549 LECTURE NOTES IN ECONOMICS
AND MATHEMATICAL SYSTEMS

Georg N. Krieg

Kanban-Controlled Manufacturing Systems

Springer

Lecture Notes in Economics and Mathematical Systems 549

Founding Editors:

M. Beckmann
H. P. Künzi

Managing Editors:

Prof. Dr. G. Fandel
Fachbereich Wirtschaftswissenschaften
Fernuniversität Hagen
Feithstr. 140/AVZ II, 58084 Hagen, Germany

Prof. Dr. W. Trockel
Institut für Mathematische Wirtschaftsforschung (IMW)
Universität Bielefeld
Universitätsstr. 25, 33615 Bielefeld, Germany

Editorial Board:

A. Basile, A. Drexl, H. Dawid, K. Inderfurth, W. Kürsten, U. Schittko

Georg N. Krieg

Kanban-Controlled Manufacturing Systems

Springer

Author

Dr. Georg N. Krieg
Chair of Production and Operations Management
Faculty of Business Administration and Economics
Catholic University of Eichstaett-Ingolstadt
85049 Ingolstadt, Germany
georg.krieg@web.de

Doctoral dissertation:
Faculty of Business Administration and Economics
Catholic University of Eichstaett-Ingolstadt, 2003
Referees: Professor Dr. Heinrich Kuhn and Professor Dr. Ulrich Küsters

Gedruckt mit Unterstützung des Förderungs- und Beihilfefonds Wissenschaft
der VG WORT

Library of Congress Control Number: 2004115950

ISSN 0075-8442
ISBN 3-540-22999-X Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera ready by author
Cover design: *Erich Kirchner*, Heidelberg

Printed on acid-free paper 42/3130Di 5 4 3 2 1 0

To my academic teachers and mentors
Prof. Dr. Werner Delfmann,
Prof. Dr. Heinrich Kuhn, and
Prof. Dr. Horst Tempelmeier
with my sincere thanks for their guidance
and support during these past years

Contents

1

Subject and Outline

The production management approach Just-In-Time (JIT) gained worldwide prominence when the rest of the world noticed the increasing success of Japanese companies in the late 1970s and early 1980s. As one major operational element of JIT, the kanban control system became a popular topic in western research and industry (e.g., Sugimori et al. 1977, Monden 1981a–d, 1998; Kimura and Terada 1981, Schonberger 1982, Hall 1983, Ohno 1988, Shingo 1989). Manufacturing companies outside Japan began to use kanbans to control production and flow of material. Several empirical studies document that kanban control bears great potential to significantly improve operations (e.g., White, Pearson, and Wilson 1999, Fullerton and McWatters 2001, White and Prybutok 2001).

Some operational improvements that follow the implementation of kanban systems are commonly attributed to organizational changes rather than to the kanban principle itself. A company, however, may reap the full benefits of kanban control only after determining an optimal or near-optimal system configuration. Finding such a configuration requires methods that can determine key performance measures, such as average fill rates and average inventory levels. Computer simulation may generally be used to analyze the performance of a system, but to identify an optimal configuration, many different system variants may have to be evaluated. To finish the search in a reasonable amount of time, the evaluation method should be fast—reliable simulation, however, is usually very time-consuming. Analytical (mathematical) evaluation methods are therefore needed that can determine key performance measures quickly, even if these methods only approximate the true performance of the system.

Some analytical evaluation methods can be found in the literature, particularly for systems with a single product. Kanban systems in industrial operations, however, usually control the production of several different products produced on shared manufacturing facilities (e.g., Anupindi and Tayur 1998). For the analysis of such multi-product kanban systems, we propose a construction-kit approach that makes it possible to build stochastic analytical models of a large class of single- and multi-product kanban systems.

Outline. In the following two chapters, we describe different implementations of kanban control, and we review the literature on stochastic models of kanban systems. The review shows that most models published so far represent single-product systems. In Chapter 4, we introduce the center part of our research: a construction-kit approach that yields new models of single- and multi-stage kanban systems with single- and multi-product manufacturing facilities.

The details of the construction-kit approach are given in Chapters 5–8. First, we develop three different one-product models that are the basic building blocks ("components") of the construction kit (Chapter 5). Then, in Chapter 6, we describe two procedures to build modules ("subassemblies") consisting of several instances of the second and the third one-product model, respectively. The subassemblies are models of kanban-controlled multi-product manufacturing systems with one and two production stage(s). They may be used to build composite models of systems with multiple stages. The general technique for linking models of single- and two-stage (sub-)systems is explained in Chapter 7. Technical restrictions limit the applicability of the basic version of the model construction kit to systems without multi-product facilities in immediate succession. A modeling trick, however, may be used to work around this limitation so that the extended version of the construction kit may be used to build models of systems with multi-product facilities in series (Chapter 8).

Since most models built with the construction kit only approximate the true behavior of the modeled systems, the quality of the approximation is of primary concern. We conducted systematic tests to examine the approximation quality for several important modeling examples. The results of these tests are reported in Chapter 9. Heuristic procedures were used to identify plausible kanban configurations for the test instances. The algorithms of these procedures are given in the appendix. In Chapter 10, we demonstrate how models generated with the construction kit may be used to study the behavior of kanban systems. We give numerous examples for different system variants. Finally, in Chapter 11, we conclude with a summary and directions for future research. A comprehensive list of all symbols and abbreviations is provided after the appendix (pp. 223–230).

2

Kanban-Controlled Manufacturing Systems: Basic Version and Variations

2.1 Basic Kanban System

The least complex variant of a kanban-controlled manufacturing system with multiple products is a system with a single multi-product manufacturing facility. Besides the production facility, the system contains a scheduling board, an output store for finished products, containers to store and carry finished items, and one set of kanbans for each product in the system (Fig. 2.1).

Traditionally, a kanban is a tag-like card (*kanban* is Japanese meaning "card" or "visible record" [Schonberger 1982, p. 17]). One kanban must be attached to each container in the output store. The number of kanbans is limited, restricting the maximum amount of finished items in the system. When a container is withdrawn, the accompanying kanban is detached from the container and placed on the scheduling board. Alternatively, the kanban may be detached when the last item is removed from the container (this is equivalent to using a fixed number of containers to limit the maximum inventory of a product). Also, removed kanbans may be put in a kanban collection box located in the output store before they are transferred to the scheduling board, either when a given number of kanbans has accumulated or when a specified

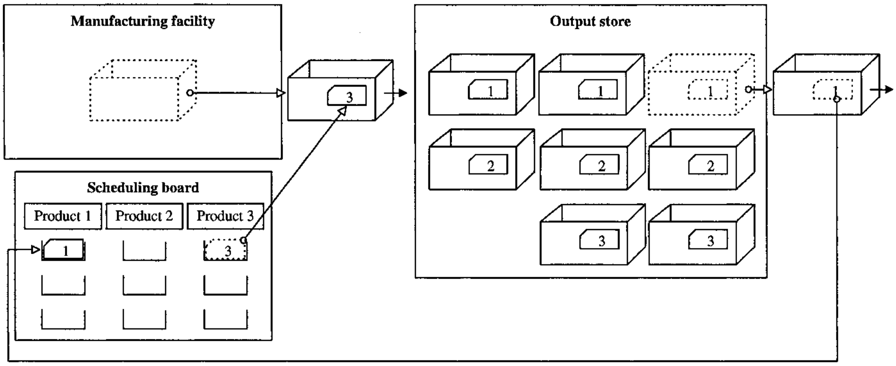


Fig. 2.1. Basic kanban system with three different products

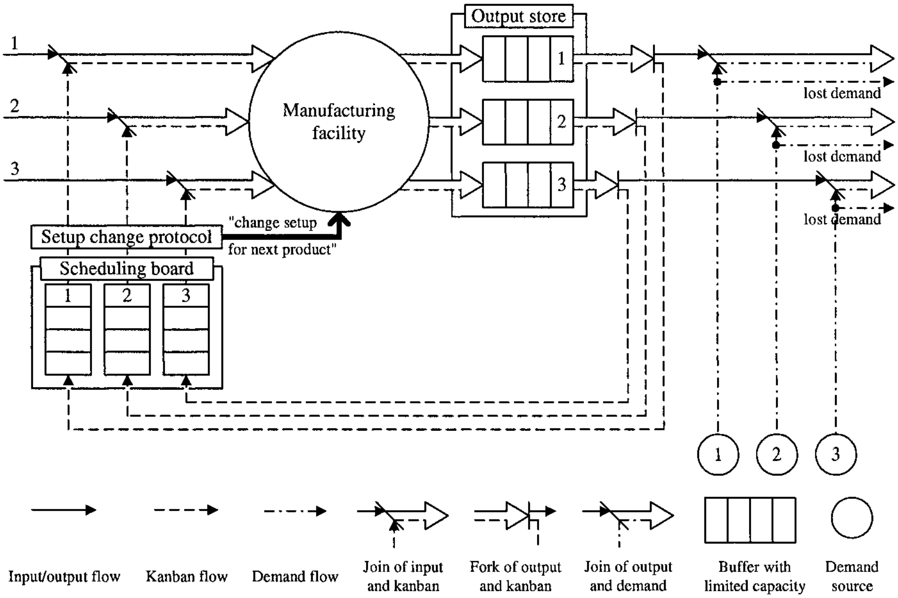


Fig. 2.2. Model of a single-stage multi-product kanban system without backorders (adapted from Mitra and Mitrani 1990, Fig. 3)

amount of time has elapsed from the last transfer. A detached or "active" kanban authorizes manufacture of one standard container of the product indicated on the card. When a container has been filled with the prescribed number of items, the now "inactive" kanban is affixed to the container and the container is transferred to the output store (Fig. 2.2).

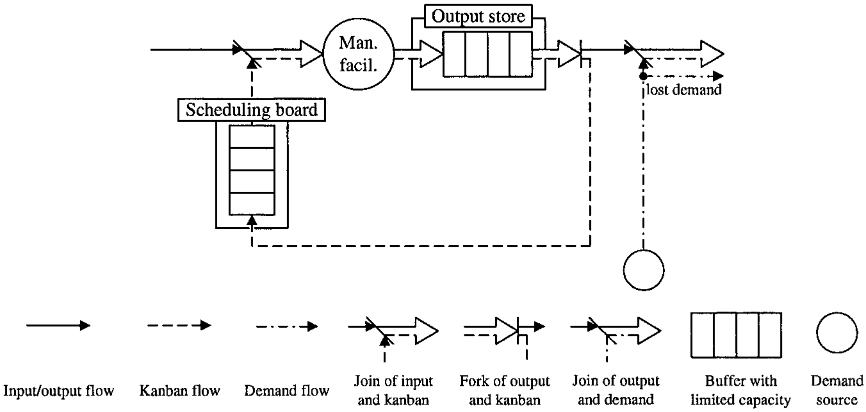


Fig. 2.3. Model of a single-stage single-product kanban system without backorders

Between the production of different products, setup changes must be performed that may consume a significant amount of time. A *setup change protocol* defines the rules for deciding when and to which other setup the current setup of the manufacturing facility should be changed next. One option is that items of a product are processed until there are no more active kanbans for this product on the scheduling board (*exhaustive processing*). Then the manufacturing facility is being set up for the next product according to a predetermined fixed *cyclic* setup sequence, for example, product 1, product 2, product 3 (repeated). Should no kanban be active for the next product, then this product is skipped. Should no kanban be active for any product, then the manufacturing facility idles. During the idle period, the current setup is maintained so that the manufacturing facility may immediately resume production if the next active kanban authorizes production of the product that was manufactured last. We refer to this setup change protocol as *cyclic-exhaustive processing*. Several other setup change protocols are suggested in the literature (e.g., Amin and Altiok 1997).

Single-product systems. A single-stage single-product kanban system is a special case of the basic kanban system. The manufacturing facility processes items as long as active kanbans are available and idles when all kanbans are inactive (Fig. 2.3). Setup changes and, hence, setup change protocols are not required.

2.2 Backorders

In the basic kanban system, customers whose demand cannot be filled from stock do not wait until the system can meet their demand. They either withdraw their request, or they turn to a different supplier who offers the same product, possibly at somewhat less favorable conditions (e.g., higher price, lower quality). As a result, no backorders accumulate in the basic kanban system.

In a different system, customers may have no alternative but to wait until their demand is satisfied. This is the standard situation for manufacturing stages that draw raw material or parts from a single supplier (the supplier may be the preceding manufacturing stage or an outside supplier). In those systems, the maximum number of backorders depends on the number of customers who generate requests. If there is only a single customer—the standard in serial manufacturing systems with one producing and one consuming manufacturing facility for each product—then the maximum number of backorders is one, and the demand source runs dry during the backorder situation. If there are several customers, then the maximum number of backorders equals the number of customers, and the average arrival rate of demand changes with the number of backorders. If customers have a local inventory of input material with a given target inventory level, then the maximum number of backorders a single customer can cause equals the target inventory level plus one (the additional backorder is due to the customer waiting for input material). If the number of customers is very large—a common situation for final products sold to private customers—then the number of backorders may become practically infinite, and the average arrival rate of demand is not significantly affected by the number of waiting customers.

It is also possible that a customer with outstanding orders continues generating new orders with constant average rate. This behavior, for example, may be observed in some assembly lines: rather than stopping the line, missing parts are added after the incomplete products have passed the last station and are waiting in a designated buffer area for unfinished products. In those systems, the number of backorders (for input material) is unlimited, even if there is only a single customer (the assembly system), and the average arrival rate of demand is independent of the number of backorders.

Finally, customers may accept a waiting time only if the total number of unfilled orders in the system is below a certain limit. Otherwise, they either withdraw their request or contact a different supplier. Assuming that all costumers consider the same number of backorders in the system prohibitive, then the maximum number of

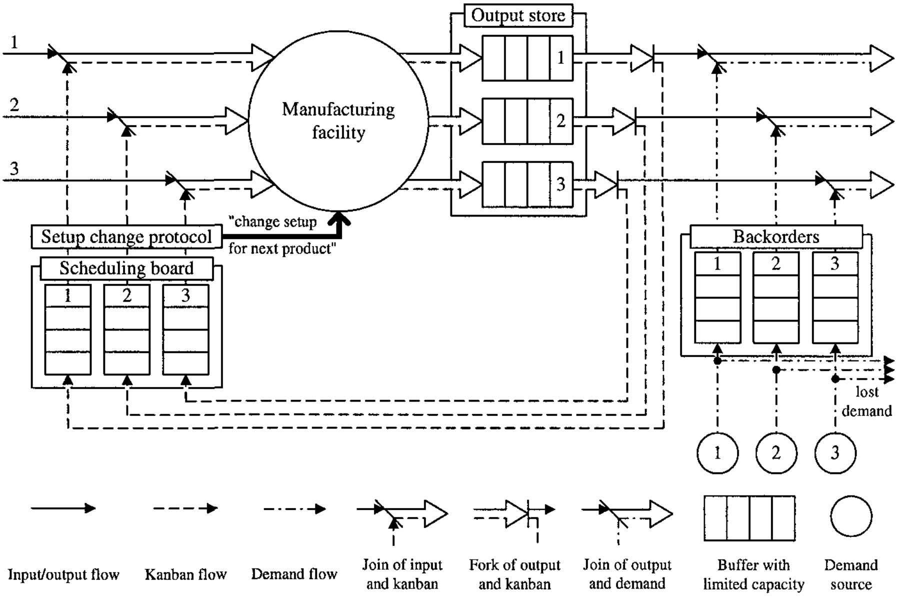


Fig. 2.4. Model of a single-stage multi-product kanban system with a limited number of backorders and lost demand

backorders equals the number of customers if the number of customers is less than the prohibitive number of unprocessed orders. Otherwise, the maximum number of backorders equals the prohibitive value. The average arrival rate of demand depends on the number of waiting customers, unless the number of customers is very large. If customers keep generating new requests irrespective of outstanding orders, then the maximum number of backorders always equals the prohibitive number of unprocessed orders, and the average arrival rate of demand is independent of the number of backorders in the system.

The model of a single-stage multi-product kanban system with a limited number of backorders and lost demand is illustrated in Figure 2.4.

2.3 Multiple Stages

In a kanban system with two or more production stages, the processed items of one stage are the (main) input material of the following stage. Unlike the manufacturing facility in the basic kanban system, the manufacturing facilities may experience shortage of input material (*starvation*). For modeling purposes, we define that in-

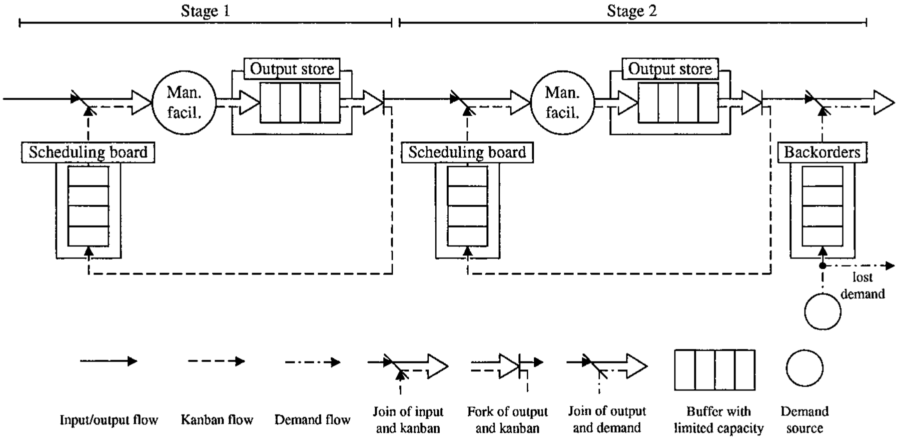


Fig. 2.5. Model of a two-stage single-product kanban system with type-1 material transfer, a limited number of backorders, and lost demand

put material for the first stage is always available in the required quantities, that is, manufacturing facilities in the first stage never starve. If manufacturing facilities in the first stage of the modeled system may experience shortage of input material, we declare that the first stage of the model corresponds to the procurement process of the input material for the first stage of the real system, and that the second stage of the model represents the first stage of the modeled system.

Production of each product in each stage is controlled by a distinct kanban loop with a fixed number of kanbans (Figs. 2.5 and 2.6). Immediately before the beginning of production, a container with input material is withdrawn from the output store of the preceding stage. Should no material be available, then the manufacturing facility either waits until new material arrives, or the setup is changed to process items of a different product.

The setup change protocol in multi-stage kanban systems must consider that, at some times, input material may not be available for products with active kanbans. One possible setup change protocol is *cyclic-exhaustive processing with limited input material.* With this setup change protocol, the precondition for a setup is that at least one active kanban and one container with input material must be available. Once the manufacturing facility has been set up for a specific product, it processes items of this product until either the number of active kanbans is zero, that is, all empty containers for the product have been filled, or the input material is depleted. Then the manufacturing facility is being set up for the next product that meets the setup con-

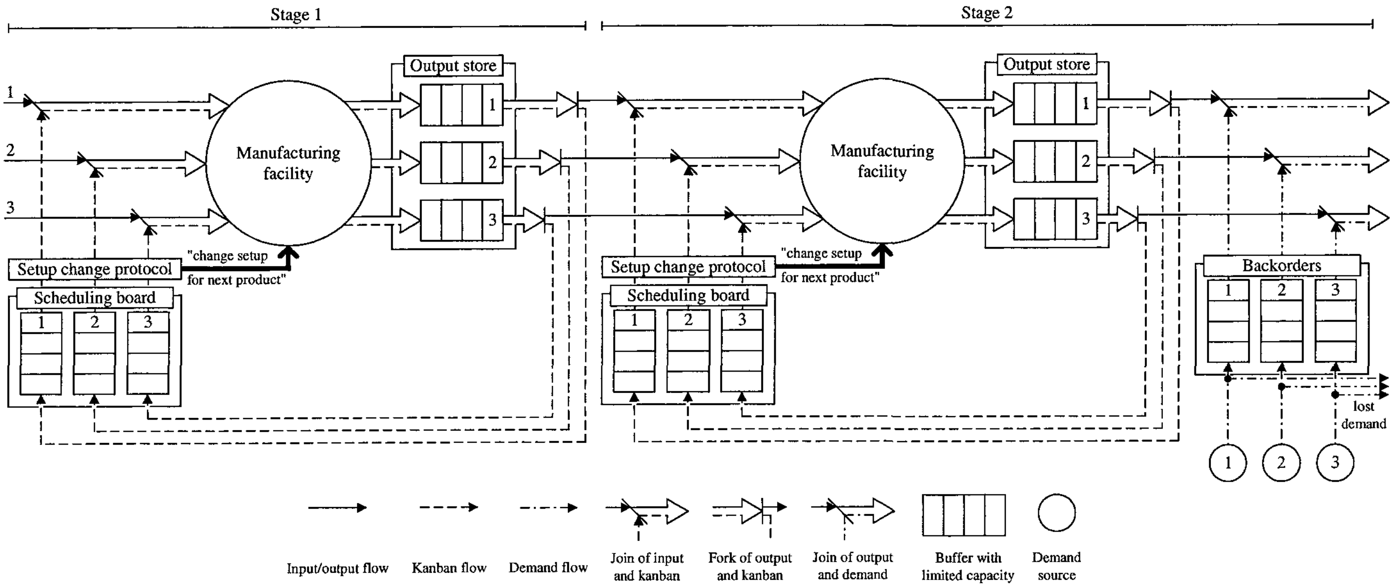


Fig. 2.6. Model of a two-stage multi-product kanban system with type-1 material transfer, a limited number of backorders, and lost demand

Table 2.1. Classification of Material Transfer Schemes

Output Store Stage m $=$ Input Store Stage $m+1$	Output Store Stage m $\neq$ Input Store Stage $m+1$		
Type 1 *Withdrawal immediately before start of production*	Type 2 *Withdrawal immediately after activation of kanban*	Type 3 *Fixed quantity, variable withdrawal cycle*	Type 4 *Fixed withdrawal cycle, variable quantity*
One-Card System		Two-Card System	

dition. The order in which products are considered for production is stipulated by a predetermined fixed cyclic setup sequence, for example, product 1, product 2, product 3 (repeated). Should no product meet the setup condition, then the manufacturing facility idles until one product can offer at least one active kanban and one container with input material. If the first product to meet the setup condition is the same product that was manufactured last, then no setup activities are required and production may resume instantly. Otherwise, a setup change is initiated and production starts upon completion of the setup process.

2.4 Material Transfer Schemes

Multi-stage kanban systems may be classified by the rules for transferring containers from the output store of one stage, say stage m, to the input store of the following stage, say stage $m+1$. At least four different set of rules may be found in the literature. In some systems, the output store of a stage is also the input store of the next stage. Consequently, there is no need for a transfer mechanism. In systems with separate output and input stores, the transfer of containers from the output store to the input store may be executed at different points in time. In Table 2.1, we summarize four different material transfer schemes. Each type is shortly explained in the following paragraphs. A discussion of the different transfer types is provided by Berkley (1991).

Type-1 material transfer. The output store of a stage is also the input store of the following stage, and material is withdrawn from the store immediately before start of production. This scheme has been labeled "late material transfer" by Gstettner and Kuhn (1996).

Type-2 material transfer. The output store of a stage is physically separated from the input store of the next stage. The material from the output store of stage m is withdrawn immediately after activation of a kanban in stage $m+1$ (the active kanban authorizes the withdrawal of a container with input material). The kanban is attached to the container and both join the queue in front of the manufacturing facility of stage $m+1$. If the output store of stage m is empty upon activation of a kanban in stage $m+1$, then the transfer is delayed until the manufacturing facility of stage m completes a container with the appropriate parts. This scheme, first described by Mitra and Mitrani (1990), has been labeled "immediate material transfer" by Gstettner and Kuhn (1996).

Type-3 and type-4 material transfer. The output store of a stage is physically separated from the input store of the following stage, and an additional set of cards, called *withdrawal*, *conveyance*, *delivery*, *move*, or *transportation kanbans*, is used to organize the transfer of containers between the stages (e.g., Monden 1998, Chap. 2). These systems are commonly referred to as *two-card* or *dual-card kanban systems*, in contrast to *one-card* or *single-card kanban systems* that only use a single card type. A withdrawal kanban must be attached to each container in the input store of a stage. When a container is taken up for production, the withdrawal kanban is removed and put into a kanban collection box. Eventually, a carrier takes the withdrawal kanbans out of the box and moves to the output store of the preceding stage. There, he withdraws a full container for each withdrawal kanban in his possession, removes the regular kanban from each container, and attaches one of the withdrawal kanbans instead (the regular kanbans are often called *production kanbans* in two-card kanban systems). Then he carries the containers to the input store of stage $m+1$. The removed production kanbans are put into a box from which they are eventually collected by a worker who places them on the scheduling board of stage m. If the carrier finds fewer containers in the output store than he holds withdrawal kanbans in his possession, then he returns the extra kanbans to stage $m+1$ and puts them back into the kanban collection box in the input store of stage $m+1$.

The point in time when the carrier removes the withdrawal kanbans from the kanban collection box is determined by one of two different schemes: (1) fixed quantity, variable withdrawal cycle (type-3 material transfer), or (2) fixed withdrawal cycle, variable quantity (*periodic material handling*, type-4 material transfer). In the first scheme, the carrier removes the withdrawal kanbans when a predetermined fixed number of cards has accumulated. The length of the withdrawal cycle, that is, the time between consecutive material transfers, may therefore vary. With the second scheme, the carrier removes the kanbans periodically, following a predetermined fixed schedule. Here, the withdrawal cycle is fixed and the number of cards may vary. Note that type-3 material transfer is equivalent to type-2 material transfer if the fixed withdrawal quantity is set to one.

3

Literature Review: Models of Kanban Systems

In this chapter, we review the literature on analytical stochastic models of kanban-controlled manufacturing systems. The review reveals that several mathematically tractable analytical models exist, however, almost exclusively for the evaluation of single-product systems. For multi-product kanban systems, only a small number of models may be found.

Other reviews are provided by Uzsoy and Martin-Vega (1990), Berkley (1992b), Singh and Brar (1992), Groenevelt (1993), Huang and Kusiak (1996), and Akturk and Erhun (1999). They also include simulation studies and deterministic models.

Unless explicitly stated otherwise, the manufacturing facilities are assumed to be perfectly reliable, input material (first stage) is always available in the quantities needed, containers contain only good parts, no parts are scrapped, transportation times between stages and withdrawal times are negligible, all processes are stochastic, all random variables are mutually independent, each product is produced and demanded by one manufacturing facility only, and a single (main) input is permitted for each processing operation (other inputs are admissible if they are always available). Systems that deviate from these assumptions are listed as "different systems."

In the systems that we consider here as kanban systems, several single- and/or multi-product manufacturing facilities (in parallel) may belong to a production stage, but a set of kanbans always controls the production of exactly one product in exactly one manufacturing facility, and the manufacturing facility can only process one item at a time. Other authors have a different, more general understanding of kanban systems. Schömig (1997, Section 4.5), for example, analyzes systems in which the same set of kanbans is used for all products of a stage. Di Mascolo, Frein, and Dallery (1992, 1996) study systems in which one set of kanbans is used to control the production of one product in one production stage, but there may be several manufacturing facilities and intermediate buffers in the production stage, arranged in series or in more complex formations (also De Araújo, Di Mascolo, and Frein 1993, Duri, Frein, and Di Mascolo 1995, Di Mascolo 1996, and Baynat et al. 2001). Hence, more than one item of a product may be being processed at the same time. In the review, we consider the results presented in these papers as far as they apply to systems with one manufacturing facility per stage.

3.1 Single-Product Kanban Systems

3.1.1 Single-Stage Systems

In this section, we review articles on analytical evaluation procedures for single-product kanban systems with one kanban-controlled production stage. A comprehensive presentation with additional system variations may be found in Buzacott and Shanthikumar 1993 (Section 4.3) under the label "single-stage single-product-type produce-to-stock systems."

Limited number of backorders. Jordan (1988) notes that closed-form results for queueing networks with exponential and Erlang distributions may be used to obtain steady-state performance measures for single-stage single-product kanban systems in which the demand is generated by a downstream manufacturing facility that is never blocked (it is always producing, provided input material is available). In this system, the maximum number of backorders is one ($B^{\max} = 1$) because the consuming manufacturing facility stops generating additional demands—as it stops producing (and thus consuming inputs)—when it lacks one container of input material (this is the situation where the number of backorders is one for the kanban-controlled manufacturing facility). For the case that the processing times of the producing and the consuming manufacturing facility are exponentially distributed with identical container processing rates, he gives equations for the average production rate and for the average number of full containers in the system (including the container in the downstream facility). The results are exact.

Karmarkar and Kekre (1989) model single-stage single-product kanban systems in which withdrawal kanbans are used to control the removal of full containers from the output store of the manufacturing facility. The removed containers are transported to a different, physically separated storage area from which external demand is met. The maximum number of backorders is given by the number of withdrawal kanbans. This implies that no customer waits for a full container if none is available at the time of his arrival. Since at least one withdrawal kanban must be present in the system, the maximum number of kanbans is always greater than zero ($B^{\max} \geq 1$). Demand arrivals are assumed to be Poisson, and processing times are exponential. The authors observe that their model is a truncated (finite) birth-and-death process, and they derive closed-form expressions for the steady-state probabilities of the system states (Equation (7) in the paper should be revised to read $p(i) = \left[\rho^{N-i}(1-\rho)\right]/(1-\rho^{N+M+1})$, $i = -M, -M+1, \ldots, 0, 1, \ldots, N$). The results are exact. The model of the kanban system is equivalent to the model of the standard $M/M/1/N$ queueing system: the customers in the queueing system correspond to the active kanbans and backorders in the kanban system, and the maximum number of customers in the queueing system, N, is equal to the sum of the number of kanbans and the maximum number of backorders, $K + B^{\max}$, in the kanban system.

Unlimited number of backorders. Karmarkar and Kekre (1989) observe that single-stage single-product kanban systems with exponential processing times, Poisson demand arrivals, and an unlimited number of backorders may be described exactly by a semi-infinite birth-and-death process. The authors provide a closed-form expression for the steady-state probability distribution of the system. The results are exact. This particular infinite-state Continuous-Time Markov Chain (CTMC) is equivalent to the CTMC of the standard $M/M/1$ queueing system and, hence, the same closed-form expressions for the steady-state probabilities of the system states apply.

Kim and Tang (1997) employ Erlang-k distributed times between the activation of kanbans where the number of Erlang phases, k, is equal to the container size as they assume that demand for single *items*, as opposed to demand for single *containers*, arrives according to a Poisson process. Each container is filled with k items. The processing time for a container of items (including a setup time) is assumed to be exponentially distributed. The model of this kanban system is equivalent to the model of a standard $E_k/M/1$ queueing system. This type of queueing system is also known as "bulk service system" since the server provides service to bulks of size k. Closed-form expressions exist for the steady-state probabilities of the states (e.g., Kleinrock 1975, Section 4.4). The results are exact.

Seki and Hoshino (1999) use the equivalence to an $M/E_k/1$ queueing system ("bulk arrival system") to model single-product single-stage kanban systems with Erlang-k processing times and Poisson demand arrivals. The authors assume that demand occurs for single items, and that a kanban is attached to each single finished item. The same queueing model may be employed to analyze single-product single-stage kanban systems with demand occurring for containers of size k and identically distributed exponential *item* processing times (hence, Erlang-k container processing times). Closed-form expressions exist for the steady-state probabilities of the states (e.g., Kleinrock 1975, Section 4.3). The results are exact. Besides the stationary behavior of the kanban system, Seki and Hoshino (1999) mainly focus on the *transient* system behavior.

Different systems. So and Pinault (1988) consider single-stage single-product kanban systems in which a minimum number of kanbans must be active before processing may start (threshold policy). They assume exponential container processing times, Poisson demand arrivals (for containers), and an unlimited number of backorders. The authors observe that their system is equivalent to an $E_k/M/1$ queueing system. For systems with general demand interarrival and processing times, they propose to use a scaling factor in the calculations based on the coefficients of variation of the processing time and the demand interarrival time. They suggest an expression for this scaling factor based on empirical results. In addition, the authors describe further modifications to include the possibility of machine breakdowns and multiple parallel manufacturing facilities in the stage. The results are exact for systems with one manufacturing facility, no breakdowns, exponential container processing times, and Poisson demand arrivals (for containers). For all other systems, the results are approximate.

Jordan (1988) sketches a discrete-time Markov chain for a basic assembly system with two kanban-controlled supplying manufacturing facilities and one assembly facility that is never blocked. As soon as one supplier is out of stock, the demand process is interrupted for both supplying manufacturing facilities. Hence, the maximum number of backorders is one for each supplier. The results are exact if the processing times of the modeled system follow a geometric distribution.

Wang and Wang (1990, 1991a, 1991b) model single-stage single-product kanban systems for which demand is generated by a manufacturing facility that is never blocked. In contrast to most other kanban systems, production is not controlled by cards, but by a limited number of containers, although production and withdrawal kanbans do circulate in the system. The manufacturing facility may start processing items only if an empty container is available. The consuming stage releases a container only after the last item has been removed. In this type of kanban system, the maximum number of full containers in the output store equals the number of containers minus one. As long as at least one container is partially filled, this container resides in the consuming stage. A backorder condition exists when all containers are empty. The authors assume exponential processing times in both stages and model the system as a CTMC. They determine the steady-state probabilities of the system states by solving the balance equations of the CTMC. The results are exact. Nori and Sarker (1998) note that the model is equivalent to the standard $M/M/1/N$ queueing model. They suggest to use the closed-form expressions available for this queueing model to find the steady-state probabilities of the states of the system.

Wang and Wang (1990, 1991b) also describe a CTMC for assembly systems with several kanban-controlled manufacturing facilities supplying parts or subassemblies to one assembly facility. The processing times are assumed to be exponentially distributed. The authors solve the balance equations numerically to obtain the steady-state probabilities of the system states. The results are exact.

3.1.2 Two-Stage Systems

Type-1 material transfer, no backorders. Karmarkar and Kekre (1989) illustrate a finite-state two-dimensional CTMC for two-stage single-product kanban systems with exponential processing times and Poisson demand arrivals. Demand that cannot be filled immediately is lost (no backorders), and the second stage withdraws material from the output store of the first stage just before start of production (type-1 material transfer). Since no closed-form solution exists for this CTMC, the authors determine the steady-state probabilities of the system states by solving the system of linear equations given by the balance equations. The results are exact.

3.1.3 Multi-Stage Systems

Type-1 Material Transfer

No backorders. Buzacott and Shanthikumar (1993, Section 5.5) treat kanban systems with type-1 material transfer as "produce-to-stock flow lines." They explain how their decomposition approach for finite-buffer flow lines with general processing times may be adapted for the analysis of multi-stage single-product kanban systems with type-1 material transfer, general processing times, Poisson demand arrivals, and no backorders (Buzacott and Shanthikumar 1993, Section 5.5.1). They model the output store of the last stage with the arrival and the departure of containers as an $G/M/1/N$ queueing system where the number of customers in the system (including the one in service) corresponds to the number of full containers in the output store, and the service completion instant corresponds to the instant when an arriving demand is met from stock. The authors also describe how a $G/G/1/N$ queueing system may be used to represent a general demand arrival process with independent and identically distributed interarrival times. The results of both procedures are approximate.

Limited number of backorders. Several authors consider multi-stage single-product kanban systems with type-1 material transfer and infinite demand at the last stage (Berkley 1990, 1991; Mitra and Mitrani 1990 [they refer to kanban systems with type-1 material transfer as systems with "modified kanban discipline"], and Wang and Wang 1991a). Alternatively, those systems may be viewed as multi-stage systems with a limited number of backorders ($B^{\max} = 1$), where the last stage is the customer of the system and a backorder condition exists when the last stage is waiting for processed items of the preceding stage. The authors observe an equivalence to tandem queueing networks and finite-buffer flow lines with tandem configuration and propose the use of evaluation procedures developed for these systems to approximate steady-state performance measures of the considered multi-stage kanban systems. Gstettner and Kuhn (1996) show in detail how the algorithm given in Buzacott and Shanthikumar 1993 (p. 200) for finite-buffer flow lines may be applied to the analysis of kanban systems. They also report results of comparisons between approximations obtained with this algorithm and point estimates determined with stochastic simulation.

Unlimited number of backorders. Buzacott and Shanthikumar (1993, Section 5.5.2) describe how their decomposition approach for finite-buffer flow lines with general processing times may be adapted for the analysis of kanban systems with type-1 material transfer, general processing times, a general demand arrival process, and an unlimited number of backorders. They use a $G/G/1$ queueing system to model the output store of the last stage. The results are approximate.

Different systems. Siha (1994) proposes a CTMC for multi-stage single-product kanban systems with exponential processing times and Poisson demand arrivals where the number of parallel manufacturing facilities in a stage is equal to the number of kanbans. As a consequence, processing may start instantly when a kanban is activated, provided that input material is available. The author determines the steady-state probability of each state by solving the system of linear equations given by the balance equations. Thus, he obtains exact values for steady-state performance measures, but he also encounters the dimensionality problem of CTMC which limits this approach to systems with a few stages and a modest number of kanbans.

Deleersnyder et al. (1989) describe a finite-state discrete-time Markov chain for multi-stage single-product kanban systems with type-1 material transfer, constant and for all machines identical processing times, symmetric binomial demand arrivals, a limited number of backorders ($B^{\max} \geq 1$), and unreliable machines. The results are exact. The curse of dimensionality, however, also applies to this discrete-time Markov chain.

Schmidbauer and Rösch (1994) consider multi-stage single-product kanban systems with type-1 material transfer, constant and for all machines identical processing times, Poisson demand arrivals, no backorders, and unreliable machines. They propose a decomposition approach where the single-stage subsystems are modeled as discrete-time Markov chains. The results are approximate.

Type-2 Material Transfer

Limited number of backorders. Mitra and Mitrani (1990) consider multi-stage single-product kanban systems with type-2 material transfer, exponential processing times, and infinite demand for full containers of the last stage. Alternatively, the manufacturing facility of the last stage may be interpreted as the costumer of the system. The manufacturing facility of the last stage generates demands with exponential interarrival times and may cause a limited number of backorders ($B^{\max} =$ number of kanbans of the last stage). The authors develop a decomposition approach to approximate steady-state performance measures. Albino, Dassisti, and Okogbaa (1995) and Fujiwara et al. (1998) consider the same type of kanban system and propose modifications to the decomposition approach introduced by Mitra and Mitrani (1990). The authors of the first paper also describe the details of an exact model of the system, a finite-state CTMC. The authors of the second paper extend the model by considering that raw parts for the first stage are procured from several suppliers with exponential procurement lead times.

Di Mascolo, Frein, and Dallery (1992, 1996) analyze systems very similar to the ones studied by Mitra and Mitrani (1990). The main difference is that processing times, and, thus, the interarrival times of the last stage's demands, may be phase-type distributed. The authors also use a more general definition of kanban-controlled systems: one set of kanbans may control the production of a product on several manufacturing facilities so that several items may be in process at the same time. This definition includes the type of kanban system considered by Mitra and Mitrani (1990)—one manufacturing facility (and one product) per set of kanbans—as a special case. Di Mascolo, Frein, and Dallery (1992, 1996) also propose an approximate decomposition method.

Unlimited number of backorders. So and Pinault (1988) develop a heuristic approach for estimating the average fraction of backordered demand for two-stage single-product kanban systems with type-2 material transfer, exponential processing times, Poisson demand arrivals, and an unlimited number of backorders. They state that a similar heuristic may be used for systems with more than two stages.

Mitra and Mitrani (1991) present a decomposition approach for multi-stage single-product kanban systems with type-2 material transfer, exponential processing times, Poisson demand arrivals, and an unlimited number of backorders. The results are approximate. The proposed procedure is a modified version of the one given in Mitra and Mitrani 1990 for systems with a limited number of backorders.

Di Mascolo, Frein, and Dallery (1992, 1996) also propose a decomposition approach for the analysis of multi-stage single-product kanban systems with type-2 material transfer, Poisson demand arrivals, and an unlimited number of backorders. The main difference to the systems analyzed by Mitra and Mitrani (1991) is that processing times may be phase-type distributed. Di Mascolo (1996) extends this approach for systems with more general demand arrival processes (approximated by phase-type distributions). The results of both procedures are approximate. Baynat et al. (2001) present a different way of deriving the decomposition approach of

Di Mascolo, Frein, and Dallery (1992, 1996). They develop a computational algorithm that appears to be considerably faster than the original.

Different systems. So and Pinault (1988) describe a procedure for estimating the average fraction of backordered demand of two-stage systems with unreliable machines. They also indicate how this approach may be extended to systems with more than two stages.

De Araújo, Di Mascolo, and Frein (1993) and Baynat et al. (2001) modify the decomposition method proposed by Di Mascolo, Frein, and Dallery (1992, 1996) for systems with two suppliers for the (identical) input material of the first stage (merge-structure) and with two customers for the (identical) output of the last stage (split-structure). Di Mascolo and Dallery (1996) and Baynat et al. (2001) describe how the decomposition method may be extended to represent the manufacture of products with more than one input (assembly-structure).

Xiaobo, Gong, and Wang (2002) consider multi-stage single-product kanban systems with type-2 material transfer, exponential processing times, and one vehicle between successive stages. Here, transfer times are modeled explicitly. The times for one way are assumed to be exponentially distributed, hence, the demand interarrival times for the last manufacturing facility are Erlang-2. The number of backorders for finished products of the last kanban-controlled manufacturing facility is limited to one, since the vehicle between the output store of this manufacturing facility and a storage area with unlimited capacity waits at the output store for the arrival of the next container with finished products, if it finds the output store empty after returning from the storage area. The authors propose a decomposition approach to approximate steady-state performance measures.

Type-3 Material Transfer

Limited number of backorders. Berkley (1990, 1991, 1994) describes a CTMC for multi-stage single-product kanban systems with type-3 material transfer, fixed withdrawal quantity equal to one, exponential or phase-type processing times, zero transportation and kanban handling times, and infinite demand at the last stage. Alternatively, the manufacturing facility of the last stage may be viewed as the demand source for the upstream part of the system that generates a new demand each time it completes filling a container, that is, immediately before starting production for the next container to be filled. Demand interarrival times for the upstream part of the system are therefore either exponentially or phase-type distributed, depending on the processing time distribution of the manufacturing facility of the last stage, and the number of backorders is limited to the number of withdrawal kanbans of the last stage plus one. The model provides exact results for steady-state performance measures. The dimensionality problem of Markov-chain analysis, however, limits the applicability of this approach to systems with very few stages and kanbans. Even with one-phase—that is, exponential—processing times, the number of states of the CTMC is soon too large to construct and solve the model in reasonable time.

Berkley (1991) shows that these kanban systems are equivalent to tandem queueing networks (it is crucial that the fixed withdrawal quantity is one and that transportation and kanban handling times are zero). The same decomposition methods that have been proposed for tandem queueing networks, finite-buffer flow lines, and multi-stage single-product kanban systems with type-1 material transfer and a limited number of backorders may therefore be used to approximate the steady-state behavior of these kanban systems.

Type-4 Material Transfer

Limited number of backorders. Berkley (1992a) introduces a decomposition approach for multi-stage single-product kanban systems with *periodic material handling*. He assumes that material transfers between all stages take place simultaneously according to the same withdrawal cycle, and that the transfer times are zero. Processing times are Erlang-k and the maximum number of backorders is equal to the number of withdrawal kanbans of the consuming manufacturing facility plus one. The results of the presented procedure are approximate.

3.2 Two-Product Kanban Systems

3.2.1 Single-Stage Systems

No backorders. Duri, Frein, and Di Mascolo (1995) develop a finite-state CTMC of a single-stage two-product kanban system with mutually independent Poisson demand arrivals, no backorders, zero setup times, and either exponential or Cox-2 processing times (they employ the two-phase Cox distribution to approximate general processing times). When kanbans are active for both products, the product to be processed next is determined randomly (to approximate first-come-first-served processing policy in the modeled system). The probability that a particular product is chosen is set equal to the proportion of this product's number of active kanbans to the total number of active kanbans. The proposed model is a finite-state CTMC, and the authors solve the balance equations numerically to determine the steady-state probabilities of the different states. The results are exact for the assumed random processing discipline, they are approximate for the first-come-first-served processing policy.

3.2.2 Multi-Stage Systems

Type-2 material transfer, unlimited number of backorders. Duri, Frein, and Di Mascolo (1995) extend the decomposition method introduced by Di Mascolo, Frein, and Dallery (1992, 1996) for the analysis of multi-stage kanban systems with type-2 material transfer, manufacturing facilities shared by two products, mutually independent Poisson demand arrivals, an unlimited number of backorders, zero setup times, first-come-first-served processing policy (approximated by a random processing discipline), and either exponential or general processing times (approximated by two-phase Cox distributions). The results are approximate.

3.3 Multi-Product Kanban Systems

3.3.1 Single-Stage Systems

Negligible Setup Times

No backorders or a limited number of backorders. In their book on stochastic models of manufacturing systems, Buzacott and Shanthikumar have a section on “multiple-product-type produce-to-stock systems” (1993, Section 4.4). These systems are equivalent to single-stage multi-product kanban systems with a fully flexible manufacturing facility (zero setup times). The authors assume mutually independent Poisson demand arrivals (with potentially different average arrival rates) and identically distributed processing times for the products. They employ an analogy to multi-class queueing systems and give exact equations for the steady-state probabilities of the system states for systems with exponential processing times, first-come-first-served service protocol, and no backorders ($B^{\max} = 0$) or a limited number of backorders ($B^{\max} \geq 1$). They also provide approximations for systems with general processing times, an unlimited number of backorders, and first-come-first-served or non-preemptive priority processing protocol.

State-Independent Setups

No backorders. Krieg and Kuhn (2002a) describe a decomposition method for single-stage multi-product kanban systems with a shared manufacturing facility, exponential processing and setup times, mutually independent Poisson demand arrivals, no backorders, exhaustive processing, and state-*in*dependent setups (setup changes are performed even if no kanban is active). This setup policy implies that the manufacturing facility never idles, it is either processing items or being set up for the next product. While a system with state-independent setups is easier to analyze—compared to a system with state-dependent setups—it is less realistic. Should the total traffic intensity be low, potential manufacturing time is lost and setup costs are incurred due to unnecessary setup changes.

Unlimited number of backorders. In the context of stochastic lot scheduling problems (a review is given by Sox et al. 1999), Federgruen and Katalan (1996) and Markowitz, Reiman, and Wein (2000) analyze systems very similar to the single-stage multi-product kanban systems studied by Krieg and Kuhn (2002a). Setup changes are also performed independently of the current state of the system. However, an idle time of deterministic length may be inserted between a production run and the next setup change, and the number of backorders may not be limited.

State-Dependent Setups

Altiok and Shiue (1994, 1995, 2000; also Altiok 1997, Chap. 7) analyze stochastic "multi-product pull-type production/inventory systems." These systems are equivalent to single-stage multi-product kanban systems with a shared manufacturing facility. The setup is changed only if a different product is scheduled for production, otherwise, the present setup is conserved. The products have different priorities. When the manufacturing facility is ready to start a new production run and two or more products satisfy the setup condition, generally the product with the highest priority is scheduled next. The setup condition is that a given number of active kanbans must be present for a product (threshold policy). If the threshold is set to one for all products, production may generally start as soon as one active kanban is present. The system then corresponds to the typical kanban system. Once the manufacturing facility is set up for a product, it keeps processing items of this product until all active kanbans for this product are cleared (*exhaustive processing*). At this point in time, the manufacturing facility is ready to start a new production run. This setup change protocol may be referred to as "threshold exhaustive processing with priorities."

No backorders. In Altiok and Shiue 1995, the authors assume phase-type processing times (in all given examples, however, the processing times are exponential), exponential setup times, mutually independent Poisson demand arrivals, and no backorders. The number of products is limited to three. The model is a finite-state CTMC (exact model) that suffers from state space explosion for larger systems. To circumvent the computational problems caused by the dimensionality problem, the authors develop a recursive scheme that significantly reduces the number of linear equations to be solved. The results seem to be unaffected by the recursive scheme so that they may be expected be exact for systems with the assumed properties and probability distributions.

Unlimited number of backorders. In Altiok and Shiue 1995 and 2000, the authors allow for arbitrarily distributed processing and setup times, and assume an unlimited number of backorders. Here, a product with a higher priority may not be scheduled more than once while a lower priority product is waiting to be produced. Demand arrivals for the different products, again, follow mutually independent Poisson processes. The authors develop a decomposition approach that approximates the performance of the system. In Altiok and Shiue 1995, the number of products is limited to three. This restriction is relaxed in Altiok and Shiue 2000, where examples are given for systems with five and eight products. In the later paper, the authors also consider systems with *threshold cyclic-exhaustive processing* and an unlimited number of backorders. They indicate that a slightly modified version of the decomposition approach for systems with threshold exhaustive processing with priorities may be employed to approximate performance measures of systems with threshold cyclic-exhaustive processing.

3.3.2 Multi-Stage Systems

Negligible setup times, type-1 material transfer, unlimited number of backorders. Askin, Mitwasi, and Goldberg (1993) consider multi-stage multi-product kanban systems with several multi-product manufacturing facilities in series or in parallel. Items of a product may be requested by one or several manufacturing facilities. A single type of kanban is used, and one inventory storage point exists between successive production stages (type-1 material transfer). Setup times are negligible and total setup costs are independent of the production sequence. Hence, products may be scheduled in any order without reducing production capacity or increasing operating costs. Requests for each type of (finished) product arrive with product-specific average rates, independently of the demand for the other products and of the number of backorders in the system. Each request must be satisfied. As a result, an infinite number of backorders may accumulate in the system. Production capacity of any manufacturing facility is such that any demand may be met eventually. The authors propose to analyze each manufacturing facility separately, with mutually independent exponential demand interarrival times for the different products and an infinite supply of input material. To obtain approximate values for the steady-state probabilities of the number of full containers and backorders for each product of each manufacturing facility, the authors solve a separate infinite-state CTMC for each product. In these single-product models, the fact that the modeled product competes in the original system with other products for processing time on the same manufacturing facility is incorporated approximately by the use of an aggregated state-dependent average processing rate.

4

New Models of Kanban Systems: A Construction-Kit Approach

In this chapter, we outline the construction-kit approach that is described in detail in the following chapters. This approach makes it possible to build solvable stochastic models of a large class of kanban systems by combining a small number of construction elements. The models may be used to approximate performance measures that characterize the average long-term behavior of single- and multi-stage single- and multi-product kanban systems (steady-state analysis). Apart from the opportunity to construct models of systems that could not be analyzed efficiently before, the suggested approach offers a general and extendable framework for developing stochastic models of a broad variety of kanban-controlled manufacturing systems.

4.1 General Assumptions

The following general assumptions describe the group of kanban systems for which solvable models may be constructed using the proposed approach. Here and in the following sections and chapters, r denotes the number of different products in the system and M denotes the number of stages.

- One type of kanban is used (*production kanbans*) and a single storage area exists between successive production stages. A container with input material is withdrawn from the output store of the preceding stage immediately before start of production (type-1 material transfer).
- A set of kanbans controls the production of one product in one stage.
- Demand arrivals (external demand) are mutually independent.
- Each demand is a request for one full container of one specific product.
- Demand interarrival times (external demand) for product i ($i = 1,\ldots,r$) are exponentially distributed with mean $1/\lambda_i^{\text{ext}}$ (this is equivalent to Poisson demand arrivals with rate λ_i^{ext}).
- Either no backorders accumulate in the system, or a limited positive number.
- A kanban is detached from a full container upon withdrawal of the container from the output store. The kanban is instantly placed on the scheduling board of the preceding stage. The time for removing the kanban and placing it on the scheduling board is negligible.
- A production stage contains one single- or multi-product manufacturing facility or several manufacturing facilities (single- and/or multi-product) in parallel. Each type of product may only be produced in one manufacturing facility in a stage.
- A manufacturing facility may only produce one item at a time.
- Multi-product manufacturing facilities require a setup change before items of a different product may be produced.
- A setup change is performed only if at least one active kanban and one container with input material are present for the next product of the setup sequence (*state-dependent setups*). Otherwise, this product is skipped.
- Setup times for product i in stage m are exponentially distributed with mean $s_i^{(m)}$.
- Multi-product manufacturing facilities in stage 1 work according to the setup change protocol *cyclic-exhaustive processing* (p. 5), multi-product manufacturing facilities in stages $2,\ldots,M$ work according to the setup change protocol *cyclic-exhaustive processing with limited input material* (p. 8).

- Container processing times for product i in stage m ($m = 1,\dots,M$) are exponentially distributed with mean $1/\mu_i^{(m)}$. The time to fill a container may include machine downtimes and time for (immediate) rework.
- The minimum production lot size of a product is the quantity recorded on the kanban (*container size*). Each production lot size is an integer multiple of the minimum production lot size.
- Transportation times between stages and between manufacturing facilities and output stores are negligible.
- Input material for stage 1 is always available.
- Each product is produced and demanded by one manufacturing facility only (neither merge- nor split-structures).
- Only a single (main) input is permitted for each processing operation (no assembly-structures; other inputs are admissible if they are always available).
- A container contains only good parts.
- No parts are scrapped in the process.

4.2 Construction Principle

The fundamental principle of the construction-kit approach is the notion of *decomposition*: systems for which exact models are too complex to be analyzed directly are broken up into manageable subsystems. The subsystems are calibrated to concertedly represent the essential characteristics of the original system. While decomposition, in principle, may result in exact models, decomposition models are typically approximate. This is because the interrelationships and interdependencies between the parts of the original system that are represented by the subsystems cannot be retained completely through the process of decomposition. Nevertheless, the approximations obtained with a decomposition model may still be sufficiently accurate for the intended purpose.

An example may illustrate why decomposition is necessary for the analysis of multi-product kanban systems (a similar example could be given for multi-stage kanban systems). The number of states of a continuous-time Markov chain that represents a single-stage multi-product kanban system with a shared manufacturing facility, exponential processing and setup times, mutually independent Poisson demand arrivals, no backorders, exhaustive processing, and state-dependent setups is

$$|\mathscr{S}| = r + \sum_{i=1}^{r} 2K_i \times \prod_{j=1;j\neq i}^{r} (1+K_j),$$

where r denotes the number of products and K_i (K_j) is the number of kanbans for product i (j) (Krieg and Kuhn 2004). If $K_i = K$ for all $i = 1,\ldots,r$, then

$$|\mathscr{S}| = r + r \times 2K \times (1+K)^{r-1}.$$

The number of states grows fast when the number of products and the number of kanbans increase. For example, for a system with five different products and five kanbans for each product, the number of states is 64 805. For a system with ten different products and ten kanbans each, the number of states is greater than 471 billion. In contrast to this, the model of a one-product subsystem—later referred to as component C2—is much smaller. For the two example systems, the number of states are 35 (vs. 64 805) and 120 (vs. 471×10^9). The number of states of the model of the one-product subsystem for any product i with K_i kanbans is

$$|\mathscr{S}_i| = 1 + 2K_i + (r-1)(1+K_i) = r \times (1+K_i) + K_i.$$

4.3 Construction Elements

The elements of the construction kit are either basic building blocks ("components") or modules consisting of several components of the same kind ("subassemblies"). Component C1 (Section 5.1) is the well-known model of the basic single-stage single-product kanban system with exponential demand interarrival and container processing times and no or a limited number of backorders. The other construction elements—components C2 and C3 and subassemblies SA1 and SA2—are new developments (key elements of component C2 and subassembly SA1 appear in less general form in Krieg and Kuhn 2004, essential features of component C3 and subassembly SA2 are discussed in a different format in Krieg and Kuhn 2002b).

The technique used for linking C1-components and the subassemblies (Section 7.1) has its roots in the work of Hillier and Boling (1967), who introduce a decomposition method for single-class finite-buffer open tandem queueing networks with exponential service times. Hillier and Boling's approach has later been augmented and extended by a number of authors (with and without direct reference), for example, Takahashi, Miyahara, and Hasegawa (1980), Boxma and Konheim (1981), Altiok (1982), Altiok and Perros (1986), Dallery and Frein (1993), and Buzacott, Liu, and Shanthikumar (1995). Reviews of contributions in this field are provided by Perros (1990) and Dallery and Gershwin (1992).

4.3.1 Components

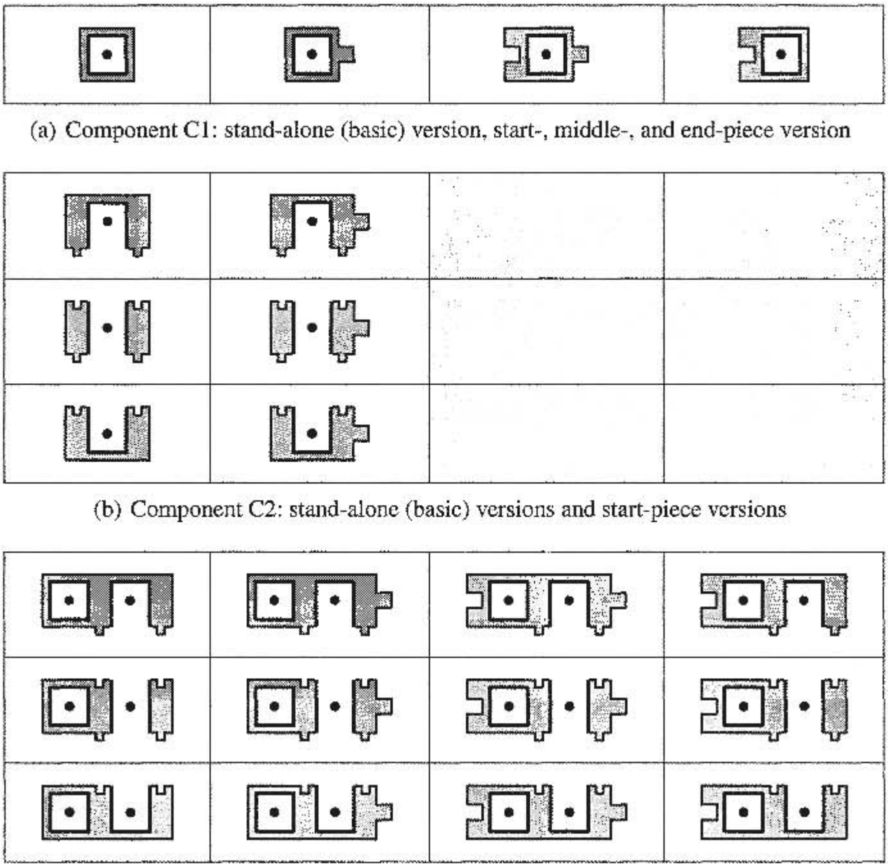
Three different components with up to four different variants are introduced in the next chapter. The basic version of each component is a stand-alone version. In contrast to the start-, middle-, and end-piece versions, the stand-alone version cannot be connected with other construction elements representing preceding or succeeding production stages.

From design, all components are finite-state Continuous-Time Markov Chains (CTMC). As a consequence, the exponential distribution is the default probability distribution for setup and processing times, and the Poisson distribution is the default for demand arrival processes. Note, however, that the method of phases (or stages) may be used to represent other distributions in a CTMC, either exactly or approximately (e.g., Cox 1955, Sauer and Chandy 1975, Bux and Herzog 1977, Altiok 1985, Neuts 1994, O'Cinneide 1999). Nevertheless, we restrict ourselves to models with exponentially distributed or Poisson-distributed random variables.

The first component (C1, Fig. 4.1a) is a model of a single-stage single-product kanban system satisfying the assumptions in Section 4.1. It may be used either as a start, a middle, or an end piece in building models of larger systems.

The second component (C2, Fig. 4.1b) has been developed to build a composite model of a single-stage multi-product kanban system in which all products are manufactured on a single shared manufacturing facility. Each C2-component contains the essential parts of the manufacturing facility with respect to one specific product. Component C2 is available in a stand-alone and in a start-piece version.

The third component (C3, Fig. 4.1c) is a combination of components C1 and C2. Components C3 may be used to construct a composite model of a two-stage kanban system where the input material of a multi-product manufacturing facility (stage 2) is processed on separate kanban-controlled single-product manufacturing facilities (stage 1). This component is needed because component C1 cannot directly precede component C2. The reason for this restriction is that information on the availability of full containers in the output stores of the single-product manufacturing facilities in the first stage is imperative for the setup change protocol of the multi-product facility in the second stage. Also, the arrival processes of orders for products of the first stage are mutually dependent in this system configuration: orders for first-stage



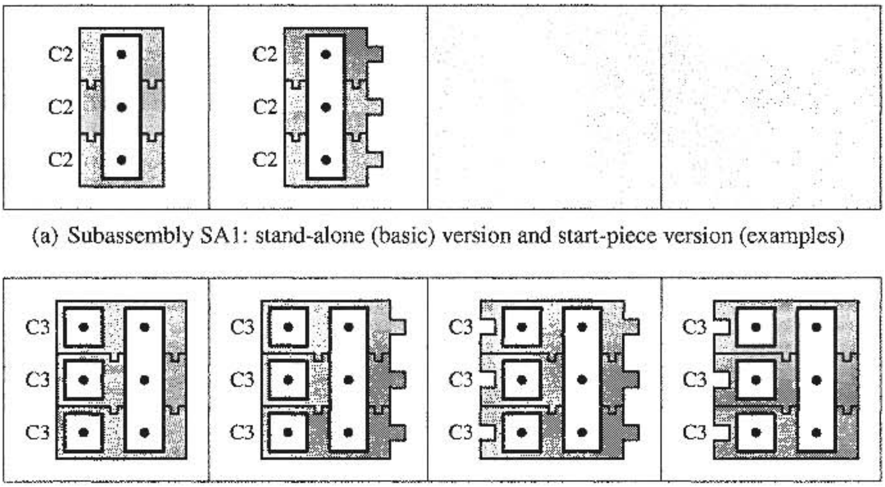
(a) Component C1: stand-alone (basic) version, start-, middle-, and end-piece version

(b) Component C2: stand-alone (basic) versions and start-piece versions

(c) Component C3: stand-alone (basic) versions, start-, middle-, and end-piece versions

Fig. 4.1. Components C1, C2, and C3

products may arrive only for one type of product at a time since the manufacturing facility of the second stage can only produce one product at a time. And—because of exhaustive processing—in some time intervals, several requests for the same type of product may arrive in close succession, while in other time intervals, no requests arrive for this product at all. In contrast to this, orders in a C1-component are assumed to arrive independently of the demand for other C1-components. Component C3 is available in all four versions (stand-alone, start-, middle-, and end-piece).



(a) Subassembly SA1: stand-alone (basic) version and start-piece version (examples)

(b) Subassembly SA2: stand-alone (basic) version, start-, middle-, and end-piece vers. (exx.)

Fig. 4.2. Subassemblies SA1 and SA2

4.3.2 Subassemblies

Subassemblies consist of two or more components of the same type. They are used as models of multi-product systems with one or two kanban-controlled manufacturing stages and as modules for models of larger systems (Fig. 4.2).

In Chapter 6, two subassemblies are described in detail. Subassembly SA1 is a model of a single-stage multi-product kanban system. It consists of two or more C2-components. Because of certain restricting requirements of the linking technique for stage-construction elements, subassembly SA1 may only be used as a start piece for larger models (Fig. 4.2a). This restriction does not apply to the second subassembly. Subassembly SA2 is a model of a two-stage multi-product system with several single-product facilities (stage 1) feeding one multi-product manufacturing facility (stage 2). Subassembly SA2 is the result of connecting two or more C3-components (Fig. 4.2b).

4.4 Composite Models

Composite models consist of two or more components or subassemblies, or a combination of both. By this definition, subassemblies are a subgroup of the group of composite models.

