The consortium coordinator (depicted by the dark-gray shaded area surrounding all agents in Fig. 1.5) has global knowledge about demand and availability of service agents. A service agent dispatcher only knows about the orders received from the coordinator (depicted by the medium-grey shaded area surrounding the service and resource agents). Similarly, an information asymmetry is observed for the relationship between the service agency and the resource representative (the small, light-gray shaded area around it represents the knowledge of the resource representative). The service agency knows about the overall order(s) but the resource representative (e.g., a machine operator or a truck driver) only knows about the requests assigned to this resource.

Hierarchy in the strict sense cannot be found in the supply consortium. A superior actor (i.e. the coordinator) cannot intervene directly in the dispatching decisions of the subordinate service agencies. This fact is mainly based on the desire of each service agency to maintain its organizational and economic independence. Referring again to the differently colored areas in Fig. 1.5, we can interpret this independence as *areas of autonomy* as follows. Each coalition member can only influence (and sometimes even perceive) what happens in the area painted in its own color. Thus, the supply consortium coordinator knows only about the dark-gray shaded part of the supply consortium. A service agency knows only about the medium-gray shaded part and a resource representative can only recognize and understand information arising and affecting the light-gray shaded part of the supply consortium.

The semi-hierarchically organized supply consortium is the most often chosen form for governing the deployment in a supply partnership. It is the only form of

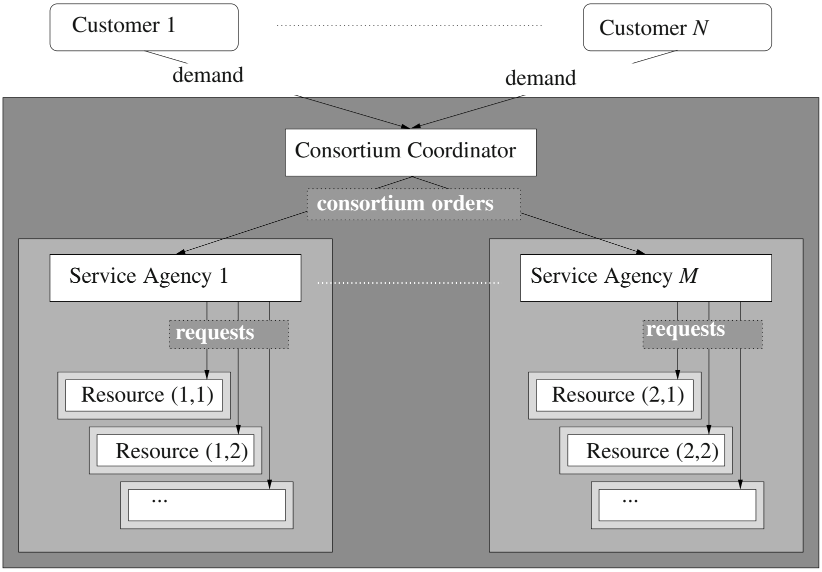

Fig. 1.5 Semi-hierarchical structure of a supply consortium (Bloos et al., 2009)

governance that merges the need for integration of the resource deployment processes of the partners with the desire to allow each partner a maximal degree of autonomy. However, the planning objectives of the global coordinator and of the coordinator of a subordinate service providing partner often do not fit or are even contradicting. Then, special efforts are required to ensure that the subordinate coordinator decisions comply with the decisions of the global coordinator. The discussions of opportunities for coupling the independent deployment decision making of the two kinds of coordinators is not yet well investigated. With this research study, we will develop contributions to the efforts to improve the deployment of resources under special consideration of the consortium-type governance of a supply consortium.

1.3 Transport Processes in Supply Consortia: Relevance and Challenges

The spatially distributed locations of the facilities of the involved supply consortium partners require the transport of raw materials, semi-finished or complete products between the different facilities. Transport processes have to be set up in order to realize the necessary material flow through the value creation stages of the physical network connecting the value creating locations. For this reason, high quality and reliable as well as profitable transport services represent a key contributor to the success of a supply consortium.

The importance of transport services for recent value creation concepts can be demonstrated by a comparison between the annual growth of value creation and the annual growth of freight transport activities. In the European Union the Gross Domestic Product (GDP) increased by 2.5% annually in the period from 1995 until 2007. In parallel, the yearly escalation of the freight transport performance measured in ton kilometers (tkm) was 2.7% (EU, 2009). This reveals a correlation between provided transport services and economic growth.

Although the performance of the transport sector is continuously increasing, its contribution to the overall Gross Value Added (GVA) is declining. For 2006, the provision of the transport sector to the GVA of the 27 states of the European Union was less than 5%. Accelerated by the extension of further low-cost members, the prices and costs of transport services decrease. It becomes harder to operate transport resources in a profitable way. Similar observations are made for the North American economic zone.

The derivation of excellent and high performing transport processes becomes more and more important for the transport providing companies. On the other hand, the expectations and requirements of customers increase as well as the complexity of the process planning. This drives the need for a continuous improvement in process planning. Computerized decision support systems capture the operations planning of transport providing partners in supply consortia (Hall and Partyka, 2008).

1.3.1 Transport Process Derivation in a Supply Consortium

A transport service provider is a specialized service providing consortium partner. It maintains and controls its own transport resources (trucks and trailers, vans, vehicles, ships coaches and/or planes) or has access to external partners who offer these resources. The supply consortium coordinator allocates these resources in order to execute the necessary movement of goods between the spatially scattered value creation locations in the consortium. The *fleet management agent* is the coordinator of a transport service providing consortium partner.

The possibility for the coordinator to reserve or block the transport resources is regulated in the contracts agreed between the consortium members. Service-related quality requirements to be considered by the fleet management agent, like maximal response times or maximal transportation times, are specified as well as cost and accounting-related issues (Stank and Goldsby, 2000) or social issues like the compliance with maximal driving hours (Goel, 2009; Kopfer and Meyer, 2009; Meyer and Kopfer, 2008). The fleet management agent is free to select a resource to fulfill transport orders as long as the agreed requirements are respected.

Different and contradicting planning objectives guide the decision making of the two collaborating decision-making agents. The coordinator aims at offering a reliable demand fulfillment to the customers with the intention of ensuring a longer lasting relationship with customers. In contrast, the fleet management agent aims at achieving a maximal profit from its engagement in the supply consortium. Since the given budget is fixed, its only possibility to increase its profitability is to keep the request fulfillment costs as low as possible. Whenever possible (especially if not forbidden in the aforementioned contracts) additional expenditures are prevented.

The derivation of orders from demand as well as the derivation of requests from orders have to respect the specific requirements of the kind of service agent and of the kind of deployed service agent resource. In this subsection, we present the processing steps necessary to extract transport orders from customer demand. Furthermore, we draw our particular attention to the generation of transport requests from transport orders and to the deployment planning of the transport resources. The derivation of profitable and reliable transport processes from customer demand is figuratively presented in Fig. 1.6.

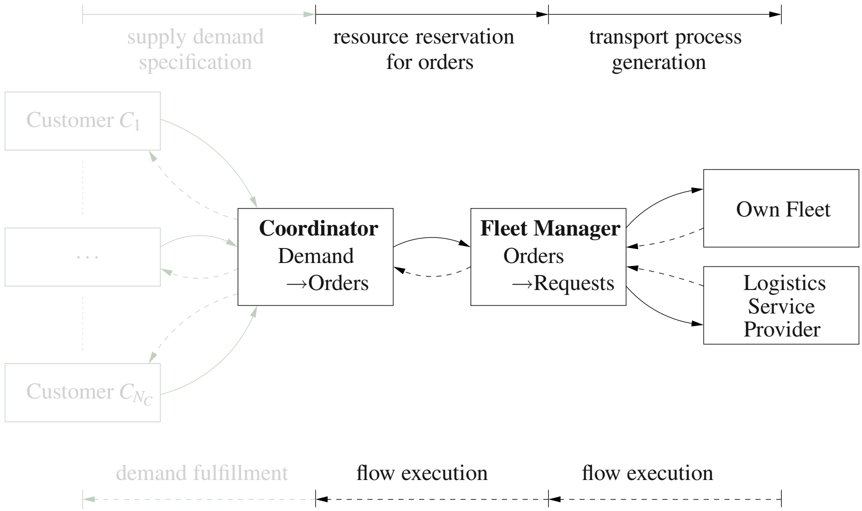

Fig. 1.6 Derivation of Transport Processes from Customer Orders

Network Capacity Disposition - From Supply Demand to Transport Orders. We assume that the coordinator agent has received supply demand from the customer(s) and that it has decided which service agents are incorporated in the fulfillment of the demand. Now, the coordinator allocates (reserves) capacities at the involved value creation facilities (production plants, raw material suppliers). Additionally, it reserves quantities of finished or semi-finished products waiting on stock locations (warehouses) to be used in the demand fulfillment. In the example presented in Fig. 1.7, facilities of different service agents distributed among nine spatially scattered origins of goods S_1 to S_9 are involved in the demand fulfillment planning. Some locations only provide goods (S_1, S_4, S_8 and S_9) while others also require to be provided with products from a previous value creation stage (S_2, S_3,

S_5, S_6 and S_7). The delivery point(s) specified by the customer is (are) the only destination(s) that does (do) not provide any goods used in subsequent stages of the value creation process.

In order to decide which origin facility supplies another destination requiring supply, the coordinator has to solve a multi-commodity network flow optimization problem (Glover et al., 1992; Kennington, 1978) for the network generated by the involved locations. A solution of the flow optimization problem determines the flow among the involved locations, e.g. the quantities of the different raw-materials, semi-finished or finished goods to be moved between pairs of nodes. In order to ensure that materials produced and/or provided by upstream value creation stages arrive in time additional time, constraints like earliest loading times, latest delivery times or even time windows are considered during the derivation of the material flow (Ceselli et al., 2008; Grünert and Irnich, 2005; Schönberger, 2005).

The determined quantities to be move along the connections of pairs of nodes (the arcs) and the temporal requirements are merged in transport orders. A transport order is a 6-tuple $(\sigma, \delta, c, q, tw_{\sigma}^{dep}, tw_{\delta}^{arr})$ in which the origin σ and the destination δ of the goods of type c and the quantity q to be moved are determined. In order to enable a synchronization of consecutive demand fulfillment steps, a loading time window tw_{σ}^{dep} at the source and an unloading time window tw_{δ}^{arr} at the destination are specified.

The solid arcs in Fig. 1.7 represent the need for moving specified quantities of goods between facilities. They represent a solution of the aforementioned multi-commodity flow optimization problem. The example order shown for the arc from S_1 to S_2 expresses the transport demand for 100 pallets to be picked up between 6.00 and 10.00 at S_1 and to be delivered at S_2 not before 17.00 and not later than 18.30.

The solving of the time constrained network flow problem leads to a solving of the demand-to-order task of the supply consortium coordinator represented by the left bold-printed box in Fig. 1.6. All generated transport orders are put into the *transport order pool* and are processed to the fleet managing agent (represented by the arc labeled by *resource reservation for transport orders* in Fig. 1.6). The dashed arcs represent the contribution of the executed movements to the fulfillment of requests and orders.

Resource Disposition of the Fleet Manager: Generation of Requests and Processes.

The immediate and direct assignment of transport orders to transport resources is not accepted by a fleet operating service agent. The reasons are two-fold. At first, the fleet manager represents an independent company. He is responsible for all operations executed in his enterprise. Therefore, it is strongly necessary that he decides about the usage of his resources independently. Secondly, the fleet manager is not only responsible for the orders of one supply coordinator but, since he is independent, he also serves coordinators of other supply consortia. Consequently, his overall resources are involved in different value creation projects and only the fleet manager can make a suitable assignment of capacities to orders (respecting the contract requirements agreed among the partners).

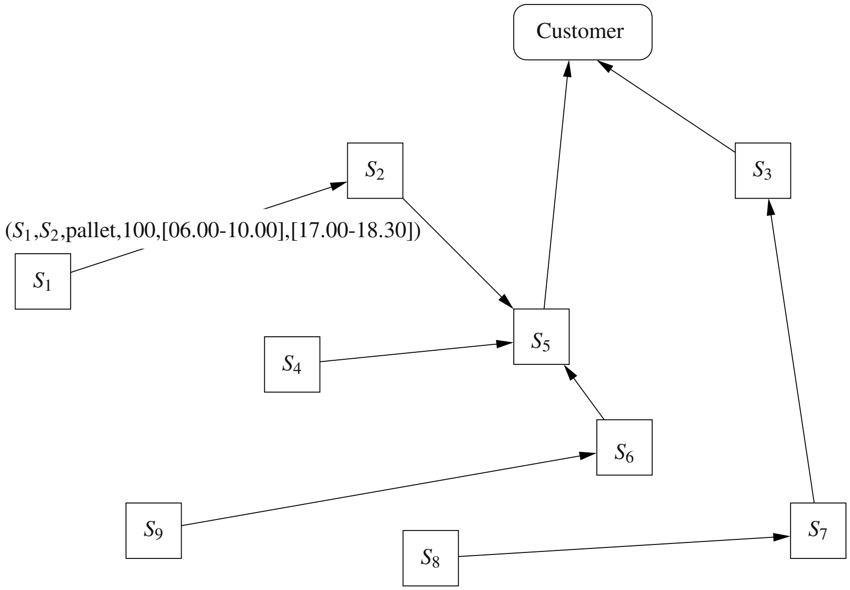

Fig. 1.7 Network flow to be covered by transport orders

All transport orders have been generated by matching the customer demand, the quantities provided by the involved service agents (locations) and the temporal requirements of the value creation process (time windows, precedence relations, ...). However, since the coordinator agent does not know how the fleet management agent wants to fulfill the transport orders, it is not possible to consider physical constraints like truck capacities, exact schedules or loading incompatibilities (for hazardous goods). The fleet manager agent's first task is to derive transportation requests from the transport orders contained in the accessible transport order pools. We define a transportation request as the smallest indivisible entity of goods (a box, a pallet, a container). It is not possible to split up a request, so therefore a complete request has to be assigned to a certain resource for any movement in the network.

The major goal of the derivation of requests from orders is to achieve executable (feasible) transport tasks. Therefore, for each transport order it is initially checked whether it is possible to serve this order using an available resource.

Capacity conflicts. In the event that there is a resource that can execute the order en bloc then this order is copied as a request into the transport request pool maintained by the fleet manager. In case it is not possible to serve the complete request because the specified quantity exceeds the capacity of each single transport resource, the considered transport order is copied several times. After the copy, the quantities in the copies are reduced so that the quantities given in the copies sum up to the original quantity; but in each copy the capacity does not exceed the capacity of at least one of the available transport resources. The copying is repeated recursively until the two aforementioned conditions are met. After copying and quantity reduction, the copied orders are inserted as requests into the maintained request pool (Archetti and Speranza, 2007; Chen et al., 2007). In some applications, the split of orders is combined with the later processed resource deployment (Archetti et al., 2008; Nowak et al., 2008) so that even requests with capacity conflicts are copied into the request pool.

Temporal conflicts occur if the loading (or unloading) cannot be scheduled within the determined pickup window tw_{σ}^{dep} (or within the delivery time window tw_{δ}^{arr})). There are at least two reasons for such a conflict. At first, the velocity of the available transport resources is too low. The second reason is that the estimated length of the transport exceeds the maximal allowed driving time of a vehicle (Meyer and Kopfer, 2008). A remedy for temporal conflicts in transport orders requires the involvement of the affected global coordinator in order to re-determine new time constraints for the orders. After the time windows have been updated and if no capacity conflicts are detected anymore, the transport order is copied as a request into the transport request pool.

Loading conflicts are caused by different commodities included into a single transportation request that cannot be moved simultaneously on the same transport resource. A loading conflict typically occurs if the transport order includes different hazardous goods which are not allowed to be moved together in order to avoid destruction following an accident or collision. Similar conflicts possibly occur in the transport of foods and pre-food products or other perishable goods. Here, conflicting requirements, like maximal or minimal temperatures to be ensured during the complete transport chain, have to be respected in the physical execution of the goods movement (Jedermann et al., 2009; Hsu et al., 2007). If loading conflicts occur, transportation orders are recursively copied and from each copy, conflicting goods are removed so that at the end, all goods to be moved are contained in a copy but all copies are free of conflicts. The generated copies are then inserted into the request pool.

At the end, after having checked all available transport orders the set of transportation requests must be composed into executable processes (deployment). Several interdependent decisions must be made in order to enable the available transport resources (e.g. trucks or vehicles) to fulfill the transport demanded by the partners of the supply consortium (Krajewska and Kopfer, 2008; Kopfer and Krajewska, 2007; Schönberger, 2005; Feillet et al., 2005; Pankratz, 2002). Fig. 1.8 presents an overview of the dispatching steps, including the request generation as a preprocessing step.

In the disposition step, each request is assigned to a certain transport fulfillment cluster. Some requests are assigned to the fleet of vehicles fully controlled by the control unit of the considered transport company (*self-fulfillment cluster*). Requests that require an immediate fulfillment but which are not assigned to one of the available vehicles are directed into the *subcontracting cluster* and are then shifted to a Logistic Service Provider (LSP). The LSP is paid for taking over the fulfillment responsibility for these requests. All remaining requests are postponed (*Postponement Cluster*). The postponed requests are not considered for the current deploy-

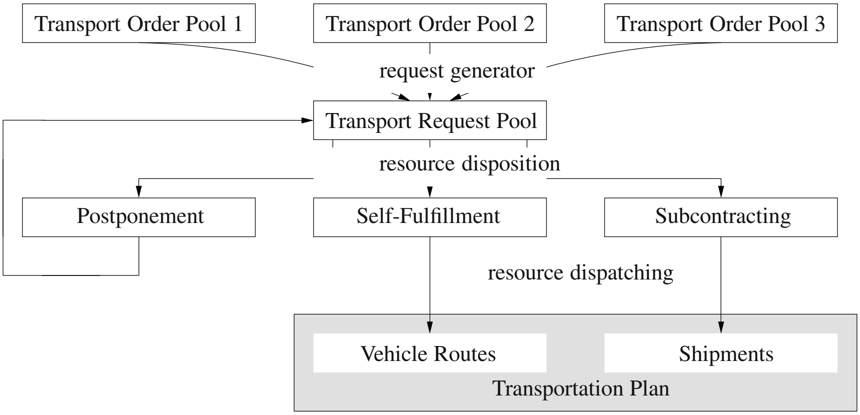

Fig. 1.8 Request-based transport resource deployment

ment planning but will be re-considered in subsequent deployment. The decision about the selection of the right cluster is referred to as mode selection (Schönberger, 2005).

Routes describing the sequence of visit of customer sites are generated from all requests assigned to the self-fulfillment cluster and each route is assigned to one vehicle of the own fleet (Golden et al., 2008). Each vehicle executes zero, one or several routes according to a schedule that determines the en-route arrival and departure times. The compilation of routes is guided by the goal to keep the route execution efforts (costs, duration, make-span and travel distances) as small as possible. Each generated route is one-to-one related to a physical movement (flow) of transport resources through the given transport network (road, track or air-based connections). Therefore, routes represent the processes of the used transport resources.

The costs for the incorporation of LSPs to fulfill the requests assigned to the subcontracting cluster are determined by a previously known tariff. The tariff typically specifies the amount to be paid for the transport service providing agent to an LSP for moving a collection of goods from an origin to a destination (shipment). The size, volume, weight and/or value are considered in the determination of the final payment. Often, tariffs are degressive with respect to the quantities contained in a shipment so that a bundling of several requests into a shipment leads to the exploitation of economies of scale (freight optimization). In order to keep the costs for the fulfillment of the subcontracted requests as low as possible, the compilation of adequate shipments is the major planning task in the subcontracting cluster. From the point of view of the transport service providing agent, the shipment generation is carried out for calculating the LSP-costs (Schönberger and Kopfer, 2005). However, the physical flow of goods through the transportation network maintained by the LSPs is sometimes different from the flow determined in the tariff calculation step. The transport service providing agent has no control over the physical movement of transport resources of an LSP.

Both planning tasks (route compilation and shipment generation) form the second step in the deployment procedure. This step is referred to as resource dispatching.

All three decision problems (cluster selection, routing and scheduling and shipment generation) are interdependent. For this reason, requests are tentatively inserted into the self-fulfillment cluster. If it is possible to combine this requests with other requests into profitable routes then this request is definitively assigned into this cluster. In the event that a profitable consolidation of this request with other requests is impossible, the considered request is re-assigned into the subcontracting cluster. The tentative request-to-cluster assignment enables an integration of both planning tasks: routing and shipment generation (integrated resource disposition and dispatching). Krajewska (2008) identifies several integration approaches, analyzes the challenges and benefits of the found types in detail and provides a classification scheme.

The planned processes are stored in the *transportation plan*. This contains the generated routes for the fully controlled own resources and the information about the pickup as well as the deliveries of goods associated with subcontracted requests (Crainic and Laporte, 1997).

1.3.2 Process Endangering Events and Data Uncertainty

The transportation plan instructs the fleet-manager-controlled fleet and the incorporation of the LSPs and determines how (and when) the transport requests are fulfilled. Thus, it contributes to the fulfillment of the transport orders specified by the supply consortium coordinator. However, unexpected or unpredictable events endanger and compromise the complete execution of the found processes and the order fulfillment is interrupted. The processes in the transportation plan have to be checked for being compromised and an update of disturbed processes becomes necessary.

Unexpected events impact the means of transport differently. Deep sea shipping processes are mainly affected by disturbances in the loading or unloading ports. Prolonged turnaround times or delays in the availabilities of berths occur and often lead to delays in the subsequent process parts (Vernimmen et al., 2007). Sometimes, the delays can be temporarily offset by increasing travel speeds. However, additional dispatching interventions are often impossible due to the fact that alternative process decisions are not available. A similar observation is made for rail-based transport systems. Due to the strict and inflexible track-based infrastructure and due to deficiencies in the determination of exact positions and causes of disruptions it is often impossible to react appropriately in order to reduce the negative impacts of unexpected events (Jacobs, 2003). The same intervention deficiencies are mentioned in the context of reacting to disruptions affecting inland waterways. In addition, this means of transport is highly dependant on suitable weather conditions. Floods, low tides and ice drifts regularly interfere with the schedules of barges. Bad weather conditions, security alerts and high volumes of traffic are the major drivers for disruptions in passenger air transportation. In order to meet the passenger requirements (no re-routing, and no delay, and no change of aircraft and no transfers) sophisticated disruption management strategies have been developed for passenger airline operations (Kohl et al., 2007). However, in air-cargo industries, more opportunities to react to disruptions and unexpected events are given since the freight can be re-routed and re-scheduled without producing complaints (Derigs et al., 2009).

Due to the very flexible underlying infrastructure, the economic importance and the number of involved entities, road-based freight transportation is the most interesting means of transport from the viewpoint of analyzing the impacts of uncertainty. In the remainder of this study we will focus on challenges in the management of disruptions affecting the road-based freight carriage. Furthermore, we will present new concepts for the management of these disruptions.

Zeimpekis et al. (2007) distinguishes three different kinds of events that compromise the execution of planned road transport processes.

Portfolio Modifications. Changes in the maintained portfolio of requests are the major driver for a transport process update. Additional requests must be integrated into previously derived transport processes (Saéz et al., 2008; Richter, 2005; Mitrović-Minić et al., 2004; Savelsbergh and Sol, 1998; Lund et al., 1996; Ausiello et al., 1994), the cancellation of requests sometimes causes a process adaptation in order to keep or re-achieve the process profitability (Rego and Roucairol, 1995), and incomplete or wrong predictions of the actually demanded transport resource capacity make an update of the once fixed processes necessary if the once-allocated capacity is exceeded by the additional demand (Lund et al., 1996). Portfolio modifications occur in supply consortium planning if additional customer demand is submitted to the coordinator. In such a situation, the coordinator has to update the transport order pool in order to announce the need for additional transport services to the corresponding coalition partners. Consequently, the request pool must be updated and additional requests occur or existing requests are modified. Similarly, the request pool requires an update if it turns out that the transport demand announced to the coordinator by other service agents does not cover the needed transport services.

Infrastructure Modifications. Events restricting the infrastructure availability form a second group of incidences compromising the execution of once planned transport processes. Traffic congestion and adverse traffic conditions that prolong the travel times along roads are in the focus of scientific research (Lo and Hall, 2008; Fleischmann et al., 2004b).

Transport Resource Shortages. Vehicle breakdowns and vehicle outages shrink the capacity of available transport resources so that some requests cannot be picked up and moved as planned (Zeimpekis and Giaglis, 2005).

Uncertainty of planning data is caused by the occurrence of the events implied by request portfolio modifications, infrastructure modifications or transport resource shortages. It refers to the vagueness and/or inappropriateness of the data used to decide about the deployment necessary to fulfill the current actual request portfolio.

Request Uncertainty is the general term subsuming the data uncertainty caused by a spontaneous and unforeseen change of the composition of the transport requests pool (appearance of additional customer requests, request cancellation, request variation) considered so far in the processes (Fleischmann et al., 2004b; Madsen et al., 1995). A detailed differentiation of the data vagueness is carried out with respect to a differentiation of the place or activities in the physical value creating system from which the data uncertainty occurs. Delays of down-stream activities during the production or preparation of the goods associated with transport requests often cause delays in subsequent value creation stages, e.g. a delayed provision of the goods for the transportation is called *Downstream Uncertainty* (Williams, 1984). Last-minute changes of the quantities to be moved are critical if the difference between the ex ante announced and the actual volume to be moved is not available on the disposed vehicle (*Quantity Uncertainty*) as observed by Goel and Gruhn (2008). *Handling Uncertainty* refers to on-site operations of loading or unloading which last longer (or shorter) than planned (Hadjiconstantinou and Roberts, 2002) and *Transshipment Uncertainty* expresses possible delays caused by the non-availability of ramps, gates or special loading or unloading equipment in transshipment or cross-docking terminals (Yu et al., 2008). If different types of goods cannot be loaded as planned or are not allowed to be loaded by the same resource then *Loading Uncertainty* (L.Delaitre et al., 2007; Gendreau et al., 2004) compromises the process execution. The movement of picked up goods from the loading place to the unloading location is compromised by blocked or congested roads so that the vehicles' travel time is prolonged (*Travel Uncertainty*). Consequently, a late arrival at the delivery place results (*Upstream Uncertainty*) as investigated for example by Lo and Hall (2008). Often, one kind of event causes another kind of process compromising event so that several sources of uncertainty have to be handled simultaneously.

In the remainder of this book, we limit our investigations to modifications of the request pool. We assume that neither transport resource shortages nor infrastructure bottlenecks occur abruptly.

Part I
Model-Based Transport Process Planning: Approaching the Limits

Chapter 2
Transport Processes and Uncertainty

This chapter addresses the management of transport processes in volatile environments. In particular, we analyze how the supply consortium coordinator agent and the fleet management agent can be integrated despite the contradicting objectives concerning the planning of transport operations.

We start with a classification of different types of decision problems and decision models (Section 2.1). In particular, we explain the differences between a static and a dynamic decision problem. Generic approaches for mapping the uncertainty (related to dynamic decision situations) into formal decision models are presented and classified.

Demand peaks are a major source of transport process disturbance. A compilation of approaches to cope with this specific interference is presented in Section 2.2. We investigate whether demand peak management concepts for production scenarios can be transferred to transportation planning.

In order to prepare the setup of a simulation study of a transport process planning scenario in a volatile environment, we introduce a specific planning problem in Section 2.3.

2.1 Formalization of Uncertainty within Decision Problems

A decision problem P is called *static* if it can be solved completely at one time point t_0 (Sellmaier, 2007). In all other cases, a decision problem is called a *dynamic problem* or an *online (decision) problem*. Such a decision problem is characterized by the need for repeated decision making and decision revision at consecutive times $t_1, t_2, \dots$ (Brehmer, 1992). Consequently, a dynamic decision problem P can be expressed as a sequence $P_0, P_1, \dots$ of static decision problems. The decision problem P_{i+1} represents P_i enriched by the additional data acquired between time t_i and time t_{i+1}. A dynamic decision problem is referred to as *real-time decision problem* if the time to re-compute a solution (deliberation time) of the instance P_i of the online problem is limited (Séguin et al., 1997). Decision problems are often dynamic

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*,
Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_2,

because decision-relevant information is missing at a decision time t_i (Lund et al., 1996). Typically, decisions made at time t_i have impacts on to the decision alternatives available in later decision situations (Busemeyer, 2001). If the appearance of future events can be approximated by probability distributions then a dynamic decision problem is called a *Markov Decision Process* (Littman et al., 1995). The portion of information revealed after the initial decision time t_0 is expressed by the *degree of dynamism* (Larsen et al., 2008; Lund et al., 1996).

In transportation logistics, dynamic assignment problems (McKendall Jr. and Jaramillo, 2006; Fleischmann et al., 2004b; Mitrović-Minić et al., 2004; Pankratz, 2002; Powell and Cheung, 2000; Lund et al., 1996; Powell, 1996; Madsen et al., 1995) as well as combined dynamic vehicles routing and scheduling problems (Rego and Roucairol, 1995) are major research fields. Furthermore, dynamic emergency team relocation problems (Rajagopalan et al., 2008) are in the focus of scientific interest. Surveys on the differences between static and dynamic vehicle deployment problems are given in (Psaraftis, 1995, 1988). If all data that are considered for the definition of the decision problem P are assumed to be given then P is called a *deterministic* (Illa, 1966) decision problem, otherwise P is named *non-deterministic* (Munera, 1984).

Let M be a decision model for the decision problem P. The model M is called a *static model* if the realization time of a decision is irrelevant for the feasibility and the evaluation of the decision (Saéz et al., 2008; Sellmaier, 2007). The mixed integer linear program formulation of the traveling salesman problem (Garfinkel, 1985) is a representative example for a static model: Only the originating and the terminating node determine the costs for traveling along an arc but not the time when the arc is traversed. If a model M is not a static model then M is called a *dynamic model* (Sellmaier, 2007) that is applied if e.g. capacity availabilities or realization costs alter over time. Dynamic optimization models are used to represent routing tasks if time or costs to pass through roads or paths alter over time (Moreira et al., 2007; Potvin et al., 2006; Fleischmann et al., 2004a; Ichoua et al., 2003). Other applications of dynamic models comprise (among others) core control in computer sciences (Ramamritham and Stankovic, 1994) or machine scheduling (Seiden, 1996). If uncertain data are described by probability distributions then M is called a *stochastic decision model* (Laux, 1982). Let M_i denote a decision model for the instance P_i of an online decision model. The sequence of decision models $M_0, M_1, \ldots$ is called an *online decision model* for the online decision problem $P_0, P_1, \ldots$.

2.2 Operational Management of Peaks in Transport Demand

The generated transport processes specify the transport services to be carried out. Pickup locations, delivery locations, quantities to be moved and operation times are determined so that the deployed transport resources are used with highest efficiency. However, the completion of the processes is in danger of being compromised by uncertainty resulting from the occurrence of unforeseen events (e.g. additional requests) as discussed above.

Independent of the type of uncertainty, the incomplete knowledge of the complete and exact planning makes the planning of transport processes more difficult. During a decision making process the degree of uncertainty (e.g. event probabilities) has to be considered in the evaluation of decision alternatives. In addition, strategies to manage events that have not been expected but which nevertheless have been appeared have to be available in order to ensure that the operations remain running even under unexpected circumstances. In this section, we direct our interest to concepts discussed to ensure that a transport process planning remains effective even if a significant workload increase suddenly happens as a result of capacity uncertainty. A specific situation is investigated in which not only the time and location of the additional requests but also the number of additional requests released at a certain time is unknown in advance. Actually, a balanced stream of requests, regularly entering the request pool, is enriched by a temporary workload peak that begins at a previously unknown time and lasts over an unknown period. During the peak period, the number of incoming requests at a certain time is significantly higher than before or after the peak period.

If the capacity of the available transport resources is already exhausted by the so far known requests then the additional requests cannot be integrated into the processes. The flexibility of the processes is in danger of being decreased (Schönberger and Kopfer, 2009c), which means that the probability for serving the additional requests in compliance with customer time window requirements runs the risk of being decreased, too.

Obviously, the reservation (blockage) of *emergency capacity* for the vehicles belonging to the controlled fleet supports the compensation of workload peaks. However, two aspects compromise this remedying idea. (i) The additional resources are not profitable during off-peak periods. Therefore, it is not possible to maintain such an extra capacity over a longer period. (ii) If none of the vehicles with emergency capacity is located in the nearer surrounding of the request-associated location(s) then reserving additional capacity only for peak situations is futile.

How to cope with an overloading of the own fleet? The acceptance and exploitation of shortfalls and under-capacity situations is proposed (Zäpfel, 1982) but, due to the currently customer-dominated transportation market, not enforceable. Also temporal staffing arrangements (temporary workers) (Kalleberg et al., 2003; Zäpfel, 1982) cannot remedy transportation capacity bottlenecks since additional vehicles must be available, too. Additional deliveries and postponement are also proposed (Zäpfel, 1982) but, again, not applicable in freight transportation logistics with supply function. In-time and complete deliveries are unconditional necessary in order to keep the process performance on a high level. Deferred deliveries corrupt the flow of goods through the value creation stages. The spatial pre-distribution of deliverable goods over the operations area and a pre-loading of trucks is proposed and evaluated in Calza and Passaro (1997). However, these approaches are very capital-intensive. Pre-peak resource build-ups (Ronen et al., 2001) are also impossible since the peak occurrence is not predictable and transport services cannot be stored. These previously mentioned strategies to mitigate the negative impacts of demand peaks have their origin in production management and exploit the special conditions in this sector. In particular, the significant larger geographic extent of transportation networks prevents the ad-hoc short-term re-allocation and increase of transport resources. For this reason, the strategies successfully applied in production applications within factories or concentrated locations cannot be transferred into the area of goods transportation.

Instead of holding or blocking a certain amount of capacity constantly in readiness, a temporal and demand-oriented provision of additional transport capacity is preferable (external procurement, third party supply, externalization). LSPs (External forwarders or transport service providers) are incorporated in order to execute selected requests (subcontracting). The LSPs provide capacity in their transport resources (e.g. vehicles). Thus, the intensification of the usage of the subcontracting cluster in transport resource dispatching is the preferred activity to achieve a quick capacity expansion in response to a beginning demand peak. As a consequence, high service reliability is ensured even during a demand peak disturbance.

In order to ensure that the subcontracting opportunity is available whenever and wherever it is needed, a consortium coordinator sets up and maintains longer term contracts with one or more transport service providing companies (the LSPs). Their transport networks cover the considered area of service (Ronen et al., 2001). The set up contracts specify the provided capacities and the fees which are paid to the service providers. A survey on subcontracting arrangements can be found in (Kopfer and Krajewska, 2007).

With respect to the incorporation of LSPs in transport request fulfillment, we make the following assumptions for the sequel of this research.

- The fleet managing agent and a sufficiently large number of logistic service providers have agreed respective contracts on a request-oriented base (Krajewska and Kopfer, 2008; Kopfer and Krajewska, 2007). According to such a contract, each single request can be given to an LSP. The fleet managing agent has to decide which requests are subcontracted.
- Only one LSP can be chosen to fulfill a certain request so that no provider selection becomes necessary for the fleet manager. Immediately after a request has been transferred to the corresponding LSP the service provider starts with the execution of the request and ensures a timely fulfillment.
- We refrain from explicitly considering preparation or setup times for incorporating an LSP.
- A previously fixed fee is paid by the coordinator agent to the LSP for each selected request.

The contracts with the LSPs secure the fleet manager agent an opportunity to make use of the resources of the LSP immediately if necessary. Thereby, the fleet managing agent has the *right but not the obligation* to choose LSP services. For this reason, the externalization (subcontracting) of a request in the context of the aforementioned contracts is interpreted as an *option* (Wöhe, 1993). The supply consortium (by means of the fleet managing agent) is enabled to exert this option at any time during the duration of the contract (*American option*). It is possible that such an option remains unused. The price for exercising the option is fixed at the time when the contract is signed (*call option*) as explained in Black and Scholes (1973).

Prices to be paid to the LSP in case those options are used are quite high, because the prices must include the costs for hedging the LSP's risk that the fleet-manager does not exercise an option. If both the own fleet and the options are available then the first mentioned alternative would be preferentially selected by the fleet manager. However, if the own resources are exhausted then the higher LSP-fee is justified and acceptable because the timely request execution supports the goal to keep the quota of reliably completed customer orders high.

2.3 A Dynamic Vehicle Routing Problem with Subcontractor Options

We want to study the impacts of a demand peak on transportation processes within simulation experiments. Therefore, we introduce a specific dynamic transport process planning scenario. The environment in which the planning decisions are made is outlined in 2.3.1. Next, the goals and interests of the involved agents and proposals for the consolidation and integration of the aims are discussed (2.3.2) and the planning task of the transport service agent is described (2.3.3). We conclude the scenario introduction with the description of the generation of parameterizable artificial test instances for the described scenario (2.3.4).

2.3.1 General Scenario Outline

We assume that the supply consortium coordinator has agreed a contract with a customer. This customer specifies demand and submits this demand irregularly and at unpredictable times to the coordinator. Immediately after the reception of the demand, the coordinator specifies the consortium orders and inserts the generated orders into the order pools of the involved service agents.

In order to enable the supply consortium to execute the necessary transport of goods between the partners, the coordinator sets up a temporary contract with one (and only one) transport service partner (service agent). The transport service agent receives a certain amount as a budget. From this budget it has to pay its costs for fulfilling the necessary transport orders (travel expenditures and fees to be paid for subcontracted shipments). The difference between the overall budget and the request fulfillment costs remain as profit at the service agent.

According to the contract between the customer and the coordinator a service degree is fixed, e.g. a given percentage p^{target} of the customer's transport demand has to be fulfilled within the customer specified time restrictions. The current process punctuality rate p_t is defined by $p_t := \frac{f_t^{punc}}{f_t}$ (representing the current reliability of the transport system). There are f_t requests whose completion times (already realized or scheduled) fall into the period $[t-500, t+500]$ (moving time window). Among these f_t requests the number of f_t^{punc} requests is completed within the previously agreed time windows. If the reliability requirement is met then the quotient p_t must not fall below the threshold value p^{target}. This quota can be explained as follows: Each demand comprises goods necessary to keep the production processes at the customer's factories running and goods used to build up security stocks. The first kind of goods must be provided in time while the second kind of goods can be delivered later without causing corruption of the running production processes.

For the reason of simplification, we assume that each transport order is an executable task. Hence, neither capacity nor temporal nor loading conflicts occur and each transport order is copied into the request pool of the transport service agent immediately after the order has been inserted into the order pool. A request instructs a transport resource to visit a given location during a customer specified time window. Examples, in which such a kind of request occur are related to situations in which a large number of small-sized packages is loaded at the beginning of a day-trip so that packages (like spare parts) can be delivered to a large number of customers without the necessity to re-visit a loading berth. Other applications are the collection of used consumable items (collection of used laser or ink-cartridges during office-hours) and the dispatching of service crew and repairmen dispatching (Madsen et al., 1995). For a summary of further related scenarios we refer to Beullens et al. (2004)).

We investigate the impacts of request uncertainty. A stream of arriving requests must be served. Additional requests are released regularly and the requests must be integrated reactively into the existing transportation plans (processes). Whenever an additional request corrupts the execution of the so far followed processes a process revision (re-planning) becomes necessary. Typically, the processes of several transport resources have to be updated because a re-assignment of requests is necessary if the originally selected vehicle is no longer scheduled to fulfill one or more requests as decided in the last plan update. Subsequently arriving requests are handled by updating the existing transportation plan. Therefore, the planning situation outlined above is a dynamic decision situation (Brehmer, 1992). The decision problem is formulated as an online decision problem. Each instance P_i of this problem is static and deterministic (all data relevant at the re-planning time t_i are assumed to be known).

2.3.2 Coordinator's and Service Agent's Interest Integration

The two considered agents follow different and to a certain extent contradicting planning goals. While the transport service agent aims at maximizing his profit by keeping costs as low as possible the coordinator agent targets the fulfillment of the promised least punctuality degree ($p_t \geq p^{target}$). The endeavor to minimize the operational costs restrains the transport service providing agent from investing additional expenditures to increase the punctuality rate if this rate has fallen below the threshold value. For this reason, the contract between the coordinator and the service agent must contain some specific measures in order to ensure that the service agent's endeavor for profit maximization does not lead to a negligence of the coordinator's requirements.

2.3.2.1 Reference Configuration

The direct way to ensure the achievement of the desired least punctuality rate is to force the service providing agent to generate only processes fulfilling the least punctuality requirement. Every process proposal which does not observe the least punctuality condition and which leads to a lower punctuality will be rejected by the coordinator. The minimization of costs is only addressed as a second ranking desire. It is not a mandatory planning requirement. The achievement of least punctuality is the superior planning goal. We refer to this configuration as the hard condition configuration (HARD-configuration) of the investigated supply consortium scenario.

The least punctuality rate has been agreed between the coordinator and the transport-providing partner. Therefore, an average workload has been assumed while deriving the agreed service level p^{target}. A significant increase in the number of customer sites (workload peak) augments the process costs and therefore lowers the profit of the transport partner. It is quite unfair that the additional expenses are not shared with the coordinator (and thereby among all supply consortium partners). Thus, a strict enforcement of the least punctuality discriminates the transport partner and enforces him to leave the consortium as soon as possible in order to avoid serious financial damage. For this reason, the HARD configuration is neither realistic nor applicable. We use it as reference configuration to provide comparable results for simulation experiments. These reference results enable an estimation of the costs necessary for ensuring the achievement of the service goal.

2.3.2.2 Penalization of Late Arrivals

Actually, the coordinator must provide incentives to each service providing partner in the supply coalition in order to act in the sense of the common goals instead of acting only in the sense of its own interest. For each partner, a major motivation to participate in the supply consortium is to maintain or increase the own profit. Vice versa, the endeavor of a partner to maximize its profit enables the supply consortium controller to influence and regulate the behavior of a partner. The partner is promised a higher benefit if it acts in accordance with the common supply consortium goals but its profit is reduced if the partner acts otherwise.

The supply consortium coordinator receives fees paid by the customers for the fulfillment of the customer orders. Using the sum of earned fees, budgets are funded that are used to cover the material flow process costs specified by the service agents. In order to stimulate a partner agent to determine processes of highest efficiency, the difference between the budget and the process costs remains at the service partner as its gain (profit). The main idea of the penalty configuration (PEN-configuration) is to penalize the transport partner for each request whose on-site fulfillment starts with a delay. Thereby, this partner is motivated to fulfill as many requests as possible on time so that the punctuality rate p^{target} can be guaranteed. If a demand peak occurs then the service-providing partner can freely decide whether to accept the profit reduction or to spend more effort on maintaining or even increasing the service level. Here, the negative impacts of a workload peak are shared between the coordinator (and therefore among all partners) and the transport partner: The latter pays penalties for late arrivals but the supply chain consortium accepts a temporarily reduced punctuality.

2.3.3 Dispatching Task of the Fleet-Managing Agent

The coordinator receives customer demand continuously over time. Every Δt time units he generates orders from the customer demand and the resulting requests are to be executed by the transport providing partner. The reception of additional requests triggers a process revision to incorporate the additional requests into the so far followed transport processes. The process-planning problem of the transport service providing agent is therefore a dynamic decision problem, which is solved in online fashion, e.g. a process revision is carried out in event-driven fashion in response to the additionally submitted requests. Consequently, a sequence of concatenated decision problems is defined. Each instance is formulated as an optimization model. Solving such a model means to find the most profitable process decisions for the transport operations considering the so far actually known planning data. Each instance represents a generalized common vehicle routing problem with time windows (Solomon, 1987). It is the multi-vehicle version of the traveling repairman problem (Irani et al., 2004), which is additionally extended by subcontracting. The transport partner's dynamic decision problem has been previously formulated and investigated in Krumke et al. (2002).

A release of one or more additional requests initiates the revision of the so far constructed routes for the own vehicles. If needed, some requests which are so far assigned to the self-fulfillment cluster, are excluded from the routes of the own vehicles and forwarded to an LSP. A re-assignment of requests formerly assigned to the subcontractor cluster into the self-fulfillment cluster is not possible.

The arrival times at some customer sites may be postponed in order to serve one or several additional customers earlier by the same vehicle. Furthermore, the number of additionally released requests temporarily increases unpredictably, so that workload peaks occur from time to time. In a *high quality (HQ) period* the requirement for the least punctuality is fulfilled ($p_t \geq p^{target}$) but in a *low quality (LQ) period* the required punctuality rate is not attained ($p_t < p^{target}$).

Subsequently arriving requests are accepted and handled by updating the existing transportation plan. A sequence of transportation plans $TP_0, TP_1, TP_2, \ldots$ is generated reactively at the ex ante unknown update times $t_0, t_1, t_2, \ldots$ and each single transportation plan is executed as long as it is not updated. In order to determine the transportation plan TP_i at time t_i a static decision problem P_i must be solved. The problem P_i represents the task of selecting the least cost transportation plan from the set of all possible transportation plans. Thus, P_i is an optimization problem and the sequence $P_0, P_1, \dots$ is an online optimization problem representing the dynamic decision problem of the subordinate transport service agent. In the following, we propose a mathematical model M_i for each instance P_i of this online optimization problem.

A single request r attains consecutively different states that change with ongoing time. Initially, r is **k**nown but it is not yet scheduled (K). Then, r is assigned to an own vehicle (I, short for **i**nternal fulfillment) or subcontracted (E, short for **e**xternalization). If the operation at the corresponding customer site has already been started but not yet been finished the state S (short for **s**tarted request) is assigned to r. The set $R^+(t_i)$ is composed of additional requests released at time t_i. Requests completed after the last transportation plan update at time t_{i-1} are stored in the set $R^C(t_{i-1}, t_i)$. At time t_i, the recent request stock $R(t_i)$ is determined by $R(t_i) := R(t_{i-1}) \cup R^+(t_i) \setminus R^C(t_{i-1}, t_i)$. Each request belongs at each time to exactly one of the sets $R^K(t_i)$, $R^E(t_i)$, $R^I(t_i)$ or $R^S(t_i)$, in which the requests having a common state are collected.

The problem P_i of updating a transportation plan at time t_i is as follows. Let V denote the set of all own vehicles, $P_v(t_i)$ the set of all paths (sequence of visited sites beginning with the position of the vehicle at time t_i and ending with the central depot) executable by vehicle v in TP_i and let $P(t_i)$ denote the union of the sets $P_v(t_i)$ ($v \in V$). If the request r is served in a path p then the binary parameter a_{rp} is set to 1, otherwise it is set to 0. A request r, already known at time t_{i-1} that is not subcontracted in TP_{i-1} is served by vehicle v_r. The travel costs associated with path p are denoted as $C^1(p)$. Finally, $C^3(r)$ gives the subcontracting costs of request r.

In order to code the necessary decisions for determining a transportation plan in the representation M_i of P_i, we deploy two families of binary decision variables. Let $x_{pv} = 1$ if and only if path $p \in P(t_i)$ is selected for vehicle $v \in V$ and let $y_r = 1$ if and only if request r is subcontracted.

$$\sum_{p \in P(t_i)} \sum_{v \in \mathscr{V}} C^1(p) x_{pv} + \sum_{r \in R(t_i)} C^3(r) y_r \rightarrow \min \tag{2.1}$$

$$\sum_{p \in P_v(t_i)} x_{pv} = 1 \quad \forall v \in \mathscr{V} \tag{2.2}$$

$$x_{pv} = 0 \quad \forall v \in \mathscr{V}, p \notin P_v(t_i) \tag{2.3}$$

$$y_r + \sum_{p \in P(t_i)} \sum_{v \in \mathscr{V}} a_{rp} x_{pv} = 1 \quad \forall r \in R(t_i) \tag{2.4}$$

$$y_r = 1 \quad \forall r \in R^E(t_i) \tag{2.5}$$

$$\sum_{p \in P_{v(r)}} a_{rp} x_{pv_r} = 1 \quad \forall r \in R^S(t_i) \tag{2.6}$$

$$p_t \geq p^{target} \tag{2.7}$$

$$x_{pv} \in \{0,1\} \; \forall p \in P(t_i), \; y_r \in \{0,1\} \; \forall r \in R(t_i) \tag{2.8}$$

For the HARD-configuration the process-planning problem is represented by the mathematical optimization model (2.1)-(2.8). The costs for TP_i are minimized (2.1). One route is selected for each vehicle (2.2) and vehicle v is able to execute the selected path p (2.3). Each single request known at time t_i is either served by a selected vehicle or forwarded to the LSP (2.4) but a once subcontracted request cannot be re-inserted into the path of an own vehicle (2.5). An (S)-labeled request cannot be re-assigned to another vehicle or LSP (2.6) and overall, the percentage p^{target} of all requests must be scheduled in time (2.7).