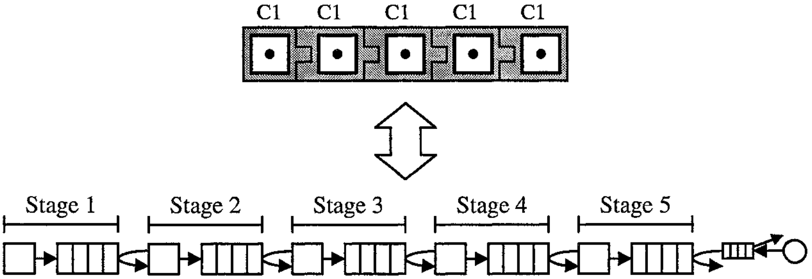


Fig. 4.3. Composite models, example 1; composite model and corresponding kanban system

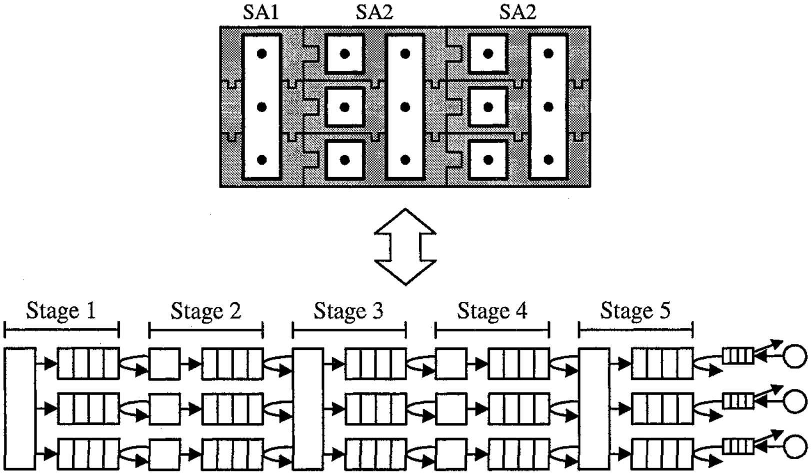


Fig. 4.4. Composite models, example 2; composite model and corresponding kanban system

Examples of composite models, along with the corresponding kanban systems, are given in Figures 4.3–4.6. In the representations of the kanban systems, a square symbolizes a single-product manufacturing facility, a rectangle represents a multi-product manufacturing facility. The multi-product manufacturing facilities may only produce items of one product at a time. The rectangular shape that consists of four medium-size rectangles stands for the storage area of one of the products in the output store of a stage. The rectangular shape that is made up of four small rectangles illustrates that the number of backorders for a product is limited.

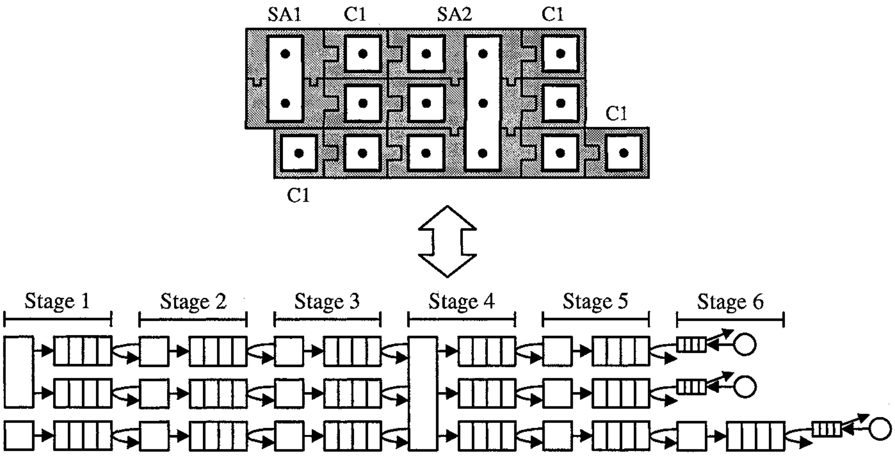


Fig. 4.5. Composite models, example 3; composite model and corresponding kanban system

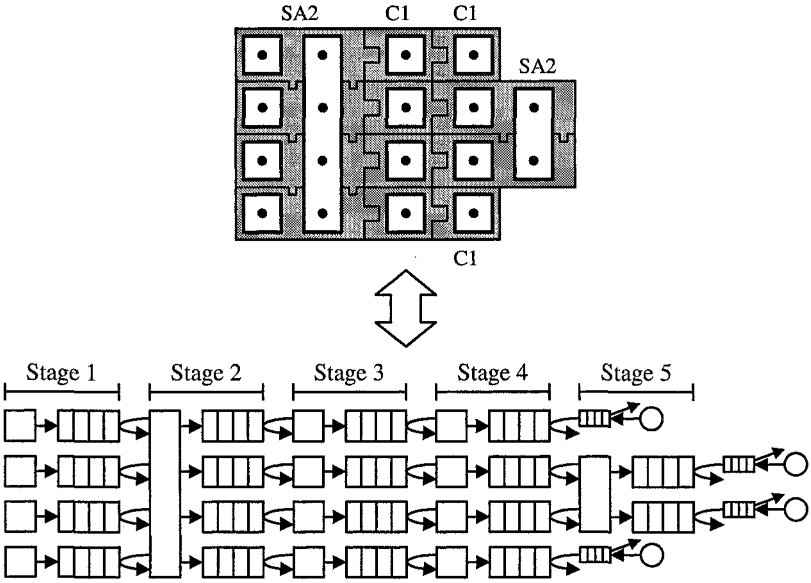


Fig. 4.6. Composite models, example 4; composite model and corresponding kanban system

4.5 Extended Application

Component C2 and subassembly SA1 are only available in the stand-alone and the start-piece version (the reasons are given in Section 5.2.3). Because of this, models of kanban systems with multi-product manufacturing facilities in series—such as the one depicted in Figure 4.7—cannot be constructed directly with the given components and subassemblies. The construction possibilities may, however, be extended by considering that a single-product manufacturing facility with processing times close to zero is almost nonexistent. With this insight, substitute models may be obtained with the help of subassembly SA2. For example, a substitute model of a kanban system consisting of M multi-product manufacturing facilities in series (Fig. 4.7) may be constructed with one SA1-subassembly and $M - 1$ SA2-subassemblies, provided that the processing times of the single-product manufacturing facilities are very small (Fig. 4.8). This approach is treated in detail in Chapter 8.

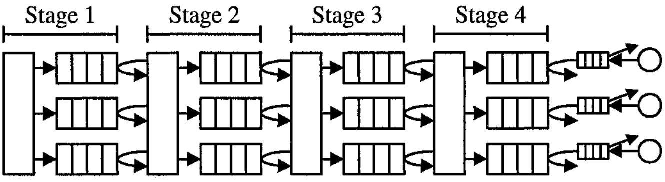


Fig. 4.7. Kanban system with multi-product manufacturing facilities in series

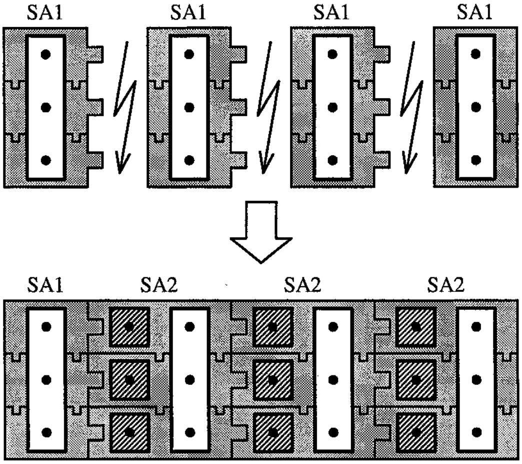


Fig. 4.8. Composite model of a four-stage kanban system with three products

5

Components: Basic Building Blocks

5.1 Component C1: Model of a Single-Product Manufacturing Facility

5.1.1 Stand-Alone Version (Basic Version)

The stand-alone version of component C1 is a model of a single-stage single-product kanban system with unlimited input material, exponentially distributed container

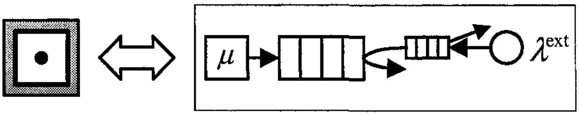


Fig. 5.1. Stand-alone (basic) version of component C1

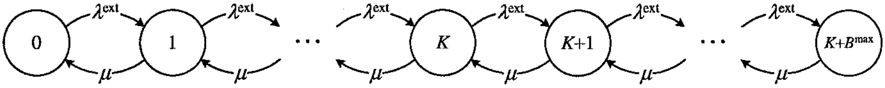


Fig. 5.2. State-transition rate diagram of the stand-alone version of component C1

processing times with rate μ, Poisson demand arrivals with rate λ^{ext}, and no or a limited number of backorders (Fig. 5.1). The state of the system is fully described by the number of active kanbans, the number of full containers in the output store, and the number of backorders. In fact, it is sufficient to know the number of backorders and either the number of active kanbans or the number of full containers because the number of active kanbans is always equal to the total number of kanbans minus the number of full containers since each full container must have a kanban attached to it.

Let $N(t)$ denote the number of active kanbans and backorders in the system at time t $(t \geq 0)$, and let K be the total number of kanbans and $B^{\max}$ the maximum number of backorders $(0 \leq B^{\max} < \infty)$. Then the number of backorders at time t is $\max\{N(t) - K, 0\}$ and the number of full containers at time t is $\max\{K - N(t), 0\}$. Whenever a customer arrives and the number of backorders is less than $B^{\max}$, the value of $N(t)$ increases by one since either a full container is removed from the output store and the accompanying kanban becomes active or, if the demand cannot be met from stock, a backorder is generated. Whenever a container is filled, the value of $N(t)$ decreases by one because either the number of backorders or the number of active kanbans, if the number of backorders is zero, is reduced by one. Hence, the stochastic process $\{N(t), t \geq 0\}$ is a finite birth-and-death process with birth rate λ^{ext}, death rate μ, and maximum population $K + B^{\max}$ (Fig. 5.2), and, if $\rho = \lambda^{\text{ext}}/\mu$,

$$p(n) = \begin{cases} (1-\rho)\rho^n / \left(1-\rho^{K+B^{\max}+1}\right), & \text{if } \rho \neq 1, \\ 1/(K+B^{\max}+1), & \text{if } \rho = 1, \end{cases}$$

if $p(n) = \lim_{t\to\infty} P\left[N(t) = n\right]$, $n = 0, \ldots, K + B^{\max}$, is the steady-state probability distribution of $\{N(t), t \geq 0\}$, that is, if $p(n)$ is the probability that at any point in

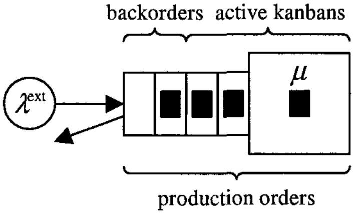


Fig. 5.3. Equivalent $M/M/1/N$ queueing system of a kanban system with $K = 3$ and $B^{\max} = 2$

time—assuming the system is in steady state—the sum of active kanbans and backorders is equal to n (e.g., Gross and Harris 1998, p. 77).

Note that the described single-stage single-product kanban system is equivalent to an $M/M/1/N$ queueing system with total system capacity (including the server position) $N = K + B^{\max}$ (Fig. 5.3). Active kanbans and backorders in the kanban system, that is, the current production orders, correspond to the customers in the queueing system. The first K customers, counted from the customer in service, correspond to active kanbans, the remaining customers correspond to backorders. This queueing analogy is described in similar form by Altiok (1997, Section 7.3.2).

5.1.2 Start-, Middle-, and End-Piece Versions

Start-piece version of component C1. In the start-piece version of component C1 (Fig. 5.4), the maximum number of backorders, $B^{\max,1}$, is fixed at one. A backorder is generated in stage 1 when the manufacturing facility of stage 2 needs new input material but finds the output store of stage 1 empty. The backorder situation lasts until the manufacturing facility in stage 1 transfers a full container to the output store. As we consider serial systems, that is, one supplier and one customer for each product within the system, there can never be more than one backorder present for a product except in stage M; the manufacturing facility of stage $m+1$ can generate a new request for the same input material only *after* the backorder situation has been resolved and the supplied material has been consumed.

Unlike the demand arrival process in the stand-alone version of component C1, the demand arrival process in the start-piece version is generally *not* Poisson. The demand arrival process is determined by the manufacturing process of stage 2: when the manufacturing facility in stage 2 intends to start processing items to fill a new container, it tries to withdraw a container with input material from stage 1. Only if the times between consecutive (intended) processing starts in stage 2 were exponentially distributed, then the demand arrival process in stage 1 would be Poisson. Since this

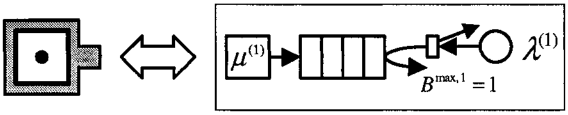


Fig. 5.4. Start-piece version of component C1

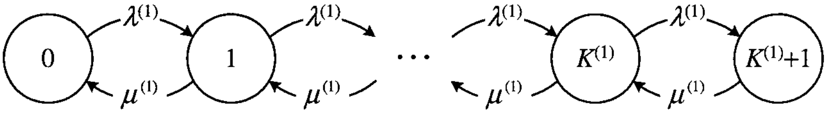


Fig. 5.5. State-transition rate diagram of the start-piece version of component C1

is not true in general, the demand arrival process in stage 1 is not Poisson. As a direct consequence, the stochastic process $\{N^{(1)}(t), t \geq 0\}$ is not a Markov process.

To be able to at least approximate performance measures of stage 1, we replace the true distribution of the demand arrival process with a Poisson distribution with rate $\lambda^{(1)}$. The resulting stochastic process, $\{\tilde{N}^{(1)}(t), t \geq 0\}$, is a Continuous-Time Markov Chain (CTMC) with rates $\lambda^{(1)}$ and $\mu^{(1)}$ and state space $\{1,\ldots,K^{(1)}+1\}$ (Fig. 5.5). Let $\tilde{p}^{(1)}(n) = \lim_{t\to\infty} P\left[\tilde{N}^{(1)}(t) = n\right]$, $n = 0,\ldots,K^{(1)}+1$, be the steady-state probability distribution of this stochastic process. Since $\{\tilde{N}^{(1)}(t), t \geq 0\}$ is a finite birth-and-death process with birth rate $\lambda^{(1)}$, death rate $\mu^{(1)}$, and maximum population $K^{(1)}+1$, we know that, if $\rho = \lambda^{(1)}/\mu^{(1)}$,

$$\tilde{p}^{(1)}(n) = \begin{cases} (1-\rho)\rho^n/\left(1-\rho^{K^{(1)}+2}\right), & \text{if } \rho \neq 1, \\ 1/(K^{(1)}+2), & \text{if } \rho = 1. \end{cases}$$

Middle-piece version of component C1. In the middle-piece version of component C1 (Fig. 5.6), the demand side is equivalent to the demand side in the start-piece version: the maximum number of backorders, $B^{\max,m}$ $(m = 2,\ldots,M-1)$, is fixed at one, and the demand arrival process is generally not Poisson. Unlike the manufacturing facility in stage 1, the manufacturing facility in stage m $(m = 2,\ldots,M-1)$ may find the output store of stage $m-1$ empty when it tries to withdraw a container with input material. The manufacturing facility in stage m is then forced to idle until new material arrives. The time the manufacturing facility spends waiting until a new container with input material becomes available may be viewed as part of an *effective* container processing time that spans the time interval from the instant the manufacturing facility could start a new container, if material was available, until the instant the container is filled completely.

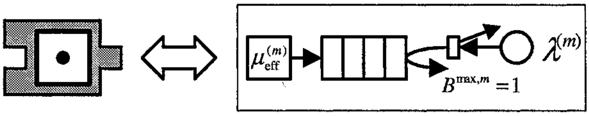


Fig. 5.6. Middle-piece version of component C1

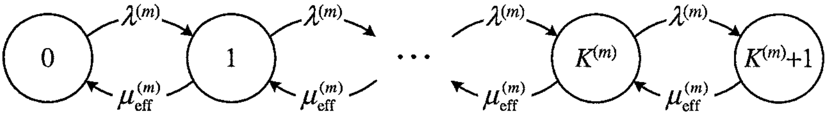


Fig. 5.7. State-transition rate diagram of the middle-piece version of component C1

Let $\mu_{\text{eff}}^{(m)}$ denote the *effective* average container processing rate in stage m, that is, the reciprocal of the *effective* average container processing time. If $w_{\text{IM}}^{(m)}$ is the average time the manufacturing facility has to wait each time it needs a new container with input material until such a container is available, then

$$\mu_{\text{eff}}^{(m)} = \left(w_{\text{IM}}^{(m)} + 1/\mu^{(m)}\right)^{-1}.$$

Note that the waiting time is zero when material is available in the output store (the time to withdraw a container is assumed to be negligible).

The *effective* processing times are generally not exponentially distributed (because of the waiting times). Hence, the stochastic process $\left\{N^{(m)}(t),\, t \geq 0\right\}$ is not a Markov process because (1) the demand arrival process is not Poisson and (2) the effective processing times are not exponential. To obtain at least estimates for the true performance measures of stages $2,\ldots,M-1$, we approximate $\left\{N^{(m)}(t),\, t \geq 0\right\}$ by the stochastic process $\left\{\tilde{N}^{(m)}(t),\, t \geq 0\right\}$ which is equal to the original process except that a Poisson distribution with parameter $\lambda^{(m)}$ replaces the true distribution of the demand arrival process and an exponential distribution with parameter $\mu_{\text{eff}}^{(m)}$ replaces the true distribution of the effective processing times. The stochastic process $\left\{\tilde{N}^{(m)}(t),\, t \geq 0\right\}$ is a CTMC on state space $\left\{1,\ldots,K^{(m)}+1\right\}$ (Fig. 5.7). Let $\tilde{p}^{(m)}(n) = \lim_{t\to\infty} P\left[\tilde{N}^{(m)}(t) = n\right]$, $n = 0,\ldots,K^{(m)}+1$, be the steady-state probability distribution of this stochastic process. Since $\left\{\tilde{N}^{(m)}(t), t \geq 0\right\}$ is a finite birth-and-death process with birth rate $\lambda^{(m)}$, death rate $\mu_{\text{eff}}^{(m)}$, and maximum population $K^{(m)}+1$, we know that, if $\rho = \lambda^{(m)}/\mu_{\text{eff}}^{(m)}$,

$$\tilde{p}^{(m)}(n) = \begin{cases} (1-\rho)\rho^n/\left(1-\rho^{K^{(m)}+2}\right), & \text{if } \rho \neq 1, \\ 1/\left(K^{(m)}+2\right), & \text{if } \rho = 1. \end{cases}$$

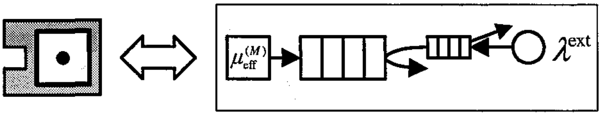


Fig. 5.8. End-piece version of component C1

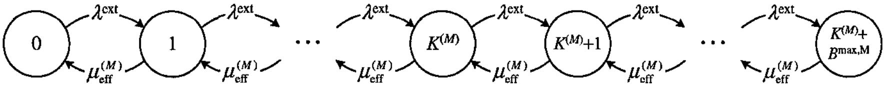


Fig. 5.9. State-transition rate diagram of the end-piece version of component C1

End-piece version of component C1. In the end-piece version of component C1 (Fig. 5.8), the demand arrival process is the external demand arrival process of the kanban system which means that it truly is a regular Poisson process. The manufacturing process is equivalent to the manufacturing process in the middle-piece version: the effective processing times are generally not exponential because of the waiting time for input material. Thus, the stochastic process $\{N^{(M)}(t),\, t \geq 0\}$ is also not a Markov process. To obtain approximations for the performance measures of the last stage, we introduce the stochastic process $\{\tilde{N}^{(M)}(t),\, t \geq 0\}$. This process is equivalent to $\{N^{(M)}(t), t \geq 0\}$ except that an exponential distribution with parameter $\mu_{\mathrm{eff}}^{(M)}$ replaces the true distribution of the effective processing times. The stochastic process $\{\tilde{N}^{(M)}(t),\, t \geq 0\}$ is a CTMC on state space $\{1,\ldots,K^{(M)}+B^{\max,M}\}$ (Fig. 5.9). Let $\tilde{p}^{(M)}(n) = \lim_{t\to\infty} P[\tilde{N}^{(M)}(t) = n]$, $n = 0,\ldots,K^{(M)}+B^{\max,M}$, be the steady-state probability distribution of this stochastic process. Since $\{\tilde{N}^{(M)}(t), t \geq 0\}$ is a finite birth-and-death process with birth rate λ^{ext}, death rate $\mu_{\mathrm{eff}}^{(M)}$, and maximum population $K^{(M)}+B^{\max,M}$, we know that, if $\rho = \lambda^{\mathrm{ext}}/\mu_{\mathrm{eff}}^{(M)}$,

$$\tilde{p}^{(M)}(n) = \begin{cases} (1-\rho)\rho^n / \left(1-\rho^{K^{(M)}+B^{\max,M}+1}\right), & \text{if } \rho \neq 1, \\ 1/\left(K^{(M)}+B^{\max,M}+1\right), & \text{if } \rho = 1. \end{cases}$$

5.1.3 Performance Measures

Performance measures of the stand-alone version of component C1 may be computed with the following equations. The results are exact. The same equations may be used to approximate the performance measures of the other three versions of component C1. For that, probability distribution p must be replaced with $\tilde{p}^{(1)}$, $\tilde{p}^{(m)}$, or $\tilde{p}^{(M)}$, and stage index (1), (m), or (M) must be added to all symbols. Moreover, for

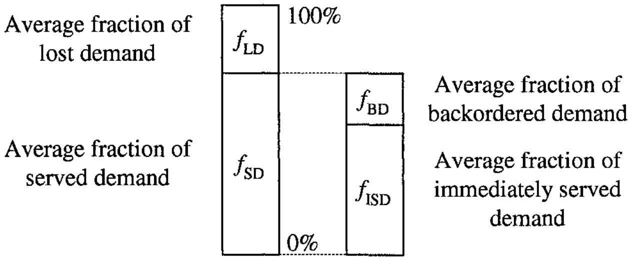

Average fraction of lost demand
f_{LD}
100%
f_{BD}
Average fraction of backordered demand
Average fraction of served demand
f_{SD}
Average fraction of immediately served demand
f_{ISD}
0%

Fig. 5.10. Relationship of different fractions of demand

the start- and the middle-piece version, tag "ext" must be removed from λ^{ext}. For the middle- and the end-piece version, tag "eff" must be attached to μ.

Average fraction of lost demand. The average fraction of lost demand is equal to the probability that all kanbans are active and that the number of backorders is equal to the maximum number of backorders,

$$f_{\text{LD}} = p\left(K + B^{\max}\right).$$

Average fraction of served demand. The average fraction of served demand (with and without waiting) is equal to the probability that an arriving customer can be served immediately or that his demand can be backordered, that is, the probability that either not all kanbans are active or that the number of backorders is not equal to the maximum number of backorders,

$$f_{\text{SD}} = \sum_{n=0}^{K+B^{\max}-1} p(n). \tag{5.1}$$

Of course, the average fraction of served demand must be equal to one (or 100%) minus the average fraction of lost demand (Fig. 5.10),

$$f_{\text{SD}} = 1 - f_{\text{LD}}.$$

Average fraction of immediately served demand (*average fill rate*). The average fraction of immediately served demand, or the *average fill rate*, is equal to the probability that the output store is not empty (1 – stockout probability), which implies that at least one kanban is inactive,

$$f_{\text{ISD}} = f = \sum_{n=0}^{K-1} p(n) = 1 - \sum_{n=K}^{K+B^{\max}} p(n). \tag{5.2}$$

This performance measure is also known as *off-the-shelf service*, *type-1 service*, and *β-service level* (e.g., Tempelmeier 2000).

Average fraction of backordered demand. The average fraction of backordered demand is equal to the probability that an arriving demand has to wait before it is met because no container with the requested items is present in the output store,

$$\begin{aligned} f_{\text{BD}} &= \sum_{n=K}^{K+B^{\max}-1} p(n) \\ &= f_{\text{SD}} - f_{\text{ISD}}. \end{aligned}$$

Average production rate. The average production rate (*average throughput*) is the average number of processed containers per unit of time which is given by the probability that the manufacturing facility is busy (at least one active kanban is present in the system) and the average container processing rate,

$$\text{TH} = \left[\sum_{n=1}^{K+B^{\max}} p(n)\right]\mu = \left[1 - p(0)\right]\mu.$$

Alternatively, the average throughput is equal to the average arrival rate of demand that is served immediately upon arrival or after a stochastic waiting time, λ_{SD},

$$\text{TH} = \lambda_{\text{SD}} = f_{\text{SD}}\lambda^{\text{ext}}. \tag{5.3}$$

The average arrival rate of served demand is the conceptional analogue to the *effective average arrival rate* in a standard queueing system with a limited waiting room.

Average inventory level. The average number of full containers in the output store is

$$\bar{y} = \sum_{n=0}^{K-1} (K-n)p(n). \tag{5.4}$$

Average backorder level. The average number of backorders is

$$\bar{b} = \sum_{n=K+1}^{K+B^{\max}} (n-K)p(n).$$

Average waiting time of backordered demand. The average waiting time of backordered demand, that is, the average time a demand that cannot be filled from stock upon arrival has to wait until being met, may be obtained by applying Little's law (Little 1961),

$$w_{\mathrm{BD}} = \frac{\bar{b}}{\lambda_{\mathrm{BD}}},$$

where λ_{BD} is the average arrival rate of demand that is backordered,

$$\lambda_{\mathrm{BD}} = f_{\mathrm{BD}} \lambda^{\mathrm{ext}}.$$

Average waiting time of served demand. The average waiting time of served demand, that is, the average delay in meeting any served demand (including the demand met immediately upon arrival), is

$$w_{\mathrm{SD}} = \frac{\bar{b}}{\lambda_{\mathrm{SD}}}. \qquad (5.5)$$

This equation follows from averaging the average waiting time of backordered demand and the average waiting time of immediately served demand (which is zero) with the appropriate weights,

$$w_{\mathrm{SD}} = \frac{\lambda_{\mathrm{BD}}}{\lambda_{\mathrm{SD}}} \frac{\bar{b}}{\lambda_{\mathrm{BD}}} + \frac{\lambda_{\mathrm{ISD}}}{\lambda_{\mathrm{SD}}} 0 = \frac{\bar{b}}{\lambda_{\mathrm{SD}}}.$$

5.2 Component C2: One-Product Submodel of a Multi-Product Manufacturing Facility

5.2.1 Stand-Alone Version (Basic Version)

The stand-alone version of component C2 contains the relevant parts of a single-stage multi-product kanban system as described in Chapter 2 with respect to one product,

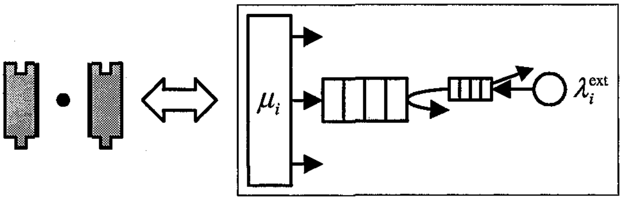
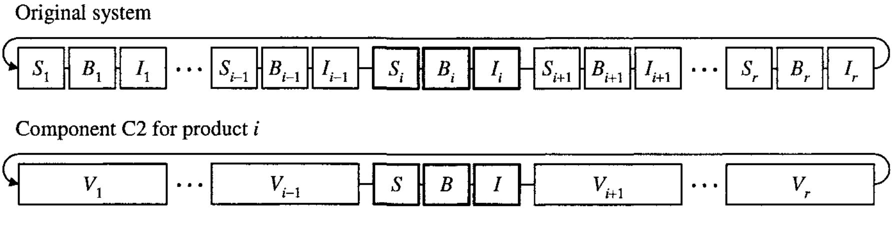


Fig. 5.11. Stand-alone (basic) version of component C2 (product i)

say product i (Fig. 5.11). This component is a building block for the approximate evaluation of such a multi-product system.

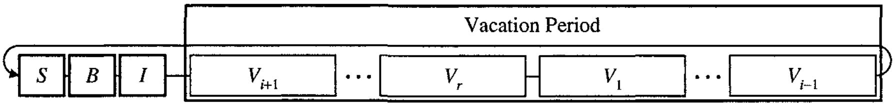
In the assumed multi-product system, the manufacturing facility cannot always be available for product i because products $1,\ldots,i-1,i+1,\ldots,r$ also demand processing time on the manufacturing facility. According to the chosen setup change protocol (cyclic-exhaustive processing, p. 5), the manufacturing facility is being set up for product i if at least one kanban (for product i) is active when the manufacturing facility turns its attention to product i. The manufacturing facility then keeps processing items of this product until all empty containers have been filled, that is, until the number of active kanbans for product i is zero. If at this point no kanban is active for any product, then the manufacturing facility idles. It may instantly resume processing product-i items if the next kanban that is activated is a product-i kanban. Otherwise, that is, if the next kanban that is activated is a product-j kanban $(j \neq i)$, the manufacturing facility is being set up for product j. Should there be active kanbans for other products at the end of a busy period for product i, then the manufacturing facility is immediately being set up for the next product—according to the predetermined fixed setup sequence—for which there is at least one active kanban. Thus, for product i, the relevant aspects of the system are not only the number of active kanbans and backorders for product i, as in component C1, but also the state of the manufacturing facility: only if the manufacturing facility is already set up for product i, then processing may start immediately as authorized by active product-i kanbans. Otherwise, that is, if the manufacturing facility is set up for a different product, product i has to wait its turn.

We say that the manufacturing facility is "on vacation" when it is not available for product i. A single vacation period lasts from the instant when a setup change for a product other than product i begins—either after the manufacturing facility has stopped processing items of product i and maybe spent some time idling or after the last time a setup change for product i was considered but not executed because no kanbans for product i were active at that time—up until the instant when the next setup change for product i may be considered. If each of the r products has one position in the setup cycle, then a vacation period consists of $r-1$ phases (a single vacation phase embraces the time periods of a cycle when the manufacturing facility is being set up for, processing items of, and being idle after processing items of a particular product). Note that the length of a vacation phase may be zero in a particular vacation period. This occurs when setup and production of the respective product were skipped because no active kanbans were present for this product when



S – Setup, B – Busy period, I – Idle period (for product i); V_j – Vacation phase for product j

Fig. 5.12. States of the manufacturing facility in the original system and in component C2 for product i (setup cycle $1,2,\ldots,r$)



S – Setup, B – Busy period, I – Idle period (for product i); V_j – Vacation phase for product j

Fig. 5.13. States of the manufacturing facility in component C2 for product i (alternative perspective)

the setup change for this product was considered. Also, the manufacturing facility may be found to start a new vacation period immediately after returning from the last vacation. This happens when production of product i is skipped because no kanbans for product i were active at the end of the last vacation period (*multiple vacations*).

In summary, the relevant states of the manufacturing facility in component C2 for product i are Setup (S), Busy period (B), and Idle period (I) for product i and a vacation period consisting of one Vacation phase (V_j) for each product j, $j = 1,\ldots,r$; $j \neq i$ (Fig. 5.12). A different perspective on the sequence of states of the manufacturing facility in component C2 for product i (Fig. 5.13) suggests that an $M/M/1/N$ queueing system with setups, idle periods, and multiple vacations is an appropriate model of component C2. In the following section, we describe this model in detail.

5.2.2 Approximate Model of the Stand-Alone Version

Stochastic processes. We first define three stochastic processes.

Let $N_i(t)$ denote the number of active kanbans and backorders in component C2 for product i at time t $(i = 1,\ldots,r; t \geq 0)$. Then $\left\{N_i(t), t \geq 0\right\}$ is a stochastic process over state space $\mathscr{N}_i = \left\{0,\ldots,K_i + B_i^{\max}\right\}$, where K_i is the number of kanbans for product i and $B_i^{\max}$ is the maximum number of backorders for product i.

Let $Z_i(t)$ denote the state of the manufacturing facility in component C2 for product i at time t. Then $\{Z_i(t), t \geq 0\}$ is a stochastic process over state space $\mathscr{Z}_i = \{S; B; I; V_j, j = 1, \ldots, r; j \neq i\}$.

We finally define the combined stochastic process $\{[N_i(t), Z_i(t)], t \geq 0\}$. Unfortunately, this is not a Markov process because the length of a vacation phase is generally not exponentially distributed. To make it a Markov process we substitute an exponential distribution with parameter $t_{\mathrm{SBI}}^{(j)}$ for the unknown true distribution of the length of vacation phase V_j, $j = 1, \ldots, r; j \neq i$. Parameter $t_{\mathrm{SBI}}^{(j)}$ is the average amount of time between two successive vacation periods in the approximate model of component C2 for product j. Then the combined stochastic process $\{[\tilde{N}_i(t), \tilde{Z}_i(t)], t \geq 0\}$ is a Continuous-Time Markov Chain (CTMC) on state space $\mathscr{S}_i = \{[(n,S),(n,B)], n \in \mathscr{N}_i \setminus \{0\}; (0,I); (n,V_j), n \in \mathscr{N}_i\ j = 1, \ldots, r; j \neq i\}$ (Fig. 5.14).

Transition $(1,B) \rightarrow (0,I)$ with transition rate μ_i' in the state-transition rate diagram of the CTMC for component C2 for product i (Fig. 5.14) represents that the manufacturing facility switches to idle at the end of a busy period. This happens if no kanban is active for any product. Otherwise, the manufacturing facility is set up for one of the other products: transition $(1,B) \rightarrow (0,V_{i+1})$ with rate μ_i''. Idle period I_i ends when a kanban is activated for any product $i = 1, \ldots, r$; either a container with product-i items is withdrawn—transition $(0,I) \rightarrow (1,B)$ with rate λ_i^{ext}—or a container is withdrawn with items of one of the other products: transition $(0,I) \rightarrow (0,V_{i+1})$ with rate $\sum_{j=1;j\neq i}^{(r)} \lambda_j^{\mathrm{ext}}$.

Steady-state probability distributions. We define steady-state probability distributions for the stochastic processes $\{\tilde{N}_i(t), t \geq 0\}$ and $\{\tilde{Z}_i(t), t \geq 0\}$ and for the combined stochastic process $\{[\tilde{N}_i(t), \tilde{Z}_i(t)], t \geq 0\}$. Table 5.1 contains a summary of the steady-state probability distributions, Figure 5.15 illustrates the hierarchical relations between them.

Let $\tilde{p}_i(n) = \lim_{t\rightarrow\infty} P[\tilde{N}_i(t) = n]$, $n \in \mathscr{N}_i$, be the steady-state probability distribution of $\{\tilde{N}_i(t)\}$.

Let $\tilde{g}_i(z) = \lim_{t\rightarrow\infty} P[\tilde{Z}_i(t) = z]$, $z \in \mathscr{Z}_i$, be the steady-state probability distribution of $\{\tilde{Z}_i(t), t \geq 0\}$.

Let $\tilde{q}_i(n,z) = \lim_{t\rightarrow\infty} P[\tilde{N}_i(t) = n, \tilde{Z}_i(t) = z]$, $(n,z) \in \mathscr{S}_i$, be the steady-state probability distribution of $\{[\tilde{N}_i(t), \tilde{Z}_i(t)], t \geq 0\}$.

Distribution $\tilde{q}_i$ may be determined by solving the balance equations for the rate of probability inflows and outflows of each state of the CTMC.

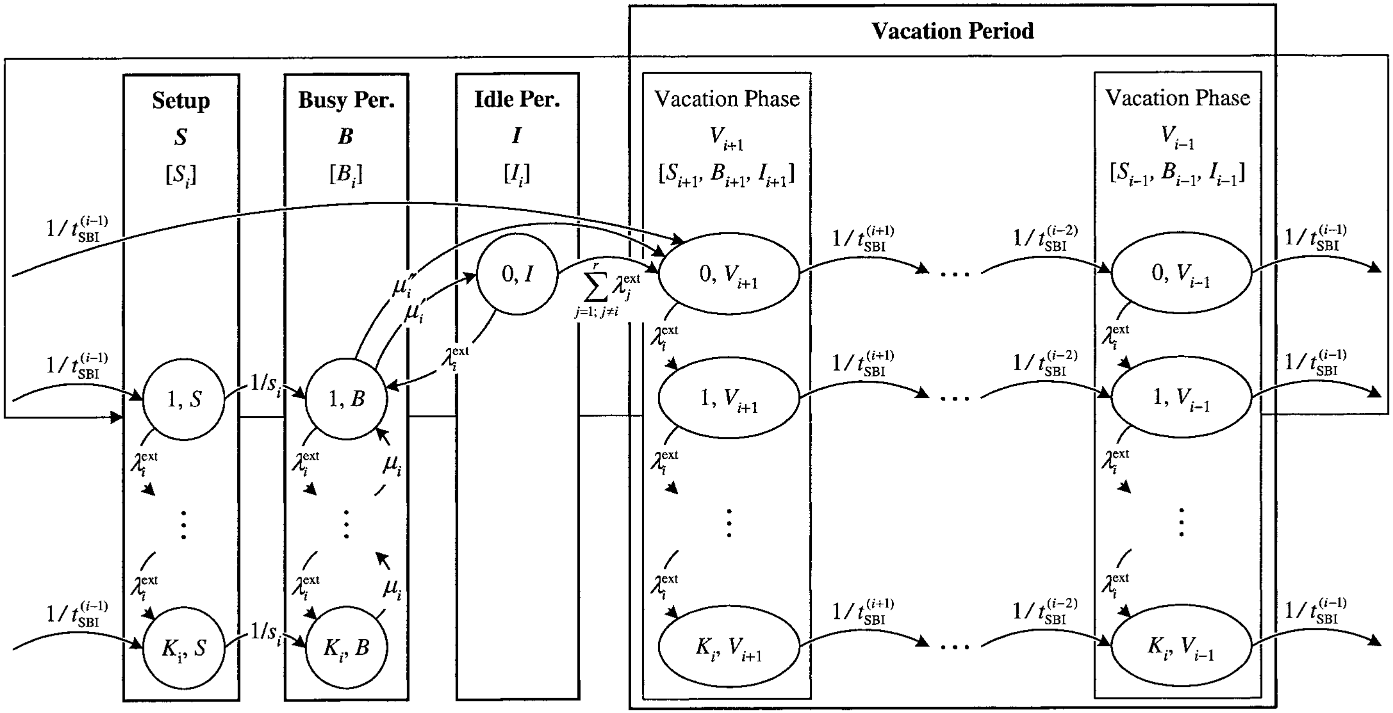


Fig. 5.14. State-transition rate diagram of the CTMC for the stand-alone version of component C2 (product i); if $B_i^{\max} > 0$, then substitute $K_i + B_i^{\max}$ for K_i.

Table 5.1. Stochastic Processes and Steady-State Probability Distributions of the Stand-Alone Version of Component C2 (Product i)

Stochastic Process	Prob. Distrib.
$\{\tilde{N}_i(t), t \geq 0\}$	$\tilde{p}_i(n)$
$\{\tilde{Z}_i(t), t \geq 0\}$	$\tilde{g}_i(z)$
$\{[\tilde{N}_i(t), \tilde{Z}_i(t)], t \geq 0\}$	$\tilde{q}_i(n,z)$

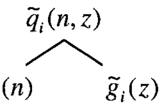


Fig. 5.15. Hierarchy of the steady-state probability distributions of the stand-alone version of component C2 (product i)

Based on distribution $\tilde{q}_i$, we can obtain distribution $\tilde{p}_i$,

$$\tilde{p}_i(n) = \begin{cases} \sum_{z \in \mathscr{Z}_i \setminus \{S,B\}} \tilde{q}_i(n,z), & \text{if } n = 0, \\ \sum_{z \in \mathscr{Z}_i \setminus \{I\}} \tilde{q}_i(n,z), & \text{if } n \in \mathscr{N}_i \setminus \{0\}, \end{cases} \tag{5.6}$$

and distribution $\tilde{g}_i$,

$$\tilde{g}_i(z) = \begin{cases} \sum_{n \in \mathscr{N}_i \setminus \{0\}} \tilde{q}_i(n,z), & \text{if } z = S, B; \\ \tilde{q}_i(0,z), & \text{if } z = I, \\ \sum_{n \in \mathscr{N}_i} \tilde{q}_i(n,z), & \text{if } z = V_j, j = 1, \ldots, r; j \neq i. \end{cases} \tag{5.7}$$

5.2.3 Start-Piece Version

The start-piece version of component C2 (Fig. 5.16) shares most properties of the start-piece version of component C1: the maximum number of backorders, $B^{\max,1}$, is fixed at one, and the demand arrival process is generally not Poisson. Again, we replace the true distribution of the demand arrival process with a Poisson distribution with rate $\lambda_i^{(1)}$.

Middle- and end-piece versions of component C2 are not available, mainly, because information on the availability of containers with input material in the output store(s) of the preceding stage is needed, yet not available, in a middle- or end-piece version of component C2. Also, the demand arrival process that a middle- or end-piece version of component C2 would generate for the preceding stage would

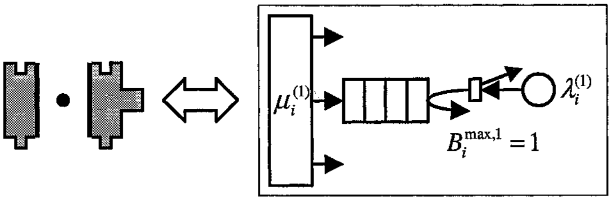


Fig. 5.16. Start-piece version of component C2 (product *i*)

not posses a constant average rate over time because consumption of input material, which leads to the generation of demand for the preceding stage, is interrupted in component C2 during setup and vacation periods since production only takes place during busy periods.

5.2.4 Performance Measures

Steady-state performance measures of component C2 for product *i* (stand-alone or start-piece version) may be approximated based on probability distribution $\tilde{p}_i$. For the start-piece version, stage index (1) must be added to all symbols, and tag "ext" must be removed from λ_i^{ext}.

Average fraction of lost demand. An estimate for the average fraction of lost demand for product *i* is

$$\hat{f}_{\text{LD},i} = \tilde{p}_i\left(K_i + B_i^{\max}\right).$$

Average fraction of served demand. An estimate for the average fraction of served demand (with and without waiting) for product *i* is

$$\hat{f}_{\text{SD},i} = \sum_{n=0}^{K_i+B_i^{\max}-1} \tilde{p}_i(n) = 1 - \hat{f}_{\text{LD},i}. \tag{5.8}$$

Average fraction of immediately served demand (*average fill rate*). An estimate for the average fraction of immediately served demand (*average fill rate*) for product *i* is

$$\hat{f}_{\text{ISD},i} = \hat{f}_i = 1 - \sum_{n=K_i}^{K_i+B_i^{\max}} \tilde{p}_i(n). \tag{5.9}$$

Average fraction of backordered demand. An estimate for the average fraction of backordered demand for product i is

$$\begin{aligned}\hat{f}_{\mathrm{BD},i} &= \sum_{n=K_i}^{K_i+B_i^{\max}-1} \tilde{p}_i(n) \\ &= \hat{f}_{\mathrm{SD},i} - \hat{f}_{\mathrm{ISD},i}.\end{aligned}$$

Average production rate. An estimate for the average production rate (*average throughput*) with regard to product i is

$$\widehat{\mathrm{TH}}_i = \left[1 - \tilde{p}_i(0)\right]\mu_i.$$

Alternatively,

$$\widehat{\mathrm{TH}}_i = \hat{\lambda}_{\mathrm{SD},i} = \hat{f}_{\mathrm{SD},i}\lambda_i^{\mathrm{ext}}, \tag{5.10}$$

where $\hat{\lambda}_{\mathrm{SD},i}$ is an estimate for the average arrival rate of demand for product i that is served immediately upon arrival or after a stochastic waiting time.

Average inventory level. An estimate for the average number of full containers with product-i items in the output store is

$$\hat{y}_i = \sum_{n=0}^{K_i-1} \left(K_i - n\right)\tilde{p}_i(n). \tag{5.11}$$

Average backorder level. An estimate for the average number of backorders for product i is

$$\hat{b}_i = \sum_{n=K_i+1}^{K_i+B_i^{\max}} \left(n - K_i\right)\tilde{p}_i(n).$$

Average waiting time of backordered demand. An estimate for the average waiting time of backordered demand for product i is

$$\hat{w}_{\mathrm{BD},i} = \frac{\hat{b}_i}{\hat{\lambda}_{\mathrm{BD},i}},$$

where $\hat{\lambda}_{\mathrm{BD},i}$ is an estimate for the average arrival rate of demand for product i that is backordered,

$$\hat{\lambda}_{\mathrm{BD},i} = \hat{f}_{\mathrm{BD},i}\lambda_i^{\mathrm{ext}}.$$

Average waiting time of served demand. An estimate for the average waiting time of served demand for product *i* is

$$\hat{w}_{\mathrm{SD},i} = \frac{\hat{\bar{b}}_i}{\hat{\lambda}_{\mathrm{SD},i}}. \tag{5.12}$$

5.3 Component C3: One-Product Submodel of a Multi-Product Manufacturing Facility Fed by Single-Product Facilities

5.3.1 Stand-Alone Version (Basic Version)

The stand-alone version of component C3 is a direct extension of component C2. In addition to the aspects considered in component C2, the manufacturing process of the input material for product *i* is included in component C3 (Fig. 5.17). Component C3 may therefore represent the relevant parts of a special two-stage multi-product kanban system with respect to one product, say product *i*. In this two-stage kanban system, raw material is processed in product-specific single-product manufacturing facilities (stage 1) and the resulting intermediate products are processed further in a single multi-product manufacturing facility (stage 2). The chosen setup change protocol for the multi-product manufacturing facility is cyclic-exhaustive processing with limited input material (p. 8), and the predetermined fixed setup cycle is $1,2,\ldots,r$.

The main justification for component C3 is the possibility to explicitly consider the availability, or lack, of input material for a multi-product manufacturing facility. Shortage of material is an important aspect in multi-stage kanban systems (Section 2.3): even if there are active kanbans for a product, if material is not available in the output store of the preceding stage, then the manufacturing facility should not be set up for this product. Moreover, a production run may have to be terminated when there are still active kanbans for the product on the scheduling board because input material cannot be obtained from the preceding stage.

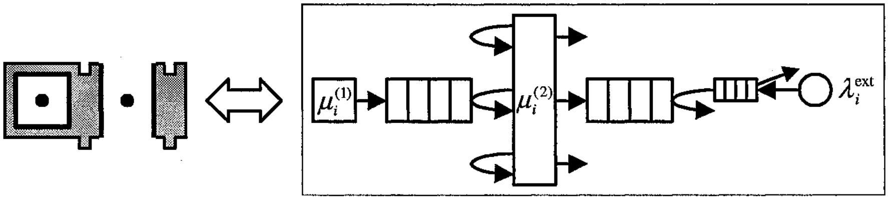


Fig. 5.17. Stand-alone (basic) version of component C3 (product *i*)

In contrast to components C1 and C2, *three* pieces of information about the original system are included in component C3 for product i: (1) the number of active kanbans and backorders for product i in stage 2, (2) the number of containers with input material in the output store of stage 1, and (3) the state of the manufacturing facility in stage 2. In the following section, we describe an approximate model of component C3.

5.3.2 Approximate Model of the Stand-Alone Version

Stochastic processes. We first define three elementary stochastic processes. To simplify the notation, we use n instead of $n^{(2)}$ to denote the number of active kanbans and backorders in stage 2, y instead of $y^{(1)}$ to denote the number of full containers in the output store of stage 1, and z instead of $z^{(2)}$ to denote the state of the manufacturing facility in stage 2.

Let $N_i(t)$ denote the number of active kanbans and backorders in stage 2 in component C3 for product i at time t $(i = 1,\dots,r; t \geq 0)$. Then $\{N_i(t), t \geq 0\}$ is a stochastic process over state space $\mathscr{N}_i = \{0,\dots,K_i^{(2)} + B_i^{\max,2}\}$, where $K_i^{(2)}$ is the number of kanbans for product i in stage 2 and $B_i^{\max,2}$ is the maximum number of backorders for product i in stage 2.

Let $Y_i(t)$ denote the number of full containers in the output store of stage 1 in component C3 for product i at time t. Then $\{Y_i(t), t \geq 0\}$ is a stochastic process over state space $\mathscr{Y}_i = \{0,\dots,K_i^{(1)}\}$, where $K_i^{(1)}$ is the number of kanbans and, thus, the maximum number of full containers for product i in stage 1.

Let $Z_i(t)$ denote the state of the manufacturing facility in stage 2 in component C3 for product i at time t. Again, we abbreviate the possible states of the manufacturing facility by S (setup), B (busy period), I (idle period), and V_j, $j = 1,\dots,r; j \neq i$ (vacation phase for product j). Then $\{Z_i(t), t \geq 0\}$ is a stochastic process over state space $\mathscr{Z}_i = \{S; B; I; V_j, j = 1,\dots,r; j \neq i\}$.

We finally define the combined stochastic process $\{[N_i(t), Y_i(t), Z_i(t)], t \geq 0\}$. Unfortunately, this is not a Markov process because the length of a vacation phase is generally not exponentially distributed. To make it a Markov process we substitute an exponential distribution with parameter $t_{\mathrm{SBI}}^{(j)}$ for the unknown true distribution of the length of vacation phase V_j, $j = 1,\dots,r; j \neq i$. Then the combined stochastic process $\{[\tilde{N}_i(t), \tilde{Y}_i(t), \tilde{Z}_i(t)], t \geq 0\}$ is a Continuous-Time Markov Chain (CTMC) on state space $\mathscr{S}_i = \{(n,y,S),\ n \in \mathscr{N}_i \setminus \{0\},\ y \in \mathscr{Y}_i \setminus \{0\};\ (n,y,B),\ n \in \mathscr{N}_i \setminus \{0\},\ y \in \mathscr{Y}_i;\ (0,y,I),\ y \in \mathscr{Y}_i;\ (n,0,I),\ n \in \mathscr{N}_i \setminus \{0\};\ (n,y,V_j),\ n \in \mathscr{N}_i,\ y \in \mathscr{Y}_i,\ j = 1,\dots,r;\ j \neq i\}$.

Figures 5.18–5.21 show the main part of a generalized state-transition rate diagram of this CTMC. To simplify the notation in the figures, we substituted Y_i for $K_i^{(1)}$, the number of kanbans for product i in stage 1, and K_i for $K_i^{(2)} + B_i^{\max,2}$, the number of kanbans plus the maximum number of backorders for product i in stage 2. Symbol Y_i may be read as the maximum number of full containers with items of product i in the output store of stage 1.

Transition rates μ_i' and μ_i'' have the same meaning as in the CTMC for component C2, except that lack of input material may also force the manufacturing facility (in stage 2) to idle (μ_i') or start another vacation period (μ_i''). Transition rate Λ_i is similar in spirit and function to transition rate $\sum_{j=1;j\neq i}^{(r)} \lambda_j^{\text{ext}}$ in component C2. The reciprocal of Λ_i is the average time until the first product other than product i meets the setup criterion (at least one active kanban and one container with input material) after the manufacturing facility stopped processing items of product i.

Steady-state probability distributions. We now define steady-state probability distributions for the stochastic processes $\{\tilde{N}_i(t), t \geq 0\}$, $\{\tilde{Y}_i(t), t \geq 0\}$, and $\{\tilde{Z}_i(t), t \geq 0\}$, and for several combined stochastic processes. Table 5.2 contains a summary of the steady-state probability distributions, Figure 5.22 illustrates the hierarchical relations between them.

Let $\tilde{p}_i(n) = \lim_{t\to\infty} P\big[\tilde{N}_i(t) = n\big]$, $n \in \mathcal{N}_i$, be the steady-state probability distribution of $\{\tilde{N}_i(t), t \geq 0\}$.

Let $\tilde{k}_i(y) = \lim_{t\to\infty} P\big[\tilde{Y}_i(t) = y\big]$, $y \in \mathcal{Y}_i$, be the steady-state probability distribution of $\{\tilde{Y}_i(t), t \geq 0\}$.

Let $\tilde{g}_i(z) = \lim_{t\to\infty} P\big[\tilde{Z}_i(t) = z\big]$, $z \in \mathcal{Z}_i$, be the steady-state probability distribution of $\{\tilde{Z}_i(t), t \geq 0\}$.

Let $\tilde{q}_i(n,z) = \lim_{t\to\infty} P\big[\tilde{N}_i(t) = n, \tilde{Z}_i(t) = z\big]$, $(n,z) \in \big\{\big[(n,S),(n,B)\big], n \in \mathcal{N}_i \setminus \{0\};\ \big[(n,I),(n,V_j)\big], n \in \mathcal{N}_i, j = 1,\ldots,r;\ j \neq i\big\}$, be the steady-state probability distribution of $\big\{\big[\tilde{N}_i(t), \tilde{Z}_i(t)\big], t \geq 0\big\}$.

Let $\tilde{h}_i(y,z) = \lim_{t\to\infty} P\big[\tilde{Y}_i(t) = y, \tilde{Z}_i(t) = z\big]$, $(y,z) \in \big\{(y,S),\ y \in \mathcal{Y}_i \setminus \{0\};\ \big[(y,B),(y,I),(y,V_j)\big], y \in \mathcal{Y}_i, j = 1,\ldots,r;\ j \neq i\big\}$, be the steady-state probability distribution of $\big\{\big[\tilde{Y}_i(t), \tilde{Z}_i(t)\big], t \geq 0\big\}$.

Let $\tilde{o}_i(n,y,z) = \lim_{t\to\infty} P\big[\tilde{N}_i(t) = n, \tilde{Y}_i(t) = y, \tilde{Z}_i(t) = z\big]$, $(n,y,z) \in \mathcal{S}_i$, be the steady-state probability distribution of $\big\{\big[\tilde{N}_i(t), \tilde{Y}_i(t), \tilde{Z}_i(t)\big], t \geq 0\big\}$.

Distribution $\tilde{o}_i$ may be determined by solving the balance equations for the rate of probability inflows and outflows of each state of the CTMC. With distribution $\tilde{o}_i$,

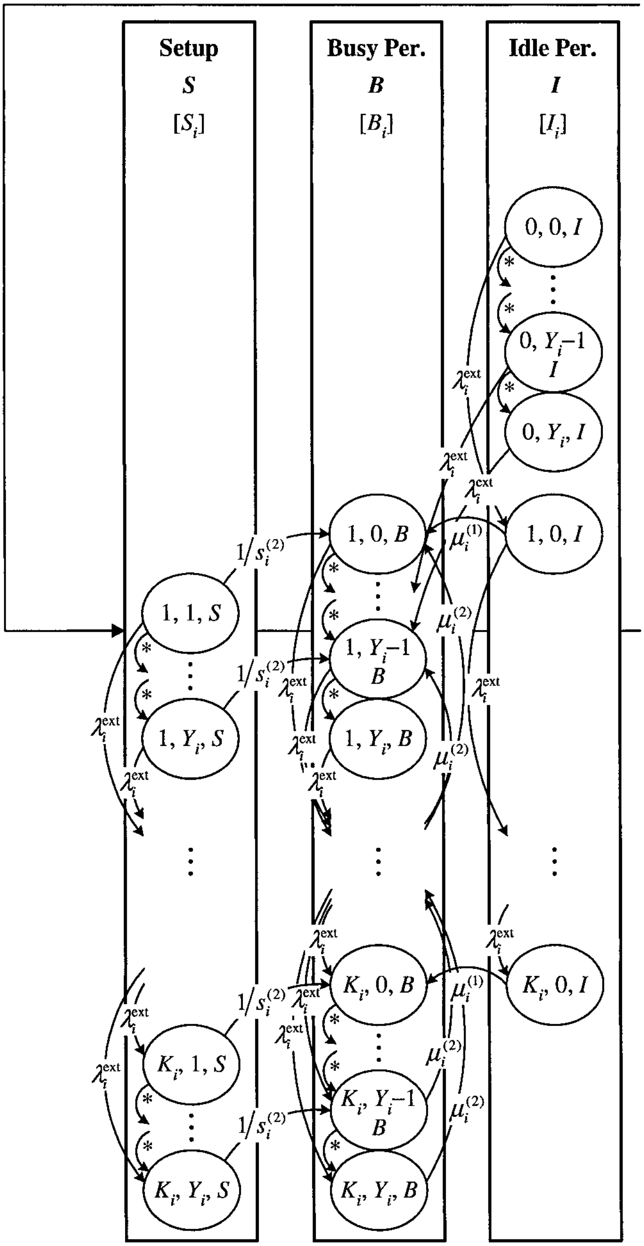


Fig. 5.18. State-transition rate diagram (layer 1, left part) of the CTMC for the stand-alone version of component C3 (product i); $* = \mu_i^{(1)}$, $K_i \equiv K_i^{(2)} + B_i^{\max,2}$, $Y_i \equiv K_i^{(1)}$.

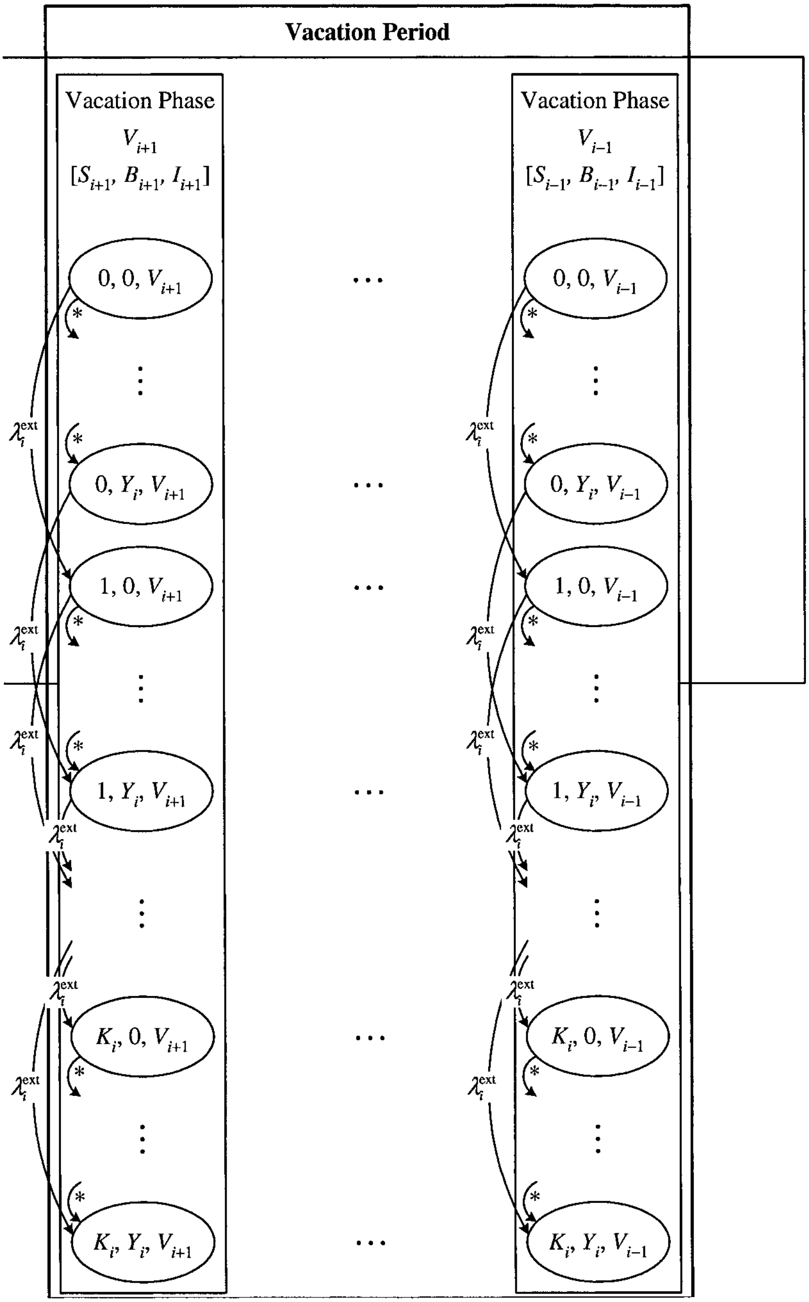


Fig. 5.19. State-transition rate diagram (layer 1, right part) of the CTMC for the stand-alone version of component C3 (product i); $* = \mu_i^{(1)}$, $K_i \equiv K_i^{(2)} + B_i^{\max,2}$, $Y_i \equiv K_i^{(1)}$.

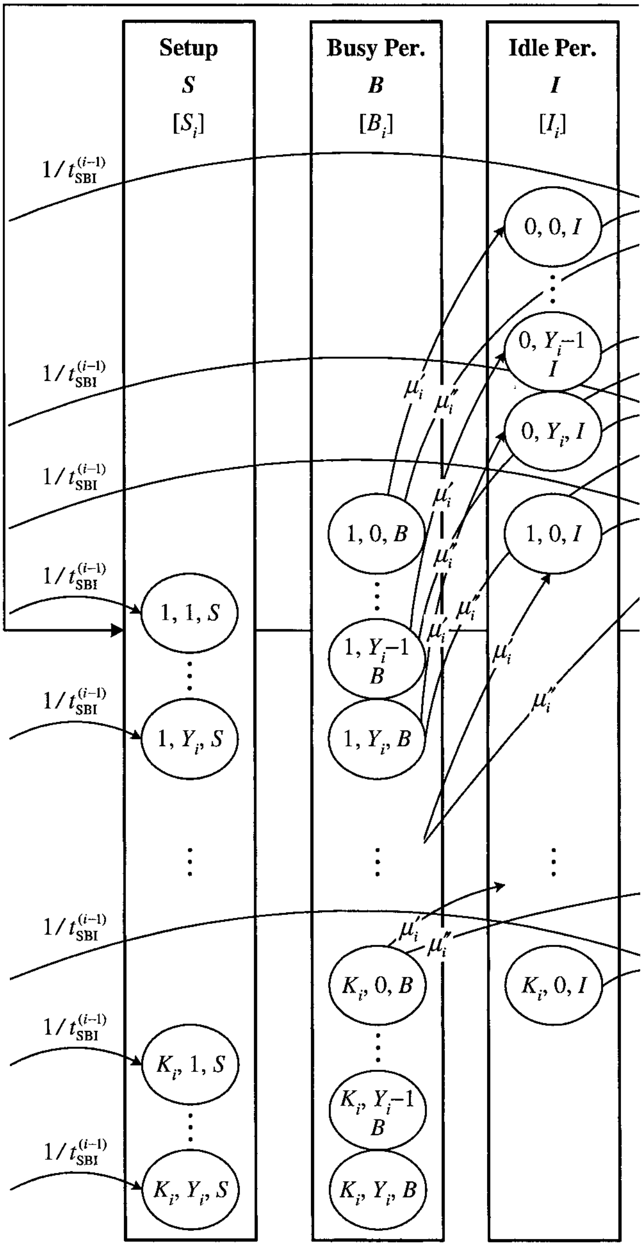


Fig. 5.20. State-transition rate diagram (layer 2, left part) of the CTMC for the stand-alone version of component C3 (product i); $K_i \equiv K_i^{(2)} + B_i^{\max,2}$, $Y_i \equiv K_i^{(1)}$.

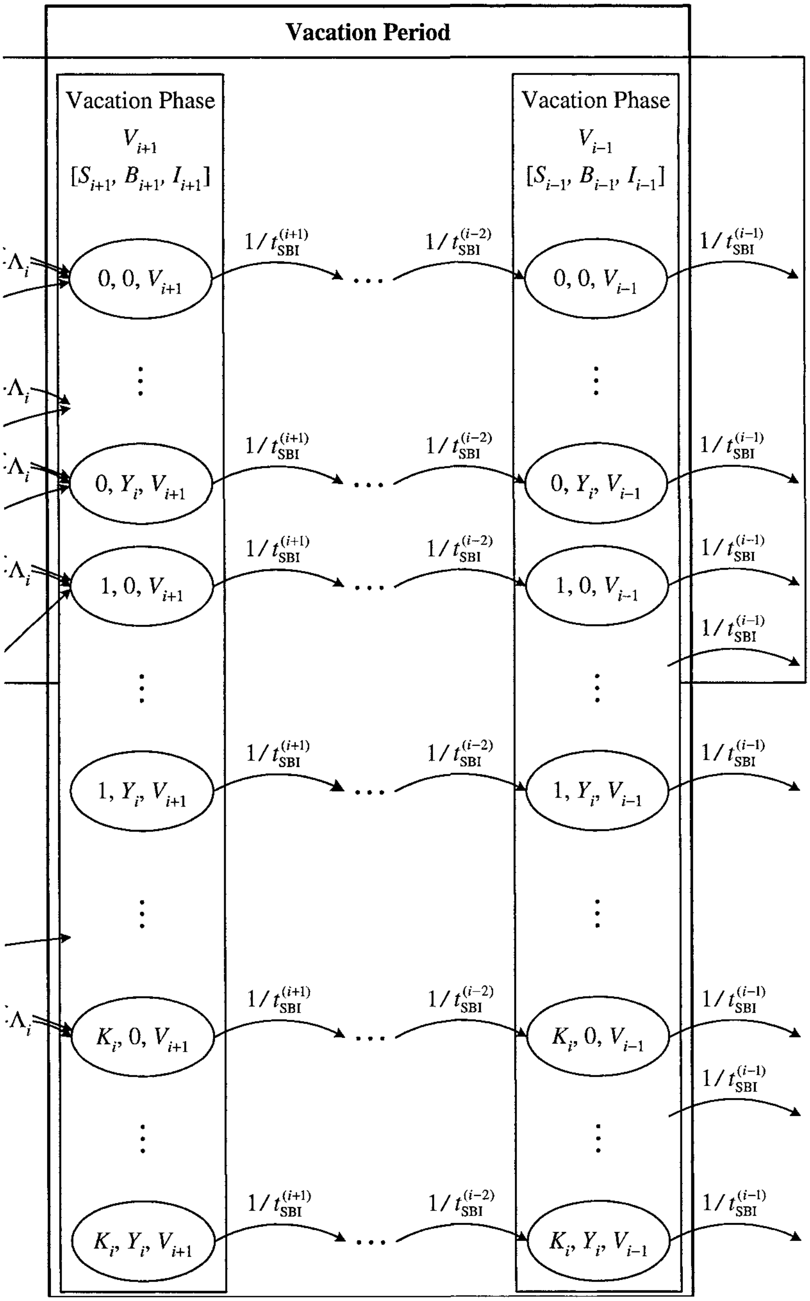


Fig. 5.21. State-transition rate diagram (layer 2, right part) of the CTMC for the stand-alone version of component C3 (product i); $K_i \equiv K_i^{(2)} + B_i^{\max,2}$, $Y_i \equiv K_i^{(1)}$.