The model (2.1)-(2.8) is NP-hard to solve since it represents the traveling salesman problem in a specific parameter setting.

$$\sum_{p \in P(t_i)} \sum_{v \in \mathcal{V}} \left(C^1(p) + C^2(p) \right) x_{pv} + \sum_{r \in R(t_i)} C^3(r) y_r \rightarrow \min \tag{2.9}$$

In the PEN-configuration the punctuality constraint (2.7) is skipped and the objective function (2.1) is replaced by the evaluation function (2.9) that incorporates the penalty payments $C^2(p)$ for lateness. Penalties associated with p are summed up to $C^2(p)$ from all late customer site visits according to p. If the request is performed in time, the penalty is zero for the associated single customer site; it increases proportionally up to 25 monetary units for a delay of 100 time units. Further delays do not lead to additional charges. According to (2.9) each late arrival at a customer's site is penalized independent of the fulfillment of $p_t \geq p^{target}$. In so doing, we have a unique quantification of the extent of lateness of the scheduled requests. We cannot uniquely identify those late requests (among all late requests) that finally cause the decrease of p_t below p^{target}. Therefore, we use the "more strict" penalization scheme coded into (2.9).

2.3.4 Generation of Parameterized Test Cases

In order to evaluate and compare the impacts of the different SC-configurations in computational simulation experiments, we have derived a set of artificial test instances. Each instance is defined by a special instantiation of a set of parameters. Different scenarios can be modeled by adjusting these parameters.

Two different kinds of dynamic routing scenarios are referred to in the literature. In the first scenario type, the workload remains stable over a specific time interval. This means that it is possible to adapt the available resources in advance so that all additional requests can be served appropriately with own vehicles. For this reason, such a scenario is called a *balanced scenario*. In the event that the workload varies over the simulation interval, the scenario is denoted as a *peak scenario*. Here, it is hardly possible to adapt the available resources in advance.

Lackner (2004), Mitrović-Minić et al. (2004) as well as Pankratz (2002) propose artificial benchmark instances for evaluating different dispatching strategies. In all these instances the number of additional requests for a given time interval remains equal, constituting balanced scenarios.

Gutenschwager et al. (2004) and Sandvoß (2004) use real world data sets for their experiments in which the intensity of incoming demands varies over time. Such instances represent examples of a peak scenario. Neither a parameterization nor a classification of these instances is possible.

To simulate and parameterize peak scenarios we first generate a balanced stream of incoming customer demands over the complete observation time period. A second stream is generated for a part of the observation period. Both streams are then overlaid so that during the period in which the second stream is alive, the balanced stream is replaced by a higher number of requests (a peak) which must be integrated into the existing transportation plans.

We start with the generation of the balanced stream of incoming customer demands over the complete observation period $[0;T_{max}]$. Therefore, at time $t_{rel} = 0$ n_0 requests are drawn randomly from request set P. The set P is a Solomon instance (Solomon, 1987) for the vehicle routing problem with time windows comprising 100 customer requests. Next, the request release time is updated to $t_{rel} := t_{rel} + \Delta t$ and for this new release time, n_0 customer requests are drawn from P at random again. However, for each of the recently selected requests r, the release time is set to t_{rel}. The original service time window $[e_r;l_r]$ of r is replaced by $[t_{rel}+e_r;t_{rel}+l_r]$. Then, t_{rel} is increased by Δt again and additional requests are generated similarly as long as $t_{rel} \leq T_{max}$. A second stream of requests is generated in order to achieve a demand peak. Again, we iteratively increase the release time t_{rel} by Δt starting at $t_{rel} = 0$. As long as $t_{rel} \leq t_{start}^{peak}$ is met no additional demand occurs. In the event that $t_{rel} \in [t_{start}^{peak};t_{start}^{peak} + d_{peak}]$, n_1 additional requests, drawn randomly from P, are selected to be released at t_{rel}. Again, the original service time window is shifted by t_{rel}. No requests are specified anymore within the second stream as soon as $t_{rel} > t_{start}^{peak} + d_{peak}$. Both streams are then overlaid so that during the period $[t_{start}^{peak};t_{start}^{peak} + d_{peak}]$ an increased number $n_0 + n_1$ of requests appear.

All vehicles specified within the instance P can be used. In order to determine a competitive and comparable tariff for calculating the LSP fare F_r associated with request r, we desist from capacity constraints and set the capacity usage of each request to zero. We multiply the Euclidian distance d_r between the depot of the LSPs, situated at location $(65,65)$, and the customer site associated with r with a normalizing factor ν_r. A subcontracting of r yields costs of $F_r := d_r \cdot \nu_r$ monetary units. We consult the best-known solution $\mathscr{S}(P)$ of P found in the literature in order to calculate ν_r. The vehicle v_r serves r according to $\mathscr{S}(P)$ and $l^{demanded}$ denotes the sum of the Euclidian distances (the demanded distances) between the depot and the customer sites of all requests served by v_r in this solution proposal. The normalizing factor assigned to request r is now set to $v_r := \alpha \cdot \frac{l^{demanded}}{l^{travelled}}$, where $l^{travelled}$ denotes the route length of vehicle v_r (Schönberger, 2005). Scenarios with different tariff levels can be generated by modifying the factor α. If $\alpha << 1$ then subcontracting is cheaper than self-fulfillment, in the event that $\alpha \approx 1$ both fulfillment modes have comparable costs but if $\alpha >> 1$ then the self completion-mode is cheaper.

Each scenario is described by the 5-tuple $(P, d_{peak}, n_0, n_1, \alpha)$. In this investigation, we use the four Solomon cases $P \in \{R103, R104, R107, R108\}$ to generate request sets with tariff levels $\alpha \in \{1, 1.25, 1.5, 1.75, 2, 3\}$. Furthermore, it is $n_0 = 50$, $n_1 = 100$ and $\Delta t = 100$ time units. The peak duration is fixed to $d_{peak} = 200$ time units starting at $t^{peak}_{start} = 1500$ time units. Finally, the total observation period lasts $T_{max} = 5000$ time units.

2.4 Conclusions

We have introduced online optimization models as a suitable formalization approach of dynamic transport process planning problems in a supply consortium. The involved decision making entities (coordinator and transport service agent) have to solve specific and interrelated online decision models in order to generate transport processes for fulfilling customer demand. Although the process planning goals of the coordinator agent and the transport service agent are partly contradicting, they have to interact in the process planning phase.

As a second major challenge in online transport process planning, the management of demand peaks has been identified. An appropriate approach to cope with significantly varying transport need is the subcontracting of parts of the maintained request pool. Depending on the amount of the costs for subcontracting a request, the goals of the two decision making units comply (low subcontracting tariffs) or they are contradicting (high subcontracting tariffs).

For preparing computational simulation experiments of the interaction between the coordinator and the service agent, we have introduced a special dynamic transportation planning scenario. Artificial test instances are proposed in which we can scale the source of dynamic as well as the subcontracting costs as the most important parameters for determining the decision behavior of the transport service agent.

Chapter 3
Decision Support: Applying the State-of-the-Art

Decision Support Systems (DSS) are collections of interacting computer systems and software tools that enable the automatic or semiautomatic derivation of solution proposals from a given decision model. A DSS accesses one or more data sources to collect all planning data predicted by the maintained decision model and derives solution proposals that take into account planning objectives as well as planning requirements. We focus on the solution derivation process with particular interest in the consideration of consecutively and unpredictably arriving planning data (e.g. transportation requests).

In this chapter, we report the development of "state-of-the-art" techniques for DSS to enable the automatic transportation plan generation and the automatic plan updating in a volatile environment. These techniques can be found in most of the papers reporting on DSS for dynamic transportation planning scenarios. They are adapted from static decision situations where they exhibit a very high efficiency and effectivity. We apply these standard techniques to the online optimization model of the dynamic vehicle routing problem introduced in the previous chapter.

The organization of this chapter is as follows. Section 3.1 presents an overview of DSS for dynamic integrated transport disposition and dispatching problems that occur in supply consortia. We explain how the generation of transportation plans is supported by computer systems and how dynamically appearing events are integrated into the already started transport processes. The setup of a special DSS for the online vehicle routing problem introduced in the previous chapter is described in Section 3.2. In particular, we develop a planning framework that is able to consider the consecutively updated planning data and, in parallel, aims to meet the given planning restrictions. A systematic and comprehensive assessment of the proposed DSS is performed. We describe the layout of the executed simulation experiments, define appropriate performance indicators and report about the observed values for these indicators in Section 3.3.

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*,
Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_3,

3.1 Decision Support Systems and Transport Process Planning

A Decision Support System (DSS) is an information processing system especially dedicated to derive or support the derivation of goal-oriented decisions in complex decision situations. Such a system combines operational data with analytical decision models in order to enable a computer-based control of value creation processes (Laudon and Laudon, 2006).

According to Sprague and Carlson (1982), a DSS consists of three parts: the data base management system (DBMS), the model-base management system (MBMS) and the dialog generation and management system (DGMS). All data previously collected and/or necessary for deriving appropriate decisions are managed by the DBMS. The MBMS hosts formalized representations of the decision tasks (decision models) to be used for deriving decisions in a specific data setting. Tools for enabling the interaction of the DSS with its users are contained and managed by the DGMS.

We explain in Subsection 3.1.1 how a DSS is used to control value creation processes in dynamic environments. In Subsection 3.1.2 we survey requirements and previous work related to DSS-based control of transport processes. The handling of events in dependence of their significance is subject of Subsection 3.1.3. Requirements for the management of requests in dynamic transportation planning are summarized in Subsection 3.1.4.

3.1.1 Process Control by Decision Support Systems

The necessity for a repeated revision of process-related decisions and the update of process instructions has been illustrated in the previous chapter. Fig. 3.1 outlines the usage of a DSS for contributing to a process revision. The gray-shaded area represents the environment in which a previously generated process is running. A process revision cycle is initiated by an event occurring in the environment and disturbing the planned execution of the current process instance (1). Immediately after the detection of the disturbing event, an update of the planning data hosted in the DBMS is triggered (2) and the necessary data revisions are established (3). After the necessity of a process revision has been determined by an analysis of the altered data, the MBMS component of the DSS is requested to instantiate a new suitable decision model (4). The MBMS selects an appropriate decision model type and parameterizes it using the stored planning data (5). The complete decision model is forwarded to a decision making algorithm and one or more proposals for an update of the process are derived using computational methods from Operations Research and/or Artificial Intelligence (6). These proposals are handed over to the DGMS that prepares the presentation of the proposals towards the responsible decision making agent (7). The responsible agent selects one proposal (8) that is then implemented and executed (9) until an additional process disturbing event is detected.

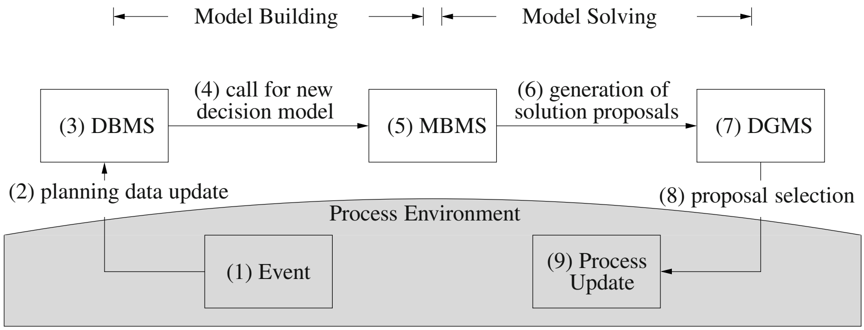

Fig. 3.1 Process-Control by a Decision Support System

The steps (1) to (5) in Fig. 3.1 form the model building phase in a process update cycle. At the end of the model building phase, a formalized representation (the decision model) of the current decision task is available. As the second phase of an update cycle, the steps (5)-(9) are grouped to the model solving phase in which an implementable solution of the set up decision model is identified, selected and implemented. The MBMS (5) connects the two phases. Thus, the management of the formal decision model plays a central role for the process control.

3.1.2 DSS-based Control of Transport Process

Significant effort has been spent to improve and accelerate the process update cycle. For the model building phase, remarkable progress in the detection of process disruption events and the collecting of planning-relevant data has been achieved in the last two decades. The development of small-size low-cost data acquisition (Wang et al., 2006; Suhong et al., 2006; Michael and McCathie, 2005) and communication devices (Timm-Giel et al., 2007; Yoshimoto and Nemoto, 2005; Erkens et al., 2005; Anderson et al., 1996), wide-spread traffic information systems (Ehmke et al., 2008; Ferman et al., 2005; Thill, 2000; Claramunt et al., 2000) and mobile communication networks (Basagni et al., 2004) enables a transport planning unit to access necessary planning data in real-time in a very highly detailed fashion. Almost all necessary planning data are available for exploitation or can be accessed without significant effort. Consequently, the automatic detection of process disturbing events and the automatic processing of associated information into the maintained decision model are start-of-the-art and part of any DSS for managing transport processes.

Progress in the model solving phase is caused by innovations in computer technology providing faster calculation machines (Raney and Nagel, 2004; Nordhaus, 2003; Cordeau et al., 2002). From operations research, efficient analytical algorithms have been contributed to accelerate the optimization of process revisions

(Koberstein, 2008; Burkhard et al., 2000). The transfer of tentative search paradigms like evolutionary search (Bäck et al., 2000a,b), tabu search (Glover and Laguna, 1997) or ant-colony-algorithms (Dorigo and Stützle, 2004) for exploring large sets of process update alternatives is due to contributions from artificial intelligence research. From communication engineering, quick, cheap and easy-to-use information and communication technologies have been developed (Giannopoulos, 2004) enabling a dispatching unit to interchange process update information with drivers and customers efficiently. Last but not least, appropriate decision models have been proposed by management scientists covering not only the pure transport but integrating transport with other value creating phases. Examples are operational freight carrier planning problems (Krajewska, 2008; Schönberger, 2005; Pankratz, 2002), inventory routing problems (Sindhuchao et al., 2005; Campbell et al., 1998) or just-in-time-production requirements (Chuah and Yingling, 2005; Vaidyanathan et al., 1999).

Each planning system designed for supporting the management of transport processes in a dynamic environment is built up by several interacting components. Each component might be hardware (computers or on-board units) or software (databases, algorithms). The latter components are typically represented by software agents that receive, process and generate information. Thus, the specification of a multi-agent system (Woolridge, 2002) for supporting the dynamic dispatching in the investigated scenario is promising.

All disposition and dispatching systems have in common that the incorporated agents require huge amounts of data for making their decisions. Therefore, a stable and reliable connection to different databases or data-warehouses is a prerequisite for the system's emergence (Zeimpekis and Giaglis, 2005; Slater, 2002). Examples for those data sources are traffic management centers providing latest traffic flow data (Ehmke et al., 2008; Fleischmann et al., 2004b), request databases and request specification interfaces (Slater, 2002) as well as map engines that simulate territory maps (Zeimpekis and Giaglis, 2005; Slater, 2002). In addition, real-time information is fetched from satellite-based Global Position Systems (GPS) as well as tracking and tracing systems (Slater, 2002). Furthermore, the agents must be prepared to process all data in real-time (Powell and Cheung, 2000; Séguin et al., 1997; Brown et al., 1987) in order to provide the physical units with recent process updates immediately. Finally, a human-computer interface has to be provided that enables human users to intervene in the computational planning process (control center user interface, Zeimpekis and Giaglis (2005)). In *interactive systems* one or several solutions of the decision model instance are presented to a human decision maker. This person modifies the proposals and selects the proposal to be implemented (Geiger and Wenger, 2007; Scott et al., 2002; Kopfer and Schönberger, 2002; Waters, 1984). In *automatic updating systems* the set up decision model is solved by a solving method. According to specified preferences (least costs, highest punctuality, ...) one solution of the model is automatically selected and implemented, e.g. Gutenschwager et al. (2004).

Two striking facts are extracted from the literature about the model building subsystem of DSS for the operational management of transport processes in dynamic environments:

- Only one decision model type is maintained in the DSS. The type depends upon the kind of requests to be served, the number of depots and the kind of transport resources.
- There are fixed rules that describe how a new decision model instance is parameterized with the updated planning data. These rules remain unchanged throughout the complete running time of the DSS. They are not changed over time or in response to an event.

3.1.3 Event-Handling in DSS for Dynamic Transport Dispatching

Séguin et al. (1997) propose a generic three-layer architecture for DSS used to handle dynamically emerging events (cf. Fig. 3.2). Adjacent layers communicate by exchanging messages. If a layer is not able to handle an event, it informs the next higher layer and asks for support from there.

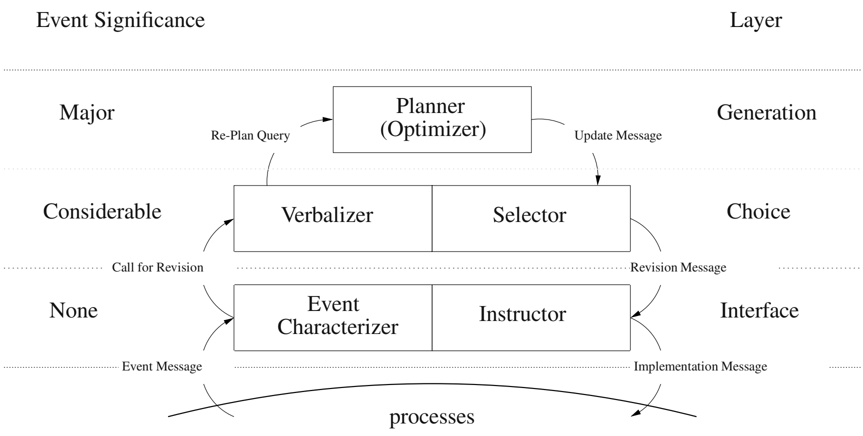

Fig. 3.2 Three-layer model of a dispatching system

The lowest layer is the *interface layer* which hosts the *event characterizer* and the *instructor*. If an unexpected event is detected then an **event message** is sent to the event characterizer that checks whether the processes must be revised or not. In the latter case the instructor sends an **all-clear implementation message** back to the field teams, who are currently waiting for an answer from the DSS. A typical event handled by the interface layer is the completion of a request. In the former case, the event leads to a process corruption and a **call for revision** of the process is sent to the next layer (*choice layer*).

In the choice layer, the *verbalizer* receives the call and analyzes the process corruption caused by an event and the *selector* tries to remedy the corruption. If it succeeds then a **revision message** is given down to the interface layer that broadcasts the relevant Implementation Message to the vehicles. A typical example for an event is the occurrence of traffic congestion that endangers the compliance of announced arrival times without shifting the arrival time at a customer site out of the time window. If the corrupting event causes major corruption that cannot be solved in the second layer then a global re-planning is triggered by sending a **re-plan query** from the choice layer into the *generation layer*.

The generation layer only hosts the *planner* (optimizer) which derives process update proposals. An additionally arriving request is an example for an event that requires the call of the planner. These proposals are fed back in an **update message** to the selector in the choice layer.

In the event that that an event cannot be handled by the choice layer the process revision cycle (1)-(9) given in Fig. 3.1 starts. A new decision model is instantiated and solved. If the event is handled by the choice layer then the application of local process revisions leads to a process recovery.

3.1.4 Challenges in Dynamic Disposition and Dispatching

The repeated update of transport processes comes along with several special requirements that decision support tools have to address.

It is important that once made decisions can be protected, e.g. the revision of such a decision is blocked. In dynamic transportation disposition and dispatching not all decisions can boundlessly revised. The reliability of announcements towards customers concerning arrival times are an important and distinguishing service aspect so that the associated decisions should be left unchanged during process revisions (Erera et al., 2008; Madsen et al., 1995). However, some decisions are even impossible to be revised at all, like the decision to exercise an LSP-option (Schönberger and Kopfer, 2007b).

The urgency of the requests in the portfolio R_t has to be considered. Requests whose completion is necessary in the near future have to be treated preferentially compared to requests whose completion is necessary far away from the current time (Mitrović-Minić et al., 2004). If not all urgent requests can be handled immediately then it is necessary to manage the resulting waiting list or queues efficiently (Fleischmann et al., 2004b). In case those late customer site arrivals cannot be prevented, delays have to be handled so that the negative impacts of unsatisfied customer requirements are minimized (Fleischmann et al., 2004b).

There is a time gap between the decision how a request is executed and the realization (Lund et al., 1996). From the decision time until the realization time the value of a decision might vary (Albers et al., 2001), e.g. the appropriateness of a decision can be changed so that decision revisions become necessary.

Each decision made at time t_i influences the number of decision alternatives and their evaluation for later re-planning times $t_{i+1}, t_{i+2}, \dots$ (Saéz et al., 2008). The anticipation of future system states is necessary (Branke and Mattfeld, 2005).

In dynamic decision making, each solution update follows one or more given global planning objectives that guide the decision making over the consecutive update stages. Lund et al. (1996) propose to deviate temporarily from the followed superior decision strategies by varying objectives and decision alternatives in order to heal negative impacts of extraordinary events more rapidly.

3.1.5 Rolling Horizon Disposition and Dispatching

The basic idea of rolling horizon planning is to generate a sequence of plans $S_0, S_1, \dots$ (S_i is a solution of the decision model M_i of problem instance P_i). At time t_i the plan S_i is derived and its realization is initiated. It is continued with the execution of S_i until, at time t_{i+1}, additional data is revealed. The plan S_i is replaced by S_{i+1} and S_{i+1} is executed until it is corrupted at time t_{i+2} by events providing additional data and so on. The solution update is carried out if a pre-specified time point is reached (Rajagopalan et al., 2008; Saéz et al., 2008; Erera et al., 2008; Mitrović-Minić et al., 2004) or if one or more certain events take place (Fleischmann et al., 2004b; Lund et al., 1996; Ausiello et al., 1994). In the first case, the update of S_i is called *time-triggered* but in the second case the revision is *event-triggered*.

Two general concepts for the update of S_i to S_{i+1} are distinguished. *Rule-based* updating follows the hypothesis that a few basic reasoning rules are valid and that it is possible to inductively reason the behavior in all other cases not explicitly stated in the basic rules (Lindstaedt, 2007). The a-priori-route-concept (Tang and Miller-Hooks, 2007; Liu, 2007) is an example of rule-based updating. Additionally, update rules like MST-algorithms (Ausiello et al., 1994) or cheapest insertion approaches (Fleischmann et al., 2004b) are representative examples for rule-based reasoning.

A deductive reasoning is carried out in *model-based* update. Here, the set of all possible update alternatives is implicitly described by a formalized problem description (the decision model) and a structured scanning of the set of alternatives leads to the desired solution. Examples of model-based approaches include the linear-programming based optimization of a traveling salesman's route and the solving of a mixed-integer linear programming model of the capacitated vehicle routing problem. Dynamic pickup and delivery problems with model-based schedule update are investigated (among others) in (Saéz et al., 2008; Mitrović-Minić et al., 2004; Lund et al., 1996)

Re-planning approaches that only consider the currently and certainly known planning data without incorporating expected data are called *myopic*. They are based on the assumption that each forecast is inappropriate because the future events that corrupt the execution of once generated processes are unpredictable (Huth and

Mattfeld, 2008; Fleischmann et al., 2004b). In contrast, re-planning approaches with anticipation exploit forecasts (Saéz et al., 2008; Powell, 1996). In such a situation, it is assumed that process decisions made according to the forecast are likely to be executed as planned.

3.2 Online Planning in HARD- and PEN-Configurations

This section contains the description of a decision support system for managing the transport processes in the scenario introduced in Section 2.3 (cf. page 27). The developed DSS framework is configured for both setups HARD and PEN proposed for integrating the planning goals of the coordinator agent and the fleet managing agent. The development of the DSS is guided by the following three assumptions.

Model Building: As events we only consider additionally arriving requests, so that each event (arriving request) causes a call of the third-layer of the event-handling scheme given in Fig. 3.2. Thus, each event triggers an execution of the complete process update cycle (1)-(9) shown in Fig. 3.1. The model building rules are static,; they are not changed during the simulation experiments.

Model-Base: Only the decision model (2.1)-(2.8) (are the HARD-configuration) and the model (2.9), (2.8) (2.2)-(2.6) for the PEN-configuration are maintained.

Model Solving: Only one solution of the current instance of the maintained decision model is generated. We want to have an automatic process update. The generated process update is immediately implemented without modifications.

In Subsection 3.2.1, we describe the configuration of a hybrid search algorithm for solving the decision model instances $M_0, M_1, \ldots$. Subsection 3.2.2 compares techniques for ensuring the consideration of the constraints (2.2)-(2.7) during the solving of M_i.

3.2.1 Memetic Algorithm for the Transportation Plan Generation

We use a Memetic Algorithm (MA) realizing a hybrid search strategy (Grosan et al., 2007; Sarker et al., 2002) consisting of a global genetic search space sampling (Eshelman, 2000) and a local 2-opt improvement procedure (Croes, 1958) for solving the scheduling model instances $M_0, M_1, \ldots$ of the online decision model introduced in 2.3.3.

The genetic search uses a $\mu + \lambda$-population model (Grefenstette, 2000) evolved by the application of the PPSX-crossover-operator (Schönberger and Kopfer, 2003) and a mutation operator that

1. arbitrarily switches fulfillment modes of requests
2. shifts requests between selected routes of own vehicles and
3. reverses the visiting order of randomly chosen subsequences of arbitrarily selected routes.

The construction of the initial population is generated using the Push Forward Insertion Heuristic (Solomon, 1987). One half of the initial set of solution proposals is generated by deploying the heuristic followed by some random proposal modifications, and the other half is generated purely at random without applying any biasing procedure. The evolution process is stopped dynamically if the average fitness of the evolved population does not improve for 10 generations.

Every time a new decision model instance M_i has arrived, the MA is re-started to solve the model of the recent instance. Computational experiments, in which parts of the final population of the last instance solved are used to seed the initial population of the recent instance failed because this initial population leads to rapid convergence on a bad level even if the crossover and mutation probability are determined adaptively. An analysis of the population development has shown that the significantly changed decision situation requires the re-initialization of the genetic material so that the new decision aspects are considered explicitly. For this reason, a complete new initial population is formed using the seeding approach described above.

3.2.2 Constraint Handling Techniques

Solutions of the models introduced in Subsection 2.3.3 are defined as sets of decision variables, instantiated by a value taken from their associated domains. Local search (improvement) algorithms generate an initial instantiation of the decision variables. Then, they apply one or more search operators and generate an offspring solution from one or more existing solutions. The sequence of generated solutions is called the search trajectory and the algorithm generates one (e.g. Tabu Search) or several trajectories in parallel (e.g. Genetic Algorithms). It is necessary that the generated solutions comply with the constraints (2.2)-(2.8) (feasibility). From a certain solution on, all further solutions within the search trajectory have to comply with the constraints specified in the model. Three different approaches enforcing the search trajectory to stay in the set of feasible solutions are presented below (Schönberger et al., 2004)

Selection of an Adequate Solution Representation and of Suitable Operators. The first idea is to design a problem representation, which can only represent solution proposals complying with all given constraints. If all operators can only generate solutions within the given representation then all maintained and generated solutions stay compatible with the associated constraints (Michalewicz, 2000a).

We use a direct problem route-based representation (Schönberger, 2005), which ensures that no violations of the constraints (2.2)-(2.6), (2.8) occur in the maintained set of solutions.

Repairing Constraint Violations. Local hill-climbers are incorporated into the superior memetic search algorithm. They repair constraint violations by modifying the generated offspring solutions and transform them to the nearest solution that is feasible with respect to the given constraints (Michalewicz, 2000b).

The application of the MA to the HARD-configuration requires the call of a repair procedure for each generated offspring solution if the percentage of in-time arrivals is smaller than p^{target}. This means, the repair procedure is invoked if and only if constraint (2.7) is violated. In this case the procedure REPAIR() shown in Fig. 3.3 is executed for each offspring solution.

1. All requests r associated to a too-late customer site arrival are collected in the set S_1.
2. For each request $r \in S_1$ the savings s_1 are calculated. Here, s_1 is defined as the difference between the travel cost savings and the subcontracting costs F_r.
3. If S_1 is empty or if the current punctuality rate p_{t_i} is at least p^{target} then goto (5). Otherwise, the requests contained in S_1 are sorted, so that the first request in the order has the maximal saving of all requests in S_1, the second request in the order has the second highest saving and so on.
4. Finally, the fulfillment mode of the first request in the sorted list is switched to subcontracting, the request is deleted from the list and the punctuality rate p_{t_i} is updated. Goto (2)
5. The repair has been completed.

Fig. 3.3 procedure REPAIR()

In general (but not in the problem investigated here) it is unclear in advance whether a repair procedure call can completely repair a given solution using the given repair function. For this reason, the repair attempt is a search process itself. Its goal is to identify the nearest solution, which complies with the given constraints. The computational effort for this (maybe unsuccessful) search is often quite high.

Penalization of Constraint Violations. Solutions that propose late customer site arrivals are penalized by depreciating their evaluation value (Smith and Coit, 2000). The penalization lowers the attractiveness of such an individual and decreases the individual chance of being selected for reproduction. It is expected that, in the long run, penalized individuals will not be used any more so that at the end, only solutions without any constraint violations are found in the search trajectories.

In the HARD-configuration for the supply consortium we repair infeasibilities of the least-punctuality constraint (2.7) by applying the procedure REPAIR() to the defective solution proposal. If the consortium is PEN-configured then a penalty scheme is incorporated for devaluating defective solution proposals in order to enforce the observance of the least-punctuality requirement.

3.3 Simulation Experiments Report

This section reports about the setup and the results of computational simulation experiments to assess the proposed DSS. The simulated scenarios are outlined in Subsection 3.3.1. All indicators used to describe the performance of the DSS are introduced in Subsection 3.3.2. The observed values for these indicators are presented and analyzed in Subsection 3.3.3. Impacts of penalty function parameter variations are presented and discussed in Subsection 3.3.4.

3.3.1 Simulated Scenarios

The HARD-configuration of the supply net setting deploys the decision model (2.1)-(2.8) and the MA incorporating the procedure REPAIR(). Similarly, the PEN-configuration uses the decision model (2.2)-(2.6), (2.8), (2.9) and the MA to derive new transportation plans incorporating the penalty scheme given on page 32.

In order to assess the performance of the two supply chain configurations, we perform several simulation experiments using the artificial test cases introduced in 2.3.4. The target punctuality p^{target} is set to 0.8 (80% of all requests have to be completed within the customer-specified time windows).

A **scenario** (α, P, exp, ω) is determined by applying a planning system setting (exp, ω) to an **incoming stream of requests** (α, P). We have combined each set of requests generated from the Solomon instance $P \in \mathscr{P} := \{R103, R104, R107, R108\}$ with each tariff level $\alpha \in \{1, 1.25, 1.5, 1.75, 2, 3\}$. In doing so, overall $\|\{R103, R104, R107, R108\}\| \cdot \|\{1, 1.25, 1.5, 1.75, 2, 3\}\| = 4 \cdot 6 = 24$ different request situations are set up. To control the allocation of resources for the requests, we deploy both integration approaches HARD and PEN. Since both methods incorporate the randomized MA, we determine different random number generation seedings $\omega \in \{1, 2, 3\}$. This leads to overall $2 \cdot 3 = 6$ **resource allocation strategies**. Each of these strategies is tested once to manage the 24 different request streams, so that in total $6 \cdot 24 = 144$ scenarios (α, P, exp, ω) are simulated. In the remainder of this subsection, we report the observed results.

3.3.2 Performance Indicators

We define several indicators which are recorded throughout the simulation experiments. The first group of indicators aims at representing the process quality in terms of punctuality, waiting requests, exploitation of subcontracting, costs and so on. In contrast, the second group of indicators contains several kinds of costs. Their evaluation enables an analysis of the cost impacts of the two resource allocation strategies for different tariff levels.

3.3.2.1 Process Quality Indicators

The punctuality rate recorded at time t within the scenario (α, P, exp, ω) is denoted as $p_t(\alpha, P, exp, \omega)$. Let $p_t(\alpha, exp) := \frac{1}{12} \sum_{\omega=1}^{3} \sum_{P \in \mathscr{P}} p_t(\alpha, P, exp, \omega)$ denote the average punctuality observed at time t for the parameter combination (α, exp).

In order to study the impact of the demand peak on the punctuality, we calculate the deviation of $p_t(\alpha, P, exp, \omega)$ from the reference value $p_{1000}(\alpha, P, exp, \omega)$ for all times in the observation time interval $[1000, 5000]$ by

$$p_t(\alpha, P, exp, \omega)/p_{1000}(\alpha, P, exp, \omega) - 1. \tag{3.1}$$

The largest past-peak deviation from the reference value is then calculated by $\frac{\min_{t\geq 1500}\{p_t(\alpha,P,exp,\omega)\}}{p_{1000}(\alpha,P,exp,\omega)} - 1$. Now, the average $\delta(\alpha, exp)$ of the largest observed deviation from the reference values for the parameter combination (α, exp) is defined as shown in (3.2).

$$\delta(\alpha, exp) := \frac{1}{12} \sum_{\omega=1}^{3} \sum_{P \in \mathscr{P}} \left(\frac{\min_{t\geq 1500}\{p_t(\alpha, P, exp, \omega)\}}{p_{1000}(\alpha, P, exp, \omega)} - 1 \right). \tag{3.2}$$

Let $T^{below}_{\alpha,exp}$ denote the first time in which $p_t(\alpha, exp)$ falls below p^{target} and $T^{heal}_{\alpha,exp} := \min\{t \in [1000, 5000] \mid \nexists\, l \in [t, 5000], p_l < p^{target}\}$ referring to the time in which an HQ state is finally re-achieved by $p_t(\alpha, exp)$. We define

$$\pi(\alpha, exp) := \frac{T^{heal}_{\alpha,exp} - T^{below}_{\alpha,exp}}{4000} \tag{3.3}$$

as the percentage of LQ periods within the observation interval $[1000, 5000]$.

Throughout the simulation time, we have recorded the percentage of subcontracted requests in $q_t(\alpha, P, exp, \omega)$. These values have been summarized in

$$q_t(\alpha, exp) := \frac{1}{12} \sum_{\omega=1}^{3} \sum_{P \in \mathscr{P}} q_t(\alpha, P, exp, \omega) \tag{3.4}$$

for each setting (α, exp). The maximally observed subcontracting rate is defined by $\sigma(\alpha, exp)$, calculated and calculated according to (3.5). It indicates the exploitation of the subcontractor fulfillment mode.

$$\sigma(\alpha, exp) := \max_{t \geq 1500} q_t(\alpha, exp).$$

In order to get information about requests still uncompleted at time t, we fetch the number $w_t(\alpha, P, exp, \omega)$ of already scheduled but not fulfilled requests (*pending requests*) during the simulation of scenario (α, P, exp, ω). We calculate the averagely observed number $w_t(\alpha, exp)$ of requests pending at time t for each combination of tariff level α and consortium configuration exp in the same way as done for $p_t(\alpha, exp)$.

3.3.2.2 Financial Process Evaluation Indicators

We trace the costs for the request fulfillment. In particular, we observe the travel costs $c_t^{travel}(\alpha,P,exp,\omega)$ of the own fleet, the costs $c_t^{sub}(\alpha,P,exp,\omega)$ of LSP incorporation and the penalty payments $c_t^{pen}(\alpha,P,exp,\omega)$ for arrivals later than the agreed time windows. The cumulated request fulfillment costs $c_t(\alpha,P,exp,\omega)$ at time t are the sum of $c_t^{travel}(\alpha,P,exp,\omega)$, $c_t^{sub}(\alpha,P,exp,\omega)$ and $c_t^{pen}(\alpha,P,exp,\omega)$. All four cost indicators are averaged in $c_t(\alpha,exp)$, $c_t^{travel}(\alpha,exp)$, $c_t^{sub}(\alpha,exp)$ and $c_t^{pen}(\alpha,exp)$.

Let $\bar{c}_t^{travel}(\alpha,exp) := \frac{c_t^{travel}(\alpha,exp)}{c_t(\alpha,exp)}$ be the percentage of the overall costs caused by the travel expenses of the own fleet. We similarly define $\bar{c}_t^{sub}(\alpha,exp)$ and let $\bar{c}_t^{pen}(\alpha,exp)$ be the percentage of the overall costs caused by LSP charges and penalty payments.

In order to compare PEN and HARD with respect to the resulting request fulfillment costs, we calculate the deviation $\gamma_t(\alpha,PEN) := \frac{c_t(\alpha,PEN)}{c_t(\alpha,HARD)} - 1$ of the costs from the reference value observed in the HARD experiment. Similarly, we calculate the deviations $\gamma_t^{travel}(\alpha,PEN)$, $\gamma_t^{sub}(\alpha,PEN)$ and $\gamma_t^{pen}(\alpha,PEN)$ for the three kinds of costs distinguished just before.

Finally, we approximate the marginal costs $mc_t(\alpha,exp)$ of a request for different combinations of tariff level α and consortium configuration exp at time t. Therefore, we first determine the increase of the cumulated costs $c_t(\alpha,exp) - c_{t-100}(\alpha,exp)$ compared to the last re-planning time $t-100$. Secondly, we determine the number of requests $\mid R^C_{(t-100,t)}(\alpha,exp)\mid$ completed in the period between time $t-100$ and t. The marginal cost indicator $mc_t(\alpha,exp)$ is then defined in (3.5). This value indicates the costs for fulfilling one additional request.

$$mc_t(\alpha,exp) := \frac{c_t(\alpha,exp) - c_{t-100}(\alpha,exp)}{\mid R^C_{(t-100,t)}(\alpha,exp)\mid} \tag{3.5}$$

3.3.3 *Computational Assessment of HARD and PEN*

We report about the observed indicator values in this subsection. The process-quality and the financial indicators are analyzed consecutively. For both types of indicators, we perform an *online*- as well as an *offline*-assessment. In the first mentioned type of evaluation, we observe the development of the indicators during the simulation runs, e.g., we present and evaluation the values for every time point t of the observation period. The offline-analysis is an ex-post analysis, e.g., we evaluate the collected indicator values after a simulation has been completely executed.

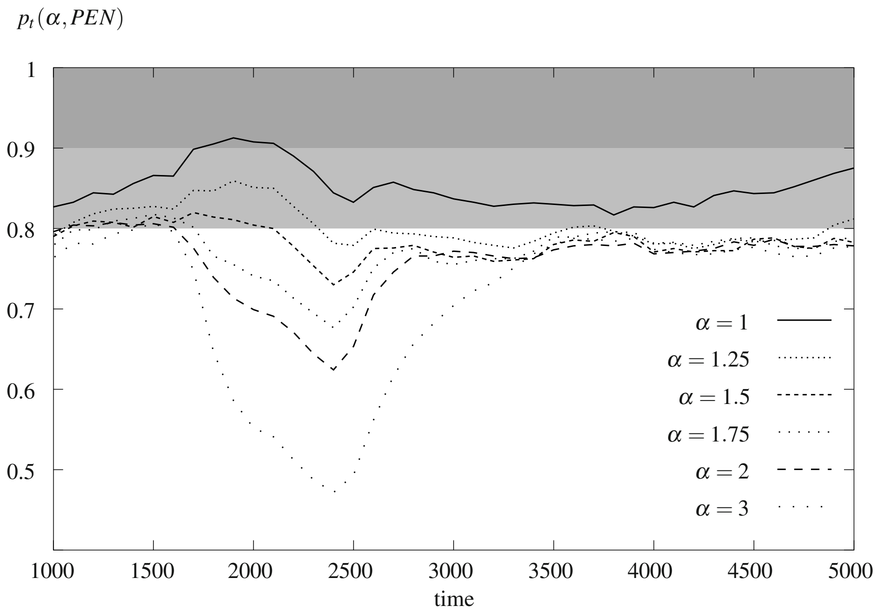

Fig. 3.4 Development of the punctuality $p_t(\alpha, PEN)$

3.3.3.1 Online-Process Quality Assessment

The PEN-configuration can guarantee 80% punctuality only for comparable tariff levels ($\alpha = 1$) as shown in Fig. 3.4. Just after the demand peak is over ($t \geq 1800$) the punctuality even increases because the routes of the own vehicles are compiled from a larger number of available requests. So, a higher number of matching requests can be found.

As soon as the tariff level α increases, subcontracting becomes more and more unattractive. Its costs become higher than the travel costs for the own vehicles plus penalty payments. The self-fulfillment mode is preferred, although it does not come up with an in-time service. Actually, collapses of p_t down to less than 50% are observed for $\alpha = 3$.

The temporal increase of the number of additional requests entering the considered logistic system from 50 up to 150 for the duration of 200 time units leads to a significant increase of waiting requests. In the event that the LSP tariff level is comparable ($\alpha = 1$) we observe a temporary increase of the number of pending requests (Fig. 3.5). For both strategies HARD as well as PEN, the number of pending requests grows up from around 75 scheduled but not yet finished requests to 200 at time $t = 1800$. But soon after the load peak is over, the number of pending requests re-descends to the "off-peak" level, which is 75. With respects to the pending requests, the load peak has been managed at latest at time $t = 2000$. If the tariff level is quite high ($\alpha = 3$) then the non-consideration of the subcontracting

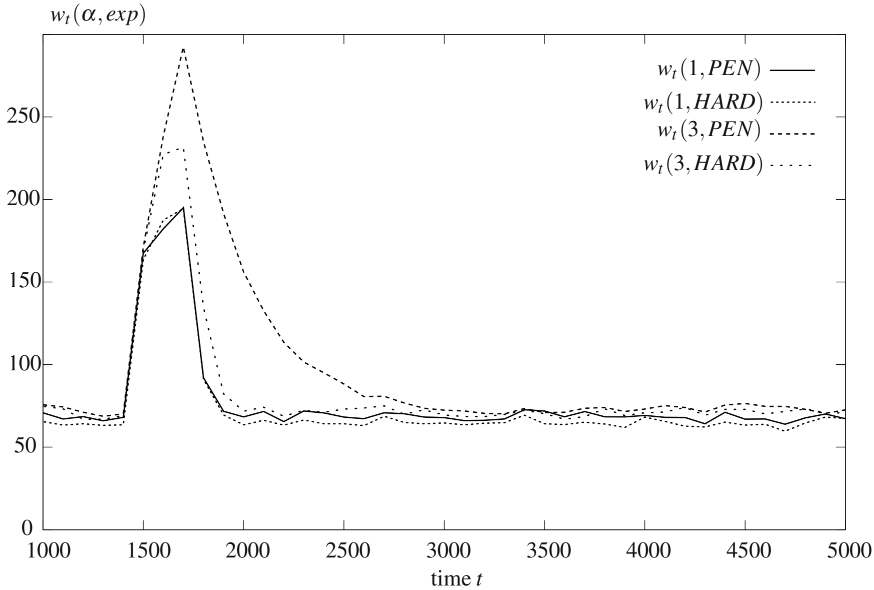

Fig. 3.5 Number of pending requests $w_t(\alpha, exp)$

services causes a blockage of the request fulfillment. While in off-peak situations the average number of pending requests is $w_t(3, PEN) = 70$, this number escalates up to nearly $w_{1800}(3, HARD) = 230$ in the HARD-configuration and even up to $w_{1800}(3, PEN) = 290$ in the PEN-configuration. In the first mentioned configuration, the number of 70 waiting requests is re-achieved at time 2000 but in the latter one the limit of nearly 70 waiting requests is not reached before time 3000. In summary, the PEN strategy is hardly able to overcome the demand peak with respect to the number of pending requests.

Fig. 3.6 shows the number of routed (used) vehicles of the own fleet of the considered transport service agent for both resource allocation strategies HARD and PEN and for the two extreme tariff levels $\alpha = 1$ and $\alpha = 3$. In the scenarios $(1, HARD)$ and $(1, PEN)$ with a comparable LSP tariff, the off-peak number of deployed vehicles from the transport service agent is 3 (HARD) respectively 4 (PEN). Immediately after the initialization of the request peak, these numbers grow up to 10 and re-descend to their original values just after the load peak is over. A completely different system behavior is observed in the scenarios $(3, HARD)$ and $(3, PEN)$. The off-peak number of used own vehicles is 10 for both resource allocation strategies. After the peak start, the number of used own vehicles grows up to the maximal available 25 vehicles independent of the application of HARD and PEN. However, whether the load peak is over the velocity of the reconvalescence of $v_t(\alpha, exp)$ to the off-peak number depends upon the applied resource allocation strategy. In the event that HARD is used, the off-peak number of used own vehicles is re-attained not

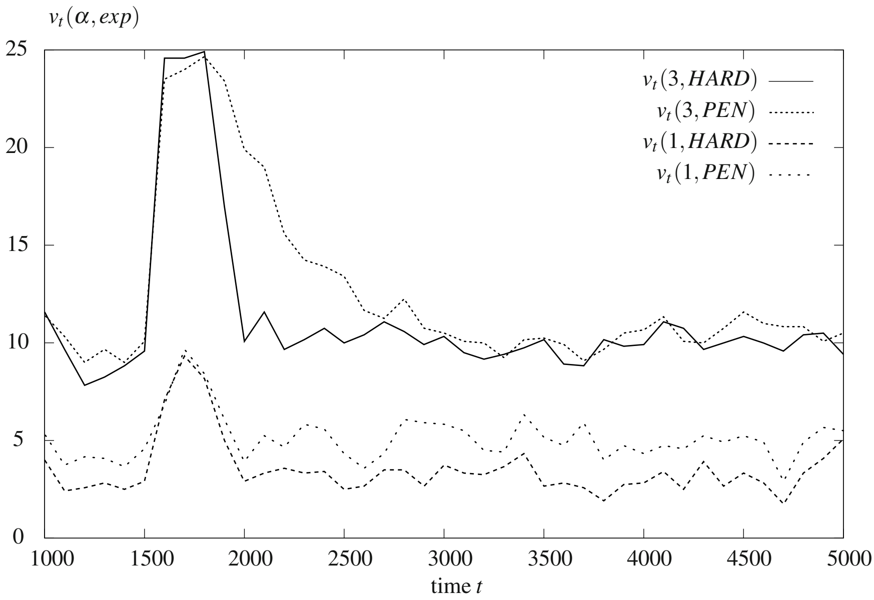

Fig. 3.6 Number of used own vehicles $v_t(\alpha, exp)$

later than $t = 2000$ but if PEN is applied, the regular off-peak number is re-attained not before $t = 3000$. This means that the transport system operates longer out of its "normal" configuration for at least 1000 time units.

The increase of the LSP tariff level from $\alpha = 1$ to $\alpha = 3$ has a significant influence on the proportion of the request portfolio given to an LSP. In Fig. 3.7, the evolution of the percentage $q_t(\alpha, exp)$ of subcontracted requests is shown for both tariff levels. If the LSP tariffs are comparable ($\alpha = 1$) then $q_t(1, PEN)$ and $q_t(1, HARD)$ oscillate between 5% and 22%. Immediately after the start of the load peak, the percentage of subcontracted requests climbs up to a value of around 22%. Here, the subcontracting possibility is used to overcome the load peak. If the LSP tariff is quite high ($\alpha = 3$) then q_t oscillates between 1% and 5%. The higher LSP costs make the exploitation of this request fulfillment mode very unattractive. Even worse, the transport system does not react to the load peak with a significant increase of the percentage of externalized requests. In the HARD-configuration, a small increase of $q_t(3, HARD)$ up to 7% is observed but in the PEN-configuration, the percentage $q_t(3, PEN)$ of externalized requests remains unaffected und stays below 5%.

To conclude our online analysis of the process quality, we state that the PEN-strategy is hardly able to react adequately to the disturbance caused by the load peak if the LSP-tariff is high. Here, the penalization of late arrivals is not able to re-direct the additional requests into the desired fulfillment mode "subcontracting". The primary goal of minimizing the total request fulfillment costs dominates the selection of the fulfillment mode that supports keeping the punctuality rate p_t on a

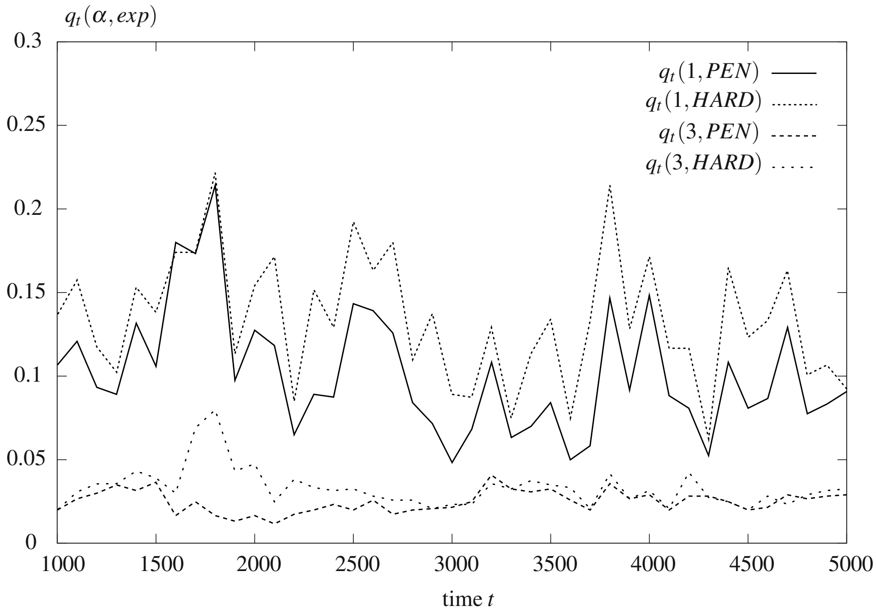

Fig. 3.7 Percentage $q_t(\alpha, exp)$ of externalized requests in the schedule generated at time t

high level. The HARD-strategy ensures that enough requests are served on a timely basis, but as discussed before, the implementation of this strategy is hindered by its unfairness towards the transport service agent.