Table 5.2. Stochastic Processes and Steady-State Probability Distributions of the Stand-Alone Version of Component C3 (Product i)

Stochastic Process	Prob. Distrib.
$\{\tilde{N}_i(t), t \geq 0\}$	$\tilde{p}_i(n)$
$\{\tilde{Y}_i(t), t \geq 0\}$	$\tilde{k}_i(y)$
$\{\tilde{Z}_i(t), t \geq 0\}$	$\tilde{g}_i(z)$
$\{[\tilde{N}_i(t), \tilde{Z}_i(t)], t \geq 0\}$	$\tilde{q}_i(n,z)$
$\{[\tilde{Y}_i(t), \tilde{Z}_i(t)], t \geq 0\}$	$\tilde{h}_i(y,z)$
$\{[\tilde{N}_i(t), \tilde{Y}_i(t), \tilde{Z}_i(t)], t \geq 0\}$	$\tilde{o}_i(n,y,z)$

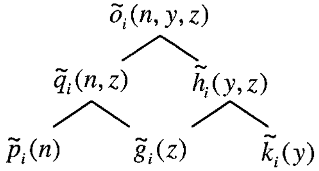


Fig. 5.22. Hierarchy of the steady-state probability distributions of the stand-alone version of component C3 (product i)

we can obtain distribution $\tilde{q}_i$,

$$\tilde{q}_i(n,z) = \begin{cases} \sum_{y \in \mathscr{Y}_i \setminus \{0\}} \tilde{o}_i(n,y,z), & \text{if } n \geq 1, z = S, \\ \tilde{o}_i(n,0,z), & \text{if } n \geq 1, z = I, \\ \sum_{y \in \mathscr{Y}_i} \tilde{o}_i(n,y,z), & \text{if } n \geq 1, z = B \text{ or} \\ & \text{if } n = 0, z = I \text{ or} \\ & \text{if } n \geq 0, z = V_j, j = 1, \ldots, r; j \neq i, \end{cases} \tag{5.13}$$

and distribution $\tilde{h}_i$,

$$\tilde{h}_i(y,z) = \begin{cases} \sum_{n \in \mathscr{N}_i \setminus \{0\}} \tilde{o}_i(n,y,z), & \text{if } y \geq 1, z = S \text{ or} \\ & \text{if } y \geq 0, z = B, \\ \tilde{o}_i(0,y,z), & \text{if } y \geq 1, z = I, \\ \sum_{n \in \mathscr{N}_i} \tilde{o}_i(n,y,z), & \text{if } y = 0, z = I \text{ or} \\ & \text{if } y \geq 0, z = V_j, j = 1, \ldots, r; j \neq i. \end{cases} \tag{5.14}$$

From distribution $\tilde{q}_i$, we can derive distributions $\tilde{p}_i$ and $\tilde{g}_i$,

$$\tilde{p}_i(n) = \begin{cases} \sum_{z\in\mathscr{Z}_i\setminus\{S,B\}} \tilde{q}_i(n,z), & \text{if } n = 0, \\ \sum_{z\in\mathscr{Z}_i} \tilde{q}_i(n,z), & \text{if } n = 1,\dots,K_i^{(2)} + B_i^{\max,2}, \end{cases} \tag{5.15}$$

$$\tilde{g}_i(z) = \begin{cases} \sum_{n\in\mathscr{N}_i\setminus\{0\}} \tilde{q}_i(n,z), & \text{if } z = S,B, \\ \sum_{n\in\mathscr{N}_i} \tilde{q}_i(n,z), & \text{if } z = I; V_j, j = 1,\dots,r; j \neq i. \end{cases} \tag{5.16}$$

Based on distribution $\tilde{h}_i$, we can determine distribution $\tilde{k}_i$,

$$\tilde{k}_i(y) = \begin{cases} \sum_{z\in\mathscr{Z}_i\setminus\{S\}} \tilde{h}_i(y,z), & \text{if } y = 0, \\ \sum_{z\in\mathscr{Z}_i} \tilde{h}_i(y,z), & \text{if } y = 1,\dots,K_i^{(1)}. \end{cases} \tag{5.17}$$

5.3.3 Start-, Middle-, and End-Piece Versions

The start-, middle-, and end-piece versions of component C3 (Figs. 5.23–5.25) share most properties of the equivalent versions of component C1: the maximum number of backorders ($B_i^{\max,m}$, $m = 2,4,\dots,M-2$) is fixed at one in start- and middle-piece versions of component C3, the demand arrival processes are generally not Poisson in start- and middle-piece versions of component C3, and the effective processing times are generally not exponentially distributed in middle- and end-piece versions of component C3. Again, we replace the true distribution of the demand arrival process in the start- and middle-piece versions with a Poisson distribution with parameter $\lambda_i^{(m)}$ ($m = 2,4,\dots,M-2$), and we substitute an exponential distribution with parameter $\mu_{\text{eff},i}^{(m-1)}$ ($m = 3,5,\dots,M-1$) for the true distribution of the effective processing times in the middle- and end-piece versions.

5.3.4 Performance Measures

Steady-state performance measures of component C3 for product i (stand-alone, start-, middle-, or end-piece version) may be approximated based on probability distributions $\tilde{p}_i$ and $\tilde{k}_i$. For the start-piece version, tag "ext" must be removed from λ_i^{ext}. For the middle-piece version, stage indices (1) and (2) must be replaced with $(m-1)$ and (m), and tag "ext" must be removed from λ_i^{ext}. For the end-piece version, stage indices (1) and (2) must be replaced with $(M-1)$ and (M).

Average fraction of lost demand. An estimate for the average fraction of lost demand for product i (stage 2) is

$$\hat{f}_{\text{LD},i}^{(2)} = \tilde{p}_i\left(K_i^{(2)} + B_i^{\max,2}\right).$$

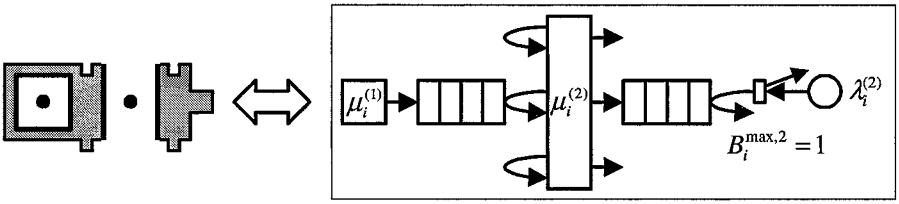


Fig. 5.23. Start-piece version of component C3

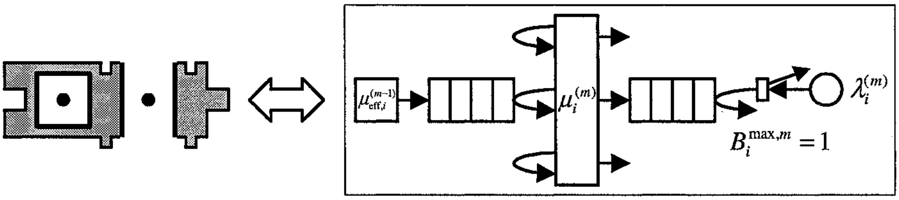


Fig. 5.24. Middle-piece version of component C3

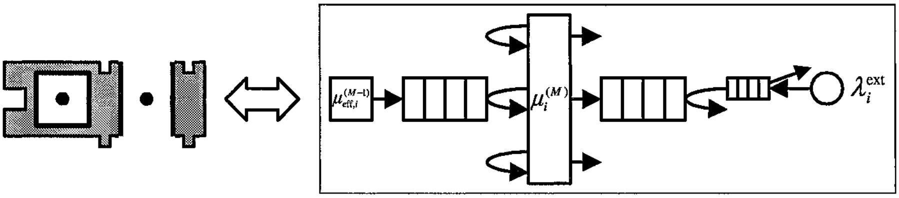


Fig. 5.25. End-piece version of component C3

Average fraction of served demand. An estimate for the average fraction of served demand (with and without waiting) for product i (stage 2) is

$$\hat{f}^{(2)}_{\mathrm{SD},i} = \sum_{n=0}^{K_i^{(2)}+B_i^{\max,2}-1} \tilde{p}_i(n) = 1 - \hat{f}^{(2)}_{\mathrm{LD},i}. \tag{5.18}$$

Average fraction of immediately served demand (*average fill rate*). An estimate for the average fraction of immediately served demand (*average fill rate*) for product i (stage 2) is

$$\hat{f}^{(2)}_{\mathrm{ISD},i} = \hat{f}^{(2)}_i = 1 - \sum_{n=K_i^{(2)}}^{K_i^{(2)}+B_i^{\max,2}} \tilde{p}_i(n). \tag{5.19}$$

Average fraction of backordered demand. An estimate for the average fraction of backordered demand for product i (stage 2) is

$$\begin{aligned}\hat{f}^{(2)}_{\mathrm{BD},i} &= \sum_{n=K_i^{(2)}}^{K_i^{(2)}+B_i^{\max,2}-1} \tilde{p}_i(n) \\ &= \hat{f}^{(2)}_{\mathrm{SD},i} - \hat{f}^{(2)}_{\mathrm{ISD},i}.\end{aligned}$$

Average production rate. An estimate for the average production rate (*average throughput*) with regard to product i is

$$\widehat{\mathrm{TH}}_i = \left[1 - \tilde{p}_i(0)\right] \mu_i^{(2)}.$$

Alternatively,

$$\widehat{\mathrm{TH}}_i = \hat{\lambda}^{(2)}_{\mathrm{SD},i} = \hat{f}^{(2)}_{\mathrm{SD},i} \lambda_i^{\mathrm{ext}}, \tag{5.20}$$

where $\hat{\lambda}^{(2)}_{\mathrm{SD},i}$ is an estimate for the average arrival rate of demand for product i in stage 2 that is served immediately upon arrival or after a stochastic waiting time.

Average inventory level in stage 1. An estimate for the average number of full containers with product-i items in the output store of stage 1 is

$$\hat{y}_i^{(1)} = \sum_{y=1}^{K_i^{(1)}} y \tilde{k}_i(y). \tag{5.21}$$

Average inventory level in stage 2. An estimate for the average number of full containers with product-i items in the output store of stage 2 is

$$\hat{y}_i^{(2)} = \sum_{n=0}^{K_i^{(2)}-1} \left(K_i^{(2)} - n\right) \tilde{p}_i(n). \tag{5.22}$$

Average backorder level. An estimate for the average number of backorders for product i (stage 2) is

$$\hat{b}_i^{(2)} = \sum_{n=K_i^{(2)}+1}^{K_i^{(2)}+B_i^{\max,2}} \left(n - K_i^{(2)}\right) \tilde{p}_i(n).$$

Average waiting time of backordered demand. An estimate for the average waiting time of backordered demand for product i (stage 2) is

$$\hat{w}^{(2)}_{\mathrm{BD},i} = \frac{\hat{b}^{(2)}_i}{\hat{\lambda}^{(2)}_{\mathrm{BD},i}},$$

where $\hat{\lambda}^{(2)}_{\mathrm{BD},i}$ is an estimate for the average arrival rate of demand for product i that is backordered in stage 2,

$$\hat{\lambda}^{(2)}_{\mathrm{BD},i} = \hat{f}^{(2)}_{\mathrm{BD},i} \lambda^{\mathrm{ext}}_i.$$

Average waiting time of served demand. An estimate for the average waiting time of served demand for product i (stage 2) is

$$\hat{w}^{(2)}_{\mathrm{SD},i} = \frac{\hat{b}^{(2)}_i}{\hat{\lambda}^{(2)}_{\mathrm{SD},i}}. \tag{5.23}$$

6

Subassemblies: Models of Multi-Product Manufacturing Facilities

In this chapter, we describe how multiple C2-components (subassembly SA1) and multiple C3-components (subassembly SA2) may be linked to obtain models of kanban-controlled multi-product manufacturing facilities. Subassembly SA2 is identical to subassembly SA1, except that the multi-product manufacturing facility is fed by several kanban-controlled single-product manufacturing facilities.

6.1 Subassembly SA1: Model of a Multi-Product Manufacturing Facility

Subassembly SA1 is the result of linking two or more C2-components (Fig. 6.1). The process of linking C2-components requires the determination of parameters $t_{\mathrm{SBI}}^{(i)}$, μ_i',

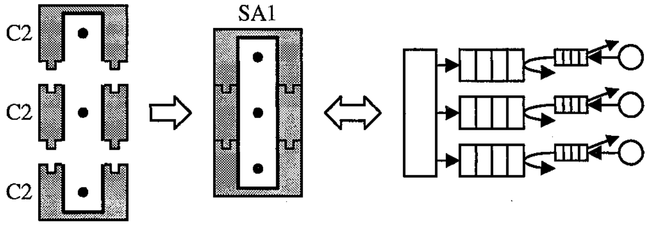


Fig. 6.1. C2-components, subassembly SA1, and corresponding kanban system

and μ_i'' for all $i = 1,\ldots,r$. An iterative procedure is employed to arrive at sufficiently precise estimates for the linking parameters.

6.1.1 Equation for Parameter $t_{\mathrm{SBI}}^{(i)}$

Let $t_S^{(i)}$ $\left(t_B^{(i)}, t_I^{(i)}\right)$ be the average amount of time the manufacturing facility in the model of component C2 for product i spends in state S (B, I) between two vacation periods (Fig. 6.2). Also, let $t_{\mathrm{SBI}}^{(i)}$ be the average amount of time between the end of a vacation period until the start of the next vacation period in the model of component C2 for product i. Since the manufacturing facility may only be in states S, B, or I between two successive vacation periods, it follows that $t_{\mathrm{SBI}}^{(i)} = t_S^{(i)} + t_B^{(i)} + t_I^{(i)}$.

Let T_i denote the average amount of time from the end of a vacation period until the end of the next vacation period in component C2 for product i. This value may also be referred to as average cycle length. If $t_V^{(i)}$ denotes the average length of a vacation period in the model of component C2 for product i, with

$$t_V^{(i)} = \sum_{j=1;j\neq i}^{r} t_{\mathrm{SBI}}^{(j)}, \tag{6.1}$$

then the average cycle length T_i is equal to $t_{\mathrm{SBI}}^{(i)} + t_V^{(i)}$.

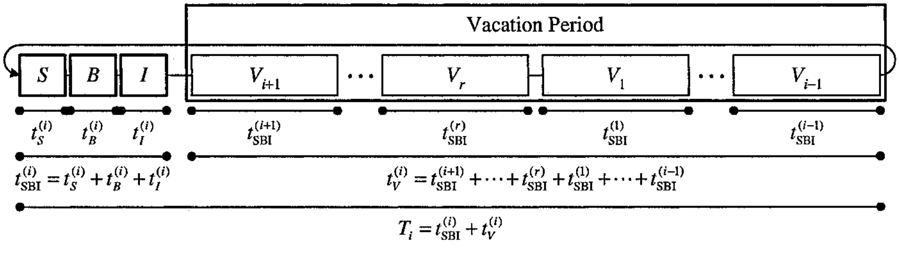

Vacation Period
S
B
I
V_{i+1}
V_r
V_1
V_{i-1}
$t_S^{(i)}$
$t_B^{(i)}$
$t_I^{(i)}$
$t_{\mathrm{SBI}}^{(i+1)}$
$t_{\mathrm{SBI}}^{(r)}$
$t_{\mathrm{SBI}}^{(1)}$
$t_{\mathrm{SBI}}^{(i-1)}$
$t_{\mathrm{SBI}}^{(i)} = t_S^{(i)} + t_B^{(i)} + t_I^{(i)}$
$t_V^{(i)} = t_{\mathrm{SBI}}^{(i+1)} + \cdots + t_{\mathrm{SBI}}^{(r)} + t_{\mathrm{SBI}}^{(1)} + \cdots + t_{\mathrm{SBI}}^{(i-1)}$
$T_i = t_{\mathrm{SBI}}^{(i)} + t_V^{(i)}$

Fig. 6.2. States of the manufacturing facility in component C2 for product *i*

From distribution $\tilde{g}_i$, we can obtain the average fraction of time of a vacation period in the model of component C2 for product i: $\tilde{g}_V^{(i)} = \sum_{j=1;j\neq i}^{r} \tilde{g}_i(V_j)$. Since $\tilde{g}_V^{(i)} T_i = t_V^{(i)}$, we have

$$T_i = \frac{t_V^{(i)}}{\tilde{g}_V^{(i)}}. \tag{6.2}$$

Finally, we can determine $t_{\mathrm{SBI}}^{(i)}$,

$$t_{\mathrm{SBI}}^{(i)} = T_i - t_V^{(i)}. \tag{6.3}$$

Rough estimates. At the beginning of the algorithm, we need rough estimates for $t_S^{(i)}$ and $t_B^{(i)}$ for all $i = 1,\dots,r$ and for $t_{\mathrm{SBI}}^{(i)}$ for all $i = 1,\dots,r$; $i \neq 2$. A rough estimate for $t_{\mathrm{SBI}}^{(2)}$ is not required because the algorithm starts with the analysis of the CTMC for component C2 for product 2 and this CTMC does not contain transition rate $t_{\mathrm{SBI}}^{(2)}$. Simplifying greatly, we pretend that the sequence of states of the manufacturing facility in the original system is S_1, B_1, S_2, B_2, …, S_r, B_r (repeated), which implies that at least one active kanban is available for the product that is to be produced next, so that no setup of the setup sequence is skipped and the manufacturing facility never idles. Based on this scenario, rough estimates for $t_S^{(i)}$ and $t_I^{(i)}$ are

$$t_S^{i:\mathrm{est}} = s_i \tag{6.4}$$

and

$$t_I^{i:\mathrm{est}} = 0.$$

The number of active kanbans at the beginning of a busy period, say B_i, must be between 1 and $K_i + B_i^{\max}$, the sum of the total number of kanbans and the maximum number of backorders for product i. Thus, the average number of active kanbans at the beginning of B_i may be (very) roughly estimated as $\frac{1}{2}\left(K_i + B_i^{\max}\right)$. A busy period ends when the number of active kanbans is reduced to zero. We estimate the average time to reduce the number of active kanbans in component C2 for product i by one unit as $\left(\mu_i - \lambda_{\mathrm{eff},i}\right)^{-1}$, where $\lambda_{\mathrm{eff},i}$ is the effective arrival rate in an $M/M/1/N$ queueing system (component C1) with average arrival rate λ_i^{ext}, average service rate μ_i, and system capacity $N = K_i + B_i^{\max}$ (including the server position). Hence, we get

$$t_B^{i:\mathrm{est}} = \frac{1}{2}\left(K_i + B_i^{\max}\right)\left(\mu_i - \lambda_{\mathrm{eff},i}\right)^{-1}, \tag{6.5}$$

as a rough estimate for $t_B^{(i)}$, where

$$\lambda_{\text{eff},i} = \begin{cases} \lambda_i^{\text{ext}} \left[1 - \dfrac{(1-\rho_i)\rho_i^N}{1-\rho_i^{N+1}} \right], & \text{if } \rho_i \neq 1, \\ \lambda_i^{\text{ext}} \left(1 - \dfrac{1}{N+1} \right), & \text{if } \rho_i = 1, \end{cases}$$

with $\rho_i = \lambda_i^{\text{ext}} / \mu_i$ and $N = K_i + B_i^{\max}$.

Finally, since $t_S^{i:\text{est}} = s_i$ and $t_I^{i:\text{est}} = 0$, we have

$$t_{\text{SBI}}^{i:\text{est}} = t_S^{i:\text{est}} + t_B^{i:\text{est}} + t_I^{i:\text{est}} = s_i + \frac{K_i + B_i^{\max}}{2(\mu_i - \lambda_{\text{eff},i})}. \tag{6.6}$$

6.1.2 Equations for Transition Rates μ_i' and μ_i''

Before the balance equations for the CTMC of component C2 for product i may be solved, transition rates μ_i' and μ_i'' for transitions $(1,B) \rightarrow (0,I)$ and $(1,B) \rightarrow (0,V_{i+1})$ must be determined (Fig. 6.3).

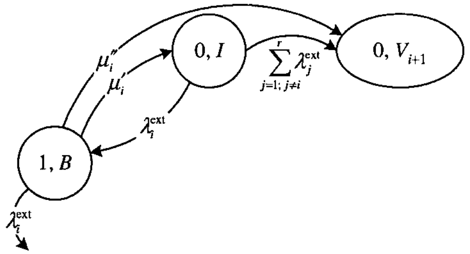


Fig. 6.3. Isolated section of the state-transition rate diagram of the CTMC for the stand-alone version of component C2 (product i)

Transition $(1,B) \rightarrow (0,I)$ represents that the manufacturing facility switches into the idle state at the end of a busy period. This happens if no kanban is active for any product. Otherwise, the manufacturing facility is being set up for one of the other products, that is, the manufacturing facility begins a vacation period: transition $(1,B) \rightarrow (0,V_{i+1})$. Let E_i denote the event that no kanban is active at the end of a busy period for product i, and let $P(E_i)$ denote the probability of this event. Then we have

$$\mu_i' = P(E_i)\mu_i, \tag{6.7}$$

and

$$\mu_i'' = [1 - P(E_i)]\,\mu_i. \tag{6.8}$$

Derivation of Equations (6.7) **and** (6.8). First, we establish a relation between processing rate μ_i and transitions rates μ_i' and μ_i''. The amount of time a system spends in a state is called the *sojourn time* in this state. In a CTMC, the sojourn time in any state is exponentially distributed and the distribution parameter is the sum of the transition rates for transitions out of the state (e.g., Kulkarni 1999, Section 6.2). Hence, the sojourn time in state $(1,B)$ is exponentially distributed with parameter $\lambda_i^{\text{ext}} + \mu_i' + \mu_i''$, that is, the average sojourn time in state $(1,B)$ is $\left(\lambda_i^{\text{ext}} + \mu_i' + \mu_i''\right)^{-1}$. In the next paragraph, we show that $\lambda_i^{\text{ext}} + \mu_i' + \mu_i'' = \lambda_i^{\text{ext}} + \mu_i$. From this equation, we get

$$\mu_i = \mu_i' + \mu_i''. \tag{6.9}$$

Consider the events that may occur when component C2 for product i is in state $(1,B)$: either a kanban for product i is activated (event 1), or an empty container for product i is filled (event 2). The time until event 1 occurs is exponentially distributed with rate λ_i^{ext}. The time until event 2 occurs is exponentially distributed with rate μ_i. By definition, the sojourn time in state $(1,B)$ must be equal to the time until the first of these two events occurs. Since the minimum of a set of independent exponential random variables is an exponential random variable, and its distribution parameter is the sum of the distribution parameters of the independent random variables (e.g., Kulkarni 1999, Section 6.3), the time until the first of the two events occurs is exponentially distributed with parameter $\lambda_i^{\text{ext}} + \mu_i$. Therefore, $\lambda_i^{\text{ext}} + \mu_i' + \mu_i'' = \lambda_i^{\text{ext}} + \mu_i$.

We now establish a relation between μ_i' and μ_i''. In the CTMC for component C2 for product i, the probability that the system enters state $(0,I)$ when it leaves state $(1,B)$ is $\mu_i' / \left(\lambda_i^{\text{ext}} + \mu_i\right)$, that is, the rate of the transition from state $(1,B)$ to state $(0,I)$, μ_i', in relation to the parameter of the sojourn time distribution of state $(1,B)$ (e.g., Kulkarni 1999, Section 6.2). Likewise, the probability that the system enters state $(0,V_{i+1})$ when it leaves state $(1,B)$ is $\mu_i'' / \left(\lambda_i^{\text{ext}} + \mu_i\right)$.

In the original multi-product kanban system, the manufacturing facility switches into the idle state only if no kanbans (for any product) are active at the end of a busy period. Otherwise, the manufacturing facility is being set up for one of the other products. In the model of component C2 for product i, this behavior is represented by the transitions from state $(1,B)$ to state $(0,I)$ and from state $(1,B)$ to state $(0,V_{i+1})$, respectively. If E_i denotes the event that no kanbans are active at the end of a busy period for product i and if $P(E_i)$ denotes the probability of this event, then

$$\frac{\mu_i'}{\lambda_i^{\text{ext}} + \mu_i} \bigg/ \frac{\mu_i''}{\lambda_i^{\text{ext}} + \mu_i} = \frac{P(E_i)}{1 - P(E_i)}$$

and, thus,

$$\frac{\mu_i'}{\mu_i''} = \frac{P(E_i)}{1-P(E_i)}. \tag{6.10}$$

From Equations (6.9) and (6.10), we get Equations (6.7) and (6.8).

Probability $P(E_i)$. We now develop an approach to estimate probability $P(E_i)$. Let E_{ij} denote the event that no kanban for product j is active at the end of a busy period for product i. Clearly, event E_i occurs only if events E_{ij} for all $j = 1,\dots,r$ occur simultaneously, that is, $E_i = \bigcap_{j=1}^{r} E_{ij}$. Events E_{ij} are mutually independent for a fixed i and $j = 1,\dots,r$. Thus, the probability of event E_i is

$$P(E_i) = \prod_{j=1}^{r} P(E_{ij}). \tag{6.11}$$

Note that, by definition, $P(E_{ii}) = 1$. For any combination $i,j = 1,\dots,r$; $j \neq i$, we approximate $P(E_{ij})$ by the conditional probability that no kanban is active for product j (event A), given that the manufacturing facility is dedicated to product i (event B), that is,

$$P(E_{ij}) \approx P(A|B).$$

For $P(B) > 0$, the conditional probability of event A given event B is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

The probability of both events A and B occurring simultaneously, $P(A \cap B)$, is approximately $\tilde{q}_j(0,V_i)$, the probability of event B, $P(B)$, is approximately $\tilde{g}_j(V_i)$. Thus, we get for probability $P(E_{ij})$ that

$$P(E_{ij}) \approx \frac{\tilde{q}_j(0,V_i)}{\tilde{g}_j(V_i)}. \tag{6.12}$$

Rough estimates. Unfortunately, no values for $\tilde{q}_j(0,V_i)$ and $\tilde{g}_j(V_i)$ are available until after the first analysis of component C2 for product j. Therefore, a different approach for obtaining values for probability $P(E_{ij})$ is needed at the beginning of the algorithm. Recall that $P(E_{ij})$ is the probability that no kanban for product j is active at the end of a busy period for product i. This may happen only if no demand for product j has arrived in the time period from the end of the last busy period for product j until the end of the busy period for product i. Let t_{ji} be an estimate for the average length of this time period with

$$t_{ji} = \begin{cases} 0, & \text{if } i = j, \\ \sum_{u=j+1}^{i-1} t_{\text{SBI}}^{(u)} + t_S^{(i)} + t_B^{(i)}, & \text{if } i > j, \\ \sum_{u=j+1}^{r} t_{\text{SBI}}^{(u)} + \sum_{u=1}^{i-1} t_{\text{SBI}}^{(u)} + t_S^{(i)} + t_B^{(i)}, & \text{if } i < j. \end{cases}$$

We get a rough estimate for $P(E_{ij})$ by calculating the probability that no demand for product j arrives during a time period of length t_{ji}. Since demand for product j arrives according to a Poisson process with rate λ_j^{ext}, a rough estimate for $P(E_{ij})$ is

$$P^{\text{est}}(E_{ij}) = e^{-\lambda_j^{\text{ext}} t_{ji}}. \tag{6.13}$$

6.1.3 Algorithm for Subassembly SA1

The algorithm for subassembly SA1 consists of two parts. In the first part (steps 1, 2, and 3), initial values are determined for parameter $t_{\text{SBI}}^{(i)}$ for all $i = 2, \dots, r$. In the second part (steps 4, 5, and 6), the models of component C2 for products $1, \dots, r$ are analyzed repeatedly until all performance measures of interest change by less than ε_p (relative change). The scheme of the algorithm is illustrated in Figure 6.4.

Algorithm for Subassembly SA1

[Part I: Initialization]

Step 1. Compute rough estimates for $t_S^{(i)}$ and $t_B^{(i)}$ for all $i = 1, \dots, r$ using Equations (6.4) and (6.5).

Step 2. Compute a rough estimate for $t_{\text{SBI}}^{(i)}$ for all $i = 1, \dots, r; i \neq 2$ using Equation (6.6).

Step 3. [The 0th rotation]
For $i = 2$ to r:

- Approximate probabilities $P(E_{ij})$ for all $j = 1, \dots, r$; $j \neq i$ using Equation (6.13) if $j = 1$ or $j > i$ and Equation (6.12) if $1 < j < i$. Approximate probability $P(E_i)$ using Equation (6.11). Then calculate parameters μ_i' and μ_i'' using Equations (6.7) and (6.8).

- Solve the balance equations of the CTMC for component C2 for product i to obtain probability distribution $\tilde{q}_i$.

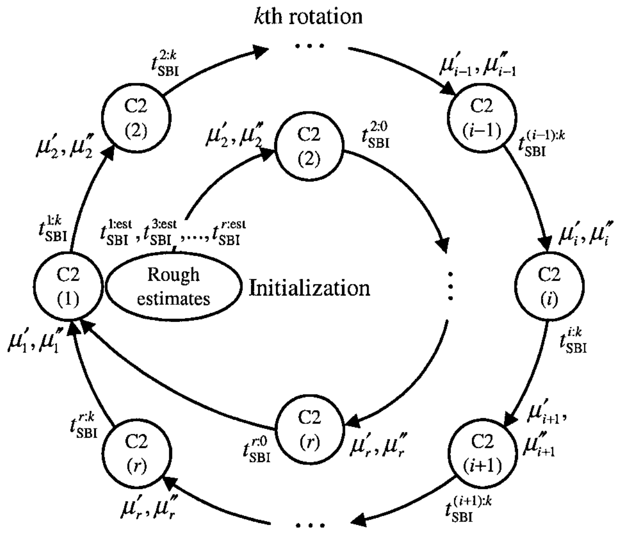


Fig. 6.4. Scheme of the algorithm for subassembly SA1; C2(i) = component C2 for product i.

- Compute probability distribution $\tilde{g}_i$ and parameters $t_{\mathrm{V}}^{(i)}$ and T_i using Equations (5.7), (6.1), and (6.2). Then calculate parameter $t_{\mathrm{SBI}}^{(i)}$ using Equation (6.3).

Next i.

[Part II: Convergence process]

Step 4. Set $k = 1$.

Step 5. [The kth rotation]
For $i = 1$ to r:

- Approximate probabilities $P\big(E_{ij}\big)$ for all $j = 1, \ldots, r$; $j \neq i$ and probability $P\big(E_i\big)$ using Equations (6.12) and (6.11). Then calculate parameters μ_i' and μ_i'' using Equations (6.7) and (6.8).

- Solve the balance equations of the CTMC for component C2 for product i to obtain probability distribution $\tilde{q}_i$.

- Compute probability distribution $\tilde{p}_i$ using Equation (5.6). Then calculate the current values (rotation k) for the performance measures of interest, for example, $\hat{f}_i^{(k)}$ and $\hat{y}_i^{(k)}$ using Equations (5.9) and (5.11).

- If $k > 1$, then: if for all performance measures of interest

$$|\text{current value} - \text{last value}|/\text{current value} < \varepsilon_p,$$

  for example, if for all $i = 1,\ldots,r$ $\left|\hat{f}_i^{(k)} - \hat{f}_i^{(k-1)}\right|/\hat{f}_i^{(k)} < \varepsilon_p$ and $\left|\hat{y}_i^{(k)} - \hat{y}_i^{(k-1)}\right|/\hat{y}_i^{(k)} < \varepsilon_p$, then STOP.

- Compute probability distribution $\tilde{g}_i$ and parameters $t_V^{(i)}$ and T_i using Equations (5.7), (6.1), and (6.2). Then calculate parameter $t_{\text{SBI}}^{(i)}$ using Equation (6.3).

Next i.

Step 6. Set $k = k + 1$. Go to Step 5.

6.2 Subassembly SA2: Model of a Multi-Product Manufacturing Facility Fed by Single-Product Facilities

Subassembly SA2 is the result of linking two or more C3-components (Fig. 6.5). The main difference to subassembly SA1 is that input material for the multi-product manufacturing facility in stage 2 may be out of stock in the output stores of stage 1. The linking-process for C3-components is conceptually identical to the linking-process for C2-components: values for a set of linking parameters are determined by means of an iterative procedure. In fact, the same equations may be used for parameters $t_{\text{SBI}}^{(i)}$, μ_i', and μ_i''; only the approach for generating rough estimates for $t_{\text{SBI}}^{(i)}$ must be adapted slightly. The approximation of probabilities $P(E_{ij})$ must be modified, and a value for the additional transition parameter Λ_i must be obtained before component C3 for product i may be analyzed.

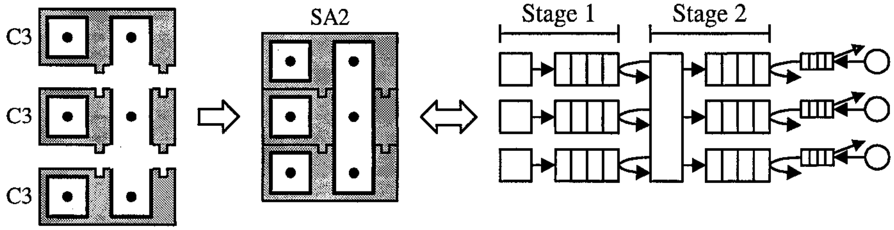


Fig. 6.5. C3-components, subassembly SA2, and corresponding kanban system

6.2.1 Rough Estimates for Parameter $t_{\mathrm{SBI}}^{(i)}$

The number of active kanbans and backorders (in stage 2) at the beginning of a busy period, say B_i, must be between 1 and $K_i^{(2)} + B_i^{\max,2}$, the sum of the total number of kanbans and the maximum number of backorders for product i in stage 2. Hence, the average number of active kanbans at the beginning of B_i may be (very) roughly estimated as $\frac{1}{2}\left(K_i^{(2)} + B_i^{\max,2}\right)$. A busy period ends when either the number of active kanbans or the number of containers with input material is reduced to zero. We estimate the average time to reduce the number of active kanbans in component C3 for product i by one unit as $\left(\mu_i^{(2)} - \lambda_{\mathrm{eff},i}\right)^{-1}$, where $\lambda_{\mathrm{eff},i}$ is the effective arrival rate in an $M/M/1/N$ queueing system (component C1) with average arrival rate λ_i^{ext}, average service rate $\mu_i^{(2)}$, and system capacity $N = K_i^{(2)} + B_i^{\max,2}$ (including the server position). Assuming that input material is always available, we get as a rough estimate for $t_B^{(i)}$

$$t_B^{i:\mathrm{est}} = \frac{1}{2}\left(K_i^{(2)} + B_i^{\max,2}\right)\left(\mu_i^{(2)} - \lambda_{\mathrm{eff},i}\right)^{-1}, \tag{6.14}$$

where

$$\lambda_{\mathrm{eff},i} = \begin{cases} \lambda_i^{\mathrm{ext}}\left[1 - \dfrac{(1-\rho_i)\rho_i^N}{1-\rho_i^{N+1}}\right], & \text{if } \rho_i \neq 1, \\ \lambda_i^{\mathrm{ext}}\left(1 - \dfrac{1}{N+1}\right), & \text{if } \rho_i = 1, \end{cases}$$

with $\rho_i = \lambda_i^{\mathrm{ext}}/\mu_i^{(2)}$ and $N = K_i^{(2)} + B_i^{\max,2}$.

Finally, with $t_I^{i:\mathrm{est}} = 0$ and $t_S^{i:\mathrm{est}} = s_i^{(2)}$, we have

$$t_{\mathrm{SBI}}^{i:\mathrm{est}} = t_S^{i:\mathrm{est}} + t_B^{i:\mathrm{est}} + t_I^{i:\mathrm{est}} = s_i^{(2)} + \frac{K_i^{(2)} + B_i^{\max,2}}{2\left(\mu_i^{(2)} - \lambda_{\mathrm{eff},i}\right)}. \tag{6.15}$$

6.2.2 Approximation of Probabilities $P(E_{ij})$

Let E_{ij} denote the event that, at the end of a busy period for product i in the multi-product manufacturing facility in stage 2, product j does not meet the setup condition. In contrast to component C2, the setup condition in component C3 consists of two parts: at least one kanban must be active (in stage 2) *and* at least one container with input material must be present in the output store of stage 1.

By definition, $P(E_{ii}) = 1$. To approximate probability $P(E_{ij})$ for all $j = 1,\ldots,r$; $j \neq i$, note that

$$P(E_{ij}) = P\left(E_{ij}^{n_j=0 \vee y_j=0}\right) = P\left(E_{ij}^{n_j=0 \wedge y_j>0}\right) + P\left(E_{ij}^{n_j>0 \wedge y_j=0}\right) + P\left(E_{ij}^{n_j=0 \wedge y_j=0}\right) \tag{6.16}$$

if n_j denotes the number of active kanbans and backorders for product j in stage 2, y_j denotes the number of full containers in the output store of stage 1 for product j, $E_{ij}^{n_j=0 \vee y_j=0}$ denotes the event that $n_j = 0$ or $y_j = 0$ (inclusive *or*) at the end of a busy period for product i, and, for example, $E_{ij}^{n_j=0 \wedge y_j>0}$ denotes the event that $n_j = 0$ and $y_j > 0$ at the end of a busy period for product i. To simplify the notation, we substitute $P(E_{ij}^{n})$, $P(E_{ij}^{y})$, and $P(E_{ij}^{n,y})$ for $P(E_{ij}^{n_j=0 \wedge y_j>0})$, $P(E_{ij}^{n_j>0 \wedge y_j=0})$, and $P(E_{ij}^{n_j=0 \wedge y_j=0})$.

Consider probability $P(E_{ij}^{n})$. The conditional probability that input material is available, but no kanban is active for product j given that the manufacturing facility is dedicated to product i may serve as an approximation for the probability that both conditions are met *at the end* of a busy period for product i. This conditional probability for the approximate model of component C3 for product i is equal to $\left[\tilde{q}_j(0,V_i) - \tilde{o}_j(0,0,V_i)\right] / \tilde{g}_j(V_i)$. Hence,

$$P(E_{ij}^{n}) \approx \frac{\tilde{q}_j(0,V_i) - \tilde{o}_j(0,0,V_i)}{\tilde{g}_j(V_i)}. \tag{6.17}$$

Equivalently,

$$P(E_{ij}^{y}) \approx \frac{\tilde{h}_j(0,V_i) - \tilde{o}_j(0,0,V_i)}{\tilde{g}_j(V_i)} \tag{6.18}$$

and

$$P(E_{ij}^{n,y}) \approx \frac{\tilde{o}_j(0,0,V_i)}{\tilde{g}_j(V_i)}. \tag{6.19}$$

Rough estimates. No values are available for the steady-state probability distributions of the model of component C3 for product j until after the first analysis of this component. Therefore, other approximations are required for probabilities $P(E_{ij}^{n})$, $P(E_{ij}^{y})$, and $P(E_{ij}^{n,y})$ at the beginning of the algorithm.

If $P(E_{ij}^{n_j=0})$ denotes the probability that no kanban is active for product j at the end of the busy period for product i (regardless of the number of containers with input material) and if t_{ji} is an estimate for the average time from the last busy period for product j until the end of the busy period for product i, then—assuming that n_j was zero at the end of the last busy period for product j—a (very) rough estimate for $P(E_{ij}^{n_j=0})$ is

$$P^{\text{est}}(E_{ij}^{n_j=0}) = e^{-\lambda_j^{\text{ext}} t_{ji}},$$

where

$$t_{ji} = \begin{cases} 0, & \text{if } i = j, \\ \sum_{u=j+1}^{i-1} t_{\text{SBI}}^{(u)} + t_S^{(i)} + t_B^{(i)}, & \text{if } i > j, \\ \sum_{u=j+1}^{r} t_{\text{SBI}}^{(u)} + \sum_{u=1}^{i-1} t_{\text{SBI}}^{(u)} + t_S^{(i)} + t_B^{(i)}, & \text{if } i < j. \end{cases}$$

By similar argument, we get

$$P^{\mathrm{est}}\left(E_{ij}^{y_j=0}\right) = e^{-\mu_j^{(1)} t_{ji}}.$$

Then, using basic probability calculus, we obtain

$$P^{\mathrm{est}}\left(E_{ij}^{n}\right) = P^{\mathrm{est}}\left(E_{ij}^{n_j=0}\right)\left[1 - P^{\mathrm{est}}\left(E_{ij}^{y_j=0}\right)\right] = e^{-\lambda_j t_{ji}} - e^{-(\lambda_j^{\mathrm{ext}} + \mu_j^{(1)}) t_{ji}}, \tag{6.20}$$

$$P^{\mathrm{est}}\left(E_{ij}^{y}\right) = \left[1 - P^{\mathrm{est}}\left(E_{ij}^{n_j=0}\right)\right] P^{\mathrm{est}}\left(E_{ij}^{y_j=0}\right) = e^{-\mu_j^{(1)} t_{ji}} - e^{-(\lambda_j^{\mathrm{ext}} + \mu_j^{(1)}) t_{ji}}, \tag{6.21}$$

and

$$P^{\mathrm{est}}\left(E_{ij}^{n,y}\right) = P^{\mathrm{est}}\left(E_{ij}^{n_j=0}\right) P^{\mathrm{est}}\left(E_{ij}^{y_j=0}\right) = e^{-(\lambda_j^{\mathrm{ext}} + \mu_j^{(1)}) t_{ji}}. \tag{6.22}$$

6.2.3 Equation for Transition Rate Λ_i

The idle period in component C3 for product i ends as soon as one product meets the setup condition. If this product is product i, then the manufacturing facility immediately resumes production, that is, the manufacturing facility switches back into state B $\left(\text{with transition rate } \mu_i^{(1)}\right)$. If, however, the first product that satisfies the setup condition is one of the other products, then the manufacturing facility starts a vacation period, that is, the state of the manufacturing facility changes from I to V_{i+1} (Figs. 5.20 and 5.21). The rate for this transition must be the reciprocal of the mean of an exponentially distributed random variable that contains the time from the beginning of the idle period until one product other than product i meets the setup condition. We denote this reciprocal by Λ_i. If l_{ij} denotes the average time from the beginning of the idle period after processing items of product i until product j meets the setup condition and if this time actually followed an exponential distribution, then

$$\Lambda_i = \sum_{j=1; j \neq i}^{r} \frac{1}{l_{ij}} \tag{6.23}$$

since the minimum of a set of independent exponential random variables is an exponential random variable, and the parameter of this random variable is the sum of the parameters of the individual random variables (e.g., Kulkarni 1999, Section 6.3). The exact average time l_{ij}, however, is unknown. It may only be approximated because no information is available in the model of component C3 for product i on the number of active kanbans for product j in stage 2 and the number of containers with input material for product j in the output store of stage 1 $(j = 1, \ldots, r; j \neq i)$.

Approximation of l_{ij}. When product j does not meet the setup condition, this may be because (1) input material is available, but no kanban is active in stage 2 or (2) one or more kanbans are active in stage 2, but no input material is available or (3) because no kanban is active *and* no input material is available. The average time until product j meets the setup condition is consequently the weighted average of the average times until (a) a kanban is activated in stage 2, (b) a container with input material is completed in stage 1, and (c) a kanban is activated and a container with input material is completed. The weights are the conditional probabilities that the system is in the three states given that product j does not meet the setup condition at the end of a busy period for product i.

The average time until a kanban for product j is activated in stage 2 is $1/\lambda_j^{\text{ext}}$. The average time until a container with input material for product j is completed in stage 1 is $1/\mu_j^{(1)}$. The average time until a kanban is activated and a container with input material is completed is given by the expected value of a random variable containing the maximum of the time until a kanban is activated and the time until a container with input material is completed. Since both times follow exponential distributions, the average time until a kanban is activated and a container with input material is completed is $1/\lambda_j^{\text{ext}} + 1/\mu_j^{(1)} - 1/\left(\lambda_j^{\text{ext}} + \mu_j^{(1)}\right)$. Hence,

$$l_{ij} \approx \frac{P(E_{ij}^n)}{P(E_{ij})}\frac{1}{\lambda_j^{\text{ext}}} + \frac{P(E_{ij}^y)}{P(E_{ij})}\frac{1}{\mu_j^{(1)}} + \frac{P(E_{ij}^{n,y})}{P(E_{ij})}\left(\frac{1}{\lambda_j^{\text{ext}}} + \frac{1}{\mu_j^{(1)}} - \frac{1}{\lambda_j^{\text{ext}} + \mu_j^{(1)}}\right). \quad (6.24)$$

6.2.4 Algorithm for Subassembly SA2

The structure of the algorithm for subassembly SA2 is identical to the structure of the algorithm for subassembly SA1. Only the different approximation of probabilities $P(E_{ij})$ and the additional calculations for updating parameter Λ_i require some modifications.

Algorithm for Subassembly SA2

[Part I: Initialization]

Step 1. Compute rough estimates for $t_S^{(i)}$ and $t_B^{(i)}$ for all $i = 1, \ldots, r$ using Equations (6.4) and (6.14).

Step 2. Compute a rough estimate for $t_{\text{SBI}}^{(i)}$ for all $i = 1, \ldots, r; i \neq 2$ using Equation (6.15).

Step 3. [The 0th rotation]
For $i = 2$ to r:

- Approximate probabilities $P\big(E_{ij}^{n}\big)$, $P\big(E_{ij}^{y}\big)$, and $P\big(E_{ij}^{n,y}\big)$ for all $j = 1,\ldots,r$; $j \neq i$ using Equations (6.20)–(6.22) if $j = 1$ or $j > i$ and Equations (6.17)–(6.19) if $1 < j < i$. Then employ Equation (6.16) to compute probabilities $P\big(E_{ij}\big)$ for all $j = 1,\ldots,r$; $j \neq i$.

- Approximate probability $P\big(E_i\big)$ using Equation (6.11) and employ Equations (6.7) and (6.8) to calculate parameters μ_i' and μ_i''. Compute the average waiting times l_{ij} for all $j = 1,\ldots,r$; $j \neq i$ using Equation (6.24) and employ Equation (6.23) to determine transition rate Λ_i.

- Solve the balance equations of the CTMC for component C3 for product i to obtain probability distribution $\tilde{o}_i$. Then use Equations (5.13) and (5.14) to determine probability distributions $\tilde{q}_i$ and $\tilde{h}_i$.

- Compute probability distribution $\tilde{g}_i$ and parameters $t_V^{(i)}$ and T_i using Equations (5.16), (6.1), and (6.2). Then employ Equation (6.3) to calculate parameter $t_{\mathrm{SBI}}^{(i)}$.

Next i.

[Part II: Convergence process]

Step 4. Set $k = 1$.

Step 5. [The kth rotation]
For $i = 1$ to r:

- Approximate probabilities $P\big(E_{ij}^{n}\big)$, $P\big(E_{ij}^{y}\big)$, and $P\big(E_{ij}^{n,y}\big)$ for all $j = 1,\ldots,r$; $j \neq i$ using Equations (6.17)–(6.19). Then employ Equation (6.16) to compute probabilities $P\big(E_{ij}\big)$ for all $j = 1,\ldots,r$; $j \neq i$.

- Approximate probability $P\big(E_i\big)$ using Equation (6.11) and employ Equations (6.7) and (6.8) to calculate parameters μ_i' and μ_i''. Compute waiting times l_{ij} for all $j = 1,\ldots,r$; $j \neq i$ using Equation (6.24) and employ Equation (6.23) to determine transition rate Λ_i.

- Solve the balance equations of the CTMC for component C3 for product i to obtain probability distribution $\tilde{o}_i$. Then use Equations (5.13) and (5.14) to determine probability distributions $\tilde{q}_i$ and $\tilde{h}_i$.

- Compute probability distributions $\tilde{p}_i$ and $\tilde{k}_i$ using Equations (5.15) and (5.17). Then determine the current values (rotation k) for the performance measures of interest, for example, $\hat{f}_i$, $\hat{\bar{y}}_i^{(1)}$, and $\hat{\bar{y}}_i^{(2)}$ using Equations (5.19), (5.21), and (5.22).

- If $k > 1$, then: if for all performance measures of interest

  $$|\text{current value} - \text{last value}|/\text{current value} < \varepsilon_p,$$

  then STOP.

- Compute probability distribution $\tilde{g}^{(i)}$ and parameters $t_V^{(i)}$ and T_i using Equations (5.16), (6.1), and (6.2). Then employ Equation (6.3) to calculate parameter $t_{\text{SBI}}^{(i)}$.

Next i.

Step 6. Set $k = k + 1$. Go to Step 5.

7

Composite Models: Models of Multi-Stage Kanban Systems

In this chapter, we show how several C1-components may be linked to obtain models of multi-stage single-product kanban systems (Fig. 7.1). We also demonstrate how SA1- and SA2-subassemblies may be connected to obtain models of multi-stage multi-product kanban systems (Fig. 7.2). After explaining the linking technique, we give detailed descriptions of the algorithms. Algorithms for models containing C1-components *and* SA1/SA2-subassemblies (Fig. 7.3) may be derived accordingly.

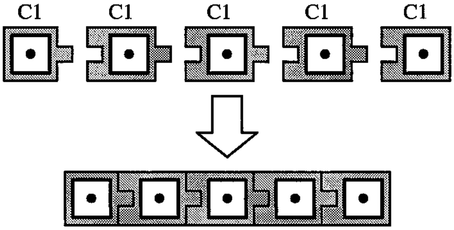


Fig. 7.1. Linking C1-components

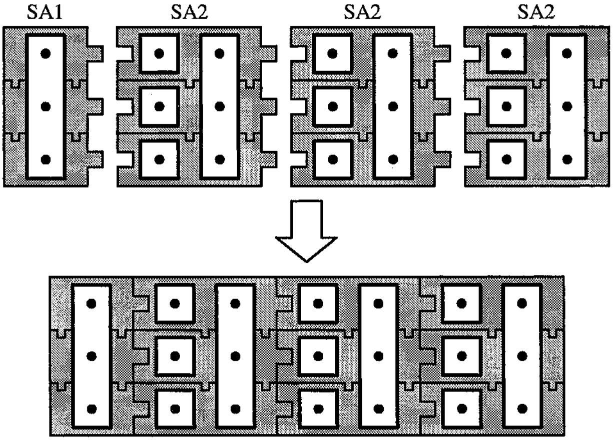


Fig. 7.2. Linking SA1- and SA2-subassemblies

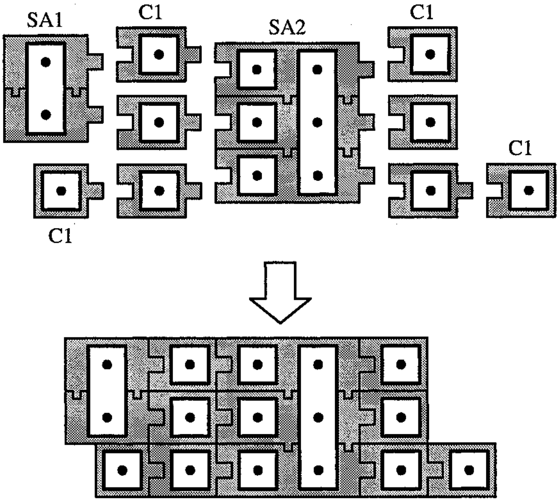


Fig. 7.3. Linking C1-components and SA1- and SA2-subassemblies

7.1 Linking Technique

Conservation of flow-property. The conceptual anchor of the linking technique is the *conservation of flow*-property of the systems under study: the exact same number of items that enter the system leave the system eventually; no parts are scrapped and no parts are newly created in the process. As a consequence, the average production rates of all stages must be identical for the same product,

$$\mathrm{TH}_i^{(1)} = \mathrm{TH}_i^{(2)} = \cdots = \mathrm{TH}_i^{(M-1)} = \mathrm{TH}_i^{(M)}.$$

Average demand arrival rates. The average demand arrival rates $\lambda_i^{(m)}$ must be determined for the start- and middle-piece versions of component C1 and the subassemblies. For each product of a stage, the average arrival rate of demand that is served immediately upon arrival or after a stochastic waiting time must be equal to the average production rate of the following stage,

$$\lambda_{\mathrm{SD},i}^{(m)} = \mathrm{TH}_i^{(m+1)}.$$

Rate $\lambda_{\mathrm{SD},i}^{(m)}$ and the (total) average demand arrival rate, $\lambda_i^{(m)}$, are linked by equation

$$\lambda_{\mathrm{SD},i}^{(m)} = f_{\mathrm{SD},i}^{(m)} \lambda_i^{(m)},$$

where $f_{\mathrm{SD},i}^{(m)}$ denotes the average fraction of served demand for product i in stage m. Thus, we obtain the equation

$$\lambda_i^{(m)} = \lambda_{\mathrm{SD},i}^{(m)} / f_{\mathrm{SD},i}^{(m)} = \mathrm{TH}_i^{(m+1)} / f_{\mathrm{SD},i}^{(m)}.$$

Effective average processing rates. The effective average processing rates $\mu_{\mathrm{eff},i}^{(m)}$ must be determined for the single-product manufacturing facilities in the middle- and end-piece versions of component C1 and subassembly SA2. For each single-product manufacturing facility, the effective average processing rate is determined based on the average waiting time until input material is available, $w_{\mathrm{IM},i}^{(m)}$, and the nominal average processing rate, $\mu_i^{(m)}$,

$$\mu_{\mathrm{eff},i}^{(m)} = \left(w_{\mathrm{IM},i}^{(m)} + 1/\mu_i^{(m)} \right)^{-1}.$$

The average waiting time until input material is available in stage m is conceptually identical to the average waiting time of served demand in stage $m-1$, $w_{\mathrm{SD},i}^{(m-1)}$. Thus, we obtain the equation

$$\mu_{\mathrm{eff},i}^{(m)} = \left(w_{\mathrm{SD},i}^{(m-1)} + 1/\mu_i^{(m)} \right)^{-1}.$$

Initial values. At the beginning of the stage-linking algorithms, initial values are needed for the average demand arrival rates $\lambda_i^{(m)}$ of the start- and middle-piece versions of component C1 and the subassemblies and for the effective average container processing rates $\mu_{\text{eff},i}^{(m)}$ of the single-product manufacturing facilities in the middle- and end-piece versions of component C1 and subassembly SA2. Rather arbitrarily, we use the average production rates of stage $m+1$ as initial values for the average demand arrival rates in stage m,

$$\hat{\lambda}_i^{m:\text{init}} = \text{TH}_i^{(m+1)}.$$

As initial values for the effective average container processing rates we employ the nominal average container processing rates,

$$\hat{\mu}_{\text{eff},i}^{m:\text{init}} = \mu_i^{(m)}.$$

7.2 Algorithm for Linking C1-Components

In this section, we give a detailed description of the algorithm for linking C1-components. The scheme of the algorithm is sketched in Figure 7.4, the essential steps are visualized in Figure 7.5. Since approximations are used in the analysis of the different versions of component C1, only approximate values may be obtained for the performance measures of interest, for example, $\hat{f}_{\text{SD}}^{(M)}$ and $\hat{y}^{(m)}$.

Algorithm for Linking C1-Components

```
Step 1. [Initialization pass]
        Set $\hat{\mu}_{\text{eff}}^{m:\text{init}} = \mu^{(m)}$ for all $m = 2,3,\ldots,M$.
        For $m = M,\ldots,1$:
          • If $m < M$: set $\hat{\lambda}^{m:\text{init}} = \widehat{\text{TH}}^{(m+1)}$.
          • If $m > 1$: compute $\widehat{\text{TH}}^{(m)}$ using Equation (5.3).
        Next $m$.

Step 2. [Forward pass]
        For $m = 1,\ldots,M$:
          • If $m > 1$: compute $\hat{\mu}_{\text{eff}}^{(m)} = \left(\hat{w}_{\text{SD}}^{(m-1)} + 1/\mu^{(m)}\right)^{-1}$.
          • If $m < M$: compute $\hat{f}_{\text{SD}}^{(m)}$ and $\hat{w}_{\text{SD}}^{(m)}$ using Equations (5.1) and (5.5).
        Next $m$.
```

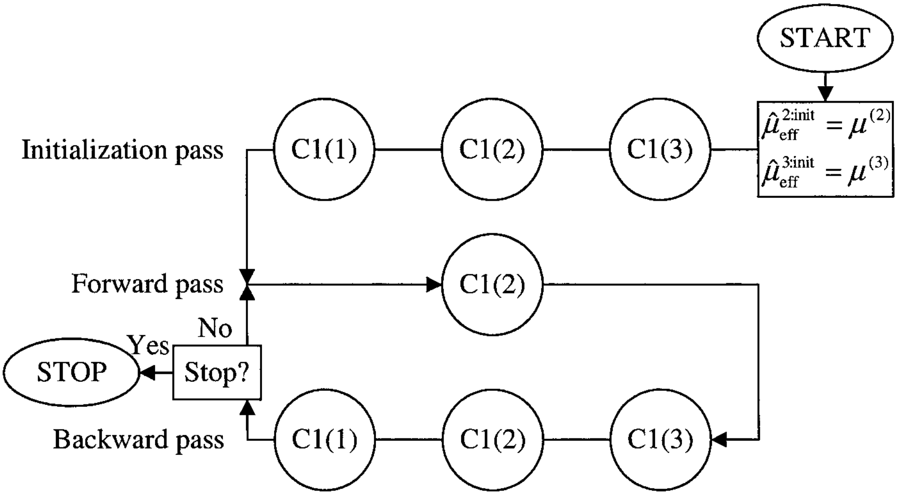

START
$\hat{\mu}_{\text{eff}}^{2:\text{init}} = \mu^{(2)}$
$\hat{\mu}_{\text{eff}}^{3:\text{init}} = \mu^{(3)}$
Initialization pass
C1(1)
C1(2)
C1(3)
Forward pass
C1(2)
No
Yes
STOP
Stop?
Backward pass
C1(1)
C1(2)
C1(3)

Fig. 7.4. Scheme of the algorithm for linking C1-components; C1(*m*) = component C1 for stage *m*.

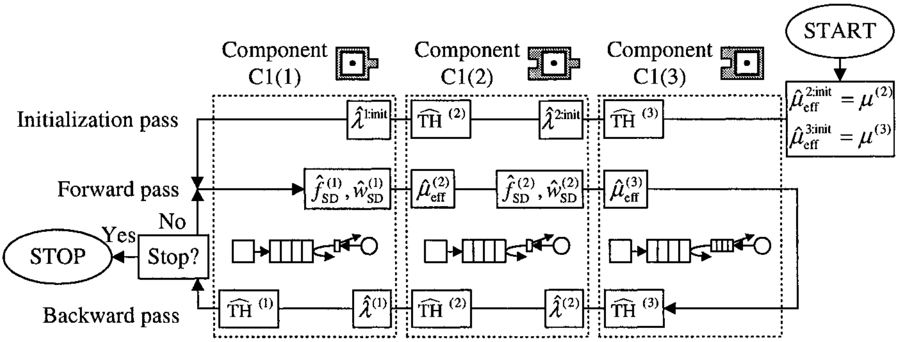

START
Component C1(1)
Component C1(2)
Component C1(3)
$\hat{\mu}_{\text{eff}}^{2:\text{init}} = \mu^{(2)}$
$\hat{\mu}_{\text{eff}}^{3:\text{init}} = \mu^{(3)}$
Initialization pass
$\hat{\lambda}^{1:\text{init}}$
$\widehat{\text{TH}}^{(2)}$
$\hat{\lambda}^{2:\text{init}}$
$\widehat{\text{TH}}^{(3)}$
Forward pass
$\hat{f}_{\text{SD}}^{(1)}, \hat{w}_{\text{SD}}^{(1)}$
$\hat{\mu}_{\text{eff}}^{(2)}$
$\hat{f}_{\text{SD}}^{(2)}, \hat{w}_{\text{SD}}^{(2)}$
$\hat{\mu}_{\text{eff}}^{(3)}$
No
Yes
STOP
Stop?
Backward pass
$\widehat{\text{TH}}^{(1)}$
$\hat{\lambda}^{(1)}$
$\widehat{\text{TH}}^{(2)}$
$\hat{\lambda}^{(2)}$
$\widehat{\text{TH}}^{(3)}$

Fig. 7.5. Essential steps of the algorithm for linking C1-components; C1(*m*) = component C1 for stage *m*.

Step 3. [Backward pass]

For $m = M, \dots, 1$:

- If $m < M$: compute $\hat{\lambda}^{(m)} = \widehat{\mathrm{TH}}^{(m+1)} / \hat{f}_{\mathrm{SD}}^{(m)}$.
- Compute $\widehat{\mathrm{TH}}^{(m)}$ using Equation (5.3).

Next m.

Step 4. [Stopping criterion]

If $\widehat{\mathrm{TH}}^{(1)} \approx \cdots \approx \widehat{\mathrm{TH}}^{(M)}$, that is, $\left|\widehat{\mathrm{TH}}^{(m)} - \widehat{\mathrm{TH}}^{(1)}\right| / \widehat{\mathrm{TH}}^{(1)} < \varepsilon_s$ for all $m = 2, \dots, M$, then go to Step 5.
Otherwise, go to Step 2.

Step 5. [Determination of performance measures]

Compute performance measures of interest, for example, $\hat{f}^{(M)}$ and $\hat{y}^{(m)}$ for all $m = 1, \dots, M$ using Equations (5.2) and (5.4), then STOP.

7.3 Algorithm for Linking SA1- and SA2-Subassemblies

In this section, we give a detailed description of the algorithm for linking a start-piece version of subassembly SA1 and one or more SA2-subassemblies. Figure 7.6 visualizes the scheme of the algorithm, Figure 7.7 illustrates the essential steps.

Algorithm for Linking SA1- and SA2-Subassemblies

Step 1. [Initialization]

Set $\hat{\mu}_{\mathrm{eff},i}^{m:\mathrm{init}} = \mu_i^{(m)}$ for all $i = 1, \dots, r$ and $m = 2, 4, \dots, M-1$.

For $m = M, M-2, M-4, \dots, 3, 1$:

- If $m < M$: set $\hat{\lambda}_i^{m:\mathrm{init}} = \widehat{\mathrm{TH}}_i^{(m+1)}$ for all $i = 1, \dots, r$.
- If $m > 1$: analyze subassembly SA2 for stages $m-1$ and m.
  If $m = 1$: analyze subassembly SA1 for stage 1.
- If $m > 1$: compute $\widehat{\mathrm{TH}}_i^{(m)}$ for all $i = 1, \dots, r$ using Equation (5.20).
  Set $\widehat{\mathrm{TH}}_i^{(m-1)} = \widehat{\mathrm{TH}}_i^{(m)}$ for all $i = 1, \dots, r$.

Next m.