3.3.3.2 Offline Process Quality Assessment

Overall, the HARD-configuration outperforms the PEN-configuration. We first analyze the maximal decrease of the punctuality rate p_t. The evolution of this parameter is completely different for the two resource allocation strategies. If PEN is used then the maximal decreasing rate of p_t falls from -1.0% ($\alpha = 1$) down to -38.8% ($\alpha = 3$) as can be seen in Tab. 3.1. In contrast, we observe that a tariff level increase hardly influences the punctuality if the consortium is organized according to the rules of HARD. Here, no significant decrease of p_t is detected.

We now turn our attention to the appearance of LQ-situations. Clearly, it is $\pi(\alpha, HARD) = 0$ for all investigated tariff levels α. In addition, the HARD-configuration is able to slightly enlarge the punctuality compared to the referential value at time $t = 1000$ (Tab. 3.1). The additional knapsack-constraint enables the memetic search to evaluate different separations of the request portfolio into self-fulfilled and subcontracted requests. Since the constraint (2.7) ensures that at least 80% of the requests are in time, no penalty costs contradict the route composition.

Table 3.1 Maximal punctuality deviation $\delta(\alpha, exp)$

exp	α					
	1	1.25	1.5	1.75	2	3
HARD	3.6%	-0.0%	5.3%	4.4%	4.2%	3.5%
PEN	-1.0%	-1.8%	-7.8%	-13.6%	-22.0%	-38.8%

Due to the parameter sensitivity of the punctuality $p_t(\alpha, PEN)$ to tariff level increases, the percentage $\pi(\alpha, PEN)$ of LQ-situations increases from $\pi(1, PEN) = 0$ up to $\pi(3, PEN) = 97.5\%$ (Tab. 3.2). Therefore, a short-time demand peak has a long lasting negative impact on the punctuality of the service.

Table 3.2 Portion $\pi(\alpha, PEN)$ of low quality situations

exp	α					
	1	1.25	1.5	1.75	2	3
PEN	0%	60.0%	70.0%	97.5%	82.5%	97.5%

The maximal rate σ of subcontracted requests decreases with increasing tariff level α (Tab. 3.3). For comparable freight tariffs ($\alpha \leq 1.5$) both configurations behave similarly with respect to the subcontracting of requests. The maximal externalization rate $\sigma(\alpha, exp)$ is nearly the same in both cases for each $\alpha \leq 1.5$. However, if the tariff levels climb further then $\sigma(\alpha, HARD)$ remains stable at $\approx 8\%$ while $\sigma(\alpha, PEN)$ falls further down to $\approx 4\%$.

A reason for the bad performance of the PEN-configuration is its non-observance of the reliable subcontracting services if its costs are significantly higher than the sum of self-fulfillment costs and penalty payments. For $\alpha = 1$ the maximal percentage of subcontracted requests is $\sigma(1, PEN) = 21.4\%$ but for high tariff levels, this portion is significantly reduced down to $\sigma(3, PEN) = 4.1\%$ (Tab. 3.3). In the same situation, the HARD resource allocation strategy identifies the externalization as the better mode for nearly double the number of requests (8.0%).

Table 3.3 Maximal externalization rate $\sigma(\alpha, exp)$

exp	α					
	1	1.25	1.5	1.75	2	3
HARD	22.2%	14.9%	10.0%	8.0%	7.2%	8.0%
PEN	21.4%	15.5%	10.0%	5.8%	5.1%	4.1%

Overall, the PEN-configured resource allocation setting is not able to maintain the target punctuality $p^{target} = 0.8$ if the tariff levels for subcontracting requests are lifted. The penalization of delayed arrivals does not ensure the target punctuality in situations with a non-comparable tariff level. The search for least-cost transportation plans ignores the selection of subcontracting requests if the sum of costs for self-fulfillment and lateness is less than the freight charge to be paid to the hired LSPs.

3.3.3.3 Online-Evaluation of the Process Costs

We trace two indicator families to gain an insight into the process cost evolution during the simulation experiments in dependence on the applied consortium configuration and on the applied tariff level. At first, we record the marginal costs $mc_t(\alpha, exp)$ for fulfilling an addition request. Secondly, the contributions of the three cost drivers $\gamma_t^{travel}(\alpha, exp)$, $\gamma_t^{sub}(\alpha, exp)$, $\gamma_t^{pen}(\alpha, exp)$ are observed.

The marginal costs of an additionally completed request mainly depend on the LSP tariff (Fig. 3.8). If the LSP tariff level is 1 then $mc_t(1, exp)$ oscillates around 15 money units even during the load peak. This means, the marginal costs are not affected by this disturbance. A cost-neutral increase of the externalization rate (Fig. 3.7) is possible. However, if the temporary intensification of the subcontracting usage is compromised by costly LSP-charges ($\alpha = 3$) then the load peak does have an impact on the marginal costs. In this context, the first observation is that the off-peak values of $mc_t(3, exp)$ are increased compared to the experiments where we have $\alpha = 1$. They vary from 18 to 20 money units for each additional request. Secondly, we discover that the load peak leads to a temporary increase of $mc_t(3, exp)$ immediately after the peak's instantiation. Finally, the duration and the intensity of the marginal cost increase depend on the applied resource allocation strategy. The application of HARD causes a dramatic but short increase of $mc_t(3, HARD)$. Here, the pre-peak marginal costs are re-achieved immediately after the peak is over. In the event that PEN is used, the increase of $mc_t(3, PEN)$ is less severe but is takes significantly longer until the pre-peak value is re-attained.

3.3.3.4 Offline-Assessment of the Process Costs

The application of both configurations leads to nearly the same costs (Tab. 3.4). There is no empirical evidence that one of the configurations generally produces a larger amount of process realization costs for a given tariff level α. We observe the following variation of the cumulated costs: $\gamma_{5000}(1, PEN) = -8.7\%$, $\gamma_{5000}(1.25, PEN) = -5.5\%$, $\gamma_{5000}(1.5, PEN) = 0.1\%$, $\gamma_{5000}(1.75, PEN) = 4.9\%$, $\gamma_{5000}(2, PEN) = 5.0\%$ and $\gamma_{5000}(3, PEN) = -1.0\%$. The decisions observed in the PEN-configuration yield less costs than for the HARD-configuration if $\alpha = 1$, 1.25, 1.5 or 3. In the remaining cases, the HARD-configured DSS produces a sequence of solutions causing fewer costs than the PEN-configured DSS.

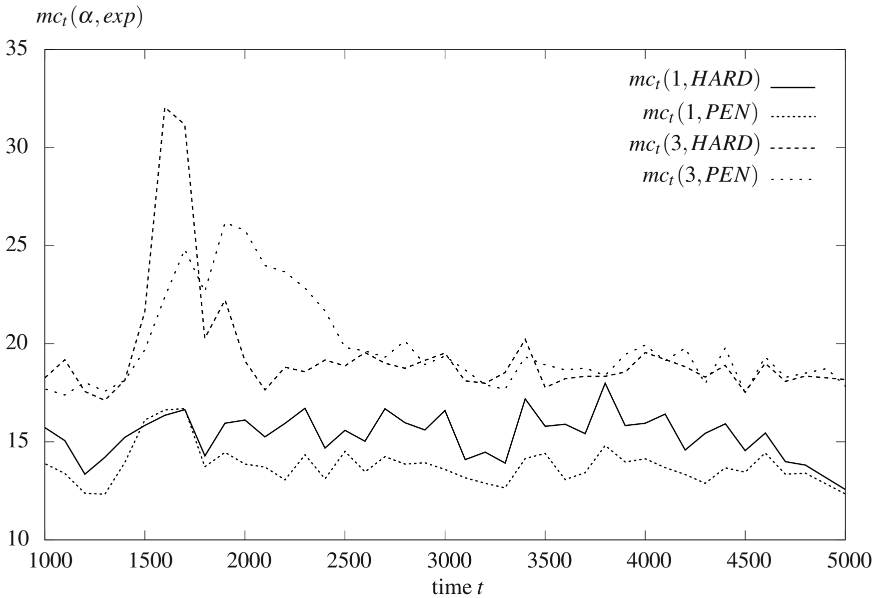

Fig. 3.8 Marginal costs $mc_t(\alpha, exp)$

Table 3.4 Cumulated requests fulfillment costs $c_{5000}(\alpha, exp)$ and deviation $\gamma_{5000}(\alpha, PEN)$

	α					
	1	1.25	1.5	1.75	2	3
$c_{5000}(\alpha, HARD)$	42851.9	43105.1	44283.0	45253.5	47316.1	56301.4
$c_{5000}(\alpha, PEN)$	39115.3	41120.9	44330.6	47483.4	49677.3	55748.4
$\gamma_{5000}(\alpha, PEN)$	-8.7%	-4.6%	0.1%	4.9%	5.0%	-1.0

For both strategies PEN and HARD, the overall sum of request fulfillment costs $c_{5000}(\alpha, exp)$ increases if the tariff level α is lifted. For HARD, tripling the LSP tariff level from $\alpha = 1$ to $\alpha = 3$ results in an increase of $c_{5000}(\alpha, HARD)$ by 31.4% (from 42851.9 to 56301.4). The increase observed in the PEN-experiments is around 42.5%. In order to find out why the two strategies have these different compensatory capabilities, we recall the results of the analysis of the number of pending requests (Fig. 3.5) and of the marginal costs (Fig. 3.8). If PEN is used, then a larger number of requests is fulfilled after the load peak is over (period from t=2000 to t=3000). In this period, PEN comes along with higher marginal costs for each completed requests. Consequently, PEN generates a higher amount of costs than HARD does.

We terminate our cost analysis with an investigation of the cumulated costs $c_{5000}(\alpha, exp)$ for the different configurations HARD and PEN. Tab. 3.5 contains the increase of the costs in the PEN experiments compared to the HARD-experiments.

The travel costs of the transportation plans generated in the PEN-experiments are always higher than the travel costs in the transportation plans generated in the HARD-experiments. As we have already seen in Tab. 3.3, the maximal externalization rate achieved in the PEN-simulations is notably lower than the externalization rate seen in the HARD-simulations. Consequently, a significant number of requests is late in the PEN-simulations leading to a significantly higher amount of penalty payments observed in the PEN-experiments.

Table 3.5 Deviation of costs $\gamma_{5000}^{travel}(\alpha, PEN)$, $\gamma_{5000}^{sub}(\alpha, PEN)$, $\gamma_{5000}^{pen}(\alpha, PEN)$

	α					
	1	1.25	1.5	1.75	2	3
$\gamma_{5000}^{travel}(\alpha, PEN)$	43.6%	31.1%	20.4%	13.8%	8.8%	3.2%
$\gamma_{5000}^{sub}(\alpha, PEN)$	-30.2%	-46.4%	-52.4%	-49.9%	-48.8%	-77.5%
$\gamma_{5000}^{pen}(\alpha, PEN)$	628.4%	5947.3%	364.8%	234.8%	193.9%	232.4%

The analysis of the contribution of the three cost drivers (travel expenses, LSP charges and penalty payments) to the overall costs reveals that a lifting of the LSP tariff results in a shifting of the costs from subcontracting to self-fulfillment (Tab. 3.6). If the LSP charges are comparable ($\alpha = 1$) then the major cost driver is the subcontracting (74.6% in the HARD-simulations and 57.1% in the PEN-simulations). If the tariff level is increased then the contribution of this cost driver decreases down to 15% (HARD) resp. 3.5% for $\alpha = 3$. At the same time, we observe a small increase in the contribution of penalty payments to the overall costs (from 0.5% up to 3.6% in the HARD-experiments and from 3.8% up to 12.0% in the PEN-experiments). However, the contribution of the travel costs becomes larger if the tariff level is raised. In the HARD-experiments, $\bar{c}_{5000}^{travel}$ (the cumulated travel costs) increase from 24.9% (α=1) to 81.8% (α=3) and in the PEN-experiments we observe an increase of $\bar{c}_{5000}^{travel}$ from 39.2% up to 84.5%. From these values, we conclude that both HARD and PEN compensate LSP charge increases by intensifying the usage of their own fleet.

3.3.4 Varying Penalties

The reported results from the simulation experiments show that the quality of the processes generated using the PEN-controlled resource allocation are significantly worse than the results observed in the HARD-configured simulation runs. In order to find out if this performance gap can be closed by finding more appropriate parameters for the penalty function, we have configured and executed additional simulation experiments. In these experiments, we have varied parameters of the applied penal-

Table 3.6 Contribution of the cost drivers

exp		α					
		1	1.25	1.5	1.75	2	3
HARD	$\bar{c}^{travel}_{5000}(\alpha, exp)$	24.9%	46.1%	63.1%	74.4%	79.9%	81.1%
	$\bar{c}^{sub}_{5000}(\alpha, exp)$	74.6%	53.8%	35.3%	23.0%	16.9%	15.4%
	$\bar{c}^{pen}_{5000}(\alpha, exp)$	0.5%	0.1%	1.6%	2.6%	3.2%	3.6%
PEN	$\bar{c}^{travel}_{5000}(\alpha, exp)$	39.2%	63.3%	75.8%	80.7%	82.8%	84.5%
	$\bar{c}^{sub}_{5000}(\alpha, exp)$	57.1%	30.2%	16.8%	11.0%	8.3%	3.5%
	$\bar{c}^{pen}_{5000}(\alpha, exp)$	3.8%	6.5%	7.4%	8.3%	8.9%	12.0%

ization scheme with the goal to identify the best suitable setting (Schönberger and Kopfer, 2008b).

A penalization of a constraint violation can be too weak or too severe. In the investigated example, the maximal penalty value of 25 money units might be too low to prevent the used decision algorithm identify the constraint violation. Therefore, we vary the maximal amount $P_{max}$of penalty payments. If we increase P_{max} from 25 up to 50,75,100 or 125 money units, each violation of the punctuality constraint (2.7) is depreciated with increasing intensity. However, if the penalization scheme is too strict, then a depreciation of a transportation plan caused by a small (local) constraint violation might prevent the decision algorithm from searching further in the area of the search space where the transportation plan with the small defect is situated. The search process runs into danger of being excluded from exploring larger parts of the search space. Consequently, the generated solutions run the risk of being suboptimal and having a significant distance to the optimal transportation plans. We introduce a fault tolerance threshold T^{tol}_{max}. As long as a solution proposal predicts an arrival time at a customer location, which is less than T^{tol}_{max} time units later than the closure of the time window associated with this customer visit, no penalties are accounted for this delay. We consecutively increase T^{tol}_{max} from 0 to 25, 50, 75. Using these two parameters, we obtain the piecewise-linear penalty function $h^*(x)$ defined in (3.6). The delay is x time units.

$$h^*_{P_{max},T^{tol}_{max}}(x) := \begin{cases} 0, & \text{if } x \leq T^{tol}_{max} \\ \frac{P_{max}}{100-T^{tol}_{max}} \cdot x - \frac{P_{max} \cdot T^{tol}_{max}}{100-T^{tol}_{max}}, & \text{if } T^{tol}_{max} < x < 100 \\ P_{max}, & \text{if } x \geq 100 \text{ time units} \end{cases} \tag{3.6}$$

We apply $h^*_{P_{max},T^{tol}_{max}}(\cdot)$ for each combination $(P_{max}, T^{tol}_{max}) \in \{50, 75, 100, 125\} \times \{0, 25, 50, 75\}$ twice. At first, we simulate scenarios with comparable freight tariffs ($\alpha = 1$) and secondly, we simulate scenarios with biased freight tariffs ($\alpha = 3$).

Throughout the simulations we record the maximal punctuality decrease (in percent) $\delta(P_{max}, T^{tol}_{max})$ after the demand peak and the cumulated overall costs $C^*(P_{max}, T^{tol}_{max}) := \frac{P_{max}, T^{tol}_{max}}{P^*_{max}, T^{tol*}_{max}} - 1$ (Among all combinations of maximal penalty and tolerance the parameter $(P^*_{max}, T^{tol*}_{max})$ setting leads to the minimal costs).

The results observed for the experiments with $\alpha = 1$ are presented in Fig. 3.9. The left isoline-plot shows that the observed maximal punctuality decreases $\delta(P_{max}, T^{tol}_{max})$. In $A^{\delta}_1(-0.6)$ maximal punctuality variations between -0.6% and 0 (light grey shaded area) are observed. Punctuality variations between -1% and -0.6% appear in $A^{\delta}_1(-1)$. Decreases of p_t between 1% and 1.4% take place in $A^{\delta}_1(-1.4)$

The right isoline plot compiles the average increase of the cumulated costs $C^*(\cdot,\cdot)$ incurred during the simulation runs. Additional costs of less than 5% ($A^C_1(0)$) are observed for small penalties and high tolerance values (light grey shaded areas). A cost increase of more than 15% is realized if $T^{tol}_{max} \leq 25$ and $P_{max} \geq 75$. We conclude, that if the LSP tariff level $\alpha = 1$ applies (same costs for both fulfillment modes) then the static parameter setting $P_{max} = 50$ and $T^{tol}_{max} = 75$ for h^* performs sufficiently well with respect to a high service quality as well as to the service cost minimization.

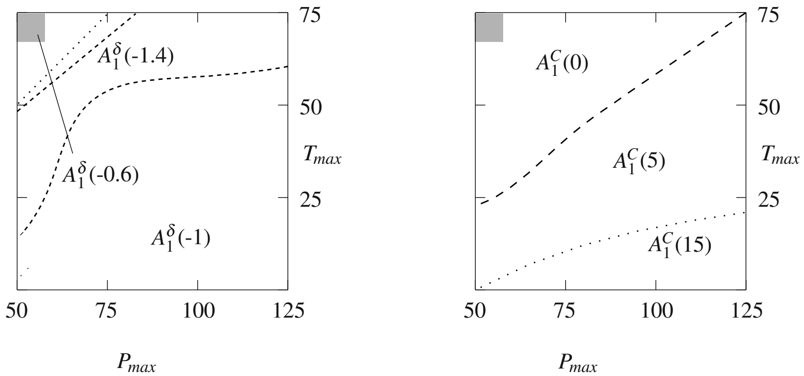

Fig. 3.9 Punctuality decrease (left) and costs increase (right) in the experiments with $\alpha = 1$.

Quite different results are observed in the experiments with $\alpha = 3$ (Fig. 3.10). The service quality optimal parameter setting is $P_{max} = 125$ and $T_{max} = 25$ with a maximal punctuality reduction of 2.9% (light grey shaded area in the left plot in Fig. 3.10). However, this setting causes a cost increase of 15% (cf. right plot in Fig. 3.10). On the other hand, the cost optimal parameter setting ($P_{max} = 50, T_{max} = 75$), indicated by the light grey shaded area in the right plot in Fig. 3.10, results in a punctuality collapse of around 20%. It is therefore impossible to find a parameter setting for $h*$ that satisfies both goals a) costs minimization and b) punctuality preservation to the maximal extent at the same time.

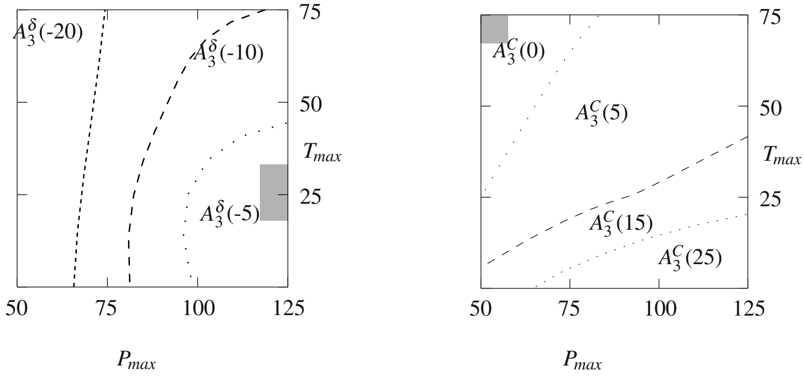

Fig. 3.10 Punctuality decrease (left) and costs increase (right) in the experiments with $\alpha = 3$

For reasonable tradeoff parameter settings (100,50) and (75,50) we observe a significantly higher punctuality decrease (compared to the punctuality preserving setting) or quite enlarged costs (compared to the cost optimal setting). Thus, it seems to be impossible to find a parameter setting for the PEN-approach that is appropriate for both planning requirements (least process costs and sufficiently high punctuality). This observation suggests that a planning approach based on the penalization of punctuality deficiencies is not appropriate for applications in a dynamic decision scenario, especially if load peaks occur.

3.4 Conclusions

In this chapter we have developed a DSS that is based on the state-of-the-art techniques for the previously introduced dynamic decision problem from transportation logistics. We followed the general idea to embed a decision making algorithm for a static decision problem into a rolling horizon framework. Whenever additional problem data has been collected the selected decision making algorithm is re-started. This concept has been applied to the dynamic decision problem from transportation logistics. An appropriate decision making algorithm has been configured. Two strategies for ensuring a continuous high reliability (measured by the observed customer site visit punctuality) are proposed. The first one (HARD) is not applicable because it does not consider some practical requirements. Although the performance of the HARD-configuration convinces with respect to efficiency as well as the service goal, its application is not possible because the partners forming the supply consortium under consideration are not treated fairly in case of demand peaks. The complete risk associated with a demand peak lies with the transport service agent.

However, we can use HARD as a reference strategy to evaluate an alternative strategy (PEN), where the negative impacts of a workload peak are shared among the partners of the supply chain. The results generated with HARD are interpreted as lower bounds to be reached by other control strategies (for example PEN).

Simulation experiments have been setup and have been executed in order to assess the two consortium configurations PEN and HARD in artificial benchmark scenarios. Offline- and online-evaluation indicators have been defined and evaluated.

The main conclusion from the observed simulation results is that the approaches proposed in the literature are not capable of handling the specific challenges of dynamic decision situations. The decrease of the punctuality rate of transport services leads to delays in subsequent process steps. From the viewpoint of the supply consortium coordinator, this behavior disqualifies PEN as an appropriate control strategy for the incorporated service agent.

For this reason, it is necessary to re-engineer the control strategy. However, every new control strategy must respect the interests of both the coordinator as well as the transport service agent.

From the achieved results we conclude that the simple re-application of a given search paradigm is not enough for managing the special challenges coming along with a dynamic decision problem. The presented results suggest that it is necessary to adapt the applied decision technique continuously to the varying decision situation instead of re-applying the same technique again and again. There are indications that it is necessary to take the current instance of a decision problem into account and to re-configure a decision algorithm so that significant dynamically appearing changes in the planning assumptions (reduced capacities, varied workload, etc.) can be integrated with superior decision making objectives (quality agreements, etc.). For example, a penalization scheme must fit to the current system state and the currently observed performance of the system. In the investigated problem at hand this condition is not fulfilled during and immediately after a load peak. If the search space appearance is significantly modified (i.e., because a relative capacity shortage occurs) then the evaluation scheme of proposed transportation plans must consider this changed planning assumption. However, such a temporal variation of the evaluation measure requires a temporal re-definition of the used formal optimization model.

Part II
Extending the Application Boundaries of Model-Based Planning

Chapter 4
Decision Support in Principal-Agent-Relationships

In this chapter we propose control strategies that adjust their decision behavior to the current process performance and/or to the state of the dynamic process environment. We propose an automatic re-configuration of the process control system by changing the process decision model maintained by the used DSS. It is demonstrated that such a process control system is able to integrate the process-related decisions of a superior supply consortium coordinator and of a subordinate service agent. Applying such an integrated process planning system simultaneously supports the service agents desire for maximal profitable routes as well as the coordinators need for reliability of the generated processes. By means of the previously introduced example scenario, we discuss the suitability of the proposed DSS framework.

The proposed extension of DSS enables the coordinator to intervene in the dispatching decisions of the service agent. Such an intervention is helpful in the event that the coordinator's requirements are no longer met by the decisions of a service agent. More concretely, the coordinator interventions are adapted to the reciprocal of the degree to which the coordinator requirements are fulfilled. In order to promote such an intervening strategy towards subordinate service agents, the coordinator pays a monetary compensation to the service agent, so that the additional expenditures of the service agent, which results from the deviation of deciding strictly in the sense of profit-maximization, are covered. Thus, no financial reasons oppose the granting of additional dispatching rights to the coordinator.

Current DSS-concepts do not support the integration of the coordinator into the dispatching and disposition framework used by a service providing agent. A methodological extension of DSS for online decision making is necessary. The development of appropriate extensions is the subject of this chapter.

In Section 4.1, we analyze the deficiencies of PEN from the perspective of a process control system. It turns out that the "failure" of PEN is caused by a conceptual shortcoming of DSS which appears in the event that the DSS is applied to rapidly evolving decision problems. We develop ideas to enhance the architecture of DSS with the goal to overcome the aforementioned deficiencies. The transfer of the innovations to logistics processes controlled by online optimization is carried out in Section 4.2. In particular, we discuss how we can use the DSS-extensions

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*, Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_4,

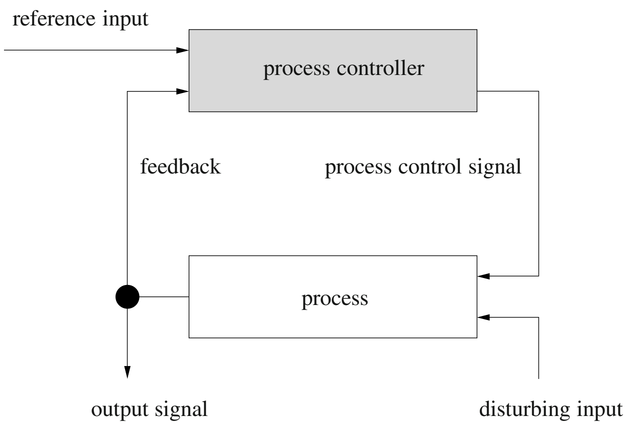

Fig. 4.1 Process-control circuit of the transport service agent Bierwirth (2000)

to support the coordination and integration of the decision making among supply consortium partners in the operational process planning in the event that principal-agent-relationships have to be considered.

4.1 Adaptive Process Control Systems

We start with the description of a generic process control system (Subsection 4.1.1). Next, we focus on the analysis of the core ("process controller") of the process control unit (Subsection 4.1.2). Then, we develop suggestions as to how the update decisions of a process controller can be biased (Subsection 4.1.3). This section terminates with a proposal to integrate the process controller alternation into the event handling scheme of a DSS (Subsection 4.1.4.)

4.1.1 Reliable and Unreliable Adaptive Systems

In order to find out the reasons for the failure of PEN to meet the goals of the supply consortium coordinator, we analyze this resource allocation strategy in dynamic environments from a more general perspective. The transport service agent and the process are interpreted as a two-component-system living in an evolvable environment (Fig. 4.1). The first component of the considered system is the **process** that is disturbed by *disturbing input* from the dynamic environment. The process cannot react to the disturbance but it returns a *feedback signal* describing its current state to the second component, which is the **process controller** (representing the transport service agent). It is able to manipulate the process by submitting a *process*

procedure process_management($reference\_input$);
(a) i:=0;
(b) t_i:=get_current_time();
(c) $process_{t_i}$:=generate_initial_process();
(d) $i = i+1$;
(e) **wait until** (disturbing input di_{t_i} appears) **or** (process terminates);
(f) **if** (process terminates) **then goto** (n)
(g) t_i:=get_current_time();
(h) f_{t_i}:=generate_feedback_signal($process_{t_i}$,di_{t_i});
(i) submit_feedback_to_controller(f_{t_i});
(j) pcs_{t_i}:= derive_process_control_signal($reference\_input$,f_{t_i});
(k) $process_{t_i}$:=update_process($process_{t_{i-1}}$,pcs_{t_i});
(l) $i = i+1$;
(m) **goto** (e);
(n) stop();

Fig. 4.2 Pseudo-Code representing a process-control circuit

control signal to the process. The process controller compares the current process feedback signal (the current process status) with a given reference input (representing the intended process status). If the two signals are congruent, the control goal is reached and the process is running as desired. In the event that the two signals are different, then a process control signal is generated. It is determined with respect to the severity of the feedback signal derivation from the reference signal. The process control signal is submitted to the process. A manipulation of the process based on the signal content is carried out with the goal to correct the process. The correction is completed as soon as the returned feedback signal once again complies with the reference input. This feedback-based process control structure is called a *process control-circuit.*

A pseudo-code-procedure *process_management($reference\_input$)* realizing the process control circuit is given in Fig. 4.2. This procedure is invoked with the $reference\_input$ as input parameter. At first, the iteration counter i is initialized (a), the current time is fetched (b) and the initial process is fixed (c). Then, the iteration counter is increased (d) and the procedure waits until a process disturbing input is detected or the process management period is over (e). If the process terminates (f) then the procedure is stopped (n). Otherwise, the current time is fetched (g) and the process feedback signal is derived (h). Afterwards, this feedback signal is forwarded to the process controller (i). The controller compares the received signal with the given reference input and derives an appropriate process control signal (j). Now, the existing process is updated by applying the derived control signal (k). In preparation of the next process update cycle, the iteration counter is increased (l) and the procedure falls back into a waiting (idle) state (m).

The configuration of the process control-circuit for the online-optimization problem introduced in Section 2.3.3 is as follows; as reference input, the least punctuality rate p^{target} is announced to the controller. The process feedback signal contains two types of information: a) the set $R^+_{t_i}$ of additionally arrived requests and b) the current punctuality rate p_{t_i}. The first information is used to derive an update of the so far allowed processes. In the event that additional requests have arrived the process controller decides how they are integrated into the existing process and submits a respective process control signal. The second type of information is used to decide if the current punctuality rate p_t complies with the reference input ($p_t \geq p^{target}$) or not ($p_t < p^{target}$). The returned process control signal contains the updated processes that replace the so far followed processes.

Both resource allocation strategies HARD and PEN represent realizations of the process controller. More precisely, the interacting decision model (2.1)-(2.8), the memetic algorithm introduced in 3.2.1 and the repair procedure shown in Fig.3.3 realize the *HARD-controller*. Similarly, the decision model (2.2)-(2.6), (2.8), (2.9) interacting with the memetic algorithm without the repair procedure establish the *PEN-controller*. These two decision models and algorithms are used in the procedure *derive_process_control_signal()*.

In both cases, the process control signal is generated automatically in response to the feedback signal received by the process controller. Bierwirth (2000) defines such a system represented by the two aforementioned components (process and process controller) as *adaptive (process control) system*. No exogenous intervention into the control-circuit occurs. This means that the procedure *derive_process_control_signal()* is obliged to derive a suitable output. If this procedure interacts with an external system then this interaction must be triggered and executed automatically. The interaction with a human decision maker is not intended.

A *reliable process controller* is able to produce processes whose feedback signals always comply with the reference input signal. If the adaptive system comes along with a reliable process controller then we call this system a **reliable adaptive system**. In contrast, an *unreliable process controller* is a process controller that is not reliable, e.g., there are disturbances affecting the process to which the process controller cannot respond with the generation of adequate process control signals in reaction to the disturbance (e.g., caused by the release of additional requests). An **unreliable adaptive system** maintains an unreliable process controller.

Using the recently introduced vocabulary, we recognize that the adaptive transport process control system developed in Chapter 3 (Section 3.2) falls into the category of reliable adaptive systems if HARD is incorporated by the process controller. If PEN is used to derive the control signal generator then the system belongs to the group of unreliable adaptive systems. In the first case (HARD) the procedure *derive_process_control_signal()* always generates a process control signal that complies with the reference input. It is $p_{t_i} \geq p^{target}$. In the latter case (PEN) there are combinations of the two input parameters $reference\_input$ and f_{t_i} for which the procedure *derive_process_control_signal()* does not return a process control signal complying with the reference input. We have then $p_{t_i} < p^{target}$. There are two potential reasons for this malfunction.

- During the specification of the process controller these combinations of input parameters have not been taken into account (illegal input).
- The internal rules of the process controller produce a wrong answer to the input values (illegal controller configuration).

In case of the first failure type, the controller does not know how to handle the provided input parameter combination and does not return any process update signal. An illegally configured process controller returns an inappropriate process update signal that does not match the requirements of the current planning situation. Consequently, the process management procedure stops.

The failure of the adaptive transport process control system developed in Chapter 3 (Section 3.2) is caused by an illegal controller configuration. During a load peak situation, an inappropriate process control signal is returned, because the returned signal does not modify the processes so that a punctuality rate $p_{t_i} \geq p^{target}$ is achieved.

4.1.2 Process Controller of Adaptive Systems

A process controller of an adaptive system consists of two components: the internal process model and the parameter adjustment. These components interchange information with the goal to identify a process control signal that updates the current process state to a state that complies with the reference signal. The gray shaded box in Fig. 4.3 shows the general structure of the controller as proposed by Bierwirth (2000).

We have an active controller component that manipulates a second passive controller element. The passive element is the *internal process model*. It maintains a (simplified) representation of the process (and of the relevant part of its environment). The representation is based on some parameters whose variation simulates the variation of the process control signal. From a given instantiation of the parameters (selection of a certain value for each parameter), a specific process control signal is formed. Instead of directly submitting this signal to the process, it is tentatively applied in the internal process model. The impacts of the simulated signal are evaluated and the evaluation result is submitted as *ex-ante feedback* to the second component which is called the *parameter adjustment*. If the ex-ante feedback complies with the reference input announced to the parameter adjustment then the tentative signal is converted into a process control signal (*instruction*) which is submitted to the process for implementation. In the event that the tentative signal does not lead to the desired process manipulation, the parameters describing the process control signal are varied by the parameter adjustment component of the controller. The new signal proposal is tentatively applied to the internal process model and so on until an appropriate process control signal is found. This signal is finally submitted as process control signal to the controlled process.

Maintaining and using the internal process model, the process controller is able to evaluate tentatively process control signal decisions before they become effective. The prefix *ex-ante* refers to feedback signals received and tentatively submitted before the process control signal selected to become effective is sent to the process execution system. After the process has been manipulated by the controller signal, the *ex-post*-feedback is generated and given back to the process controller (the term

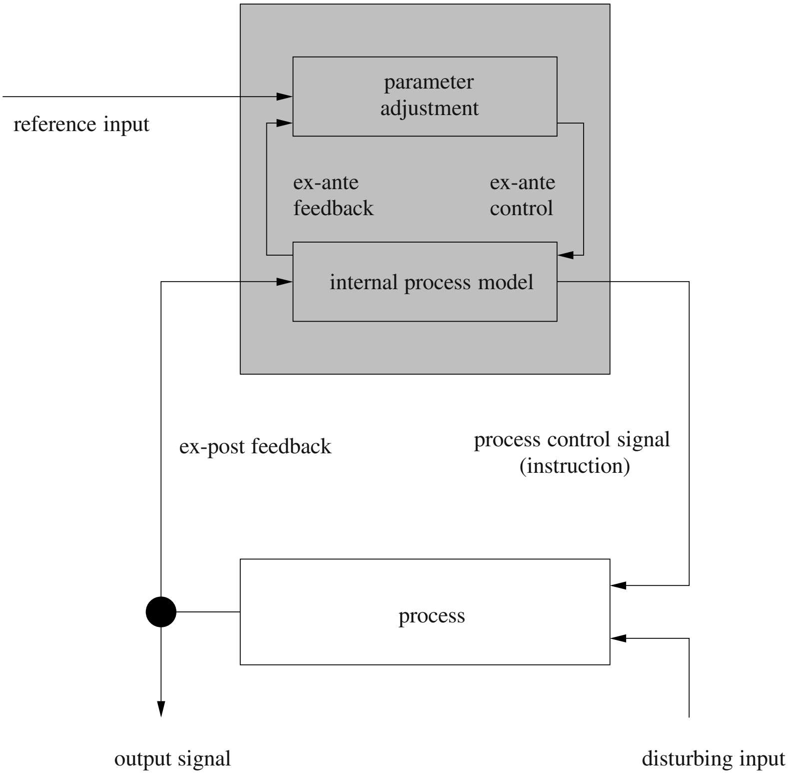

Fig. 4.3 Framework of an adaptive process control system

ex-post refers to the time after the process manipulation has become effective). In addition, it is provided as *output signal* to other system components who can exploit this signal for evaluation or other purposes.

A pseudo-code representation of the process-controller is depicted in Fig. 4.4. After the initiation of the procedure, a first signal proposal is derived (a) and used to parameterize the internal process model (b). An evaluation of the parameter instantiation follows (c). Now it is checked whether the evaluation result of the generated process control signal complies with the reference input (d). In the event that this is true, the current process control signal is returned (i). Otherwise, the following three steps are repeated until an appropriate process control signal is found or if it turns out that no such signal exists. In an iteration cycle, the tentative process control signal is updated (e), the internal model is updated according to the content of the tentative process control signal (f) and the re-parameterized model is evaluated (g).

procedure derive_process_control_signal(*reference_input*,*f*);
(a) pcs:=init_process_control_signal(*reference_input*,*f*);
(b) IPM:=parameterize_internal_process_model(pcs);
(c) e:=evaluate(IPM);
(d) **while** (*e* does not comply with *reference_input*) and (*pcs* can still be varied) **do**
(e) pcs:= update_process_control_signal(*reference_input*,*f*,*pcs*);
(f) IPM:=parameterize_internal_process_model(pcs);
(g) e:=evaluate(IPM);
(h) **end while**;
(i) return(*pcs*);

Fig. 4.4 Pseudo-code of the procedure *derive_process_control_signal()*

The analysis of line (d) of the aforementioned pseudo-code of the process controller reveals that the controller decides for a process control signal by means of one of the following two criteria:

1. The returned process control signal is able to vary the existing process so that the modified process complies with the requirements specified by the *reference_input* (left condition in algorithm step (d) in Fig. 4.4 is not fulfilled anymore).
2. The returned process control signal does not convert the disturbed process to a feasible updated process because no tested signal is able to do this (right condition in algorithm step (d) in Fig. 4.4 is not fulfilled anymore). In order to prevent a deadlock situation, the "best" found process control signal is returned.

If the process controller is in the HARD-configuration, then the internal process model is equal to the decision model (2.1)-(2.8). A *parameter instantiation* is represented by a set of values assigned to the decision variables of this mathematical decision model and e (determined in step (g)) is the corresponding objective function value according to (2.1). The testing cycle (d)-(h) of the algorithm in Fig. 4.4 is realized by the hybrid algorithm consisting of the Memetic Algorithm (Subsection 3.2.1) and the repair heuristic (Fig. 3.3) that tentatively evaluates several parameter constellations until the given termination criteria is met. Since we ensure the feasibility of each tentative parameter setting (by applying the repair procedure), we can be sure that the returned process control signal, which is the best solution found by the hybrid algorithm, updates the existing transportation plan (the process) by a transportation plan that respects the punctuality requirement. Furthermore, we drop the left condition check in step (d) in Fig. 4.4 because by construction, each proposed process control signal surely complies with the reference input. We left the second condition because we are looking for *the best possible* process update signal.

However, if we use the PEN-configured controller, then the used MA proposes process control signals with a punctuality quota p_t less than p^{target}. In the event that no new process control signal (proposed by the memetic algorithm without incorporated repair heuristic) is able to generate further proposals, while at the same time the proposals do not comply with the referential punctuality requirement (the right condition in line (f) within Fig. 3.3), the controller returns an inappropriate process control signal. This signal updates the running process into a new one that comes along with a too low punctuality rate.

If the PEN resource allocation strategy is used in the process controller then the controller is unreliable because it fails in load peak situations in the event that the LSP tariff is significantly higher than the travel costs (and the penalty payments) for deploying an own vehicle. However, outside load peak situations, the controller performs well. Therefore, a complete replacement of a PEN-configured controller is not intended. Instead, it would be better to “support” the PEN-strategy to improve its performance in load peak situations.

4.1.3 System Intervention by Image Modification

The supply consortium coordinator is not part of the adaptive process control system shown in Fig. 4.3. Its integration into the process control requires the extension of the adaptive process control system by a component that represents this coordinator. We propose to add a so-called model-controller to the process control system. The model controller represents the coordinator and emits signals to influence the process controller (representing the subordinate service agent) by modifying the internal process model.

We generalize the aforementioned idea of a temporal controller manipulation. By default, the process controller employs a given internal process model to identify a process control signal that updates a process in a way so that the updated process is *feasible*. This model is defined exclusively in accordance with the resource allocation strategy and the preferences of the subordinate service providing agent. A process is feasible if the ex-post feedback signal complies with the reference-input given to the process controller. The model (2.2)-(2.6), (2.8) and (2.9) is the default model used in a PEN-configured process controller.

An exceptional situation occurs as soon as the controller is not able to generate a process control signal that makes the controlled process feasible. The planning goals of the superior coordinator agent and the planning goals of the subordinate are conflicting now. Such a situation is achieved using the PEN-configured process controller after the initialization of the demand peak if the LSP-charge level $\alpha >> 1$ prevents the LSP incorporation.

In such an exceptional situation, the process controller is re-configured temporarily in order to enable it to generate process control signals that have not yet been able to be generated. During this phase, the coordinator goals are preferentially addressed without respecting the desires of the subordinate agent. For example, the coordinator wants to subcontract selected requests in order to ensure a timely request completion even if the LSP-charges for the selected requests are higher than the sum of travel costs and penalty payments for a late arrival.

As soon as the exceptional situation is over (e.g., a process control signal is proposed that leads to a feasible process), the original process controller configuration is re-established and used until the next exceptional situation is detected. Now, the

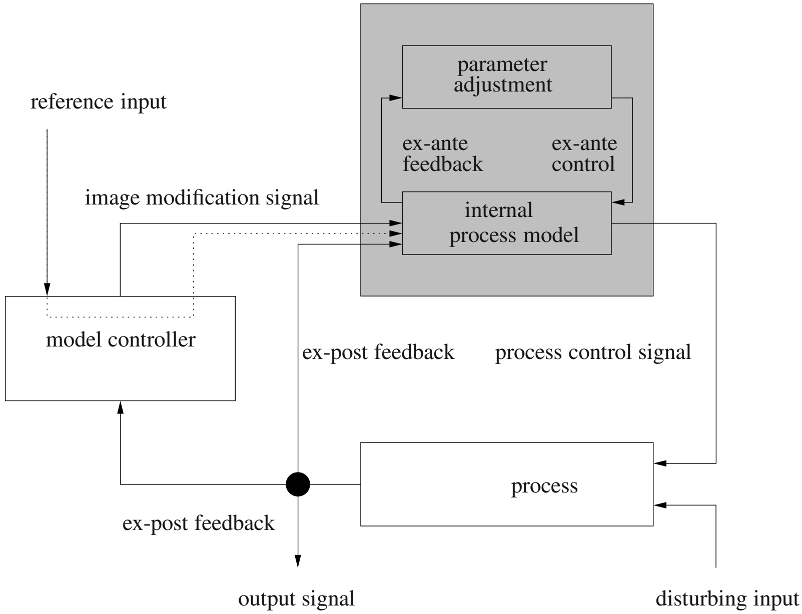

Fig. 4.5 An adaptive process control system with image modification

planning objectives and preferences of the subordinate agent are basically addressed in the process planning.

By comparing the reference input with the ex-post process feedback signal it is possible to detect exceptional situations in which the process operates unsatisfactorily. According to the aforementioned idea, the process controller must be updated until it again operates at an acceptable performance. Bierwirth (2000) explains that there are two generic approaches to influence the behavior of the process controller and to intervene in the adaptive process control. Each of the two intervention approaches addresses one of the two only intervention possibilities: modifying the reference input or modifying the internal process model of the controller. In both cases, a model modification signal, which is generated by the model controller (representing the supply consortium coordinator), is submitted to the internal process model and this model is altered according to the content of the model modification signal.

The first mentioned intervention approach is called **goal modification**. It aims at adjusting the behavior of the process control system by altering by the reference input in exceptional situations. Goal modification is of interest in engineering control systems but has only minor importance in the application of supply consortium operations planning and logistics process planning because the guiding goal of the process generation does not change here, only the circumstances under which the goal must be achieved. For a more detailed discussion of goal modification interventions and its consequences for the design of adaptive process control systems we refer to the book of Bierwirth (2000).

To improve the management of logistics processes in evolving and/or volatile environments the second intervention possibility, which is referred to as **image modification** is more suitable. Controller interventions by image modification are established by manipulating the internal process model maintained by the controller.

The generic architecture of an adaptive process control system using image modification is presented in Fig. 4.5. A third component, the model controller, is added to the process-control system. The ex-post feedback signal from the process is now directed to the model controller. In addition, the reference input is re-directed from the process-controller to the model controller. After having received a new ex-post feedback signal, the model controller carries out a comparison of the ex-post feedback signal with the intended process performance. If both signals comply then the model controller does not emit any output signal except for the reference input that is required by the process controller. However, if the model controller detects a deviation of the ex-post feedback from the reference input, it interprets the deviation and submits an adequate *image modification signal* to the process controller. The process controller analyzes the image modification signal and establishes modifications of the maintained internal process model before this model is used again. Until another image modification signal is received, the process controller uses the varied internal process model to determine the ex-ante feedback.

Three types of intervention (applied in combination or alone) are possible:

- Excluding parameter instantiations from evaluation as process control signal.
- Allowing additional parameter instantiations in the evaluation cycle of the process control procedure.
- Varying the scheme used to evaluate parameter combinations: This is equivalent to a replacement of the procedure $evaluate(\cdot)$ (step (c) and (g) in the procedure $derive\_process\_control\_signal(\cdot)$ in Fig. 4.4). Since the comparison of the outcome of the evaluation function with the reference input decides about the selection of a tentative process controller signal, the variation of this function enables the generation of process control signals that have not yet been selected to be returned by the controller.

From an algorithmic perspective, the implementation of image modification requires several modifications of the procedures presented before, which are mainly caused by the necessity

1. to generate and submit a model control signal mcs in each update cycle describing the required model changes,
2. to setup a new internal process model IM for each cycle in the procedure $process\_management()$ and
3. to update the currently used internal process model IM in the process control procedure $derive\_process\_control\_signal()$ according to the current value of mcs_{t_i}.

Fig. 4.6 shows the updated process management algorithm that now incorporates image modification. Instead of submitting the process feedback signal directly to

procedure process_management_im(*reference_input*);
(a) i:=0;
(b) t_i:=get_current_time();
(c) $process_{t_i}$:=generate_initial_process();
(d) $i = i + 1$;
(e) **wait until** (disturbing input di_{t_i} appears) **or** (process terminates);
(f) **if** (process terminates) **then goto** (n)
(g) t_i:=get_current_time();
(h) f_{t_i}:=generate_feedback_signal($process_{t_i}$,di_{t_i});
(i) submit_feedback_to_controller(f_{t_i});
(i*) mcs_{t_i}:=generate_model_control_signal(t_i,f_{t_i});
(i**) IM_{t_i}:=update_model(t_i,$IM_{t_{i-1}}$,mcs_{t_i});
(j*) pcs_{t_i}:= derive_process_control_signal_im(*reference_input*,f_{t_i}, IM_{t_i});
(k) $process_{t_i}$:=update_process($process_{t_{i-1}}$,pcs_{t_i});
(l) $i = i + 1$;
(m) **goto** (e);
(n) stop();

Fig. 4.6 Pseudo-code representing a process-control circuit with image modification

the process controller, the procedure *submit_feedback_to_controller*() invoked in step (i) in Fig. 4.6 forwards the ex-post-feedback signal to the model controller. The new procedure *generate_model_control_signal*() called in step (i*) returns the necessary model control signal *mcs* to process management procedure and therefore addresses (1.) in the previous list of extensions/modifications. This procedure represents the model controller shown in Fig. 4.5. The signal is given to the model update procedure *update_model*() (i**) which returns the new process control model (addresses (2.) in the above list of extensions). In a final modification (3.), we replace the command to call the procedure *derive_process_control_signal*() by the command to execute the new process control signal generating procedure *derive_process_control_signal_im*() that requires the specification of the internal process model M_{t_i} to be used (j*). The last modification enables the process controller to exploit the recent version of the process model.

The extended control signal generating procedure *derive_process_control_signal_-im(reference_input,f,IM)* is presented in Fig. 4.7. Both calls of the model instantiation now include the specification of the process model to be used in (b*) and (e*).

The model control procedure *generate_model_control_signal*(t_i, f_{t_i}) exploits the two input parameters t_i (current time) and the f_{t_i} (ex-post feedback signal). In the event that the model control signal is independent from the current time t_i as well as from the ex-post feedback signal f_{t_i} the model controller is called a **static model controller**. If only the current time t_i determines the model control signal then the procedure represents a **dynamic model controller**. Finally, if the returned model control signal depends upon the submitted ex-post feedback signal, the model controller is defined as an **adaptive model controller**.

```
procedure derive_process_control_signal_im(reference_input,f,IM);
(a)  pcs:=init_process_control_signal(reference_input,f);
(b*) IPM:=parameterize_internal_process_model(pcs,IM);
(c)  e:=evaluate(IPM);
(d)  while (e does not comply with reference_input) and (pcs can still be varied) do
(e*)   pcs:= update_process_control_signal(reference_input,f,pcs,IM);
(f)    IPM:=parameterize_internal_process_model(pcs);
(g)    e:=evaluate(IPM);
(h)  end while;
(i)  return(pcs);
```

Fig. 4.7 Pseudo-code of the procedure *derive_process_control_signal_im()* using an externally specified process model

4.1.4 Image Modification and Event Handling in DSS

As we have seen in the three-layer-model introduced in Fig. 3.2 the type of the DSS response depends on the severity (or impacts) of a disturbing event. A considerable event is managed without incorporating the process model but according to some pre-given process update rules. Only an event of major significance leads to the re-call of the process planner (the process controller) with the aim to generate a process control signal which enables the update of the corrupted process. However, not all events can be successfully managed by the proposed event handling as we have seen by means of the example of a load peak. If such a *threatening* event is detected, the model controller intervenes in the regular DSS-process-management cycle (1)-(9) shown in Fig. 3.1 and varies the internal process model of the process by submitting a model control signal to the controller. In order to enable the DSS to handle such a threatening event, we extend the three-layer model of Séguin et al. (cf. Subsection 3.1.3) and add a fourth layer. This additional layer (*coordination layer*, cf. Fig. 4.8) is invoked by the third layer if a threatening event has appeared that cannot be handled in the current configuration (expressed by the current internal process model) of the planning system. The third layer is extended by an *error signal generator* that compares the actual system state with the reference input and maps the detected deviation into a numerical value (the **error signal**). This value is forwarded into the fourth layer to the controller that transforms the error signal into instructions for modifying the planning logics (**model control signal**). This signal is sent back into the generation layer. The planner is reconfigured according to the control signal and then the required process re-planning is carried out using the re-adjusted planner.

4.2 Image Modification and Process Re-Planning

This section describes the transfer of the image modification concept into the application field of supply consortium process planning. The main contribution is the

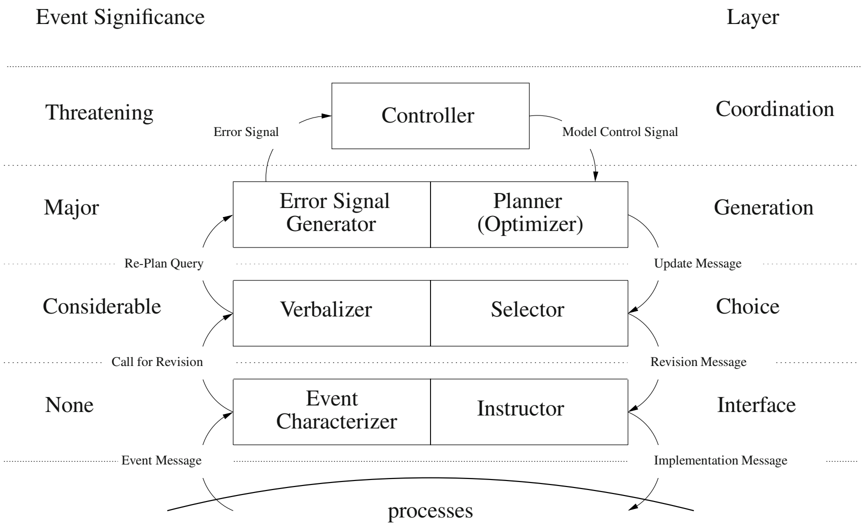

Fig. 4.8 Four-layer event handling model for a process management system

consolidation of the decision support systems of the superior supply consortium coordinator (the principal) and of a subordinate service agent in one single planning system (*Integrated Principal-Agent Resource Allocation*). It is aimed to improve the responsiveness of the supply consortium in the event that extraordinary disturbance compromises the fulfillment of customer demand and disturb the well-balanced cooperation between the coordinator and a subordinate service agent.

Initially, we explain the idea to use image modification to intervene in the resource allocation logic of the subordinate transport service agent (Subsection 4.2.1). In so doing, the coordinator is represented by the model controller and the service agent is modeled as the process controller. In 4.2.2, we propose a framework to integrate the decision making of the superior and the subordinate planning entities. Finally, a proposal to model the supply consortium coordinator's behavior in a model controller is given in Subsection 4.2.3.

4.2.1 Image Modification and Integrated Principal-Agent Resource Allocation

A major challenge in supply consortium process planning is the coordination of the decision making between the superior coordinator and the subordinate service agents in the allocation of resources. All resources are provided by the service agents and the coordinator has no direct access to the resources, e.g. decisions about the deployment of a resource are made by the owning service agent exclusively.

In exceptional situations it is useful to grant the coordinator a direct allocation opportunity. Such a situation occurs e.g., if unexpected material shortages at an intermediate process step in the fulfillment of a request will lead to a delayed fulfillment of the overall customer demand. In order to prevent the payment of penalties (caused by a late demand fulfillment) it is necessary that resulting emergency requests are fulfilled immediately and with highest priorities. Typically, the service agent might not be aware of the urgency of the request fulfillment or cannot fulfill such a request in a profitable way because this request cannot be consolidated with other requests in one of its profitable routes or shipments.

The transport service agent compares the costs for fulfilling an urgent request with its own resources (travel costs as well as potentially applying penalty payments) with the costs for using an LSP (LSP charges). By default, it will select the cheaper fulfillment mode independent of the punctuality of the request. In the event that the supply consortium coordinator has the opportunity to directly allocate a resource for the emergency request it selects the LSP fulfillment because the LSP guarantees a timely fulfillment of the requests. As long as the LSP is the cheaper mode, the decisions of the coordinator and of the transport service agent comply with each other. In the event that the LSP charges are higher than the costs for using the own fleet of the transport service agent a conflict between the coordinator and the transport service agent occurs. This conflict cannot be solved to the satisfaction of both the coordinator and the service agent. A coordinator exploits its superiority and executes the granted right to overrule the subordinate agent: the LSP fulfillment mode is selected for some requests. Here, a direct allocation and deployment of an LSP contributes to the preservation of the performance of the overall supply consortium, although the transport service agent does not gain an immediate benefit. However, due to the intervention of the coordinator, the reliability and therefore the acceptability of the supply net by the customer remains high so that in the longer perspective, also the service agent achieves a benefit because he can earn profits as a participant of the ongoing supply consortium.

In order to motivate a (transport) service agent to agree to such a direct resource allocation opportunity, the additional costs of the subordinate agent (resulting from the coordinator intervention) are covered and paid by the coordinator. In order to keep the additional costs of the coordinator as low as possible, the direct resource access is only used if necessary. Therefore, we propose to use image modification to control the application of a direct resource allocation by the coordinator. As long as the decisions of the subordinate service agent lead to processes that comply with the least punctuality requirement of the coordinator (represented by the reference input in Fig. 4.6) the transport service agent uses its own process controller (represented by the process controller in Fig. 4.6) to adjust the transportation plan (represented by the process in Fig. 4.6) after a disturbance (e.g., an additional request) has appeared. However, if the generated transportation plan does not comply anymore with the specified least punctuality rate p^{target} (current punctuality p_t is less then p^{target}) then a coordinator intervention (represented by the model control signal submitted to the internal process model as described in Fig. 4.6) becomes necessary. Thus, the model controller in the adaptive process control system with image modification represents the supply consortium coordinator.

4.2.2 Preparing Integrated Principal-Agent Resource Allocation

The cooperation between the coordinator of a supply consortium and the service agents are based on a set of rules regulating the responsibilities and describing the obligations of each partner of the supply consortium. In order to enable an integrated principal agent resource allocation it is necessary to specify exactly, whether, when and how the principal intervenes in the allocation of a service agent's resources and how the subordinate agent is compensated or rewarded for granting the access to its resources.

When negotiating the conditions of the cooperation, the first decision is about the application of integrating principal-agent decision making in the resource allocation. If both the coordinator and the service agent agree to set up such a resource allocation then the reference input must first be specified.

For an integrated principal-agent resource allocation carried out by the supply consortium coordinator and by the transport service agent it is necessary to declare a priori in which cases the coordinator is granted the right to intervene in the resource allocation of the transport service agent. No intervention rights are granted as long as the reclusively generated processes comply with the reference input agreed in the consortium contracts. With increasing deviation of the quality of generated processes from the required/stipulated process quality more interventions into the resource allocation should be allowed.

It is necessary that the supply consortium coordinator and the service agent(s) agree on

- a clear definition of the intended quality of processes (reference input) and
- a measure that determines the deviation of the quality of current processes from the reference and which determines the intensity of the direct resource access in dependence of the observed deviation.

For deciding about the compliance of generated processes with the specified reference qualities, we use N indicators that map the performance of a process (e.g. a transportation plan) at a time t into the $N+1$-tuple $t,(i_1(t),\ldots,i_N(t))$ of real values (the system's state at time t). Let Im_u denote the set of possible values for the indicator $i_u : t \mapsto i_u(t)$. The subset $\mathscr{F}(t_i)$ of $Im_1 \times \cdots \times Im_N$ contains exactly all those performance instantiations that comply with the consortium contracts. Now, the set $\mathscr{D}(t_i) := [t_i;\infty) \times \mathscr{F}(t_i)$ describes all future states of the system that comply with the reference quality according to the agreed consortium contracts. It is called the **system development corridor at time** t_i.

A countermeasure maintaining a sufficiently high punctuality should be established before the system leaves the system development corridor. In order to be able to start the necessary actions as early as possible, we define a **core** $\mathscr{C}(t_i) \subseteq \mathscr{D}(t_i)$

of the previously specified system development corridor as a subset $\mathscr{C} := [t_i;\infty) \times \bar{\mathscr{F}}(t_i)$, where $\bar{\mathscr{F}}(t_i) \subseteq \mathscr{F}(t)$. If the current system state belongs to the core $\mathscr{C}(t_i)$ then it is assumed that there is no danger of the system performance falling out of the system development corridor in the next re-planning cycle.

The intensity of the model adaptation is determined by measuring the distance of the current system state from the core of the system development corridor. If this distance is "zero" then no intervention into the resource allocation carried out by the service agents is allowed. If the distance is "small", only a few intervention opportunities become allowed. However, if the distance is "large" then significant coordinator interventions into the resource allocation carried out by the service agent are granted.

We now describe formally what the distances "zero", "small" and "large" mean and how the model controller transforms the process feedback signal $f_{t_i} := p_{t_i}$ into a model control signal mcs_{t_i} according to the determined distance of the current system state from the core. Three steps are executed consecutively in the procedure $generate\_model\_control\_signal(t_i, f_{t_i})$ in order to generate the model control signal.

In a first step, the model controller compares the observed ex-post feedback signal received from the process with the set of intended stated described by the system development corridor for the current time. The deviation of the current state from the desired states is called the error signal $e(t_i)$ at time t_i. It is 0 as long as the current state falls into the core of the system development corridor. If the deviation of the current state from the core increases then the error signal also grows, so that the error signal prematurely indicates whether the system is in danger of leaving the system development corridor as soon as the next external disturbance like a peak in the system workload occurs.

In a second step, a function maps the error signal (representing the current system state) to a (real) value quantifying the necessary controller intervention intensity. This function is called the **intensity function** h. In general, the intensity function should be defined according to the core $\mathscr{C}(t_i)$ of the system development corridor $\mathscr{D}(t_i)$. The intensity function value should be 0 as long as the system state falls into the core of the system development corridor ($e(t_i) = 0$). As soon as the system state leaves the core of the system development corridor the intensity function value should start increasing. Once it has finally left the system development corridor h should have reached its maximal value β.

In the third step, the previously determined intensity value $h_\beta(p_{t_i})$ is used to parameterize the so called **implementation function** H specifying the model modifications to be implemented into the internal process model. These modifications depend on the current time t, the recent system performance and on the intervention intensity expressed by the current intensity function value. Thus, $mcs_{t_i} := H(t_i; h_\beta(p_{t_i}))$ is used as the model control signal submitted to the process controller.

4.2.3 Algorithmic Update of the Agent's Decision Model

Kaluza et al. (2003) propose two different strategies to be used by a coordinator of a supply consortium in order to manipulate the behavior of a subordinate service agent in principal-agent relationships. Depending on the kind of business relationship between the principal and the agent, one of the strategies is applied.

In the first kind of relationship (**non-cooperative relationship**) it is not possible to grant a direct access to the agent's resources for the coordinator. Here, it is necessary to promise monetary incentives for the agent if the agent behaves in accordance with the coordinator's requirements. The access to the agent's resources is achieved indirectly by making a specific decision of the agent much more profitable for the agent. The service agent has to update its evaluation system whenever the coordinator adjusts its incentive system.

The second kind of analyzed relationship between a superior coordinator agent and a subordinate service agent is called **cooperative relationship**. Here, the subordinate service agent agrees to temporarily grant the coordinator a direct access to its resources because the service agent is convinced that it can benefit from such an agreement. If the service agent deploys its resources the granted access rights must be considered e.g. by reserving capacities to be managed by the coordinator.

Clearly, the boundaries of an incentive system modification as well as the amount of capacity to be granted to the consortium controller must be fixed in advance in the supply consortium contracts. In both kinds of relationship, the intervention of the coordinator causes costs accruing to the coordinator. In the non-cooperative relationship, additional income is promised by the coordinator to the service agent if it desists to follow exclusively its own decision preferences. This additional income of the service agent must be paid by the whole consortium. In the cooperative relationship, the service agent has to be paid for reserving parts of its resources for the exclusive coordinator access (opportunity costs).

Often, the core element of the used process controller is a mathematical optimization model. Thus, every incentive scheme modification and/or every temporarily granted direct resource access of the coordinator requires a re-parameterization of the model. As we have seen in Section 2.3 the dynamic decision problem of the service agent can be modeled as an online optimization model, e.g., as a sequence of mathematical optimization models. To implement a temporary coordinator intervention into the resource management of the service agent it is necessary to adjust the objective function (in the non-cooperative relationship) or the constraint set (in the cooperative relationship) of the next instance of the online model to the desired intervention. Consequently, the implementation function H (representing the model control signal) describes the variation of the objective function and/or the variation of the constraint set. The variation of an objective function of an optimization model changes the search trajectory of an exact or heuristic model solver through the search space (described by the constraint set). Therefore, we call the feedback-controlled modification of the objective function **Search Direction ADaption (SDAD)** in an online optimization model. The feedback-controlled modification of the constraint

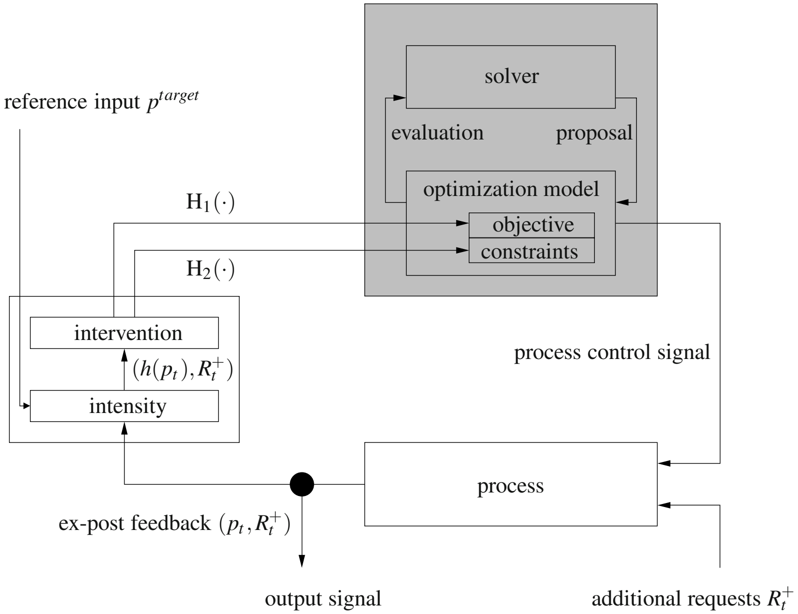

Fig. 4.9 Intervention of an adaptive process control system (gray shaded area) with SDAD and CSAD

set of instances of an online optimization model is referred to as **Constraint Set ADaptation (CSAD)**.

Schönberger and Kopfer (2008a) configure an adaptive process control system with image modification. Within this system, SDAD and CSAD are used to adjust a default optimization model to the current process performance. Fig. 4.9 shows the proposed configuration.

Additional requests disturb the execution of the running processes at time t. As soon as this disturbance is detected an update of the currently executed process transportation plan is triggered. Both, the current punctuality p_{t_i} and the set $R_{t_i}^+$ of the additionally released requests are forwarded to the model controller.

The model controller consists of two consecutively processed stages. In the first stage, the intensity function compares the current process punctuality p_t with the externally specified reference punctuality rate p^{target}. The generated intervention intensity $h(p_t)$ is forwarded to the intervention function together with the set R_t^+ of additionally released requests. Next, the intervention function derives two model control signals $H_1(\cdot)$ and $H_2(\cdot)$ which are both forwarded to the process controller (gray shaded area in Fig. 4.9). The process controller consists of a mathematical optimization model (representing the current decision task) and of a model solver. The first signal H_1 is submitted to the objective function. This signal contains information on how to adjust the objective function (SDAD). Instructions to adjust the constraint set to the current process feedback signal (CSAD) are contained in the second signal H_2, which is directed to the constraint set. The application of H_1 and/or H_2 corresponds to a call of the function $update\_model(\cdot)$ in step (i**) in Fig. 4.6.

After the required adjustments have been established, the solver generates solution proposals which are evaluated using the updated decision model until the best proposal with respect to the current model instance is identified. This proposal is submitted as process control signal to the process and updates it. The updated process is executed until additionally arriving requests require a further process revision.

4.3 Summary

Adaptive process control systems provide a base architecture for automatic process management systems applied in dynamic and evolving process environments, where a recurrent process re-planning is necessary. An adaptive process control system supports the decision behavior of a service agent by maintaining a decision model to evaluate and select service agent decision alternatives. In order to integrate the decision making of a superior coordinator agent and of a subordinate service agent, we have suggested adding an additional control component (*the model controller)* to the adaptive control system. The model controller manipulates the decision model used to support the service agent's decision making. This kind of manipulation is called image modification and enables the coordinator to intervene in the resource allocation decisions made the subordinate service agent. The usage of an adaptive process control system with image modification facilitates a DSS to handle so-called process threatening events that cannot be handled by DSS without image modifications.

Chapter 5
Adaptive Controllers for Mathematical Optimization Models

Adaptive model controllers are promising candidates for extending DSS for dynamic process planning problems. They are designed to feed back variations of model assumptions about the severity of a process control problem to the model base during the process control phase. Using image modification, this class of process controller is able to reflect the altered assumptions into the maintained process control logic. Thus, a synchronization of the process control problem with the maintained internal process control problem description becomes possible.

Although the synchronization of a formalized problem situation with the motivating real-world situation has been evaluated positively in the control of computer systems (Arnold et al., 2005; Šegvić et al., 2006) the precise adjustment of mathematical optimization models for the control of logistic systems has received only minor attention so far.

In this chapter, we report the configuration of adaptive model controllers for supporting the solving of the instances of the online transport process planning problem introduced in Chapter 2. We interpret a process control model as the planning logic guiding a subordinate transport service proving agent. The model controller is interpreted as the superior principal (the supply consortium coordinator) which temporarily intervenes in the service agent's disposition decision making in order to overcome the negative impacts of a spontaneously appearing demand peak.

We start with the definition of an adequate system development corridor. Its core is fixed and a suitable intensity function (Section 5.1) is recommended. Then, we propose two different implementation functions. In Section 5.2, we develop an SDAD-approach, which manipulates the scheme that is used to determine the reimbursement of the subordinate service agent by the superior coordinator agent. A CSAD intervention approach is introduced in Section 5.3. We test and evaluate both intervention strategies within comprehensive computational experiments whose setup and results are discussed and reported in Section 5.4. Finally, both controller adjustment strategies SDAD and CSAD are combined into a hybrid approach. We report the observed simulation results in Section 5.5.

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*, Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_5,

5.1 Preparations

In order to equip the transport process control system introduced in Subsection 3.2 with image modification features it is necessary to define an appropriate system development corridor and its core (Subsection 5.1.1). Furthermore, a suitable intensity function must be defined which measures the distance of the current system state from the system development corridor and from its core (Subsection 5.1.2).

5.1.1 System Development Corridor

The system development corridor for the problem introduced in Section 2.3 is defined as follows. We use the only indicator p_t, $Im_1 := [p^{target};1]$ and set $\mathscr{F}(t) := [p^{target};1]$. The corridor $\mathscr{D}(t_i)$ is then given by $\mathscr{D}(t_i) := [t_i;\infty) \times [p^{target};1]$. The union of the two grey shaded areas in Fig. 3.4 represents the system development corridor for the investigated relationship between the supply consortium coordinator and a transport service agent.

The core of the system development corridor $\mathscr{D}(t_i)$ defined for the investigated relationship of a supply consortium coordinator and the transport service agent is defined by $\mathscr{C}(t_i) := [t_i;\infty) \times [p^{target} + 0.1;1]$. It serves as a reference that is used to decide whether a direct access is granted or not. The dark grey shaded area in Fig. 3.4 represents the core $\mathscr{C}(t_i)$ of the system development corridor $\mathscr{D}(t_i)$.

5.1.2 Definition of a Suitable Intensity Function

In order to check whether the current ex-post feedback signal p_t still complies with the intended system state, the deviation of p_t from the core of the system development corridor is measured and saved in the *error signal* $e(t_i)$ defined in (5.1).

$$e(t_i) := -min(p_{t_i} - (p^{target} + 0.1);0) \tag{5.1}$$

This error signal prematurely indicates if the system is in danger of leaving the system development corridor as soon as the next external disturbance like a peak in the system workload occurs.

For the investigated dynamic transport process planning problem we propose the subsequently described intensity function h_β (Schönberger and Kopfer, 2007b). We define h_β as the piecewise linear function (5.2) which is calculated by calling GET_-INTERVENTION_INTENSITY($e(t_i)$) .

$$h_\beta(p_{t_i}) = \begin{cases} 0, & e(t_i) \leq 0 \\ 1 \cdot \beta, & e(t_i) \geq 0.2 \\ 5 \cdot e(t_i) \cdot \beta, & \text{in all other cases} \end{cases} \tag{5.2}$$

The intensity function value $h_\beta(p_{t_i})$ is 0 if $p_{t_i} \geq p^{target} + 0.1$ (HQ period, $(t_i, p_{t_i}) \in \mathscr{C}(t_i)$), $h(p_{t_i}) = 1$ if $p_{t_i} \leq p^{target} - 0.1$ (LQ period, $(t, p_t) \notin \mathscr{D}(t)$) and it decreases linearly from β down to 0 if p_{t_i} increases from $p^{target} - 0.1$ up to $p^{target} + 0.1$ (transition phase).

5.2 Accounting Scheme Adaptation

We start with the introduction of accounting schemes (Subsection 5.2.1). Then, we explain how such a scheme is used to modify the objective function of an optimization model (Subsection 5.2.2). Finally, we use an accounting scheme to adapt online transport disposition and dispatching problem instances (introduced in Subsection 2.3) to the currently observed process punctuality rate p_t (Subsection 5.2.3).

5.2.1 Accounting Schemes

A subordinate transport service agent in the supply consortium decides autonomously about the fulfillment modes of the waiting requests. High quality services (express courier or individual same-day delivery) are quite reliable but very costly. The usage of standardized request execution processes is cheaper due to the realization of economies of scale but in this mode of transport individual request requirements cannot be fulfilled.

An accounting scheme determines how expenses of a subordinate agent are accounted to the given budget. The main idea of the accounting scheme adaptation is to define a rule that determines the refunding of the transport partner only taking into account its reliability. Actually incurred costs are not relevant for the determination of the refunding. The accounting scheme is regularly adapted to the currently observed punctuality rate p_{t_i}. Consequently, if the transport partner's performance varies then the rule for refunding the transport partner's expenses also varies.

In the event that the punctuality rate p_{t_i} is higher than p^{target} then the transport partner's expenses are refunded completely for each fulfillment mode (individual service by an LSP and consolidated transport with a vehicle of the service agent). The refunded amount is accounted to the budget designated for covering the transport costs. However, if the punctuality rate is at risk to fall below p^{target} or has even fallen below this threshold, then expenses for the reliable services (LSP-options) are reimbursed at a higher percentage or even completely but expenses for the cheap and unreliable transport services (self entry) are only partly covered.

Each participating service agent decides independently about the planning of the processes (resource deployment, etc.) within its area of authority. However, a superior coordinator agent modifies the accounting scheme used to determine the amounts, with which the budget for transport services is charged. Doing so, the coordinator balances or shifts the financial attractiveness among the two requests fulfillment modes. In the event that the punctuality rate p_{t_i} is low the additional expenses of using LSPs are not or only partly charged to the budget of the transport service agent. As a result, the LSP-mode is more frequently selected by the service agent and more requests are served on a timely basis. Thus, the variation of the accounting scheme forces the subordinate agents to adapt its process decisions to the guidelines of the superior coordination agent. At the end, the reliability of the supply consortium re-increases because transport operations become more timely again.

5.2.2 *Objective Function Re-Parameterization*

In order to enable the usage of an adaptive accounting scheme for an automatic re-parameterization of the objective function (2.9) in the instances of the online optimization model introduced in Section 2.3, we reformulate the model (2.2)-(2.6), (2.8 and (2.9)). The objective function (5.3) replaces (2.1). This objective function does not determine the actually incurred costs of the generated transportation plan but the fictitious costs, which are actually accounted to the budget available for the transport service providing supply consortium partner (Schönberger and Kopfer, 2009a). The service agent aims at minimizing this amount in order to maximize its own remaining profit.

$$\lambda_{t_i}^{a} \cdot \sum_{p \in P(t_i)} \sum_{v \in \mathscr{V}} \left(C^1(p) + C^2(p)\right) x_{pv} \;+\; \lambda_{t_i}^{b} \cdot \sum_{r \in R(t_i)} C^3(r) y_r \rightarrow \min \tag{5.3}$$

The ordered pair $(\lambda_{t_i}^a, \lambda_{t_i}^b)$ is called the **accounting scheme** used for adjusting (5.3) to p_{t_i}. The values of the two parameters are re-calculated before a new instance of the online optimization model is instantiated. Thus, recent information about the process punctuality is reflected into the next instance of the online model.

If $\lambda_{t_i}^a$ is increased relatively to $\lambda_{t_i}^b$ then the fulfillment of a request with own vehicles becomes financially less attractive than the subcontracting of this request. In the event that $\lambda_{t_i}^a$ is reduced relatively to $\lambda_{t_i}^b$ the attractiveness of the self-fulfillment mode increases.

5.2.3 *Adaption of the Accounting Scheme to the Current Process Punctuality*

This subsection is about the definition of an implementation function H_1, which modifies the so far used objective function by updating the so far used accounting scheme $(\lambda_{t_{i-1}}^a, \lambda_{t_{i-1}}^b)$ to the new scheme $(\lambda_{t_i}^a, \lambda_{t_i}^b)$. This updated scheme is used for re-weighting the costs of the two fulfillment modes in the decision situation at time t_i.

We define the weight $\lambda_{t_0}^b$ of the subcontracting costs to be 1 and do not vary this value anymore, so that we have $\lambda_{t_i}^b = 1$ for all $i = 0, 1, \ldots$. In an HQ period, the weight of the self-fulfillment mode should also be equal to 1. If the tariff level α applies then one additionally subcontracted request produces costs that are α times larger than the additional costs produced by the usage of the self-fulfillment mode. Taking into account the tariff level α, we use the following principle to determine $\lambda_{t_i}^a$. Initially, we weight both fulfillment modes equally and define $\lambda_{t_0}^a = 1$. As soon as the punctuality p_{t_i} falls below $p^{target} + 0.1$ and tends to leave the core $\mathscr{C}(t_i)$ of the system development corridor $\mathscr{D}(t_i)$ the weight $\lambda_{t_i}^a$ is increased proportionally to the decrease of p_{t_i}. The maximal value of $\lambda_{t_i}^a$ is set to $1 + \alpha$ and this value is reached in the event that p_{t_i} has fallen below $p^{target} - 0.1$. If an increase of p_{t_i} is detected then $\lambda_{t_i}^a$ is reduced proportionally. Its minimal value is 1.

If we define the function $H_1(t_i, p_{t_i})$ as described in equation (5.4) then we get $\lambda_{t_i}^a = 1$ during HQ periods. The value of $\lambda_{t_i}^a$ increases correspondingly if p_{t_i} decreases. If an LQ period is finally reached, then $lambda_{t_i}^a$ equals $1 + \alpha$. We use the function H_1 as implementation function for adapting the objective function (5.3). This adaptive accounting scheme is a realization of a model controller designed according to the Search Direction ADaptation (SDAD) strategy.

$$H_1(t_i, p_{t_i}) := (\lambda_{t_i}^a, \lambda_{t_i}^b) = (1 + \alpha \cdot h_\beta(p_{t_i}), 1) \tag{5.4}$$

5.3 Adaptive Exercising of LSP-Options

A coordinator agent can only indirectly intervene in the dispatching decisions of the fleet managing agent by adjusting the evaluation system for service agent decisions (SDAD). If the implementation function H_1 is used then the achievement of the intended manipulation of the dispatching behavior cannot be guaranteed. To remedy this deficiency, we propose to use a model preprocessing approach that determines values of some decision variables so. These values cannot be updated by the used model solving approach any more. By fixing the value $y_r = 1$ for selected variables y_r the coordinator ensures that its intervention will surely become effective.

The pre-selection of subcontracted requests modifies the constraint set of the next instance of the online optimization model. Therefore, this approach realizes a model controller, which uses the principle of **Constraint Set ADaptation (CSAD)**.

We discuss the capabilities of a decision model preprocessing to be used as coordinator intervention channel (Subsection 5.3.1). Next, an adjustable constraint family is developed that enables the injection of coordinator interventions into the online model of the investigated online vehicle routing problem (Subsection 5.3.2) during the run-time of the processes. The necessary extensions of the online planning framework introduced in Section 3.2 are outlined in Subsection 5.3.3. Different rules for selecting those decision variables for which a values is determined before the model solving routine is started are proposed in Subsection 5.3.4.

5.3.1 Decision Model Preprocessing and Presolving

The optimization model (2.1)-(2.8) represents the planning task of the subordinate transport service agent. Similarly to the first model controller configuration using adaptive accounting schemes, we prepare the decision model instance by proposing a suitable optimization model reformulation. Again, we replace (2.1) by (2.9) and skip the least quality restriction (2.7).

Coordinator interventions address the decisions about the exercise of LSP-options for some requests $r \in R(t_i)$. The coordinator decisions about the LSP-option drawing are reflected into the current decision model instance by fixing $y_r = 1$ for some requests. Thus, a preselection of values for some decision variables is carried out before the solving of the next instance of the online model. A decision variable for which a value is determined becomes a problem parameter whose value cannot be changed.

Techniques to preselect values of decision variables are subsumed under the term *presolving* (Andersen and Andersen, 1995). Presolving is normally used to erase redundancies in a decision model without shrinking or even extending the set of feasible model solutions. However, we can also use this technique to intervene in the process of online decision making by fixing the values of selected variables of the model (2.2)-(2.6), (2.8), (2.9). Doing so, we shrink the set of feasible solutions that is scanned by the applied search procedure (which is the Memetic Algorithm introduced in Chapter 3) during the solving of the current model instance.

In the context of principal-agent relationships, we can adjust the decision space of the subordinate agent if the principal agent applies presolving. Actually, we equip the principal to analyze the model data and to manipulate the domain of some decision variables y_r, $r \in \tilde{R}$, before the data is used to define the current model instance. Such a data analysis and model manipulation is referred to as *decision model preprocessing* (Solnon, 2002). Here, model preprocessing is applied if the process punctuality p_{t_i} falls below the given threshold p^{target} or if it is in danger of falling below p^{target} after the next additional requests have been released. By fixing $y_r = 1$ for some requests $r \in \tilde{R} \subseteq R^+(t_i)$ the coordinator extends the LSP-usage in order to ensure or in order to re-increase the reliability of the processes.

5.3.2 An Adjustable Constraint Family

The original online-formulation of the considered dynamic deployment problem proposed in Hiller et al. (2006); Krumke et al. (2002); Grötschel et al. (2002) is extended by an adjustable constraint family. To ensure that the coordinator decisions about the exercise of LSP-options are considered by the fleet managing agent during the re-deployment planning, the constraint (2.5) has been sharpened by Schönberger and Kopfer (2007b) to the constraint (5.5). This constraint enables the consideration of coordinator decisions for every new instance of the update model.

$$y_r = 1 \quad \forall r \in R^E(t_i) \cup \underbrace{\tilde{R}}_{\text{intervention}} \tag{5.5}$$

5.3.3 Preparing the Coordinator Agent's Interventions

For determining the intensity of the coordinator agent's intervention we use the intensity function $h_\beta(p_{t_i})$. The principal contribution of the coordinator agent to the disposition and to the dispatching of the fleet is to scan the incoming requests and to select requests for which an LSP-option is exercised in accordance with the recent value of h_β. In order to keep the once generated processes as stable as possible the number of options exercised for already scheduled requests should be kept as small as possible. Only for additionally arrived requests which have not yet been initially scheduled is it beneficial to decide about the exercise of LSP-options. The decision tasks of the consortium coordinator now comprise

1. the determination of an adequate portion of the requests in $R^+(t_i)$, for which LSP-options are exercised (realized) at the re-planning time t_i (*intensity determination*).
2. the selection of the previously fixed number of requests from $R^+(t_i)$ and their insertion into the set $\tilde{R} := R(t_i, h_\beta(p_{t_i}))$ in dependence on the current intervention intensity $h_\beta(p_t)$ (*request choice*).

In the event that the current punctuality rate p_{t_i} is higher than the intended threshold rate $p^{target} + 0.1$, no option will be exercised by the coordinator. The set $R(t_i, h_\beta(p_{t_i}))$ remains empty. If p_{t_i} has fallen below $p^{target} - 0.1$ then almost all additionally released requests should be directed into $R^+(t_i, h_\beta(p_{t_i}))$, so that we get "$R(t_i, h_\beta(p_{t_i})) \approx R^+(t_i)$": for all additional requests an LSP-option is exercised. If p_{t_i} increases (decreases) then the number of requests put into $R(t_i, h_\beta(p_{t_i}))$ should be decreased (increased) proportionally with respect to p_{t_i}.

In the request choice, new requests which are not compatible with the existing vehicle routes should be identified and preferentially put into $R(t_i, h_\beta(p_{t_i}))$. Here, a new request r is compatible with the routes if r can be served so that "only slight modifications to the routes become necessary". Therefore, the coordinator agent needs to identify all new requests that cannot be served by the existing routes because

- the associated pickup and/or delivery location require detours and/or
- their associated delivery time window closes before a vehicle can reach the delivery site.

For the identified "incompatible" requests LSP-options are exercised.

The value $h_\beta(p_{t_i})$ is interpreted as percentage of the recently released requests at time t_i for which the SC-mode is determined by the coordinator agent. Now, the number $N_{t_i}^{PRE}(p_{t_i})$ of requests directed into the subcontracting mode at time t_i is determined as specified in (5.6). At time t_i an LSP-option is exercised for at most $\lceil \beta \cdot \mid R^+(t_i) \mid \rceil$ additionally arrived requests.

$$N_{t_i}^{PRE}(p_{t_i}) := \lceil h_\beta(p_{t_i}) \cdot \mid R^+(t_i) \mid \rceil. \tag{5.6}$$

No request is enforced into the SC-mode if $p_{t_i} \geq p^{target} + 0.1$. The percentage of enforced externalization decreases (increases) smoothly and proportionally with an increasing (decreasing) punctuality rate p_{t_i}.

A sequencing rule Φ determines the order $SEQ(R^+(t_i), \Phi)$ of the elements in $R^+(t_i)$. It assigns a numerical value σ_r to each request $r \in R^+(t_i)$ and specifies whether $R^+(t_i)$ is ordered by increasing or by decreasing σ_r-values.

We first arrange the $m(i)$ elements contained in the set $R^+(t_i)$ of recently released requests in the sequence $SEQ(R^+(t_i), \Phi) := (r_{i_1}, r_{i_2}, \ldots, r_{i_{m(i)}})$ according to Φ. Next, we initialize $R(t_i, h_\beta(p_{t_i})) := \emptyset$. Then, we consecutively insert the requests $r_{i_1}, r_{i_2}, \ldots$ into the set $R(t_i, h_\beta(p_{t_i}))$. As soon as the number of elements in $R(t_i, h_\beta(p_{t_i}))$ exceeds the number $N_{t_i}^{PRE}(p_{t_i})$ we stop inserting requests into $R(t_i, h_\beta(p_{t_i}))$.

We define the function $\mathscr{F}_{SEQ}(R, z)$ to return the subset of R in which the first z requests of R according to the order defined by SEQ are contained. Using this function, the intervention function $H_2(t_i, h_\beta(p_{t_i})) :=$ "Set $y_r = 1$ for all $r \in R(t_i, h_\beta(p_{t_i}))$" is formally defined in (5.7) and the coordinator intervention expressed as the set $R(t_i, h_\beta(p_{t_i}))$ is determined by H_2.

$$R(t_i, h_\beta(p_{t_i})) := H_2(t_i, p_{t_i}) := \mathscr{F}_{SEQ}(R^+(t_i), N_{t_i}^{PRE}(p_{t_i})) \tag{5.7}$$

The tentative definition (5.5) of the adaptive constraint family is now finalized in (5.8).

$$y_r = 1 \; \forall r \in R^E(t_i) \cup R(t_i, h_\beta(p_{t_i})). \tag{5.8}$$

All the remaining requests from the set $R^+(t_i) \setminus R(t_i, h_\beta(p_{t_i}))$ can be freely assigned into one of the two fulfillment modes by the fleet management agent.

5.3.4 Intervention Specification

The marginal costs of a certain request can hardly be calculated so that substitutional measures for the profitability / compatibility of a request r must be exploited instead to decide about the applied request fulfillment mode. Those requests that seem to be least profitable or that seem to be least compatible with the already generated routes are selected accordingly and put into the set $R(t_i, h_\beta(p_{t_i}))$. In the following, we present some simple rules for identifying portfolio incompatible requests (Schönberger and Kopfer, 2009b).

5.3.4.1 Random Request Sequencing (RRS)

If this rule is applied then a randomly selected value is drawn from the interval $[0,1]$ and assigned to σ_r (assuming a uniform distribution). The requests from $R^+(t_i)$ are then sorted by increasing σ_r-values. This rule merely serves as a reference in order to find out whether biasing the request sequencing affects the overall planning results.

5.3.4.2 Distance-to-be-Bridged Sequencing (DBS)

Requests in the middle of the operations area can more often be combined with other requests into profitable routes than requests which are far away from the center of the operations area. We first calculate the geometric median $med(R^+(t_i))$ from all requests contained in $R^+(t_i)$. Let μ_r be the location of the site associated with the request r and let $dist(med(R^+(t_i)), \mu_r)$ denote the Euclidian distance of the site of request r to the calculated median $med(R^+(t_i))$. We define the following sorting criterion (5.9) and sort the requests from $R^+(t_i)$ so that the σ_r-values decrease. Requests situated on the periphery of the operations area are the first for which LSP-options are exercised, in the expectation that they cannot be combined with other requests in profitable routes.

$$\sigma_r := dist(med(R^+(t_i)), \mu_r) \tag{5.9}$$

5.3.4.3 Vehicle Availability Sequencing (VAS)

For each request $r \in R^+(t_i)$ the number vn_r of vehicles that can reach the site μ_r from their current positions before the time window of r closes is calculated. This number defines the sorting criterion (5.10). Then, the requests in $R^+(t_i)$ are sorted by increasing vn_r-values. Consequently, those requests which cannot be reached on time or only by few own vehicles are subcontracted preferentially. This sorting attempts to prevent penalty payments for late arrivals.

$$\sigma_r := vn_r \tag{5.10}$$

5.3.4.4 Remaining Time Based Sequencing (RTS)

RTS sorts the requests in $R^+(t_i)$ by increasing remaining time in which the site of request r can be visited without violating the associated time window $TW(r) = [t_r^+; t_r^-]$. Therefore, the sorting criterion (5.11) is defined as the difference between the closing time of $TW(r)$ and the current time t_i. The requests in $R^+(t_i)$ are then sorted by increasing σ_r-values, so that the most urgent requests are preferentially subcontracted by the coordinator.

$$\sigma_r := t_r^- - t_i \tag{5.11}$$

Costs and benefits of a single request can hardly be evaluated since the coupling effects of combining the fulfillment of several requests are quite high. This observation motivates the development and implementation of a rule that tries to identify those requests which cannot be combined efficiently with other requests. For these isolated requests LSP-options are preferentially exercised.

5.3.4.5 Isolation Based Sequencing (IBS)

In order to evaluate the "degree of isolation" of the site of a request $r \in R(t_i, p_{t_i})$, we first calculate for each request r its distance $d_1(r)$ from the median $med(R(t_i))$ of the current request portfolio $R(t_i)$. After having calculated this distance for each request in $R^+(t_i)$, we calculate the normalized distance $d_1^*(r) := \frac{d_1(r)}{\max\{d_1(r) | r \in R^+(t_i)\}}$ for each request $r \in R^+(t_i)$. If $d_1^*(r)$ is close to 1 then r is situated at the edge of the operations area, which is often a first hint for isolation.

To find out whether $r \in R^+(t_i)$ can be combined with other known requests into an efficient route, we calculate the distance $mindist(r)$ to the nearest other request site in the complete request portfolio $R(t_i) \setminus R^E(t_i)$. We consider only requests that have not yet been subcontracted. It is $mindist(r) := \min\{d_2(r, r_j) + d_3^{tw}(r, r_j) | r_j \in R(t_i) \setminus R^C(t_i)\}$, where $d_2(r, r_j)$ gives the travel distance between μ_r and μ_{r_j}. The term $d_3^{tw}(r, r_j)$ is used to depreciate the spatial distance in the event that the time windows $TW(r) := [t_r^+, t_r^-]$ and $TW(r_j) = [t_{r_j}^+, t_{r_j}^-]$ of r and r_j interdict the combination of the two requests in one route. It is $d_3^{tw}(r, r_j) := 0$, if $\min\{| t_r^+ - t_{r_j}^- |, | t_j^+ - t_r^- |\} \geq dist(r, r_j)$ (that is, there is enough time for a vehicle to travel from μ_r to μ_{r_j} or vice versa) and in all other cases it is $d_3^{tw}(r, r_j) := dist(r, r_j) - \min\{| t_r^+ - t_{r_j}^- |, | t_{r_j}^+ - t_r^- |\}$. Finally, we calculate the normalized minimal distance indicator $mindist^*(r) := \frac{mindist(r)}{\max\{mindist(r) | r \in R(t_i) \setminus R^C(t_i)\}}$.

$$\sigma_r := d_1^*(r) \cdot mindist^*(r) \tag{5.12}$$

The value (5.12) is then assigned as sorting value to the request r. If σ_r is small (close to 0) then the site μ_r is either in the center of the operations area or it is close

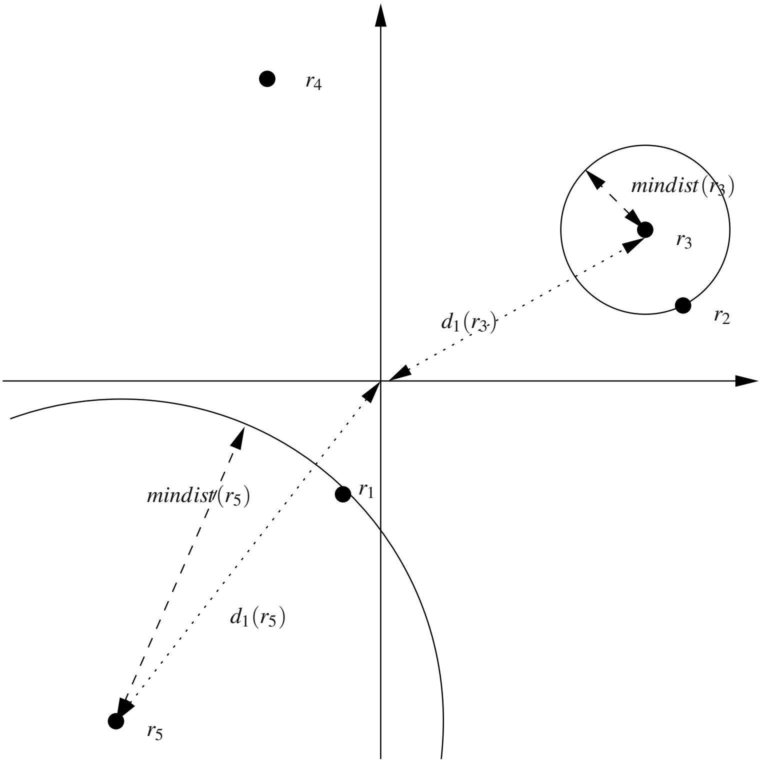

Fig. 5.1 Isolation-based sequencing

to the site of at least one another requests. If a request site μ_r is situated at the edge of the operations area and not closely situated to the sites of other requests then r can be classified as isolated ($\sigma_r \approx 1$).

Fig. 5.1 shows an example with five requests. The sites of r_4 and r_5 are isolated. They are far away from the other request sites as well as from the median of the five request sites. Since the site of r_1 is situated close to the median of all five request locations it is not treated as isolated. The sites of the two remaining requests r_2 and r_3 are also not isolated because they are situated closed to each other.

We now sort the requests in $R^+(t_i)$ by decreasing σ_r-values. At the beginning of the sequence of requests the most isolated requests are found and LSP-options for these requests are exercised first.

5.4 Evaluation and Assessment of the Model Controllers

We repeat the computational simulation experiments outlined in Section 3.3. In addition to the already executed simulation runs for HARD and PEN, we now performs experiments using the image modification approaches with adaptive accounting schemes (SDAD) and with an adaptive exercising of LSP-options (CSAD). In order to enable a clear evaluation of both adaptive approaches, we apply them separately and not together, e.g., if SDAD is used then we set $H_2(t_i, p_{t_i}) := \emptyset$ and if we apply CSAD them we set $H_1(t_i, p_{t_i}) := (1,1)$. We begin with the description of the experimental setup (Subsection 5.4.1). Afterwards, we describe and discuss the observed performances for the different strategies (Subsection 5.4.2).

5.4.1 Simulated Scenarios

We extend and re-use the simulation setting introduced in 3.3. The process planning is most challenging if the costs for drawing a subcontracting option are significantly higher than the costs for the self-fulfillment by the own fleet. In all other cases, the application of PEN supports the integration of the planning objectives of the controller and of the fleet managing agent. Therefore, we restrict our computational experiments and simulate only scenarios where $\alpha = 3$ (the exercise of an LSP-option costs three times the costs of the own vehicle usage including necessary penalty payments).

The determination of the system setting requires the specification of the intervention strategy *exp* applied by the supply consortium coordinator. In addition to the already evaluated strategies HARD and PEN, we now evaluate the strategy SDAD. Furthermore, we execute simulation experiments with CSAD. We repeat the simulation runs for each CSAD request selection strategy RRS, DBS, RTS and IBS as well as for SDAD. Again, we execute experiments with different seedings $\omega \in \mathscr{O} := \{1,2,3\}$ of the applied MA meta-heuristic.

We execute simulation runs with different maximal intervention intensities $\beta \in \{0.2, 0.4, 0.6, 0.8, 1\}$ in order to identify the most appropriate intervention intensity (The $\beta = 0$-case is equivalent to PEN). Overall, we setup $\mid \{CSAD-RRS, CSAD-DBS, CSAD-RTS, CSAD-IBS, CSAD-VAS, SDAD\} \mid \cdot \mid \{0.2, 0.4, 0.6, 0.8, 1\} \mid \cdot \mid \{1,2,3\} \mid = 6 \cdot 5 \cdot 3 = 90$ additional planning system settings.

Combining the previously mentioned parameter settings with the four request streams collected in $\mathscr{P}$, we setup $90 \cdot \mid \mathscr{P} \mid = 90 \cdot 4 = 360$ scenarios for simulation. In each scenario, a target punctuality $p^{target} = 0.8$ must be met.

We deploy the same online and offline performance indicators introduced in Subsection 3.3.2 for assessing SDAD and CSAD. Here, we compare the observed performance indicator values for the four strategies HARD, PEN (the two static strategies), SDAD and CSAD (the two “adaptive strategies”).

5.4.2 Presentation and Discussion of Results

We first investigate different parameterizations of SDAD and CSAD in order to find out the best parameter combinations (Subsubsection 5.4.2.1). For each of the two adaptive strategies, we identify the most appropriate parameters (maximal intervention intensity β, request selection rule if CSAD is applied). A detailed report about the online- and offline performance of the two resulting strategies is given. Therein, we compare the new strategies with the state-of-the-art strategies HARD and PEN (Subsubsections 5.4.2.2-5.4.2.5).

5.4.2.1 Parameterization of SDAD and CSAD

Initially, we aim at determining the best parameter setting for SDAD and CSAD. For SDAD, we check for which maximal intervention intensity β we achieve the best results. For CSAD we compare the different combinations of the maximal intervention intensity β with the previously introduced request sequencing rules RRS, DBS, VAS, RTS and IBS.

In order to determine and to distinguish the performance of the different parameterizations (exp, β), we compare the maximal punctuality decrease $\delta(exp, \beta)$, the percentage $\pi(exp, \beta)$ of LQ-situations and the cumulated costs $c_{5000}(exp, \beta)$ offline after the end of the simulations. After we have identified the "best setting" for each strategy *exp*, we restrict the presentation of further details of the simulations experiments to these superior parameterizations.