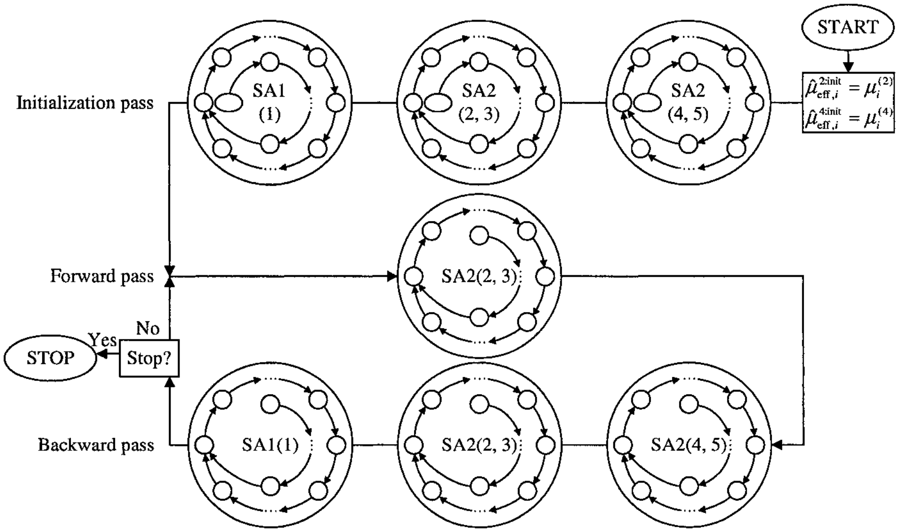


Fig. 7.6. Scheme of the algorithm for linking SA1- and SA2-subassemblies; SA1(1) = subassembly SA1 for stage 1, SA2($m-1,m$) = subassembly SA2 for stages $m-1$ and m.

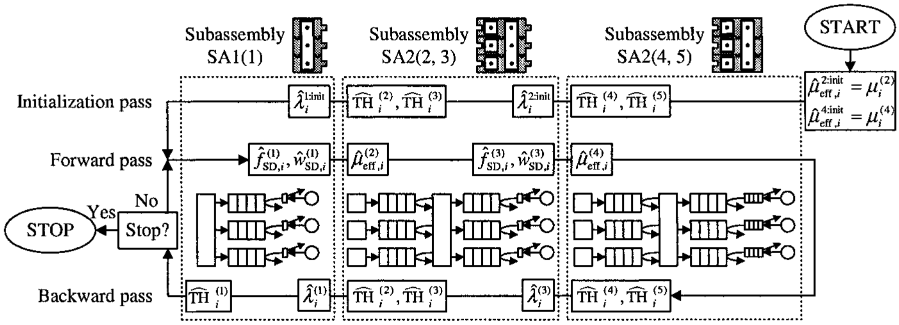

Subassembly SA1(1)
Subassembly SA2(2, 3)
Subassembly SA2(4, 5)
START
$\hat{\mu}^{2:\text{init}}_{\text{eff},i} = \mu^{(2)}_i$
$\hat{\mu}^{4:\text{init}}_{\text{eff},i} = \mu^{(4)}_i$
Initialization pass
$\hat{\lambda}^{1:\text{init}}_i$
$\widehat{\text{TH}}^{(2)}_i, \widehat{\text{TH}}^{(3)}_i$
$\hat{\lambda}^{2:\text{init}}_i$
$\widehat{\text{TH}}^{(4)}_i, \widehat{\text{TH}}^{(5)}_i$
Forward pass
$\hat{f}^{(1)}_{\text{SD},i}, \hat{w}^{(1)}_{\text{SD},i}$
$\hat{\mu}^{(2)}_{\text{eff},i}$
$\hat{f}^{(3)}_{\text{SD},i}, \hat{w}^{(3)}_{\text{SD},i}$
$\hat{\mu}^{(4)}_{\text{eff},i}$
No
Yes
STOP
Stop?
Backward pass
$\widehat{\text{TH}}^{(1)}_i$
$\hat{\lambda}^{(1)}_i$
$\widehat{\text{TH}}^{(2)}_i, \widehat{\text{TH}}^{(3)}_i$
$\hat{\lambda}^{(3)}_i$
$\widehat{\text{TH}}^{(4)}_i, \widehat{\text{TH}}^{(5)}_i$

Fig. 7.7. Essential steps of the algorithm for linking SA1- and SA2-subassemblies; SA1(1) = subassembly SA1 for stage 1, SA2($m-1,m$) = subassembly SA2 for stages $m-1$ and m.

Step 2. [Forward pass]

For $m = 1,3,5,\ldots,M-2,M$:

- If $m > 1$: compute $\hat{\mu}_{\text{eff},i}^{(m-1)} = \left(\hat{w}_{\text{SD},i}^{(m-2)} + 1/\mu_i^{(m-1)}\right)^{-1}$ for all $i = 1,\ldots,r$.
- If $m > 1$ and $m < M$: analyze subassembly SA2 for stages $m-1$ and m.
- If $m < M$: compute $\hat{f}_{\text{SD},i}^{(m)}$ and $\hat{w}_{\text{SD},i}^{(m)}$ for all $i = 1,\ldots,r$ using Equations (5.8) and (5.12) for $m = 1$ and (5.18) and (5.23) for $m > 1$, respectively.

Next m.

Step 3. [Backward pass]

For $m = M, M-2, M-4,\ldots,3,1$:

- If $m < M$: compute $\hat{\lambda}_i^{(m)} = \widehat{\text{TH}}_i^{(m+1)}/\hat{f}_{\text{SD},i}^{(m)}$ for all $i = 1,\ldots,r$.
- If $m > 1$: analyze subassembly SA2 for stages $m-1$ and m.
  If $m = 1$: analyze subassembly SA1 for stage 1.
- Compute $\widehat{\text{TH}}_i^{(m)}$ for all $i = 1,\ldots,r$ using Equation (5.10) for $m = 1$ and (5.20) for $m > 1$, respectively.

Next m.

Step 4. [Stopping criterion]

If $\widehat{\text{TH}}_i^{(1)} \approx \cdots \approx \widehat{\text{TH}}_i^{(M)}$, that is, $|\widehat{\text{TH}}_i^{(m)} - \widehat{\text{TH}}_i^{(1)}|/\widehat{\text{TH}}_i^{(1)} < \varepsilon_s$ for all $i = 1,\ldots,r$ and $m = 3,5,7,\ldots,M-2,M$,
or if $|\widehat{\text{TH}}_i^{m:\text{new}} - \widehat{\text{TH}}_i^{m:\text{last}}|/\widehat{\text{TH}}_i^{m:\text{new}} < 0.1\varepsilon_s$ for all $i = 1,\ldots,r$ and $m = 1,3,5,\ldots,M-2,M$, then STOP.
Otherwise, go to Step 2.

8

Extended Application: Models of Systems with Multi-Product Manufacturing Facilities in Series

The described construction elements and linking techniques do not permit to directly model kanban systems with multi-product manufacturing facilities in series, such as the "original systems" in Figures 8.1 and 8.2. The stage-linking technique (Chap. 7) is valid only for connecting single-product manufacturing facilities (C1-components) and for connecting multi-product and single-product manufacturing facilities (subassembly SA2 and C1-components [only in that order] or several SA2-subassemblies). An indirect approach may, however, be employed to extend the applicability of the construction kit to systems with two or more multi-product manufacturing facilities in immediate succession.

The general notion of the indirect approach is to "insert" a single-product manufacturing facility with extremely small average processing time into each product-specific output store of those multi-product manufacturing facilities that supply input material to another multi-product manufacturing facility. Models of the resulting "substitute systems" (two examples are given in Figs. 8.1 and 8.2) may then be built with the elements of the construction kit (Figs. 8.3 and 8.4). The only purpose of the inserted single-product manufacturing facilities is to obtain systems that may be modeled directly with the elements of the construction kit. Because of the extremely small average processing times, the performance measures of the substitute systems

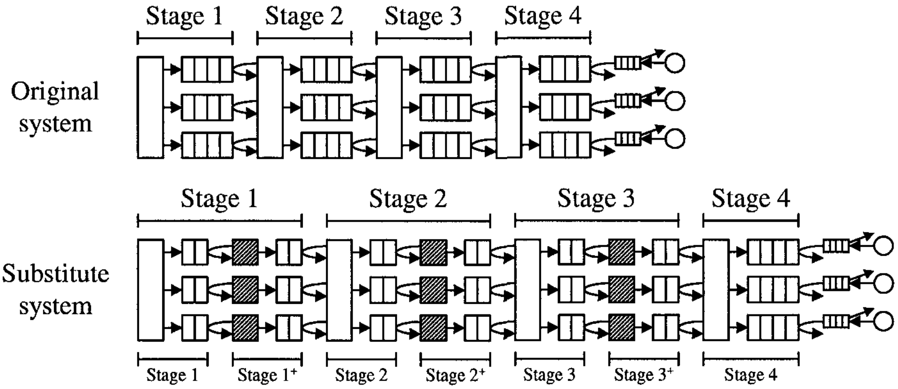


Fig. 8.1. Extended application, example 1; original system and substitute system

should be approximately equal to the respective performance measures of the original systems.

The suggested indirect approach consists of three steps:

1. Build a substitute system by inserting single-product manufacturing facilities into the output stores of multi-product manufacturing facilities that supply input material to another multi-product manufacturing facility.
2. Analyze the substitute system.
3. Derive performance measures of the original system based on the performance measures of the substitute system.

In the following three sections, we describe the details of these steps. Symbols relating to the original system are indicated by the additional superscript O, symbols relating to the substitute system are indicated by the additional superscript S. The inserted single-product manufacturing facilities in the substitute systems are referred to by the stage index of the preceding multi-product facility supplemented by a plus sign, for example, $K_i^{S:m^+}$ denotes the number of kanbans for product i in the single-product manufacturing facility inserted into the output store of stage m, that is, the number of kanbans for product i in stage m^+ of the substitute system.

8.1 Building the Substitute System

When inserting single-product manufacturing facilities into the product-specific output stores of a multi-product manufacturing facility, the original capacity of the out-

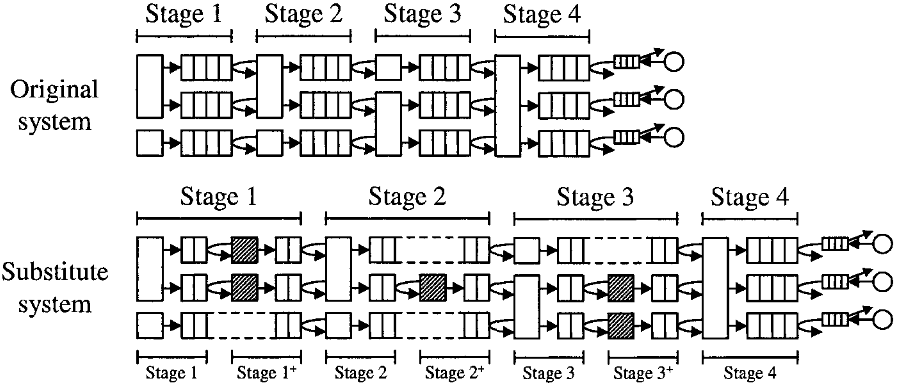


Fig. 8.2. Extended application, example 2; original system and substitute system

put store—and, hence, the available kanbans—must be divided among the multi-product facility and the inserted single-product facilities. For each product, the sum of the number of kanbans in both stages of the substitute system must be identical to the number of kanbans in the corresponding stage of the original system, that is, $K_i^{O:m} = K_i^{S:m} + K_i^{S:m^+}$, so that the maximum inventory level for each product in the considered section of the manufacturing system is identical in both the original and the substitute system.

It remains to define how the kanbans for a product should be distributed. We suggest to divide them equally among the multi-product manufacturing facility and the inserted single-product manufacturing facility. If the number of kanbans is uneven, we favor the multi-product manufacturing facility,

$$K_i^{S:m} = \left\lceil K_i^{O:m}/2 \right\rceil \quad \text{and} \quad K_i^{S:m^+} = \left\lfloor K_i^{O:m}/2 \right\rfloor .$$

In a kanban system, the number of kanbans must be greater than or equal to one (for each product in each stage). The proposed indirect approach is therefore restricted to kanban systems with at least *two* kanbans for each product of a multi-product manufacturing facility that is followed directly by another multi-product manufacturing facility.

8.2 Analyzing the Substitute System

The substitute system may be analyzed using the construction-kit approach. For the example system in Figure 8.1, a model of the substitute system (Fig. 8.3)

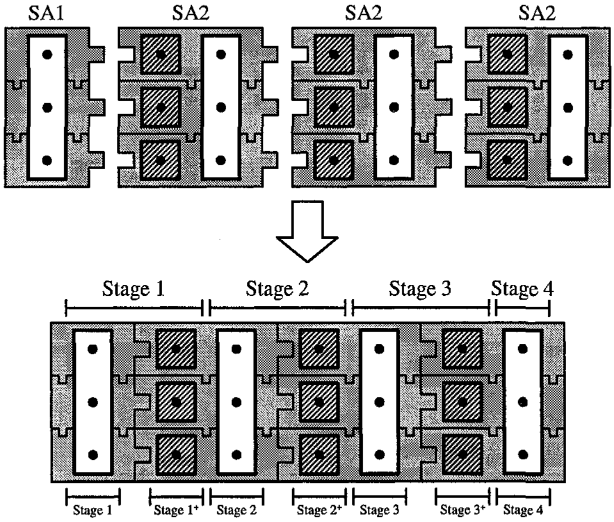


Fig. 8.3. Extended application, example 1; model of the substitute system

may be constructed with one SA1-subassembly (start-piece version) and three SA2-subassemblies (two middle-piece versions and one end-piece version). The algorithm given in Section 7.3 may then be employed to obtain approximate values for performance measures of the substitute system. For the example system in Figure 8.2, the model of the substitute system built with elements of the construction kit is depicted in Figure 8.4.

8.3 Deriving Performance Measures of the Original System

Based on the performance measures of the substitute system, approximate values for performance measures of the original system may be derived easily. Most estimates for the substitute system may even be used directly as estimates for the original system. Examples are the average fill rates, the average backorder levels, and the average waiting times of backordered demand in the last stage,

$$\hat{f}_i^{O:M} = \hat{f}_i^{S:M}, \quad \hat{b}_i^{O:M} = \hat{b}_i^{S:M}, \quad \text{and} \quad \hat{w}_{\mathrm{BD},i}^{O:M} = \hat{w}_{\mathrm{BD},i}^{S:M}.$$

To obtain approximate values for the average inventory levels in stages with inserted single-product manufacturing facilities, the average inventory level of the output store of the single-product manufacturing facility is added to the average inven-

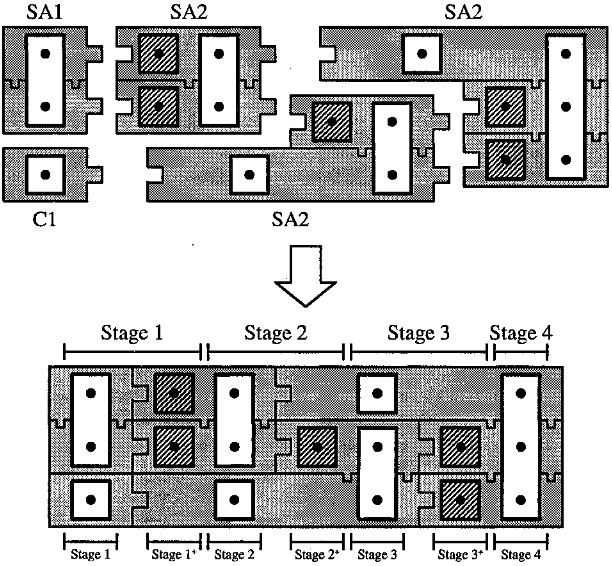


Fig. 8.4. Extended application, example 2; model of the substitute system

tory level of the output store of the multi-product manufacturing facility,

$$\hat{y}_i^{O:m} = \hat{y}_i^{S:m} + \hat{y}_i^{S:m^+}.$$

9

Accuracy of the Models: Numerical Results

9.1 Experimental Design

For each algorithm, we conducted a series of tests to determine the magnitude of the approximation errors that may be expected for selected performance measures. As reference, we either took the results of a discrete-event simulation or, for very small systems, the results of the exact model. To enable a meaningful comparison of the approximation errors across different performance measures, we report the relative, or percentage, deviation of each approximate value from its respective reference value, for example, $(\hat{f}-f)/f$. Note that a positive relative deviation signals that the approximate model overestimated the true value and that a negative relative deviation indicates that the approximate model underestimated the true value.

Test instances. Rather than generating a multitude of different system configurations randomly, we defined a base system for each algorithm and created different system configurations by systematically changing single system parameters or ratios of system parameters, for example, the number of products or the ratio of the average setup times for product 1 and product r. The main advantage of this systematic approach, compared to the random generation of test instances, is that changes in the approximation quality may be attributed to specific changes in the system configuration.

The only parameter of a base system that is determined directly is the average container processing rate for product 1 (in the last stage). The remaining parameters—average demand arrival rates, average container processing rates (for products $2,\dots,r$), and average setup times—are specified indirectly by a number of ratios. In this way, it is possible to characterize a group of structurally similar systems independently of the "dimensions" of the system, that is, independently of the number of stages and the number of products.

Most ratios used in the following sections are self-explanatory, for example, the setup time ratio s_1/s_r. Others, such as the traffic intensity, require an explicit definition. In accordance with common usage, we use the term *traffic intensity* for the ratio of the average demand arrival rate (for full containers) and the average container processing rate. Hence, the traffic intensity of product i, denoted by ρ_i, is $\rho_i = \lambda_i^{\text{ext}}/\mu_i$ (note that this definition excludes setup times). Additionally, for multi-product manufacturing facilities, we define the term *total traffic intensity*, denoted by ρ. Let the *total traffic intensity* of a manufacturing facility be the sum of the traffic intensities of all products processed in this manufacturing facility, for example, $\rho = \sum_{i=1}^{r} \rho_i$, if all r products share the same manufacturing facility.

Ratios such as the setup time ratio s_1/s_r only relate parameters (or ratios) of product 1 and product r. We obtain a connection to the parameters (or ratios) for products $2,\dots,r-1$ by demanding that the differences between the parameters (or ratios) of products with consecutive index values must be equal, for example, $s_2 - s_1 = s_3 - s_2 = \cdots = s_r - s_{r-1}$; hence, given s_1 and s_r, the average setup time for product i, for example, is $s_i = s_1 + (i-1)(s_r - s_1)/(r-1)$.

Number of kanbans. In addition to the average demand arrival rates, container processing rates, and setup times, the number of kanbans for each product in each stage must also be specified for each test instance. Rather than choosing the number of kanbans arbitrarily, we tried to find realistic kanban configurations in the sense that each test system had to guarantee certain average fill rates with minimum cost. We generally postulated that a system should be able to satisfy 95% of the demand for each product from stock, that is, that the system should achieve minimum average fill rates of 0.95. Simplifying the cost side, we only considered inventory holding costs and assumed that it was sufficient to minimize the total average inventory for all products in all stages (that is, similar holding costs for all products in all stages). As a result, the problem of finding a realistic kanban configuration could be formulated as: *find the combination of the number of kanbans for each product in each stage that minimizes the total average inventory in the system while guaranteeing given average fill rates*. For the different types of systems studied in the following sections, we constructed five local-search heuristics. The algorithms of these heuristics are given in the Appendix. The kanban configurations that resulted from applying the heuristic procedures are listed in the following sections along with selected results of the tests.

Implementation issues. The Markov chain models were generated with the approach implemented in the software package MARCA (Stewart 1991, 1996). The resulting balance equations were solved with the Gauss-Seidel method (e.g., Stewart 1994, Chap. 3). A minimum number of 20 iterations were performed and the iteration process was stopped when for every element of the steady-state probability vector the difference between the current and the last value divided by the current value changed by less than 10^{-7}.

The algorithms were implemented in Visual Basic 5.0 and run on a PC with a Pentium III processor at 733 MHz. They converged for all test instances. For each test instance, we report the number of rotations (subassemblies SA1 and SA2) or iterations (linking C1-components, extended application). The number of rotations is non-integer for many test instances. This indicates that the stopping criterion was met before the last rotation was completed. We also list the observed computing times. Note that in writing the code, no particular attention was given to performance aspects, and that the programs were executed in the Visual Basic development environment, that is, without creating an executable file. Therefore, the reported computing times should not be taken as benchmarks. Their sole purpose is to indicate relative differences in the computational requirements of the algorithms.

Simulation results. Simulation results were gathered with the replication/deletion approach (e.g., Law and Kelton 2000, Chap. 9). The minimum number of replications was set to 25, the maximum number to 80. When the point estimates did not satisfy a given precision after 25 replications, additional replications were performed until either the maximum relative errors of the point estimates were satisfactory, or the maximum number of replications was reached. Warmup periods and observation periods (length of a replication – warmup period) were generally set generously to obtain observations representative of steady-state behavior.

9.2 Test Results for Subassembly SA1

The definition of the base system and the test sets for subassembly SA1 are given in Tables 9.1 and 9.3. The parameter values of the three-product version of the base system are listed in Table 9.2. For test sets 2b and 2c, we reduced the total traffic intensity from 0.80 to 0.30. For systems with low total traffic intensity, the idle states are more important because the average fraction of time the manufacturing facility spends in the idle states increases when the total traffic intensity is reduced. This effect is documented in Table 9.4 for the test instances of test set 1a. For $\rho = 0.30$, with $K_1 = K_2 = K_3 = 3$, the respective values are 13.561% (I_1), 11.336% (I_2), and 8.680% (I_3).

The algorithm for subassembly SA1 was implemented with $\varepsilon_p = 10^{-4}$. The kanban configurations, number of rotations, and computing times are listed in Tables 9.5–9.12. Most reference values were obtained from the exact model. Only for systems with more than three products, the comparison was based on point estimates obtained from simulation. In the simulation experiments, additional replications were made until each point estimate had a relative error of at most 0.05% at an *overall* confidence level for each test instance of at least (approximately) 90%. The confidence level for each confidence interval of a test instance was determined according to *Bonferroni's Inequality* (e.g., Law and Kelton 2000, Section 9.7), for example, $1-(1-0.9)/20 = 99.5\%$ for a test instance with ten products and, thus, 20 individual confidence intervals $(f_1,\ldots,f_{10};\bar{y}_1,\ldots,\bar{y}_{10})$.

The estimates generated with the proposed algorithm appear to be fairly accurate over a wide range of different systems. Many relative deviations in the experiments are close to or below one percent. Unbalancing the system with respect to the average demand arrival rates and varying the required fill rates shows hardly any effect on the relative deviations (Figs. 9.2–9.5 and 9.8). When the number of products was increased, the estimates for the average inventory levels were almost identical to the exact values (Fig. 9.9).

Table 9.1. Base System for Subassembly SA1

Average processing rate of product 1, μ_1	2.00
Total traffic intensity, ρ	0.80
Traffic intensity ratio ρ_1/ρ_r	1.60
Processing rate ratio μ_1/μ_r	0.90
Setup to processing time ratio of product 1, s_1/μ_1^{-1}	2.00
Setup time ratio s_1/s_r	0.80
Required fill rates, $f_i^{\min}$, $i = 1, \dots, r$	0.95

Table 9.2. Base System with Three Products for Subassembly SA1

Product i	1	2	3
Average demand rates, λ_i^{ext}	0.66	0.56	0.46
Average processing rates, μ_i	2.00	2.11	2.22
Traffic intensities, $\rho_i = \lambda_i^{\text{ext}}/\mu_i$	0.33	0.27	0.20
Average setup times, s_i	1.00	1.125	1.25
Required fill rates, $f_i^{\min}$	0.95	0.95	0.95

Table 9.3. Test Sets for Subassembly SA1

Test Set	Modified Parameter	Range	Increment
1	Total traffic intensity, ρ	0.05–0.95	0.10
2a	Traffic intensity ratio ρ_1/ρ_r [$\rho = 0.80$, $r = 3$]	2–20	2
2b	Traffic intensity ratio ρ_1/ρ_r [$\rho = 0.30$, $r = 3$]	2–20	2
2c	Traffic intensity ratio ρ_1/ρ_r [$\rho = 0.30$, $r = 10$]	2–20	2
3	Setup to proc. time ratio prod. 1, s_1/μ_1^{-1}	0.5–5.0	0.5
4	Setup time ratio s_1/s_r	0.2–1.0	0.1
5	Required fill rates, $f_i^{\min}$	0.81–0.95	0.02
6	Number of products, r	3–10	1

Table 9.4. Test Set 1 (Subassembly SA1), Average Fraction of Time (%) of Each Idle State (Exact Values)

ρ	I_1	I_2	I_3
0.05	34.555	29.346	23.435
0.15	25.381	21.314	16.637
0.25	17.009	14.228	10.953
0.35	8.792	6.725	10.479
0.45	5.880	4.922	3.746
0.55	2.387	1.825	2.846
0.65	0.771	1.199	1.007
0.75	0.243	0.376	0.317
0.85	0.046	0.071	0.060
0.95	0.002	0.003	0.003

Table 9.5. Test Set 1 (Subassembly SA1)

ρ	K_1, K_2, K_3	Rotations	Time (sec.)
0.05	2,2,2	11.3	0.040
0.15	2,2,2	6.3	0.030
0.25	3,3,3	7.3	0.031
0.35	4,3,3	7.0	0.030
0.45	4,4,4	5.3	0.041
0.55	6,5,5	4.3	0.050
0.65	7,7,6	5.7	0.080
0.75	10,10,9	6.3	0.151
0.85	16,16,14	7.0	0.361
0.95	33,32,29	7.0	1.392

Table 9.6. Test Set 2a (Subassembly SA1)

ρ_1/ρ_3	K_1,K_2,K_3	Rotations	Time (sec.)
2	13, 12, 10	6.7	0.220
4	13, 12, 7	6.3	0.180
6	13, 12, 6	6.7	0.180
8	13, 12, 5	6.7	0.190
10	13, 12, 4	6.7	0.181
12	13, 12, 4	6.7	0.170
14	13, 12, 4	6.7	0.170
16	13, 12, 3	6.7	0.181
18	12, 12, 3	6.7	0.160
20	12, 12, 3	6.7	0.170

Table 9.7. Test Set 2b (Subassembly SA1)

ρ_1/ρ_3	K_1,K_2,K_3	Rotations	Time (sec.)
2	3, 3, 3	7.3	0.030
4	3, 3, 2	7.3	0.030
6	3, 3, 2	7.3	0.050
8	3, 3, 2	7.3	0.060
10	3, 3, 2	7.3	0.030
12	3, 3, 2	6.7	0.040
14	3, 3, 2	5.3	0.020
16	3, 3, 2	5.3	0.050
18	3, 3, 2	5.3	0.030
20	3, 3, 2	5.3	0.040

Table 9.8. Test Set 2c (Subassembly SA1)

ρ_1/ρ_{10}	$K_1,\ldots,K_{10}$	Rotations	Time (sec.)
2^s	2,2,2,2,2,2,2,2,2,2	8.7	0.230
4^s	3,3,2,2,2,2,2,2,2,2	8.7	0.270
6^s	3,3,2,2,2,2,2,2,2,2	8.6	0.261
8^s	3,3,2,2,2,2,2,2,2,2	8.6	0.251
10^s	3,3,2,2,2,2,2,2,2,2	8.6	0.261
12^s	3,3,3,2,2,2,2,2,2,1	8.6	0.291
14^s	3,3,3,2,2,2,2,2,2,1	8.6	0.280
16^s	3,3,3,2,2,2,2,2,2,1	8.6	0.261
18^s	3,3,3,2,2,2,2,2,2,1	8.6	0.250
20^s	3,3,3,2,2,2,2,2,2,1	8.6	0.260

^{s}Comparison based on simulation results.

Table 9.9. Test Set 3 (Subassembly SA1)

s_1/μ_1^{-1}	K_1,K_2,K_3	Rotations	Time (sec.)
0.5	5,5,5	6.0	0.050
1.0	7,7,6	7.0	0.090
1.5	10,10,9	7.3	0.161
2.0	13,12,11	7.0	0.230
2.5	15,15,13	6.0	0.281
3.0	18,17,16	6.0	0.390
3.5	21,20,18	5.7	0.481
4.0	24,23,20	5.3	0.580
4.5	26,25,23	4.7	0.621
5.0	29,28,25	4.0	0.661

Table 9.10. Test Set 4 (Subassembly SA1)

s_1/s_3	K_1,K_2,K_3	Rotations	Time (sec.)
0.2	32,29,26	5.0	0.891
0.3	23,22,19	5.3	0.540
0.4	19,18,16	6.0	0.421
0.5	17,16,14	6.3	0.350
0.6	15,14,13	6.7	0.300
0.7	14,13,12	6.7	0.261
0.8	13,12,11	7.0	0.241
0.9	12,12,10	7.0	0.210
1.0	11,11,10	6.7	0.190

Table 9.11. Test Set 5 (Subassembly SA1)

$f_i^{\min}$	K_1,K_2,K_3	Rotations	Time (sec.)
0.81	6,6,5	4.7	0.060
0.83	6,6,6	4.7	0.060
0.85	7,7,6	4.7	0.080
0.87	8,7,7	4.3	0.080
0.89	8,8,7	4.7	0.090
0.91	10,9,8	5.0	0.110
0.93	11,10,9	5.7	0.150
0.95	13,12,11	7.0	0.230
0.97	15,15,13	8.0	0.341
0.99	21,20,18	10.3	0.701

Table 9.12. Test Set 6 (Subassembly SA1)

r	$K_1,\dots,K_r$	Rotations	Time (sec.)
3	13,12,11	7.0	0.231
4[s]	13,13,12,11	5.8	0.371
5[s]	13,13,12,11,11	6.2	0.621
6[s]	13,13,12,12,11,10	7.0	1.001
7[s]	13,13,12,12,11,11,10	7.4	1.512
8[s]	13,13,12,12,12,11,11,10	7.6	2.083
9[s]	13,13,12,12,12,11,11,11,10	7.8	2.874
10[s]	13,13,12,12,12,11,11,11,10,10	7.9	3.545

[s] Comparison based on simulation results.

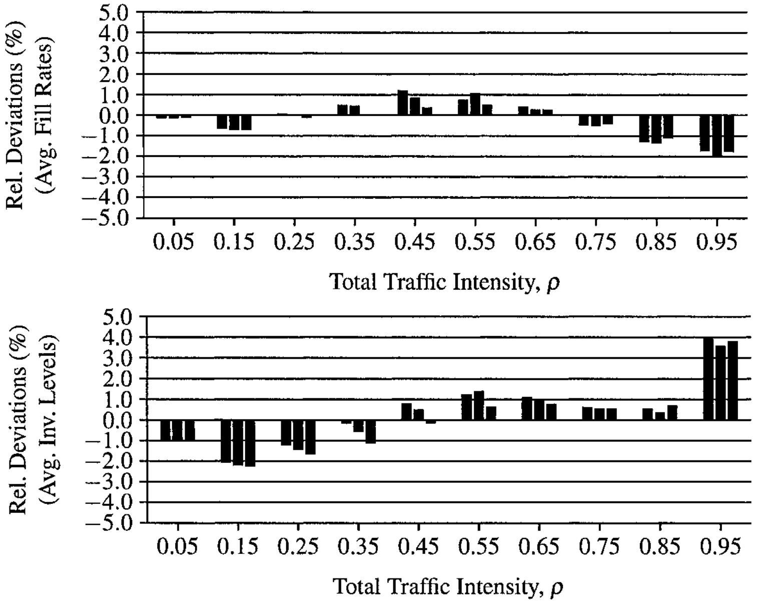


Fig. 9.1. Test set 1 (subassembly SA1); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

The estimates of the average inventory levels, however, seem to deteriorate significantly for systems with total traffic intensity close to 1 (Fig. 9.1). For the test instances with relatively large average setup times (three to five times the average container processing time) and for test instances with significantly different setup times (Figs. 9.6 and 9.7), the estimates of the average fill rates differed from the exact values by two to three percent.

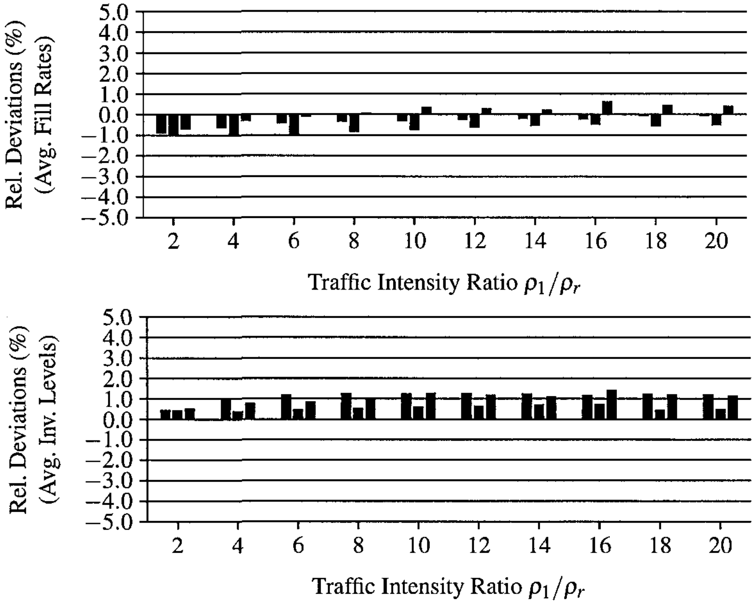


Fig. 9.2. Test set 2a (subassembly SA1); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

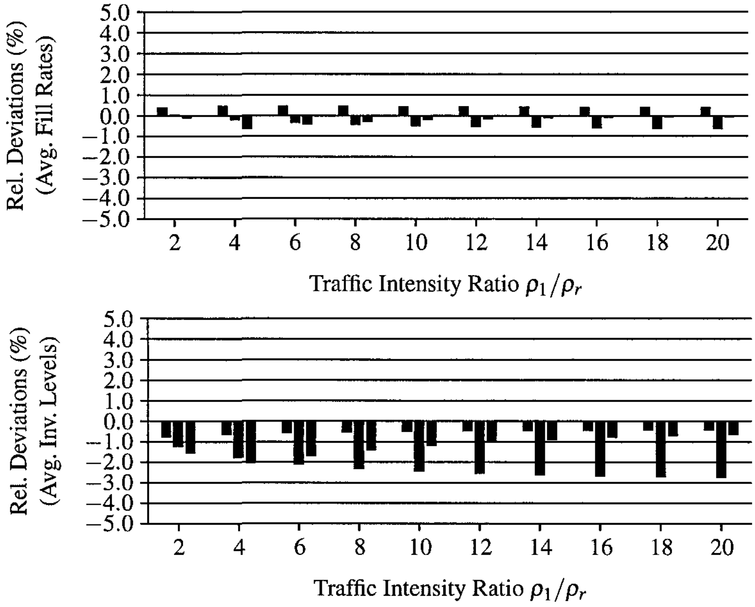


Fig. 9.3. Test set 2b (subassembly SA1); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

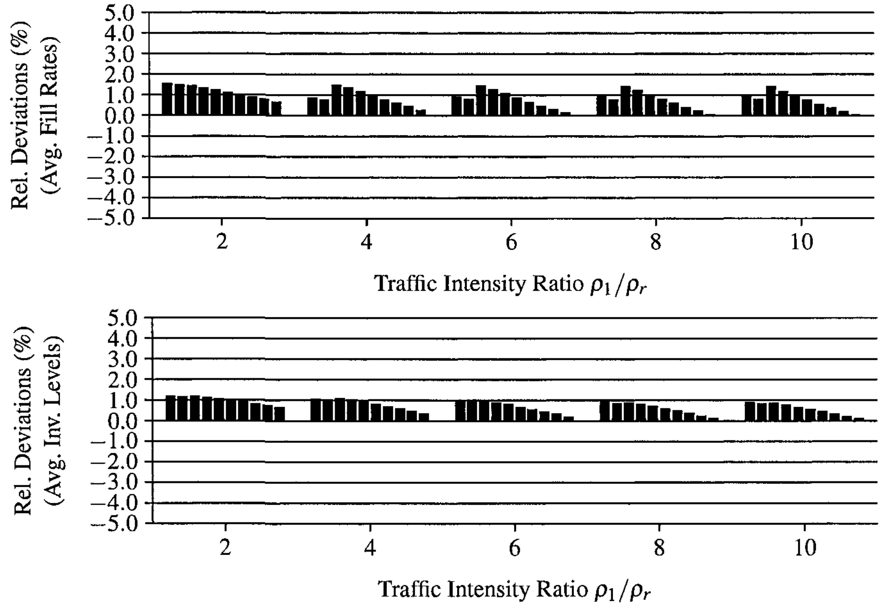


Fig. 9.4. Test set 2c (subassembly SA1), part 1; *first bar*, product 1; *second bar*, product 2; ...; *last bar*, product 10.

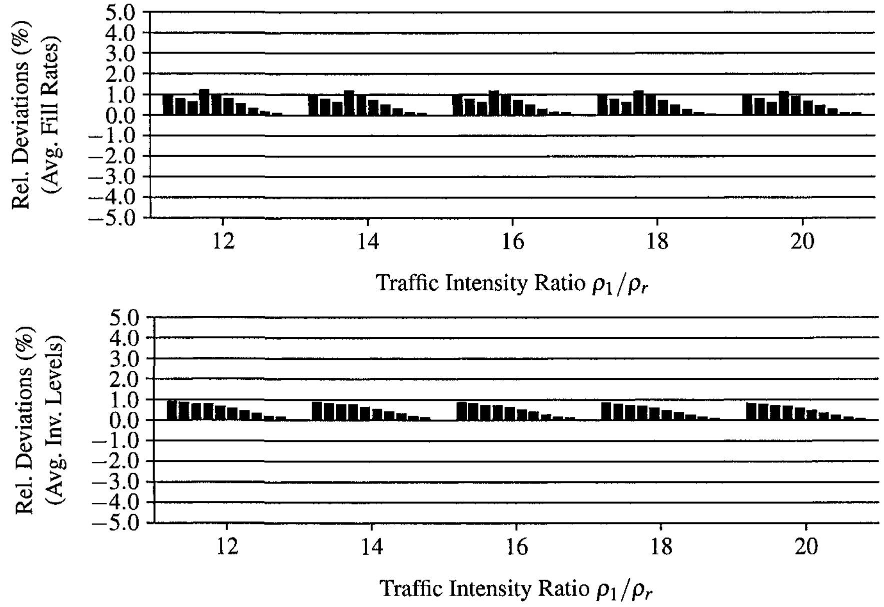


Fig. 9.5. Test set 2c (subassembly SA1), part 2; *first bar*, product 1; *second bar*, product 2; ...; *last bar*, product 10.

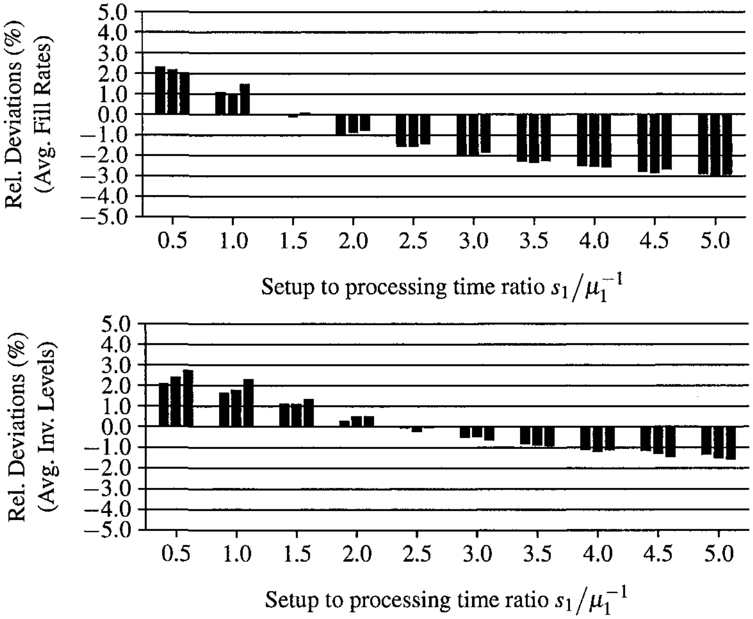


Fig. 9.6. Test set 3 (subassembly SA1); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

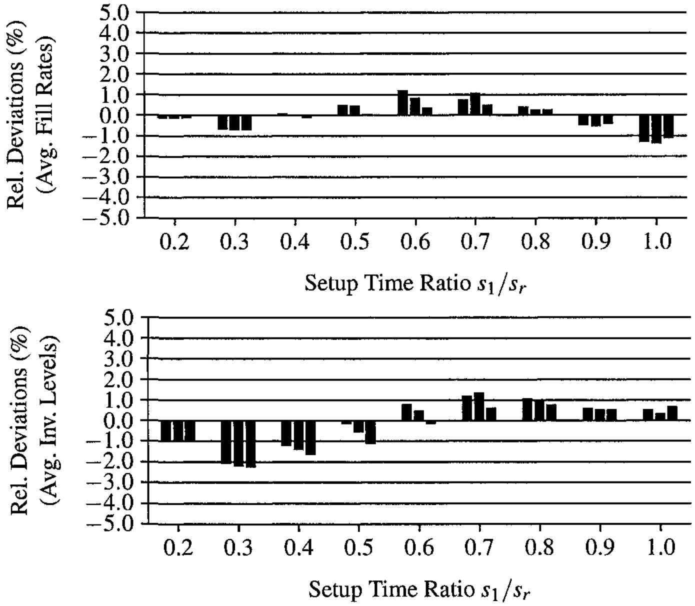


Fig. 9.7. Test set 4 (subassembly SA1); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

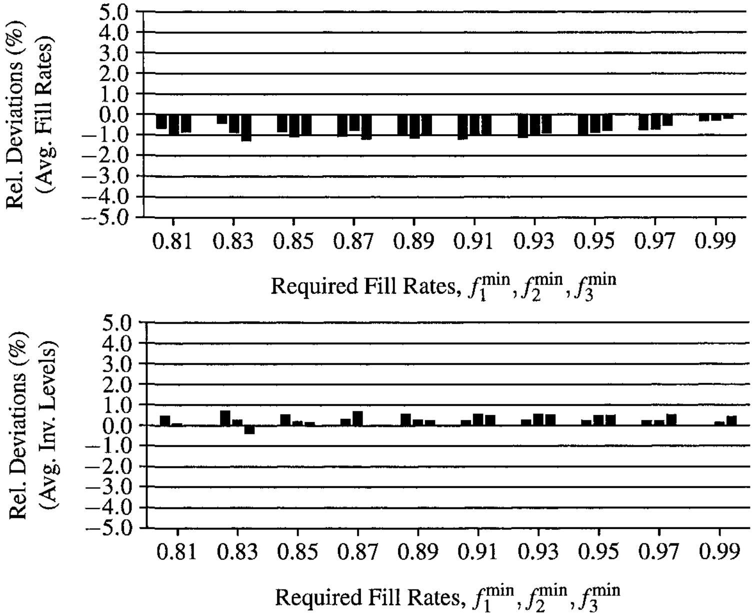


Fig. 9.8. Test set 5 (subassembly SA1); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

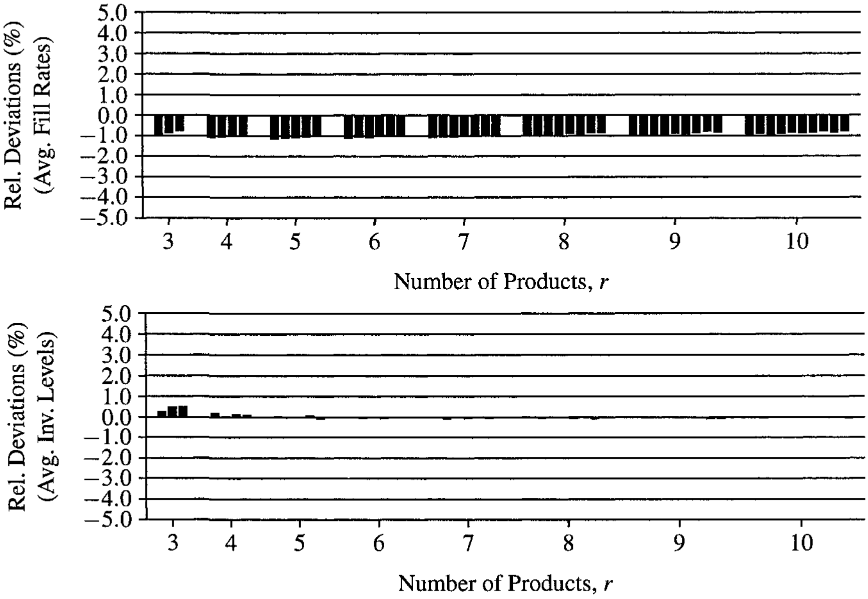


Fig. 9.9. Test set 6 (subassembly SA1); *first bar*, product 1; *second bar*, product 2; ...; *last bar*, product *r*.

9.3 Test Results for Subassembly SA2

In the following two sections, we give the results of nine test sets without backorders and two test sets with backorders for subassembly SA2. The definition of the base system is given in Table 9.13. The parameter values of the three-product version of the base system are listed in Table 9.15. The algorithm for subassembly SA2 was implemented with $\varepsilon_p = 10^{-4}$.

9.3.1 Tests without Backorders

The test sets for subassembly SA2 without backorders are listed in Table 9.14. For test sets 1c and 2b, we reduced the total traffic intensity in stage 2 from 0.80 to 0.30. For test set 2c, the initial values of the traffic intensities in stage 1 $\left(\rho_i^{(1)} = \lambda_i^{\text{ext}}/\mu_i^{(1)}\right)$ were set to 0.50. While $\rho_2^{(1)}$ was kept constant at 0.50, the difference between $\rho_1^{(1)}$ and $\rho_3^{(1)}$ was successively increased in increments of 0.10. For example, for test instance 2 of test set 2c, the traffic intensities in stage 1 were $\rho_1^{(1)} = 0.55$, $\rho_2^{(1)} = 0.50$, and $\rho_3^{(1)} = 0.45$, for test instance 10, the respective values were $\rho_1^{(1)} = 0.95$, $\rho_2^{(1)} = 0.50$, and $\rho_3^{(1)} = 0.05$.

The kanban configurations, number of rotations, and computing times are listed in Tables 9.17–9.25. A number of reference values were obtained from the exact model. For larger systems, the comparison was based on point estimates obtained from simulation. In the simulation experiments, additional replications were made until each point estimate had a relative error of at most 0.1% at an individual confidence level of (approximately) 99%. For some performance measures, the maximum relative error still exceeded this value after 80 replications. These cases are listed in Table 9.16 along with the respective maximum relative errors.

In test set 1a, the total traffic intensity of stage 2 was varied from 0.10 to 0.90. In test sets 1b $\left(\rho^{(2)} = 0.80\right)$ and 1c $\left(\rho^{(2)} = 0.30\right)$, the traffic intensities for the products in stage 1 were changed from 0.10 to 1.00. The results indicate that the approximation quality is only moderately sensitive to changes in the (total) traffic intensity (Figs. 9.10–9.12). In test sets 1a and 1b, the approximation errors of the average inventory levels in stage 1 increase for higher traffic intensities. In test sets 1a and 1c, the approximation errors of the average inventory levels in stage 2 peak for traffic intensities around 0.40–0.50 and 0.50–0.60, respectively.

In test sets 2a–2c, the test instances were increasingly unbalanced with respect to the (per product) traffic intensities in stage 2 (test sets 2a, 2b) and in stage 1 (test set 2c), respectively. The results suggest that the approximation is insensitive to

Table 9.13. Base System for Subassembly SA2

Average processing rate of product 1 in stage 2, $\mu_1^{(2)}$	2.00
Total traffic intensity stage 2, $\rho^{(2)}$	0.80
Traffic intensity ratio $\rho_1^{(2)}/\rho_r^{(2)}$	1.60
Processing rate ratio $\mu_1^{(2)}/\mu_r^{(2)}$	0.90
Setup to processing time ratio of product 1 in stage 2, $s_1^{(2)}\mu_1^{(2)}$	2.00
Setup time ratio $s_1^{(2)}/s_r^{(2)}$	0.80
Traffic intensities in stage 1, $\rho_i^{(1)}$, $i = 1, \ldots, r$	0.80
Required fill rates, $f_i^{\min}$, $i = 1, \ldots, r$	0.95

Table 9.14. Test Sets for Subassembly SA2 without Backorders

Test Set	Modified Parameter	Range	Increm.
1a	Total traffic intensity stage 2, $\rho^{(2)}$	0.10–0.90	0.10
1b	Traffic intensities stage 1, $\rho_i^{(1)}$, $i = 1, \ldots, r$ $[\rho^{(2)} = 0.80]$	0.10–1.00	0.10
1c	Traffic intensities stage 1, $\rho_i^{(1)}$, $i = 1, \ldots, r$ $[\rho^{(2)} = 0.30]$	0.10–1.00	0.10
2a	Traffic intensity ratio $\rho_1^{(2)}/\rho_r^{(2)}$ $[\rho^{(2)} = 0.80]$	2–20	2
2b	Traffic intensity ratio $\rho_1^{(2)}/\rho_r^{(2)}$ $[\rho_i^{(1)} = 0.30, i = 1, \ldots, r;\ \rho^{(2)} = 0.30]$	2–20	2
2c	Spread of traffic intensities stage 1 $\rho_1^{(1)} - \rho_r^{(1)}$ $[\rho_2^{(1)} = 0.50, \rho_1^{(1)} - \rho_2^{(1)} = \rho_2^{(1)} - \rho_3^{(1)};$ $\rho^{(2)} = 0.80]$	0.0–0.9	0.1
3	Setup to proc. time ratio product 1, $s_1^{(2)}\mu_1^{(2)}$	0.5–4.0	0.5
4	Setup time ratio $s_1^{(2)}/s_r^{(2)}$	0.3–1.0	0.1
5	Number of products, r	3–10	1

Table 9.15. Base System with Three Products for Subassembly SA2

Product i	1	2	3
Average demand rates, λ_i^{ext}	0.66	0.56	0.46
Average processing rates stage 1, $\mu_i^{(1)}$	0.82	0.70	0.57
Average processing rates stage 2, $\mu_i^{(2)}$	2.00	2.11	2.22
Traffic intensities, $\rho_i^{(2)} = \lambda_i^{\text{ext}}/\mu_i^{(2)}$	0.33	0.27	0.20
Average setup times, $s_i^{(2)}$	1.00	1.125	1.25
Required fill rates, $f_i^{\min}$	0.95	0.95	0.95

Table 9.16. Test Sets for Subassembly SA2 without Backorders, Point Estimates with Maximum Relative Error > 0.1%

Test Set	Test Instance	Performance Measures (Maximum Relative Error)
1b	$\rho_i^{(1)} = 1.00$	$\bar{y}_1^{(1)}(0.102\%)$, $\bar{y}_2^{(1)}(0.123\%)$, $\bar{y}_3^{(1)}(0.125\%)$
1c	$\rho_i^{(1)} = 0.90$	$\bar{y}_2^{(1)}(0.127\%)$, $\bar{y}_3^{(1)}(0.123\%)$
1c	$\rho_i^{(1)} = 1.00$	$\bar{y}_1^{(1)}(0.213\%)$, $\bar{y}_2^{(1)}(0.261\%)$, $\bar{y}_3^{(1)}(0.276\%)$, $\bar{y}_2^{(2)}(0.111\%)$, $\bar{y}_3^{(2)}(0.112\%)$
2a	$\rho_1^{(2)}/\rho_r^{(2)} = 12$	$\bar{y}_3^{(1)}(0.111\%)$
2a	$\rho_1^{(2)}/\rho_r^{(2)} = 14$	$\bar{y}_3^{(1)}(0.104\%)$
2a	$\rho_1^{(2)}/\rho_r^{(2)} = 16$	$\bar{y}_3^{(1)}(0.129\%)$
2a	$\rho_1^{(2)}/\rho_r^{(2)} = 18$	$\bar{y}_3^{(1)}(0.127\%)$
2a	$\rho_1^{(2)}/\rho_r^{(2)} = 20$	$\bar{y}_3^{(1)}(0.170\%)$

Table 9.17. Test Set 1a (Subassembly SA2 without Backorders)

$\rho^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0.10^e	3,3,3	5,5,5	7.7	1.782
0.20^e	4,3,3	4,5,5	6.7	0.902
0.30^e	5,5,4	4,4,5	7.3	0.872
0.40^e	5,5,4	5,5,6	5.7	0.831
0.50	6,5,5	6,7,6	5.7	1.072
0.60	6,7,6	9,7,7	6.0	1.703
0.70	8,8,7	11,10,9	7.0	3.575
0.80	11,11,11	15,14,12	7.7	9.804
0.90	18,19,16	23,22,21	8.0	44.985

^{e}Comparison based on exact values.

changes of the traffic intensity ratio in stage 2 and moderately sensitive to drastically different average processing rates for the products in stage 1 (Figs. 9.13–9.15). Note the large difference between the average processing rates for products 1 and 3 in the last instance of test set 2c: $\mu_1^{(1)} = 0.69$ and $\mu_3^{(1)} = 9.12$.

In test sets 3 and 4, the setup times were changed. In test set 3, they were simultaneously increased for all products, in test set 4, the ratio of the setup times for product 1 and 3 was varied, where $s_2^{(2)} = \frac{1}{2}\left(s_3^{(2)} - s_1^{(2)}\right)$. The results in Figures 9.16 and 9.17 indicate a high sensitivity of the approximation to changes in the setup times. The errors are most pronounced for estimates of the average inventory levels in stage 2 when the average setup times are smaller than the average container processing times and when the average setup times are about three times the average container processing times, or larger (Fig. 9.16).

Finally, we systematically increased the number of products from three to ten (test set 5). The results show no significant changes in the approximation quality for systems with more products (Fig. 9.18).

Table 9.18. Test Set 1b (Subassembly SA2 without Backorders)

$\rho_i^{(1)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0.10	3,4,4	13,12,11	7.0	1.272
0.20	4,6,5	14,12,11	7.0	1.823
0.30	6,7,6	13,12,11	7.0	2.373
0.40	7,7,6	13,13,12	7.0	2.904
0.50	8,8,8	13,13,11	7.0	3.585
0.60	10,8,8	13,14,12	7.0	4.987
0.70	9,10,9	15,13,12	7.3	6.609
0.80	11,11,11	15,14,12	7.7	9.824
0.90	14,12,12	15,16,14	8.0	16.294
1.00	18,18,16	20,19,18	9.7	43.883

Table 9.19. Test Set 1c (Subassembly SA2 without Backorders)

$\rho_i^{(1)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0.10[e]	1,1,1	4,3,3	7.7	0.080
0.20[e]	2,2,1	3,3,4	7.7	0.110
0.30[e]	2,2,2	3,3,3	7.3	0.120
0.40[e]	3,2,2	3,4,3	7.3	0.201
0.50[e]	2,3,2	4,3,4	7.7	0.241
0.60[e]	3,3,2	4,4,5	7.3	0.381
0.70[e]	4,3,3	4,5,4	7.3	0.541
0.80[e]	5,5,4	4,4,5	7.3	0.892
0.90	6,5,5	6,7,7	7.3	2.484
1.00	11,10,10	10,11,11	7.7	15.672

[e] Comparison based on exact values.

Table 9.20. Test Set 2a (Subassembly SA2 without Backorders)

$\rho_1^{(2)}/\rho_r^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
2	12,10,10	14,15,11	7.3	8.933
4	11,11,7	16,14,9	7.3	8.503
6	10,13,9	17,13,6	7.3	8.913
8	10,11,6	16,14,6	7.3	7.631
10	10,11,6	16,14,5	7.3	7.621
12	12,12,5	14,13,5	7.3	7.661
14	11,12,5	14,13,5	7.3	7.230
16	10,11,6	15,13,4	7.3	6.830
18	10,11,5	15,13,4	7.3	6.740
20	10,11,5	15,13,4	7.3	6.770

Table 9.21. Test Set 2b (Subassembly SA2 without Backorders)

$\rho_1^{(2)}/\rho_r^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
2[e]	2,2,2	4,3,3	7.3	0.150
4[e]	2,2,2	4,3,2	7.3	0.130
6[e]	1,2,2	5,3,2	7.7	0.151
8[e]	1,2,2	4,3,2	7.3	0.130
10[e]	1,2,1	4,3,2	7.3	0.120
12[e]	1,2,1	4,3,2	7.3	0.130
14[e]	1,2,1	4,3,2	7.3	0.130
16[e]	1,2,1	4,3,2	7.3	0.131
18[e]	1,2,1	4,3,2	7.3	0.140
20[e]	1,2,1	4,3,2	7.3	0.160

[e] Comparison based on exact values.