We start with the analysis of the impacts of different maximal intensity values in the SDAD-experiments. Tab. 5.1 contains the maximal punctuality decrease $\delta(SDAD, \beta)$, the percentage $\pi(SDAD, \beta)$ of LQ-situations and the cumulated costs $c_{5000}(SDAD, \beta)$ observed for different β-values. With the exception of the cumulated costs, we see that a maximal intervention intensity $\beta = 1.0$ is advantageous. Therefore, we use SDAD with the maximal intervention intensity $\beta = 1.0$ in the remainder of this book. The additional costs represent the "price" of the increased process reliability.

Table 5.1 Offline evaluation of different maximal intervention values β in the SDAD-experiments

	β					
	0	0.2	0.4	0.6	0.8	1.0
$\delta(SDAD, \beta)$	-38.20%	-21.13%	-16.92%	-10.78%	-6.34%	-5.63%
$\pi(SDAD, \beta)$	100%	100%	100%	90.00%	65.00%	57.50%
$c_{5000}(SDAD, \beta)$	55748.3	53518.5	53282.8	56250.2	60239.2	64225.6

The identification of the best parameter setting for CSAD requires the selection of a request selection strategy and the determination of the maximal intervention intensity. Tab. 5.4.2.1 contains the maximal decrease $\delta(exp, \beta)$ of the punctuality rate $p_t(exp,\beta)$ for a given combination of resource allocation strategy $exp \in \{CSAD-RRS, CSAD-DBS, CSAD-VAS, CSAD-RTS, CSAD-IBS\}$ and of an intervention intensity $\beta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ compared to the initially observed punctuality rate $p_{1000}(exp,\beta)$. The shown values increase if the intervention intensity β is increased. Thus, maximal intervention intensity seems to be beneficial with respect to the desire to guarantee a least punctuality. The random request selection strategy performs best (least decrease after the initiation of the demand peak). We observe that DBS and IBS also perform very well. However, if the vehicle availability is used to determine the selection of subcontracted requests (VAS) or if the remaining service time (RTS) is exploited then a significantly more severe timeliness decrease has to be observed.

Table 5.2 Maximal punctuality quota decrease $\delta(exp,\beta)$

exp	β					
	0.0	0.2	0.4	0.6	0.8	1.0
CSAD-RRS	-38.20%	-28.03%	-19.15%	-15.57%	-11.09%	-8.88%
CSAD-DBS	-38.20%	-25.08%	-19.50%	-13.67%	-12.72%	-9.47%
CSAD-VAS	-38.20%	-24.84%	-18.91%	-14.66%	-12.66%	-10.42%
CSAD-RTS	-38.20%	-21.61%	-16.14%	-15.12%	-11.63%	-10.27%
CSAD-IBS	-38.20%	-26.06%	-18.96%	-15.00%	-11.42%	-9.27%

From the results delivered in Tab. 5.3 we learn that the percentage of LQ-situations is reduced if the intervention intensity grows. Again, VAS and RTS (and partly RRS, too) show a very bad performance for non-maximal intensities. However, for the maximal intervention intensity all five selection strategies demonstrate a similar behavior. Now, RTS shows best performance. The presented π-values suggest choosing the maximal possible intervention intensity in order to keep the system performance in the system development corridor as long as possible. For $\beta = 1$ RTS performs best but all four other request selection strategies can compete with RTS.

Table 5.3 Percentage $\pi(exp,\beta)$ of LQ-situations

exp	β					
	0.0	0.2	0.4	0.6	0.8	1.0
CSAD-RRS	100%	80.00%	35.00%	17.50%	17.50%	15.00%
CSAD-DBS	100%	25.00%	17.50%	17.50%	15.00%	15.00%
CSAD-VAS	100%	85.00%	65.00%	17.50%	60.00%	15.00%
CSAD-RTS	100%	62.50%	60.00%	17.50%	15.00%	10.00%
CSAD-IBS	100%	15.00%	17.50%	22.50%	17.50%	15.00%

We learn from the three aforementioned tables that an intensive coordinator intervention is beneficiary for the process quality. Actually, more requests are enforced into the subcontracting fulfillment mode, which finally leads to higher request fulfillment costs. The cumulated request fulfillment costs are summarized in Tab. 5.4. An increase of the maximal intervention intensity leads to an increase of c_{5000}. The least costs are produced by DBS, followed by RTS, IBS, RRS and VAS.

Table 5.4 Cumulated costs $c_{5000}(exp,\beta)$

exp	β					
	0.0	0.2	0.4	0.6	0.8	1.0
CSAD-RRS	55748.3	65655.7	74301.5	80722.3	85564.6	89599.4
CSAD-DBS	55748.3	65545.0	70862.2	75052.5	79515.2	82696.6
CSAD-VAS	55748.3	69292.4	76340.8	82941.0	88211.3	92016.4
CSAD-RTS	55748.3	68539.9	74233.7	77135.5	80492.6	83254.9
CSAD-IBS	55748.3	64328.7	70587.9	76371.9	79928.6	84251.8

The offline performance of CSAD improves if the intervention intensity is increased. The best performance is observed for $\beta = 1$. Similarly to SDAD, we will configure CSAD with $\beta = 1$ so, that the maximal possible intervention intensity is exploited.

In order to identify the "best" request selection strategy, the indicator-specific performances of the strategies are consolidated into a single numerical indicator. Therefore, we first determine the "performance rank" of each strategy for each of the three analyzed indicators. As an example, we consider the default strategy RRS. With respect to δ, this strategy exhibits the best performance and RRS is assigned the δ-rank 1. Observing the achieved π-values its performance is second best and gets the π-rank 2. The σ-rank of RRS is 3 (DBS and IBS perform better here) and the c_{5000}-rank of RRS is 4 (only VAS generates higher costs). Secondly, we weight each rank with an indicator-specific weight and sum up the weighted ranks for each request selection strategy. The request selection strategy coming along with the lowest weighted rank-sum will be selected as default request selection strategy applied together with CSAD. Tab. 5.5 presents the weighted ranks and the sum of weighted ranks. We weight the punctuality loss δ by 40% (primary improvement need), the cumulated costs c_{5000} and the maximal observed subcontracting quota by 30% each. Using these parameters, we observe that DBS exhibits the best overall performance. For this reason, DBS is selected as the default request selection strategy incorporated by CSAD.

Table 5.5 Consolidation of the ranking of the performance indicators

	δ	π	c_{5000}	Σ
weight	40%	30%	30%	
RRS	0.4	0.6	1.2	2.4
DBS	1.2	0.6	0.3	2.2
VAS	2.0	0.6	1.5	2.3
RTS	1.6	0.3	0.6	2.5
IBS	0.8	0.6	0.9	2.3

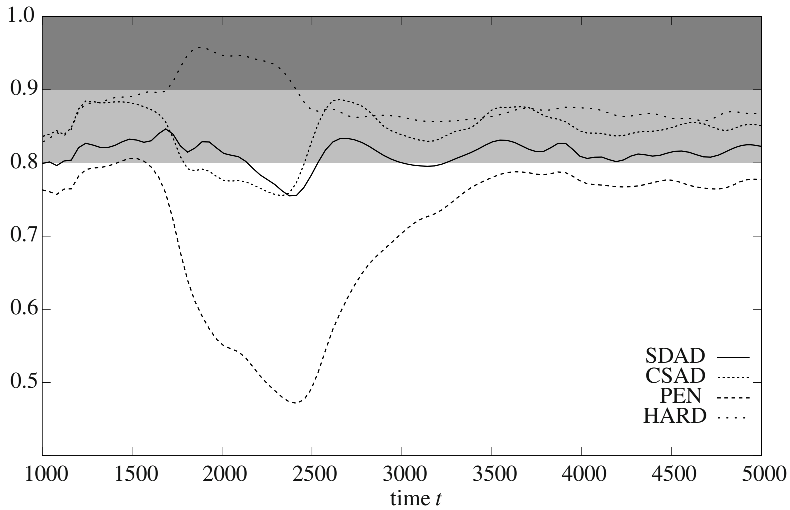

Fig. 5.2 Development of the punctuality $p_t(exp)$

5.4.2.2 Online Comparison of Static and Adaptive Strategies

Fig. 5.2 contains the punctuality rates $p_t(exp)$ observed online during the simulation experiments. We observe that CSAD performs worse that HARD but significantly better than PEN. A comparable behavior of CSAD and SDAD is revealed. Similar to SDAD, the punctuality falls out of the system development corridor (the gray shaded areas in Fig. 5.2) immediately after the load peak has started. SDAD is able to keep $p_t(SDAD)$ within the corridor until time $t = 2200$ but $p_t(CSAD)$ leaves the system development corridor already at time $t = 1900$. However, in both cases (CSAD and SDAD) the least observed punctuality is 0.75 and in both cases the system corridor is re-entered after a short period (around time $t = 2400$ for CSAD, around time $t = 2500$ for SDAD). After the immediate reaction to the load peak is over, CSAD maintains a higher punctuality rate than SDAD. Furthermore, $p_t(SDAD)$ leaves the system development corridor for a certain period around $t = 3100$ but $p_t(CSAD)$ remains completely within the corridor for the remaining simulation time. We conclude from these results, that both implementation strategies SDAD and CSAD are effective with respect to protecting the process timeliness in peak load situations.

Since the punctuality $p_t(CSAD)$ leaves the core of the system development corridor (the dark gray shaded area in Fig. 5.2) a non-zero error signal $e(t_i)$ defined by (5.1) is generated and implies a non-zero control signal (the intervention intensity) $h_1(p_{t_i})$ (5.2). The CSAD intervention intensity $h_{1,CSAD}(p_{t_i})$ is compared with the SDAD intervention intensity $h_{1,SDAD}(p_{t_i})$ in Fig. 5.3. In off-peak-periods, $h_{1,CSAD}(p_{t_i}))$ is remarkable smaller than $h_{1,SDAD}(p_{t_i}))$ but in an acute peak management period ($1700 \leq t \leq 2400$) the CSAD-intervention intensity is higher than the SDAD-intervention intensity.

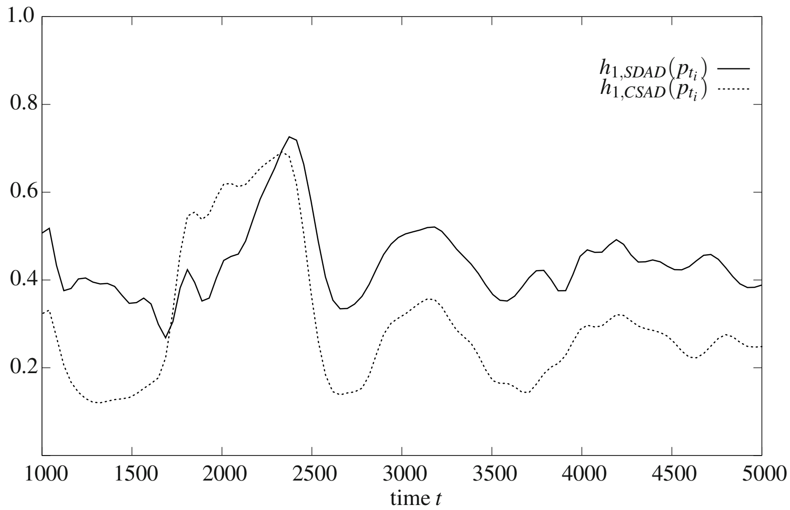

Fig. 5.3 Evolution of the intervention intensities

The increase of the intervention intensity granted to the supply consortium coordinator immediately after the beginning of the load peak contributes to keep the number of scheduled but not completed requests (pending requests) in the system on a significantly lower level than PEN is able to achieve (Fig. 5.4). If we apply HARD, SDAD or CSAD then the number of pending requests does not climb above 240 (290 pending requests are observed for PEN) and the averagely observed pre-peak number of around 70 pending requests is also re-achieved before $t = 2000$.

The analysis of the number of used own vehicles $v_t(exp)$ and of the percentage $q_t(exp)$ of externalized requests reveals the impacts of the higher intervention intensity of CSAD compared to SDAD. A higher intervention intensity lifts up the post-peak number percentage of subcontracted requests (Fig. 5.5) during the interval from $t = 1900$ until $t = 2400$. In this period the number of externalized requests is doubled compared to SDAD and quadruplicated compared to PEN and to HARD. In the same period, the application of CSAD or SDAD reduces the number of routed own vehicles (Fig. 5.6) after the peak. Compared to $v_t(SDAD)$ the number $v_t(CSAD)$ is reduced by 50% during the aforementioned period, compared with $v_t(HARD)$ it is reduced by 66% and in comparison with $v_t(PEN)$ it is reduced by 80%. In later stages of the simulation experiments, the numbers of used own vehicles $v_t(SDAD)$ and $v_t(CSAD)$ are nearly equal. However, the percentage $q_t(SDAD)$ is significantly lower than $q_t(CSAD)$.

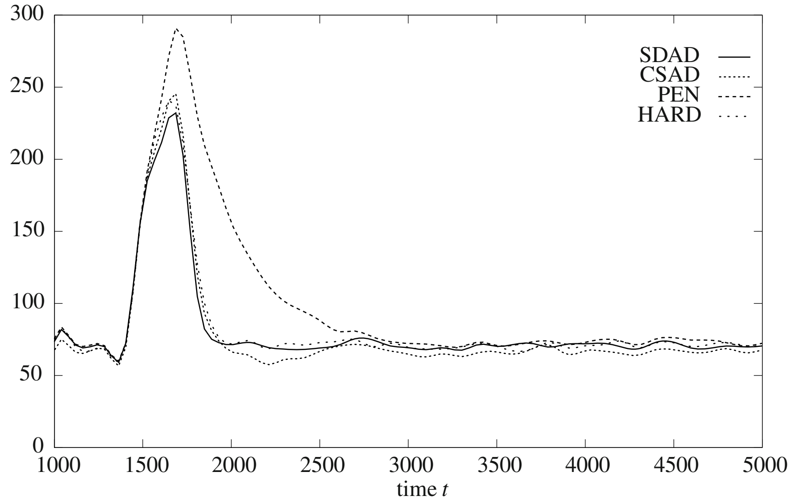

Fig. 5.4 Number of pending requests $w_t(exp)$

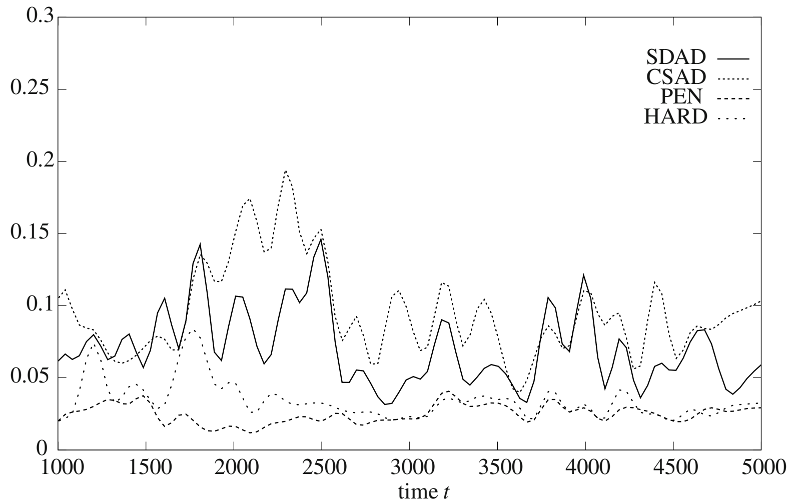

Fig. 5.5 Percentage $q_t(exp)$ of externalized requests in the schedule generated at time t

5.4.2.3 Offline Comparison of Static and Adaptive Strategies

Tab. 5.6 contains all indicator values resulting from the offline evaluation of the planning system after the simulation experiments have been finished. Again, we state that the performance of the adaptive strategies SDAD and CSAD clearly outperforms PEN but they are dominated by HARD. In detail, we first see that CSAD

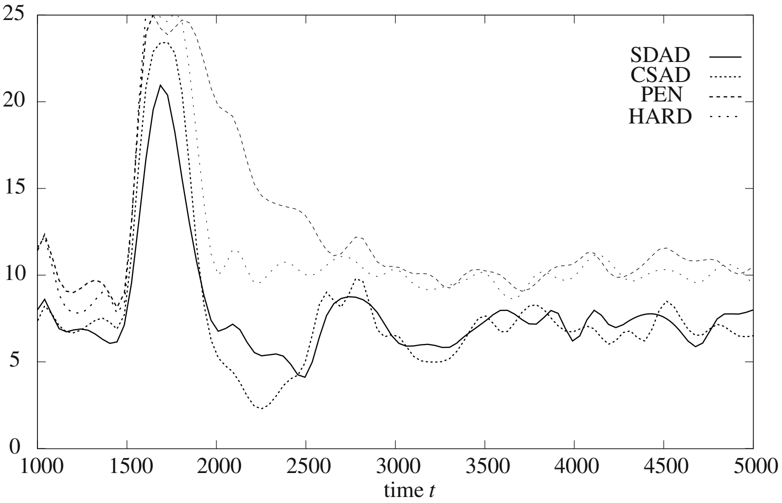

Fig. 5.6 Number of used own vehicles $v_t(exp)$

comes along with a slightly more severe maximal punctuality deviation than SDAD, e.g., $\delta(SDAD) = -5.6\% \geq -9.5\% = \delta(CSAD)$.

Next, we observe fewer LQ-situations if CSAD is used (compared to SDAD): $\pi(CSAD) = 15.0\% \leq 57.5\% = \pi(SDAD)$. Thus, CSAD produces more reliable sequences of transportation plans. Finally, we observe that CSAD is able to intervene more severely and more often draws an LSP option. It is $\sigma(CSAD) = 15.2\% \geq 14.5\% = \sigma(SDAD)$.

In conclusion, CSAD demonstrates a better performance with respect to reliability and intervention severity. However, it is outperformed by SDAD with respect to the maximal punctuality decrease after the load peak.

Table 5.6 Offline Process Quality Performance Indicator Values

	exp			
	HARD	PEN	SDAD	CSAD
$\delta(exp)$	3.5%	-38.8%	-5.6%	-9.5%
$\pi(exp)$	–	97.5%	57.5%	15.0%
$\sigma(exp)$	8.0%	4.1%	14.5%	15.2%

5.4.2.4 Online-Evaluation of the Process Costs

We have traced the average marginal costs of the completed requests throughout the simulations (Fig. 5.7). CSAD produces the highest marginal costs among all four analyzed approaches and the largest oscillation amplitude is also observed for

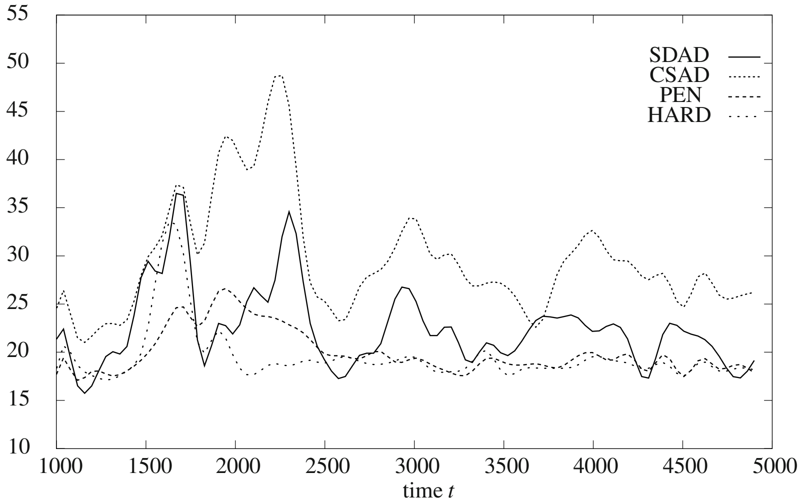

Fig. 5.7 Marginal costs $mc_t(exp)$

CSAD. The increased intervention intensity $h_{1,CSAD}(p_{t_i})$ during the past peak period $t \in [1900; 2400]$ leads to a significantly increased LSP option usage (compared the all three other strategies), which results in additional expenditures caused by the high LSP tariffs. We detect severely varying marginal costs $mc_t(SDAD)$ and $mc_t(CSAD)$ for the adaptive strategies. In contrast, the two non-adaptive strategies keep the marginal costs nearly constant on the pre-peak level after the acute demand peak is over.

In conclusion, the two adaptive strategies CSAD and SDAD cause higher marginal costs than the non-adaptive strategies HARD and PEN. In addition, CSAD is the most cost intensive strategy among the four analyzed approaches.

5.4.2.5 Offline Cost Comparison of the Four Strategies

Tab. 5.7 contains the cumulated average process execution costs observed after the completion of the simulation runs. For both non-adaptive strategies PEN and HARD, we observe nearly the same costs $c_{5000}(PEN)$ and $c_{5000}(HARD)$. We see, that the costs from the PEN-experiments are decreased by $\gamma_{5000}(PEN) = 1\%$ compared to the results achieved in the HARD-experiments. The two adaptive strategies produce significantly higher costs: $c_{5000}(SDAD) = 64225.6$ ($\gamma_{5000}(SDAD) = 14.1\%$) and $c_{5000}(CSAD) = 82699.6$ ($\gamma_{5000}(CSAD) = 46.9\%$).

Table 5.7 Offline Process Costs Performance Indicator Values

	exp			
	HARD	PEN	SDAD	CSAD
$c_{5000}(exp)$	56301.5	55748.3	64225.6	82696.6
$\gamma_{5000}(exp)$	–	-1.0%	14.1%	46.9%

5.5 Hybridization of Model Adaptation Strategies

The assessment of the two adaptive strategies SDAD and CSAD has revealed that each single adaptive strategy is capable of integrating the decision making of the coordinator agent and the subordinate transport service providing agent. In this section, we investigate the hybridization of search direction adaptation and constraint set adaption (SDCS), e.g., both implementation functions simultaneously generate a non-static model alternation signal, which depends on the current value of the intensity function (Schönberger and Kopfer, 2007a).

We start with the identification of the best parameterization of the hybrid strategy SDCS (Subsection 5.5.1). Afterwards, we compare the performance and costs of SDCS with the performance of the remaining four strategies HARD, PEN, SDAD and CSAD (Subsection 5.5.2 - Subsection 5.5.4).

5.5.1 *Parameterization of the Hybrid Strategy*

Initially, the identification of the most suitable maximal intervention intensity β for SDCS is addressed. In an offline fashion (after the termination of an experiment) we fetch the values of two process quality indicators $\delta(SDCS,\beta)$ (maximal punctuality decrease during the experiment relative to the pre-peak punctuality) and $\pi(SDCS,\beta)$ (percentage of LQ-situations). Additionally, the cumulated request fulfillment costs $c_{5000}(SDCS,\beta)$ are calculated.

Table 5.8 contains the observed values of the three observed indicators. We can see that the best process reliability is detected if the maximal β-value 1.0 is applied. The punctuality p_t declines only by 5.79% and does not leave the system development corridor (p_t remains above or equal 80% so that $\pi(SDCS,1) = 0\%$). Expectedly, the increase of the punctuality and reliability does not come for free: the cumulated request fulfillment costs $c_{5000}(SDCS,\beta)$ increase from 55748.3 money units (no intervention $\beta = 0$) up to 78350.8 money units (full intervention $\beta = 1$). Since we aim at increasing the transport process reliability in a demand peak situation, we set $\beta := 1$ for the remaining investigations.

Table 5.8 Offline evaluation of different maximal intervention values β in the SDCS-experiments

	β					
	0	0.2	0.4	0.6	0.8	1.0
$\delta(SDCS,\beta)$	-38.20%	-18.13%	-8.88%	-7.65%	-6.1%	-5.79%
$\pi(SDCS,\beta)$	100%	35%	17.5%	5.0%	2.5%	0%
$c_{5000}(SDCS,\beta)$	55748.3	62972.0	67398.4	70.782.8	73645.1	78350.8

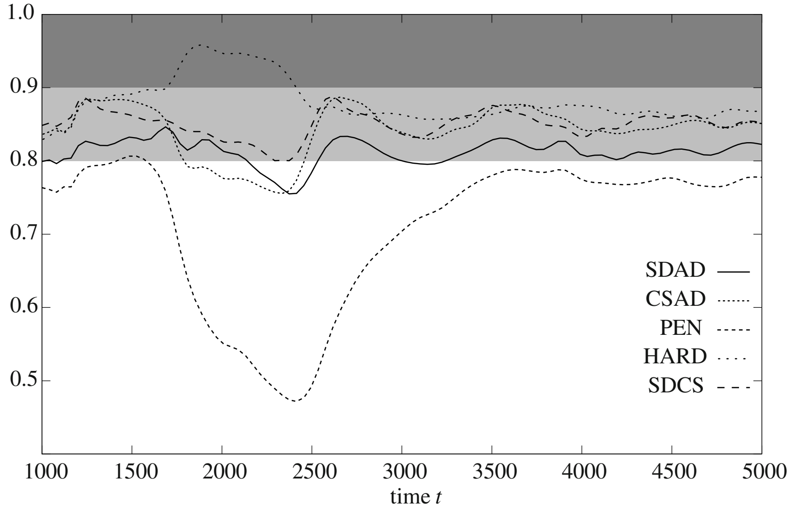

Fig. 5.8 Development of the punctuality $p_t(exp)$

5.5.2 Online Performance Comparison

The development of the punctuality p_t of all five investigated strategies is printed in Fig. 5.8. Two details are striking. At first, $p_t(SDCS)$ never leaves the system development corridor. Only a slight decrease of $p_t(SDCS)$ is observed after the peak of incoming demand/incoming requests is over. Secondly, SDCS outperforms not only PEN but also SDAD and CSAD with respect to the maintained punctuality. From these results, we conclude that the simultaneous application of both model adjustment strategies is more suitable than the application of one of the two strategies CSAD or SDAD.

In order to find out the reasons for the improved performance of SDCS (compared to SDAD and CSAD) we first analyze the intervention intensities determined by the intensity function h_1. The comparison of the intervention intensities generated during the simulation (Fig. 5.9) reveals that SDCS most often intervenes less than SDAD or CSAD. Especially during the past-peak period between time 1700 and 2500, $h_{1,SDCS}(p_{t_i})$ is significantly lower that $h_{1,SDAD}(p_{t_i})$ and/or $h_{1,CSAD}(p_{t_i})$. These observations lead to the conclusion that the interventions proposed and ap-

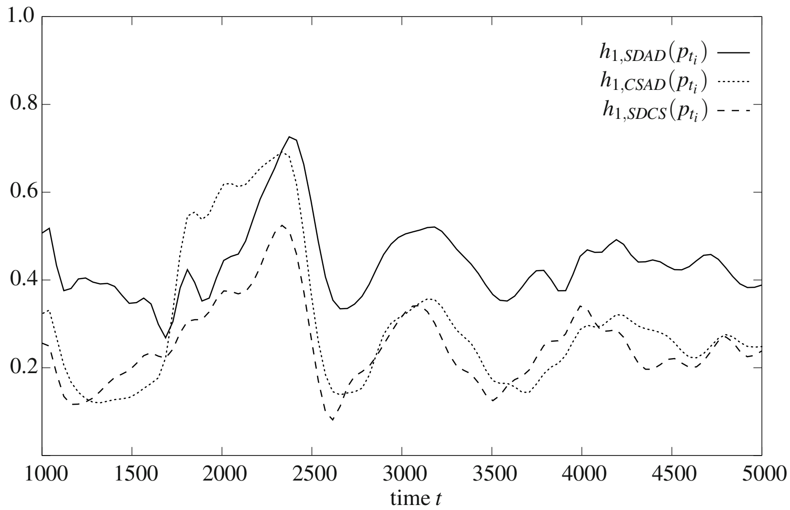

Fig. 5.9 Evolution of the intervention intensities h_1

plied by the hybrid strategy SDCS are more appropriate than the interventions proposed by SDAD and/or CSAD. More precisely, SDCS is able to identify a higher percentage $q_t(SDCS)$ of requests to be given away to an LSP immediately after the demand peak is over than SDAD does (Fig. 5.10). However, compared to CSAD, the percentage of requests selected by the coordinator for being subcontracted in the aforementioned period is lower ($q_t(SDCS) \leq q_t(CSAD)$). With respect to all other process performance criteria (number of pending requests, routed vehicles, ...) SDCS shows nearly the same values and behavior as CSAD and SDAD do. This means that the analytical capabilities of SDCS to identify portfolio incompatible requests are improved compared to CSAD. However, compared to SDAD, the strategy SDCS demonstrates a stricter and more assertive behavior: SDCS identifies more incompatible requests than SDAD, so that a higher percentage of requests are subcontracted if SDCS is applied.

5.5.3 Offline Assessment

An offline process quality and reliability assessment of SDCS reveals that the hybrid adaptive strategy SDCS outperforms PEN and the two “individual” adaptive strategies SDAD and CSAD. On the other hand, it turns out that the quality gap between the HARD strategy and the hybrid adaptive strategy is closer than the gap between HARD and SDAD or HARD and CSAD respectively.

From the results compiled in Tab. 5.9 we learn that the relative decrease $\delta(SDCS)$ of $p_t(SDCS)$ is significantly smaller than the relative punctuality rate decrease

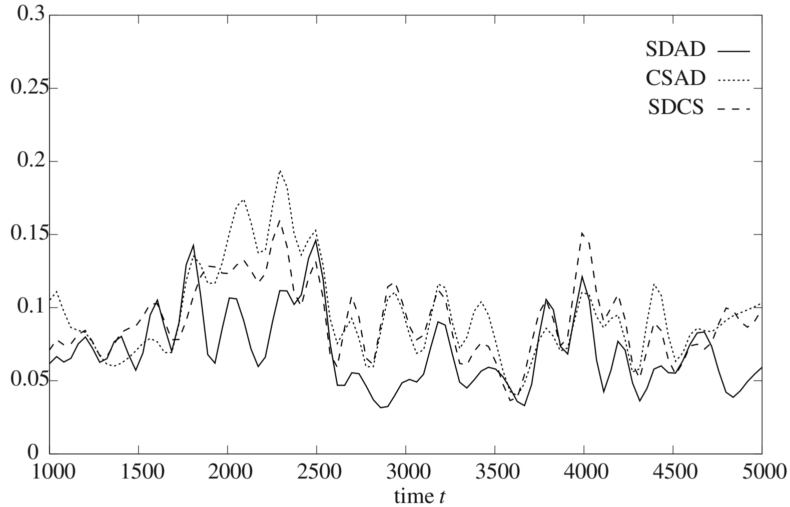

Fig. 5.10 Percentage $q_t(exp)$ of externalized requests in the schedule generated at time t

$\delta(PEN)$. In addition, the relative decrease of the punctuality in the SDCS-experiments is less than the relative punctuality rate decrease $\delta(CSAD)$. The relative decrease $\delta(SDAD)$ is only slightly less than $\delta(SDCS)$.

Similarly to the HARD strategy, also SDCS ensures that no LQ-situations occur and that the punctuality rate never leaves the intended system development corridor within the experiments. All three other strategies (PEN, SDAD and CSAD) unable to guarantee keeping the punctuality rate p_t above 80% throughout the complete simulation experiments.

Finally, we state that SDCS leads to the highest maximal externalization quota $q_t(\cdot)$ among all five applied strategies. Although SDCS combines SDAD and CSAD, the observed maximal externalization quota $\sigma(SDCS)$ does not fall between the quotas $\sigma(SDAD)$ and $\sigma(CSAD)$ but it is significantly higher than $\sigma(SDAD)$ and even higher than $\sigma(CSAD)$.

Table 5.9 Value of the indicators for the offline process quality performance

	exp				
	HARD	PEN	SDAD	CSAD	SDCS
$\delta(exp)$	3.5%	-38.8%	-5.6%	-9.5%	-5.79%
$\pi(exp)$	0%	97.5%	57.5%	15.0%	0%
$\sigma(exp)$	8.0%	4.1%	14.5%	15.2%	15.9%

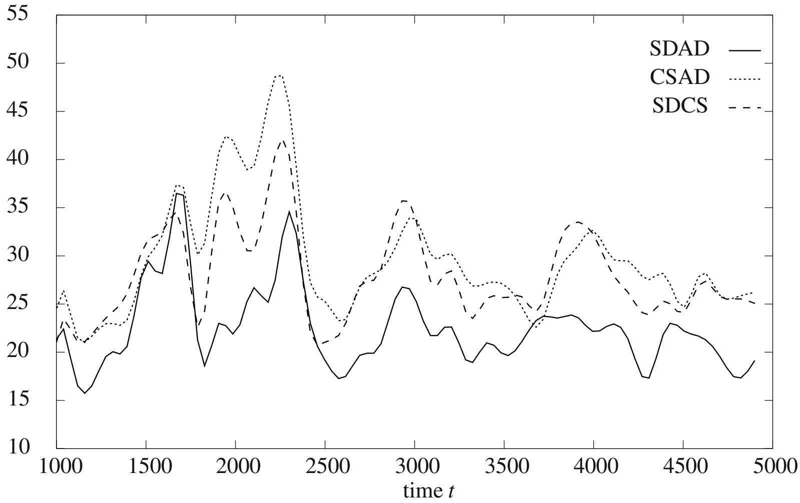

Fig. 5.11 Marginal costs $mc_t(exp)$

5.5.4 Cost Evaluation

As we have seen in the previous simulation results, the price for the improved process reliability and for the higher punctuality induced by the application of the adaptive strategies is quite high. Due to the intensified usage of expensive LSP-services, additional request fulfillment costs occur. In this subsection, we check whether the combination of SDAD and CSAD in one hybrid adaptive model controller supports keeping the additional request fulfillment costs low, e.g., we test whether we can reduce the "price of higher reliability" by combining CSAD and SDAD.

Fig. 5.11 shows the development of the marginal costs mc_t for the three adaptive strategies. The marginal costs of a request in an SDCS-controlled system are lower during the past-peak period from 1700 to 2500 then the marginal costs of a CSAD-controlled system in the same period. Due to the relatively lower usage of request subcontracting in the SDCS experiments (compared to the CSAD-cases), less expenditures have to be paid. Since the externalization quota in the SDCS-strategy is higher then the externalization quota in the SDAD-strategy, the marginal costs $mc_t(SDAD)$ are lower than $mc_t(SDCS)$ in most of the re-planning cycles.

Table 5.10 consolidates the achieved cumulated costs (c_{5000}), the cost increase compared to the HARD-experiments (γ_{5000}) and the contributions of the three cost drivers, which are travel costs ($\bar{c}^{travel}_{5000}$), penalty payments ($\bar{c}^{pen}_{5000}$) and LSP-charges ($\bar{c}^{ext}_{5000}$). The reduced marginal costs of a request finally leads to a decrease of the cumulated request fulfillment costs $c_{5000}(SDCS)$ of SDCS compared to the costs observed for the CSAD-configured scenarios. A reduction from $c_{5000}(CSAD) = 82696.6$ to $c_{5000}(SDCS) = 78350.8$ money units is observed. However, the application of SDAD leads to significantly less costs than the application of SDCS.

SDCS exhibits the least portion of travel costs c_{5000}^{travel} among all five strategies. Only 37.6% of the total sum of expenditures are used to cover the distance-related expenses of the own fleet. At the same time, SDCS causes the least penalty expenditure percentages among the investigated approaches (2.7%). On the other hand, SDCS comes along with the highest portion of LSP-charges. In the experiments, 59.7% of the total costs are used by LSP to fulfill incompatible requests. This value is maximal among the five adaptive strategies.

Table 5.10 Cumulated costs and contributions of cost drivers

	exp				
	HARD	PEN	SDAD	CSAD	SDCS
$c_{5000}(exp)$	56301.5	55748.3	64225.6	82696.6	78350.8
$\gamma_{5000}(exp)$	–	-1.0%	14.1%	46.9%	39.2%
$\bar{c}_{5000}^{travel}(exp)$	81.1%	84.5%	44.4%	40.7%	37.6%
$\bar{c}_{5000}^{pen}(exp)$	3.6%	12.0%	3.8%	3.6%	2.7%
$\bar{c}_{5000}^{ext}(exp)$	15.4%	3.5%	51.8%	55.7%	59.7%

5.6 Summary of Findings

We have developed and assessed adaptive model controllers that realize the integration of the decision making of a coordinator and of a transport service providing member of a supply consortium. At first, a suitable problem-specific system development corridor was derived. Next, an appropriate intervention intensity determining function was proposed. Finally, two different implementation functions were suggested for the investigated online optimization problem. Initially, each implementation function has been evaluated in isolation. Secondly, a hybridization of these two implementation functions has been tested.

Three lessons have been learnt from the results of the executed computational simulation experiments. On the one hand, the integration of the decision making of the two decision making actors is beneficial for the quality of the demand fulfillment offered by a supply consortium. On the other hand, a higher coordinator intervention leads to a higher fulfillment of the coordinator planning goals. Finally, the coordinator interventions induce additional process costs.

In conclusion, we have shown that the integration of image modification features into the control of supply consortium processes is possible. Applying this feature contributes towards keeping the quality of the generated transport processes on a high level even if a workload peak compromises the supply consortium operations. However, the maintaining of the high process performance does not come for free. Additional expenditures are necessary. It has to be decided whether this “price” is justified by the achieved process improvement.

Part III
Adaptive Model Controllers in Action

Chapter 6
Responsiveness Improvement

Flexibility subsumes the abilities of a value-creating or service-providing system to cope with the abruptly varying needs (requests) appearing from its uncontrollable environment (Schneeweiß and Schneider, 1999). Flexibility issues are of interest for both the system deployment level (operational planning) as well as with regard to the system configuration level (tactical planning).

Systemflexibility is a property of the investigated system that describes the general ability of the system to meet the requirements of requests released over a (representative) period of time. The determination of provided capacities as well as the selection of strategies and decision preferences for the deployment are exploited to achieve a sufficiently high degree of systemflexibility. Investing into the increase of the degree of systemflexibility during the system configuration is a kind of hedging the system against uncertainty related to future requests. It enables the usage of the system in a larger variety of environments.

Planflexibility is a property of a plan executed in the system at a specific time. In the deployment of resources, a flexible plan is able to be updated so that the requirements of additionally appearing requests are met (responsiveness). However, it depends upon the configuration of the system which additional requests can be integrated in a plan. A sufficiently high capacity must be available and the preparation of effective and efficient decision preferences for integrating additional operations into the existing plan is necessary.

Since the additional demand cannot be forecasted, the configuration of the system requires a temporal reconfiguration if the problem difficulty changes significantly, e.g. if the available capacities are exhausted. In this chapter, the exploitation of the information about the current planflexibility in the adjustment of the system configuration by varying the decision preferences used to integrate the additional demand into existing processes is proposed. We want to prove or disprove that the application of image modification has the potential to support and to improve the responsiveness of a system. In particular, the verification of the following research hypothesis is targeted: *The long term responsiveness of a (transport) system is able to be increased if the applied decision preferences for the demand integration are adjusted to the intermediately observed responsiveness (e.g., by applying image modification). The*

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*, Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_6,

better the information about the intermediate responsiveness is, the higher is the achieved degree of long term responsiveness.

The discussion of system- and planflexibility starts with a structuring of the meaning and usage of the term flexibility in the literature and a distinction between flexibility of a system and of a schedule (Section 6.1). In order to quantify the flexibility-related properties of a system or a schedule, we propose flexibility measurements (Section 6.2). We add the flexibility measurements into our portfolio of performance indicators and report flexibility-related results observed in the performed computational simulation experiments (Section 6.3).

6.1 Flexibility and Logistic Operations

A compilation of the scientific discussion of the meaning of the term flexibility is given in Subsection 6.1.1. The major conclusion from this literature analysis is that it is useful to distinguish flexibility as a property of a system and as a property of a specific plan. Thus, we propose flexibility definitions from the viewpoint of a system and from the viewpoint of a schedule (Subsection 6.1.2).

6.1.1 Literature Review

Flexibility-related investigations explore the range of different exogenous demand under which a given system performs well or "as intended" or "as required". Furthermore, bottlenecks that limit the system's ability to adjust to the changed exogenous demand situation are searched for. The goal is to analyze and widen or bypass bottlenecks, so that the responsiveness of the system increases with respect to the need originating from the system environment.

De Groote (1994) proposes to distinguish (i) the investigated system ("technology"); (ii) the system environment and (iii) one or several evaluation criteria to evaluate the behavior of a system in its environment. Most of the work related to (i) and (ii) is linked to applications from production or inventory planning (Beach et al., 2000; Pibernik, 2002; de Toni and Tonchia, 1998; Sethi and Sethi, 1990; Gupta and Goyal, 1989). Uncertainty about environmental needs is connected with the inability to forecast the demand time as well as the demand volume. Morlok and Chang (2004) also consider the uncertainty of the locations of demand for rail freight transport and thus cover the third dimension of uncertainty, which is the spatial uncertainty of demand. Different transport network types are compared by Feitelson and Salomon (2000) with regard to different non-numeric flexibility criteria. Flexibility is measured by counting the variety of a system's possible reactions to a particular environmental need, by technical indicators or by the costs to enable the system to fulfill the requirements of a new environmental demand (Schneeweiß and Schneider, 1999).

Since a system evolves and changes its configuration over time and the flexibility property is given only for specific configurations and/or a specific time it becomes necessary to add specific information to a flexibility statement: the time (representing the system state in which the flexibility property was observed), the description of the disturbance for which the adaptability has been investigated as well as the rules for the integration of additional requests (integration logic and decision preferences).

6.1.2 Flexible Plans and Flexible Systems

Let $S = (C, R)$ be a system consisting of a set C of system components and a set R of possible relations between the components. The components out of a subset of C are connected by means of activated relations taken from R at time t. This situation is referred to as **state** $\mathbf{X}(S,t)$ of system S at time t.

A system control unit is able to change the state of S from $X(S,t')$ to $X(S,t^*)$ with $t' \leq t^*$. The transformation of the system during the period from t' to t^* is described by the **transformation plan** $\mathbf{P}[X(S,t'),X(S,t^*)]$. The intermediate state of S during the transformation according to this plan at time t is $P[X(S,t'),X(S,t^*)](t)$.

An **environmental need (request)** is an exogenous demand whose appearance (in time, spatial variability and demand intensity) cannot be controlled by the system's control unit. Each request claims the fulfillment of specific requirements, that are fulfilled by the current system state or not. If and only if a request e appears at time t and if $X(S,t)$ fulfills the requirements of e then $\mathbf{X}(\mathbf{S},\mathbf{t})$ **is compatible with e at time t**. A reconfiguration (transformation of the state) of S is not necessary. The compatibility of a system state comprises any requirements like time constraints (time windows), sufficient volume or the meeting of logical dependencies. If a request e is not compatible with $X(S,t)$ then it is called a **disturbance of S at time t**.

6.1.2.1 Planflexibility

The basic idea for defining the flexibility of a plan with regard to an additional request e appearing at time t is to check, whether the so far unexecuted part of this plan can be replaced by another transformation plan so that after the completion of the updated plan also the requirements of e are fulfilled. If such an update of the performed transformation plan exists then the plan is denoted as **e-planflexible**.

Let $P[X(S,t'),X(S,t^*)]$ be the currently processed transformation plan for the reconfiguration of S from $X(S,t')$ to $X(S,t^*)$. The request e appearing at time t is a disturbance of S at time t, because it is not compatible with the intended final configuration $X(S,t^*)$ of S so that the reconfiguration of S into the state $X(S,t^*)$ is void now. A transformation of S from the current configuration $P[X(S,t'),X(S,t^*)](t)$ into a new target configuration which is also compatible with e is necessary.

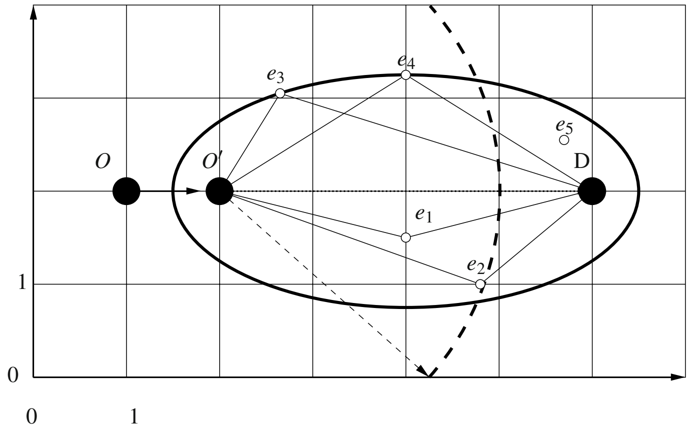

Fig. 6.1 An example scenario from transportation planning with compatible requests (e_1-e_4) and an incompatible request (e_5)

If and only if there is a transformation plan Q converting the current system configuration $P[X(S,t'),X(S,t^*)](t)$ into a configuration $Z_e(S,t(e))$ that is compatible also with e then the so far followed transformation plan is adaptable at time t with regard to the disturbance e. If such a transformation plan Q exists then the plan $P[X(S,t'),X(S,t^*)]$ is e-planflexible at time t.

By means of the situation outlined in Fig. 6.1 the meaning of planflexibility is explained. The system S comprises a vehicle (server) that travels in the plane and fulfills requests by visiting a customer site associated with the request. Its travel speed is 1 length unit per time unit.

The server waits at position O at time 0. A request has been released and this request requires a server's visit at position D within the time window $[5;6]$. The current position of the vehicle and the request are incompatible: For this reason, the vehicle starts to move towards the location D following the bold arc representing the transformation plan P. According to its schedule, the server will reach D on a timely basis at time 5. At time 1, a second request appears which requires the visit to the customer site e within the time window $[3;4]$. The position of the server is now O'. If the customer site associated with e is situated in the intersection of the ellipsis and the circle indicated by the dashed radius in Fig. 6.1, then both the additional customer site (e_1, e_2, e_3 or e_4) and D can be reached within the agreed time windows. In all other cases (e.g., e_5) it is impossible to visit both customer sites without a delay. If e is situated outside the circle then its own time window is violated but if e lies outside the ellipsis then the time window associated with the first request cannot be met.

The set E is formed by all additional customer requests with time window equal to $[3;4]$ and a location belonging to the previously mentioned intersection. If and only if P is e-planflexible at time 1 for all requests belonging to E then the plan P is denoted as E-planflexible at time 1.

In the given example the additional customer site request is inserted into the existing plan so that a minimal overall travel distance must be bridged. Therefore, the integration of the additional request follows the decision preference "least sum of travel distances". An application of another preference or integration logic like "first-come/first-serve" cannot guarantee the E-planflexibility of P at time 1. It is therefore necessary to add information about the applied decision preferences V to the planflexibility statement. The refined planflexibility definition is now:

If and only if there is a transformation plan Q which has been generated according to a given set of decision preferences V and that converts the system configuration $P[X(S,t'),X(S,t^*)](t)$ into a configuration $Z_e(S,t(e))$ compatible also with e then the so far performed transformation plan is adaptable at time t with respect to the disturbance e by utilizing the decision preferences V. If such a transformation plan Q exists then $P[X(S,t'),X(S,t^*)]$ is called e-planflexible at time t with respect to V.

6.1.2.2 Systemflexibility

Planflexibility is not able to inform about the ability of the decision preferences V to successfully integrate any additional request within a given time period into the so far followed transformation plan. To describe such general adaptation ability for the system S, the concept of systemflexibility is introduced. Systemflexibility describes the system's ability to react to any environmental need in a given period without having information about the explicit followed plan if the application of decision preferences V is obligatory.

The system S is called **e-systemflexible** at time t under the decision preferences V, if and only if there is a state $X(S,t^*)$ with $t' \leq t^*$, so that $X(S,t^*)$ and e are compatible. This definition does not exploit any information about the structure of S. It is not required that S is compatible with e at the appearance time t of e but it is necessary that S can be transformed into a state that is compatible with the request e. Systemflexibility describes the system's ability to change its state in order to meet the requirements of an environmental demand ("action volume" according to Schneeweiß and Schneider (1999)). The "reactivity" of S determines the speed of this change. In the event that the adjustment time of S after the appearance of e is limited, e.g. if for every constant $K > 0$ the requirement $t \leq t^* \leq t+K$ is kept, then the system S is called **real-time e-systemflexible** at time t under the decision preferences V.

This definition of systemflexibility generalizes the "system capacity flexibility" of Morlok and Chang (2004) who state that "system capacity flexibility" is the ability of a transport system to accommodate variations or changes in traffic demand while maintaining a satisfactory level of performance.

At a particular time, e-systemflexibility might be observed for different requests e. In order to enable a compact description of the systemflexibility-property at a given time t for a variety of requests, the set U formed by all these requests is introduced. The system S is called U-systemflexible at time t under the decision preferences V, if and only if S is e-systemflexible at time t under the decision preferences V for all $e \in U$. In order to enable a more general description of the systems ability to meet the demand of environmental needs, the restriction that the environmental need must appear at a particular time t is dropped in the following. Instead it is assumed that it appears in a period T. If S is U-systemflexible at time t under the decision preferences V at any time $t \in T$, then the system S is called U-systemflexible during T under the decision preferences V.

6.2 Quantification of Flexibility

The consideration of responsiveness issues during the deployment preparation as well as the configuration of a system enables the creation of planflexible plans and schedules and systemflexible systems. However, it is necessary to quantify planflexibility as well as systemflexibility. The definition of a scalar or of a vector of scalars to represent the degree of flexibility is a prerequisite for a comparison of several deployment or configuration alternatives with respect to their responsiveness. Quantification requires knowledge about the totality of possible requests and disturbances and a feedback about those requests for which the associated requirements can be met. No assumptions about internal structures of the system S are made. Thus, only the comparison of the total demand the system is faced with and the demand satisfied as request by S ("output flexibility" according to Grubbström and Olhager (1997)) can be exploited.

A plan or a system might be able to appropriately incorporate no (additional) requests (0%), all additional requests (100%) or a subset of future requests. Therefore, Schneeweiß and Schneider (1999), Barad and Sapir (2003) as well as Corsten and Gössinger (2006) propose to quantify flexibility by the degree of satisfaction of the requirements of additional requests which is equivalent to the relative frequency that an occurred request can be integrated appropriately.

6.2.1 *Measures for Planflexibility*

The planflexibility of a plan P refers to the ability of the decision preference V to integrate one or more additional requests into P at a particular time t by replacing the so far unexecuted part of P.

Let N^A be the overall number of all requests that could appear at time t jeopardizing the current plan $P[X,Y]$, n^A be the number of these requests e for which $P[X,Y]$ is e-planflexible at t and E contains all requests that could appear at time t. Then the A-degree for E-planflexibility of $P[X,Y]$ at time t under V is defined in (6.1).

$$F^A(V,E,P[X,Y],t) := \frac{n^A}{N^A}. \tag{6.1}$$

The integration of additional requests into the plan P affects the already included operations as well as the previously made assignments of tasks to resources and therefore makes the re-scheduling of execution times necessary. The A-planflexibility does not consider these crowding-out effects, since it does not consider whether the requirements of previously integrated environmental needs are still met. Let N^B be the overall number of all requests contained in P and let n^B be the number of those requests whose requirements are all satisfied at time t. The B-degree of E-planflexibility of $P[X,Y]$ at time t in the event that the decision preference(s) V are applied is defined as in (6.2).

$$F^B(V,E,P[X,Y],t) := \frac{n^B}{N^B}. \tag{6.2}$$

Neither the A-degree nor the B-degree for planflexibility considers already completed requests within the calculation of the relative frequency (probability) for the satisfaction of the requirements associated with a request.

The inclusion of recently fulfilled requests in the calculation of the flexibility degree prevents an overweighting of temporal demand peaks. On the other hand, if the fulfillment time of a request is scheduled far from now, then information about the satisfaction of the associated requirements is of reduced worth. It is reasonable to consider only those requests, whose completion is scheduled for the near future. In the C-degree for E-planflexibility of P at t the satisfied demand is compared with the total demand that falls into the time window $[t-\Delta t; t+\Delta t]$. Let N^C be the number of all requests completed within the period $[t-\Delta t; t]$ or scheduled to be completed during the period $]t; t+\Delta t]$ and let n^C be the number of those requests for which all requirements have been satisfied or are expected to be satisfied. The C-degree for E-planflexibility of P at time t in the event that the decision preferences V are applied is defined in (6.3).

$$F^C(V,E,P[X,Y],t) := \frac{n^C}{N^C} \tag{6.3}$$

6.2.2 Systemflexibility Quantification

Barad and Sapir (2003) define the logistics dependability of a system as the probability that all requirements of the released requests are fulfilled by using the system S during an observation period. This idea is taken up and the systemflexibility-degree for U-systemflexibility during T of S is similarly defined. Let M denote the number of all possible requests belonging to U and appearing during the period T. If system S can handle m requests appearing during T as requested applying the decision preferences V to integrate the requests then the systemflexibility-degree for U-systemflexibility of S during T, denoted as $F^{system}(V,U,T)$, is defined by (6.4).

$$F^{system}(V,U,T) := \frac{m}{M}. \tag{6.4}$$

Using the vocabulary introduced from the concepts of planflexibility as well as systemflexibility and recognizing the decision preferences as part of the systems configuration, the initially stated research hypothesis is refined: *The U-systemflexibility of a (transport) system during a period T is able to be increased if the decision preferences V of this system are adapted to the intermediately calculated degree of planflexibility. The B-degree-application leads to a higher F^{system}-value than the A-degree-application and the C-degree-application leads to a higher F^{system}-value than the B-degree-application.*

It is self-evident that this hypothesis cannot be verified for any transport system but the general correctness of the idea to exploit planflexibility knowledge to adjust the process planning system can be demonstrated by simulating the transport system outlined in Section 2.3. We use adaptive model controllers proposed in Chapter 5 to automatically adjust the process control system (the decision preferences). The currently observed degree of planflexibility is used as input parameter for the intensity function of the process control system.

6.3 Computational Experiments

We check the appropriateness of the proposed flexibility concept within computational simulation experiments. In Subsection 6.3.1, the setup of the simulation environment and of the executed experiments are described. A presentation and discussion of the observed results is given in Subsection 6.3.2.

6.3.1 Experimental Setup

The requirements of a transport request are met if the request is completed within its associated time window. We configure the introduced plan- and systemflexibility measures (6.1) - (6.4) for the online vehicle routing problem introduced in Section 2.3. The online optimization framework outlined in Section 3.2 is used to automatically update the generated transportation plans in response to the additionally arrived requests. We evaluate this process control system with and without adaptive model controllers in order to reveal the impacts of the application of model adjustments with respect to the achievement of flexibility.

As reference approach we use the HARD strategy. In addition, we repeat the experiments for PEN and for the three adaptive techniques SDAD, CSAD and SDCS.

The scenarios introduced in Subsection 5.4.1 are simulated. Again, the subcontracting rate is three times larger than the sum of penalty payments and travel costs. Therefore, the transport system control unit tends to refrain from the subcontracting of requests but accepts delays. Each scenario for the transport system has been simulated over a period of 5000 time units. After a startup phase of 1000 time units the planflexibility degree has been recorded throughout the next 4000 time units while every 100 time units additional requests have been released. In the investigated artificial scenario U is defined to be the set of all possible additional requests, so that this information can be dropped in the planflexibility statements. Each simulation run has been executed several times with different seedings because the memetic dispatching algorithm is a randomized procedure. Here, we report about the average results observed at particular times, so that also the information about the particularly updated plan in the planflexibility statement is dropped.

The degrees $F^A(V,t)$, $F^B(V,t)$ and $F^C(V,t)$ are calculated for $t = 1000$, 1100, $1200, \ldots, 5000$ in separate experiments applying $V \in \{HARD, PEN, CSAD, SDAD, SDCS\}$. We set up three classes of experiments. In the first one ("A-degree"), the value $F^A(V,t)$ is used as input parameter for the intensity function h_β that controls SDAD as well as CSAD and SDCS. In the second class ("B-degree") $F^B(V,t)$ determines the model controller input and in the third class ("C-degree") $F^C(V,t) = p_t$ is used to control SDAD, CSAD and SDCS.

Let $max(CSAD,A)$ denote the maximal value observed for the A-planflexibility degree $F^A(CSAD,t)$ during the period [1000,5000] and let $min(CSAD,A)$ denote the minimal value observed for $F^B(CSAD,t)$ within this time period. Furthermore, $var(CSAD,A)$ is defined as the difference between $max(CSAD,A)$ and $min(CSAD,A)$. The indicator $bel(CSAD,A)$ gives the percentage of re-planning cycles in the interval [1000,5000] in which $F^A(CSAD,t)$ has fallen below 0.8. Finally, we report $\pi(CSAD,A)$ which represents the percentage of LQ-states. The same values have been calculated for the results observed in the SDAD-, the SDCS as well as for the results from the non-adaptive PEN- and HARD-experiments. After having reconfigured the planning system from the application of the A-degree to the B-degree and, furthermore, after changing from the B-degree to the C-degree we re-evaluate the indicator values.

At the end of a simulated scenario, the quotient between the number of requests served within the assigned time window and the overall number of requests released during the observation period [1000,5000] has been calculated for the configuration with A-, B- and $C-$planflexibility degree which represents the observed systemflexibility degree $F^{system}(\cdot,\cdot)$ during this period.

6.3.2 Results

All calculated key indicator values for the planflexibility degrees from the experiments are presented in Tab. 6.1. The evaluation results from the HARD simulations serve as referential values. Here, the target punctuality $p^{target} = 0.8$ is met in every re-planning cycle ($\pi(HARD, none) = 0$). However, the HARD approach is not applicable in supply network consortia. As a compromise, PEN is used to generate the processes. As we have already seen in the previous experiments, it is unable to generate process updates that meet the target punctuality in load peak situations. From the viewpoint of responsiveness, PEN does not support ensuring a sufficiently high responsiveness and a PEN-controlled system is of low systemflexibility.

The application of an adaptive approach (SDAD, CSAD or SDCS) leads to an improvement (reduction) of the number the LQ-periods. Even if the feedback signal is compiled only from the punctuality degree observed for the recently arrived requests (A-planflexibility degree), an increase of π is observed. This means more requests can be served in time. A small improvement is observed for SDAD. Here, we have $\pi(SDAD, A) = 92.7\%$ which is lower than $\pi(none) = 97.6\%$. For CSAD and SDAD, a π-value of 70.7% is achieved. The realization of further improvements of the π-values is impeded by the inability of the feedback signal to warn about a punctuality degree decline. From the values in the third and fourth column we learn that the feedback signal oscillates around a value which is far away from the target punctuality. This is mainly caused by the fact that the A-degree considers only the punctuality of new requests while the punctuality of already scheduled requests is ignored. Consequently, no suitable error signal can be generated.

In order to improve the responsiveness of the generated schedules, we replace the A-planflexibility degree by the B-planflexibility degree. This new degree also considers already scheduled (but not yet completed) requests in the feedback signal generation. We observe a significant reduction of the number of LQ-periods if the B-planflexibility degree is incorporated. Now, only 43.9% re-planning cycles do not meet the target punctuality (compared to 92.7% in the A-planflexibility experiments). If CSAD is used, then only 29.3% (so far 70.7%) updates do not come along with a percentage of 80% or more. A further reduction of π is achieved if we use SDCS. Now, only 24.4% of the generated updates violate the target punctuality condition (compared to 70.7% before). The B-planflexibility degree is a better indicator for variations in the process performance. We observed a quite active oscillation of the feedback signal around the target value 80% (third and fourth column).

If we also consider recently completed requests into the generation of the feedback signal (C-planflexibility degree) then we improve the π-values once again. In only 17% of all re-planning cycles are schedules with a punctuality rate below p^{target} generated. The least punctuality requirement is met in all re-planning cycles if the hybrid adaptive approach SDCS is used. Since the feedback value closely oscillates around the target value (third and fourth column in the C-degree section), a quite appropriate feedback signal is generated that adequately indicates a decrease of the process punctuality.

We finally analyze the systemflexibility achieved by applying the three feedback signals. These values are summarized in Tab. 6.2. Again, we use the results achieved by applying the non-adaptive strategies as upper (HARD) and lower (PEN) reference values. The application of HARD leads to the best (highest) systemflexibility degree of 89.8% and PEN exhibits the worst systemflexibility degree (74.1%). Independent of the applied adaptive model controller, we observe a systemflexibility in-

Table 6.1 Observed degrees of planflexibility

type	V	$max(\cdot,\cdot)$	$min(\cdot,\cdot)$	$\mathrm{var}(\cdot,\cdot)$	$bel(\cdot,\cdot)$	$\pi(\cdot,\cdot)$
none	HARD	99.5%	88%	7.5%	0%	0%
	PEN	88.1%	30.5%	57.6%	97.6%	97.6%
A-degree	SDAD	95.8%	82.6%	13.2%	0%	92.7%
	CSAD	95.5%	81.3%	14.2%	0%	70.7%
	SDCS	96.0%	81.3%	15.3%	0%	70.7%
B-degree	SDAD	90.6%	73.3%	17.3%	4.8%	43.9%
	CSAD	91.3%	67.0%	24.3%	9.7%	29.3%
	SDCS	90.5%	76.1%	14.4%	4.9%	24.4%
C-degree	SDAD	84.6%	72.4%	9.2%	17.0%	
	CSAD	87.4%	75.3%	12.1%	17.1%	
	SDCS	94.6%	75.2%	19.4%	0%	

crease if we replace the A-degree by the B-degree or if we use the C-degree instead of the B-planflexibility degree. CSAD and SDAD demonstrate a similar systemflexibility degree varying from $F^{sys}(CSAD,A) = 79.2$ respectively $F^{sys}(SDAD,A) = 79.9\%$ to $F^{sys}(CSAD,C) = 81.1\%$ respectively $F^{sys}(SDAD,C) = 83.3\%$. Higher systemflexibility degrees are observed if the hybrid and adaptive strategy SDCS is applied. Then, the systemflexibility degree increases from $F^{sys}(SDCS,A) = 81.1\%$ to $F^{sys}(SDCS,B) = 82.8\%$ and to $F^{sys}(SDCS,C) = 85.5$. The last value is very close to the referential value $F^{sys}(HARD) = 89.8\%$.

The presented results and the described observations verify the initially stated research hypothesis: It has been shown that the systemflexibility is increased if the decision preferences, which are a part of the system configuration and which are coded in the model (2.2)-(2.7) and (2.9), are adapted continuously to the intermediate degree of planflexibility. Furthermore, it has been demonstrated that the systemflexibility increases if the planflexibility measurements become more sophisticated and if more local information is used in the reflected process feedback signal.

Table 6.2 Observed degree of systemflexibility using different planflexibility measures

V	A-degree	B-degree	C-degree
HARD	89.8%	89.8%	89.8%
PEN	74.1%	74.1%	74.1%
CSAD	79.2%	82.3%	83.8%
SDAD	79.9%	82.0%	83.3%
SDCS	81.1%	82.8%	85.5%

6.4 Conclusion of Findings

The responsiveness of a transport system operating in a volatile environment can be increased if adaptive model controllers are incorporated into the process control.

Flexibility degrees have been defined from the short term perspective (planflexibility) and from the long term perspective (systemflexibility). The initially stated research hypothesis has been verified for the investigated system. We have proven that, the adaptation of the transport system's configuration (particularly the applied deployment decision preferences) to the particular planflexibility degrees leads to an increase of the systemflexibility.

With respect to principal-agent relationships the observed simulation results lead to the following conclusion. Interventions of a superior supply consortium coordinator into the deployment decision-making of a subordinate service providing agent are beneficial. They lead to an increased responsiveness of the consortium. The intensity of the intervention should be small if the current planflexibility degree is high. As soon as the responsiveness of the agent-generated schedules starts declining the superior coordinator should increase the intensity of its interventions. If this simple rule is applied then the incorporation of adaptive model controllers contributes to an improvement of the interaction between a principal and a subordinate agent, so that the responsiveness of the supply consortium is increased.

Chapter 7
Nervousness Reduction in Re-Scheduling

A logistic system can be considered as a black-box-system S that transforms a given input signal (requests) into an output signal (logistics processes). A major challenge in the management of logistic systems is to keep the quality of the output on a high and balanced level even if the input signal oscillates with large amplitude. If the system is able to fulfill this property then it is called responsive or systemflexible. Responsiveness and systemflexibility refer to the degree to which changes in the system's environment can be compensated by modifications of the scheduled operations.

In order to cope with the uncertainty of future planning data, the system coordination unit sets up tentative schedules considering all data known at the schedule generation time. The execution of a tentative schedule is started but interrupted immediately if additional planning data are released. Now, previously made operation scheduling decisions are revised in order to integrate the operations associated with the additional requests into the processes. If the frequency of scheduling decision revisions increases then the acceptance of and the trust in a once made decision decreases. This planning instability phenomenon is referred to as *schedule nervousness* or *schedule instability* (Kadipasaoglu and Sridharan, 1997).

Interdependence between responsiveness and nervousness is obvious. In order to be responsive it is necessary to revise a given schedule. However, the executed revisions must be chosen very carefully in order to keep the schedule nervousness on a low level. Within this chapter, we investigate the nervousness observed in the online vehicle routing problem with time windows introduced in Section 2.3. Furthermore, we analyze the impacts of applying image modification from the viewpoint of stability and nervousness. A verification or disproof of the following research hypothesis is addressed: *The nervousness observed in the transport system is reduced if the utilization degree of the subcontracting request fulfillment mode is adapted to the intermediately detected responsiveness of a transport system (e.g., by coordinator interventions into the deployment decision derivation).*

Planning stability and instability issues are investigated from a general perspective in Jensen (1996). Nervousness in inventory management is addressed in de Kok

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*, Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_7,

and Inderfurth (1997) and Inderfurth (1994) but nervousness in production planning is investigated by Inderfurth and Jensen (1997).

Nervousness is a symptom appearing during the transition from the so far followed schedule to an updated schedule after additional requests appeared. The first mentioned schedule will be called *preschedule* (Jensen, 2001b) and the latter one is referred to as *new schedule* in the remainder of this chapter.

The comparison between a preschedule and the associated new schedule reveals several differences. In the following, the comparison of these two concatenated schedules is discussed.

The organization of this chapter is as follows. Section 7.1 introduces nervousness issues that affect customers and nervousness issues that are of special interest only for the process planning unit. Three nervousness degrees relevant for transport processes are proposed in Section 7.2. A system nervousness measure is derived in Section 7.3. The relationship between responsiveness and stability is addressed in Section 7.4. Simulation experiments, in which the proposed quantification concepts are assessed, are reported in Section 7.5.

7.1 External and Internal Nervousness

Some revisions of scheduled operations directly affect the customers served by the logistic system S. Typical examples are brought forward or postponed arrival times of transport vehicles at customer sites to pickup or deliver goods or sending repair, maintenance or emergency response teams. Nervousness or instability of data already given to the customers is referred to as *external nervousness*.

Other revisions do not affect the customer or do not receive the customer's attention. The instability of these decisions is called *internal nervousness*. A typical example is the re-assignment of operations to another resource without modification of the completion times of once scheduled operations.

The prevention of external nervousness is as important as the prevention of internal nervousness. In the first case, the satisfaction of the customers is endangered but in the second case, additional setup (loading) costs and start-up costs occur.

7.2 Schedule Transition Nervousness

Let P be a preschedule and Q the associated new schedule. The preschedule consists of a set of operations for which several decisions have been determined at time t_p. For each operation belonging to P, a starting time has been fixed, resource capacities has been reserved (corresponding to the volume that is handled by this operation) and the location(s) where the operation is executed has (have) been determined.

All operations which have not been completed before the new schedule Q is set up at time t_Q are also contained in Q. However, the assigned starting times, the volumes of the reserved capacity and the assigned locations have been checked for compatibility with the new requests. If the so far made decisions and the new requests are not compatible then these decisions are revised: an earlier or later starting time is assigned to one or more operations, the volume to be handled by this operation is increased or reduced and the involved locations are subject of revision.

In the event that no decision contained in the preschedule is revised in the new schedule, the preschedule is completely stable. If all decisions made for the preschedule are revised in the new schedule then the preschedule is instable. However, in most of the transitions from a preschedule to a new schedule only a fraction of the operations contained in both consecutive schedules are revised. This observation leads to the following general nervousness definition. Let $N_{P,Q}$ be the number of operations contained in both schedules and let $n_{P,Q,\mathscr{V}}$ be the number of operations subject of a revision during the transition from P to Q using the update preference $\mathscr{V}$. Now, the **schedule nervousness degree $\mathbf{deg}_{\mathscr{V}}(\mathbf{P},\mathbf{Q})$ using update preferences** $\mathscr{V}$ for the transition from P to Q is generally defined in (7.1).

$$deg_{\mathscr{V}}(P,Q) := \frac{n_{P,Q,\mathscr{V}}}{N_{P,Q}} \tag{7.1}$$

Often, only particular decisions are observed during the transition from a preschedule to a new schedule but the definition of the specific schedule nervousness degree remains the same. In transport scheduling the first decision task to be carried out for each schedule revision is the selection of the right fulfillment mode for each request (self-fulfillment and subcontracting). **M**ode **S**election **N**ervousness (MSN) quantifies the variations in the mode decisions between the preschedule P and the new schedule Q. Let $N^{MSN}_{P,Q}$ be the number of all requests contained in P as well as in Q whose fulfillment mode is allowed to be altered. Furthermore, let $n^{MSN}_{P,Q,\mathscr{V}}$ be the number of requests for which the selected fulfillment mode is different in P and Q after $\mathscr{V}$ has been applied to update P. Then, the degree of MSN observed in the transition from P to Q using the update preferences $\mathscr{V}$ is defined in (7.2).

$$MSN_{\mathscr{V}}(P,Q) := \frac{n^{MSN}_{P,Q,\mathscr{V}}}{N^{MSN}_{P,Q}}. \tag{7.2}$$

The second decision task in transport schedule generation is the assignment of the operations associated with a request to available resources. Those requests, for which the self-fulfillment has been selected are distributed among the vehicles and a logistic service provider is selected for all subcontracted requests. In operational freight transport there are two kinds of resources: subcontractors and owned vehicles (Schönberger, 2005). The decision to subcontract a request cannot be revised but a request that has been assigned to a certain owned vehicle is allowed to be reassigned to another owned vehicle during a plan update. Let $N^{RAN}_{P,Q}$ be the number of all requests which can be re-assigned from an owned vehicle to another owned vehicles during the transition from P to Q. The expression $n^{RAN}_{P,Q,\mathscr{V}}$ contains the number of requests which have been assigned to different owned vehicles in P and Q by $\mathscr{V}$.

We define the degree $RAN_{\mathscr{V}}(P,Q)$ of **R**esource **A**ssignment **N**ervousness (RAN) in the transition from P to Q associated with the update preferences $\mathscr{V}$ in (7.3).

$$RAN_{\mathscr{V}}(P,Q) := \frac{n^{RAN}_{P,Q,\mathscr{V}}}{N^{RAN}_{P,Q}}. \tag{7.3}$$

Both measures MSN as well as RAN quantify internal nervousness. Neither a mode change nor a resource variation might be of interest for the associated customers. In both cases it is possible to keep a once announced request completion time, so that the synchronization of the transport processes with the internal processes of the customers is preserved. In contrast, the variation of announced request completion times represents an external nervousness issue. The symbol $N^{ATN}_{P,Q}$ denotes the number of all requests for which an arrival time shifting is allowed. If $n^{ATN}_{P,Q,\mathscr{V}}$ represents the number of the requests whose completion time is revised during the transition from P to Q, then the **A**rrival **T**ime **N**ervousness (ATN) using $\mathscr{V}$ is defined in (7.4).

$$ATN_{\mathscr{V}}(P,Q) := \frac{n^{ATN}_{P,Q,\mathscr{V}}}{N^{ATN}_{P,Q}}. \tag{7.4}$$

7.3 Transport System Nervousness

MSN, RAN and ATN describe and quantify the instability of a particular decision with respect to a specific schedule update. A more general quantification is necessary to describe the stability / instability of the system S during a longer period T in which several updates are carried out. Instead of observing and counting the number of revised decisions during the transition from P to Q (at a given time t_Q), it is necessary to consolidate the executed schedule revisions during the generation of the concatenated sequence of schedules $P_i, P_{i+1}, \ldots, P_{i+k}$ whose update times fall into T. Let $M_{S,T,\mathscr{V}}$ denote the number of all update decisions that must be made during period T and let $m_{S,T,\mathscr{V}}$ be the number of all changes of a decision during the transition from a preschedule to a new schedule during the period T. Then, the **degree $\mathbf{deg^{sys}_{\mathscr{V}}(S,T)}$ of nervousness of system S during period T using update preferences $\mathscr{V}$** is calculated by

$$deg^{sys}_{\mathscr{V}}(S,T) := \frac{m_{S,T,\mathscr{V}}}{M_{S,T,\mathscr{V}}}. \tag{7.5}$$

The system nervousness degree expresses the inability of the system to maintain and/or preserve once made decisions during subsequent schedule revisions. Here, $MSN^{sys}_{\mathscr{V}}(S,T)$, $RAN^{sys}_{\mathscr{V}}(S,T)$ and $ATN^{sys}_{\mathscr{V}}(S,T)$ denote the degree of system nervousness with respect to the mode selection, the resource assignment and the operation sequencing (scheduling). They are defined as described generally in (7.5).

7.4 Flexibility and Nervousness of Logistic Operations

Flexibility addresses the ability to integrate additional input into the systems configuration, so that the requirements of the additional input are met (responsiveness). We distinguish between planflexibility and systemflexibility (Subsection 6.1.2).

At first glance, a higher systemflexibility degree suggests a higher system nervousness degree since the higher responsiveness requires intensified schedule revisions. However, there are indicators suggesting a contrary interdependence if more requests are forwarded to some logistic service providers. Then, there is a reduced need for updating the routes of the own vehicles so that at the end a lower number of scheduling decisions requires a revision. To clarify this issue, the same transport system as investigated in the previous chapters is analyzed with respect to nervousness. Now, special attention is paid to the dependencies between the achieved systemflexibility degree and the observed system nervousness degree.

With the vocabulary introduced in this section, the initial research hypothesis can be refined: *If the systemflexibility degree increases then the nervousness degrees $MSN_{\mathscr{V}}^{sys}(S,T)$, $RAN_{\mathscr{V}}^{sys}(S,T)$ and $ATN_{\mathscr{V}}^{sys}(S,T)$ decrease, provided the utilization frequency of subcontracting is adapted to the intermediately observed planflexibility degree instead of deciding strictly on costs.* In the event that this hypothesis is true, interventions of a supply consortium coordinator into the deployment decision derivation of a subordinate service providing agent contribute to a stabilization of the generated logistics processes.

7.5 Numerical Experiments

We use the same simulation layout described in Subsection 5.4.1. In order to enable a thorough analysis of the simulation results, the following performance indicators have been recorded during the experiments: The averagely observed values for $MSN_{\mathscr{V}}(t)$, $RAN_{\mathscr{V}}(t)$ and $ATN_{\mathscr{V}}(t)$ are calculated for each of the five update strategies $\mathscr{V} \in \{HARD, PEN, SDAD, CSAD, SDCS\}$ (instead of describing the preschedule P and the new schedule Q, we link the degrees to the schedule transition time t). After a simulation experiment is completed the achieved degrees for the system mode selection nervousness ($MSN_{\mathscr{V}}^{sys}$), the system resource assignment nervousness ($RAN_{\mathscr{V}}^{sys}$) and the system arrival time nervousness ($ATN_{\mathscr{V}}^{sys}$) are calculated.

Fig. 7.1 shows the degree of mode change nervousness observed during the simulated period. In most of the off-peak re-planning times the two adaptive strategies SDAD and CSAD exhibit a higher mode selection nervousness degree than all other strategies. Immediately after the initiation of the demand peak, HARD as well as SDAD shift a very large portion of requests from an own vehicle to an LSP. After the demand peak is over, the non-adaptive strategies PEN and HARD come along with a noticeable lower number of mode changes than the two adaptive strategies SDAD and CSAD. Remarkably, the application of the hybrid adaptive strategy SDCS leads

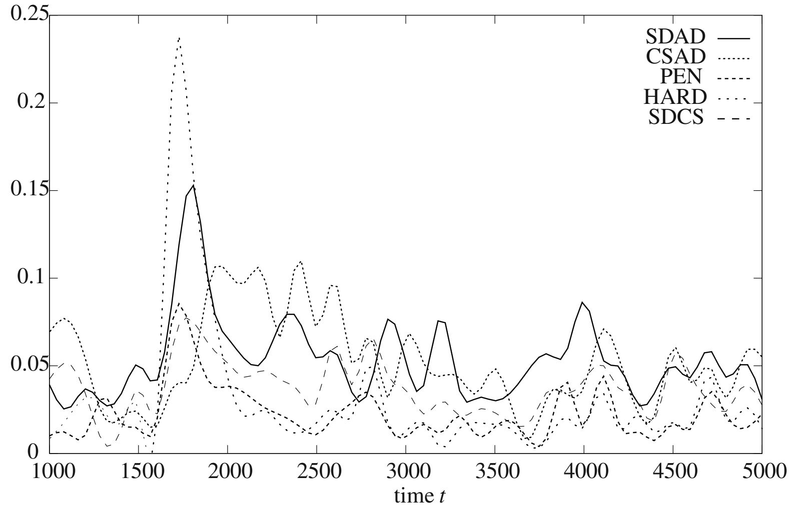

Fig. 7.1 Development of the MSN-degree $MSN_{\mathscr{V}}(t)$

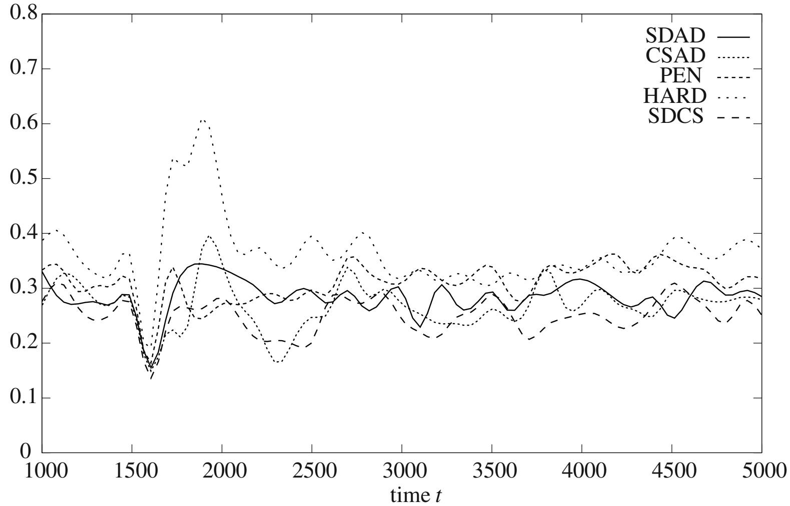

Fig. 7.2 Development of the RAN-degree $RAN_{\mathscr{V}}(t)$

to a mode change frequency that lies between the frequencies of the adaptive and of the non-adaptive strategies.

From Fig. 7.2 we learn that the three adaptive strategies (SDAD, CSAD and SDCS) are able to keep the resource assignment nervousness degree on a lower level than the non-adaptive strategies (HARD and PEN) are able to do. Although the differences are not big, we observe the different default degree in the off-peak

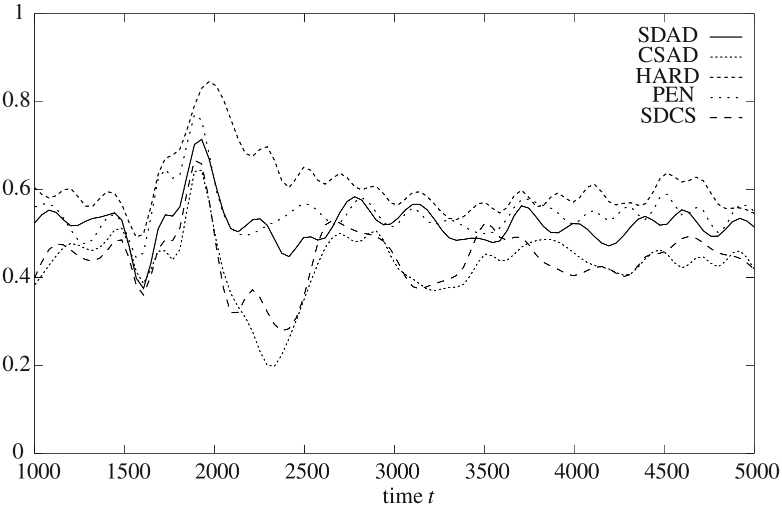

Fig. 7.3 Development of the arrival time nervousness degree $ATN_{\mathscr{V}}(t)$

situations. Immediately after the start of the demand peak, the resource assignment nervousness degree collapses because the number of available requests increases more rapidly than the number of resource assignment changes. However, soon after the acute demand peak is over, the nervousness degree recovers to the pre-peak levels. However, the nervousness degree in the HARD experiments explodes and the nervousness doubles for some re-planning cycles.

The development of the arrival time nervousness is shown in Fig. 7.3. In general, the non-adaptive update strategies HARD and PEN exhibit a slightly higher tendency to re-vise a once fixed arrival time (between 50% and 60%). In the event that one of the adaptive strategies (CSAD, SDAD or SDCS) is used, a default arrival time nervousness degree between 40% and 50% is observed. The appearance of the demand peak leads to an oscillation of $ATN_{\mathscr{V}}$ with a high amplitude. Immediately after the start of the demand peak (time 1600) a remarkable decrease of $ATN_{\mathscr{V}}$ is observed for all update strategies. From time 1600 until 2000 the arrival time nervousness degree increases. In the next re-planning cycles, a decrease down to the pre-peak levels is observed. While the HARD-controlled processes require more than 1000 time units to re-attain the pre-peak arrival time nervousness level, the CSAD and the SDCS-controlled processes demonstrate a collapse of ATN_{CSAD} respectively ATN_{SDCS} down to 0.3 respectively. 0.2. This collapse can be explained by the fact that CSAD as well as SDCS shift a large portion of incoming requests to LSPs (cf. Fig. 5.10) so that after the acute demand peak is over the routes of the vehicles are faintly used and contain enough slack to integrate further requests without re-scheduling operations.

To conclude the online analysis of schedule nervousness, we state that the price of securing the least punctuality rate by applying *HARD* is the high process nervousness after the introduction of the peak. The adaptive strategies exhibit a reasonable tradeoff-performance: a short (CSAD and SDAD) or even no reduction (SDCS) of p_t on a level below p^{target} is ensured simultaneously with the lowest schedule nervousness degrees.

Table 7.1 Observed system nervousness values

$\mathscr{V}$	$MSN_{\mathscr{V}}^{sys}$	$RAN_{\mathscr{V}}^{sys}$	$ATN_{\mathscr{V}}^{sys}$
HARD	4.3%	39.8%	51.75%
PEN	1.4%	35.5%	56.8%
CSAD	5.3%	28.5%	38.2%
SDAD	6.3%	31.4%	48.8%
SDCS	3.8%	27%	40.3%

In order to compare the different degrees of system nervousness, the averagely observed degrees for mode selection, resource assignment and arrival time system nervousness are presented in the columns 2 to 4 in Table 7.1. Since all three adaptive strategies CSAD, SDAD and SDCS enforce and intensify the utilization of the subcontracting mode, an increase of $MSN_{\mathscr{V}}^{sys}$ from 1.4% (PEN) to 3.8% (SDCS), 5.3% (CSAD) respectively 6.25% (SDAD) is observed. SDCS outperforms HARD, because MSN_{SDCS} is lower than MSN_{HARD}. In conclusion, the research hypothesis stated in the introduction of this chapter cannot be verified for this particular internal system nervousness degree.

A different observation is made for the resource assignment system nervousness degree (third column in Table 7.1): RAN_{PEN}^{sys} is 35.5% but this degree decreases down to 31.4% (SDAD), to 28.5% (CSAD) and even down to $RAN_{SDCS}^{sys} = 27\%$ if an adaptive model controller adapts the decision model to the intermediately observed planflexibility degree. Thus, all adaptive strategies clearly outperform the non-adaptive strategies and the HARD-strategy comes along with the highest resource assignment nervousness among all analyzed update concepts. The conclusion of these observations is that the research hypothesis is verified for this specific nervousness degree.

With respect to the degree of the external arrival time nervousness (fourth column in Table 7.1), the observed results enable a clear verification of the research hypothesis. If the knowledge of the intermediate planflexibility degree is not exploited for the variation of the integration preferences (PEN and HARD) then the initially announced request completion time is revised for $ATN_{PEN}^{sys} = 56.8\%$ respectively $ATN_{HARD}^{sys} = 51.8\%$ of all incoming requests. More than half of the announced requests are re-scheduled. If an adaptive update strategy is applied then the arrival time revision degree is reduced below 50% to ATN_{SDAD}^{sys}=48.8% and $ATN_{CSAD}^{sys} = 40.3\%$. In the event that CSAD is used to adjust the next decision model a further reduction below 40% is achieved: ATN_{CSAD}^{sys}=38.2%. It is concluded, that the research hypothesis is verified for the arrival time nervousness.

The decrease of the ATN^{sys}-values is significant, so that this particular aspect has been analyzed in more detail. The second column in Table 7.2 contains the percentage of requests whose final completion time is earlier than their initially announced

Table 7.2 Frequency of requests which have been proponed or deferred compared to their initial completion time

$\mathscr{V}$	earlier	later	unvaried
HARD	21%	31%	48%
PEN	16%	41%	43%
SDAD	17%	32%	51%
CSAD	10%	28%	62%
SDCS	12%	18%	60%

completion time. If HARD is applied then the percentage of this *left shifting* is maximal among all five strategies (21%). PEN and SDAD exhibit a similar behavior (16% and 17%). A significant reduction of left shifting is observed if one the two adaptive strategies CSAD and SDCS is incorporated. From the values presented in the third column it is concluded that the incorporation of adaptive model controllers supports the prevention of deferments. The postponement percentage for a request decreases from 41% (PEN) to 32% (SDAD) and to 31% (HARD). Further reductions are achieved if one of the adaptive strategies CSAD or SDCS is applied. Altogether, the percentage of un-rescheduled requests remains on a higher level if adaptive update strategies are used than if either HARD or PEN is incorporated.

The observed numerical results do not allow a general verification of the research hypothesis. However, it has been observed that the adaptation of the integration preferences to the intermediate planflexibility degree supports the reduction of the external nervousness degree of the arrival times at customer sites. The exploitation of feedback information about the process quality is a reasonable tool to achieve a higher reliability with respect to the announced arrival times so that the customers' trust in a once submitted arrival time increases.

7.6 Conclusions

We have learned that the price for maintaining high process reliability in a volatile environment is an increase of process nervousness from the internal as well as from the external point of view. The application of adaptive model controllers (SDAD, CSAD and SDCS) contributes to an improvement of the performance of the controlled logistic system. In addition to the findings from Chapter 6, we learn from the results reported in this chapter that the temporal and feedback-controlled variation of the decision model (determining the decision preferences of the subordinate fleet-managing service agent) has a positive influence on the stability of once made resource assignment and arrival time decisions. The HARD configuration comes along with the highest nervousness.

From the perspective of the organization of a supply consortium the allowance of principal interventions into the decision making of the subordinate fleet manager is beneficial. As already mentioned above, such interventions contribute to stabilize once made decisions. This benefit justifies the additional expenses to be paid for the demand fulfillment by the supply consortium.

Chapter 8
Impacts on Robustness

Processes in value creating systems are compromised by exogenous disturbances. As a consequence, a recurrent revision of the schedules determining the processes is necessary. Three streams of evaluating such a so-called dynamic decision situation are subject of scientific interest. *Planning Nervousness* or *Planning Stability* addresses the validity of once made process decisions. Special attention is paid to keep the negative impacts of the revision of decisions small (Chapter 7). *Flexibility*-related research is particularly interested in the question if it is possible to generate a feasible (e.g. executable) update of a process (Chapter 6). *Robustness* extends the concept of flexibility and adjoins the consideration of cost variation and benefit variation to the analysis of flexibility issues.

The motivation of the investigation reported in this chapter is twofold. At first and in contrast to other robustness-related investigations we aim at defining robustness of a schedule or system without referring to a special application. Consequently, we cannot use specific domain knowledge for the definition of robustness and its quantification. Second, we want to investigate robustness issues in the area of transportation in supply consortia.

We address robustness issues from a system's perspective (systemrobustness) as proposed by Schillo et al. (2001) and from the perspective of a particular schedule (planrobustness) that describes the system's state transfer Scholl (2001). The contributions of this chapter are the structuring of the multifaceted discussion of ideas of robustness, the generation of a general definition of robustness-related terms, the provision of robustness indicators that help to quantify robustness issues and the application of the proposed concepts in the already reported simulations of a transport system.

Besides describing the robustness of a transportation system we target to extend the ability of a transport process planning unit to compensate for disturbances. To do this, we propose to use adaptive model controllers exploiting SDAD, CSAD and SDCS. It has already been demonstrated that the application of such update approaches supports the increase of the flexibility of a transport system (Chapter 6 as well as Schönberger and Kopfer (2009c) and supports the reduction of planning nervousness (Chapter 7 as well as Schönberger and Kopfer (2008c)).

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*,
Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_8,

In this chapter we want to verify (or disprove) the following two research hypotheses:

1. The robustness of a schedule (determining processes) can be increased if we apply adaptive decision model controllers in the process re-planning.
2. A system in which adaptive model controllers are employed (using SDAD, CSAD or SDCS) is able to compensate for more disturbances than a system in which a static model controller is used for the process control (HARD and/or PEN).

We start with a summary of the discussion of robustness-related issues in the literature (Section 8.1). In Section 8.2, we introduce acceptable schedule updates. Afterwards in Section 8.3, we define the terms *planrobustness* and *systemrobustness* using the definition of an acceptable schedule update. In Section 8.4, we propose a scaled quantification of robustness properties of a schedule or a system. Finally, we configure robustness evaluation schemes for the transportation system introduced in Section 2.3 and present evaluation results from simulation experiments (Section 8.5).

8.1 Robustness in the Literature

Contributions to the discussion of robustness fall into two categories. On the one hand, robustness is referred to as a property of a particular schedule (planrobustness) but on the other hand, robustness is defined as a system's property (systemrobustness). Some authors attempt to analyze robustness issues of a decision method. However, such a discussion falls back to planrobustness and/or systemrobustness. For this reason, we do not discuss robustness aspects of scheduling algorithms here.

The discussion of robustness issues is strongly related to the discussion of online decision situations where the planning data are uncertain at the time when planning decisions are derived. Jensen (2001b) remarks that the relation between robustness and uncertainty is discussed in two directions. On the one hand, uncertainty lies in the problem data but on the other hand, uncertainty is also related to the imprecise implementation of a schedule in a system. Our approach to robustness will cover both directions.

8.1.1 Robustness of Schedules

Robustness is interpreted as a countermeasure to uncertainty that supports keeping changes and revisions of once made decisions small (Graves, 1981; Hnich et al., 2004; Leon et al., 1994). In this context, the prevention of repair costs is referred to as one important idea of generating robust schedules and systems (Ginsberg et al., 1998). Changes in the planning data (or of planning premises) often lead to exceeding the available resources because once coordinated activities run more and more uncoordinated while capacity-restricted resources become bottleneck resources (Keys et al., 2005). Schillo et al. (2001) remark that robustness is related to the measuring of performance of a schedule or a system. In this context, Jensen (2001b) calls a schedule *A* "more robust" than a schedule *B* if the repair costs of *A* are less than the repair costs of *B* after a disturbing event has appeared.

According to Ginsberg et al. (1998), small data variations implying schedule corruption must be capable of being repaired quickly (fast repair, temporal aspect) and with a small number of modifications to the schedule (small repair, effort aspect).

A schedule is considered to be "robust" by Yu and Yang (1998) if it has the best worst-case-performance among all available alternative schedules. The worst-case refers to the external disturbances. Montemanni and Gambardella (2005) specify the uncertainty and investigate shortest path problems where uncertainty is expressed in intervals of possible arc travel times instead of a fixed travel time.

Planrobustness (robustness of a plan or schedule) is defined by Pinedo and Chao (1999) as the schedules' property to remain applicable with only small modifications after the occurrence of a disturbance. More concretely, they call a schedule robust if the schedule's makespan is not affected by a disturbance. Branke (1998) suggests that the quality of a robust schedule does not collapse if the environment in which the schedule is executed changes slightly. Morales (2007) links planrobustness to a min-max-property of a vehicle routing plan. A plan is compared to other plans and called robust if and only if the consideration of additional customer site visits leads to a minimal route length increase among all available plans.

In the context of job shop scheduling, Leon et al. (1994) call a schedule robust if its performance remains on a high level even after a disruption has occurred. Yu and Yang (1998) call a schedule robust if it has the best worst-case performance among the available alternative schedules. Jensen (2001b) calls a schedule robust if conflicts (after the occurrence of a disturbing event) can be solved by a simple right-shifting of so far not started operations. Hart et al. (1998) calculate the Hamming-distance (Whitley, 2000) of the original schedule and its update. If the Hamming distance is low then the original schedule is referred to as robust.

Dynamic maintenance operation scheduling is subject of the investigation carried out by Marmier et al. (2007). Here, a schedule is declared to be "robust" as long as it is insensible to uncertainties and data variations.

Scholl (2001) connects planning goals with disturbances and defines the key property of a robust schedule as its ability to be realizable for all thinkable disturbances without significant variations in the achieved planning goals (compared to the undisturbed schedule).

8.1.2 Robustness of Systems

Scott et al. (2006) establish a connection between the discussion of the term "robustness" and the reliability of a transport network. Tarhini and Fouchal (2006) call a system robust if it is able to remain reliable in improper or stressful environments, e.g. if it is able to operate correctly in the presence of invalid inputs.

Network robustness is defined as the property of a collection of components that maintain their connections among each other even under erroneous environmental conditions and/or input data (Caballero et al., 2008).

Schillo et al. (2001) highlight a connection between robustness and multi-agent systems. They mention that robustness is not a self-evident property of a system but that special efforts are required to avoid performance losses after the occurrence of disturbing events. They call a multi-agent system robust if the group of agents ensures that some basic restrictions (on capacity, performance, etc.) are permanently respected. These so called *safety requirements* (Woolridge et al., 1999) represent the backbone of the considered system and are vital for the system's survival.

Demetrius and Manke (2005) investigate biological systems. They call such a system (e.g. a human or animal) robust if it remains functional in the face of random perturbations. They connect robustness with "the insensitivity of measurable parameters of the system to changes in its internal organization". Furthermore, they distinguish *dynamic robustness* and *topological robustness*. The first property is related to changes of the behavior of network components while the last property is related to the analysis of changes in the network composition.

Kitano and Oda (2005) conjecture a special structure of a robust system. They say that a robust system maintains a small highly conserved core network. This core is linked with the other components in the overall network and these "satellite components" protect the core against external perturbations.

Robustness issues play an important role in the development and configuration of computer systems as well as in software engineering. In the IEEE-definition (Radatz, 1990) systemrobustness is interpreted as "the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions". In this context, DeVale (1998) investigates robustness measures for computer operating systems.

Robustness of a transport network is investigated by Scott et al. (2006). They highlight a specific compensatory property of the US highway system which they describe as "robust". With a relative stable highway capacity the significantly increased transport demand has been managed in the last decades. They link this compensatory effect to the reliability of the investigated highway network.

8.1.3 Achieving, Implementing and Conserving Robustness

A generic approach towards implementing robustness-related properties into a schedule or into a system is to increase its tolerance against faults and errors (Radatz, 1990). The following efforts targeting this idea have been reported in the literature.

Pinedo and Chao (1999) as well as Leon et al. (1994) propose to insert idle times at strategic points in a given schedule. These idle (or slack) times absorb right-shifted operations / tasks in the event that an unforeseen event requires a prolongation or insertion of tasks. They further propagate to schedule least flexible jobs at the beginning of s schedule so that they are processed as early as possible without being endangered by unforeseen events later on.

In order to ensure that a schedule (or system) is robust, it is necessary to invest additional effort (costs, resources) into the schedule (system). The additional expenditures that make a schedule (system) a robust schedule (system) are called the *price of robustness* (Morales, 2007).

Ginsberg et al. (1998) propose so-called supermodels to generate robust schedules. Here, the transformation of a decision supermodel after a data disturbance into a new model guarantees bounded costs of the solution of the transformed model. Thus, a new solution of the now updated problem can be achieved by solving the new model for bounded costs which are known in advance.

Redundancy is labeled as a strategy to obtain disturbance-resistance (Schillo et al., 2001) but it is remarked that redundancy alone cannot be implemented wherever disturbances endanger a schedule or a system. Scalability, flexibility, resistance, drop-out safety and delegation of tasks are proposed as further properties of a (multi-agent-)system that can support the efforts for implementing and increasing the reliability of a system.

Robust Optimization subsumes optimization approaches that consider data uncertainties explicitly in the derivation of schedules. Data perturbations are modeled and added to the used optimization models. Their solving leads to high quality solutions in which some of the aforementioned robustness achieving strategies are implemented. The produced solutions are "robust" in the sense that post-optimization variations of the planning data do not lead to a loss of the optimality of the solution (solution robustness) or even of the feasibility property of the solution (model robustness) as pointed out by Mulvey et al. (1995).

Branke (1998) successfully experiments with the idea to maintain several differently looking schedules of similar performance throughout the observation time. Whenever a disturbance occurs that compromises the execution of the so far followed schedule, one of the aforementioned schedules of the same quality is likely to be able to cope with the new situation without significant revisionary need. Jensen (2001b,a) extends this idea of neighborhood-based robustness, where the neighborhood of a schedule contains all similar-looking schedules.

8.1.4 Measuring and Quantification of Robustness

The update of a schedule is compared with the original schedule (**absolute evaluation variation**) or the update of a schedule is compared with a referential schedule that represents an (artificial) ideal schedule (**referential evaluation variation**) (Yu and Yang, 1998) with optimized performance.

8.1.4.1 Planrobustness

Leon et al. (1994) propose to compare the quantified performance of the original schedule with the performance of the updated schedule generated to integrate the disturbance into the schedule. In this context, they mention that the applied disturbance correction strategy is important for the achievement of robustness. Furthermore, they propose three explicit quantifications of robustness where several disruptive events are simultaneously considered.