Table 9.22. Test Set 2c (Subassembly SA2 without Backorders)

$\rho_1^{(1)} - \rho_r^{(1)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0.00	8,8,8	13,13,11	7.0	3.705
0.10	9,7,8	13,14,11	6.7	3.905
0.20	10,7,6	13,14,12	7.0	4.086
0.30	10,8,7	13,13,11	7.0	4.196
0.40	10,10,5	14,12,12	7.0	4.747
0.50	10,7,6	15,14,11	7.3	5.147
0.60	12,9,5	14,12,11	7.0	5.478
0.70	11,7,5	16,14,11	7.7	6.459
0.80	14,6,3	15,15,12	7.7	8.312
0.90	15,6,3	17,15,11	7.7	10.365

Table 9.23. Test Set 3 (Subassembly SA2 without Backorders)

$s_1^{(2)}\mu_1^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0.5	6,5,4	6,7,7	6.7	1.172
1.0	6,7,8	10,9,7	8.3	2.784
1.5	8,9,8	13,11,11	8.3	5.558
2.0	11,11,11	15,14,12	7.7	9.964
2.5	14,14,12	17,16,15	7.0	16.123
3.0	14,15,14	22,20,18	7.3	27.019
3.5	16,17,14	25,22,22	7.0	38.796
4.0	20,19,16	26,25,24	6.3	55.419

Table 9.24. Test Set 4 (Subassembly SA2 without Backorders)

$s_1^{(2)}/s_r^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0.3	18,17,18	27,25,20	6.7	49.031
0.4	16,15,15	21,20,17	6.7	27.339
0.5	14,14,13	19,17,15	7.0	18.517
0.6	13,12,12	17,16,14	7.3	14.180
0.7	12,12,11	15,14,13	7.3	10.715
0.8	11,11,11	15,14,12	7.7	9.804
0.9	12,10,11	13,14,11	7.3	8.502
1.0	9,9,10	15,14,11	7.7	7.881

Table 9.25. Test Set 5 (Subassembly SA2 without Backorders)

r	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$
3	11, 11, 11	15, 14, 12
4	11, 13, 11, 10	15, 13, 13, 12
5	11, 12, 10, 12, 10	16, 14, 15, 12, 12
6	12, 12, 12, 10, 11, 11	15, 14, 13, 14, 12, 11
7	13, 11, 11, 10, 12,9, 11	14, 15, 14, 15, 12, 14, 11
8	11, 11, 11, 10, 12, 11,9, 10	16, 15, 14, 15, 12, 12, 13, 11
9	11, 11, 13, 12, 10, 11, 11, 11, 10	16, 15, 13, 13, 14, 13, 12, 11, 11
10	11, 11, 11, 12, 10, 12, 11, 12, 11,8	16, 15, 14, 13, 15, 12, 12, 11, 11, 14

r	Rotations	Time (sec.)
3	7.7	9.794
4	7.3	17.064
5	7.2	29.363
6	5.8	39.356
7	3.9	51.283
8	5.1	71.833
9	6.2	105.001
10	5.5	131.770

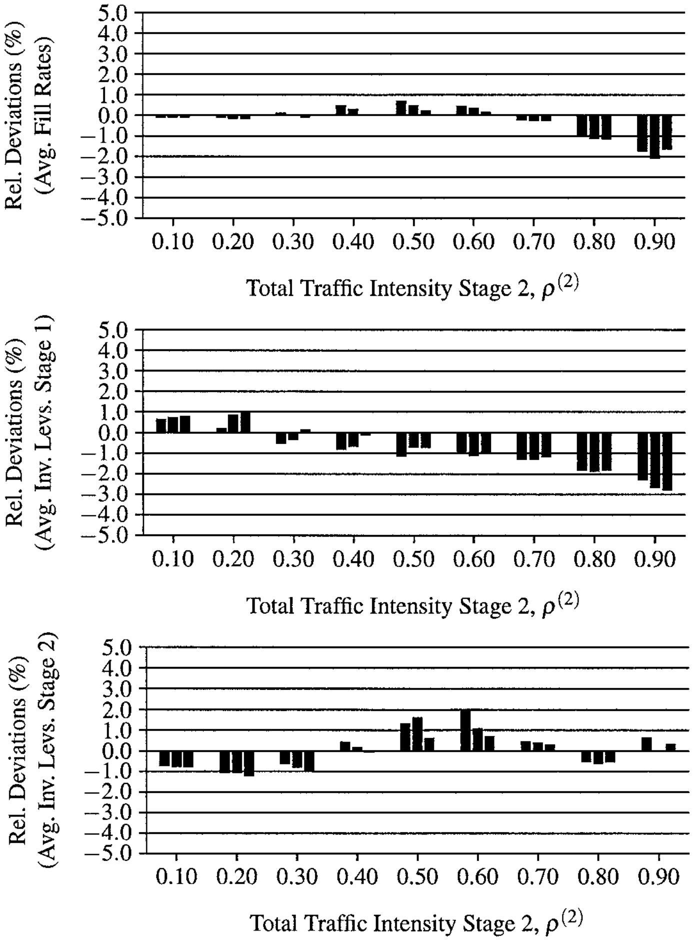


Fig. 9.10. Test set 1a (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

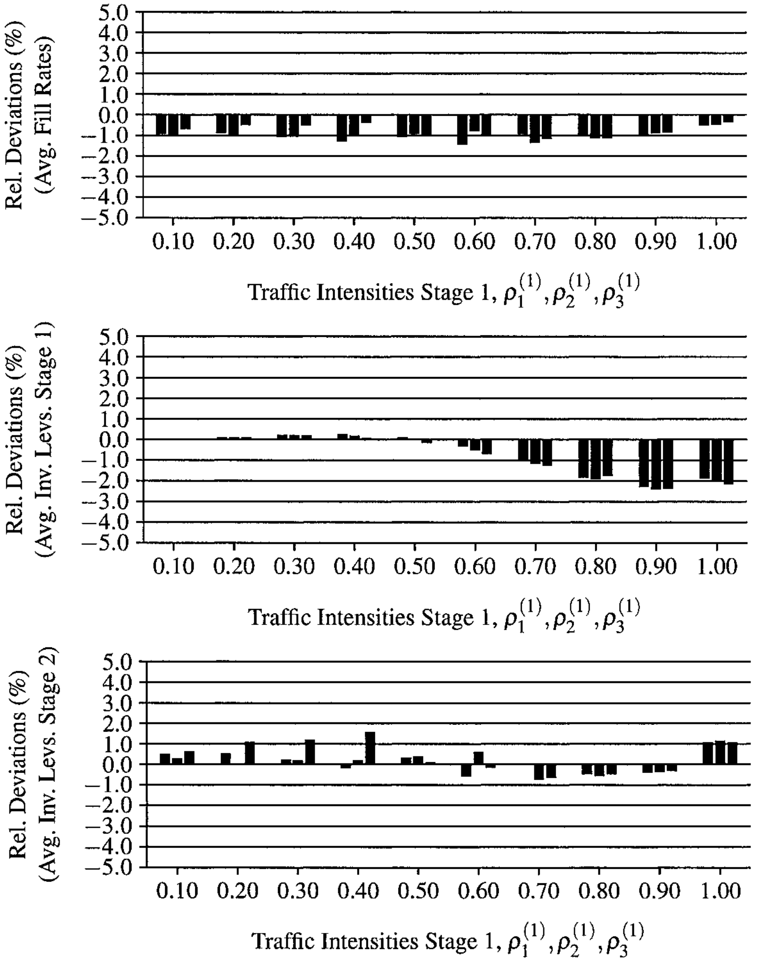


Fig. 9.11. Test set 1b (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

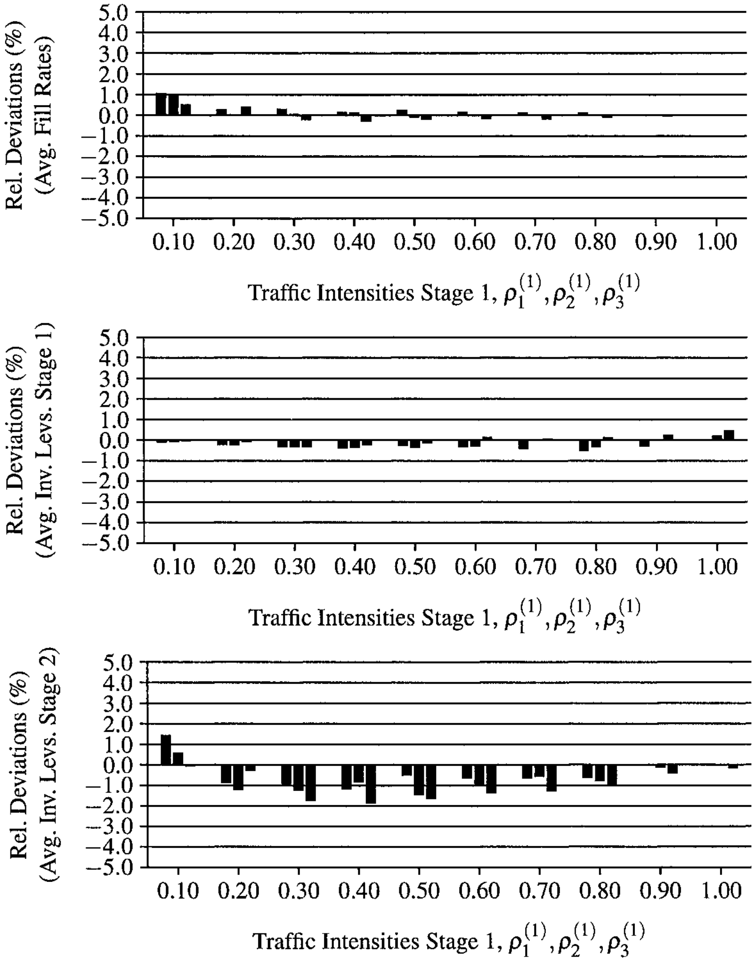


Fig. 9.12. Test set 1c (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

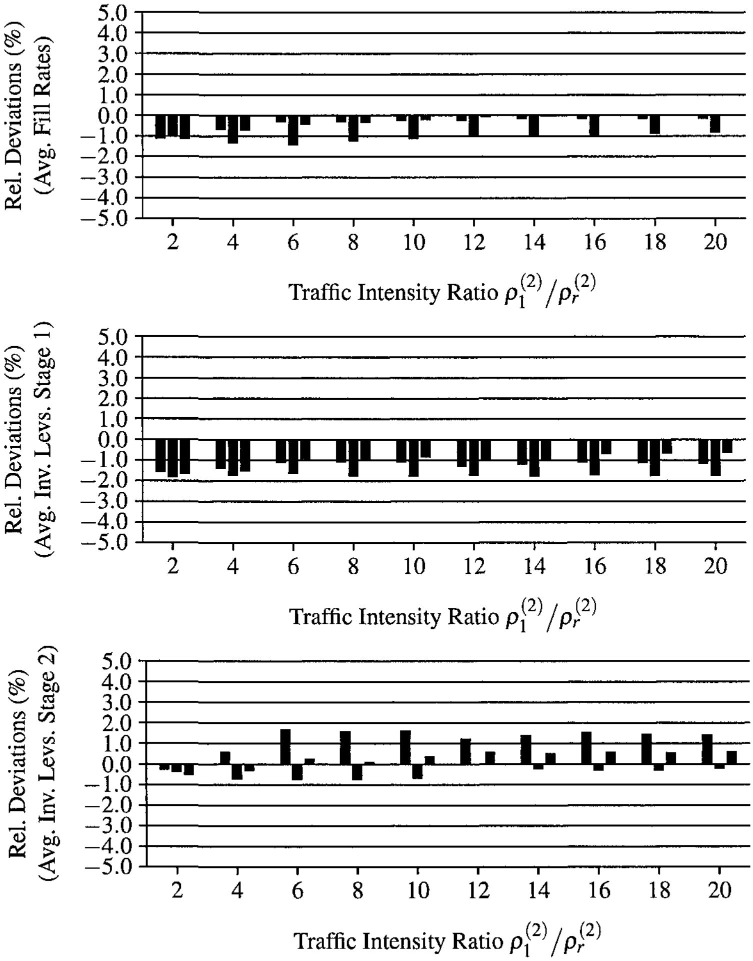


Fig. 9.13. Test set 2a (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

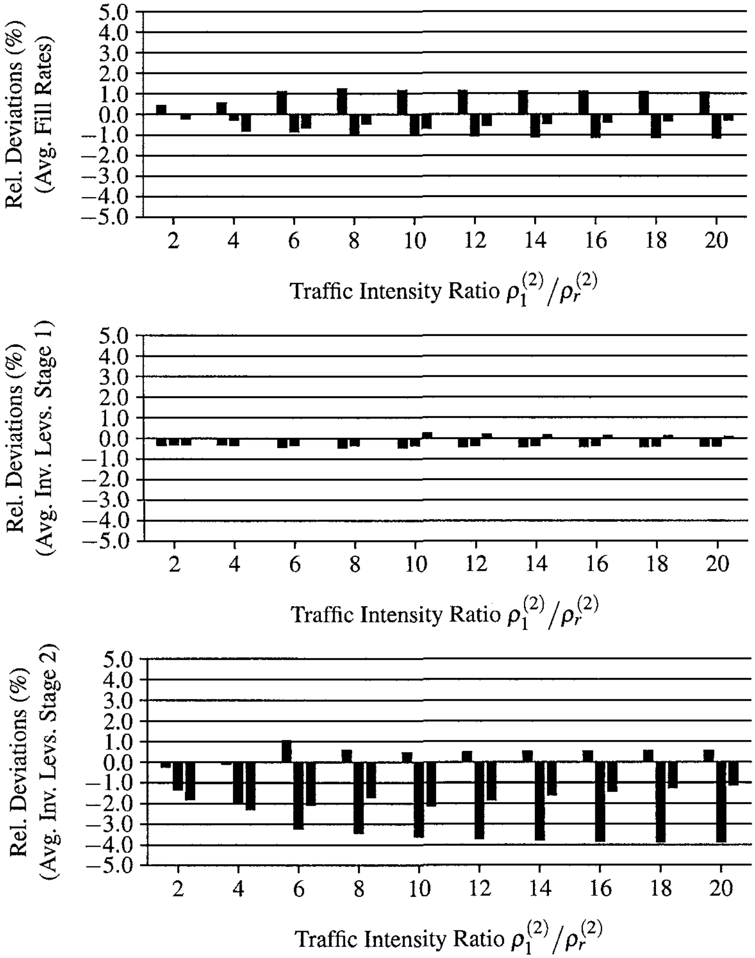


Fig. 9.14. Test set 2b (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

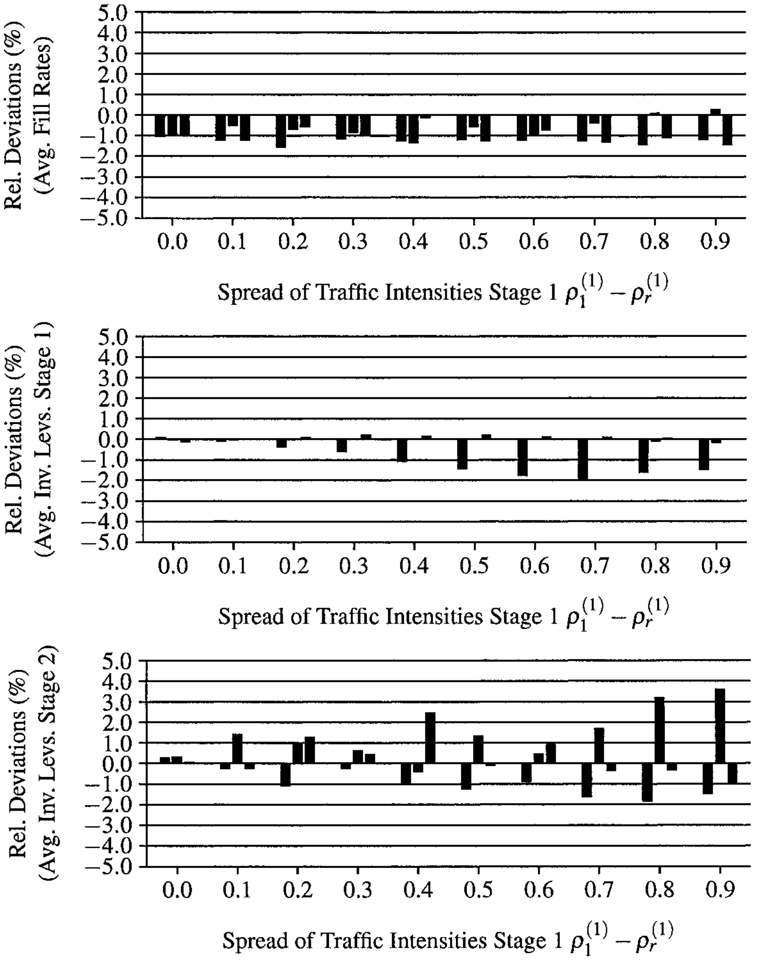


Fig. 9.15. Test set 2c (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

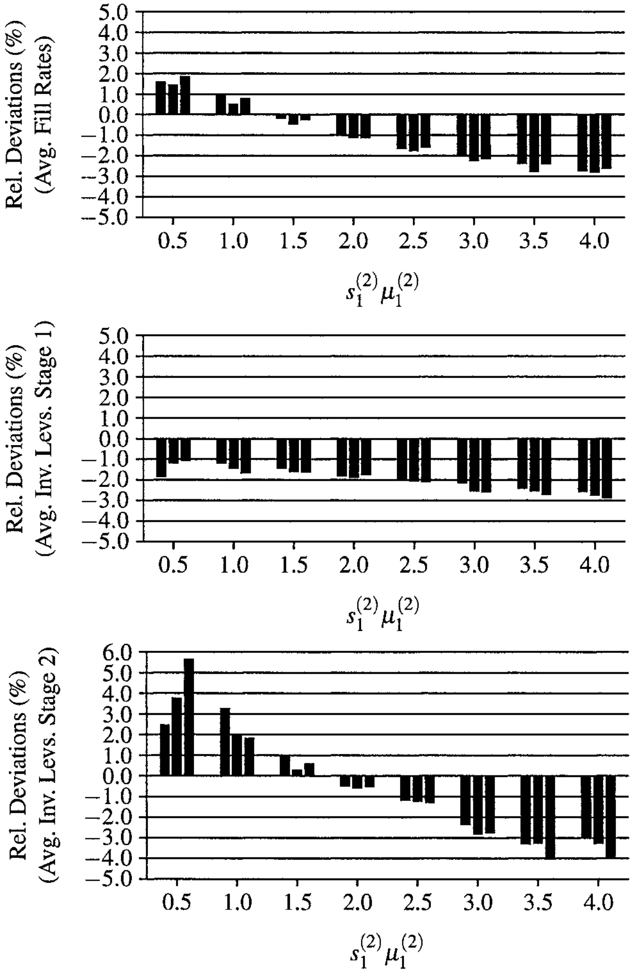


Fig. 9.16. Test set 3 (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

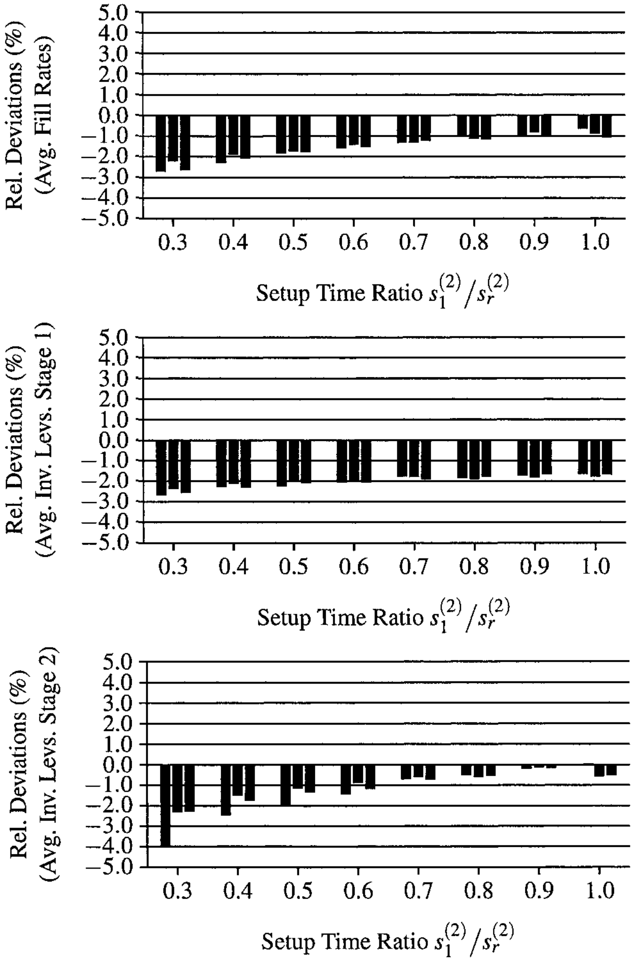


Fig. 9.17. Test set 4 (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

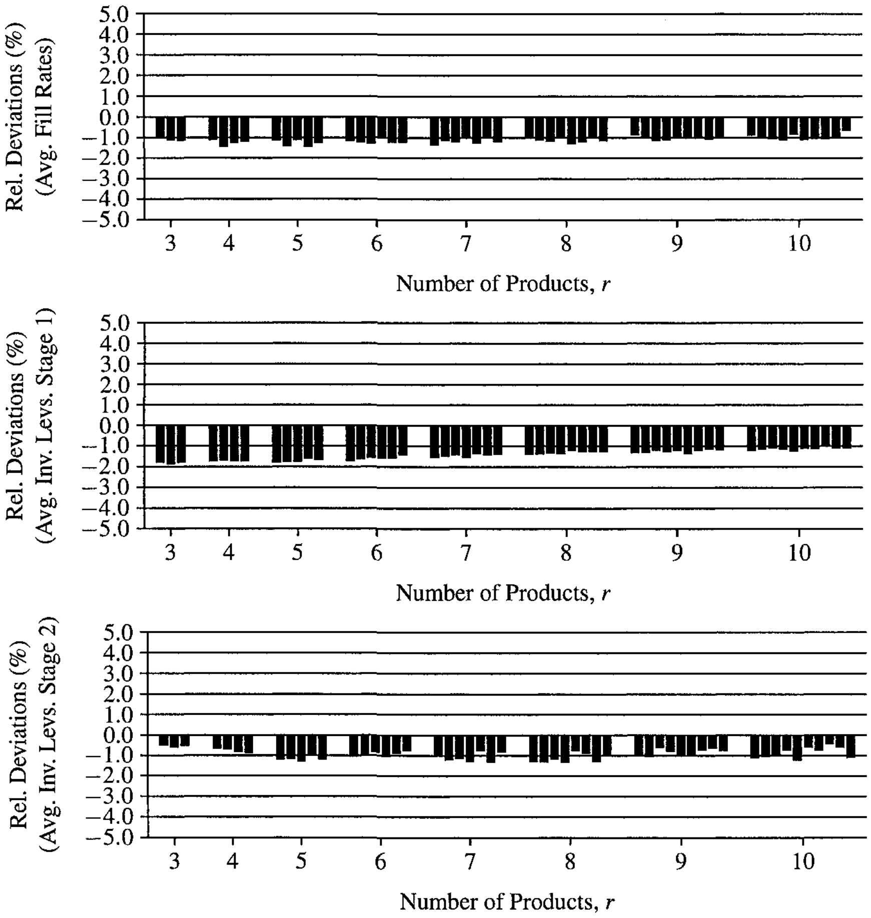


Fig. 9.18. Test set 5 (subassembly SA2 without backorders); *first bar*, product 1; *second bar*, product 2; ... ; *last bar*, product *r*.

9.3.2 Tests with Backorders

The test sets for subassembly SA2 with backorders are given in Table 9.26. The kanban configurations, number of rotations, and computing times are listed in Tables 9.27 and 9.28. All reference values, except for the first two instances of the first test set, were obtained from simulation. In the simulation experiments for test set 1, additional replications were made until each point estimate for the average fill rates, the average inventory levels (stage 1 and stage 2), and the average fractions of served demand had a relative error of at most 0.1% at an individual confidence level of (approximately) 99%. The maximum relative error of the point estimates for the average backorder levels was set to 0.5%, the maximum relative error of the point estimates for the average waiting time of backordered demand was set to 0.2%. In the simulation experiments for test set 2, additional replications were made until each point estimate had a relative error of at most 0.2% (average backorder levels: 0.35%) at an individual confidence level of (approximately) 99%. Only for the last test instance, the maximum relative errors of the estimates for the average inventory levels in stage 2 for products 1 and 2 still exceeded this value after 80 replications $\left(\bar{y}_1^{(2)}\text{: } 0.205\%; \bar{y}_2^{(2)}\text{: } 0.205\%\right)$.

First, we ran test set 1a from the last section (Table 9.14) with $B_i^{\max,2} = 1$ for all $i = 1, \ldots, r$ (test set 1). The results are given in Figure 9.19 for the average fill rates and average inventory levels and in Figure 9.20 for the average fractions of served demand, the average backorder levels, and the average waiting times of backordered demand.

For the first four performance measures, the results are almost identical to the ones for $B_i^{\max,2} = 0$ (Fig. 9.10): only for total traffic intensities above 0.70, the relative errors appear to increase systematically (except for the average inventory levels in stage 2). The relative errors, however, are still very moderate, even for $\rho^{(2)} = 0.90$. This last observation does not apply to the relative errors of the average backorder

Table 9.26. Test Sets for Subassembly SA2 with Backorders

Test Set	Modified Parameter	Range	Increment
1	Total traffic intensity stage 2, $\rho^{(2)}$, with $B_1^{\max,2} = B_2^{\max,2} = B_3^{\max,2} = 1$	0.10–0.90	0.10
2	Maximum number of backorders, $B_1^{\max,2}, B_2^{\max,2}, B_3^{\max,2}$	0–18	2

Table 9.27. Test Set 1 (Subassembly SA2 with Backorders)

$\rho^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{B}^{\max,2}$	Rotations	Time (sec.)
0.10[e]	3,3,3	5,5,5	1,1,1	9.3	2.654
0.20[e]	4,3,3	4,5,5	1,1,1	6.7	1.342
0.30	5,5,4	4,4,5	1,1,1	8.0	1.322
0.40	5,5,4	5,5,6	1,1,1	7.7	1.301
0.50	6,5,5	6,7,6	1,1,1	6.7	1.493
0.60	6,7,6	9,7,7	1,1,1	7.3	2.413
0.70	8,8,7	11,10,9	1,1,1	9.0	4.737
0.80	11,11,11	15,14,12	1,1,1	9.7	12.448
0.90	18,19,16	23,22,21	1,1,1	10.3	53.767

[e] Comparison based on exact values.

Table 9.28. Test Set 2 (Subassembly SA2 with Backorders)

$\vec{B}^{\max,2}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Rotations	Time (sec.)
0,0,0	7,7,7	9,9,9	6.3	2.163
2,2,2	7,7,7	9,9,9	8.0	3.565
4,4,4	7,7,7	9,9,9	8.7	5.218
6,6,6	7,7,7	9,9,9	8.7	7.290
8,8,8	7,7,7	9,9,9	9.0	10.265
10,10,10	7,7,7	9,9,9	9.3	13.800
12,12,12	7,7,7	9,9,9	9.3	18.016
14,14,14	7,7,7	9,9,9	9.3	22.773
16,16,16	7,7,7	9,9,9	9.3	28.390
18,18,18	7,7,7	9,9,9	9.7	35.001

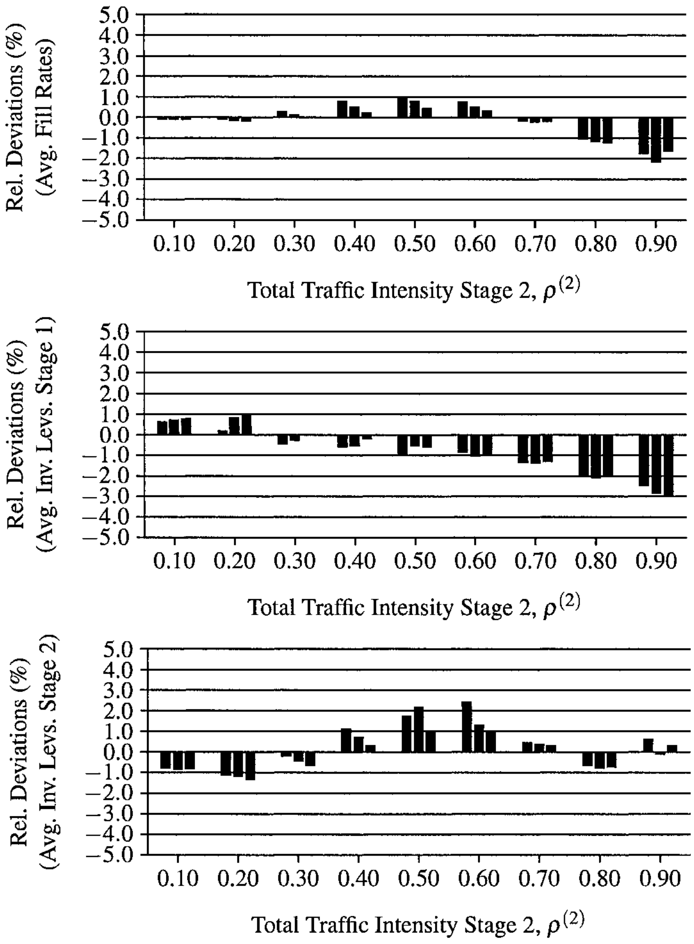


Fig. 9.19. Test set 1 (subassembly SA2 with backorders), part 1; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

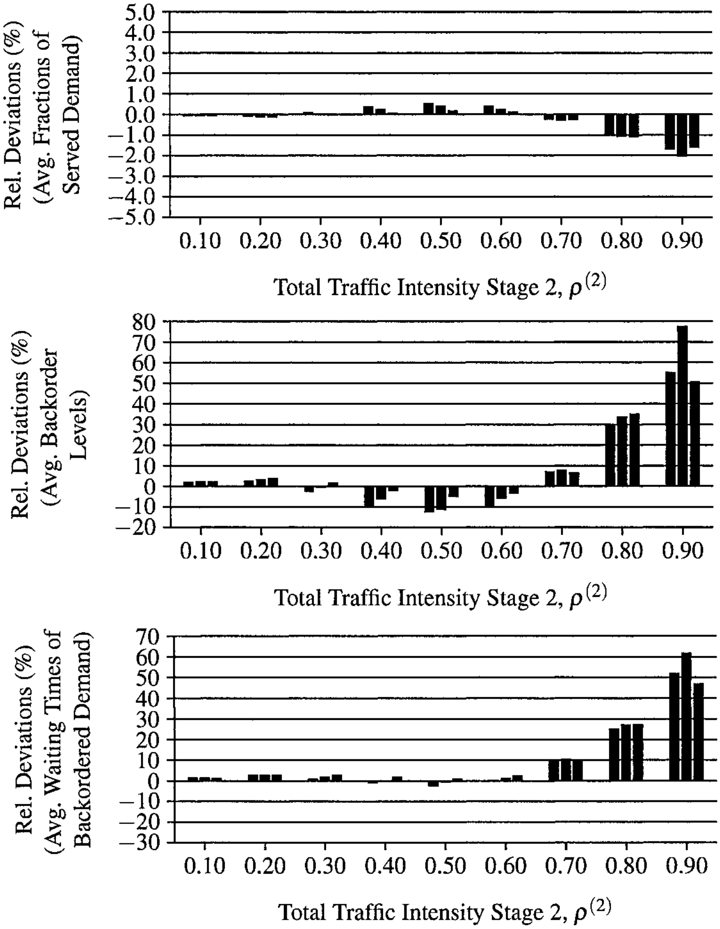


Fig. 9.20. Test set 1 (subassembly SA2 with backorders), part 2; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

levels and the average waiting times of backordered demand (Fig. 9.20, middle and bottom chart). Note that the vertical axis is scaled differently in the bar charts for these two performance measures so that the relative errors are even greater than they may appear. In Figures 9.21 and 9.22, the values of these two performance measures are plotted for product 1 for each test instance (left axis). The (absolute) deviation of the approximations from the reference values is given twice in this diagram: (1) indirectly by the gap between the data points and (2) directly by the white bars (right axis). Note that the right axis in Figure 9.22 is scaled differently than the left axis.

We also investigated if increasing the maximum number of backorders has any effect on the accuracy of the approximation (test set 2). For that purpose, we used a completely balanced system with three products and $\rho^{(2)} = 0.80$, $\rho_i^{(1)} = 0.80$, and $s_i^{(2)}\mu_i^{(2)} = 2$, $i = 1,2,3$ $\left(\mu_i^{(2)} = 2,\ \lambda_i^{\text{ext}} = 0.53,\ s_i^{(2)} = 1, \text{ and } \mu_i^{(1)} = 0.67\right)$. We set $K_i^{(1)} = 7$ and $K_i^{(2)} = 9$, $i = 1,2,3$, to yield average fill rates of about 0.90 for $B_i^{\max,2} = 0$. Then $B_1^{\max,2}$, $B_2^{\max,2}$, and $B_3^{\max,2}$ were increased simultaneously from 0 to 18 in increments of 2. The results are given in Figures 9.23 and 9.24.

In this example, increasing the maximum number of backorders has no significant effect on the relative approximation errors of the average inventory levels in stage 1 and the average fractions of served demand. The relative errors of the average fill rates and the average inventory levels in stage 2 increase continuously. The relative errors of the average backorder levels and the average waiting times of backordered demand decrease initially, but they rise again, at different rates, when the maximum number of backorders is further increased.

In Figures 9.25 and 9.26, the simulation and approximation results for these performance measures are plotted for product 1 (left axis) along with the (absolute) deviations (white bars, right axis). Note that the right axis is scaled differently than the left.

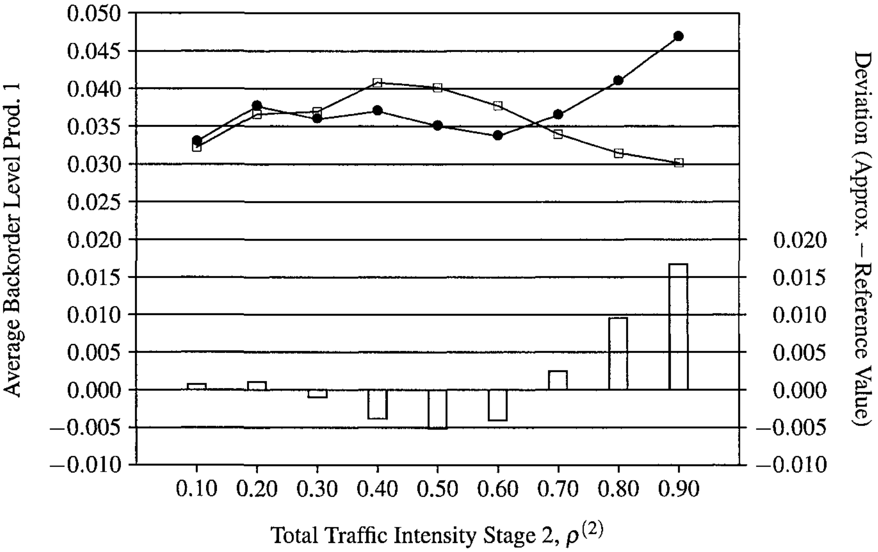


Fig. 9.21. Test set 1 (subassembly SA2 with backorders), average number of backorders for product 1; ● = approx., □ = exact value (0.10, 0.20) or simulation (0.30–0.90) (left axis); *white bars*, (absolute) deviation (right axis).

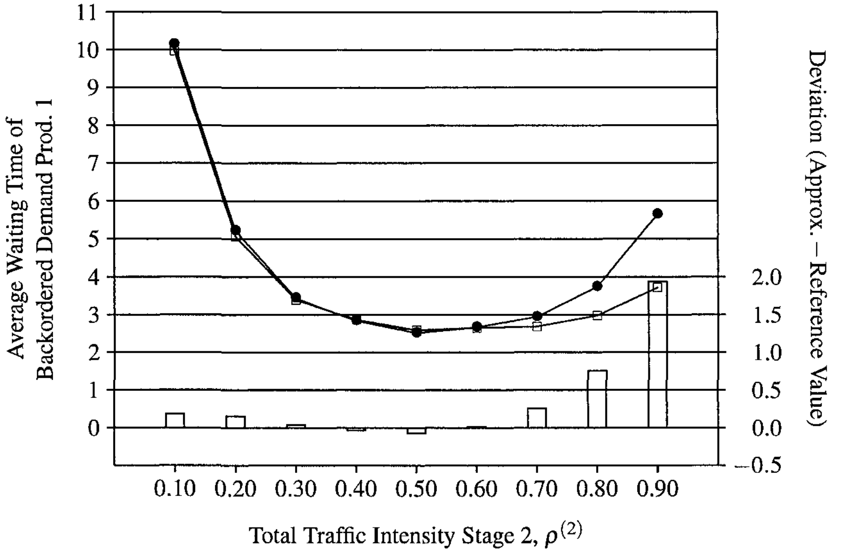


Fig. 9.22. Test set 1 (subassembly SA2 with backorders), average waiting time of backordered demand for product 1; ● = approx., □ = exact value (0.10, 0.20) or simulation (0.30–0.90) (left axis); *white bars*, (absolute) deviation (right axis).

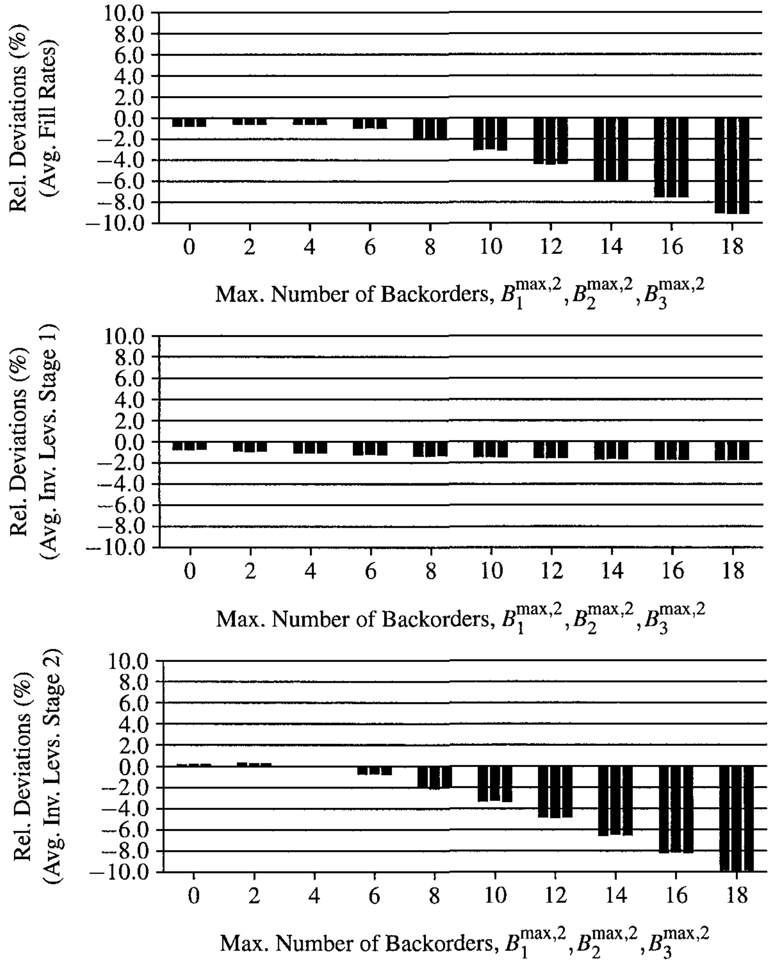


Fig. 9.23. Test set 2 (subassembly SA2 with backorders), part 1; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

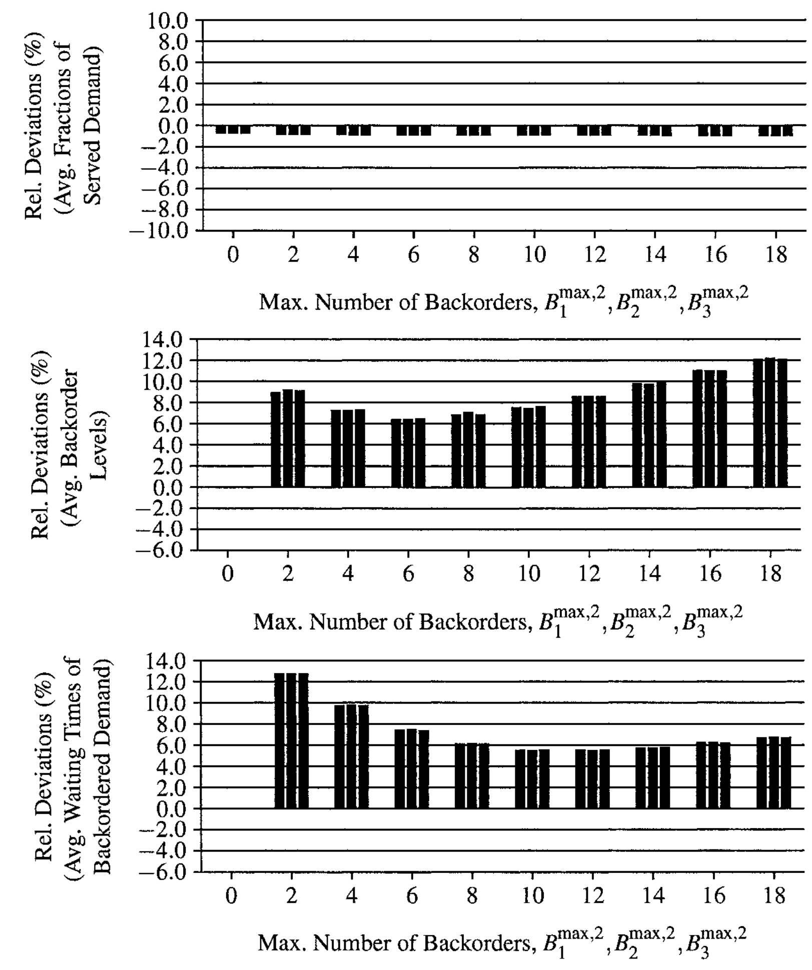


Fig. 9.24. Test set 2 (subassembly SA2 with backorders), part 2; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

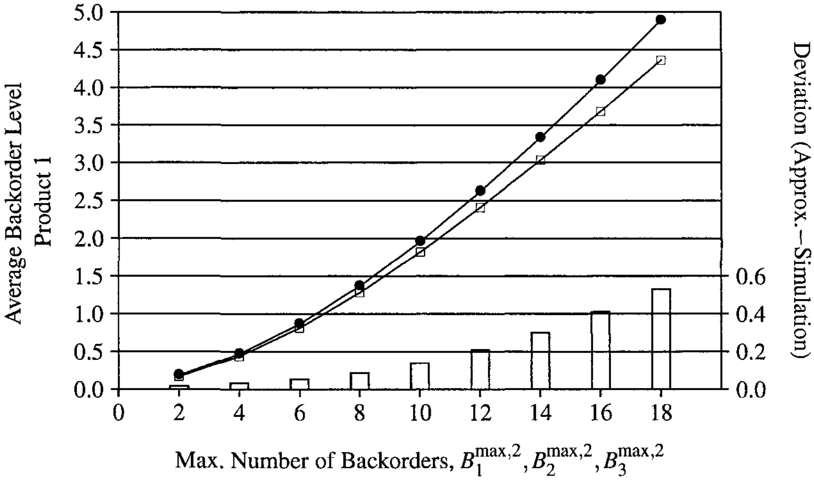


Fig. 9.25. Test set 2 (subassembly SA2 with backorders), average number of backorders for product 1; ● = approx., □ = simulation (left axis); *white bars*, (absolute) deviation (right axis).

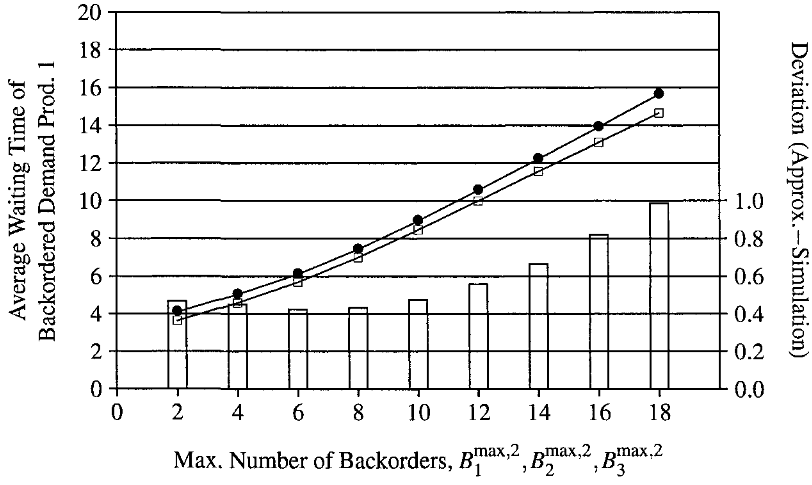


Fig. 9.26. Test set 2 (subassembly SA2 with backorders), average waiting time of backordered demand for product 1; ● = approx., □ = simulation (left axis); *white bars*, (absolute) deviation (right axis).

9.4 Test Results for Linking C1-Components

The definition of the base system and the test sets for this section are given in Tables 9.29 and 9.31. The parameter values of the three-stage version of the base system are listed in Table 9.30.

The algorithm for linking C1-components was implemented with $\varepsilon_s = 10^{-4}$. The kanban configurations, number of iterations, and computing times are listed in Tables 9.32–9.39. Most reference values were obtained from the exact model. In the simulation experiments, additional replications were made until each point estimate had a relative error of at most 0.1% at an individual confidence level of (approximately) 99%.

The results for test set 1a (two stages), test set 1b (three stages), and test sets 2a–2d $\left(\rho^{(M)} = 0.30, 0.50, 0.70, 0.90\right)$ suggest a correlation between traffic intensity and approximation errors (Figs. 9.27, 9.28, and 9.29–9.32). The largest errors seem to occur for traffic intensities around 0.50. Test sets 2a–2d indicate a strong influence of the number of stages on the relative deviations of the approximations. The worst estimates are generally received for stages in the middle of the system, whereas estimates for the first and the last stage are fairly accurate in most test instances.

Increasing the processing rates in stages 1 and 2 (test set 3) leads to more accurate approximations for the average inventory levels in the second and the third stage (Fig. 9.33). This effect may be due to smaller waiting times for input material in these stages. Increasing only the processing rate in stage 2 (test set 4) results in higher estimation errors (Fig. 9.34). Unlike the errors in test set 3, the average inventory levels in stages 2 and 3 are *over*estimated in these test instances.

Table 9.29. Base System for Linking C1-Components

Average processing rate in stage M, $\mu^{(M)}$	2.00
Traffic intensity stage M, $\rho^{(M)}$	0.70
Processing rate ratio $\mu^{(1)}/\mu^{(M)}$	1.00
Required fill rate, $f^{\min,M}$	0.95

Table 9.30. Base System with Three Stages for Linking C1-Components

Average demand rate, λ^{ext}	1.40
Average processing rate in stage m, $\mu^{(m)}$, $m = 1,2,3$	2.00
Required fill rate, $f^{\min}$	0.95

Table 9.31. Test Sets for Linking C1-Components

Test Set	Modified Parameter	Range	Increment
1a	Traffic intensity stage M, $\rho^{(M)}$ $[M=2]$	0.10–0.90	0.10
1b	Traffic intensity stage M, $\rho^{(M)}$ $[M=3]$	0.10–0.90	0.10
2a	Number of stages, M $\left[\rho^{(M)}=0.30\right]$	2–6	1
2b	Number of stages, M $\left[\rho^{(M)}=0.50\right]$	2–6	1
2c	Number of stages, M $\left[\rho^{(M)}=0.70\right]$	2–6	1
2d	Number of stages, M $\left[\rho^{(M)}=0.90\right]$	2–6	1
3	Processing rate ratio $\mu^{(1)}/\mu^{(M)}$ $\left[\mu^{(2)}=\frac{1}{2}\left(\mu^{(1)}+\mu^{(3)}\right);M=3,\ \rho^{(M)}=0.70\right]$	1.5–5.0	0.5
4	Processing rate ratios $\mu^{(2)}/\mu^{(1)},\mu^{(2)}/\mu^{(M)}$ $\left[\Rightarrow\mu^{(1)}=\mu^{(3)};M=3,\ \rho^{(M)}=0.70\right]$	1.5–5.0	0.5

Table 9.32. Test Set 1a (Linking C1-Components)

$\rho^{(2)}$	$K^{(1)},K^{(2)}$	Iterations	Time (sec.)
0.10	1,2	2	0.000
0.20	1,2	2	0.000
0.30	1,3	3	0.000
0.40	1,4	4	0.000
0.50	2,4	4	0.000
0.60	2,6	5	0.000
0.70	3,8	5	0.000
0.80	5,10	5	0.000
0.90	9,14	4	0.000

Table 9.33. Test Set 1b (Linking C1-Components)

$\rho^{(3)}$	$K^{(1)},K^{(2)},K^{(3)}$	Iterations	Time (sec.)
0.10	1,1,2	2	0.000
0.20	1,1,2	3	0.000
0.30	1,1,3	5	0.000
0.40	1,1,5	7	0.000
0.50	1,2,5	8	0.000
0.60	2,3,6	7	0.000
0.70	3,4,8	8	0.000
0.80	4,7,10	8	0.000
0.90	7,13,16	8	0.000

Table 9.34. Test Set 2a (Linking C1-Components)

M	$K^{(1)},\ldots,K^{(M)}$	Iterations	Time (sec.)
2	1,3	3	0.000
3	1,1,3	5	0.000
4	1,1,1,3	6	0.000
5	1,1,1,1,3	7	0.000
6	1,1,1,1,1,3	8	0.000

Table 9.35. Test Set 2b (Linking C1-Components)

M	$K^{(1)},\ldots,K^{(M)}$	Iterations	Time (sec.)
2	2,4	4	0.000
3	1,2,5	8	0.000
4	1,2,2,5	10	0.000
5	1,1,2,3,6	22	0.000
6	1,1,2,2,3,5	27	0.000

Table 9.36. Test Set 2c (Linking C1-Components)

M	$K^{(1)},\ldots,K^{(M)}$	Iterations	Time (sec.)
2	3,8	5	0.000
3	3,4,8	8	0.000
4	2,4,5,8	14	0.000
5	2,3,5,5,9	23	0.000
6	2,3,5,4,6,8	31	0.010

Table 9.37. Test Set 2d (Linking C1-Components)

M	$K^{(1)},\dots,K^{(M)}$	Iterations	Time (sec.)
2	9,14	4	0.000
3	7,13,16	8	0.010
4	7,11,14,16	12	0.000
5^s	8,9,13,14,17	15	0.000
6^s	8,9,12,13,15,19	18	0.000

^{s}Comparison based on simulation results.

Table 9.38. Test Set 3 (Linking C1-Components)

$\mu^{(1)}/\mu^{(M)}$	$K^{(1)},K^{(2)},K^{(3)}$	Iterations	Time (sec.)
1.5	1,3,8	7	0.000
2.0	1,2,8	5	0.000
2.5	1,2,7	4	0.000
3.0	1,2,6	3	0.000
3.5	1,1,7	4	0.000
4.0	1,1,7	4	0.000
4.5	1,1,7	3	0.000
5.0	1,1,7	3	0.000

Table 9.39. Test Set 4 (Linking C1-Components)

$\mu^{(2)}/\mu^{(1)}$, $\mu^{(2)}/\mu^{(M)}$	$K^{(1)},K^{(2)},K^{(3)}$	Iterations	Time (sec.)
1.5	2,3,7	8	0.000
2.0	2,2,7	8	0.000
2.5	2,2,7	8	0.000
3.0	2,2,7	8	0.010
3.5	1,2,7	11	0.000
4.0	1,2,7	11	0.000
4.5	1,2,7	11	0.000
5.0	1,2,7	11	0.000

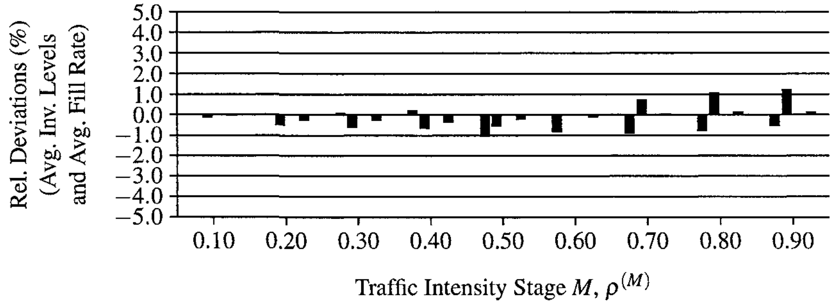


Fig. 9.27. Test set 1a (linking C1-components), $M = 2$; *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; *third bar*, rel. deviation of $\hat{f}^{(2)}$.

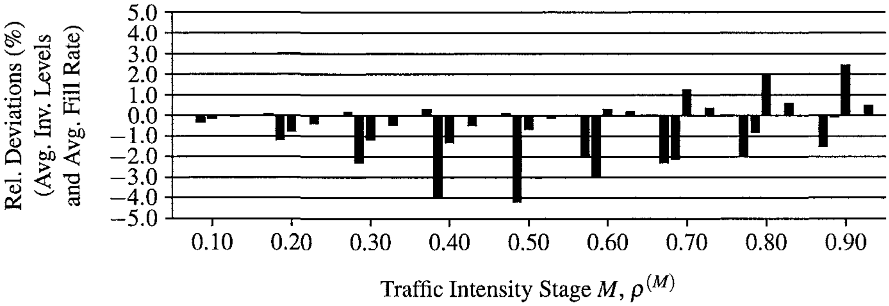


Fig. 9.28. Test set 1b (linking C1-components), $M = 3$; *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; *third bar*, rel. deviation of $\hat{\bar{y}}^{(3)}$; *fourth bar*, rel. deviation of $\hat{f}^{(3)}$.

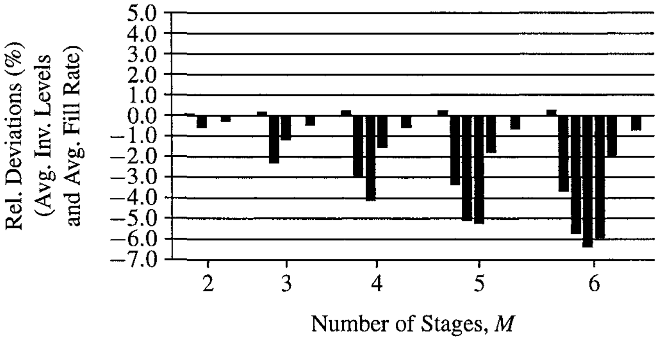


Fig. 9.29. Test set 2a (linking C1-components), $\rho^{(M)} = 0.30$; *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; ... ; *last bar*, rel. deviation of $\hat{f}^{(M)}$.

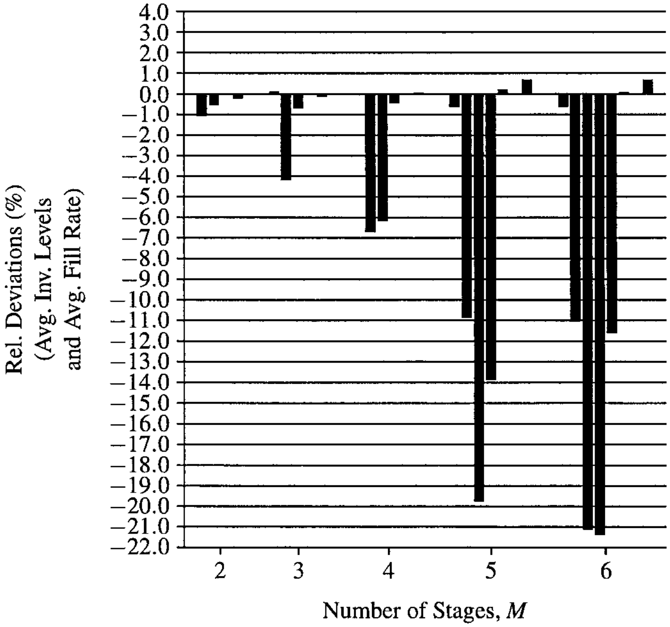


Fig. 9.30. Test set 2b (linking C1-components), $\rho^{(M)} = 0.50$; *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}^{(M)}$.

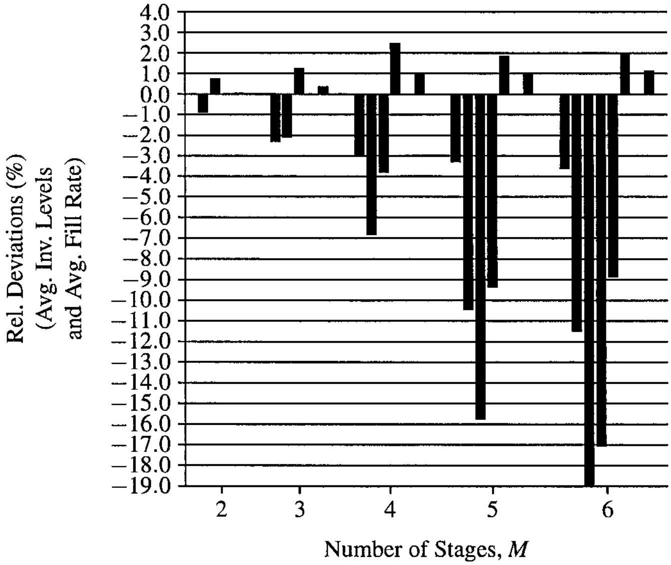


Fig. 9.31. Test set 2c (linking C1-components), $\rho^{(M)} = 0.70$; *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}^{(M)}$.

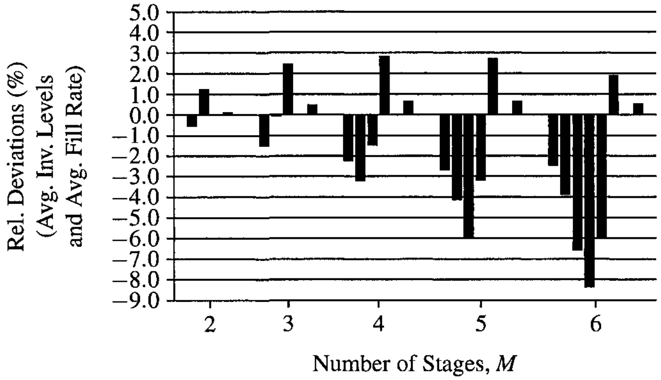


Fig. 9.32. Test set 2d (linking C1-components), $\rho^{(M)} = 0.90$; *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}^{(M)}$.

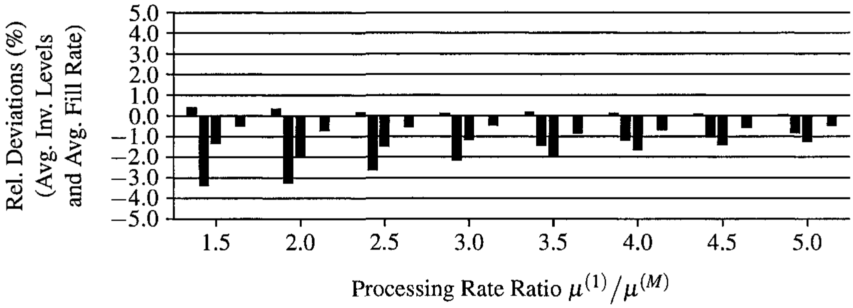


Fig. 9.33. Test set 3 (linking C1-components); *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; *third bar*, rel. deviation of $\hat{\bar{y}}^{(3)}$; *fourth bar*, rel. deviation of $\hat{f}^{(3)}$.

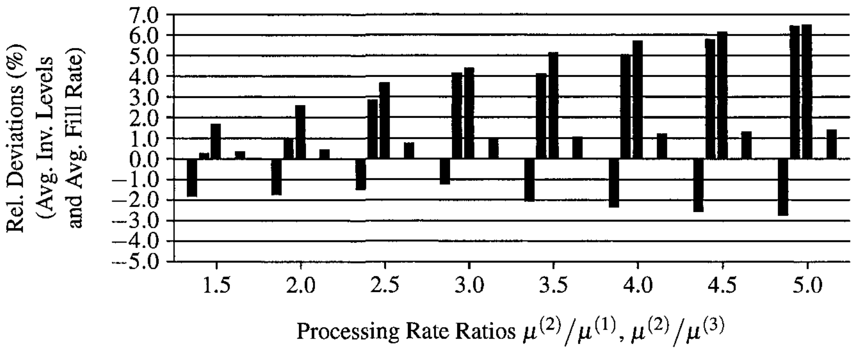


Fig. 9.34. Test set 4 (linking C1-components); *first bar*, rel. deviation of $\hat{\bar{y}}^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}^{(2)}$; *third bar*, rel. deviation of $\hat{\bar{y}}^{(3)}$; *fourth bar*, rel. deviation of $\hat{f}^{(3)}$.

9.5 Test Results for the Extended Application

In this section, we give test results for the example system in Figure 8.1 (p. 90). The algorithm for linking SA1- and SA2-subassemblies was implemented with $\varepsilon_s = 5 \times 10^{-4}$, the algorithms for subassemblies SA1 and SA2 were run with $\varepsilon_p = 10^{-4}$. The average container processing rates of the inserted single-product manufacturing facilities were set to 10^6. Most reference values were obtained via simulation. In the simulation experiments, additional replications were made until each point estimate had a relative error of at most 0.1% (test set 2d, balanced stages: 0.15%) at an individual confidence level of (approximately) 99%.

9.5.1 Tests with Balanced Stages

The definition of the base system and the test sets for this section are given in Tables 9.40 and 9.42. The parameter values of the three-stage/three-product version of the base system are listed in Table 9.41. The kanban configurations, number of iterations, and computing times are reported in Tables 9.43–9.52. The relative deviations for the average inventory levels and the average fill rate are only given for product 1 because all products of a stage share the same parameter values and, therefore, almost identical approximation errors (Figs. 9.35–9.44).

The approximation errors are generally higher than in the preceding sections, but equal or similar correlations may be observed between the total traffic intensity and the approximation errors and between the number of stages and the approximation errors (test sets 1a, 1b, and 2a–2d; Figs. 9.35–9.40).

The results of test set 3 (Fig. 9.41) resemble in broad strokes the results graphed in Figure 9.33 for single-product systems, whereas the results of test set 4 differ

Table 9.40. Base System for the Extended Application (Balanced Stages)

Average processing rate of product 1 in stage M, $\mu_1^{(M)}$	2.00
Total traffic intensity stage M, $\rho^{(M)}$	0.70
Traffic intensity ratio $\rho_1^{(M)}/\rho_r^{(M)}$ $\left[\Rightarrow \lambda_1^{\text{ext}} = \lambda_2^{\text{ext}} = \cdots = \lambda_r^{\text{ext}}\right]$	1.00
Processing rate ratio $\mu_1^{(1)}/\mu_1^{(M)}$	1.00
Processing rate ratio $\mu_1^{(m)}/\mu_r^{(m)}$, $m = 1,\ldots,M$	1.00
Setup to processing time ratio of product 1, $s_1^{(m)}\mu_1^{(m)}$, $m = 1,\ldots,M$	2.00
Setup time ratio $s_1^{(m)}/s_r^{(m)}$, $m = 1,\ldots,M$	1.00
Required fill rates, $f_i^{\min,M}$, $i = 1,\ldots,r$	0.95

Table 9.41. Base System with Three Stages and Three Products for the Extended Application (Balanced Stages)

Product i	1	2	3
Average demand rates, λ_i^{ext}	0.47	0.47	0.47
Average processing rates in stage m, $\mu_i^{(m)}$, $m = 1,2,3$	2.00	2.00	2.00
Average setup times in stage m, $s_i^{(m)}$, $m = 1,2,3$	1.00	1.00	1.00
Required fill rates, $f_i^{\min,M}$	0.95	0.95	0.95

Table 9.42. Test Sets for the Extended Application (Balanced Stages)

Test Set	Modified Parameter	Range	Increment
1a	Total traffic intensity stage M, $\rho^{(M)}$ $[M = 2]$	0.10–0.90	0.10
1b	Total traffic intensity stage M, $\rho^{(M)}$ $[M = 3]$	0.10–0.90	0.10
2a	Number of stages, M $[\rho^{(M)} = 0.30]$	2–6	1
2b	Number of stages, M $[\rho^{(M)} = 0.50]$	2–6	1
2c	Number of stages, M $[\rho^{(M)} = 0.70]$	2–6	1
2d	Number of stages, M $[\rho^{(M)} = 0.90]$	2–6	1
3	Processing rate ratio $\mu_1^{(1)}/\mu_1^{(M)}$ $[\mu_1^{(2)} = \frac{1}{2}(\mu_1^{(1)} + \mu_1^{(3)}); M = 3, \rho^{(M)} = 0.70]$	1.5–5.0	0.5
4	Processing rate ratio $\mu_1^{(2)}/\mu_1^{(1)}, \mu_1^{(2)}/\mu_1^{(M)}$ $[\Rightarrow \mu_1^{(1)} = \mu_1^{(M)}; M = 3, \rho^{(M)} = 0.70]$	1.5–5.0	0.5
5a	Number of products, r $[M = 2]$	3–10	1
5b	Number of products, r $[M = 3]$	3–10	1

Table 9.43. Test Set 1a (Ext. Appl., Balanced Stages)

$\rho^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Iterations	Time (sec.)
0.10^{e}	2,2,2	2,2,2	2	0.140
0.20^{e}	2,2,2	2,2,2	2	0.130
0.30^{e}	2,2,2	3,3,3	3	0.220
0.40	2,2,2	5,5,5	5	0.531
0.50	4,4,4	4,4,4	5	0.451
0.60	5,5,5	6,6,6	6	0.670
0.70	7,7,7	7,7,7	8	1.322
0.80	9,9,9	12,12,12	13	5.959
0.90	15,15,15	19,19,19	21	32.076

^{e}Comparison based on exact values.

Table 9.44. Test Set 1b (Ext. Appl., Balanced Stages)

$\rho^{(3)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
0.10	2,2,2	2,2,2	2,2,2	2	0.280
0.20	2,2,2	2,2,2	2,2,2	3	0.321
0.30	2,2,2	2,2,2	3,3,3	5	0.570
0.40	2,2,2	3,3,3	4,4,4	8	1.052
0.50	3,3,3	4,4,4	5,5,5	22	2.704
0.60	4,4,4	6,6,6	6,6,6	23	4.066
0.70	5,5,5	9,9,9	9,9,9	48	17.535
0.80	8,8,8	10,10,10	12,12,12	38	25.787
0.90	12,12,12	18,18,18	21,21,21	58	175.012

Table 9.45. Test Set 2a (Ext. Appl., Balanced Stages)

M	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	$\vec{K}^{(4)}$	$\vec{K}^{(5)}$	$\vec{K}^{(6)}$	Iterations	Time (sec.)
2	2,2,2	3,3,3					3	0.220
3	2,2,2	2,2,2	3,3,3				5	0.601
4	2,2,2	2,2,2	2,2,2	3,3,3			6	1.011
5	2,2,2	2,2,2	2,2,2	2,2,2	3,3,3		8	1.632
6	2,2,2	2,2,2	2,2,2	2,2,2	2,2,2	3,3,3	9	2.254

Table 9.46. Test Set 2b (Ext. Appl., Balanced Stages)

M	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	$\vec{K}^{(4)}$	$\vec{K}^{(5)}$	$\vec{K}^{(6)}$	Iterations	Time (sec.)
2	4,4,4	4,4,4					5	0.420
3	3,3,3	4,4,4	5,5,5				22	2.674
4	3,3,3	4,4,4	5,5,5	4,4,4			34	6.119
5	3,3,3	4,4,4	4,4,4	4,4,4	5,5,5		60	12.908
6	3,3,3	4,4,4	3,3,3	6,6,6	5,5,5	4,4,4	71	22.593

Table 9.47. Test Set 2c (Ext. Appl., Balanced Stages)

M	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	$\vec{K}^{(4)}$	$\vec{K}^{(5)}$	$\vec{K}^{(6)}$	Iterations	Time (sec.)
2	7,7,7	7,7,7					8	1.312
3	5,5,5	9,9,9	9,9,9				48	17.685
4	5,5,5	8,8,8	8,8,8	7,7,7			71	27.710
5	5,5,5	8,8,8	7,7,7	7,7,7	8,8,8		78	40.338
6	5,5,5	8,8,8	6,6,6	9,9,9	7,7,7	8,8,8	138	88.337

Table 9.48. Test Set 2d (Ext. Appl., Balanced Stages)

M	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	$\vec{K}^{(4)}$	$\vec{K}^{(5)}$	$\vec{K}^{(6)}$	Iterations	Time (sec.)
2	15,15,15	19,19,19					21	31.986
3	12,12,12	18,18,18	21,21,21				58	175.092
4	12,12,12	16,16,16	18,18,18	18,18,18			78	222.490
5	12,12,12	19,19,19	14,14,14	16,16,16	19,19,19		85	279.752
6	12,12,12	18,18,18	14,14,14	13,13,13	18,18,18	20,20,20	97	418.121

Table 9.49. Test Set 3 (Ext. Appl., Balanced Stages)

$\mu^{(1)}/\mu^{(M)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
1.5	3,3,3	7,7,7	8,8,8	17	3.876
2.0	2,2,2	6,6,6	7,7,7	10	2.314
2.5	2,2,2	4,4,4	8,8,8	5	1.322
3.0	2,2,2	4,4,4	7,7,7	3	0.971
3.5	2,2,2	4,4,4	7,7,7	3	0.841
4.0	2,2,2	4,4,4	7,7,7	2	0.731
4.5	2,2,2	3,3,3	8,8,8	2	0.630
5.0	2,2,2	3,3,3	8,8,8	2	0.591

Table 9.50. Test Set 4 (Ext. Appl., Balanced Stages)

$\mu^{(2)}/\mu^{(1)}$, $\mu^{(2)}/\mu^{(M)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
1.5	4,4,4	6,6,6	7,7,7	30	6.359
2.0	4,4,4	4,4,4	8,8,8	26	4.937
2.5	4,4,4	4,4,4	8,8,8	27	5.408
3.0	4,4,4	4,4,4	8,8,8	28	5.849
3.5	4,4,4	4,4,4	7,7,7	27	5.517
4.0	3,3,3	4,4,4	9,9,9	25	5.959
4.5	3,3,3	4,4,4	8,8,8	25	5.588
5.0	3,3,3	4,4,4	8,8,8	25	5.898

Table 9.51. Test Set 5a (Ext. Appl., Balanced Stages)

r	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Iterations	Time (sec.)
3	7,7,7	7,7,7	8	1.372
4	7,7,7,7	8,8,8,8	8	2.964
5	7,7,7,7,7	8,8,8,8,8	8	5.098
6	8,8,8,8,8,8	7,7,7,7,7,7	10	8.603
7	8,8,8,8,8,8,8	7,7,7,7,7,7,7	10	12.408
8	8,8,8,8,8,8,8,8	8,8,8,8,8,8,8,8	10	21.240
9	8,8,8,8,8,8,8,8,8	8,8,8,8,8,8,8,8,8	10	28.341
10	8,8,8,8,8,8,8,8,8,8	8,8,8,8,8,8,8,8,8,8	10	36.732

Table 9.52. Test Set 5b (Ext. Appl., Balanced Stages)

r	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
3	5,5,5	9,9,9	9,9,9	48	17.836
4	6,6,6,6	8,8,8,8	8,8,8,8	34	19.338
5	6,6,6,6,6	9,9,9,9,9	8,8,8,8,8	36	35.511
6	6,6,6,6,6,6	10,10,10,10,10,10	8,8,8,8,8,8	63	99.263
7	7,7,7,7,7,7,7	8,8,8,8,8,8,8	8,8,8,8,8,8,8	26	58.444
8	7,7,7,7,7,7,7,7	9,9,9,9,9,9,9,9	8,8,8,8,8,8,8,8	24	75.098
9	7,7,7,7,7,7,7,7,7	9,9,9,9,9,9,9,9,9	8,8,8,8,8,8,8,8,8	25	103.148
10	7,7,7,7,7,7,7,7,7,7	9,9,9,9,9,9,9,9,9,9	8,8,8,8,8,8,8,8,8,8	26	137.668

greatly from the ones for the single-product case (Fig. 9.42 vs. Fig. 9.34). With three products in each stage, the average inventory in stage 1 is underestimated significantly, with relative deviations up to −48.5%. The absolute approximation errors, however, are less dramatic; for example, 1.60 (simulation) vs. 0.82 (approx.) for the last test instance $\left(\mu_1^{(2)}/\mu_1^{(1)} = \mu_1^{(2)}/\mu_1^{(M)} = 5.0\right)$.

The results for test sets 5a and 5b (Figs. 9.43 and 9.44) suggest the same conclusion as the results for test set 6 for subassembly SA1 (Fig. 9.9) and test set 5 for subassembly SA2 (Fig. 9.18): increasing the number of products has only a small, if any, systematic effect on the approximation quality.