Pinedo and Chao compare the length of a delay (the disturbance) with the additional costs caused by the right shifting of subsequent operations in a machine schedule (Pinedo and Chao, 1999). The evaluation of lateness potential of a schedule is proposed by Marmier et al. (2007). Chen and Muraki (1997) suggest comparing the number of actually fulfilled scheduling constraints with the number of constraints to be fulfilled. The quotient between these two numbers is called the degree of satisfaction. A two-dimensional evaluation vector mapping costs and performance of a schedule simultaneously is proposed as robustness quantification tool by Branke (1998).

The worst-case performance of a schedule is used as a measure for robustness by Yu and Yang (1998) in the context of the generation of robust shortest paths in a given network with uncertain travel times. Zieliński (2004) measures the robustness in terms of the *robust deviation of a path p* which is the difference between the costs of a path p and the shortest possible path in the considered network.

Neighborhood-based robustness measures for a schedule C are compiled by Jensen (2001b). He defines the robustness of a schedule as the expected objective function value of all schedules contained in the neighborhood of C. Schillo et al. (2001) enhance this idea and propose to calculate the expected drop in the performance quantification after a disturbing event has become effective.

A quantification of schedule robustness, based on results observed in a Monte-Carlo-Experiment is introduced by Mignon et al. (1995). They fetch the objective function value observed in several Monte-Carlo-Experiment realizations and calculate the standard deviation from these values. Then, the quotient of the standard deviation value and the objective function value from the deterministic counterpart is calculated and interpreted as the degree of robustness of the schedule.

Wang (2004) proposes a measure for schedule robustness which is based on qualitative possibility theory.

8.1.4.2 Systemrobustness

Caballero et al. (2008) use graph connectivity metrics to define the robustness of a network under investigation.

Scott et al. (2006) propose the *Network Robustness Index (NRI)*. The NRI quantifies the compensation properties in a transport network. More precisely, it expresses the performance loss if parts of the network (e.g. a highway segment) become unusable (what-if-investigation).

This survey about the usage of the term robust in planning and system control contexts reveals that there are quite different approaches to the same goal to increase the compensatory abilities of processes. However, most of the proposed definitions and measuring concepts have an explicit background in a specific application domain und exploit domain-specific knowledge. Transportation systems and processes seem to have been of minor research interest so far where robustness is in the focus of research. Furthermore, most concepts to increase the extent of compensatory aspects have their roots in the design of the physical process execution scheme; but the development of special scheduling and control algorithms to increase robustness has received minor attention so far. Adaptation as a design paradigm of such algorithms has not yet been exploited.

8.2 Definition of Robustness

8.2.1 Basic Terms

The **schedule** $P_{A,t}$ determines all necessary activities to transform the considered system $\mathscr{S}$ from a given state ("some requests are not completed") into an intended state ("all requests completed"). It is generated at time t and all data A known at time t are considered during the schedule generation. The schedule determines all processes to be executed in order to achieve the required state transformation of the system $\mathscr{S}$.

A **disturbance** is an event that manipulates the environment in which the processes run. Such an event might be uncontrollable (exogenous disturbance) or caused by the process control unit (endogenous disturbance). In the former case, the data from which the necessary process decisions are derived, vary due to an external uncontrollable event (e.g. additional requests specified by a customer, machine breakdowns or flow congestion). In the latter case, the process control unit activates an event in order to provoke a process revision with the goal to change the process (e.g. reducing working speed of machines).

In the event that a disturbance s occurs at time $t_B \geq t_A$, which corrupts the processes specified by P_{A,t_A} it becomes necessary to replace the so far followed schedule P_{A,t_A} by an updated schedule (Leon et al., 1994). The update is achieved by deriving a new schedule $P_{A\cup\{s\},t_B}$. In general, a schedule P_{B,t_B} is an **update** of another schedule P_{A,t_A} if and only if $B \supset A$ and $t_B \geq t_A$. The inclusion $B \supset A$ represents the growing knowledge acquired by the schedule generation unit while time goes on. Specifically, if we define $B := A \cup \{s\}$ and if we assume $t_B \geq t_A$, then the schedule $P_{A\cup\{s\},t_B}$ is an update of schedule P_{A,t_A}.

Often, several alternative schedules are available to transform the system $\mathscr{S}$ from a given state into another state. In order to distinguish these alternatives, each schedule $P_{A,t}$ is evaluated and mapped into a vector of real numbers $E(P_{A,t}) \in \mathscr{R}^n$ (the set of n-dimensional real-valued vectors). The decision preferences $\mathscr{V}$ describe how the evaluated schedules are ranked. The highest ranked schedule is then selected for realization and is declared to replace the original schedule $P_{A,t}$.

A variation of the applied decision preferences leads to the selection of another schedule update. The set $\mathscr{U}(P_{A,t_A}, s, t)$ is formed by all updates of P_{A,t_A} that incorporate s at time t. It is empty, or it contains one or several schedules. For given rules $\mathscr{V}$ the set of updates includes exactly one element which is denoted by $\mathscr{V}_{s,t}(P_{A,t_A})$ (the $\mathscr{V}$-update $\mathscr{V}_{s,t}(P_{A,t_A})$ of P_{A,t_A}).

8.2.2 *Evaluation Schemes and Acceptable Updates*

All previously mentioned approaches to and auxiliary definitions of robustness have in common that disturbance-triggered changes (of schedules) should be kept as small as possible. In addition, changes in a schedule itself are not considered but changes in the evaluation values of the varied schedule (costs, reliability) have to be as small as possible.

To decide if the evaluation of two schedules P and Q (e.g. a schedule P and its update Q) are "similar enough" we deploy a "distance measure" and a "maximal allowed distance between two schedules". A function $\Delta(E(P), E(Q))$ that maps the two evaluation vectors $E(P)$ and $E(Q)$ into $\mathscr{R}^{n_1}$ serves as the distance measure. The set $\mathscr{E} \subseteq \mathscr{R}^{n_1}$ contains all acceptable $\Delta(E(P), E(Q))$-vectors and plays the role of the maximal allowed distance between P and Q. The **evaluation scheme ES** is defined as the ordered triple $(\Delta, E, \mathscr{E})$. As long as $\Delta(E(P), E(Q)) \in \mathscr{E}$ we call the schedule P to be in the ES-neighborhood of Q and the set $\mathscr{N}(P, \Delta, E, \mathscr{E})$ contains all schedules that fall in the ES-neighborhood of Q. If Q is an update of P, we call Q an **acceptable** (or admissible or intended and wished (Eslami, 1999)) update of P.

In order to demonstrate the concept of acceptable schedule updates we introduce a simple re-planning situation, which is outlined in Fig. 8.1. At time t_i a server is situated at the depot S. According to the schedule P^{orig} (generated at time $t_{i-1} < t_i$) it visits consecutively the customer sites A (at time $t_i + 1$), B ($t_i + 2$), C ($t_i + 3$) and D ($t_i + 4$). Finally, it proceeds to its target T that is reached at time $t_i + 5$.

8.2.3 *Evaluation of Disturbances*

8.2.3.1 Controlling the Variation between the Original and its Update.

For a given schedule P the set $\mathscr{A}^{ES}_{s,t} := \mathscr{A}^{\Delta,E,\varepsilon}_{s,t}(P)$ defined in (8.1) contains all updates of P that fall into the given ES-neighborhood $\mathscr{N}(P, \Delta, E, \mathscr{E})$ of P, e.g. the evaluation vector of the update is "close enough" to the evaluation vector of the original schedule.

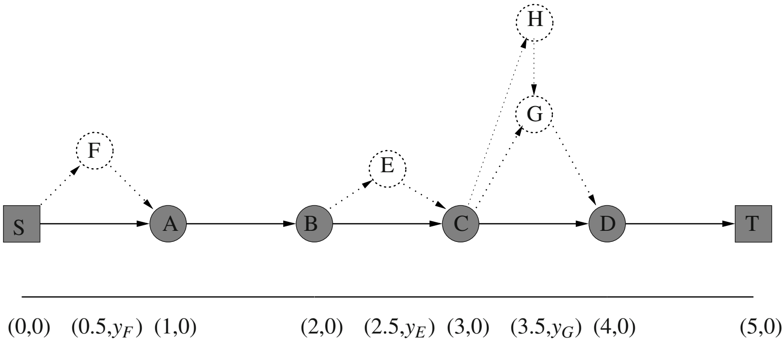

Fig. 8.1 Example of a single server scenario. The depots S and T as well as the customer sites A, B, C and D are known before time t_i.

$$\mathscr{A}_{s,t}^{ES}(P) = \mathscr{A}_{s,t}^{\Delta,E,\mathscr{E}}(P) := \mathscr{U}(P,s,t) \cap \mathscr{N}(P,\Delta,E,\mathscr{E}). \tag{8.1}$$

We demonstrate the function of this acceptance definition using a special setting of the previously described server control scenario. The schedule evaluation vector $E_0(P)$ gives the number of time window violations in a given schedule P. No time window violation occurs as long as P^{orig} is executed, so that $E_0(P^{orig}) = 0$. If P^{update} is an update of P^{orig} then the difference $\Delta_0(P^{update}, P^{orig})$ of the evaluation vectors is defined by $\Delta_0(P^{update}, P^{orig}) := E_0(P^{update}) - E_0(P^{orig})$ and the server control unit does not want to increase the number of time window violations, so that we define $\mathscr{E}_0 := \{r \in \mathscr{R} \mid r \leq 0\}$.

At time t_i the disturbance s_E occurs (cf. Fig. 8.1). It represents the demand for visiting the customer site E at position $(2.5, y_E)$. The arrival at E is expected to fall into the time window $[t_i + 2; t_i + 3]$ which is definitively agreed with the customer. An update of the schedule P^{orig} to the new schedule $P^{update} := \mathscr{V}_{s,t}(P^{orig})$ is done by following the scheduling strategy $\mathscr{V}$:="generate a shortest Hamiltonian path starting at the current server position S and terminating in T (ignoring time windows)".

According to $\mathscr{V}$, the updated schedule $P^{update} := \mathscr{V}_{s,t}(P^{orig})$ instructs the server to follow the path $S \rightarrow A \rightarrow B \rightarrow E \rightarrow C \rightarrow D \rightarrow T$. The acceptability of the update depends on the position of the customer site associated with s_E. If $y_E \leq \sqrt{0.75}$ then the server arrives at $(2.5, y_E)$ within the given time window, so that $E_0(P^{update}) = 0$. This implies $\Delta_0(E_0(P^{update}), E_0(P^{orig})) = 0 \leq 0 \leq \varepsilon$. Thus, P^{update} belongs to the set of acceptable updates: $P^{update} \in \mathscr{A}_{s,t}^{\Delta,E,\varepsilon}(P^{orig})$. However, if $y_E > \sqrt{0.75}$ then the server arrives late at E, so that a time window violation occurs and $E_0(P^{update}) = 1$ implies $\Delta_0(E_0(P^{update}), E_0(P^{orig})) = 1 > \varepsilon$: $P^{update} \notin \mathscr{A}_{s,t}^{\Delta,E,\varepsilon}(P^{orig})$.

8.2.3.2 Referential Variation.

If it is necessary to prevent that a sequence of schedule updates drifts away from the intended development trajectory, then the comparison of the evaluation vector $E(P^{update})$ of the recently generated update P^{update} with a given fixed referential evaluation vector E^{ref} is more suitable (Jensen, 2001b). If the update of a given schedule P does not leave a given ES-neighborhood $\mathscr{N}(P^{ref},\Delta,E,\mathscr{E})$ of a given fixed reference or prototypic schedule P^{ref} then $\mathscr{A}^{ES}_{s,t} := \mathscr{A}^{\Delta,E,\mathscr{E}}_{s,t}(P)$ as defined in (8.2) includes all updates of P schedule that fall into the given neighborhood $\mathscr{N}(P^{ref},\Delta,E,\mathscr{E})$ of the artificial reference schedule with reference vector $E(P^{ref})$. Figuratively speaking, the evaluation vectors of subsequently generated updates "oscillate" around a given fixed evaluation vector.

$$\mathscr{A}^{ES}_{s,t}(P) = \mathscr{A}^{\Delta,E,\mathscr{E}}_{s,t}(P) := \mathscr{U}(P,s,t) \cap \mathscr{N}(P^{ref},\Delta,E,\mathscr{E}) \tag{8.2}$$

We re-visit the example outlined above. Again, the server is completing the schedule P^{orig} in which the sites S,A,B,C,D and T require a visit. All requests contained in P^{orig} are assumed to have been integrated in P^{orig} at time $t_{i-1} = t_i - 1$. A given schedule is evaluated now by the evaluation function $E_1(\cdot)$ that maps the schedule to the average completion time of the served requests. Due to a superior decision, the server control unit has to ensure an average completion time of less than 3 time units. Thus, a referential schedule P^{ref} fulfills the property $E_1(P^{ref}) \leq 3$. In this context, the comparison operator $\Delta_1(E_1(P^{update}),E_1(P^{ref}))$ is defined by $\Delta_1(E_1(P^{update}),E_1(P^{ref})) := E_1(P^{update}) - E_1(P^{ref}) = E_1(P^{update}) - 3$ and the set of acceptable evaluation vector differences is defined as $\mathscr{E}_1 := \{r \in \mathscr{R} \mid r \leq 0\}$.

The schedule P^{orig} is evaluated by $E_1(P^{orig}) := \frac{(1+1)+(1+2)+(1+3)+(1+4)}{4} = \frac{14}{4} = 3.5$ leading to $\Delta_1(E_1(P^{update}),E_1(P^{ref})) := 3.5 - 3 \notin \mathscr{E}_1$. In order to improve the average completion time of scheduled requests the server control unit acquires an additional request at time t_i that requires the visit of the site F situated at position $(0.5, y_F)$. This additional request requires integration into P^{orig} and is therefore interpreted as a disturbance s_F of P^{orig}. Depending on the value of y_F the update P^{update} of P^{orig} generated by applying $\mathscr{V}$ falls into the set of acceptable updates defined in (8.2). In the event that $y_F \leq \sqrt{\frac{1}{1.8} - 0,25}$ it is $E_1(P^{update}) \leq 3$, so that $P^{update} \in \mathscr{A}^{\Delta_1,E_1,\mathscr{E}_1}_{s_F,t_i}(P^{orig})$. In contrast, if $y_F > \sqrt{\frac{1}{1.8} - 0,25}$ then the schedule valuation vector $E_1(P^{update})$ exceeds the threshold value 3, so that $P^{update} \notin \mathscr{A}^{\Delta_1,E_1,\mathscr{E}_1}_{s_F,t_i}(P^{orig})$.

We have demonstrated exemplarily by means of the simple server routing scenario that a performance variation measurement based on absolution variation of schedules and their updates is adequate if the consideration of a disturbance is measured by an indicator that is surely increased in each update or decreased in each update. However, a variation measurement based on a relative comparison of schedules and updates seems to be adequate if a decrease or increase of the observed indicator is possible. This observation can be generalized for multidimensional evaluation vectors: If the indicator function components are unbounded then preferentially an absolute comparison should be applied but if the indicator components are bounded then an adequate reference schedule should be searched for and a relative comparison has to be performed.

8.2.4 Comparison of Input-Output-Variations

So far, we have only considered the variation of the schedule evaluation vector in the discussion of robustness issues (*output variation*). However, the severity of the disturbance has not yet been considered in the definition of an acceptable update (*input variation*). In order to enable the integration of input data variations we generalize the definition of an acceptable schedule update. Therefore, we propose a simultaneous consideration of the input and the output variation in the definition of an acceptable schedule update.

At first, we generalize the term "ES-neighborhood" of a given schedule P and introduce the **extended evaluation scheme EES**. Let P_{B,t_B} and P_{A,t_A} be two schedules. The function $e(A)$ maps the planning data A of a plan P_{A,t_A} to an evaluation vector. In order to decide whether P_{B,t_B} and P_{A,t_A} are "close enough" we map the vector $(E(P_{A,t_A}), e(A), E(P_{B,t_B}), e(B))$ into $\mathscr{R}^n$ using the function Γ. The vector $(\Gamma, E, e, \mathscr{E})$ is called **extended evaluation scheme (EES)**. A schedule P_{B,t_B} belongs to the EES-neighborhood of P_{A,t_A}, denoted by $\mathscr{M}(P_{A,t_A}, \Gamma, E, e, \mathscr{E})$, if and only if the property (8.3) is fulfilled.

$$\Gamma(E(P_{A,t_A}), e(A), E(P_{B,t_B}), e(B)) \in \mathscr{E} \tag{8.3}$$

8.2.4.1 Comparison of Update and Original.

In the context of the comparison of input data variations and schedule modifications during the transformation of a schedule to its update, we define the set $\bar{\bar{\mathscr{A}}}^{EES}_{s,t}(P_{A,t_A}) = \bar{\bar{\mathscr{A}}}^{\Gamma,E,e,\mathscr{E}}_{s,t}(P_{A,t_A})$ of acceptable updates of P_{A,t_A} as shown in (8.4).

$$\bar{\bar{\mathscr{A}}}^{EES}_{s,t}(P_{A,t_A}) := \bar{\bar{\mathscr{A}}}^{\Gamma,E,e,\mathscr{E}}_{s,t}(P_{A,t_A}) := \mathscr{U}(P_{A,t_A}, s, t) \cap \mathscr{M}(P_{A,t_A}, \Gamma, E, e, \mathscr{E}) \tag{8.4}$$

The example introduced in 8.2.2 is re-visited in order to illustrate the functionality of the comparison of input and output variations. The server control unit only accepts an additional request if its consideration leads to an increase of profits that is at least as rapid as the increase of costs. This leads to the definition of the following EES:

- $e_2(A)$: necessary investments (travel expenditures) to fulfill the requests collected in the order backlog A (representing the input data of the schedule P_{A,t_A}) (1 money unit per each traveled distance unit).

- $E_2(P_{A,t_A})$: rewards associated with P_{A,t_A} (We assume that the visit of a customer site is rewarded with 1 money unit).
- $\Gamma_2(E_2(P_{A,t_A}), e_2(A), E_2(P_{B,t_B}), e_2(B)) := \frac{\frac{E_2(P_{B,t_B})}{E_2(P_{A,t_A})}-1}{\frac{e_2(B)}{e_2(A)}-1}$ (we assume $P_{B,t_B} := \mathscr{V}_{s,t}(P_{A,t_A})$).
- $\mathscr{E}_2 := \{r \in \mathscr{R} \mid r \geq 1\}$

For the original schedule P^{orig}, we get $E_2(S \to A \to B \to C \to D \to T) = 4$ (output) and $e_2(\{A,B,C,D\}) = 5$ (input).

At time t_i an additional request is added to the portfolio and must be integrated into P^{orig}. This new request requires the visit of the customer site G situated at $(3.5, y_G)$ (disturbance s_G). We use $\mathscr{V}$ to update P^{orig} to P^{update} and achieve the Hamiltonian path $S \to A \to B \to C \to G \to D \to T$ describing the updated server path. The evaluation of the updated schedule leads to $E_2(P^{update}) = 5$ and $e_2(\{A,B,C,D,G\}) = 4 + 2\sqrt{0.25 + y_G^2}$.

If $y_G \leq \frac{\sqrt{65}}{8}$ then we have $\Gamma_2(4,5,5,4+2\sqrt{0.25+y_G^2}) \in \mathscr{E}_2$ so that the updated schedule fulfills the property $\mathscr{V}(P_{\{A,B,C,D,G\},t_i}) \in \bar{\bar{\mathscr{A}}}_{s_G,t}^{\Gamma_2,E_2,e_2,\mathscr{E}_2}(P_{\{A,B,C,D\},t_{i-1}})$. In all other cases, the necessary detour to visit $(3.5, y_G)$ is too long so that the updated schedule does not fall into the set $\bar{\bar{\mathscr{A}}}_{s_G,t_i}^{\Gamma_2,E_2,e_2,\mathscr{E}_2}(P_{\{A,B,C,D\},t_{i-1}})$.

8.2.4.2 Comparison of Updates with a Fixed Reference Schedule.

We assume now that G has appeared at position $(3.5; 0.2)$, so that $\Gamma_2(4,5,5,4 + 2\sqrt{1.25}) \approx 16.2 \in \mathscr{E}_2$. Thus, the updated visiting order $S \to A \to B \to C \to G \to D \to T$ represents an acceptable update according to the applied EES.

We further assume that additional requests are inserted consecutively into the existing schedule one after another. An additional request H must be integrated into the plan $S \to A \to B \to C \to G \to D \to T$ at time t_i after G has been integrated. This request H requires a visit at position $(3.5; 1)$. Now, the updated schedule $S \to A \to B \to C \to H \to G \to D \to T$ is not an acceptable update of the schedule $S \to A \to B \to C \to G \to D \to T$, because $\Gamma_2(5, 4+2\sqrt{1.25}, 6, 5.5+\sqrt{1.25}) \approx 0.74 \notin \mathscr{E}_2$. However, compared to the original schedule $S \to A \to B \to C \to D \to T$ the last mentioned update is acceptable, because $\Gamma_2(4,5,6,4.8+\sqrt{0.29}+\sqrt{1.25}) \approx 1.7 \in \mathscr{E}_2$. Thus, from the situation after the insertion of the first additional request G, the integration of the second additional request H is not beneficial. However, compared to the initial schedule the integration of both requests G and H is beneficiary. Here, the comparison of the variation between the first and the second update is misleading. The analysis of the evaluation variation from the initial to the second update is more suitable. Similarly to the "compensation of disturbances"-approach for the comparison of an updated schedule with a referential schedule, the comparison of the potential update candidates with a fixed (artificial) referential schedule $P_{A^{ref},t_{A^{ref}}}$ is useful here, too. The set of acceptable updates of a schedule P_{A,t_A} (compared to a reference schedule $P^{ref}_{A^{ref},t_{A^{ref}}}$) is then defined as shown in (8.5).

$$\overset{\equiv}{\mathscr{A}}{}^{EES}_{s,t}(P_{A,t_A}) := \overset{\equiv}{\mathscr{A}}{}^{\Gamma,E,e,\mathscr{E}}_{s,t}(P_{A,t_A}) := \mathscr{U}(P_{A,t_A}, s, t) \cap \mathscr{M}(P^{ref}_{A^{ref},t_{A^{ref}}}, \Gamma, \Delta, E, e, \mathscr{E}) \tag{8.5}$$

8.2.5 *The Role of the Schedule Update Strategy*

Jensen (2001b) states that robustness statements require the specification of the applied update strategy. If we replace the schedule update strategy $\mathscr{V}$ for example by the strategy $\mathscr{V}'$ representing the First-In/First-Out (FIFO) scheduling principle (Thonemann, 2005), then an acceptable schedule update cannot be found; neither for s_E, s_F, s_G nor for s_H. For this reason, the decision about the used update strategy constitutes an important key aspect in the configuration of a responsive as well as efficient value creation system.

8.3 Robustness of Schedules and Systems

We have clearly defined sets of acceptable schedule updates in the previous section. These sets are now used for a clear and straightforward definition of planrobustness and systemrobustness, e.g. the robustness of a given schedule or system alternatively. In order to simplify the presentation, we restrict ourselves to consider only a single type of reference set $\mathscr{A}^{ES}_{s,t}(P)$. If we refer to the aforementioned set defined in (8.1) then the proposed definitions and conclusions are also applicable for the sets defined in (8.2), (8.4), and (8.5). For the same reason, we do not distinguish the applied evaluation system. In the remainder of this section, we only refer to simple evaluation schemes (ES) but the same definitions can be used for extended evaluation schemes.

8.3.1 *Robust Schedules*

Ginsberg et al. (1998) propose to define robustness independently of the used update strategy. However, we have discussed in Subsection 8.2.5 that the acceptance-property depends on the applied update strategy $\mathscr{V}$. Therefore, we propose to explicitly consider the deployed update rule $\mathscr{V}$ for two reasons. At first, the robustness definition of Ginsberg et al. (1998) requires the existence of a special type of decision model. Such models are not so easy to find and only available for very few decision problems. Secondly, the update strategy is an important part in scheduling systems and in order to compare several update strategies, it is necessary, to distinguish update-strategy-specific behaviors.

We call the schedule P **s-$\mathscr{V}$-planrobust at time** t **for the evaluation system ES** if and only if $\mathscr{V}_{s,t}(P) \in \mathscr{A}_{s,t}^{ES}(P)$. Thus, in the event that the disturbance s occurs at time t, a s-$\mathscr{V}$-planrobust schedule is able to be transformed by applying $\mathscr{V}$ into another schedule whose performance complies with the intended performance after the appearance of s at time t and the performance is measured by ES.

In order to enable an analysis of several different disturbances, we extend the planrobustness definition. If D is a set of disturbances compromising the schedule P then this schedule is called **D-$\mathscr{V}$-planrobust at time t for the evaluation system ES** if and only if P is s-$\mathscr{V}$-planrobust at time t for the evaluation scheme ES for all disturbances $s \in D$. The application of the decision preferences $\mathscr{V}$ ensures that a schedule, which is D-$\mathscr{V}$-planrobust at time t cannot be disturbed unexpectedly as long as the disturbances fall into the set D. If D consists of all possible (expected or unexpected) disturbances and if a schedule is D-$\mathscr{V}$-planrobust at time t for the evaluation system ES then this schedule is referred to as **$\mathscr{V}$-planrobust at time t for the evaluation scheme ES**.

If the schedule P is D-$\mathscr{V}$-planrobust for the evaluation system ES not only at a particular time t but during a period (or any other compilation of time points) T then we call P **D-$\mathscr{V}$-T planrobust for the evaluation scheme ES** if and only if P is D-$\mathscr{V}$-planrobust for the evaluation scheme ES at all time points $t \in T$. A *D-$\mathscr{V}$-T* planrobust schedule (for the evaluation scheme ES) is able to absorb all disturbances from D during T with an acceptable performance variation if the decision preferences $\mathscr{V}$ are invoked for updating the schedule. In the event that T comprises the complete relevant observation period, we call a *D-$\mathscr{V}$-T* planrobust schedule (for ES) **D-$\mathscr{V}$ planrobust for the evaluation scheme ES**.

Finally, if D covers all possible disturbances and if T covers the complete relevant observation period, we call a *D-$\mathscr{V}$-T* planrobust schedule (with respect to ES) simply **$\mathscr{V}$-planrobust for the evaluation system ES**.

8.3.2 Robust Systems

In the event that every schedule in the system $\mathscr{S}$ which has been generated by using the decision preferences $\mathscr{V}$ is able to cope with a specific disturbance s at time t (s-$\mathscr{V}$-planrobust at t for ES), we call $\mathscr{S}$ **s-$\mathscr{V}$-systemrobust at time t for ES**. More formally, we define $\mathscr{S}$ to be named *s-$\mathscr{V}$*-systemrobust at time t for ES if and only if each schedule P is *s-$\mathscr{V}$*-planrobust at time t for ES. A sufficient condition for *s-$\mathscr{V}$*-systemrobustness at time t is that there is an update $\mathscr{V}_{s,t}(P) \in \mathscr{A}_{s,t}^{ES}(P)$ for each schedule P at time t.

Let D be again a set of disturbances. Then we call the system $\mathscr{S}$ **D-$\mathscr{V}$-systemrobust at time t for ES** if and only if $\mathscr{S}$ is **s-$\mathscr{V}$-systemrobust at time t for ES** for all disturbances $s \in D$. In the event that D contains all possible disturbances then the system $\mathscr{S}$ is called **$\mathscr{V}$-systemrobust at time t for ES**.

In order to allow systemrobustness statements for several time points, we extend the systemrobustness definition as follows. The set T contains several time points (for example, it covers a certain time period). We name $\mathscr{S}$ **D-$\mathscr{V}$-T systemrobust for ES** if and only if $\mathscr{S}$ is D-$\mathscr{V}$-systemrobust at all time points $t \in T$ for ES.

If D covers all potential disturbances and if T spans over the complete considered time span then a D-$\mathscr{V}$-T systemrobust system $\mathscr{S}$ (for ES) is called V**-systemrobust for ES**.

Here, we have proposed a set of robustness definitions that enable a binary statement whether a schedule or system is robust. DeVale (1998) distinguishes graduations of impacts ranging from "silent errors" having no impacts on the system performance up to "catastrophic failures" that affect the complete schedule or system.

8.3.3 Robustness and Flexibility

Schneeweiß (1992) explicates that robustness is a special form of flexibility. The here proposed definitions for planrobustness as well as for systemrobustness and the definition by Schneeweiß are contradictory. To explain this, we first remember the definitions of planflexibility and systemflexibility introduced in Chapter 6. A schedule P is called e-planflexible at time t with respect to $\mathscr{V}$ if (and only if) there is an acceptable update $\mathscr{V}(P)$ of P in the event that a disturbance e occurs at time t. The term "acceptable update" refers to the feasibility of the proposed update. Each update is evaluated only with respect to the number of constraint violations.

In the context of defining the term "e-planrobustness" we generalize the evaluation of schedule updates. Now, we enable a consideration of all suitable update evaluation schemes, especially those which exploit only the number of violated constraints. Clearly, it is still possible to evaluate a schedule and its update by counting the number of violated constraints. A similar conclusion can be exercised for a robust system. Thus, the robustness-concept generalizes the flexibility-concept by extending the meaning of an "acceptable" update. For these reasons, it makes sense to define robustness as previously explained in this article.

In the application context, flexibility typically addresses the question whether there is a feasible update of a schedule in the event that a disturbance occurs. The costs or any other evaluations associated with the update are of minor interest. However, if the existence of a feasible update is obvious and if updates are easy to generate, then the question "which update should be selected" arises. Here, the evaluation values are consulted so that the robustness issues are now of major interest. Figuratively speaking, flexibility addresses the question whether there is a feasible update but robustness also addresses the costs of the update derivation.

8.4 Quantification of Robustness

The main purpose for the quantification of robustness and the distinction of several "degrees of robustness" is to enable a distinction of different schedules with respect to robustness issues. It is neither realistic to assume that a given schedule is able to be planrobust with respect to all possible disturbances nor is each system systemrobust with respect to all thinkable disturbances. Therefore, it is reasonable to quantify the relative frequency that a schedule (system) is able to compensate for disturbances in a given configuration for a given time or time period (Leon et al., 1994). In this context, Chen and Muraki (1997) define the expected delay of a schedule as a robustness measure and propose to scale this value into the interval $[0,1]$.

In the following, we assume that the applied evaluation scheme ES remains unvaried and that a specific update strategy $\mathscr{V}$ is used. We are interested in answering the question how large is the possibility that a given schedule is planrobust (systemrobust) with respect to a given disturbance randomly drawn from a set of disturbances. Doing so, we want to know the relative frequency that a disturbance can be compensated. This relative frequency is called the "robustness degree" of the given schedule or system. Schneeweiß (1992) proposes this approach to define robustness. Although he uses such a definition to describe the extent of robustness of an update strategy, we can transfer the idea to describe the extent of robustness of schedules and systems. Here, we estimate the probability that a disturbance is able to be compensated. The associated relative frequency is used to define a "degree of robustness" of schedules and systems. Tarhini and Fouchal (2006) as well as Mignon et al. (1995) propose the same idea to calculate probabilities that given constraints are still met after an update.

8.4.1 Planrobustness Quantification

Let $\mathscr{D}_t$ be the set of all disturbances that corrupt a given schedule P at time t. We define $N(t,D_t,P)$ to be the number of possible disturbances of schedule P. The set $\mathscr{D}_t^*$ contains all disturbances $s \in \mathscr{D}_t$ for which P is $s-\mathscr{V}$-planrobust at time t for the applied evaluation scheme ES. We set $n(t,D_t,P,ES,\mathscr{V}) :=\mid \mathscr{D}_t^* \mid$ and define the **$\mathbf{D}_t - \mathscr{V}$-planrobustness-degree** $robdeg^{plan}_{t,\mathscr{D}_t,\mathscr{V},ES}(P)$ of P at time t with respect to ES as shown in (8.6).

$$robdeg^{plan}_{t,\mathscr{D}_t,\mathscr{V},ES}(P) := \frac{n(t,D_t,P,ES,\mathscr{V})}{N(t,D_t,P)}. \tag{8.6}$$

This definition of the schedule's property is derived in a similar way as the planflexibility property proposed in Schönberger and Kopfer (2009c). Again, the calculation of $robdeg^{plan}_{\mathscr{D}_t,\mathscr{V},ES}(P)$ will hardly be possible if the set $\mathscr{D}_t$ is very large. If the cardinality of $\mathscr{D}_t$ exceeds a certain threshold, the quotient must be approximated, e.g. by a Monte-Carlo-based experiment. For a sufficiently large sample of $\tilde{N}(t, D_t, P)$ randomly selected disturbances s it is checked whether the considered schedule P is $s - \mathscr{V}$-planrobust at time t for the applied evaluation scheme ES. We count the number $\tilde{n}(t, D_t, P, ES, \mathscr{V})$ of those disturbances for which the planrobustness is observed and approximate $robdeg^{plan}_{t,\mathscr{D}_t,\mathscr{V},ES}(P)$ by (8.7) as proposed in Chen and Muraki (1997).

$$\frac{\tilde{n}(t, D_t, P, ES, \mathscr{V})}{\tilde{N}(t, D_t, P)} \tag{8.7}$$

If the selected sample is representative for $\mathscr{D}_t$ and if the sample is large enough then (8.7) is a reasonable approximation of (8.6).

8.4.2 Quantification of Systemrobustness

The need to quantify the degree to which a system $\mathscr{S}$ is robust is emphasized by Tarhini and Fouchal (2006). We present a definition of a systemrobustness degree that re-uses the knowledge about planrobustness. Let $\mathscr{T}$ denote a period or a set of periods for which the robustness degree of the system $\mathscr{S}$ should be quantified. The set $\mathscr{D}$ contains all disturbances for which we want to quantify the compensatory properties of $\mathscr{S}$. Similarly to the planrobustness degree definition, we want to calculate the relative frequency (probability) that an event taken from $\mathscr{D}$ occurring within $\mathscr{T}$ and disturbing a schedule in $\mathscr{S}$ can be managed by generating an acceptable update using the update preferences $\mathscr{V}$. We define $M(\mathscr{T}, \mathscr{D}, \mathscr{S})$ to be the number of all possible disturbances of $\mathscr{S}$ belonging to the disturbance set $\mathscr{D}$ and appearing during $\mathscr{T}$. The number of acceptable updates (with respect to ES) generated by $\mathscr{V}$ is stored in $m(\mathscr{T}, \mathscr{D}, \mathscr{S}, ES, \mathscr{V})$. Then the **$\mathbf{D} - \mathscr{V}$-systemrobustness-degree** $robdeg^{system}_{t,\mathscr{D}_t,\mathscr{V},ES}(\mathscr{S})$ of $\mathscr{S}$ during $\mathscr{T}$ with respect to ES is defined by (8.8). Again, we have to approximate $robdeg^{system}_{\mathscr{T},\mathscr{D},\mathscr{V},ES}(\mathscr{S})$ if the number $M(\mathscr{T}, \mathscr{D}, \mathscr{S})$ is very large.

$$robdeg^{system}_{\mathscr{T},\mathscr{D},\mathscr{V},ES}(\mathscr{S}) := \frac{m(\mathscr{T}, \mathscr{D}, \mathscr{S}, ES, \mathscr{V})}{M(\mathscr{T}, \mathscr{D}, \mathscr{S})}. \tag{8.8}$$

8.5 Robustness in a Transportation Scenario

We are now prepared to start the verification efforts for the initially stated research hypothesis. To do this, we define specific evaluation schemes and calculate planrobustness as well as systemrobustness degrees in a transport system. As test scenario, we use the setting introduced in Section 2.3.

8.5.1 Setup of the Simulation Experiments

We simulate the arrival of artificial streams of incoming requests introduced in Section 2.3.4. The same disturbance (arrival of a given number of additional requests) is activated in all experiments so that we refrain from adding disturbance-related information to the statement of planrobustness and systemrobustness values. Since we consider the same observation period $[1000, 4000]$ for all systemrobustness degree calculations, we also desist from adding observation period information to the systemrobustness statements.

For a simulation scenario, we calculate the planrobustness degree $robdeg^{plan}_{t,V,\Sigma}$ and the systemrobustness degree $robdeg^{system}_{V,\Sigma}$. Five different update strategies V are applied. The strategies are based on HARD, PEN (non-adaptive) as well as CSAD, SDAD and SDCS (adaptive). Furthermore, it is necessary to distinguish the robustness degrees with respect to the applied (extended) evaluation scheme $\Sigma \in \{ES_1, ES_2, ES_3, ES_4\}$. We apply two evaluation schemes (ES_1 and ES_2) and two extended evaluation schemes (ES_3 and ES_4).

Absolute Cost Variation ($\Sigma = ES_1$). At time t_{i+1} we compare the cumulated costs $E(P_i)$ and $E(P_{i+1})$ of the schedule P_i and its update P_{i+1}. The relative cost increase $\Delta(E(P_{i+1}), E(P_i)) := \frac{E(P_{i+1})}{E(P_i)} - 1$ should be at most 10% in order to be acceptable ($\mathscr{E} := \{r \in \mathscr{R} \mid r \leq 0.10\}$).

Referential Punctuality Variation ($\Sigma = ES_2$). We compare the punctuality rate $E(P_i) := q_{t_i}$ with the target punctuality rate $E(P^{ref}) := p^{target} = 0.8$ of (an artificial) reference schedule P^{ref} and calculate $\Delta(E(P_i), E(P^{ref})) := E(P_i) - E(P^{ref})$. The acceptable Δ-values are collected in the set $\mathscr{E} := \{r \in \mathscr{R} \mid r \geq -0.05\}$. The remaining parameters are configured as described for ES_1.

Absolute Efficiency Variation ($\Sigma = ES_3$). At time t_i, we calculate the number of requests completed between t_{i-1} and t_i. This number is interpreted as output variation ($E(P_i)$:=number of requests completed not later than t_i) between the consecutive planning steps. Furthermore, we determine the additional costs during the same period. The additionally spent money units are interpreted as input variation ($e(t_i)$ investment up to time t_i.). We define $\Gamma(E(P_i),\ e(t_i),\ E(P_{i+1},\ e(t_{i+1})) := \frac{E(P_{i+1}) - E(P_i)}{e(t_{i+1}) - e(t_i)}$ which gives the number of additionally completed requests for each invested money unit. An update is called acceptable if and only if $\Gamma(E(P_i), e(t_i), E(P_{i+1}, e(t_{i+1})) \geq 0.05$, so that it is desired that 0.05 requests are additionally completed for each invested money unit.

Referential Efficiency Variation ($\Sigma = ES_4$). At time t_i, we calculate the number of requests completed between $t_0 = 1000$ and t_i. Furthermore, we determine the additional costs spent from t_0 until t_i and interpret the resulting amount as input variation ($e(t_i)$ investment up to time t_i.). We define $\Gamma(E(P_0), e(t_0), E(P_i, e(t_i)) := \frac{E(P_i) - E(P_0)}{e(t_i) - e(t_0)}$. An update is called acceptable if and only if $\Gamma(E(P_0), e(t_0), E(P_i, e(t_i)) \geq 0.05$.

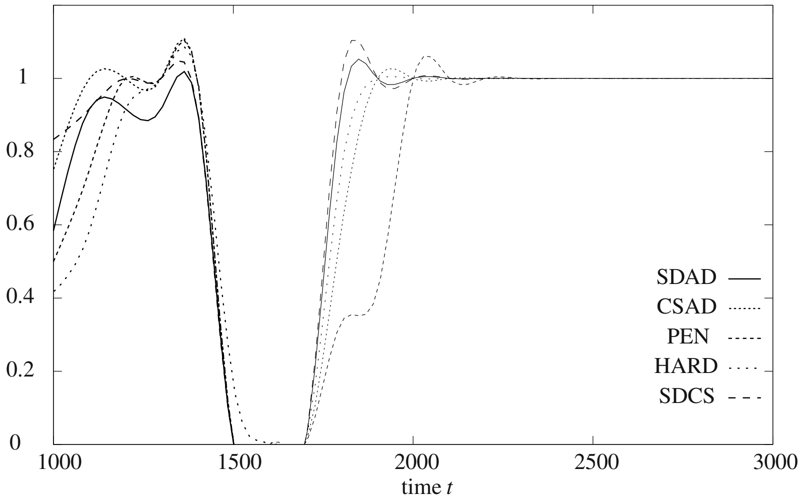

Fig. 8.2 Evolution of the ES_1-planrobustness degree

8.5.2 Presentation and Discussion of Simulation Results

The diagram in Fig. 8.2 represents the planrobustness degrees if the evaluation scheme $\Sigma = ES_1$ is used to quantify the planrobustness. We observe a dramatic decrease of $robdeg^{plan}_{V,ES_1}$ after the initiation of the load peak at time $t = 1500$ for all five update strategies $V \in \{HARD, PEN, SDAD, CSAD, SDCS\}$. The degree falls down to 0 in all cases. If HARD is applied then the pre-peak planrobustness degree is re-achieved at time $t = 1900$. In the event that PEN is incorporated the pre-peak robustness degree is re-achieved at time $t = 2000$. The adaptive update strategy CSAD behaves in a similar way to HARD and re-achieves degree 1 at time $t = 1900$. An incorporation of SDAD or SDCS shifts the time when the pre-peak degree is re-achieved to $t = 1800$. In conclusion, we observe that the two adaptive strategies SDAD and SDCS contribute towards managing the negative impacts of the demand peak more rapidly.

If we use the second evaluation scheme $\Sigma = ES_2$ then we observe the planrobustness degrees shown in Fig. 8.3. In the HARD-experiments no decrease of the ES_2-planrobustness degree is detected, because HARD is able to maintain a punctuality larger than $p^{target} = 0.8$. The application PEN leads to a decrease of $robdeg^{plan}_{PEN,ES_2}$ from 0.6 down to 0 after the initiation of the demand peak. This indicator re-climbs not before time $t = 2500$ and re-attains its pre-peak level at time $t = 3500$ and the pre-peak degree is not reached again before time 3500. If the decision preferences are adapted to the current punctuality rate p_t then the situation is completely different compared to the PEN-experiments. At first, $robdeg^{plan}_{CSAD,ES_2}$ as well as $robdeg^{plan}_{SDAD,ES_2}$ show a pre-peak level of 100%. Furthermore, after the peak's

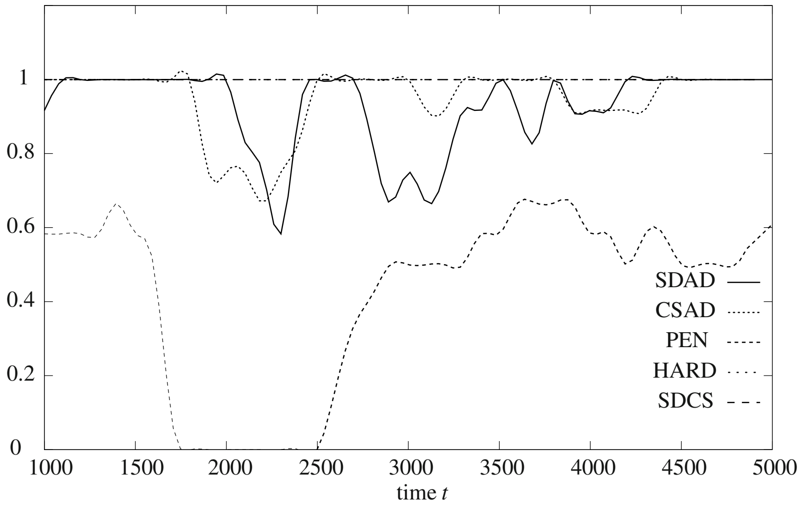

Fig. 8.3 Evolution of the ES_2-planrobustness degree

introduction a decrease of these values down to 0.6 (SDAD) and 0.7 (CSAD) respectively is observed and the pre-peak level is re-attained already at time $t = 2000$. The application of the hybrid adaptive update strategy SDCS leads to a planflexibility degree that is 100% throughout the complete simulation time. If the decision preferences are updated then the pre-peak situation is re-attained significantly earlier than in the experiments without adaptation of the decision preferences. We conclude, that the adaptation of the update strategy leads to higher planrobustness degrees compared to the results observed in the PEN-experiments if ES_1 or ES_2 are applied. Hence, for ES_1 and ES_2 we have proven the first research hypothesis.

In Fig. 8.4, we have compiled the planrobustness degrees averagely observed in the experiments where the extended evaluation scheme ES_3 is used to quantify the planrobustness. We first investigate the behavior of HARD. Before the peak, $robdeg^{plan}_{HARD,ES_3}$ oscillates around 0.8. After the demand peak initialization, the planrobustness degree collapses down to 0 but it re-climbs up to 0.6 in the next replanning iteration. Then, it successively grows until it re-attains 0.8 at time 2500. In the remaining simulation, $robdeg^{plan}_{HARD,ES_3}$ oscillates around 0.8 with high amplitude. We observe a similar behavior for the PEN-controlled process management. The value $robdeg^{plan}_{PEN,ES_3}$ decreases from the pre-peak value 0.6 down to zero after the peak has become effective. At time 2500 the pre-peak value is re-attained again and an oscillation of $robdeg^{plan}_{PEN,ES_3}$ around the pre-peak level 0.6 is observed. An identical behavior is exhibited by the first adaptive model controller based on SDAD. However, the two other adaptive update strategies CSAD as well as SDCS lead to quite different robustness developments. Now, $robdeg^{plan}_{CSAD,ES_3}$ and $robdeg^{plan}_{SDCS,ES_3}$ collapse after the initiation of the demand peak. After the acute de-

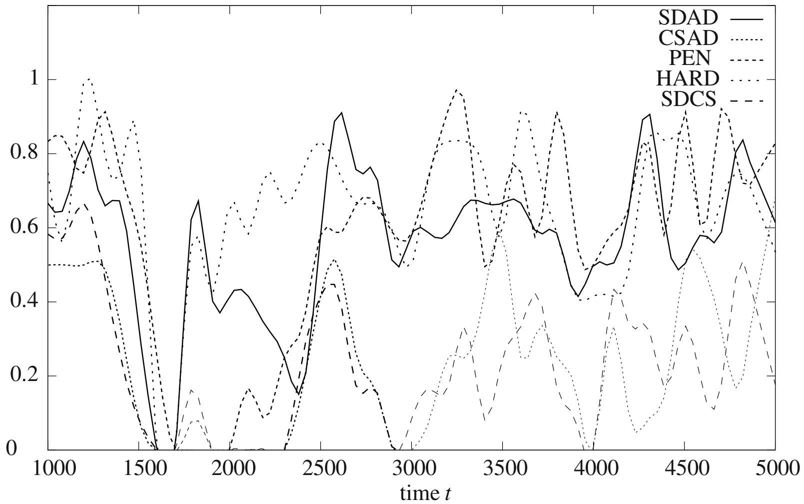

Fig. 8.4 Evolution of the ES_3-planrobustness degree

mand peak is over they hardly reach their pre-peak levels. Instead they reveal an oscillation between zero and their pre-peak levels. Here, a sustainable and persistent change in the robustness performance is observed. If we apply CSAD or SDCS then the number of requests which can be completed using one invested money unit is lower after the demand peak than before the demand peak. This observation is in line with the results from the online (Fig. 5.7) and offline (Tab. 5.10) cost analysis executed in Section 5.4 as well as in Section 5.5.

The results observed in the ES_4-experiments deliver additional experiences and induce the conclusion that the adaptation of a decision model does not lead to a higher planrobustness degree in general. We observe that $robdeg^{plan}_{HARD,ES_4}$ and $robdeg^{plan}_{PEN,ES_4}$ collapse after the initialization of the demand peak. The planrobustness degree falls from around 0.8 (HARD) and 0.75 (PEN) down to 0. After reaching 0, $robdeg^{plan}_{HARD,ES_4}$ starts climbing up immediately but $robdeg^{plan}_{PEN,ES_4}$ remains zero until time 2700 and starts its increase. However, the pre-peak degrees are not reached again during the simulation. SDAD is the only adaptive update strategy that exhibits a similar behavior than HARD and PEN do. For the two remaining adaptive strategies CSAD and SDCS, the demand peak has disastrous impacts. Both planrobustness degrees $robdeg^{plan}_{HARD,ES_4}$ and $robdeg^{plan}_{HARD,ES_4}$ sink down to 0 and do not re-increase at all during the rest of the experiments. Hence, we cannot verify the first research hypothesis for the extended evaluation scheme ES_3 nor for ES_4.

In Table 8.1, the observed systemrobustness degrees $robdeg^{system}_{V,\Sigma}$ are compiled. In the ES_1-experiment, the incorporation of the strategies based SDAD or CSAD leads to a small increase of the systemrobustness degree from $robdeg^{system}_{PEN,ES_1} = 0.86$ to $robdeg^{system}_{SDAD,ES_1} = 0.89$, $robdeg^{system}_{CSAD,ES_1} = 0.90$ and $robdeg^{system}_{SDCS,ES_1} = 0.90$ re-

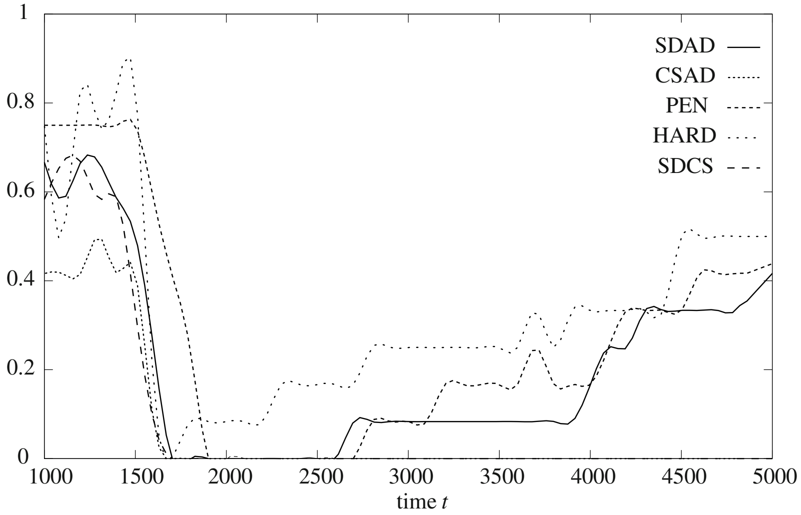

Fig. 8.5 Evolution of the ES_4-planrobustness degree

Table 8.1 Observed systemrobustness degrees $robdeg^{system}_{V,\Sigma}$.

	Σ			
V	ES_1	ES_2	ES_3	ES_4
HARD	0.89	1.00	0.65	0.33
PEN	0.86	0.42	0.58	0.27
SDAD	0.89	0.92	0.56	0.20
CSAD	0.90	0.95	0.23	0.07
SDCS	0.90	1.00	0.23	0.09

spectively. Here, the application of the adaptive model controllers enables the integration of coordinator and service agent decision processes and simultaneously provides the same process quality than the non-applicable strategy HARD. A similar situation is observed for ES_2. Here, the application of HARD provides the best result with a system robustness degree $robdeg^{system}_{HARD,ES_2} = 1.00$. If PEN is used then the systemrobustness degree is reduced to $robdeg^{system}_{PEN,ES_2} = 0.42$. Again, the application of the strategies based on decision model adaptation leads to a quite good approximation of the results achieved by applying HARD: it is $robdeg^{system}_{SDAD,ES_2} = 0.92$, $robdeg^{system}_{CSAD,ES_2} = 0.95$ and $robdeg^{system}_{SDCS,ES_2} = 1.00$. These results support the conjecture that our second research hypothesis is true.

A contrary development of the systemrobustness degrees is observed in the ES_3- and in the ES_4-experiments. In the third experiment-class (ES_3), we recognize that HARD and PEN lead to similar results: $robdeg^{system}_{HARD,ES_3} = 0.65$ and $robdeg^{system}_{PEN,ES_3} = 0.58$. However, the application of an adaptive update strategy does not lead to results as good as the results observed for HARD, so that the strategies based on decision model adaptation are not adequate enough to substitute HARD. In the forth experiment-class (ES_4), all strategies based on decision model adaptation exhibit a poor performance and do not reach the performance of HARD or PEN. These results contradict the second research hypothesis. Thus, the second research hypothesis cannot be verified here.

8.6 Conclusions

In this chapter, we have proposed a domain-independent definition of robustness and we have developed robustness measures. Within the investigated artificial transport planning scenario we have configured several robustness evaluation systems and proven the general applicability of our robustness concept. In contrast to the discussion of planning stability/nervousness and planning flexibility we have not succeeded in demonstrating in general that adaptive process planning systems have positive impacts to the robustness of processes and plans.

The proposed robustness degrees enable a comparison of the efficiency of the investigated process update strategies. From this comparison, we learn that the efficiency of the approach based on HARD outperforms the efficiency of the approaches based on decision model adaptation. This observation leads to the conclusion that it is not possible to substitute the approach based on HARD with an approach based on decision model adaptation (SDAD, CSAD or SDCS) in order to integrate the decision making of a superior supply consortium coordinator and of a subordinate service agent in the consortium. Thus, from the viewpoint of the improvement of principal agent relationships the application of process update strategies exploiting decision model adaptation is not beneficial. This conclusion is not in line with the findings from the flexibility investigations but confirms the conclusions drawn from the nervousness experiments.

Chapter 9
Summary and Conclusions

Having now completed the report about the research conducted on dynamic decision problems in volatile environments we summarize the main findings and draw major conclusions from the achieved results. In Section 9.1, we recall the challenges we have investigated during this study. The proposed methodological innovations to overcome these challenges are summarized in Section 9.2. We terminate this study with an aggregation of the announced benefits to be achieved by incorporating the new decision technology. Finally, a summary of improvement potentials associated with model-based process planning in principal-agent-relationships is given (Section 9.3).

9.1 Principals, Agents and Dynamic Decision Problems

We have established a link between dynamic decision problems for process planning in supply consortium applications and principal-agent relationships. While dynamic decision problems are typically in the focus of operations management and address operational challenges in a supply consortium, principal-agent relationships are the subject of challenges addressing the configuration or even the design of supply consortia. The establishment of the link thus enables a common discussion of operational (short-term) and longer term issues of dynamic decision problems in supply projects (Chapter 1).

In the focus of our interest was a decision scenario from transportation logistics that represents a typical planning situation in a supply consortium where a superior principal and a subordinate agent are involved. A coordinator that represents the entirety of the consortium and a transport service providing coalition member have to interact in order to determine transport processes that are vital for the efficient flow of materials through the different value creation stages. Both interacting consortium members have different planning goals, which are often even contradictory. Obviously, conflicting planning preferences exist and compromise the process planning. Typically, a coordinator wants to ensure the sustainability and survivability of the

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*, Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4_9,

entire consortium but the service providing partner has the primary goal to maximize its own benefit instead of acting in the sense of common welfare. We have discussed and modeled this conflicting situation in detail in Chapter 2.

State-of-the-art techniques are not able to deal with the aforementioned challenges in dynamic decision making for the determination of processes in supply consortia. We have learned from the analysis of the simulation results reported in Chapter 3 that two conceptual deficiencies prevent the successful application of decision support systems. Both deficiencies exhibit structural shortcomings of hierarchical planning that is the fundamental paradigm for deployment in supply consortia. The first observation is that there is a top-down provision of planning assumptions from a superior stage in the assumed hierarchy to a level stage. However, the superior decision making unit is unable to intervene directly in the deployment decisions of the subordinate member of the supply consortium. It is impossible that the superior decision unit overrules the subordinate process planning decision unit even if the superior unit (e.g., a consortium coordinator) has additional information, that predicts an explicit ad-hoc intervention. The second observation is that feedback about impacts of lower-level process decisions (derived by a subordinate service agent) is hardly exploited by a coordinator during the derivation of decisions belonging to the next decision making cycle.

In conclusion, we have learned that a superior coordinator and a subordinate service providing coalition partner interact according to a quite static top-down-strategy that ignores recent feedback information to a large extent. As a consequence, the process quality collapses if an ad-hoc change of the decision situation (e.g. a relative resource shortage in a demand peak situation) occurs.

9.2 Innovative Methods for Decision Derivation and Evaluation

To remedy the aforementioned deficiency of decision support tools discussed in the literature, we have proposed to extend the scope of process controlling. The idea is to enable a superior process controller not only to provide additional problem data to the subordinate service providers but also to adjust the decision preferences to the current decision situation and to dictate the application of the varied preferences to the subordinate service provider. Doing so, the coordinating unit is able to avoid deadlock situations caused by decision guidelines that are incompatible with the current decision situation.

The adaptation of the decision rules to the recent decision situation is achieved by carrying out image modification (Chapter 4), which manipulates the maintained formalized representation (decision model) of the recent decision situation. Actually, the logistics process emits a feedback signal that contains information about the current process quality. The coordinator fetches this signal and derives its intervention actions. If the observed quality is in danger of deviating too much from a given reference signal then image modification updates the parameters of the maintained decision model in order to bias the derivation of the next process decisions.

In order to exploit image modification capabilities in online decision making, we have proposed to add a so-called decision model controller to the process planning system. This controller rules about the necessary decision model adjustments. If the decision model controller is *adaptive* then it decides about those adjustments by evaluating the recent process feedback signal. A re-formulation of the maintained objective function leads to biased valuations of decision alternatives so that the preferences among the existing alternatives are shifted. The update of constraints enables the relaxation and sharpening of process requirements, which especially prevent infeasibilities during the solving of the decision model.

Image modification integrates two planning cycles: the process management and the decision model management. The process management cycles represents the planning task of a subordinate service providing agent in a supply consortium. In contrast, the decision model management (decision preference adjustment) is a task for which the consortium coordinator is responsible. For this reason, image modification is a decision support technology that integrates different aspects of process management executed by the two aforementioned members of a supply consortium. The specific requirements of both kinds of members are respected simultaneously: A coordinator is granted the ability to intervene in the process decisions if it is necessary from the consortium's point of view but the service providing agent maintains its autonomy in the process-related decision making to the largest justifiable extent. We have been able to verify these theoretical conclusions for the investigated scenario from transportation logistics in a simulated supply consortium setting (Chapter 5).

Five different process update strategies have been tested for their usage in the adaptive process controller of the rolling horizon planning framework. Three of them (SDAD, CSAD and SDCS) make use of the image modification capabilities we have added to the process control. Their performance has been compared to the performance of two "traditional" state-of-the-art techniques (HARD and PEN), which do not use the innovative process control technology. The assessment of the five strategies is based on the recording and comparison of performance indicators during and after computational simulation experiments. We use two classes of performance indicators. The first class comprises quantifications that are typically used to evaluate the performance of decision tools applied to static decision problems. We have measured and compared costs, punctuality, resource assignment shares and similar indicators. In order to prepare and execute a more detailed and deeper analysis of the update strategies especially designed for the application in dynamic environments we have proposed a second class of performance indicators. These performance quantification approaches explicitly address special properties of dynamic decision situations. We have developed and applied a domain-independent evaluation concept for flexibility (Chapter 6) which enables a quantification of the portion of additional requests that can be served by a transport system to the customers' satisfaction. Furthermore, we have turned our attention to the validity of once made decisions in re-planning situations and we have proposed nervousness (instability) measures (Chapter 7). Finally, we have merged cost and process reliability performance indicators and we have quantified the resulting robustness of the differently configured simulation scenarios (Chapter 8).

9.3 Principal-Agent Relationships: Improved Process Quality

The degree of integration and interaction of supply consortium coordinators and service providing partners in the process determination is crucial for the quality of the processes offered by a supply consortium. On the one hand, customers of the supply consortium claim a quite high responsiveness and reliability of services. Here, the satisfaction of customers is a prerequisite for the economic success of the consortium (external need for high quality processes). On the other hand, the satisfaction of the members of the consortium determines the success of the whole coalition. In order to ensure and maintain the contentment of the partners, reliability and efficiency has to be provided for each partner involved. Therefore, the sustainability of the supply consortium requires high quality processes as well. This is called the internal need for high quality processes.

We have proven in the executed experiments that the proposed extension of adaptive process controllers by image modification contributes to the improvement of the quality of generated processes in several ways. The starting point of our computational simulation experiments was the presentation of the two integration strategies for the decision making in principal-agent relationships found in the literature. It has been shown that the HARD strategy satisfies the external need for high quality processes in volatile environments. However, the distribution of the economic risk is quite unfair. It discriminates subordinate service providing coalition members against coordinating members. In order to protect themselves, the unfairly treated members have to leave the supply consortium. Thus, the internal need for high quality processes is not satisfied. For the second found strategy (PEN), an inverse observation is made. If PEN is used to integrate the decision making of a coordinator and of a subordinate service providing agent then the external requirements for high quality processes are not met. The reliability of the generated processes is too low, especially in demand peak situations. However, subordinate service providing partners accept the PEN strategy because it offers a quite fair share of economic risk among the coordinating unit (and thus among the consortium) and the subordinate service provider.

In order to merge the advantages of both strategies HARD and PEN while reducing their deficiencies we have developed three so-called adaptive process update strategies (SDAD, CSAD and SDCS) that exploit the features of the image modification technology. Adequate adaptive decision model controllers have been developed for all three strategies. It has been demonstrated that the application of an adaptive decision model controller is able to improve the responsiveness of a system. The incorporation of adaptive model controllers into the online process planning leads to a significant increase of the decision stability. In conclusion, the application of update strategies making use of image modification technology contributes to an increase of the external process quality. A higher number of additional transport requests can be served to the customer's satisfaction and the validity of once announced arrival times is significantly lifted. The application of the robustness measures developed in this study enables a comparison of the process quality increase with the necessary additional expenditures. We have found out that there is at least one adaptive strategy that leads to a systemrobustness degree comparable with the systemrobustness degree observed for the PEN strategy. Thus, it is possible to replace the low-performing strategy PEN by a suitable adaptive process update strategy that guarantees a higher process quality at comparable costs.

We have demonstrated that the performance of all three proposed adaptive strategies is (sometimes only slightly) lower than the performance of HARD but (often significantly) better than the performance of PEN. However, the investment of additional expenditures is necessary. Each supply consortium has to decide whether it wants to spend an additional budget to improve their process performance. This study might contribute to the corresponding internal discussion among the partners forming a supply consortium. It provides starting points for an improvement of decision support for value creation within supply consortia.

References

Albers, S., Charikar, M., Mitzenmacher, M., 2001. Delayed Information and Action in On-Line Algorithms. Information and Computation 170 (2), 135–152.

Andersen, E., Andersen, K., 1995. Presolving in linear programming. Mathematical Programming 71, 221–245.

Anderson, S., Jorna, R., Verweij, C., 1996. Satellite communication in road freight operations: the metafora experience. International Journal of Physical Distribution and Logistic Management 26 (1), 49–61.

Archetti, C., Savelsbergh, M., Speranza, M., 2008. To split or not to split: That is the question. Transportation research Part E: Logistics and Transportation review 44 (1), 114–123.

Archetti, C., Speranza, M., 2007. An Overview on the Split Delivery Vehicle Routing Problem. In: Waldmann, K.-H., Stocker, U. (Eds.), Proceedings of Operations Research 2006. Springer, pp. 123–127.

Arnold, M., Fink, S., Grove, D., Hind, M., Sweeney, P., 2005. A Survey of Adaptive optimization in Virtual Machines. Proceedings of the IEEE 93 (2), 449–466.

Ausiello, G., Feuerstein, E., Leonardi, S., Stougie, L., Talamo, M., 1994. Serving Requests with On-line Routing. In: Schmidt, E., Skyum, S. (Eds.), Proceedings of 4th Scandinavian Workshop on Algorithm Theory. Springer, pp. 37–48.

Bäck, T., Fogel, D., Michalewicz, T. (Eds.), 2000a. Evolutionary Computation 1 - Basic Algorithms and Operators. IoP.

Bäck, T., Fogel, D., Michalewicz, T. (Eds.), 2000b. Evolutionary Computation 2 - Advanced Algorithms and Operators. IoP.

Ballou, R., Gilbert, S., Mukherjee, A., 2000. New Managerial Challenges from Supply Chain Management. Industrial Marketing Management 29, 7–18.

Barad, M., Sapir, D. E., 2003. Flexibility in logistic systems – modeling and performance evaluation. International Journal of Production Economics (85), 155–170.

Basagni, S., Conti, M., Giordano, S., Stojmenoviċ, I. (Eds.), 2004. Mobile Ad Hoc Networking. Wiley-IEEE.

Beach, R., Muhlemann, A., Price, D., Paterson, A., Sharp, J., 2000. A review of manufacturing flexibility. European Journal of Operational Research 122, 41–57.

J. Schönberger, *Model-Based Control of Logistics Processes in Volatile Environments*, Operations Research Computer Science Interfaces Series 50,
DOI 10.1007/978-1-4419-9682-4,

Beullens, P., Oudheusden, D. V., Van Wassenhove, L., 2004. Collection and Vehicle Routing Issues in Reverse Logistics. In: Reverse logistics: quantitative models for closed-loop supply chains. Springer, pp. 95–134.

Bierwirth, C., 2000. Adaptive Search and the Management of Logistics System. Kluwer.

Black, F., Scholes, M., 1973. The Pricing of Options and Corporate Liabilities. The Journal of Political Economy 81 (3), 637–654.

Bloos, M., Schönberger, J., Kopfer, H., 2009. Supporting cooperative demand fulfillment in supply networks using autonomous control and multi-agent-systems. In: Proceedings of INFORMATIK 2009. pp. 3590–3604.

Branke, J., 1998. Creating Robust Solutions by Means of Evolutionary Algorithms. In: Proceedings of the 5^{th} international Conference on Parallel Problem Solving from Nature. Springer, pp. 119–128.

Branke, J., Mattfeld, D., 2005. Anticipation and flexibility in dynamic scheduling. International Journal of Production Research 43 (15), 3103–3129.

Brehmer, B., 1992. Dynamic decision making: Human control of complex systems. Acta Psychologica 81, 211–241.

Brown, G., Ellis, C., Graves, G., Ronen, D., 1987. Real-Time, Wide Area Dispatch of Mobil Tank Trucks. Interfaces 17, 107–120.

Burkhard, R., Labbè, M., Ramos, M., Sicila, J., 2000. Recent advances in theory and practice of combinatorial optimization (ecco x). European Journal of Operational Research 123 (2), 227–228.

Busemeyer, J., 2001. Dynamic Decision Making. In: International Encyclopedia of the Social and Behavioral Sciences: Methodology. Mathematic and Computer science. Elsevier, pp. 3903–3908.

Caballero, J., Kampouris, T., Song, D., Wang, J., 2008. Would Diversity Really Increase the Robustness of the Routing Infrastructure against Software Defects? In: Proc. of the 15^{th} Annual Network and Distributed System Security Symposium.

Calza, F., Passaro, R., 1997. Edi network and logistics management at unilever-sagit. Supply Chain Management: An International Journal 2 (4), 158–170.

Campbell, A., Clarke, L., Kleywegt, A., Savelsbergh, M., 1998. The Inventory Routing Problem. In: Crainic, T., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer, pp. 95–113.

Ceselli, A., Gatto, M., Lübbecke, M., Nunkesser, M., Schilling, H., 2008. Optimizing the Cargo Express Service of Swiss Federal Railways. Transportation Science 42 (4), 450–465.

Chen, S., Golden, B., Wasil, E., 2007. The Split Delivery Vehicle Routing Problem: Applications, Algorithms, Test Problems, and Computational Results. Networks 49 (4), 318–329.

Chen, W., Muraki, M., 1997. A Fuzzy Evaluation of Schedule Robustness Under Processing time variations in batch plants. Journal of Chemical Engineering of Japan 30 (2), 260–267.

Chuah, K., Yingling, J., 2005. Routing for a Just-in-Time Supply Pickup and Delivery System. Transportation Science 39 (3), 328–339.

Claramunt, C., Jiang, B., Bargiela, A., 2000. A new framework for the integration, analysis and visualisation of urban traffic data within geographic information systems. Transportation Research Part C: Emerging Technologies 8 (1-6), 167–184.
Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F., 2002. A Guide to Vehicle Routing Heuristics. The Journal of the Operational Research Society 53 (5), 512–522.
Corsten, H., 2007. Produktionswirtschaft, 11th Edition. Oldenbourg.
Corsten, H., Gössinger, R., 2006. Output flexibility of service enterprises – analysis based on production theory. International Journal of Production Economics 104, 296–307.
Crainic, T., Laporte, G., 1997. Planning models for freight transportation. European Journal of Operational Research 122, 409–438.
Croes, G., 1958. A method for solving traveling salesman problems. Operations Research 6, 791–812.
De Groote, X., 1994. The Flexibility of Production Processes: A General Framework. Management Science 40 (7), 1587–1617.
de Kok, T., Inderfurth, K., 1997. Nervousness in inventory management: Comparison of basic control rules. European Journal of Operational Research 103, 55–82.
de Toni, A., Tonchia, S., 1998. Manufacturing flexibility: a literature review. International Journal of Production Research 36 (6), 933–945.
Demetrius, L., Manke, T., 2005. Robustness and network evolution - an entropic principle. Physica A 346, 682–696.
Derigs, U., Friederichs, S., Schäfer, S., 2009. A New Approach for Air Cargo Network Planning. Transportation Science 43 (3), 370–380.
DeVale, J., 1998. Measuring Operating System Robustness. Tech. rep., Department of Electrical and Computer Engineering & Institute for Complex Engineering Systems, Carnegie Mellon University.
Dorigo, M., Stützle, T., 2004. Ant colony optimization. MIT Press.
Ehmke, J., Meisel, S., Mattfeld, D., 2008. Floating Car Data Based Analysis of Urban Travel Times for the Provision of Traffic Quality. In: Proceedings des IWT-DCS 08.
Elschen, R., 1991. Gegenstand und Anwendungsmöglichkeiten der Agency-Theorie. zfbf - Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung 43 (11), 1002–1012.
Erera, A., Karacık, B., Savelsbergh, M., 2008. A dynamic driver management scheme for less-than-truckload carriers. Computers & Operations Research 35, 3397–3411.
Erkens, E., Kopfer, H., Siek, K., Hülder, M., Schöpe, L., Mann, S., 2005. Analyse von Rahmenbedingungen. In: Herrmann, T., Schöpe, L., Erkens, E., Hülder, M. (Eds.), Mobile Speditionslogistikuntersttzung. Shaker-Verlag Aachen, pp. 2–1 – 2–77.
Eshelman, L., 2000. Genetic algorithms. In: Bäck et al. (2000a), pp. 64–80.
Eslami, M., 1999. Robustness measures in nonlinear discrete-time systems with delays and large-parameter variations. Automatica 35, 159–167.
EU, 2009. EU energy and transport in figures – statisical pocketbook 2009.

Feillet, D., Dejax, P., Gendreau, M., 2005. Traveling salesman problems with profits. Transportation Science 39, 188–205.

Feitelson, E., Salomon, I., 2000. The implications of differential network flexibility for spatial structures. Transportation Research Part A 34, 449–479.

Ferman, M., Blumenfeld, D., Dai, X., 2005. An Analytical Evaluation of a Real-Time Traffic Information System Using Probe Vehicles. Journal of Intelligent Transportation Systems 9 (1), 23–34.

Fleischmann, B., 2002. Distribution and Transport Planning. In: Stadtler and Kilger (2002), pp. 195–210.

Fleischmann, B., Gietz, M., Gnutzmann, S., 2004a. Time-Varying Travel Times in Vehicle Routing. Transportation Science 38 (2), 160–173.

Fleischmann, B., Gnutzmann, S., Sandvoß, E., 2004b. Dynamic Vehicle Routing Based on Online Traffic Information. Transportation Science 38 (4), 420–433.

Fleischmann, B., Meyr, H., Wagner, M., 2002. Advanced planning. In: Stadtler and Kilger (2002), pp. 71–104.

Garfinkel, R., 1985. Motivation and modeling. In: Lenstra, J., Kan, A. R., Shmoys, D. (Eds.), The Traveling Salesman Problem. Wiley, pp. 17–36.

Geiger, M., Wenger, W., 2007. On the Interactive Resolution of Multi-objective Vehicle Routing Problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), Evolutionary Multi-Criterion Optimization. Springer, pp. 687–699.

Gendreau, M., Laporte, G., Semet, F., 2004. Heuristics and lower bounds for the bin packing problem with conflicts. Computers & Operations Research 31 (3), 347–358.

Giannopoulos, G., 2004. The application of information and communication technologies in transport. European Journal of Operational Research 152 (2), 302–320.

Ginsberg, M., Parkes, A., Roy, A., 1998. Supermodels and Robustness. In: Proceedings of the fifteenth national/tenth conference on Artificial intelligence / Innovative applications of artificial intelligence. pp. 334–339.

Glover, F., Klingman, D., Phillips, N., 1992. Network Models in Optimization and Their Application in Practice. Wiley.

Glover, F., Laguna, M., 1997. Tabu Search, 5th Edition. Springer.

Goel, A., 2009. Vehicle Scheduling and Routing with Drivers' Working Hours. Transportation Science 43 (1), 17–26.

Goel, A., Gruhn, V., 2008. A General Vehicle Routing Problem. European Journal of Operational Research 191, 650–660.

Golden, B., Raghavan, S., Wasil, E. (Eds.), 2008. The Vehicle Routing Problem. Springer.

Graves, S., 1981. A Review of Production Scheduling. Operations Research 29, 646–675.

Grefenstette, J., 2000. Rank-based selection. In: Bäck et al. (2000a), pp. 187–194.

Grosan, C., Abraham, A., Ishibuchi, H. (Eds.), 2007. Hybrid Evolutionary Algorithms. Springer.

Grötschel, M., Krumke, S., Torres, L., 2002. Online-dispatching of automobile service units. In: Leopold-Wildburger, U., Rendl, F., Wäscher, G. (Eds.), Operations Research Proceedings. Springer, pp. 168–173.

Grubbström, R. W., Olhager, J., 1997. Productivity and flexibility: Fundamental relations between two major properties and performance measures of the production system. International Journal of Production Economics 52, 73–82.

Grünert, T., Irnich, S., 2005. Optimierung im Transport - Band II: Wege und Touren. Shaker-Verlag.

Gupta, A., Maranas, C., 2003. Managing demand uncertainty in supply chain planning. Computers and Chemical Engineering 27, 1219–1227.

Gupta, Y. P., Goyal, S., 1989. Flexibility of manufacturing systems: Concepts and measurements. European Journal of Operational Research 43, 119–135.

Gutenschwager, K., Niklaus, C., Voß, S., 2004. Dispatching of an electric monorail system: Applying metaheuristics to an online pickup and delivery problem. Transportation Science 38 (4), 434–446.

Hadjiconstantinou, E., Roberts, D., 2002. Routing under Uncertainty: An Application in the Scheduling of Field Service Engineers. In: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. SIAM, pp. 331–253.

Hall, R., Partyka, J., 2008. On the Road to Mobility. OR/MS Today 2.

Hart, E., Ross, P., Nelson, J., 1998. Producing Robust Schedules Via an Artificial Immune System. In: Proceedings of CEC98. pp. 464–469.

Hax, A., Meal, H., 1975. Hierarchical integration of production planning and scheduling. In: Geisler, M. (Ed.), Logistics. Vol. 1 of TIMS Studies in the management sciences. MIT, pp. 53–69.

Hiller, B., Krumke, S., Rambau, J., 2006. Reoptimization gaps versus model errors in online-dispatching of service units for adac. Discrete Applied Mathematics 154 (13), 1897–1907.

Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T., 2004. Hybrid Modelling for Robust Solving. Annals of Operations Research 130, 19–39.

Hsu, C.-I., Hung, S.-F., Li, H.-C., 2007. Vehicle routing problem with time-windows for perishable food delivery. Journal of Food Engineering 80, 465–475.

Hülsmann, M., Windt, K. (Eds.), 2007. Understanding Autonomous Cooperation and Control in Logistics. Springer.

Huth, T., Mattfeld, D., 2008. Integration of Routing and Resource Allocation in Dynamic Logistic Networks. In: Haasis, H.-D., Kreowski, H.-J., Scholz-Reiter, B. (Eds.), Dynamics in Logistics. Springer, pp. 85–93.

Ichoua, S., Gendreau, M., Potvin, J.-Y., 2003. Vehicle dispatching with time-dependent travel times. European Journal of Operational Research 144, 379–396.

Illa, A. B., 1966. Critical probabilities and determinism in decision theory. Calcolo 2 (2), 117–130.

Inderfurth, K., 1994. Nervousness in inventory control: analytical results. OR Spektrum 16, 113–123.

Inderfurth, K., Jensen, T., 1997. Planungsnervosität im Rahmen der Produktionsplanung. Zeitschrift für Betriebswirtschaft 67 (8), 817–843.

Irani, S., Ku, X., Regan, A., 2004. On-line algorithms for the dynamic traveling repairman problem. Journal of Scheduling 7, 243–258.

Jacobs, J., 2003. Rechnerunterstützte Konfliktermittlung und Entscheidungsunterstützung bei der Disposition des Zuglaufs. Ph.D. thesis, RWTH Aachen.

Jedermann, R., Ruiz-Garcia, L., Lang, W., 2009. Spatial temperature profiling by semi-passive RFID loggers for perishable food transportation. Computer and Electronics in Agriculture 65, 145–154.

Jensen, M., 2001a. Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. Applied Soft Computing 1, 35–52.

Jensen, M., 2001b. Robust and Flexible Scheduling with Evolutionary Computation. Ph.D. thesis, University of Aarhus.

Jensen, T., 1996. Planungsstabilität in der Materiallogistik. Physica-Verlag.

Kadipasaoglu, S. N., Sridharan, S. V., 1997. Measurement of instability in multi-level MRP systems. International Journal of Production Research 35 (3), 713–737.

Kalleberg, A., Reynolds, J., Marsden, P., 2003. Externalizing employment: flexible staffing arrangements in us organizations. Social Science Research 32, 525–552.

Kaluza, B., Dullnig, H., Malle, F., 2003. Principal-Agent-Probleme in der Supply Chain – Problemanalyse und Diskussion von Lösungsvorschlägen. Tech. Rep. 2003/03, University of Klagenfurt, Austria.

Kennington, J., 1978. A Survey of Linear Cost Multicommodity Network Flows. Operations Research 26 (2), 209–236.

Keys, K., Moore, D., Estan, C., 2005. A Robust System for Accurate Real-time Summaries of Internet Traffic. In: Proceedings of the ACM SIGMETRICS 2005. pp. 85–96.

Kitano, H., Oda, K., 2005. Self-Extending Symbiosis: A Mechanism for Increasing Robustness Through Evolution. Biological Theory 1 (1), 61–66.

Koberstein, A., 2008. Progress in the dual simplex algorithm for solving large scale lp problems: techniques for a fast and stable implementation. Computational Optimization and Applications 41 (2), 185–204.

Kohl, N., Larsen, A., Larsen, J., Ross, A., Tiourine, S., 149–162 2007. Airline disruption management - Perspectives, experiences and outlook. Journal of Air Transport Management 13.

Kopfer, H., Krajewska, M., 2007. Approaches for Modelling and Solving the Integrated Transportation and Forwarding Problem. In: Corsten, H., Missbauer, H. (Eds.), Produktions- und Logistikmanagement. Verlag Franz Vahlen, pp. 439–458.

Kopfer, H., Meyer, C., 2009. A Model for the Traveling Salesman Problem including the EC Regulations on Driving Hours. In: Proceedings of OR2008. p. to appear.

Kopfer, H., Schönberger, J., 2002. Interactive Solving of Vehicle Routing and Scheduling Problems: Basic Concepts and Qualification of Tabu Search Approaches. In: Sprague, R. (Ed.), Proceedings of HICCS 35. Vol. 3. p. 84.

Krajewska, M., 2008. Potentials for efficiency increase in modern freight forwarding. Gabler.

Krajewska, M., Kopfer, H., 2008. Transportation planning in freight forwarding companies - tabu search algorithm for the integrated operational transportation planning problem. European Journal of Operational Research, to appear.

Krumke, S., Rambau, J., Torres, L., 2002. Real-Time Dispatching of Guided and Unguided Automobile Service Units with Soft Time Windows. In: Möhring, R., Raman, R. (Eds.), Proceedings of ESA 2002. Vol. 2461 of LNCS. Springer, pp. 637–648.

Lackner, A., 2004. Dynamische Tourenplanung mit ausgewählten Metaheuristiken. Cuvillier Verlag, Göttingen.

Lambert, D., Cooper, M., 2000. Issues in Supply Chain Management. Industrial Marketing Management 29, 65–83.

Larsen, A., Madsen, O., Solomon, M., 2008. Recent Developments in Dynamic Vehicle Routing. In: Golden et al. (2008), pp. 199–218.

Laudon, K., Laudon, J., 2006. Management Information Systems, 9th Edition. Pearson - Prentice Hall.

Laux, H., 1982. Entscheidungstheorie. Springer-Verlag.

L.Delaitre, Awasthi, A., Molet, H., Breuil, D., 2007. A multiple container loading problem based algorithm for efficient allocation of goods to vehicles. In: Proceedings of the 18th IASTED International Conference. pp. 313–317.

Leon, V., Wu, S., Storer, R., 1994. Robustness Measures and Robust Scheduling for Job Shops. IIE Transactions 26 (5), 32–43.

Lin, F.-R., Shaw, M., 1998. Reengineering the Order Fulfillment Process in Supply Chain Networks. The International Journal of Flexible Manufacturing Systems 10, 197–229.

Lindstaedt, H., 2007. Problemlösen und Verstehen bei ökonomischen Agenten - eine Gegenüberstellung ökonomischer und kognitionspsychologischer Modelle regelbasierten Entscheidens. NeuroPsychoEconomics 2, 30–43.

Littman, M., Dean, T., Kaelbling, L., 1995. On the Complexity of Solving Markov Decision Problems. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence. pp. 394–402.

Liu, Y.-H., 2007. A hybrid scatter search for the probabilistic traveling salesman problem. Computers & Operations Research 34, 2949–2963.

Lo, S.-C., Hall, R., 2008. The design of real-time logistics information system for trucking industry. Computers & Operations Research 35, 3439–3451.

Lund, K., Madsen, O., Rygaard, J., 1996. Vehicle Routing Problems with Varying Degrees of Dynamism. Tech. Rep. IMM-REP-1996-1, Institute of Mathematical Modelling (IMM), Technical University of Denmark, Lyngby.

Madsen, O., Tosti, K., Vælds, J., 1995. A heuristic method for dispatching repair men. Annals of Operations Research 61, 213–226.

Marmier, F., Varnier, C., Zerhouni, N., 2007. Robustness Measure for Fuzzy Maintenance Activities Schedule. In: Bologa, O., Dumitrache, I., Filip, F. G. (Eds.), Proc. of 4^{th} Conference on Management and Control of Production and Logistics. pp. 85–90.

McKendall Jr., A., Jaramillo, J., 2006. A tabu search heuristic for the dynamic space allocation problem. Computers & Operations Research 33, 768–789.

Meyer, C., Kopfer, H., 2008. Restrictions for the operational transportation planning by regulations on drivers' working hours. In: Bortfeldt, A., Homberger, J., Kopfer, H., Pankratz, G., Strangmeier, R. (Eds.), Intelligent Decision Support - Current Challenges and Approaches. Gabler, pp. 177–186.

Michael, K., McCathie, L., 2005. The pros and cons of RFID in supply chain management. In: Proceedings of the International Conference on Mobile Business 2005. IEEE, pp. 623–629.

Michalewicz, Z., 2000a. Constraint-preserving operators. In: Bäck et al. (2000b), pp. 62–68.

Michalewicz, Z., 2000b. Repair algorithms. In: Bäck et al. (2000b), pp. 56–61.

Mignon, D., Honkomp, S., Reklaitis, G., 1995. A framework for investigating schedule robustness under uncertainty. Computers & Chemical Engineering 19, 615–620.

Mitrović-Minić, S., Krishnamurti, R., Laporte, G., 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transportation Research Part B 38, 635–655.

Montemanni, R., Gambardella, L., 2005. The robust shortest path problem with interval data via benders decomposition. 4OR 3, 315–328.

Morales, J., 2007. Robust Duration-Constrained Tours for Vehicle Routing Problems with Stochastic Demands. In: Proceedings of TRISTAN VI.

Moreira, L., Oliveira, J., Gomes, A., Ferreira, J., 2007. Heuristics for a dynamic rural postman problem. Computers & Operations Research 34, 3281–3294.

Morlok, E., Chang, D., 2004. Measuring capacity flexibility of a transport system. Transportation Research Part A 38, 405–420.

Mulvey, J., Vanderbei, R., Zenios, S., 1995. Robust Optimization of Large-Scale Systems. Operations Research 43 (2), 264–281.

Munera, H., 1984. The generalized means model (gmm) for non-deterministic decision making: its normative and descriptive power, including sketch of the representation theorem. Theory and Decision 18, 173–202.

Nordhaus, W., 2003. The Progress of Computing. Tech. Rep. 1324, Cowles Foundation Discussion Paper.

Nowak, M., Ergun, O., III, C. W., 2008. Pickup and Delivery with Split Loads. Transportation Science 42 (1), 32–43.

Otto, A., Obermaier, R. (Eds.), 2007. Logistikmanagement. DUV.

Pankratz, G., 2002. Speditionelle Transportdisposition. Deutscher Universitäts-Verlag, Wiesbaden.

Pibernik, R., 2002. Flexibilitätsplanung in Wertschöpfungsnetzwerken. DUV.

Pibernik, R., Sucky, E., 2006. Centralized and decentralized supply chain planning. International Journal of Integrated Supply Management 2 (1/2), 6–27.

Pinedo, M., Chao, X., 1999. Operations Scheduling. McGrawHill.

Potvin, J.-Y., Xu, Y., Benyahia, I., 2006. Vehicle routing and scheduling with dynamic travel times. Computers & Operations Research, 1129–1137.

Powell, W., 1996. A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers. Transportation Science 30 (3), 195–219.

Powell, W., Cheung, R., 2000. Adaptive Labeling Algorithms for the Dynamic Assignment Problem. Transportation Science 34 (1), 50–66.

Psaraftis, H., 1988. Dynamic Vehicle Routing Problems. In: Golden, B., Assad, A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, pp. 223–248.

Psaraftis, H., 1995. Dynamic vehicle routing: status and prospects. Annals of Operations Research 61, 143–164.

Radatz, J., 1990. IEEE Standard Glossary of Software Engineering Terminology. Computer Society of the IEEE.

Rajagopalan, H., Saydam, C., Xiao, J., 2008. A multiperiod set covering location model for dynamic redeployment of ambulances. Computers & Operations Research 35 (3), 814–826.

Ramamritham, K., Stankovic, J., 1994. Scheduling and Operating System Support for Real-Time Systems. In: Proceedings of the IEEE. pp. 55–67.

Raney, B., Nagel, K., 2004. Iterative route planning for large-scale modular transportation simulations. Future Generation Computer Systems 20 (7), 1101–1118.

Rego, C., Roucairol, C., 1995. Using tabu search for solving a dynamic multiterminal truck dispatching problem. European Journal of Operational Research 83, 411–429.

Richter, A., 2005. Dynamische Tourenplanung. Ph.D. thesis, Technische Universität Dresden.

Ronen, B., Coman, A., Schragenheim, E., 2001. Peak management. International Journal of Production Research 39 (14), 3183–3193.

Rudzki, R., Smoch, D., Katzorke, M., Stewart Jr., S., 2005. Straight to the Bottom Line: An Executive's Roadmap to World Class Supply Management. J. Ross Publishing.

Saéz, D., Cortéz, C., Núñes, A., 2008. Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. Computers & Operations Research 35 (11), 3412–3438.

Sadler, I., 2007. Logistics and Supply Chain Integration. Sage Pubn Inc.

Sandvoß, E., 2004. Dynamische Tourenplanung auf Basis von Online-Verkehrsinformationen. Pro Business.

Sarker, R., Mohammadian, M., Yao, X. (Eds.), 2002. Evolutionary Optimization. Kluwer.

Savelsbergh, M., Sol, M., 1998. DRIVE: Dynamic Routing of Independent Vehicles. Operations Research 46, 474–490.

Schillo, M., Bürckert, H.-J., Fischer, K., Klusch, M., 2001. Towards a Definition of Robustness for Market-Style Open Multi-Agent Systems. In: Proceedings of the 5th international conference on autonomous agents. Springer, pp. 75–76.

Schneeweiß, C., 1992. Planung 2: Konzepte der Prozeß- und Modellgestaltung. Springer-Verlag.

Schneeweiß, C., 1994. Elemente einer Theorie hierarchischer Planung. OR Spektrum 16, 161–168.

Schneeweiß C., Schneider, H., 1999. Measuring and designing flexibility as a generalized service degree. European Journal of Operational Research 112, 98–116.

Scholl, A., 2001. Robuste Planung und Optimierung. Physica-Verlag.

Schönberger, J., 2005. Operational Freight Carrier Planning. Springer.
Schönberger, J., Kopfer, H., 2003. A Memetic Algorithm for a Pickup and Delivery Selection Problem. In: Mohammadian, M. (Ed.), Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation. published on CD, pp. 704–715.
Schönberger, J., Kopfer, H., 2005. Freight Flow Consolidation in Presence of Time Windows. In: Fleuren, H., den Hertog, D., Kort, P. (Eds.), Proceedings of Operations Research 2004. Springer, pp. 184–191.
Schönberger, J., Kopfer, H., 2007a. Autonomous Decision Model Adaptation and the Vehicle Routing Problem with Time Windows and Uncertain Demand. In: Hülsmann, M., Windt, K. (Eds.), Understand Autonomous Cooperation and Control in Logistics - The Impact of Autonomy on Management, Information, Communication and Material Flow. Springer, pp. 139–161.
Schönberger, J., Kopfer, H., 2007b. Flexible Transport Process Planning in Volatile Environments and the Adaptation of a Cost-Based Objective Function. In: Otto and Obermaier (2007), pp. 263–283.
Schönberger, J., Kopfer, H., 2008a. Manipulating the Decision Behavior of a Subordinate Service Centre by a Closed-Loop Cost Accounting Scheme Control. In: Suhl, L., Mattfeld, D., Günther, H.-O., Voß, S. (Eds.), Information- und Kommunikationssysteme in Supply Chain Management, Logistik und Transport. DSOR Contributions to Information Systems, pp. 3–19.
Schönberger, J., Kopfer, H., 2008b. On the Value of Objective Function Adaptation in Online Optimisation. In: Nickel, S., Kalcsics, J. (Eds.), Operations Research Proceedings 2007. Springer, pp. 491–496.
Schönberger, J., Kopfer, H., 2008c. Schedule Nervousness Reduction in Transport Re-Planning. Communications of SIWN 3, 7–14.
Schönberger, J., Kopfer, H., 2009a. Online Decision Making and Automatic Decision Model Adaptation. Computers & Operations Research 36 (6), 1740–1750.
Schönberger, J., Kopfer, H., 2009b. Rules for the Identification of Portfolio-Incompatible Requests in Dynamic Vehicle Routing. In: Hansen, H., Karagiannis, D., Fill, H. (Eds.), Proceedings of Wirtschaftsinformatik 2009. Vol. 2. OCG, pp. 35–44.
Schönberger, J., Kopfer, H., 2009c. Transport system responsiveness improvement. International Journal of Physical Distribution and Logistic Management 39 (1), 63–79.
Schönberger, J., Mattfeld, D., Kopfer, H., 2004. Memetic Algorithm Timetabling for Non-Commercial Sport Leagues. European Journal of Operational Research 153 (1), 102–116.
Scott, D., Novak, D., Aultman-Hall, L., Guo, F., 2006. Network Robustness Index: A new method for identifying critical links and evaluating the performance of transportation networks. Journal of Transport Geography 14, 215–227.
Scott, S., Lesh, N., Klau, G., 2002. Investigating human-computer optimization. In: Wixon, D. (Ed.), Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves. pp. 155–162.

Séguin, R., Potvin, J.-Y., Gendreau, M., Crainic, T., Marcotte, P., 1997. Real-time decision problems: an operational research perspective. Journal of the Operational Research Society 48 (2), 162–174.

Seiden, S., 1996. Randomized Online Scheduling with Delivery Times. Journal of Combinatorial Optimization 3, 399–416.

Sellmaier, S., 2007. Langfristiges Entscheiden. LIT Verlag.

Sethi, A., Sethi, S., 1990. Flexibility in Manufacturing: A Survey. International Journal of Flexible Manufacturing Systems 2, 289–328.

Sindhuchao, S., Romeijn, H., Akcali, E., Boondiskulchok, R., 2005. An Integrated Inventory-Routing System for Multi-item Joint Replenishment with Limited Vehicle Capacity. Journal of Global Optimization 32, 93–118.

Slater, A., 2002. Specification for a dynamic vehicle routing and scheduling system. International Journal of Transport Management 1, 29–40.

Smith, A., Coit, D., 2000. Penalty Functions. In: Bäck et al. (2000b), pp. 41–48.

Solnon, C., 2002. Boosting ACO with a Preprocessing Step. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G. (Eds.), Applications of Evolutionary Computing. Springer, pp. 163–172.

Solomon, M., 1987. Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations Research 35 (2), 254–265.

Sprague, R., Carlson, E., 1982. Building effective decision support systems. Prentice-Hall.

Stadtler, H., 2002. Supply Chain Management – An Overview. In: Stadtler and Kilger (2002), pp. 7–28.

Stadtler, H., 2005. Supply chain mangement and advanced planning – basics, overview and challenges. European Journal of Operational Research 163, 575–588.

Stadtler, H., Kilger, C. (Eds.), 2002. Supply Chain Management and Advanced Planning, 2nd Edition. Springer.

Stank, T., Goldsby, T., 2000. A framework for transportation decision making in an integrated supply chain. Supply Chain Management: An International Journal 5 (2), 71–78.

Stock, G., Greis, N., Kasarda, J., 2000. Entreprise logistics and supply chain structure: the role of fit. Journal of Operations Management 18, 531–547.

Suhong, L., Visich, J., Basheer, M., Khumawala, M., Chen, Z., 2006. Radio frequency identification technology: applications, technical challenges and strategies. Sensor Review 26 (3), 193–202.

Sundermeyer, K., 2001. Collaborative Supply Net Management. In: Baader, F., Brewka, G., Eiter, T. (Eds.), KI2001 - Advances in Artificial Intelligence. Springer, pp. 467–470.

Tang, H., Miller-Hooks, E., 2007. Solving a generalized traveling salesperson problem with stochastic customers. Computers & Operations Research 34, 1963–1987.

Tarhini, A., Fouchal, H., 2006. Robustness evaluation of real-time protocols. In: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services.

Thill, J.-C., 2000. Geographic information systems for transportation in perspective. Transportation Research Part C: Emerging Technologies 8 (1-6), 3–12.

Thommen, J.-P., 1991. Managementorientierte Betriebswirtschaftslehre, 3rd Edition. Verlag Paul Haupt.

Thonemann, U., 2005. Operations Management. Pearson Studium.

Timm-Giel, A., Subramanian, A., Dhanasekaran, K., Navda, K., Das, S., 2007. Improving Connectivity to Road Side Communication Units Using Steerable Directional Antennas in Vehicular Applications. In: CEWIT Conference.

Vaidyanathan, B., Matson, J., Miller, D., Matson, J., 1999. A capacitated vehicle routing problem for just-in-time delivery. IIE Transactions 31, 1083–1092.

Vernimmen, B., Dullaert, W., Engelen, S., 2007. Schedule Unreliability in Liner Shipping: Origins and Consequences for the Hinterland Supply Chain. Maritime Economics & Logistics 9, 193–213.

Villa, A., 2002. Emerging trends in large-scale supply chain management. International Journal of production Research 40 (15), 3487–3498.

Šegvić, S., Remazeilles, A., Chaumette, F., 2006. Enhancing the Point Feature Tracker by Adaptive Modelling of the Feature Support. In: Leonardis, A., Bischof, H., Prinz, A. (Eds.), ECCV 2006, Part II. pp. 112–124.

Wallentowitz, H., Freialdenhoven, A., Olschewski, I., 2009. Strategien in der Automobilindustrie. Vieweg+Teubner.

Wang, J., 2004. A fuzzy robust scheduling approach for product development projects. European Journal of Operational Research 152, 180–194.

Wang, N., Zhang, N., Wang, M., 2006. Wireless sensors in agriculture and food industryrecent development and future perspective. Computer and Electronics in Agriculture 50, 1–14.

Waters, C., 1984. Interactive vehicle routing. Journal of the Operational Research Society 35 (9), 821–826.

Wathne, K., Heide, J., 2004. Relationship Governance in a Supply Chain Network. Journal of Marketing 68, 73–99.

Whitley, D., 2000. Permutations. In: Bäck et al. (2000a), pp. 139–150.

Williams, T., 1984. Special products and uncertainty in production/inventory systems. European Journal of Operational Research 15 (1), 46–54.

Wöhe, G., 1993. Einführung in die Allgemeine Betriebswirtschaftslehre. Verlag Vahlen.

Woolridge, M., 2002. An introduction to multiagent systems. Wiley.

Woolridge, M., Jennings, N., Kinny, D., 1999. A Methodology for Agent-Oriented Analysis and Design. In: Proceedings of the Third International Conference on Autonomous Agents. pp. 69–76.

Yoshimoto, R., Nemoto, T., 2005. The impact of information and communication technology on road freight transportation. IATSS Research 29 (1), 16–21.

Yu, G., Yang, J., 1998. On The Robust Shortest Path Problem. Computers and Operations Research 25 (6), 457–468.

Yu, V., Sharma, D., Murty, K. G., 2008. Door Allocations to Origins and Destinations at Less-than-Truckload Trucking Terminals. Journal of Industrial and Systems Engineering 2 (1), 1–15.

Zäpfel, G., 1982. Produktionswirtschaft: Operatives Produktions-Management. de Gruyter.

Zeimpekis, V., Giaglis, G., 2005. A dynamic real-time fleet management system for incident handling in city logistics. In: Proceedings of 61^{st} IEEE Vehicular Technology Conference (VTC2005 Spring). pp. 2900–2904.

Zeimpekis, V., Minis, I., Mamassis, K., Giaglis, G., 2007. Dynamic Management of a Delayed Delivery Vehicle in a City Logistics Environment. In: Zeimpekis, V., Tarantilis, C., Giaglis, G., Minis, I. (Eds.), Dynamic Fleet Management. Springer, pp. 197–217.

Zieliński, P., 2004. The computational complexity of the relative robust shortest path problem with interval data. European Journal of Operational Research 158, 570–576.

Index

Q

R

S