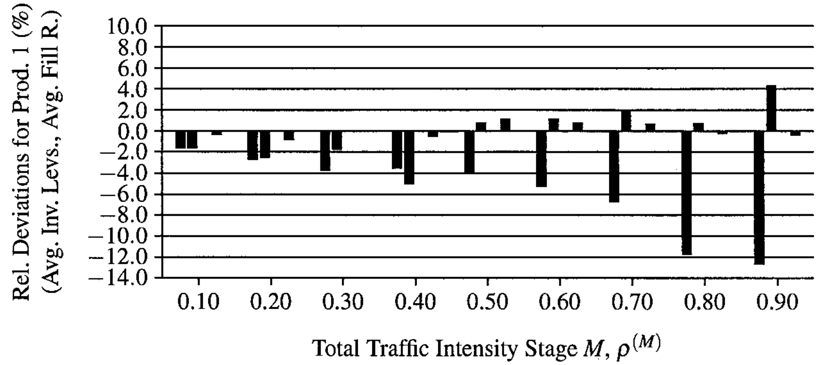


Fig. 9.35. Test set 1a (ext. appl., balanced stages), $M = 2$; *first bar*, rel. deviation of $\hat{y}_1^{(1)}$; *second bar*, rel. deviation of $\hat{y}_1^{(2)}$; *third bar*, rel. deviation of $\hat{f}_1^{(2)}$.

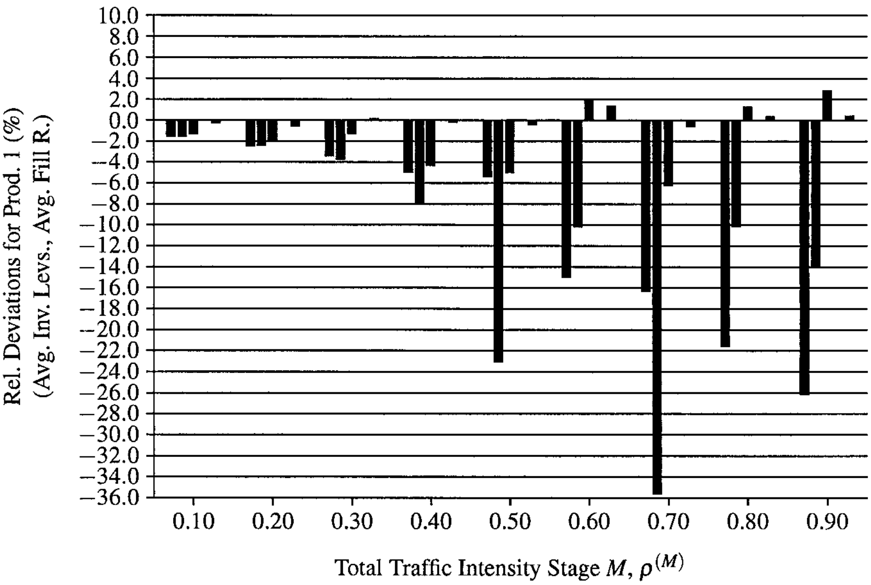


Fig. 9.36. Test set 1b (ext. appl., balanced stages), $M = 3$; *first bar*, rel. deviation of $\hat{y}_1^{(1)}$; *second bar*, rel. deviation of $\hat{y}_1^{(2)}$; *third bar*, rel. deviation of $\hat{y}_1^{(3)}$; *fourth bar*, rel. dev. of $\hat{f}_1^{(3)}$.

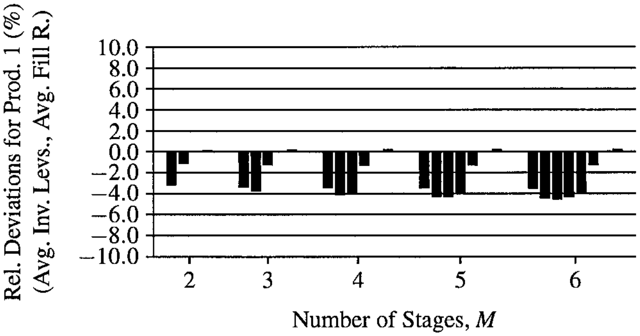


Fig. 9.37. Test set 2a (ext. appl., balanced stages), $\rho^{(M)} = 0.30$; *first bar*, rel. deviation of $\hat{y}_1^{(1)}$; *second bar*, rel. deviation of $\hat{y}_1^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}_1^{(M)}$.

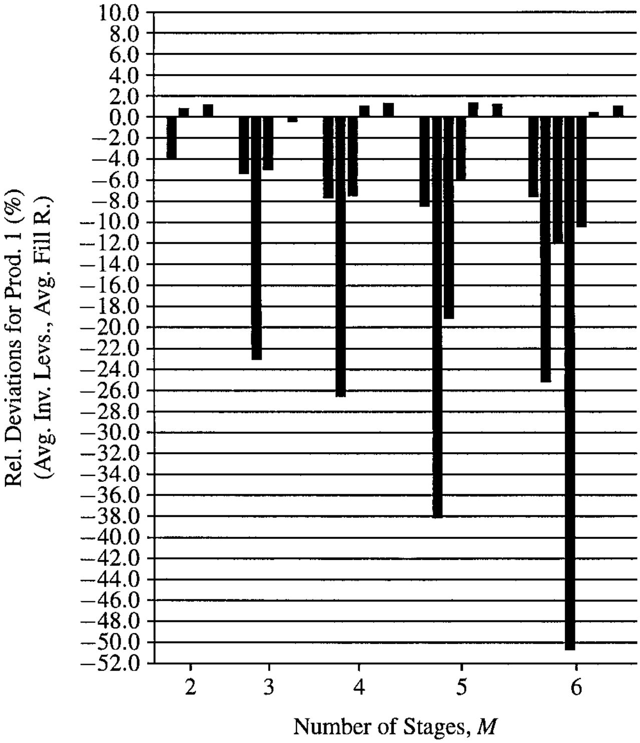


Fig. 9.38. Test set 2b (ext. appl., balanced stages), $\rho^{(M)} = 0.50$; *first bar*, rel. deviation of $\hat{y}_1^{(1)}$; *second bar*, rel. deviation of $\hat{y}_1^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}_1^{(M)}$.

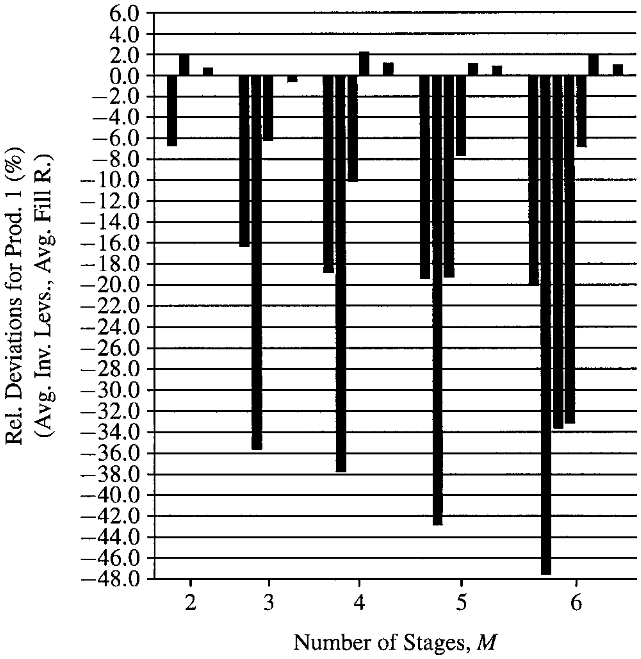


Fig. 9.39. Test set 2c (ext. appl., balanced stages), $\rho^{(M)} = 0.70$; *first bar*, rel. deviation of $\hat{y}_1^{(1)}$; *second bar*, rel. deviation of $\hat{y}_1^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}_1^{(M)}$.

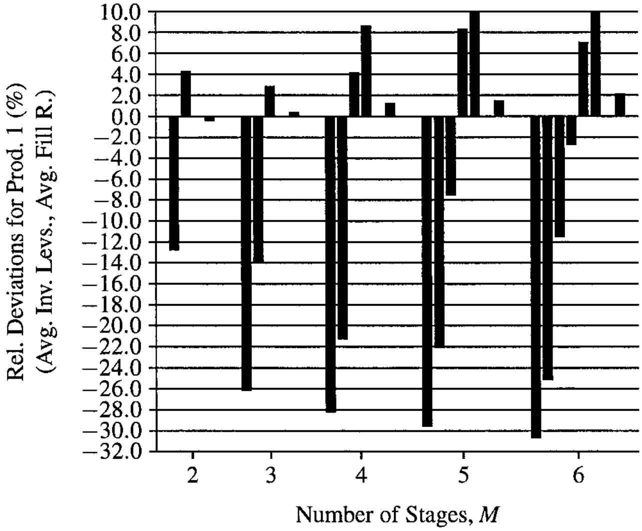


Fig. 9.40. Test set 2d (ext. appl., balanced stages), $\rho^{(M)} = 0.90$; *first bar*, rel. deviation of $\hat{y}_1^{(1)}$; *second bar*, rel. deviation of $\hat{y}_1^{(2)}$; ...; *last bar*, rel. deviation of $\hat{f}_1^{(M)}$.

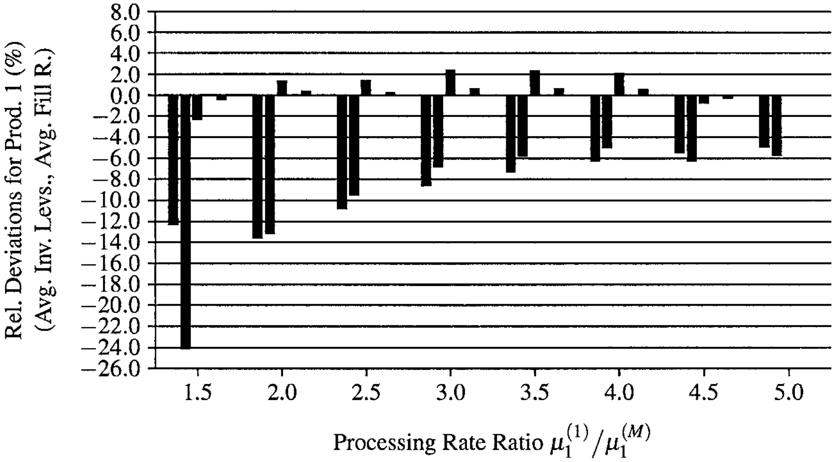


Fig. 9.41. Test set 3 (ext. appl., balanced stages); *first bar*, rel. deviation of $\hat{\bar{y}}_1^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}_1^{(2)}$; *third bar*, rel. deviation of $\hat{\bar{y}}_1^{(3)}$; *fourth bar*, rel. deviation of $\hat{\bar{f}}_1^{(3)}$.

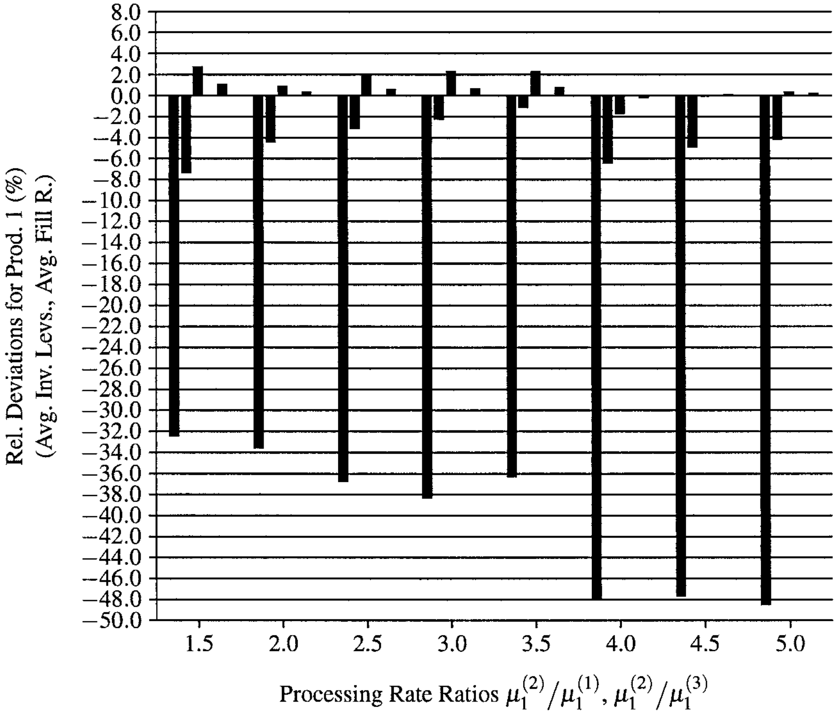


Fig. 9.42. Test set 4 (ext. appl., balanced stages); *first bar*, rel. deviation of $\hat{\bar{y}}_1^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}_1^{(2)}$; *third bar*, rel. deviation of $\hat{\bar{y}}_1^{(3)}$; *fourth bar*, rel. deviation of $\hat{\bar{f}}_1^{(3)}$.

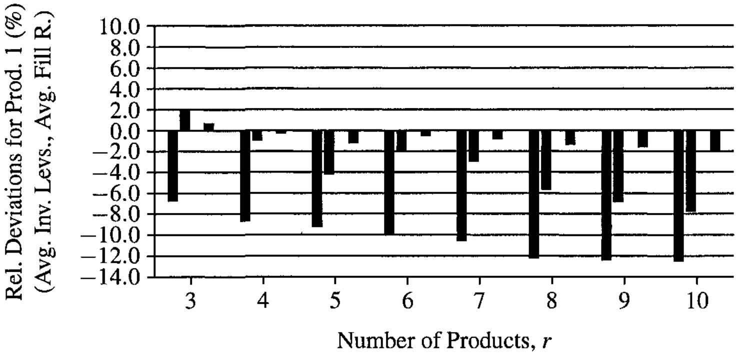


Fig. 9.43. Test set 5a (ext. appl., balanced stages), $M = 2$; *first bar*, rel. deviation of $\hat{\bar{y}}_1^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}_1^{(2)}$; *third bar*, rel. deviation of $\hat{f}_1^{(2)}$.

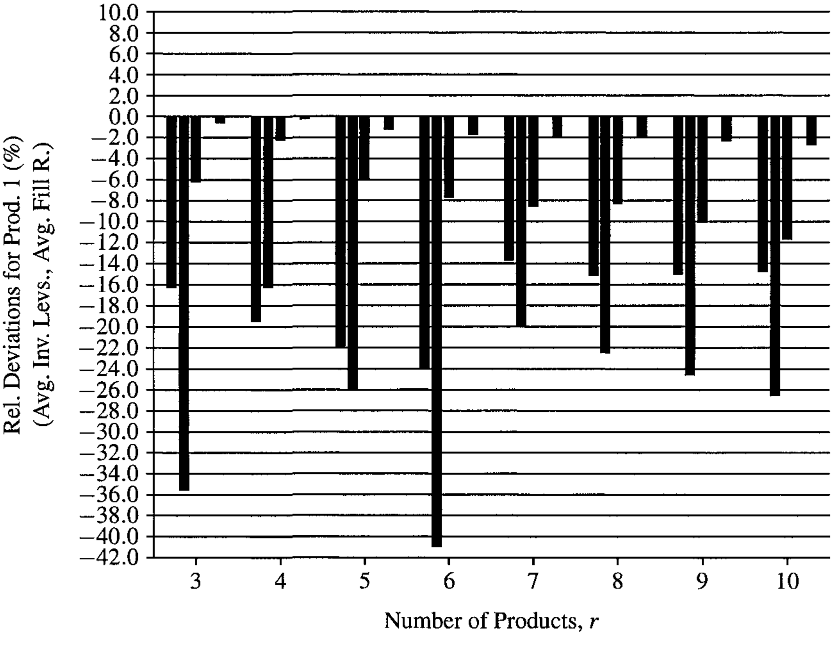


Fig. 9.44. Test set 5b (ext. appl., balanced stages), $M = 3$; *first bar*, rel. deviation of $\hat{\bar{y}}_1^{(1)}$; *second bar*, rel. deviation of $\hat{\bar{y}}_1^{(2)}$; *third bar*, rel. deviation of $\hat{\bar{y}}_1^{(3)}$; *fourth bar*, rel. dev. of $\hat{f}_1^{(3)}$.

9.5.2 Tests with Unbalanced Stages

The definition of the base system and the test sets for this section are given in Tables 9.53 and 9.55. The parameter values of the three-stage/three-product version of the base system are listed in Table 9.54. The kanban configurations, number of iterations, and computing times are reported in Tables 9.56–9.61. Figures 9.45–9.51 and 9.56–9.57 display the relative deviations of the approximations for the average fill rates and for the average inventory levels for all products. In addition, the values of the performance measures obtained from the suggested procedure and from simulation are given in Figures 9.52–9.55 for test set 2b and in Figures 9.58–9.61 for test set 3b. These plots indicate that the approximations follow the reference values closely.

Test sets 1a and 1b indicate (again) a correlation between the total traffic intensity and the approximation errors (Figs. 9.45–9.47). Unbalancing the system with respect to the traffic intensities (test sets 2a and 2b, Figs. 9.48–9.51) provokes more inconsistent approximation errors—particularly in the last stage—than spreading the average container processing rates while maintaining the ratios of the traffic intensi-

Table 9.53. Base System for the Extended Application (Unbalanced Stages)

Average processing rate of product 1 in stage M, $\mu_1^{(M)}$	2.00
Total traffic intensity stage M, $\rho^{(M)}$	0.70
Traffic intensity ratio $\rho_1^{(M)}/\rho_r^{(M)}$	1.60
Processing rate ratio $\mu_1^{(1)}/\mu_1^{(M)}$	1.00
Processing rate ratio $\mu_1^{(m)}/\mu_r^{(m)}$, $m=1,\dots,M$	0.90
Setup to processing time ratio of product 1, $s_1^{(m)}\mu_1^{(m)}$, $m=1,\dots,M$	2.00
Setup time ratio $s_1^{(m)}/s_r^{(m)}$, $m=1,\dots,M$	0.80
Required fill rates, $f_i^{\min,M}$, $i=1,\dots,r$	0.95

Table 9.54. Base System with Three Stages and Three Products for the Extended Application (Unbalanced Stages)

Product i	1	2	3
Average demand rates, λ_i^{ext}	0.57	0.49	0.40
Average processing rates in stage m, $\mu_i^{(m)}$, $m=1,2,3$	2.00	2.11	2.22
Average setup times in stage m, $s_i^{(m)}$, $m=1,2,3$	1.00	1.125	1.25
Required fill rates, $f_i^{\min,M}$	0.95	0.95	0.95

Table 9.55. Test Sets for the Extended Application (Unbalanced Stages)

Test Set	Modified Parameter		Range	Increment
1a	Total traffic intensity stage M, $\rho^{(M)}$	$[M=2]$	0.10–0.90	0.10
1b	Total traffic intensity stage M, $\rho^{(M)}$	$[M=3]$	0.10–0.90	0.10
2a	Traffic intensity ratio $\rho_1^{(M)}/\rho_r^{(M)}$ $[M=2,\ \rho^{(M)}=0.70]$		2–20	2
2b	Traffic intensity ratio $\rho_1^{(M)}/\rho_r^{(M)}$ $[M=3,\ \rho^{(M)}=0.70]$		2–20	2
3a	Processing rate ratio $\mu_1^{(M)}/\mu_r^{(M)}$ with reversed first stage $\big[M=2,\ \rho^{(M)}=0.70,\ s_i^{(m)}\mu_i^{(m)}=2.00,$ $i=1,\ldots,r; m=1,\ldots,M\big]$		1.00–0.55	0.05
3b	Processing rate ratio $\mu_1^{(M)}/\mu_r^{(M)}$ with reversed second stage $\big[M=3,\ \rho^{(M)}=0.70,\ s_i^{(m)}\mu_i^{(m)}=2.00,$ $i=1,\ldots,r; m=1,\ldots,M\big]$		1.00–0.55	0.05

Table 9.56. Test Set 1a (Ext. Appl., Unbalanced Stages)

$\rho^{(2)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Iterations	Time (sec.)
0.10[e]	2,2,2	2,2,2	2	0.121
0.20[e]	2,2,2	3,3,2	3	0.170
0.30[e]	2,2,2	4,3,3	4	0.260
0.40	3,3,2	5,5,5	5	0.431
0.50	4,4,4	6,5,5	7	0.661
0.60	6,5,5	7,7,6	7	0.971
0.70	7,7,7	11,10,8	10	2.694
0.80	11,11,10	14,13,13	15	8.933
0.90	17,17,17	23,24,22	22	54.929

[e] Comparison based on exact values.

ties (test sets 3a and 3b, Figs. 9.56–9.57). The magnitude of the relative deviations, however, is similar in these test sets.

Table 9.57. Test Set 1b (Ext. Appl., Unbalanced Stages)

$\rho^{(3)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
0.10	2,2,2	2,2,2	2,2,2	2	0.280
0.20	2,2,2	2,2,2	3,3,2	3	0.321
0.30	2,2,2	2,2,2	4,4,3	7	0.571
0.40	3,2,2	4,4,4	4,4,4	20	1.061
0.50	4,4,3	4,4,4	7,6,6	20	2.604
0.60	5,5,4	6,6,6	8,7,6	24	4.086
0.70	7,7,6	8,8,7	10,9,10	24	17.535
0.80	10,9,9	12,12,11	14,15,13	37	36.903
0.90	14,15,15	18,19,19	26,25,22	47	150.186

Table 9.58. Test Set 2a (Ext. Appl., Unbalanced Stages)

$\rho_1^{(M)}/\rho_r^{(M)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Iterations	Time (sec.)
2	8,7,6	10,10,8	11	2.774
4	8,7,4	10,10,7	11	2.534
6	8,7,4	10,9,5	10	1.953
8	7,7,3	11,9,5	9	1.802
10	7,7,2	11,9,5	9	1.923
12	7,7,2	11,9,4	9	1.833
14	7,6,2	10,10,3	11	2.143
16	7,6,2	10,9,3	10	1.772
18	7,6,2	10,9,3	10	1.753
20	7,6,2	10,9,3	10	1.762

Table 9.59. Test Set 2b (Ext. Appl., Unbalanced Stages)

$\rho_1^{(M)}/\rho_r^{(M)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
2	7,6,6	8,9,6	11,10,9	31	12.659
4	7,6,4	8,8,5	12,11,7	31	13.118
6	7,6,3	9,8,4	10,10,7	34	11.527
8	7,6,2	8,8,4	11,10,5	38	12.768
10	7,6,2	8,8,3	11,9,5	24	7.431
12	6,6,2	8,8,2	12,9,5	29	10.435
14	6,6,2	8,7,2	11,10,4	26	8.252
16	6,6,2	8,7,2	10,9,3	23	6.349
18	6,6,2	8,7,2	10,9,3	23	6.209
20	6,6,2	8,7,2	9,8,3	21	5.087

Table 9.60. Test Set 3a (Ext. Appl., Unbalanced Stages)

$\mu_1^{(M)}/\mu_r^{(M)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	Iterations	Time (sec.)
1.00	7,6,6	9,9,7	11	2.164
0.95	7,6,6	8,8,7	10	1.782
0.90	7,6,6	8,8,7	10	1.743
0.85	6,6,6	9,9,7	12	2.433
0.80	6,6,6	9,9,8	12	2.524
0.75	6,6,6	9,9,8	12	2.563
0.70	6,6,6	8,9,8	12	2.404
0.65	6,6,6	8,9,9	12	2.584
0.60	6,7,6	8,8,10	12	2.573
0.55	6,7,7	8,9,9	12	2.344

Table 9.61. Test Set 3b (Ext. Appl., Unbalanced Stages)

$\mu_1^{(M)}/\mu_r^{(M)}$	$\vec{K}^{(1)}$	$\vec{K}^{(2)}$	$\vec{K}^{(3)}$	Iterations	Time (sec.)
1.00	6,6,5	8,7,6	9,8,8	26	7.841
0.95	6,6,5	7,7,6	10,8,8	23	7.250
0.90	6,6,5	7,7,6	9,8,8	23	6.730
0.85	6,6,5	7,6,6	9,10,9	24	8.482
0.80	6,6,5	7,6,7	9,10,8	23	7.761
0.75	6,6,5	7,6,7	8,10,9	24	8.092
0.70	6,6,5	7,7,7	8,8,9	23	6.880
0.65	6,6,5	6,7,8	10,9,9	33	11.296
0.60	6,6,6	6,7,7	9,9,9	22	7.541
0.55	6,6,6	6,8,7	9,8,10	26	8.973

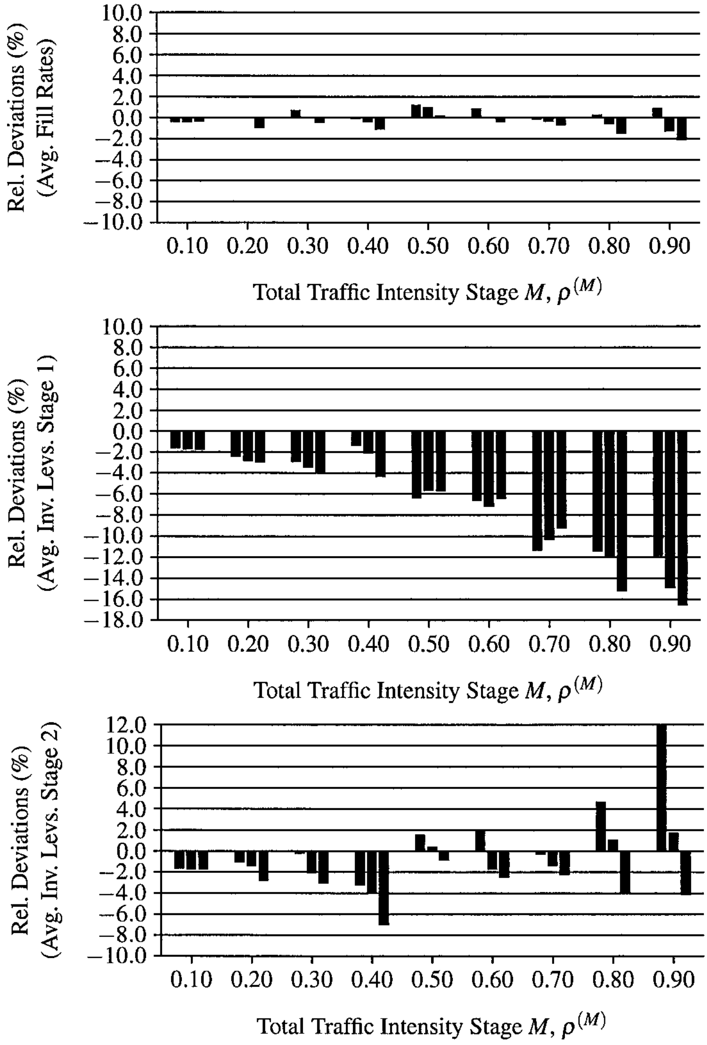
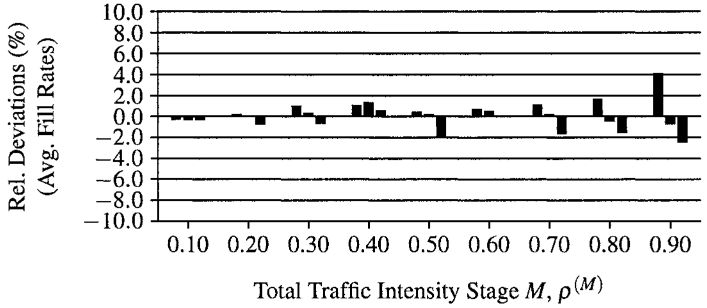


Fig. 9.45. Test set 1a (ext. appl., unbalanced stages); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.



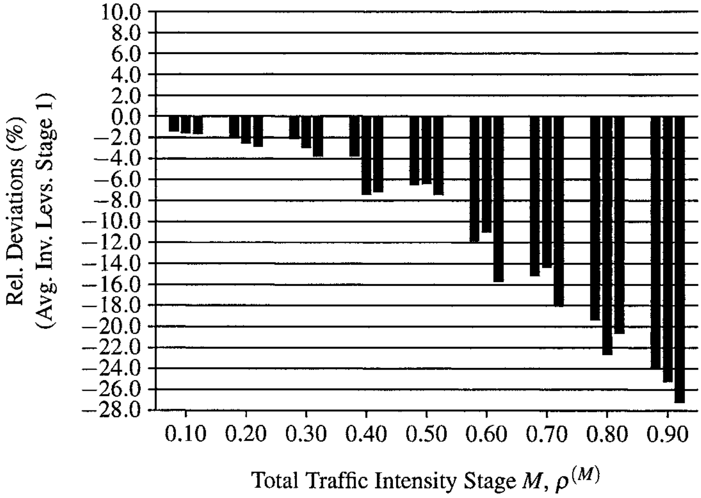
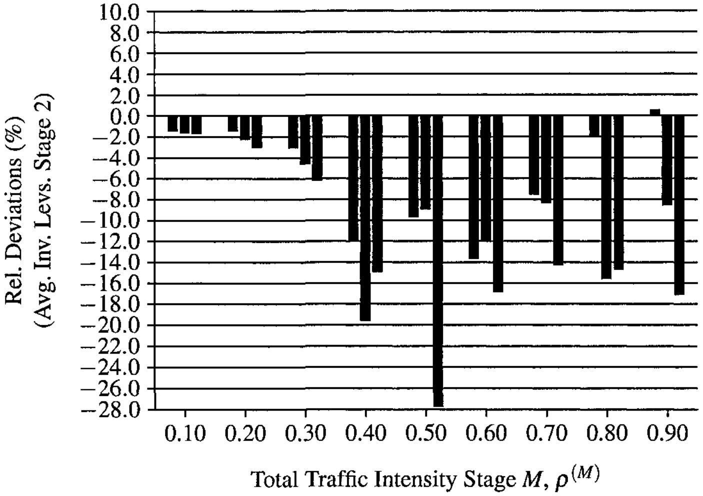


Fig. 9.46. Test set 1b (ext. appl., unbalanced stages), part 1; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.



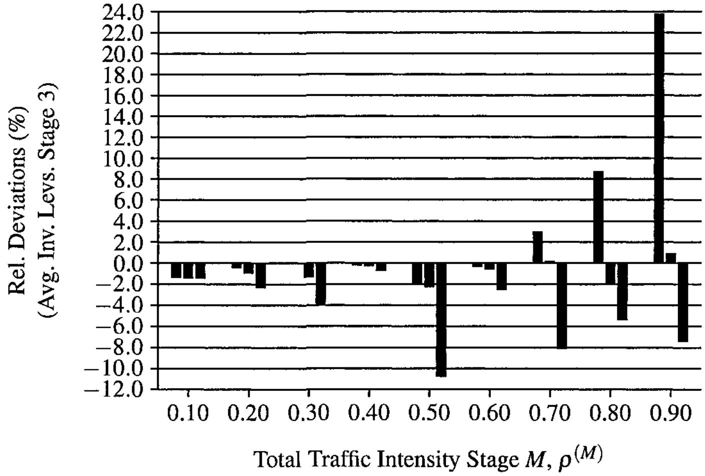


Fig. 9.47. Test set 1b (ext. appl., unbalanced stages), part 2; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

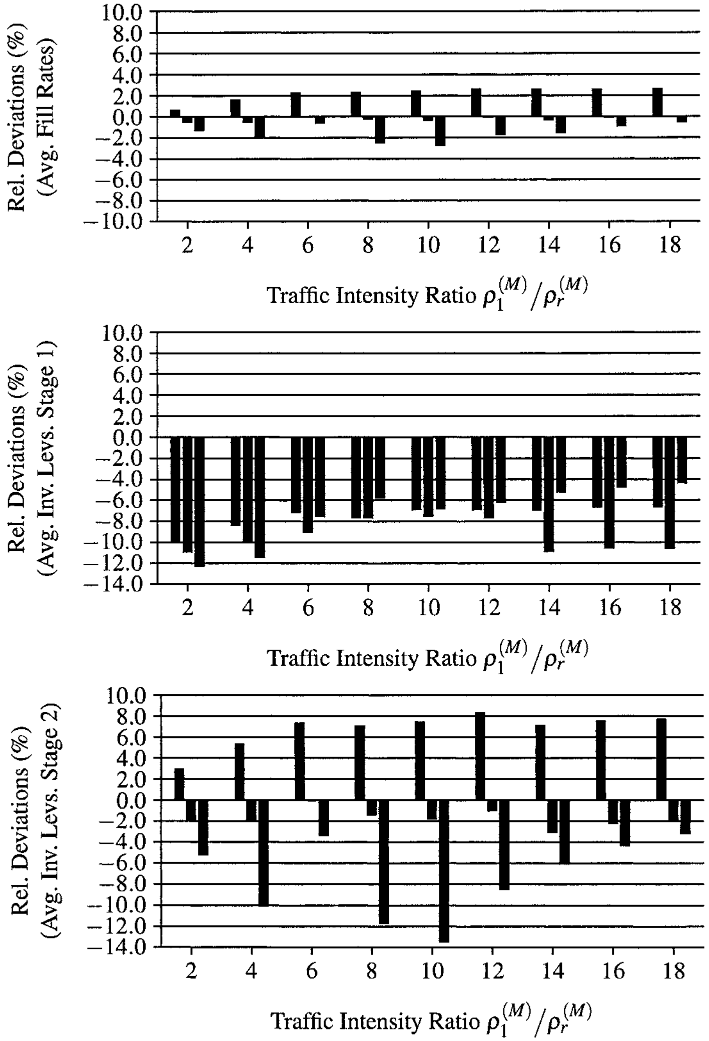


Fig. 9.48. Test set 2a (ext. appl., unbalanced stages); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

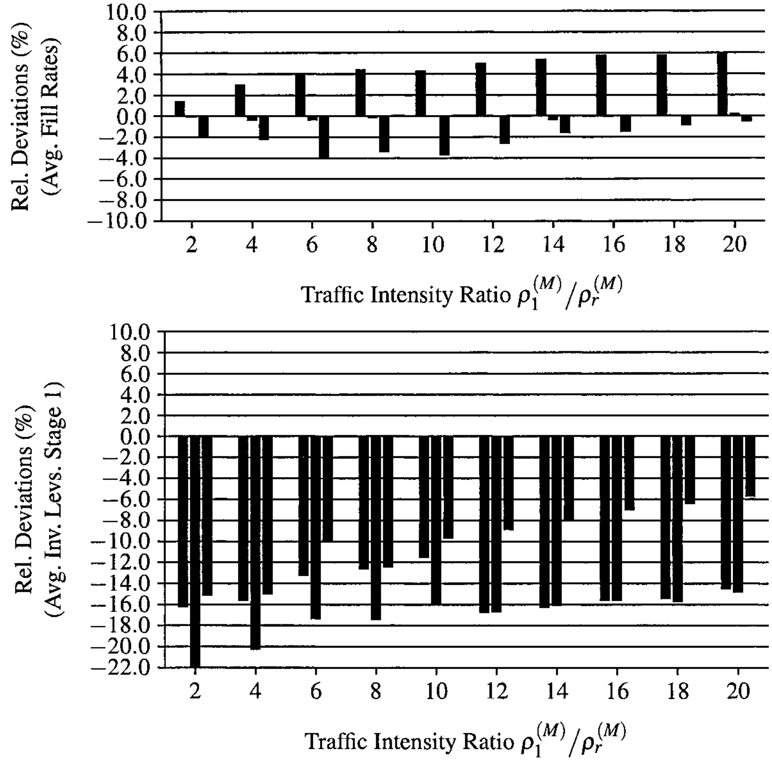


Fig. 9.49. Test set 2b (ext. appl., unbalanced stages), part 1; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

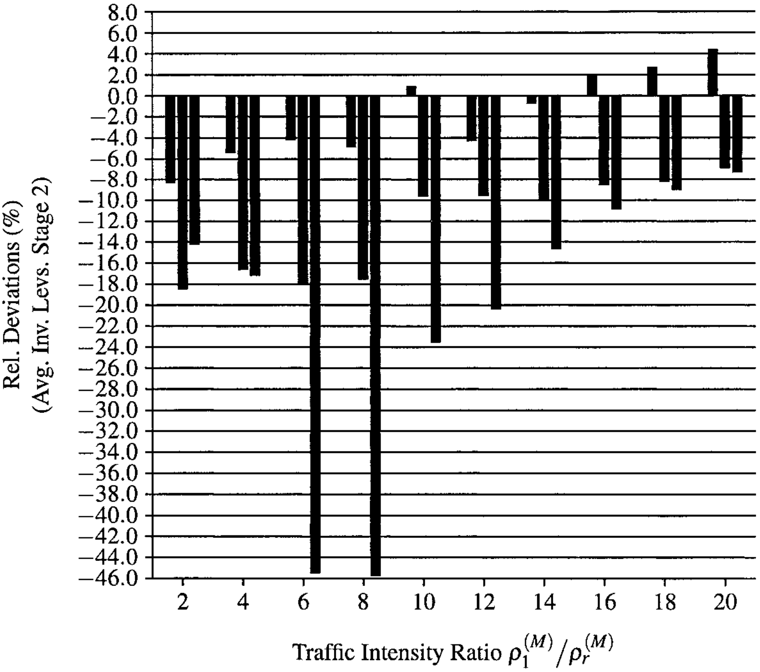


Fig. 9.50. Test set 2b (ext. appl., unbalanced stages), part 2; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

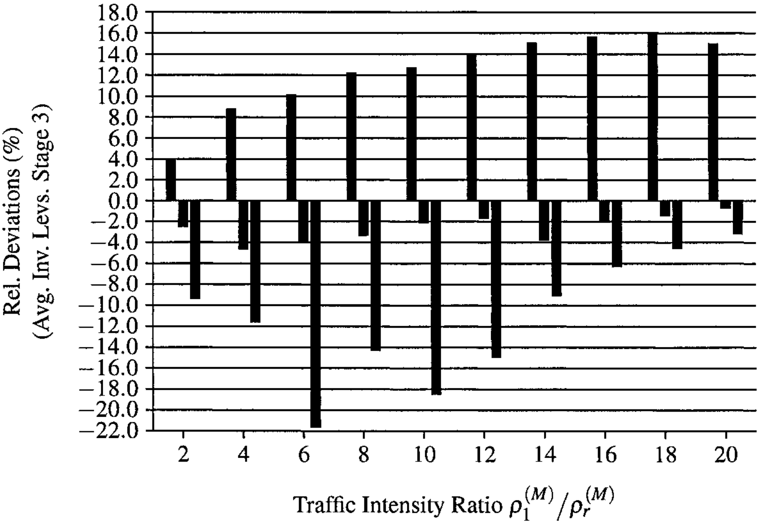


Fig. 9.51. Test set 2b (ext. appl., unbalanced stages), part 3; *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

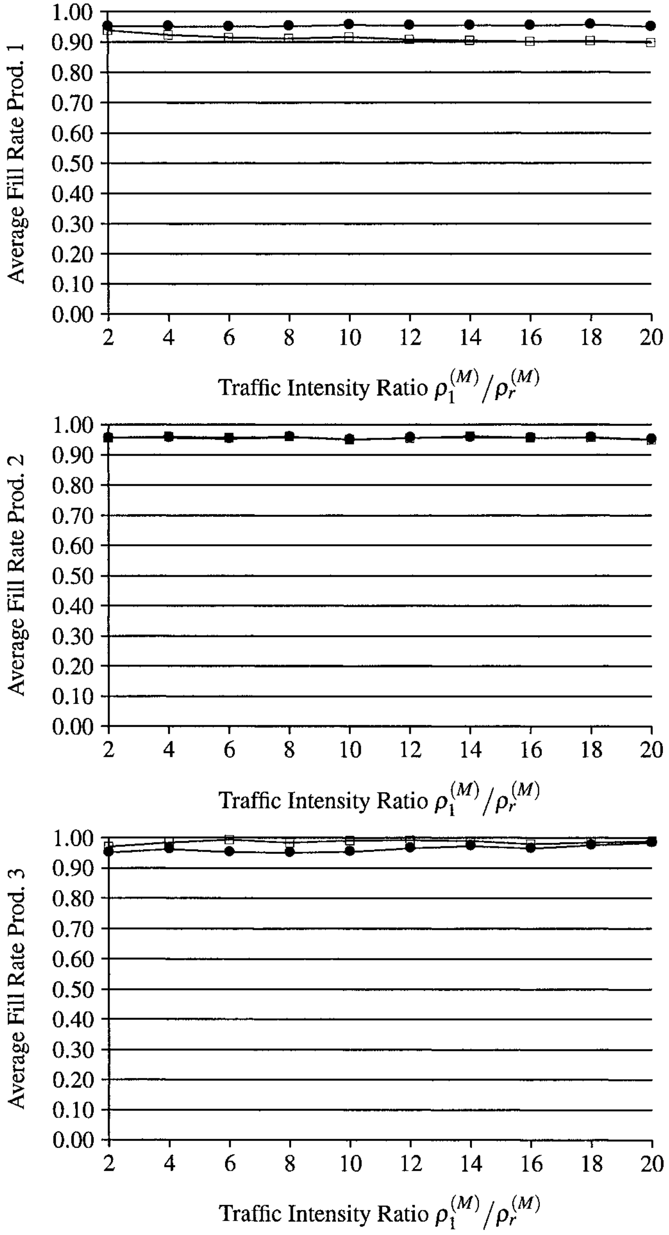


Fig. 9.52. Test set 2b (ext. appl., unbalanced stages), average fill rates; ● = approx., □ = simulation.

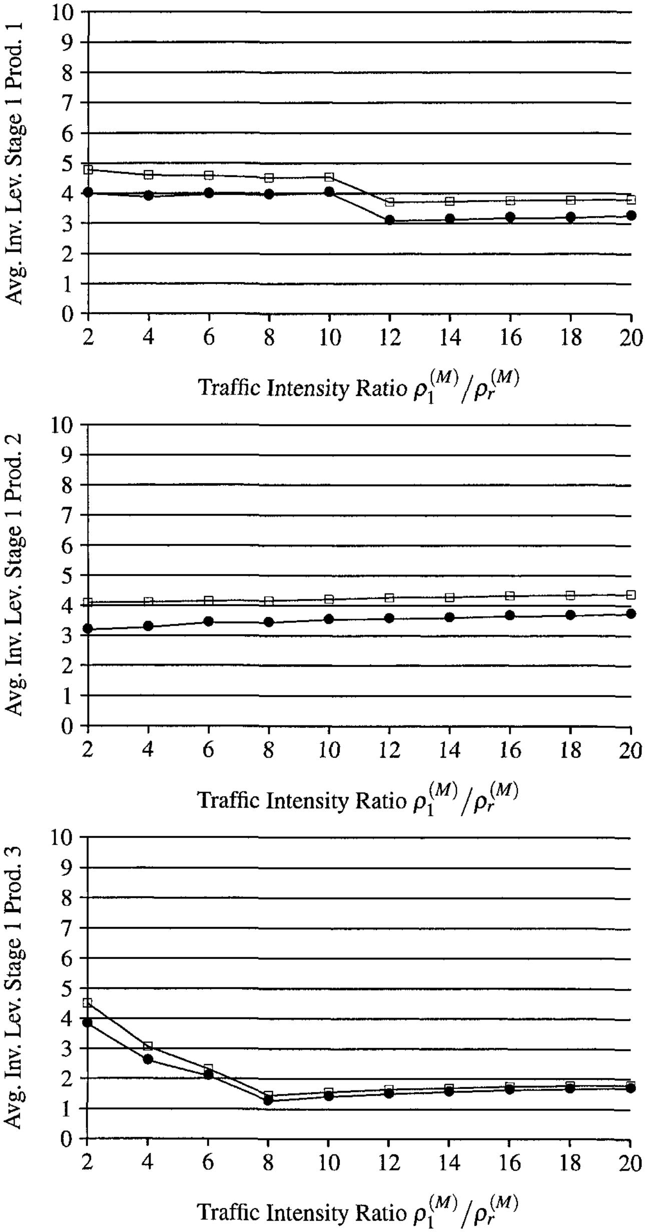


Fig. 9.53. Test set 2b (ext. appl., unbalanced stages), average inventory levels in stage 1; ● = approx., □ = simulation.

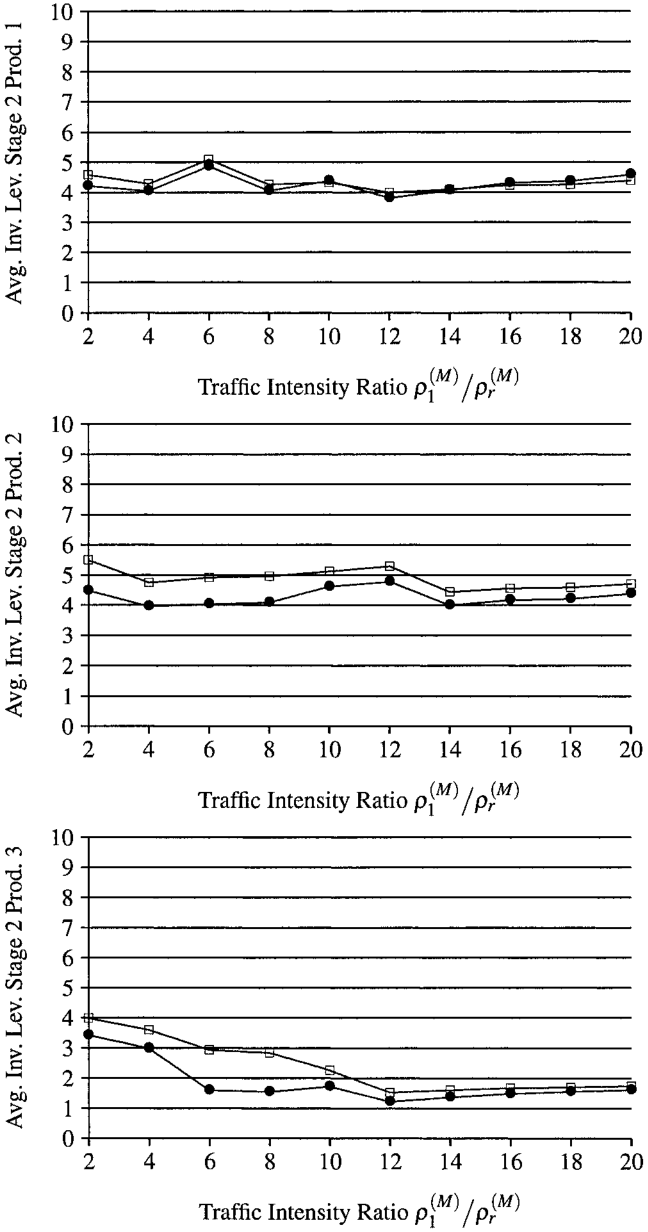


Fig. 9.54. Test set 2b (ext. appl., unbalanced stages), average inventory levels in stage 2; ● = approx., □ = simulation.

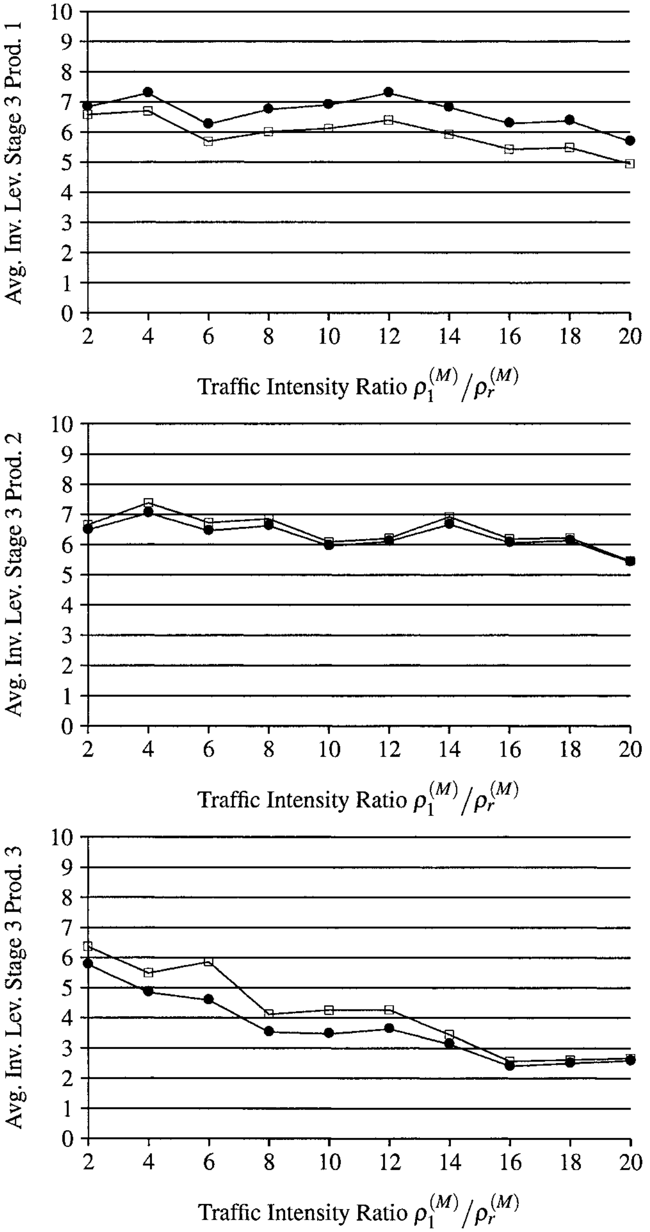


Fig. 9.55. Test set 2b (ext. appl., unbalanced stages), average inventory levels in stage 3; ● = approx., □ = simulation.

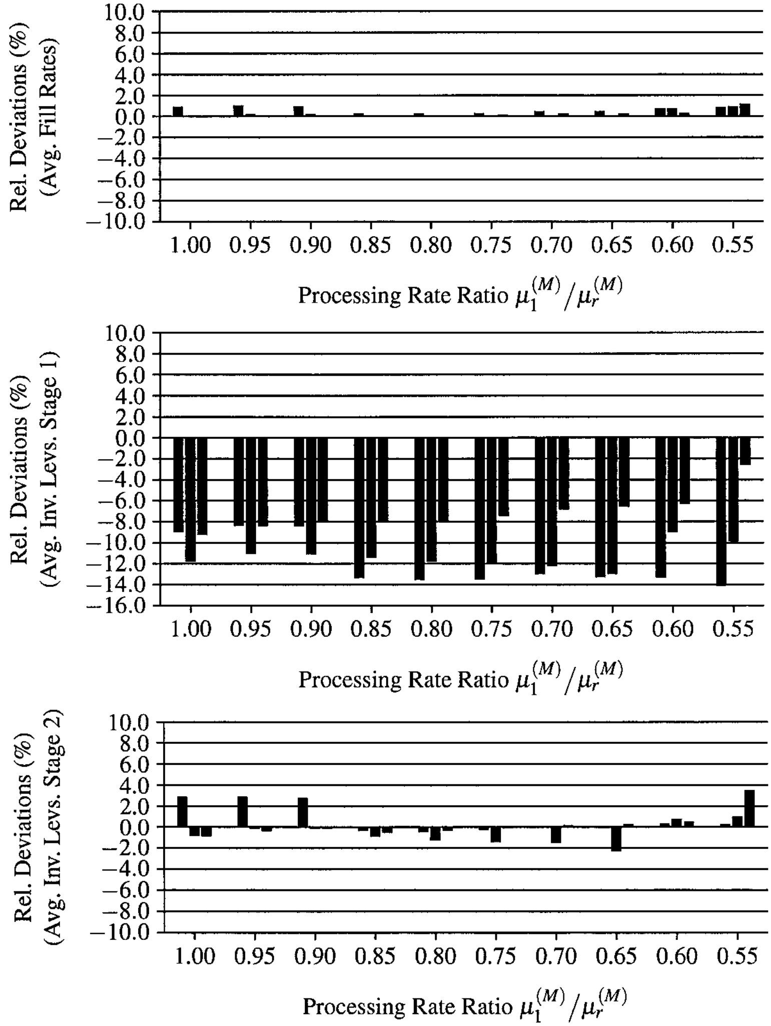


Fig. 9.56. Test set 3a (ext. appl., unbalanced stages); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

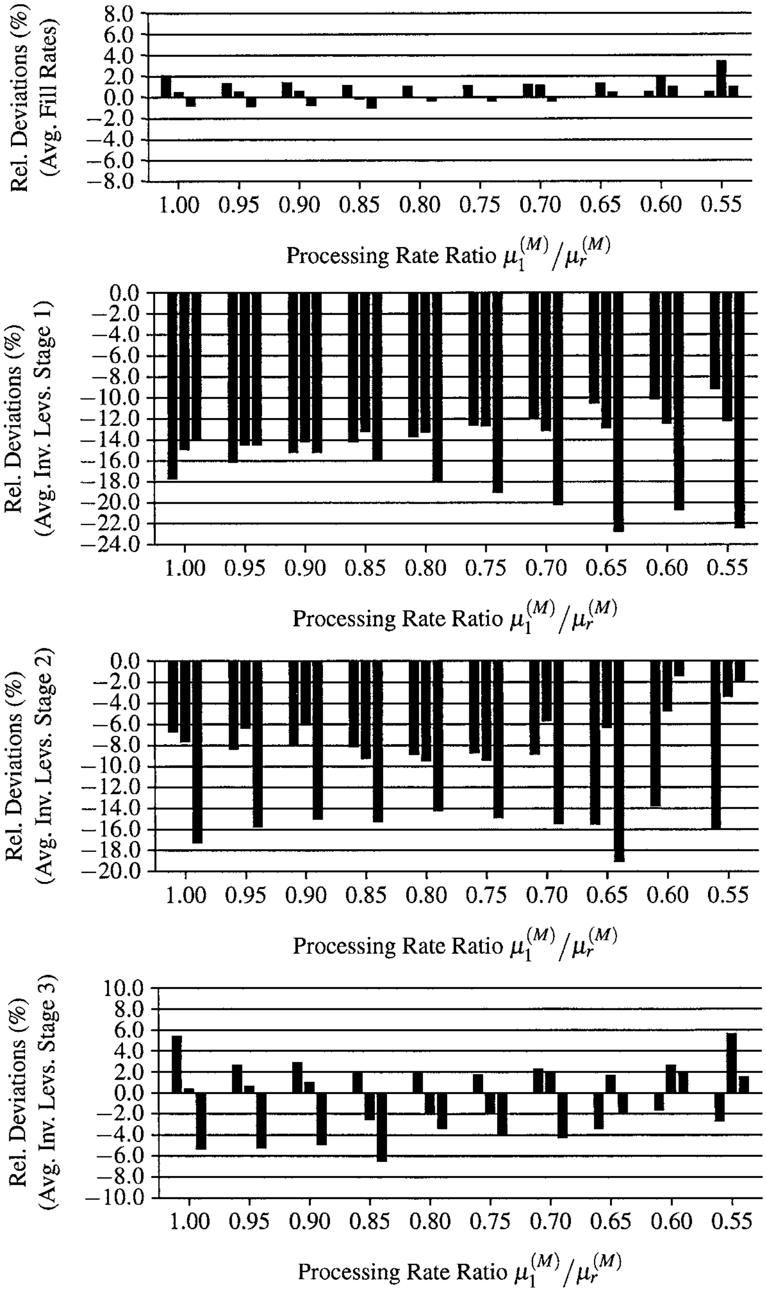


Fig. 9.57. Test set 3b (ext. appl., unbalanced stages); *first bar*, product 1; *second bar*, product 2; *third bar*, product 3.

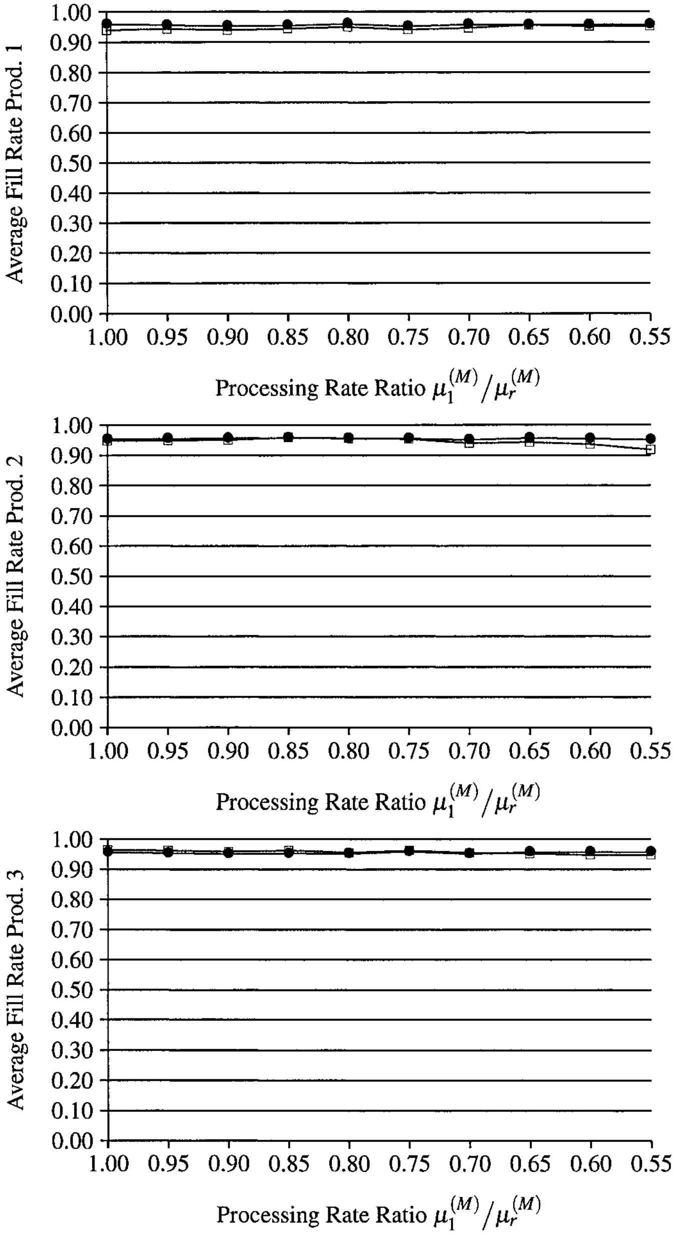


Fig. 9.58. Test set 3b (ext. appl., unbalanced stages), average fill rates; • = approx., □ = simulation.

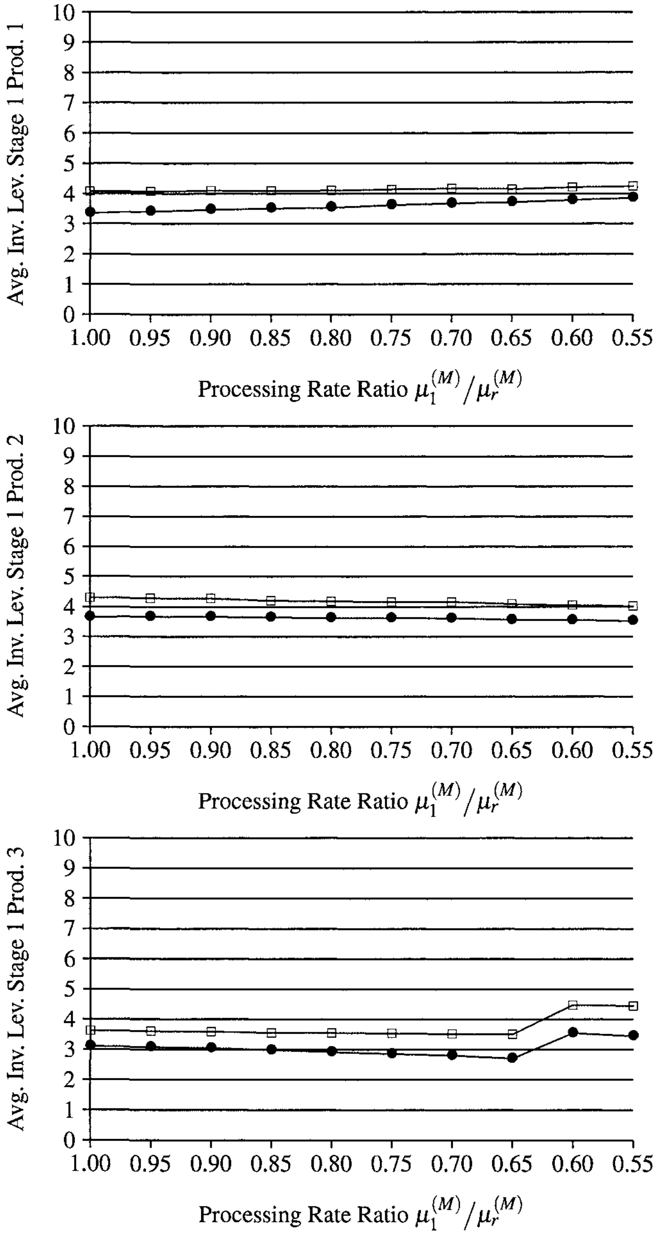


Fig. 9.59. Test set 3b (ext. appl., unbalanced stages), average inventory levels in stage 1; ● = approx., □ = simulation.

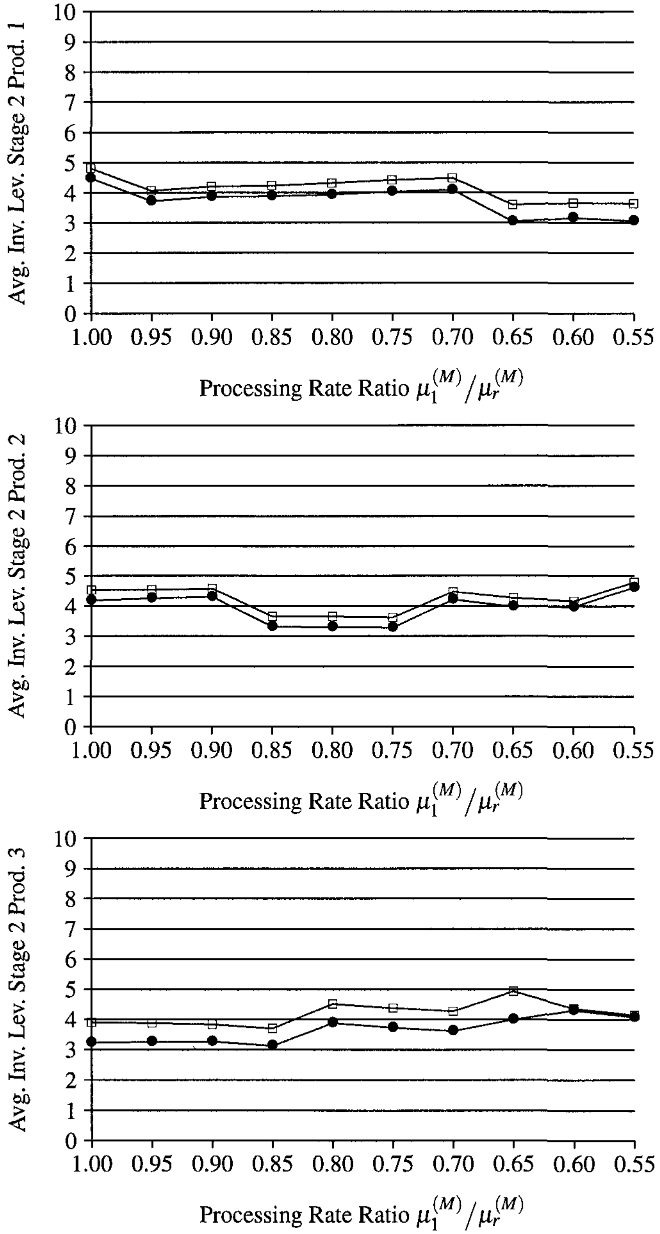


Fig. 9.60. Test set 3b (ext. appl., unbalanced stages), average inventory levels in stage 2; ● = approx., □ = simulation.

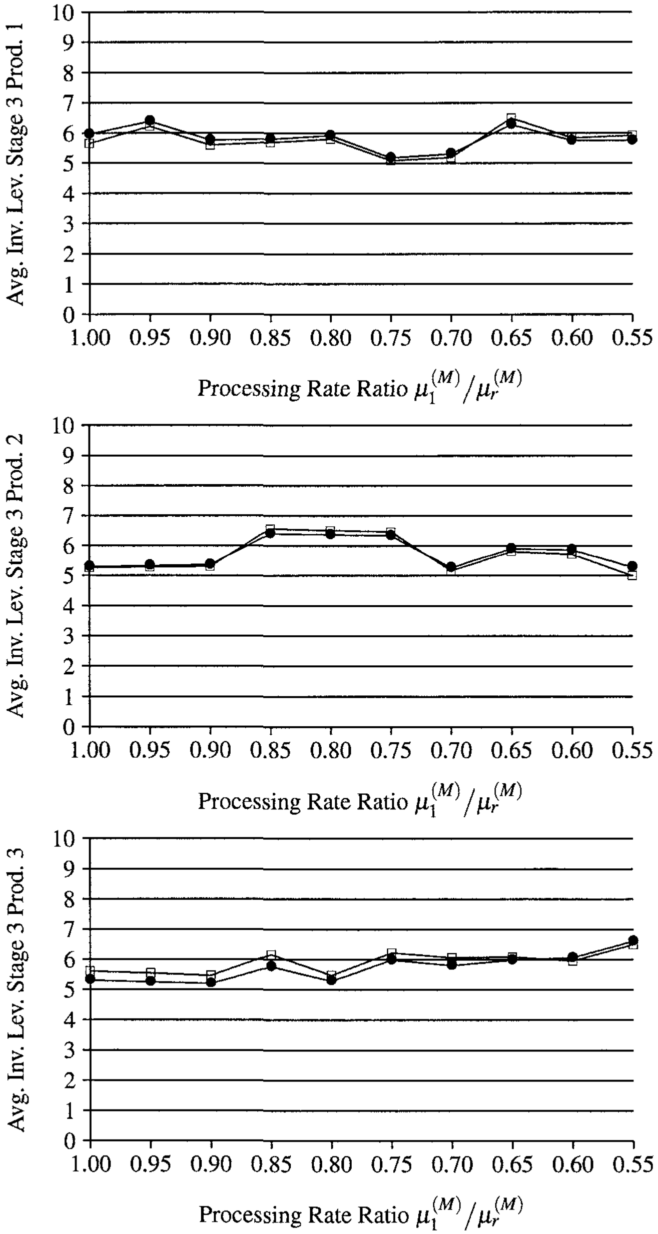


Fig. 9.61. Test set 3b (ext. appl., unbalanced stages), average inventory levels in stage 3; ● = approx., □ = simulation.

10

Application of the Models: Analysis of System Behavior

The models built with the construction kit allow the system analyst to approximate performance measures that characterize the average long-term behavior of kanban-controlled manufacturing systems (steady-state analysis). This information, for example, is needed by optimization procedures constructed to identify kanban configurations (the number of kanbans for each product in each stage) that provide a desired level of service at the lowest possible costs. Another fruitful application of the models is the systematic analysis of system behavior. Questions such as "how does the performance of a system change when a given parameter is increased or decreased" are asked and answered in this type of analysis. The insights gained in studying the effects of systematic changes are essential for understanding a system, and they may prove instrumental in designing and fine-tuning optimization procedures. In the following sections, we demonstrate how models built with the construction kit may be used to analyze system behavior.

10.1 Single-Stage Single-Product Kanban Systems

For a single-product kanban system with one kanban-controlled manufacturing facility and no or a limited number of backorders ($\lambda^{\text{ext}} = 1.9$, $\mu = 2$; $\rho = 0.95$), we examined the effects of (1) increasing the number of kanbans and (2) increasing the maximum number of backorders, respectively, on four performance measures: (a) the average fraction of served demand, (b) the average fraction of immediately served demand (*average fill rate*), (c) the average inventory level, and (d) the average backorder level (average number of backorders).

In the first experiment, we increased the number of kanbans from 1 to 10 in increments of 1 with $B^{\max} = 2$. The exact values of the four performance measures are indicated in Figures 10.1 and 10.2. The effect of additional kanbans on the performance of the system is not surprising. The number of kanbans is equivalent to the maximum inventory level, or the number of storage positions, in the output store. Hence, when the number of kanbans is increased, more containers with finished products may be put on stock. As a consequence, it is more likely that an arriving demand can be filled from stock, which translates directly into a larger average fraction of served and immediately served demand (Fig. 10.1). On the other hand, the average number of full containers in the output store rises, and the average number of backorders decreases (Fig. 10.2).

In the second experiment, we increased the maximum number of backorders from 0 to 10, while maintaining the number of kanbans at $K = 2$. Here, the average fraction of immediately served demand falls, while the average fraction of served demand rises (Fig. 10.3). Consequently, the average fraction of backordered demand, that is, the fraction of demand served after some waiting time, increases significantly. The average backorder level rises, while the average number of full containers in the output store decreases (Fig. 10.4). The smaller average fraction of immediately served demand may be explained as follows. Since orders are waiting in the system for full containers more frequently (higher average backorder level), more arriving (full) containers are withdrawn immediately from the output store (lower average inventory level). As a result, more new requests find the output store empty on arrival. The higher average fraction of served demand and the higher utilization of the manufacturing facility result from the increased number of direct (active kanbans) and indirect (backorders) production orders in the system. A larger (on average) stack of production orders reduces the probability that the manufacturing facility is forced to idle. Hence, the fraction of time the manufacturing facility is active increases and a larger fraction of the total demand may be served.

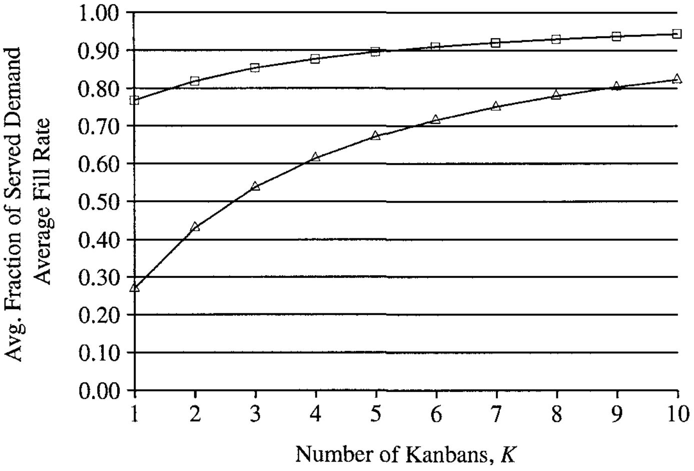


Fig. 10.1. Increasing the number of kanbans, K, while $B^{\max} = 2$; □ = average fraction of served demand, △ = average fraction of immediately served demand (*average fill rate*).

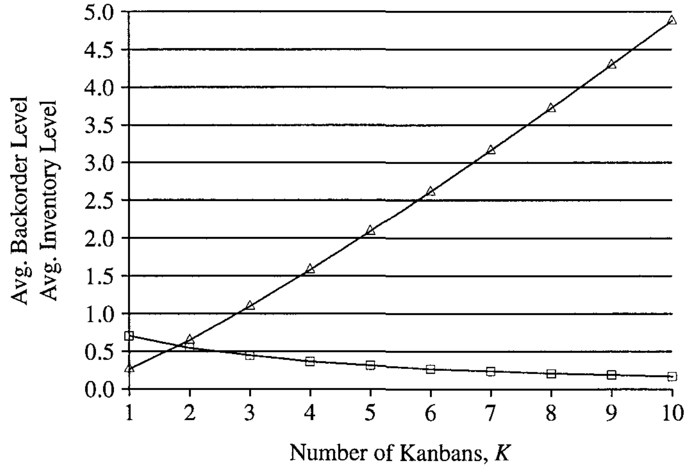


Fig. 10.2. Increasing the number of kanbans, K, while $B^{\max} = 2$; □ = average backorder level, △ = average inventory level.

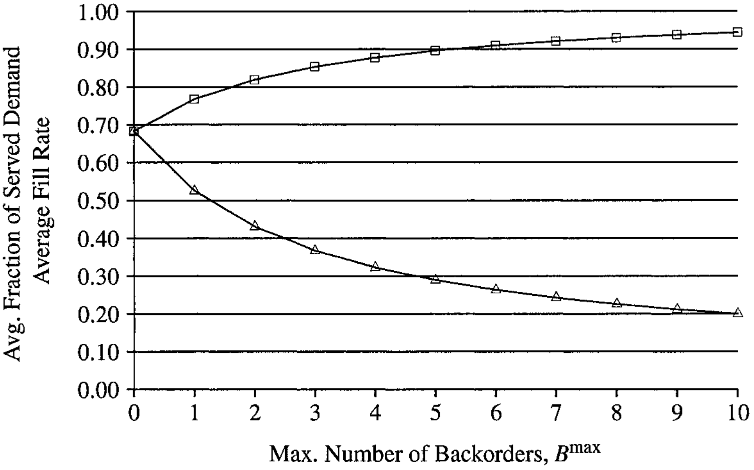


Fig. 10.3. Increasing the maximum number of backorders, B^{max}, while $K = 2$; □ = average fraction of served demand, △ = average fraction of immediately served demand (*average fill rate*).

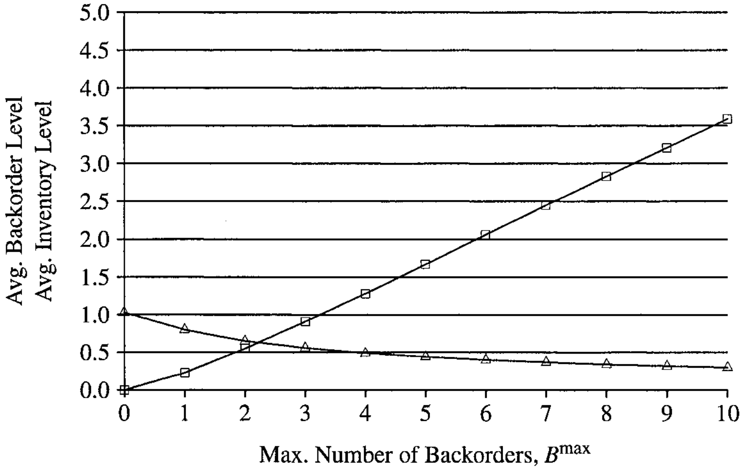


Fig. 10.4. Increasing the maximum number of backorders, B^{max}, while $K = 2$; □ = average backorder level, △ = average inventory level.

10.2 Single-Stage Multi-Product Kanban Systems

For a single-stage kanban system with three products and no backorders $\left(\lambda_i^{\text{ext}} = 0.6, \mu_i = 2, s_i = 1, i = 1,2,3; \rho = 0.90\right)$, we explored the effects of (1) increasing the number of kanbans for one product, increasing the average container processing rate for one product with (2) $K_1 = K_2 = K_3 = 2$ and (3) $K_1 = K_2 = K_3 = 8$, respectively, and decreasing the average setup time for one product with (4) $K_1 = K_2 = K_3 = 2$ and (5) $K_1 = K_2 = K_3 = 8$, respectively, on the average fill rates and the average inventory levels of the three products.

In the first experiment, we increased the number of kanbans for product 1 from 5 to 10 in increments of 1 ($K_2 = K_3 = 5$). Figure 10.5 illustrates the approximations for the average fill rates and the average inventory levels obtained with subassembly SA1. To indicate the accuracy of the approximations, the exact values of the performance measures have been plotted into the same diagrams. The average fill rate and the average inventory level for product 1 increase appreciably, while the same performance measures drop only moderately for products 2 and 3.

In the second and the third experiment, we increased the average container processing rate for product 1 from 2.0 to 4.0 in increments of 0.4 ($\mu_2 = \mu_3 = 2$). In the second experiment, the number of kanbans for each product was two, in the third experiment, the number of kanbans for each product was eight. The values of the performance measures are depicted in Figures 10.6 and 10.7. Interestingly, the average fill rates and the average inventory levels rise for *all* products, although the average container processing rate was only increased for product 1. This effect is due to a smaller average length (in units of time) of the setup cycle, that is, the average time between subsequent opportunities to set up the manufacturing facility for one product. A shorter setup cycle bears the implication that the inventory of a product may be refilled in shorter intervals, or—in other words—that the inventory level at the end of a production run is subject to external demand during a shorter interval of time before it may be replenished. As a consequence, since the inventory level at the end of a production run is determined only by the number of kanbans under the cyclic-exhaustive processing setup change protocol and since this inventory level is thus constant for each product in experiments 2 and 3, the average fill rates and the average inventory levels must rise when the average container processing rate is increased for one product.

Surprisingly, at least at first glance, the increases in the performance measures are somewhat larger for products 2 and 3, although the average container processing rate was increased for product 1. This effect is due to the characteristics of exhaus-

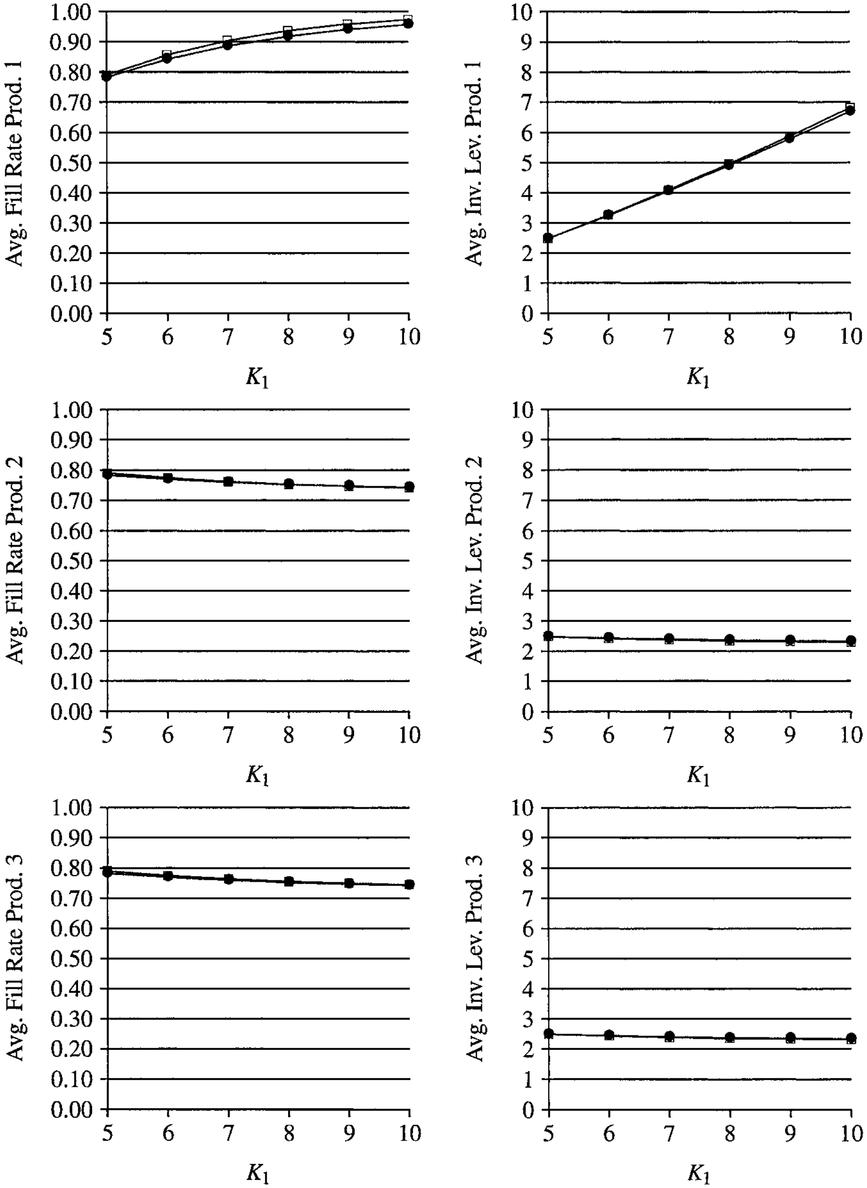


Fig. 10.5. Increasing the number of kanbans for product 1, K_1, while $K_2 = K_3 = 5$; ● = approx. □ = exact value.

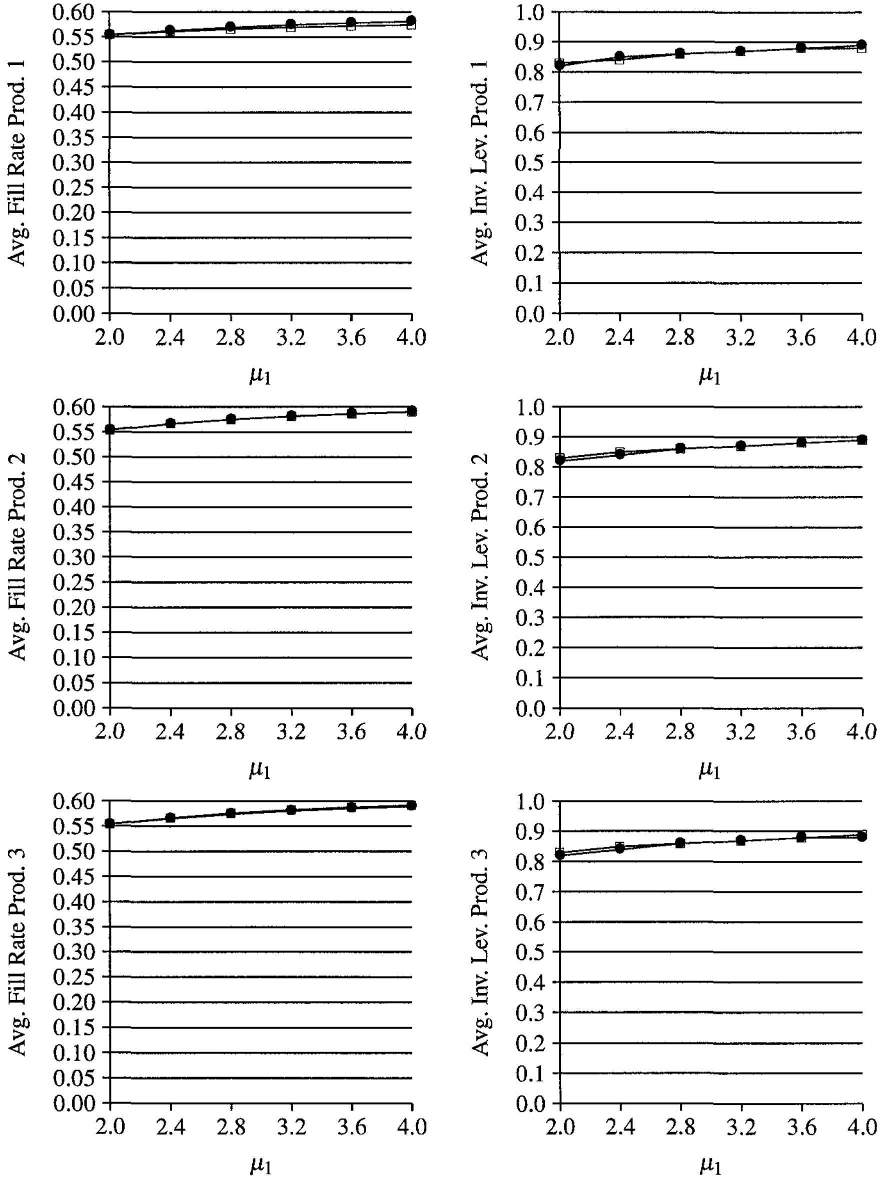


Fig. 10.6. Increasing the average container processing rate for product 1, μ_1, while $\mu_2 = \mu_3 = 2.0$ and $K_1 = K_2 = K_3 = 2$; ● = approx., □ = exact value.

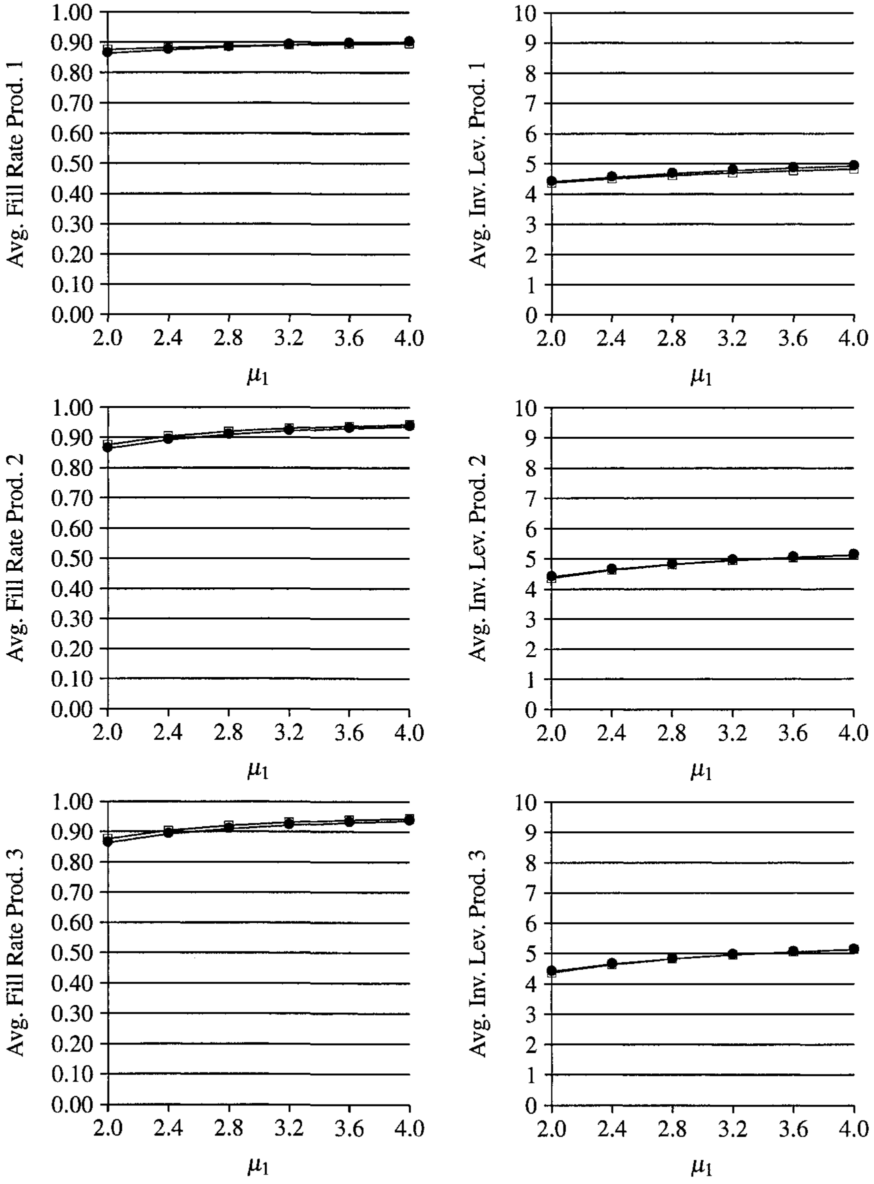


Fig. 10.7. Increasing the average container processing rate for product 1, μ_1, while $\mu_2 = \mu_3 = 2.0$ and $K_1 = K_2 = K_3 = 8$; ● = approx., □ = exact value.

tive processing. Because a production run ends only after the last active kanban has been attached to a full container, the length of a production run is generally longer for a product with a smaller processing rate (products 2 and 3) compared to a product with a larger processing rate (product 1), assuming the same number of kanbans for each product. For a product with a longer production run, a larger proportion of the demand arrives during the production run compared to a product with a shorter production run. The probability that demand can be served immediately is much higher for demand arriving during the production run than for demand arriving during the remaining part of the setup cycle. Thus, all other parameters being equal, the average fill rate of a product with a longer production run must be higher than the average fill rate of a product with a shorter production run. Note that the length of the production run for a product with a smaller average container processing rate is not only longer because of the longer average container processing time, but also because of a compound effect: during a longer production run, more kanbans will be (re-)activated because more demand will arrive (on average). Due to the exhaustive-processing policy, the re-activated kanbans extend the length of the production run further, and even more kanbans may be (re-)activated, which, again, prolong the length of the production run.

In the forth and the fifth experiment, we reduced the average setup time for product 1 from 1.0 to 0.2 in increments of 0.2, while keeping the same parameter constant at 1.0 for products 2 and 3 ($s_2 = s_3 = 1.0$). In the forth experiment, the number of kanbans for each product was two, in the fifth experiment the number of kanbans for each product was eight. The values of the performance measures are indicated in Figures 10.8 and 10.9. The effects on the average fill rates and the average inventory levels are, in principal, identical to the effects observed in experiments 2 and 3: the values for both performance measures rise not only for product 1, but also for the other products. The explanation is similar: reducing the average setup time shortens the average length of the setup cycle, and a shorter setup cycle means that the inventory at the end of a production run must satisfy demand during a shorter period of time (= less demand).

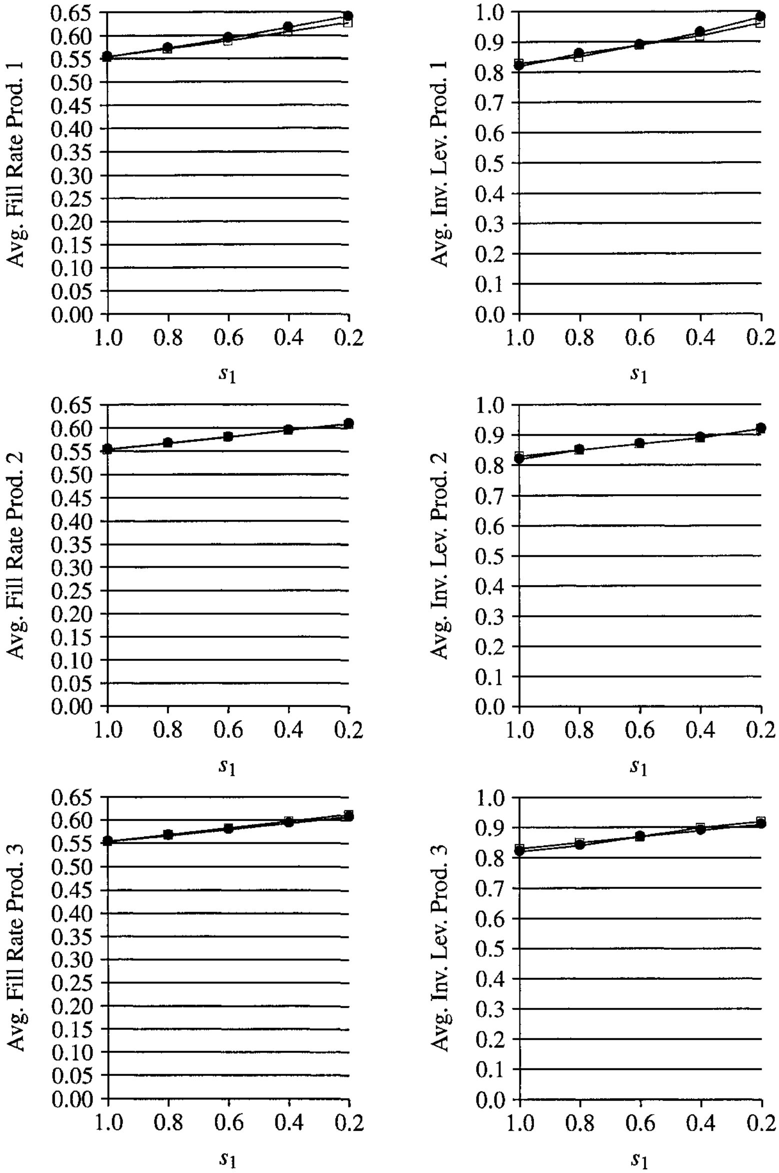


Fig. 10.8. Decreasing the average setup time for product 1, s_1, while $s_2 = s_3 = 1.0$ and $K_1 = K_2 = K_3 = 2$; ● = approx., □ = exact value.

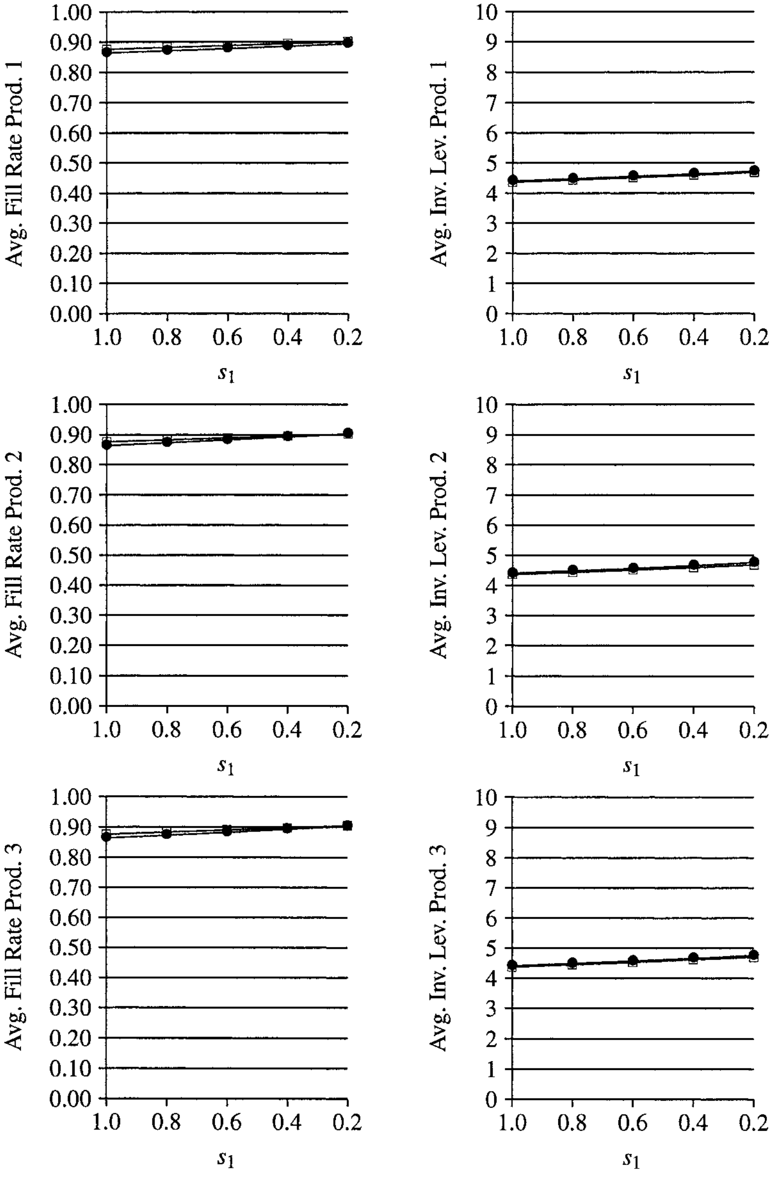


Fig. 10.9. Decreasing the average setup time for product 1, s_1, while $s_2 = s_3 = 1.0$ and $K_1 = K_2 = K_3 = 8$; ● = approx., □ = exact value.

10.3 Two-Stage Multi-Product Kanban Systems

For a completely balanced two-stage kanban system with three products and no or a limited number of backorders $\left(\lambda_i^{\text{ext}} = 0.53,\ \mu_i^{(2)} = 2,\ s_i = 1,\ i = 1,2,3;\ \rho^{(2)} = 0.80\right)$, we investigated the effects of increasing the maximum number of backorders when (1) stage 1 is the bottleneck $\left(\mu_i^{(1)} = 0.67, \text{hence}, \rho_i^{(1)} = 0.80\right)$ and when (2) stage 1 is not the bottleneck $\left(\mu_i^{(1)} = 5.3, \text{hence}, \rho_i^{(1)} = 0.10\right)$ on the four performance measures (a) average fraction of served demand, (b) average fraction of immediately served demand (*average fill rate*), (c) average inventory level, and (d) average backorder level (average number of backorders).

For the first experiment, the approximations—obtained with subassembly SA2 for product 1—are depicted in Figures 10.10 and 10.11. We increased the maximum number of backorders simultaneously for each product from 0 to 18 in increments of 2, while we kept the number of kanbans in stages 1 and 2 constant at $K_1^{(1)} = K_2^{(1)} = K_3^{(1)} = 3$ and $K_1^{(2)} = K_2^{(2)} = K_3^{(2)} = 4$, respectively, to yield average fill rates of about 0.70 for $B_1^{\max,2} = B_2^{\max,2} = B_3^{\max,2} = 0$. The results depicted in Figure 10.10 indicate that the average fraction of served demand hardly changes, whereas the average fraction of immediately served demand (*average fill rate*) drops sharply. At the same time, the average inventory level in stage 1 remains almost constant, while the average inventory level in stage 2 falls drastically and the average backorder level rises progressively (Fig. 10.11).

One would expect the average fraction of served demand to rise more profoundly when the maximum number of backorders is increased. The production runs in stage 2 should become longer because more orders may accumulate between production runs since the maximum number of orders at the start of a production run is $K_i^{(2)} + B_i^{\max,2}$ for product i. Longer production runs mean that the fraction of time that the manufacturing facility in stage 2 actually produces items increases and a smaller fraction of time is used for changing the setup. Consequently, the average production rates for each product should increase, which would then translate directly into larger fractions of served demand. In this example, however, stage 1, which is clearly the bottleneck of the system, sabotages the described mechanism. Production runs in stage 2 must be terminated before all production orders have been processed because stage 1 cannot keep up with the production of the input material for stage 2. As a consequence, the production runs in stage 2 hardly expand when the number of backorders is raised. Average inventory levels in the output store of stage 2 drop sharply because filled containers are often withdrawn instantly to satisfy accumulated backorders.

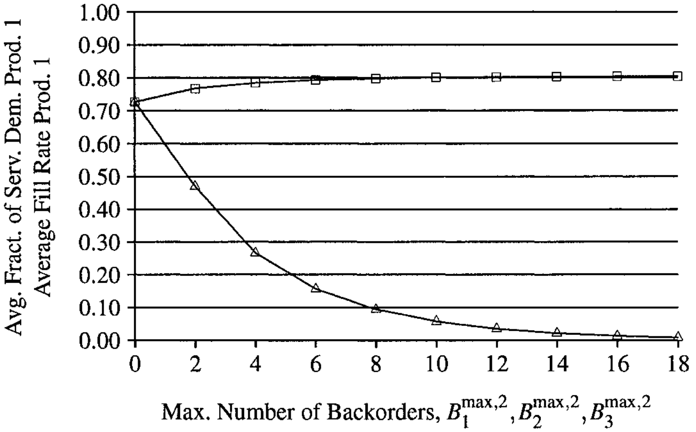


Fig. 10.10. Increasing the maximum number of backorders when stage 1 is the bottleneck $(\rho^{(1)} = 0.80, \rho^{(2)} = 0.80)$; □ = average fraction of served demand for product 1, △ = average fraction of immediately served demand for product 1 (*average fill rate*).

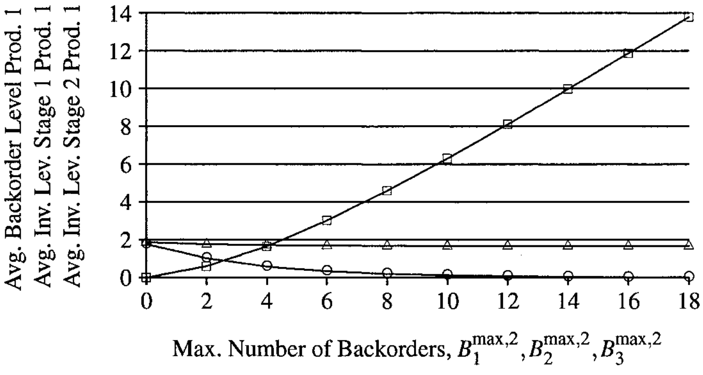


Fig. 10.11. Increasing the maximum number of backorders when stage 1 is the bottleneck $(\rho^{(1)} = 0.80, \rho^{(2)} = 0.80)$; □ = average backorder level for product 1, △ = average inventory level in stage 1 for product 1, and ○ = average inventory level in stage 2 for product 1.

For comparison, we conducted a second experiment in which stage 1 was not the bottleneck of the system. Again, we increased the maximum number of backorders for each product simultaneously from 0 to 18 in increments of 2. We held the number of kanbans constant at $K_1^{(1)} = K_2^{(1)} = K_3^{(1)} = 2$ and $K_1^{(2)} = K_2^{(2)} = K_3^{(2)} = 3$ to yield average fill rates of about 0.70 for $B_1^{\max,2} = B_2^{\max,2} = B_3^{\max,2} = 0$. Figure 10.12 illustrates that in this experiment the average fraction of served demand increases much faster, and the average fill rate falls significantly slower. Moreover, the average number of backorders rises far less dramatically (Fig. 10.13).

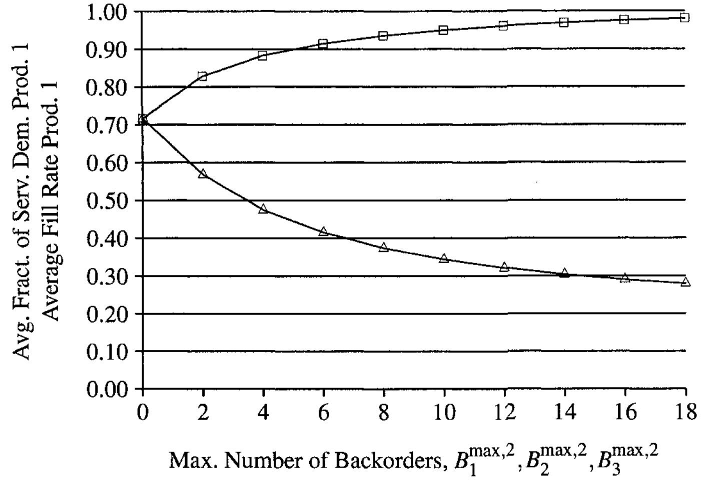


Fig. 10.12. Increasing the maximum number of backorders when stage 1 is not the bottleneck $(\rho^{(1)} = 0.10, \rho^{(2)} = 0.80)$; □ = average fraction of served demand for product 1, △ = average fraction of immediately served demand for product 1 (*average fill rate*).

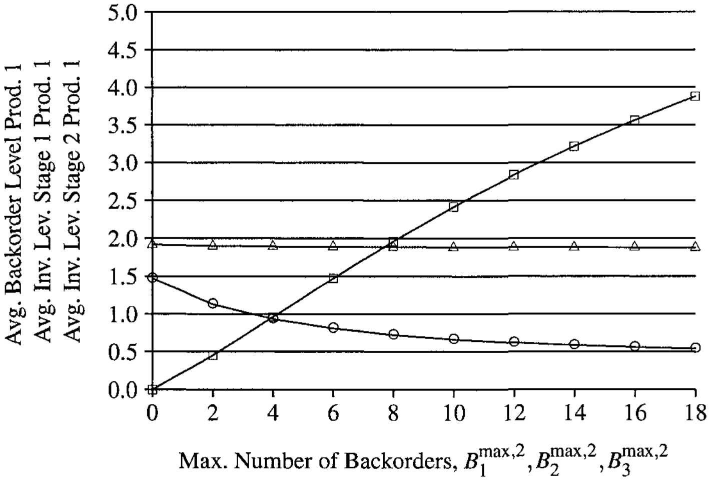


Fig. 10.13. Increasing the maximum number of backorders when stage 1 is not the bottleneck $(\rho^{(1)} = 0.10, \rho^{(2)} = 0.80)$; □ = average backorder level for product 1, △ = average inventory level in stage 1 for product 1, and ○ = average inventory level in stage 2 for product 1.

10.4 Multi-Stage Single-Product Kanban Systems

For a single-product kanban system with three kanban-controlled manufacturing facilities in series and no backorders $\left(\lambda^{\text{ext}} = 1.9,\ \mu^{(m)} = 2,\ m = 1,2,3;\ \rho^{(M)} = 0.95\right)$, we examined the effects of (1) increasing the number of kanbans in (1a) stage 1, (1b) stage 2, and (1c) stage 3 as well as (2) increasing the average container processing rate in (2a) stage 1, (2b) stage 2, and (2c) stage 3 on the four performance measures (a) average fraction of immediately served demand (*average fill rate*), (b) average inventory level in stage 1, (c) average inventory level in stage 2, and (d) average inventory level in stage 3. The experiments of the second set were conducted with two different kanban configurations.

For the first set of experiments, we increased the number of kanbans in one stage from 2 to 10 in increments of 2, while maintaining the number of kanbans in the other two stages at 10. Approximations obtained with the model consisting of three C1-components are depicted in Figures 10.14 and 10.15. Exact values are also included in the figures to indicate the accuracy of the approximation.

In all experiments of the first set, the system shows the same typical behavior: when the bottleneck becomes less severe, the average inventory rises in the bottleneck stage (the stage with fewer kanbans) and in the stages downstream the bottleneck, while in the stages upstream the bottleneck, the average inventory falls. The service level continuously improves with decreasing gains. Note that, although the average processing rates are identical and the processing times follow the same probability distribution, a stage may be a bottleneck solely because of the kanban configuration of the system. This effect is caused by the variability of the processing and demand interarrival times. The fewer kanbans are assigned to a stage, the more likely is a stockout situation, which forces the following stages to idle. The experiments also show that in a multi-stage system, the total average inventory in the system may sometimes be reduced by adding additional kanbans. This effect should be largest if the last stage is the bottleneck because then the average inventory levels in all other stages may be expected to fall when the bottleneck is removed.

For the second set of experiments, we increased the average container processing rate in one stage from 2.0 to 4.0 in increments of 0.4, while keeping the same parameter constant at 2.0 in the other two stages (Figs. 10.16 and 10.17). In these experiments, we set the number of kanbans in the stages at $K^{(1)} = 4$, $K^{(2)} = 5$, and $K^{(3)} = 7$. We chose this kanban configuration because it minimizes the total average inventory level in the system for a desired minimum average fill rate of 0.80.

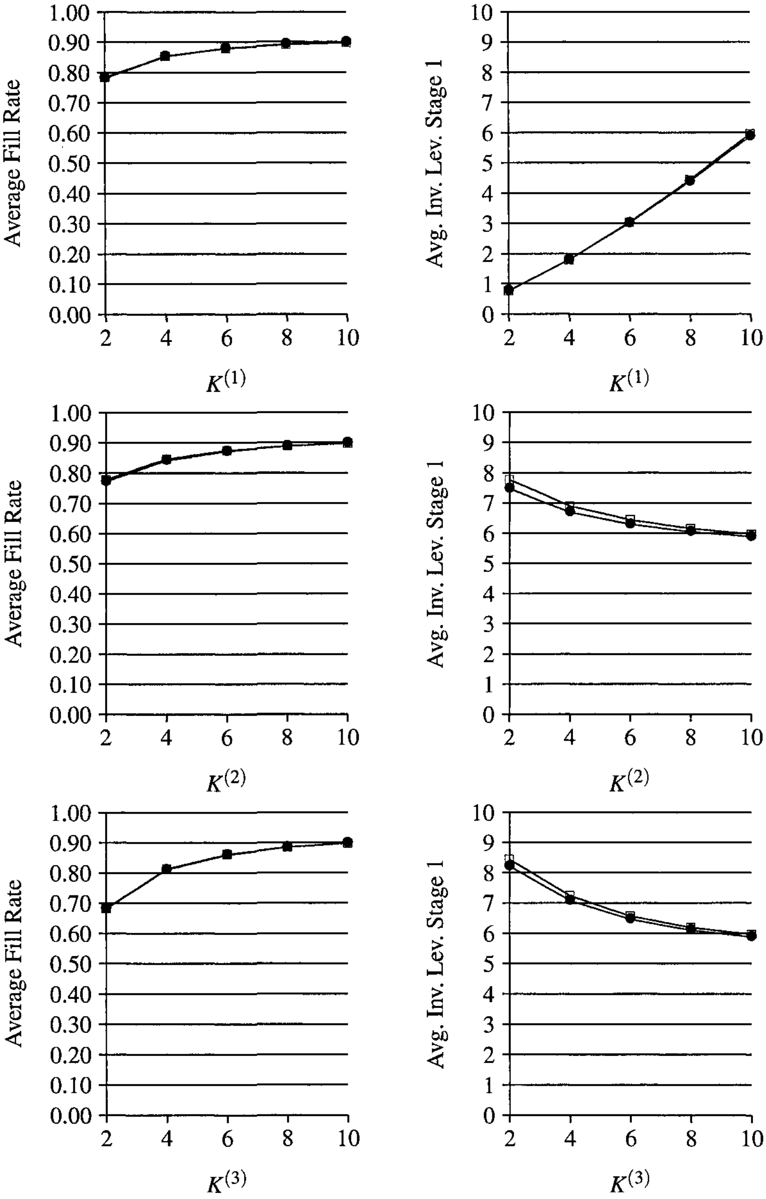


Fig. 10.14. Increasing the number of kanbans in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $K^{(m)} = 10$ in the other stages; ● = approx., □ = exact value.

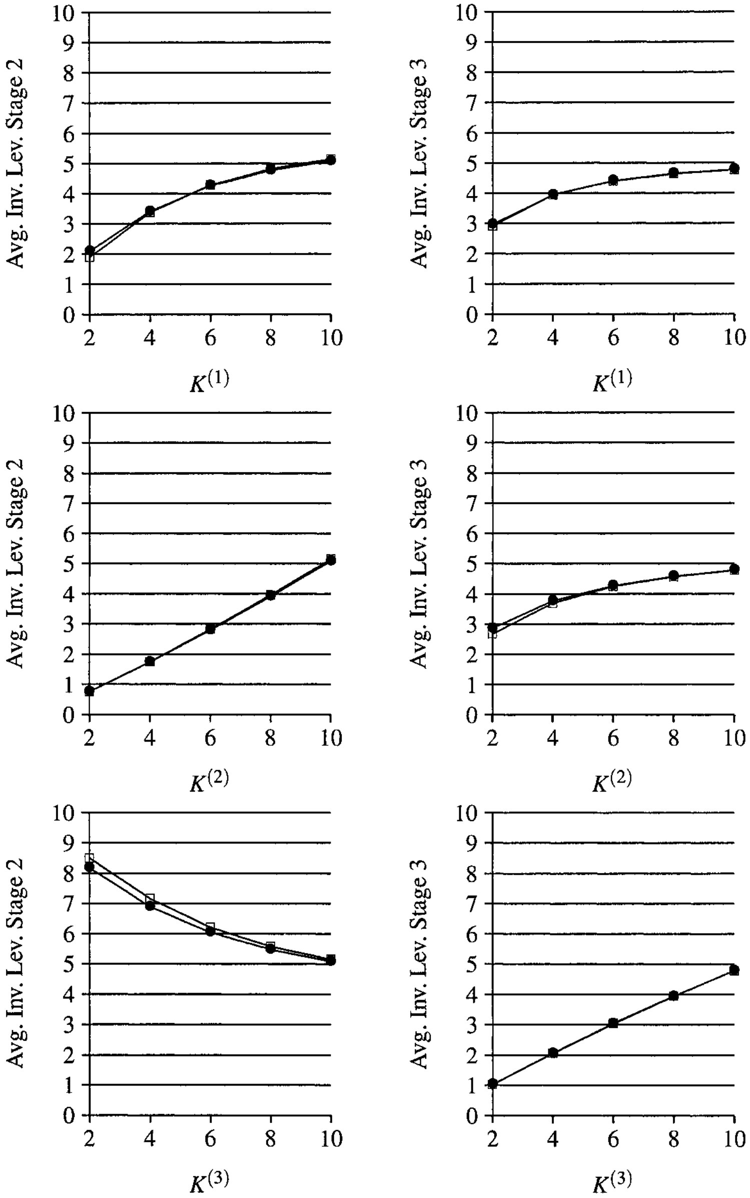


Fig. 10.15. Increasing the number of kanbans in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $K^{(m)} = 10$ in the other stages; ● = approx., □ = exact value.

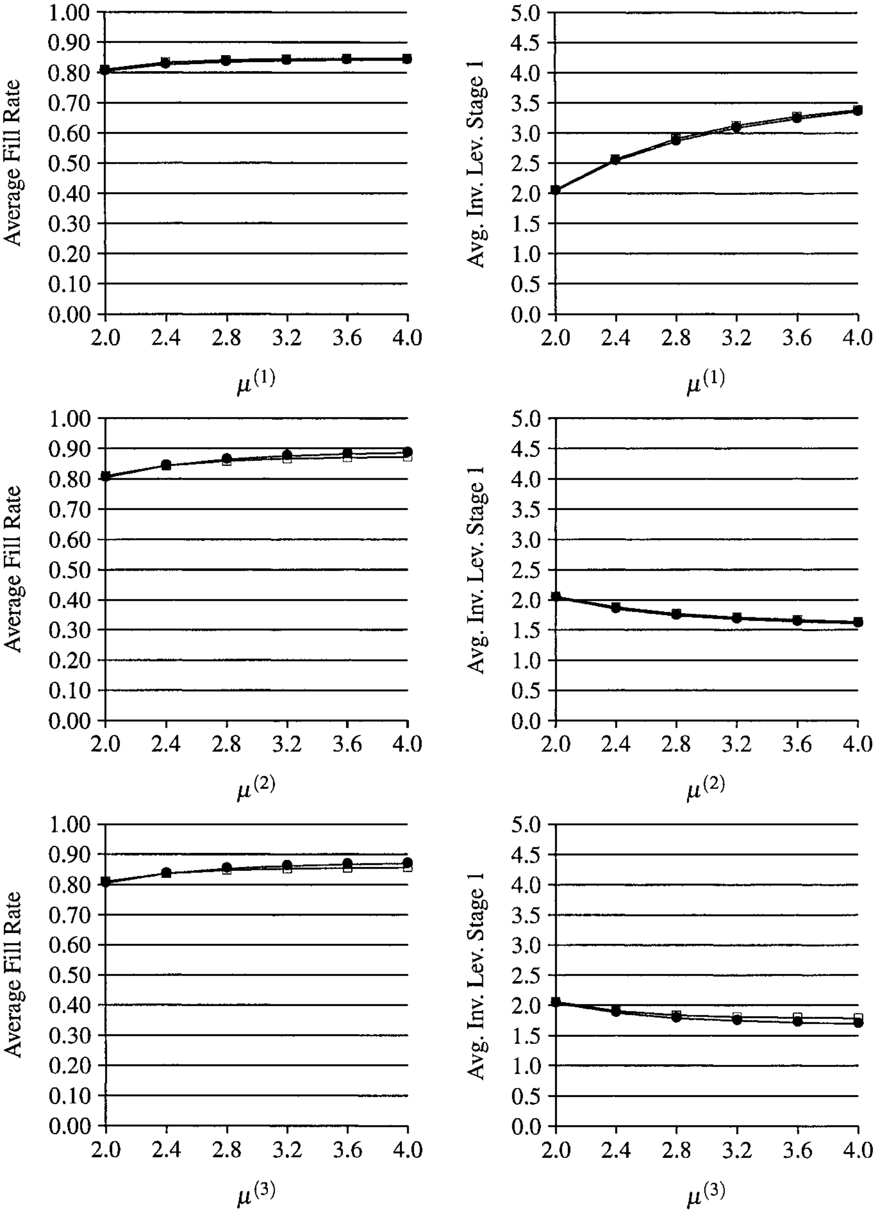


Fig. 10.16. Increasing the average container processing rate in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $\mu^{(m)} = 2.0$ in the other stages and $K^{(1)} = 4$, $K^{(2)} = 5$, and $K^{(3)} = 7$; ● = approx., □ = exact value.

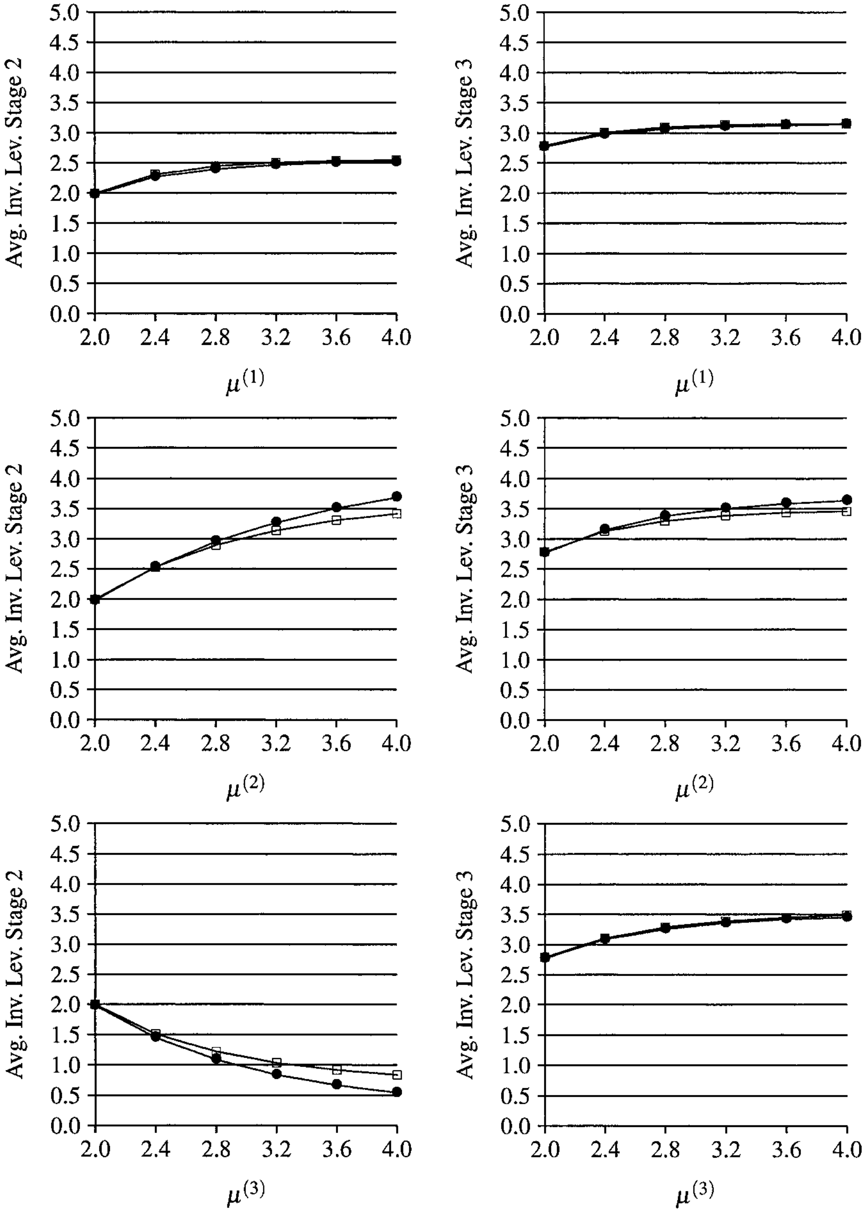


Fig. 10.17. Increasing the average container processing rate in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $\mu^{(m)} = 2.0$ in the other stages and $K^{(1)} = 4$, $K^{(2)} = 5$, and $K^{(3)} = 7$; ● = approx., □ = exact value.

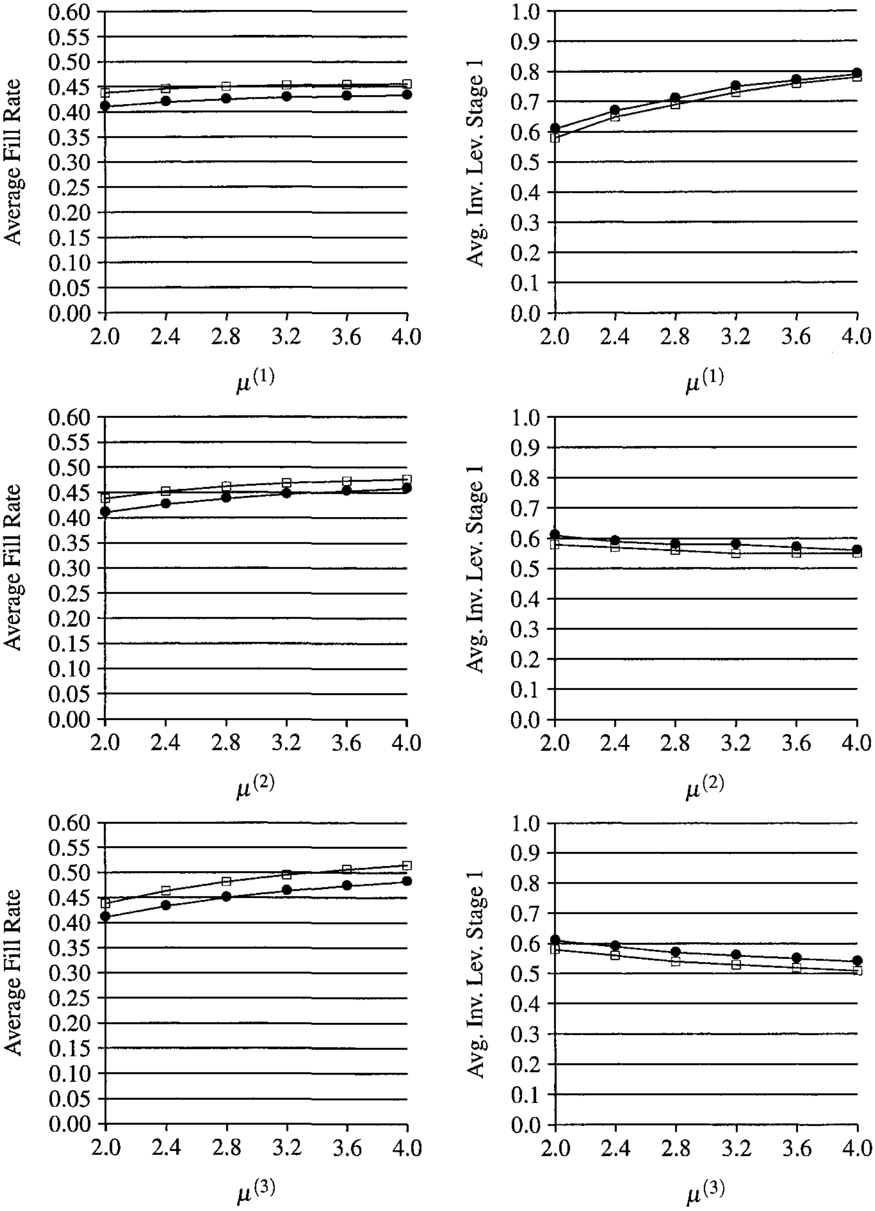


Fig. 10.18. Increasing the average container processing rate in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $\mu^{(m)} = 2.0$ in the other stages and $K^{(1)} = K^{(2)} = K^{(3)} = 1$; ● = approx., □ = exact value.

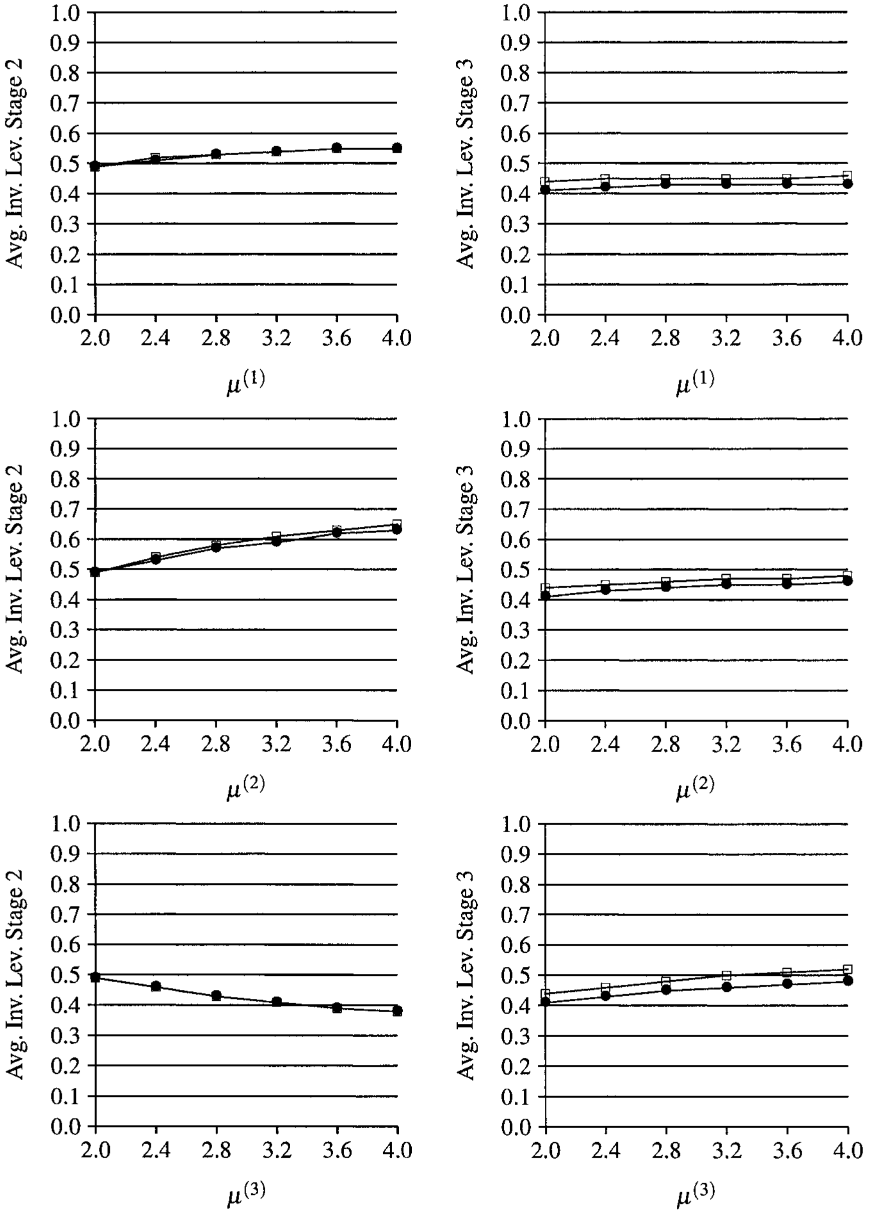


Fig. 10.19. Increasing the average container processing rate in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $\mu^{(m)} = 2.0$ in the other stages and $K^{(1)} = K^{(2)} = K^{(3)} = 1$; • = approx., □ = exact value.

In three additional experiments (Figs. 10.18 and 10.19), we set the number of kanbans in each stage to 1 to examine if the system behaves differently in response to the same changes when the number of kanbans is identical in all stages.

The results show that the system's reactions are the same for both kanban configurations: the average inventory rises in the stage where the average processing rate is increased, and in all stages downstream this stage, whereas the average inventory falls in all stages upstream this stage. The service level improves with diminishing returns. Note that for the system with the same number of kanbans in all stages (Figs. 10.18 and 10.19), increasing the average processing rate in the last stage yields the highest gains in the average fill rate. At the same time, the increases in the average inventory of the last stage are smaller, compared to the increases in the average inventory in the other stages when the processing rate is increased in the other stages. Thus, when manufacturing facilities are identical at the outset, engineering efforts aiming at shortening processing times should start at the end of the line.

10.5 Multi-Stage Multi-Product Kanban Systems

For a three-product kanban system with three kanban-controlled multi-product manufacturing facilities in series and no backorders $\left(\lambda_i^{\text{ext}} = 0.6,\ \mu_i^{(m)} = 2,\ s_i^{(m)} = 1,\ i = 1,2,3 \text{ and } m = 1,2,3;\ \rho^{(m)} = 0.90\right)$, we examined the effects of (1) increasing the number of kanbans simultaneously for all products in (1a) stage 1, (1b) stage 2, and (1c) stage 3, (2) increasing the number of kanbans for one product simultaneously in all stages, and (3) increasing the number of kanbans for one product in (3a) stage 1, (3b) stage 2, and (3c) stage 3 on four groups of performance measures: (a) the average fractions of immediately served demand (*average fill rates*) for products 1, 2, and 3, (b) the average inventory levels in stage 1 for products 1, 2, and 3, (c) the average inventory levels in stage 2 for products 1, 2, and 3, and (d) the average inventory levels in stage 3 for products 1, 2, and 3. The approximate values were obtained with the extended application of the construction kit for kanban systems with several multi-product manufacturing facilities in series (Chapter 8). We also determined point estimates for the same performance measures via simulation to indicate the true behavior of the system. In the simulation experiments, additional replications were made—after a minimum of 25 replications—until each point estimate had a relative error of at most 0.2% at an individual confidence level of (approximately) 99%.

In the experiments of the first set, we increased the number of kanbans for each product in one stage from 2 to 10 in increments of 2, while keeping the number of kanbans for the products in all other stages constant at 10. Figures 10.20 and 10.21 depict approximate values of the performance measures for one product (product 1). Since all parameters in these experiments have identical values for the three products, the values of the performance measures must also be identical, except for minor deviations. It is therefore sufficient to consider the results for one product only.

The first set of experiments corresponds directly to the first set of experiments for the multi-stage single-product system in the preceding section. The results show that the behavior of the multi-product system is almost identical to the behavior of the single-product system: when the bottleneck becomes less severe, the average inventory rises in the bottleneck stage (the stage with fewer kanbans) and in the stages downstream the bottleneck while the average inventory falls in the stages upstream the bottleneck. Also, the service level improves with diminishing returns.

In the second experiment, we increased the number of kanbans for product 1 in all stages simultaneously from 5 to 10 in increments of 1, while maintaining the number of kanbans for the other products in all stages at 5. The approximations and point estimates for the performance measures are given in Figures 10.22 and 10.23.

The second set of experiments corresponds directly to the first experiment for the single-stage multi-product system in Section 10.2, and the observed system behavior is comparable: the average fill rate and the average inventory for product 1 increase appreciably, while the same performance measures drop only moderately for the other two products.

In the last set of experiments, we increased the number of kanbans for product 1 in one stage from 2 to 10 in increments of 2, while keeping the number of kanbans for the other products in this stage and for all products in the other stages constant at 10. Figures 10.24 and 10.25 display the results for the first experiment in this set $\left(K_1^{(1)} = 2,\ldots,10\right)$, Figures 10.26 and 10.27 exhibit the results for the second experiment $\left(K_1^{(2)} = 2,\ldots,10\right)$, and Figures 10.28 and 10.29 illustrate the approximations and point estimates for the third experiment $\left(K_1^{(3)} = 2,\ldots,10\right)$.

The system's behavior in this set of experiments is a combination of the its behavior in the first two sets of experiments: when more kanbans are added for product 1, the average inventory for product 1 rises in the bottleneck stage (the stage with fewer kanbans) and in the stages downstream the bottleneck, and it falls in the stages upstream the bottleneck. At the same time, the average inventories for products 2 and 3 fall in all stages. Likewise, the average fill rates of products 2 and 3 decrease. The average fill rate of product 1, however, always increases.

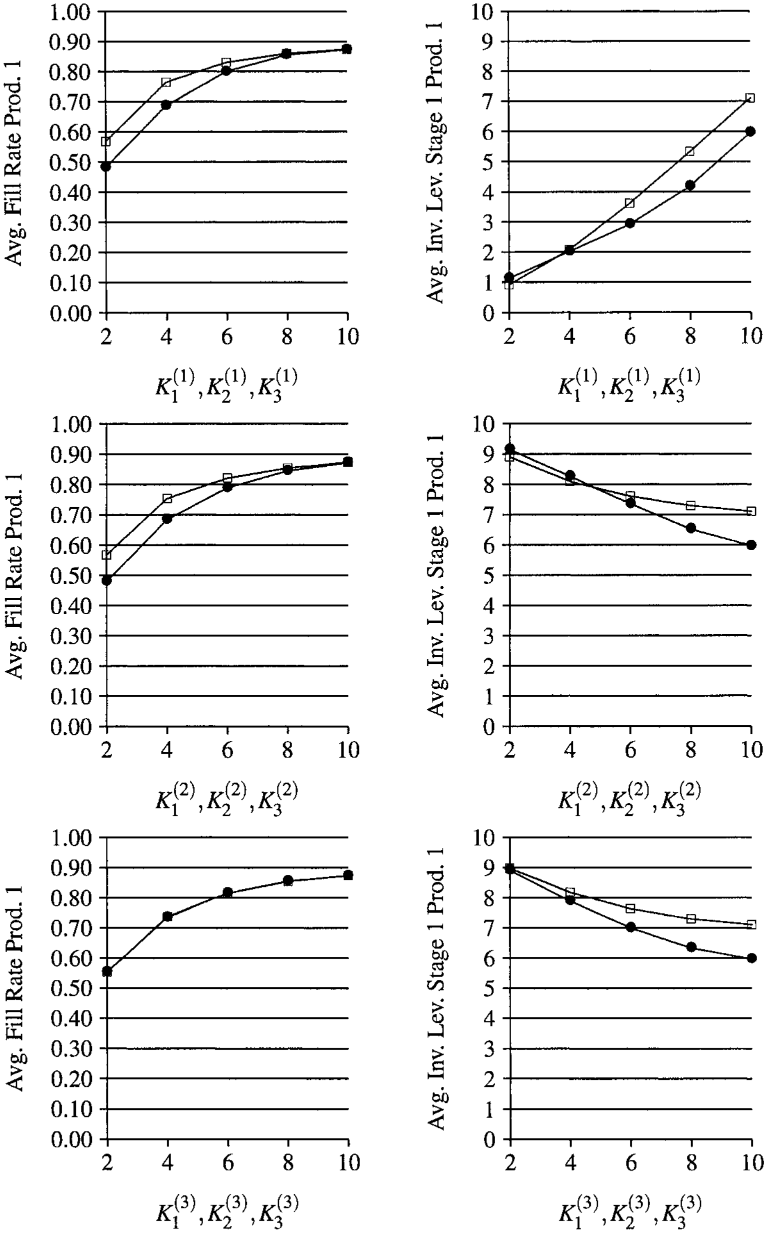


Fig. 10.20. Increasing the number of kanbans for each product in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $K_i^{(m)} = 10$ for each product in the other stages; • = approx., □ = simulation.

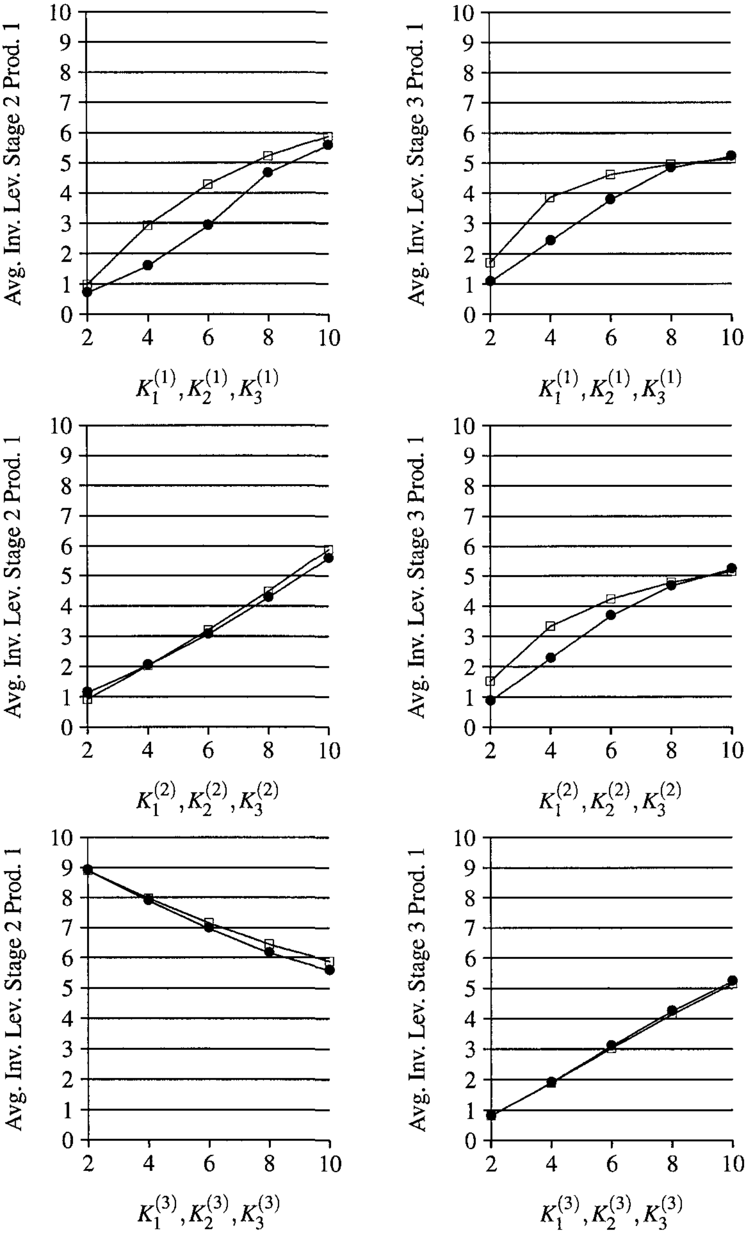


Fig. 10.21. Increasing the number of kanbans for each product in stage 1 (top diagrams), in stage 2 (middle diagrams), and in stage 3 (bottom diagrams), respectively, while $K_i^{(m)} = 10$ for each product in the other stages; ● = approx., □ = simulation.

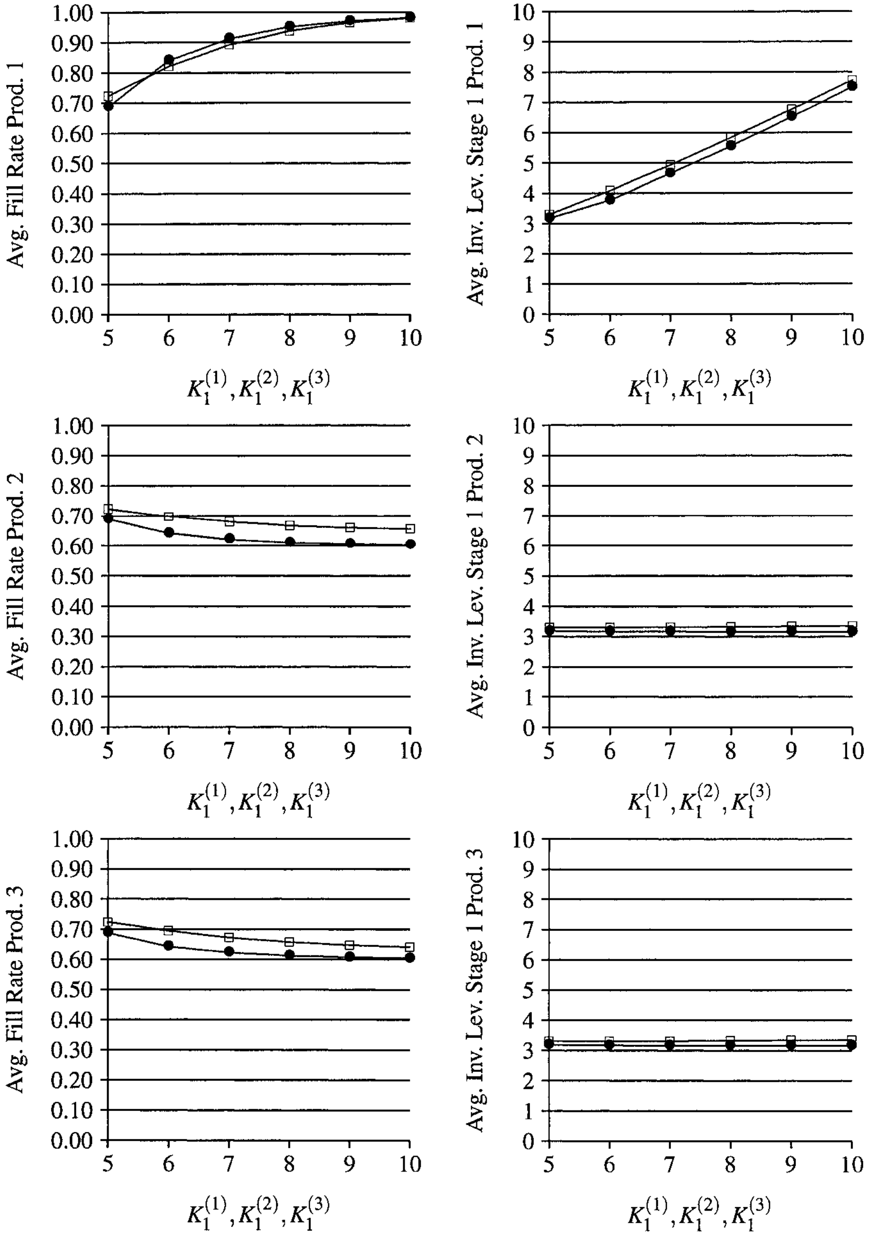


Fig. 10.22. Increasing the number of kanbans for product 1 in all stages, $K_1^{(1)}, K_1^{(2)}, K_1^{(3)}$, while $K_2^{(1)} = K_3^{(1)} = K_2^{(2)} = K_3^{(2)} = K_2^{(3)} = K_3^{(3)} = 5$; ● = approx., □ = simulation.

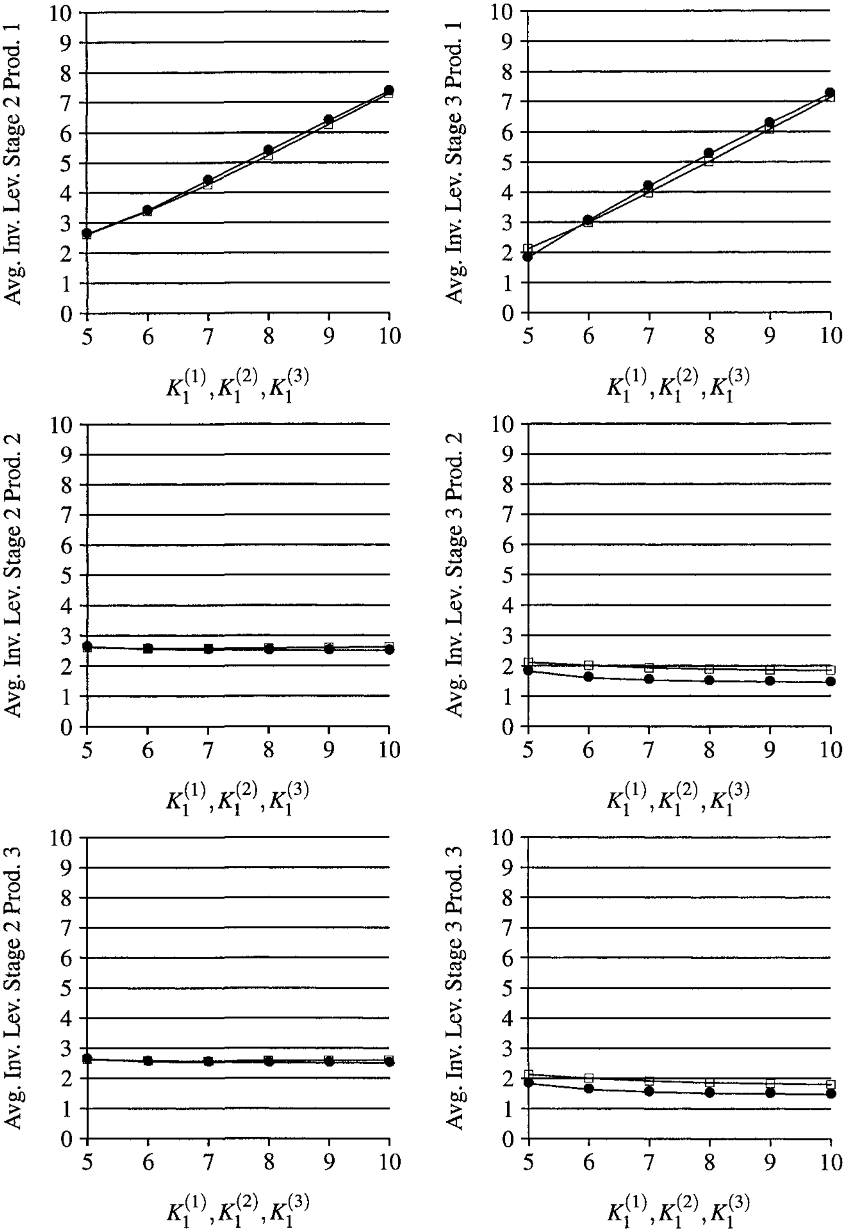


Fig. 10.23. Increasing the number of kanbans for product 1 in all stages, $K_1^{(1)}, K_1^{(2)}, K_1^{(3)}$, while $K_2^{(1)} = K_3^{(1)} = K_2^{(2)} = K_3^{(2)} = K_2^{(3)} = K_3^{(3)} = 5$; ● = approx., □ = simulation.

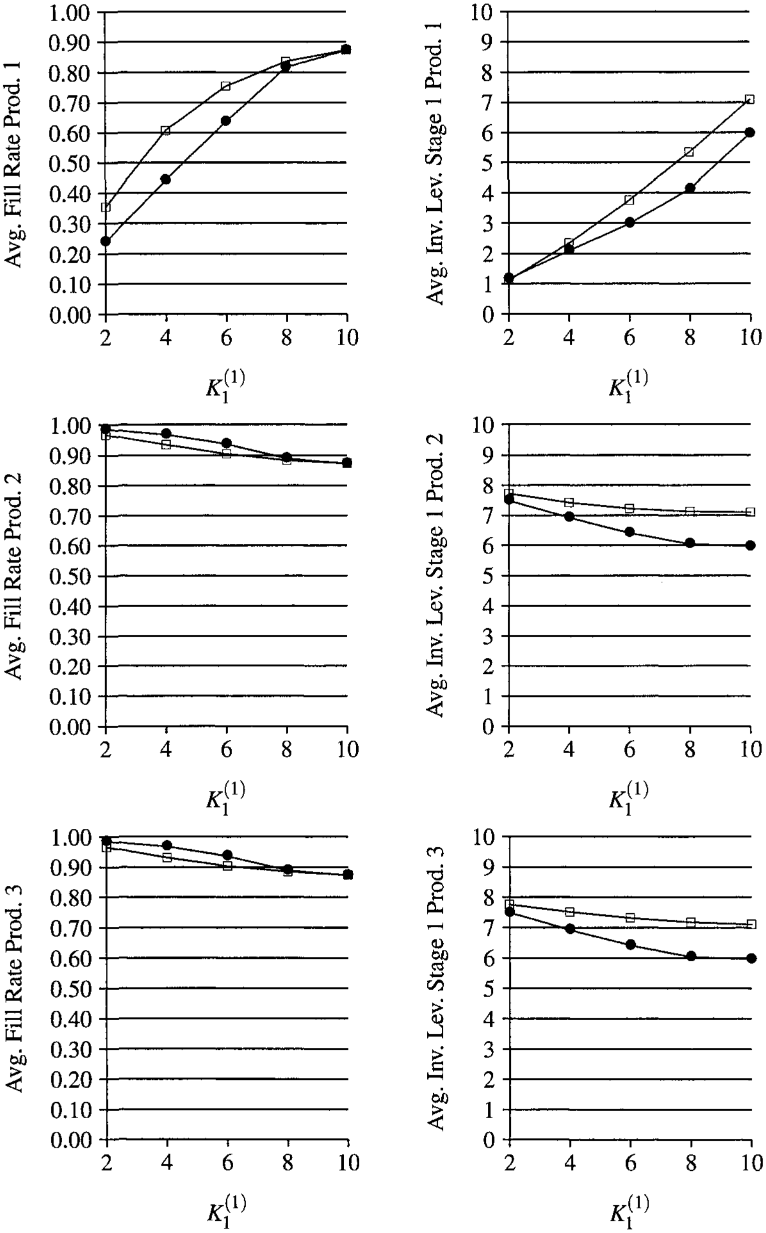


Fig. 10.24. Increasing the number of kanbans for product 1 in stage 1, $K_1^{(1)}$, while $K_2^{(1)} = K_3^{(1)} = K_1^{(2)} = K_2^{(2)} = K_3^{(2)} = K_1^{(3)} = K_2^{(3)} = K_3^{(3)} = 10$; ● = approx., □ = simulation.

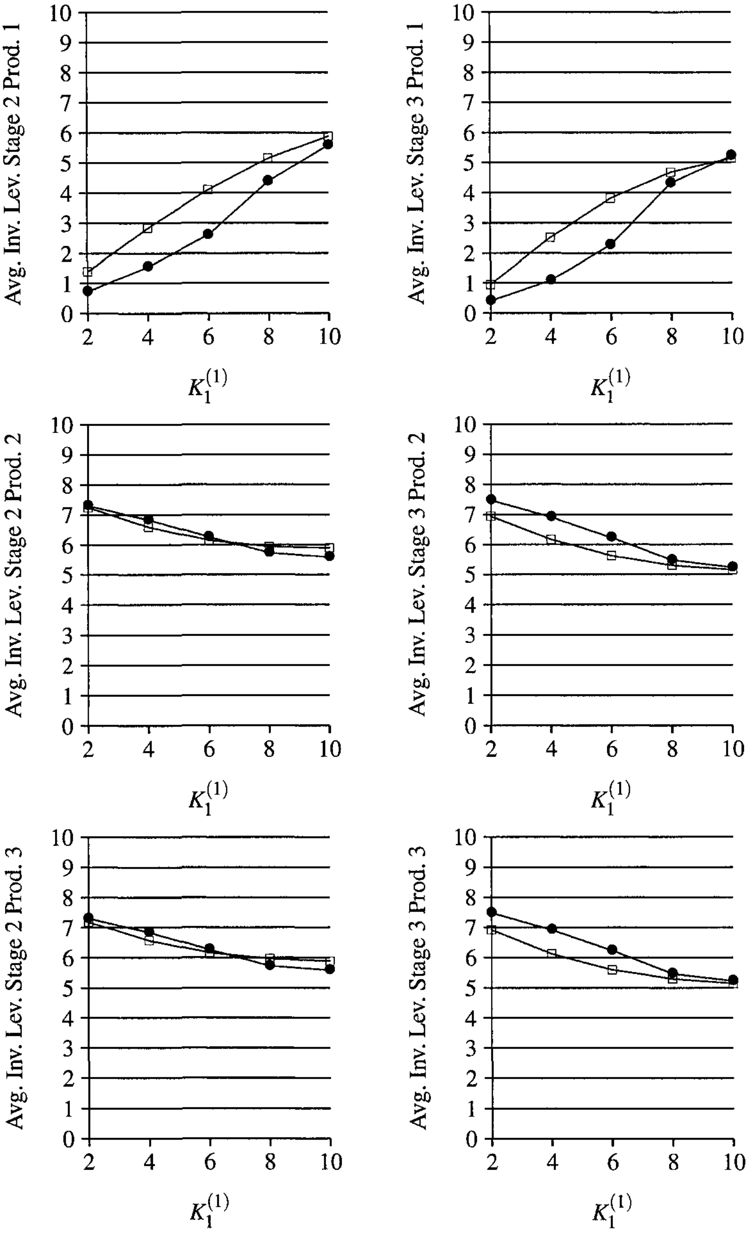


Fig. 10.25. Increasing the number of kanbans for product 1 in stage 1, $K_1^{(1)}$, while $K_2^{(1)} = K_3^{(1)} = K_1^{(2)} = K_2^{(2)} = K_3^{(2)} = K_1^{(3)} = K_2^{(3)} = K_3^{(3)} = 10$; ● = approx., □ = simulation.

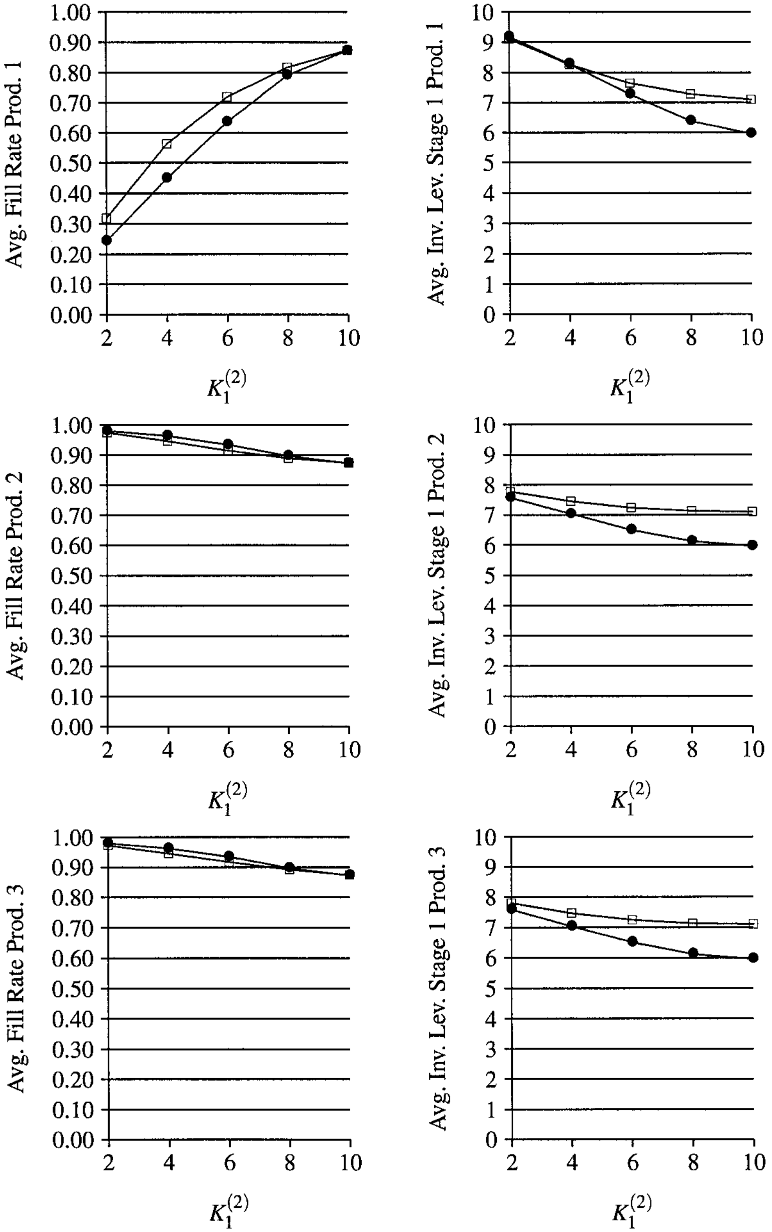


Fig. 10.26. Increasing the number of kanbans for product 1 in stage 2, $K_1^{(2)}$, while $K_1^{(1)} = K_2^{(1)} = K_3^{(1)} = K_2^{(2)} = K_3^{(2)} = K_1^{(3)} = K_2^{(3)} = K_3^{(3)} = 10$; ● = approx., □ = simulation.

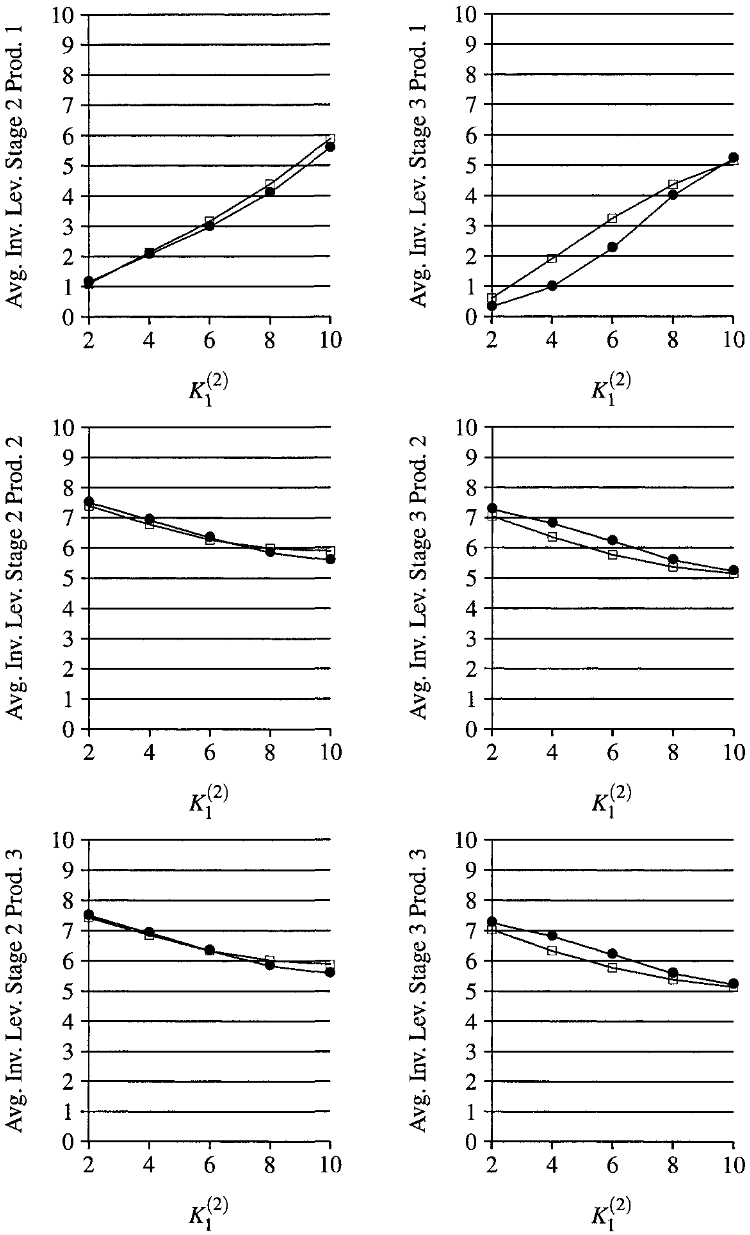


Fig. 10.27. Increasing the number of kanbans for product 1 in stage 2, $K_1^{(2)}$, while $K_1^{(1)} = K_2^{(1)} = K_3^{(1)} = K_2^{(2)} = K_3^{(2)} = K_1^{(3)} = K_2^{(3)} = K_3^{(3)} = 10$; • = approx., □ = simulation.

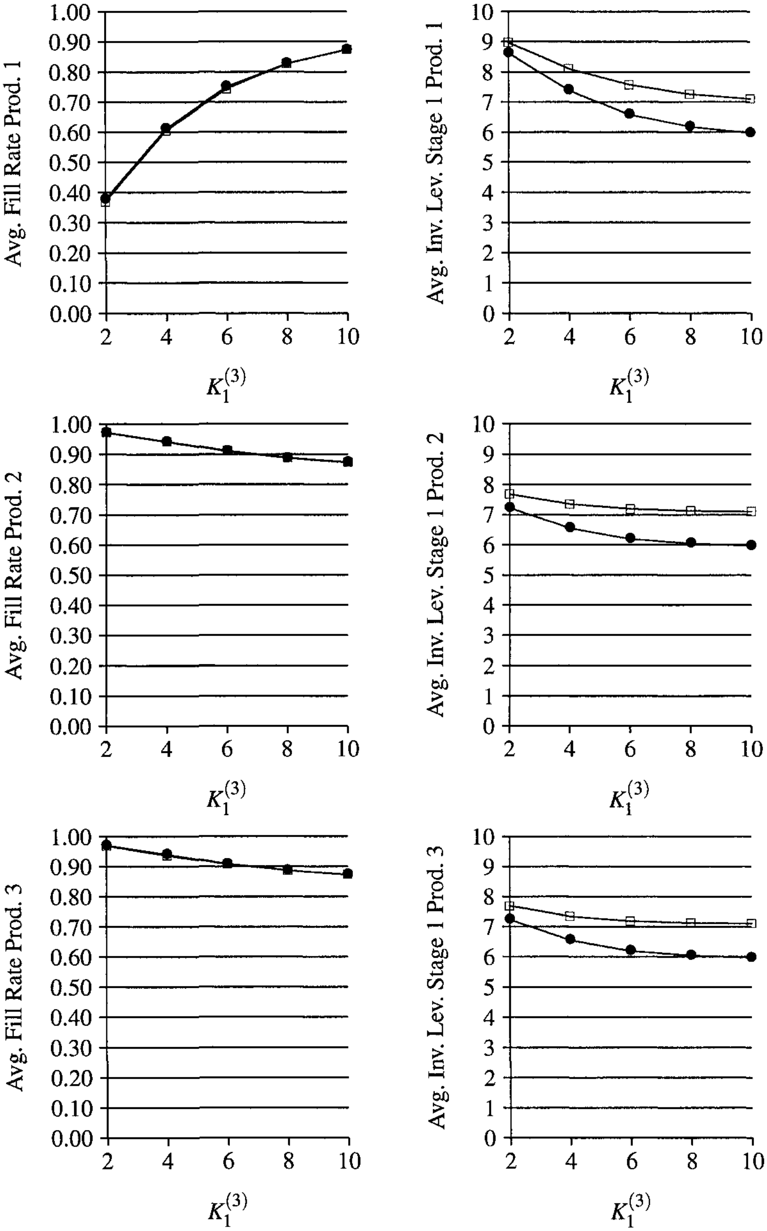


Fig. 10.28. Increasing the number of kanbans for product 1 in stage 3, $K_1^{(3)}$, while $K_1^{(1)} = K_2^{(1)} = K_3^{(1)} = K_1^{(2)} = K_2^{(2)} = K_3^{(2)} = K_2^{(3)} = K_3^{(3)} = 10$; ● = approx., □ = simulation.

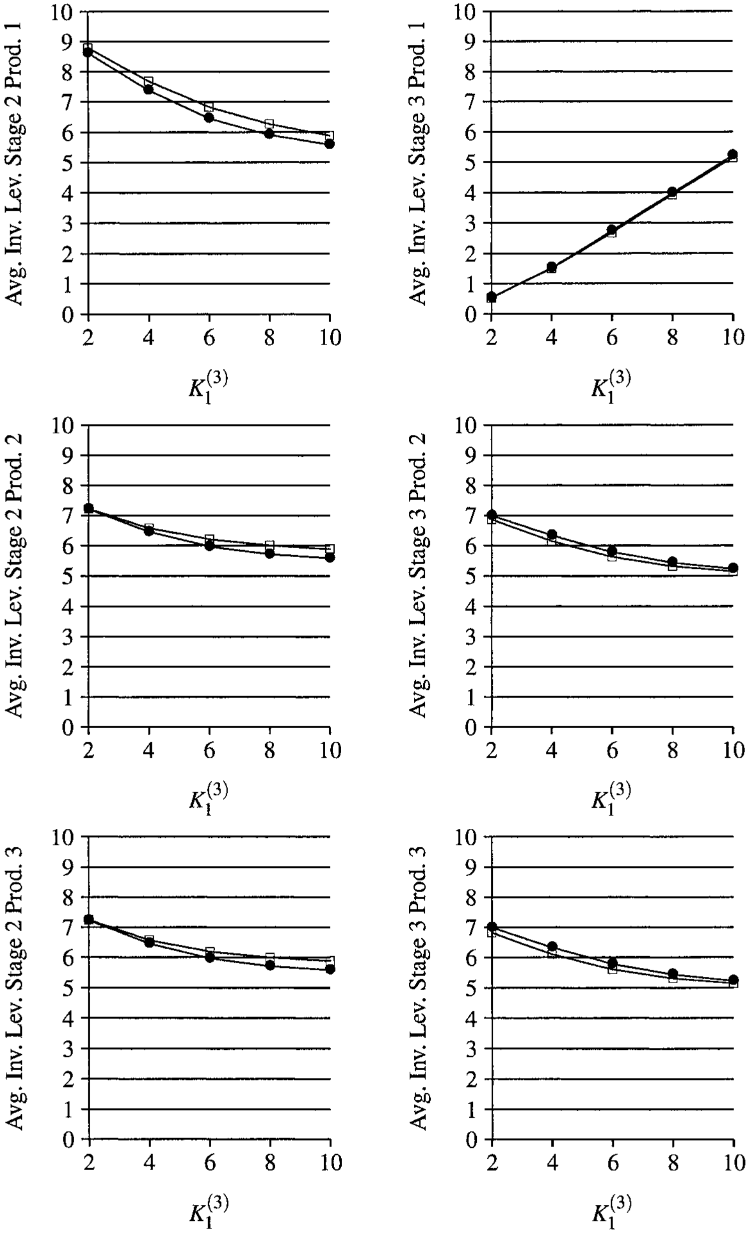


Fig. 10.29. Increasing the number of kanbans for product 1 in stage 3, $K_1^{(3)}$, while $K_1^{(1)} = K_2^{(1)} = K_3^{(1)} = K_1^{(2)} = K_2^{(2)} = K_3^{(2)} = K_2^{(3)} = K_3^{(3)} = 10$; ● = approx., □ = simulation.

It seems that, at least to some extent, the behavior of less complex systems—one product or one stage instead of several—is reflected in the behavior of more complex systems. The analysis of smaller systems may therefore be helpful in understanding the behavior of larger systems.

11

Summary and Directions for Future Research

Sound economic operation of kanban-controlled manufacturing systems requires planning instruments that can help to identify optimal or near-optimal system configurations. In particular, methods are needed to evaluate the performance and operating costs of individual system configurations. While computer simulation might generally be used to perform this task, reliable analyses via simulation are usually very time-consuming, even without counting the time for building and validating the simulation model. The alternative to simulation are analytical (mathematical) models. Most analytical models in the literature, however, may only be used for kanban systems controlling production of a single type of product. As many kanban systems in industry contain manufacturing facilities shared by several different products, there is a need for analytical models that may be employed to evaluate key performance measures of multi-product kanban systems.

In the preceding chapters, we have presented a construction-kit approach that yields new models of single- and multi-stage multi-product kanban systems. The development of basic building blocks ("components") and more complex modules ("subassemblies") along with a technique for linking models of single- and two-stage subsystems allows the analyst to construct models of a large number of different kanban-controlled manufacturing systems with shared manufacturing facilities. Additional construction elements—the result of future research—may be expected to increase the modeling options manifoldly. In the following paragraphs, we hint at some interesting extensions.

Directions for future research. The presented construction-kit approach may be extended and augmented in a number of directions. In the current version, each product is produced and demanded by one manufacturing facility only, and only a single (main) input is permitted for each processing operation. Interesting extensions are therefore building blocks that model merge- (several suppliers for the same input), split- (several consumers of the same product), and assembly-structures (more than one input for a product).

Also, cyclic-exhaustive processing is certainly only one—albeit a reasonable—setup change protocol for multi-product manufacturing facilities with non-negligible setup times. Extensions could, for example, include modules for shared manufacturing facilities with cyclic-gated processing. Under this regime, only active kanbans that were present at the time the setup change or the production run was started are considered in the current production run.

Finally, the assumption of exponential setup and processing times might be too restrictive for many real-life systems, especially if breakdowns and rework do not cause considerable variability in actual container processing times. Future research should therefore be directed at extending the proposed construction elements to cover probability distributions with less variation. A promising path for successful developments represents the method of phases (combining two or more exponential random variables). Extensive numerical tests will be necessary to examine the approximation quality of these extensions and generalizations.

Appendix: Configuration Heuristics

In this appendix, we give detailed descriptions of the local-search heuristics used in generating the test instances in Chapter 9. The common goal of the heuristics is to find the optimal or near-optimal combination of the number of kanbans for each product in each stage that minimizes the total average inventory in the system while guaranteeing certain average fill rates (service requirements).

Starting with the minimum number of kanbans for each product, the heuristics continue to select the "best" neighbor of the current kanban configuration to be the *new* current configuration until the current configuration meets the service requirements. Then the search stops, either immediately (heuristic 1) or after trying to reduce the total average inventory by decreasing the number of kanbans under the condition that the required service level is maintained (heuristics 2–5). The neighborhood of the current kanban configuration is the set of kanban configurations that may be obtained by increasing the number of kanbans for one or all product(s) in one stage by one. Matrix $\mathbf{K}_0$ denotes the current kanban configuration and matrices $\mathbf{K}_1, \ldots, \mathbf{K}_J$ denote its neighbors, where J is the total number of neighbors.

The specifics of the local-search heuristics are given in the following sections. Common elements of the five procedures are the measures that are used to evaluate the neighbors of the current configuration.

The *Total Relevant Improvement* (TRI) expresses the net progress of a neighbor towards the required average fill rates,

$$\mathrm{TRI}_j = \sum_{i=1}^{r} \mathrm{RI}_{ij},$$

where RI_{ij} is the Relevant Improvement of the average fill rate for product i with neighbor $\mathbf{K}_j$, that is, the change of the average fill rate for product i in stage M below or up to the required minimum fill rate for product i in stage M, $f_i^{\min,M}$,

$$\mathrm{RI}_{ij} = \begin{cases} f_{ij}^{(M)} - f_{i0}^{(M)}, & \text{if } f_{i0}^{(M)}, f_{ij}^{(M)} \leq f_i^{\min,M}, \\ f_i^{\min,M} - f_{i0}^{(M)}, & \text{if } f_{i0}^{(M)} \leq f_i^{\min,M} \text{ and } f_{ij}^{(M)} > f_i^{\min,M}, \\ f_{ij}^{(M)} - f_i^{\min,M}, & \text{if } f_{i0}^{(M)} > f_i^{\min,M} \text{ and } f_{ij}^{(M)} \leq f_i^{\min,M}, \\ 0, & \text{if } f_{i0}^{(M)}, f_{ij}^{(M)} > f_i^{\min,M}, \end{cases} \tag{11.1}$$
$$= \min\left\{f_i^{\min,M}, f_{ij}^{(M)}\right\} - \min\left\{f_i^{\min,M}, f_{i0}^{(M)}\right\},$$

where $f_{i0}^{(M)}$ is the average fill rate for product i in stage M with the current kanban configuration, $\mathbf{K}_0$, and $f_{ij}^{(M)}$ is the average fill rate for product i in stage M with neighbor $\mathbf{K}_j$.

Let $\bar{Y}_0$ denote the total average inventory in the system (average number of full containers) with the current kanban configuration, $\mathbf{K}_0$, and let $\bar{Y}_j$ be the total average inventory in the system with neighbor $\mathbf{K}_j$. The increase or decrease in the total average inventory when changing from $\mathbf{K}_0$ to $\mathbf{K}_j$ may then be denoted by $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.

Heuristic 1

This heuristic was used to find optimal or near-optimal kanban configurations for the test instances for subassembly SA1 (Section 9.2). A similar procedure is described in Krieg and Kuhn 2001.

Algorithm of Heuristic 1

Step 1. [Initialization]

Initialize the current kanban configuration, $\mathbf{K}_0$: Set $K_i = 1$ for all $i = 1, \ldots, r$.

Step 2. [Feasibility check]

Evaluate the initial kanban configuration. If it satisfies the service requirements, then STOP. Otherwise, continue with Step 3.

Step 3. [Generation of neighbors]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by increasing the number of kanbans for one product by one.

Generate all neighbors $\mathbf{K}_j$ ($j = 1, \ldots, J$) of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors ($J = r$). Start by increasing the number of kanbans for product 1. Hence, $j = i$, where i is the product index.

Step 4. [Evaluation of the neighbors]

Compute for each neighbor $\mathbf{K}_j$ ($j = 1,\dots,J$):

- The relevant improvement for each product, RI_{ij} ($i = 1,\dots,r$), using Equation (11.1).
- The largest relevant improvement per product, $\mathrm{RI}_j^{\max} = \max\big(\mathrm{RI}_{1j},\dots,\mathrm{RI}_{rj}\big)$.
- The total relevant improvement, $\mathrm{TRI}_j = \sum_{i=1}^{r} \mathrm{RI}_{ij}$.
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.
- The average total relevant improvement per unit of additional total average inventory, $\mathrm{TRI}_j/\Delta\bar{Y}_j$.

Step 5. ["Best" neighbor I: Largest total relevant improvement]

Determine all neighbors $\mathbf{K}_j$ with $\mathrm{TRI}_j/\Delta\bar{Y}_j = \max\big(\mathrm{TRI}_1/\Delta\bar{Y}_1,\dots,\mathrm{TRI}_J/\Delta\bar{Y}_J\big)$ and select the one with the smallest index value j.

If $\mathrm{TRI}_j/\Delta\bar{Y}_j > 0$ for the selected candidate $\mathbf{K}_j$, then set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.

Step 6. ["Best" neighbor II: Largest single relevant inprovement]

Determine all neighbors $\mathbf{K}_j$ with $\mathrm{RI}_j^{\max}/\Delta\bar{Y}_j = \max\big(\mathrm{RI}_1^{\max}/\Delta\bar{Y}_1,\dots,\mathrm{RI}_J^{\max}/\Delta\bar{Y}_J\big)$ and select the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.

Step 7. [Feasibility check]

If the current kanban configuration satisfies the service requirements, then STOP. Otherwise, go to Step 3.

Heuristic 2

This heuristic was used to find optimal or near-optimal kanban configurations for the test instances for subassembly SA2 (Section 9.3).

Algorithm of Heuristic 2

Step 1. [Initialization]

Initialize the current kanban configuration, $\mathbf{K}_0$: Set $K_i^{(m)} = 1$ for all $i = 1,\ldots,$ r and $m = 1,2$.

Step 2. [Feasibility check]

Evaluate the initial kanban configuration. If it satisfies the service requirements, then STOP. Otherwise, continue with Step 3.

Step 3. [Generation of neighbors]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by increasing the number of kanbans for one product in one stage by one.

Generate all neighbors $\mathbf{K}_j$ $(j = 1,\ldots,J)$ of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors $(J = 2r)$. Start by increasing the number of kanbans for product 1 in stage 1. Hence, $j = (m-1)r + i$, where i is the product index and m is the stage index.

Step 4. [Evaluation of the neighbors]

Compute for each neighbor $\mathbf{K}_j$ $(j = 1,\ldots,J)$:

- The relevant improvement for each product, RI_{ij} $(i = 1,\ldots,r)$, using Equation (11.1).
- The largest relevant improvement per product, $\mathrm{RI}_j^{\max} = \max\bigl(\mathrm{RI}_{1j},\ldots,$ $\mathrm{RI}_{rj}\bigr)$.
- The total relevant improvement, $\mathrm{TRI}_j = \sum_{i=1}^{r} \mathrm{RI}_{ij}$.
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.
- The average total relevant improvement per unit of additional total average inventory, $\mathrm{TRI}_j / \Delta\bar{Y}_j$.

Step 5. ["Best" neighbor I: Largest total relevant improvement]

Determine all neighbors with $\mathrm{TRI}_j \geq 0.0001$.
Let $\mathscr{N}$ denote the set of the index values of these neighbors.
If set $\mathscr{N}$ is empty, then go to Step 6.

For all neighbors $\mathbf{K}_j$ $(j \in \mathscr{N})$, sort the values TRI_j from largest to smallest (ignore identical values) and let $\mathrm{TRI}_{[k]}$ denote the kth largest value, so that $\mathrm{TRI}_{[1]} > \mathrm{TRI}_{[2]} > \mathrm{TRI}_{[3]} > \cdots$.
Set $k = 1$.

[‡] Determine all neighbors $\mathbf{K}_j$ with $\mathrm{TRI}_j = \mathrm{TRI}_{[k]}$ and select the one that results in the smallest increase in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

If for the selected candidate $\mathbf{K}_j$

$$\Delta\bar{Y}_j \leq 0 \quad \text{or} \quad \mathrm{TRI}_j/\Delta\bar{Y}_j \geq 0.5 \max_{\ell \in \mathscr{N}} \left(\mathrm{TRI}_\ell/\Delta\bar{Y}_\ell\right),$$

then set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.
Otherwise, set $k = k+1$ and go to [‡].

Step 6. ["Best" neighbor II: Largest single relevant improvement]

Determine all neighbors $\mathbf{K}_j$ $(j = 1,\ldots,J)$ with $\mathrm{RI}_j^{\max} = \max\left(\mathrm{RI}_1^{\max},\ldots,\mathrm{RI}_J^{\max}\right)$ and select the one that results in the smallest increase in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.

Step 7. [Feasibility check]

If the current kanban configuration satisfies the service requirements, then go to Step 8. Otherwise, go to Step 3.

Step 8. [Check for improvements by reducing the number of kanbans]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by *decreasing* the number of kanbans for one product in one stage by one.

[$\updownarrow$] Generate all neighbors $\mathbf{K}_j$ ($j = 1,\ldots,J$) of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors ($J = 2r$). Start by decreasing the number of kanbans for product 1 in stage 1. Hence, $j = (m-1)r + i$, where i is the product index and m is the stage index.

Compute for each neighbor $\mathbf{K}_j$ ($j = 1,\ldots,J$):

- The relevant improvement for each product, RI_{ij} ($i = 1,\ldots,r$), using Equation (11.1).
- The total relevant improvement, $\mathrm{TRI}_j = \sum_{i=1}^{r} \mathrm{RI}_{ij}$.
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.

Determine the neighbors with $\mathrm{TRI}_j = 0$ (preservation of feasibility). If no neighbor meets this criterion, then STOP.
Otherwise, select from the neighbors with $\mathrm{TRI}_j = 0$ the one that results in the largest reduction in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to [$\updownarrow$].

Heuristic 3

This heuristic was used to find optimal or near-optimal kanban configurations for the test instances for linking C1-components (Section 9.4). Since there is only one product in these systems, no distinction is made between *single* and *total* relevant improvement. Conceptually similar heuristic procedures are suggested by De Araújo, Frein, and Di Mascolo (1995) and Gstettner and Kuhn (1996).

Algorithm of Heuristic 3

Step 1. [Initialization]

Initialize the current kanban configuration, $\mathbf{K}_0$: Set $K^{(m)} = 1$ for all $m = 1,\ldots,M$.

Step 2. [Feasibility check]

Evaluate the initial kanban configuration. If it satisfies the service requirement, then STOP. Otherwise, continue with Step 3.

Step 3. [Generation of neighbors]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by increasing the number of kanbans in one stage by one.

Generate all neighbors $\mathbf{K}_j$ $(j = 1, \ldots, J)$ of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors $(J = M)$. Start by increasing the number of kanbans in stage 1. Hence, $j = m$, where m is the stage index.

Step 4. [Evaluation of the neighbors]

Compute for each neighbor $\mathbf{K}_j$ $(j = 1, \ldots, J)$:

- The relevant improvement, RI_j $(= \mathrm{RI}_{1j})$, using Equation (11.1).
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.
- The average relevant improvement per unit of additional total average inventory, $\mathrm{RI}_j / \Delta\bar{Y}_j$.

Step 5. ["Best" neighbor I: Largest decrease in total average inventory]

If the total average inventory decreases by at least 5% with one or more neighbors, then select the neighbor that results in the largest decrease $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.

Step 6. ["Best" neighbor II: Largest relevant improvement]

Determine all neighbors with $\mathrm{RI}_j \geq 0.0001$.
Let $\mathscr{N}$ denote the set of the index values of these neighbors.

For all neighbors $\mathbf{K}_j$ $(j \in \mathscr{N})$, sort the values RI_j from largest to smallest (ignore identical values) and let $\mathrm{RI}_{[k]}$ denote the kth largest value, so that $\mathrm{RI}_{[1]} > \mathrm{RI}_{[2]} > \mathrm{RI}_{[3]} > \cdots$.

Set $k = 1$.

[‡] Determine all neighbors $\mathbf{K}_j$ with $\mathrm{RI}_j = \mathrm{RI}_{[k]}$ and select the one that results in the smallest increase in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

If for the selected candidate $\mathbf{K}_j$

$$\Delta\bar{Y}_j \leq 0 \quad \text{or} \quad \mathrm{RI}_j/\Delta\bar{Y}_j \geq 0.5 \max_{\ell \in \mathscr{N}} \left(\mathrm{RI}_\ell/\Delta\bar{Y}_\ell\right),$$

then set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.
Otherwise, set $k = k + 1$ and go to [‡].

Step 7. [Feasibility check]

If the current kanban configuration satisfies the service requirement, then go to Step 8. Otherwise, go to Step 3.

Step 8. [Check for improvements by reducing the number of kanbans]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by *decreasing* the number of kanbans in one stage by one.

[↕] Generate all neighbors $\mathbf{K}_j$ ($j = 1, \ldots, J$) of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors ($J = M$). Start by decreasing the number of kanbans in stage 1. Hence, $j = m$, where m is the stage index.

Compute for each neighbor $\mathbf{K}_j$ ($j = 1, \ldots, J$):

- The relevant improvement, RI_j ($= \mathrm{RI}_{1j}$), using Equation (11.1).

- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.

Determine the neighbors with $\mathrm{RI}_j = 0$ (preservation of feasibility). If no neighbor meets this criterion, then STOP.
Otherwise, select from the neighbors with $\mathrm{RI}_j = 0$ the one that results in the largest reduction in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to [↕].

Heuristic 4

This heuristic was used to find optimal or near-optimal kanban configurations for the test instances with balanced stages for the extended application of the construction kit (Section 9.5.1).

Algorithm of Heuristic 4

Step 1. [Initialization]

Initialize the current kanban configuration, $\mathbf{K}_0$:
Set $K_i^{(m)} = 2$ for all $i = 1,\dots,r$ and $m = 1,\dots,M-1$.
Set $K_i^{(m)} = 1$ for all $i = 1,\dots,r$ and $m = M$.

Step 2. [Feasibility check]

Evaluate the initial kanban configuration. If it satisfies the service requirements, then STOP. Otherwise, continue with Step 3.

Step 3. [Generation of neighbors]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by simultaneously increasing the number of kanbans for all products in one stage by one.

Generate all neighbors $\mathbf{K}_j$ $(j = 1,\dots,J)$ of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors $(J = M)$. Start by increasing the number of kanbans for all products in stage 1. Hence, $j = m$, where m is the stage index.

Step 4. [Evaluation of the neighbors]

Compute for each neighbor $\mathbf{K}_j$ $(j = 1,\dots,J)$:

- The relevant improvement for each product, RI_{ij} $(i = 1,\dots,r)$, using Equation (11.1).
- The largest relevant improvement per product, $\mathrm{RI}_j^{\max} = \max\bigl(\mathrm{RI}_{1j},\dots,\mathrm{RI}_{rj}\bigr)$.
- The total relevant improvement, $\mathrm{TRI}_j = \sum_{i=1}^{r} \mathrm{RI}_{ij}$.
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.

- The average total relevant improvement per unit of additional total average inventory, $\mathrm{TRI}_j/\Delta\bar{Y}_j$.

Step 5. ["Best" neighbor I: Largest decrease in total average inventory]

If the total average inventory decreases by at least 5% with one or more neighbors, then select the neighbor that results in the largest decrease $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.

Step 6. ["Best" neighbor II: Largest total relevant improvement]

Determine all neighbors with $\mathrm{RI}_j^{\max} \geq 0.0001$.
Let $\mathscr{N}$ denote the set of the index values of these neighbors.

For all neighbors $\mathbf{K}_j$ ($j \in \mathscr{N}$), sort the values TRI_j from largest to smallest (ignore identical values) and let $\mathrm{TRI}_{[k]}$ denote the kth largest value, so that $\mathrm{TRI}_{[1]} > \mathrm{TRI}_{[2]} > \mathrm{TRI}_{[3]} > \cdots$.

Set $k = 1$.

[‡] Determine all neighbors $\mathbf{K}_j$ with $\mathrm{TRI}_j = \mathrm{TRI}_{[k]}$ and select the one that results in the smallest increase in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

If for the selected candidate $\mathbf{K}_j$

$$\Delta\bar{Y}_j \leq 0 \quad \text{or} \quad \mathrm{TRI}_j/\Delta\bar{Y}_j \geq 0.5 \max_{\ell \in \mathscr{N}} \left(\mathrm{TRI}_\ell/\Delta\bar{Y}_\ell\right),$$

then set $\mathbf{K}_0 = \mathbf{K}_j$ and go to Step 7.
Otherwise, set $k = k+1$ and go to [‡].

Step 7. [Feasibility check]

If the current kanban configuration satisfies the service requirements, then go to Step 8. Otherwise, go to Step 3.

Step 8. [Check for improvements by reducing the number of kanbans]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by *decreasing* the number of kanbans for all products in one stage by one.

[↕] Generate all neighbors $\mathbf{K}_j$ $(j = 1, \ldots, J)$ of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors $(J = M)$. Start by decreasing the number of kanbans for all products in stage 1. Hence, $j = m$, where m is the stage index.

Compute for each neighbor $\mathbf{K}_j$ $(j = 1, \ldots, J)$:

- The relevant improvement for each product, RI_{ij} $(i = 1, \ldots, r)$, using Equation (11.1).
- The total relevant improvement, $\mathrm{TRI}_j = \sum_{i=1}^{r} \mathrm{RI}_{ij}$.
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.

Determine the neighbors with $\mathrm{TRI}_j = 0$ (preservation of feasibility). If no neighbor meets this criterion, then STOP.
Otherwise, select from the neighbors with $\mathrm{TRI}_j = 0$ the one that results in the largest reduction in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$ and go to [↕].

Heuristic 5

This heuristic was used to find optimal or near-optimal kanban configurations for the test instances with unbalanced stages for the extended application of the construction kit (Section 9.5.2).

Algorithm of Heuristic 5

Step 1. [Initialization]

Initialize the current kanban configuration, $\mathbf{K}_0$:
Set $K_i^{(m)} = 2$ for all $i = 1, \ldots, r$ and $m = 1, \ldots, M-1$.
Set $K_i^{(m)} = 1$ for all $i = 1, \ldots, r$ and $m = M$.

Step 2. [Feasibility check]

Evaluate the initial kanban configuration. If it satisfies the service requirements, then STOP. Otherwise, continue with Step 3.

Step 3. [Generation of neighbors]

Define the neighborhood of the current kanban configuration to be the set of kanban configurations that may be obtained by increasing the number of kanbans for one product in one stage by one.

Generate all neighbors $\mathbf{K}_j$ $(j = 1,\dots,J)$ of the current kanban configuration, $\mathbf{K}_0$, where J is the total number of neighbors $(J = Mr)$. Start by increasing the number of kanbans for product 1 in stage 1. Hence, $j = (m-1)r + i$, where i is the product index and m is the stage index.

Step 4. [Evaluation of the neighbors]

Compute for each neighbor $\mathbf{K}_j$ $(j = 1,\dots,J)$:

- The relevant improvement for each product, RI_{ij} $(i = 1,\dots,r)$, using Equation (11.1).
- The largest relevant improvement per product, $\mathrm{RI}_j^{\max} = \max\left(\mathrm{RI}_{1j},\dots,\mathrm{RI}_{rj}\right)$.
- The change in the total average inventory, $\Delta\bar{Y}_j = \bar{Y}_j - \bar{Y}_0$.

Step 5. ["Best" neighbor: Largest single relevant improvement]

Determine all neighbors $\mathbf{K}_j$ $(j = 1,\dots,J)$ with $\mathrm{RI}_j^{\max} = \max\left(\mathrm{RI}_1^{\max},\dots,\mathrm{RI}_J^{\max}\right)$ and select the one that results in the smallest increase in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j.

Set $\mathbf{K}_0 = \mathbf{K}_j$.

Step 6. [Feasibility check]

If the current kanban configuration does not satisfy the service requirements, then go to Step 3.

Otherwise, check if different neighbors of the last current configuration satisfy the service requirements with less additional total average inventory. If there are such neighbors, then select (from those) the one that results in the smallest increase in total average inventory $\left(\text{smallest } \Delta\bar{Y}_j\right)$. If two or more neighbors fit this description, then select (from those) the one with the smallest index value j. Substitute this neighbor for the current configuration and STOP.

Symbols and Abbreviations

An estimate is indicated by a hat, for example, $\hat{f}$ denotes an estimate for f. The tag "init" marks an initial value, the tag "est" labels a rough estimate.

A	The event that no kanban is active for product j.
B	The event that the manufacturing facility is dedicated to product i.
$B_{[i]}$	Busy period (production run) [for product i].
$B_{[i]}^{\max\langle,m\rangle}$	Maximum number of backorders [for product i] $\langle$in stage $m\rangle$.
$b_{[i]}^{\langle(m)\rangle}$	Backorder level (number of backorders) [for product i] $\langle$in stage $m\rangle$.
$\bar{b}_{[i]}^{\langle(m)\rangle}$	Average backorder level [for product i] $\langle$in stage $m\rangle$.
$\bar{b}_i^{O:M}$	Average backorder level for product i in the last stage of the original system.
$\bar{b}_i^{S:M}$	Average backorder level for product i in the last stage of the substitute system.
C	Component.
C1(m)	Component C1 for stage m.
C2(i)	Component C2 for product i.
CTMC	Continuous-time Markov chain.
E_i	The event that no product meets the setup condition at the end of a busy period for product i.

E_{ij} The event that product j does not meet the setup condition at the end of a busy period for product i.

$E_{ij}^{n_j=0}$ The event that $n_j = 0$ at the end of a busy period for product i.

$E_{ij}^{n_j=0 \wedge y_j>0}$ The event that $n_j = 0$ and $y_j > 0$ at the end of a busy period for product i.

E_{ij}^{n} Substitute for $E_{ij}^{n_j=0 \wedge y_j>0}$.

$E_{ij}^{n_j>0 \wedge y_j=0}$ The event that $n_j > 0$ and $y_j = 0$ at the end of a busy period for product i.

E_{ij}^{y} Substitute for $E_{ij}^{n_j>0 \wedge y_j=0}$.

$E_{ij}^{n_j=0 \wedge y_j=0}$ The event that $n_j = 0$ and $y_j = 0$ at the end of a busy period for product i.

$E_{ij}^{n,y}$ Substitute for $E_{ij}^{n_j=0 \wedge y_j=0}$.

$E_{ij}^{n_j=0 \vee y_j=0}$ The event that $n_j = 0$ or $y_j = 0$ (inclusive *or*) at the end of a busy period for product i.

$E_{ij}^{y_j=0}$ The event that $y_j = 0$ at the end of a busy period for product i.

E_k Erlang-k distribution/Erlang distribution with k phases.

e Euler constant/base of natural logarithm ($e \approx 2.718$).

ε_p Constant for stopping criterion (coordination of the single-*p*roduct components): the relative change of each performance measure must be less than this value.

ε_s Constant for stopping criterion (coordination of the *s*tages): either the absolute value of the relative difference between the average production rate of stage 1 and any other stage must be less than this value, or the relative change of the average production rate of each stage must be less than $0.1\varepsilon_s$.

$f_{[i]}^{\langle(m)\rangle}$ Average fill rate [for product i] $\langle$in stage $m\rangle$.

$\hat{f}_i^{(k)}$ Value of $\hat{f}_i$ at the end of the kth rotation.

$f_{[i]}^{\min\langle,m\rangle}$ Minimum required average fill rate [for product i] $\langle$in stage $m\rangle$.

$f_{i0}^{(M)}$ Average fill rate for product i in stage M with the current kanban configuration, $\mathbf{K}_0$.

$f_{ij}^{(M)}$ Average fill rate for product i in stage M with neighbor $\mathbf{K}_j$.

$f_i^{O:M}$ Average fill rate for product i in the last stage of the original system.

$f_i^{S:M}$ Average fill rate for product i in the last stage of the substitute system.

$f_{\mathrm{BD}[,i]}^{\langle(m)\rangle}$ Average fraction of backordered demand [for product i] $\langle$in stage $m\rangle$.

$f_{\mathrm{ISD}[,i]}^{\langle(m)\rangle}$ Average fraction of immediately served demand (= *average fill rate*) [for product i] $\langle$in stage $m\rangle$.

$f_{\mathrm{LD}[,i]}^{\langle(m)\rangle}$ Average fraction of lost demand [for product i] $\langle$in stage $m\rangle$.

$f_{\mathrm{SD}[,i]}^{\langle(m)\rangle}$ Average fraction of served demand [for product i] $\langle$in stage $m\rangle$.

G General distribution (The random variables are independent and identically distributed).

$\tilde{g}_i(z)$ Steady-state probability distribution of $\{\tilde{Z}_i(t), t \geq 0\}$.

$\tilde{g}_V^{(i)}$ Average fraction of time of a vacation period in the approximate model of component C2 (C3) for product i.

$\tilde{h}_i(y,z)$ Steady-state probability distribution of $\{[\tilde{Y}_i(t), \tilde{Z}_i(t)], t \geq 0\}$.

$I_{[i]}$ Idle period [for product i].

i Product index.

J Total number of neighbors of a kanban configuration.

JIT Just-in-time.

j Index value of a neighbor.

j Product index.

$K_{[i]}^{\langle(m)\rangle}$ Number of kanbans [for product i] $\langle$in stage $m\rangle$.

$K_i^{O:m}$ Number of kanbans for product i in stage m of the original system.

$K_i^{S:m}$ Number of kanbans for product i in stage m of the substitute system.

$K_i^{S:m^+}$ Number of kanbans for product i in stage m^+ of the substitute system.

$\vec{K}^{(m)}$ Vector of the number of kanbans for products $1,\ldots,r$ in stage m.

$\mathbf{K}_0$	Current kanban configuration (matrix of the number of kanbans for each product in each stage).
$\mathbf{K}_j$	Neighbor with index value j of the current kanban configuration.
k	Auxiliary variable.
k	Rotation counter.
$\tilde{k}_i(y)$	Steady-state probability distribution of $\left\{\tilde{Y}_i(t), t \geq 0\right\}$.
Λ_i	Reciprocal of the average time until the first product other than product i meets the setup criterion (at least one active kanban and one container with input material) after the manufacturing facility stopped processing items of product i.
l_{ij}	Average time from the beginning of the idle period after processing items of product i until product j meets the setup condition.
ℓ	Index value of a neighbor.
$\lambda_{\text{eff},i}$	Effective average arrival rate of customers in an M/M/1/N queueing system.
$\lambda_{[i]}^{\text{ext}}$	Average arrival rate of external demand [for product i].
$\lambda_{[i]}^{\langle(m)\rangle}$	Average arrival rate of demand [for product i] $\langle$in stage $m\rangle$.
$\lambda_{\text{BD}[,i]}^{\langle(m)\rangle}$	Average arrival rate of demand [for product i] $\langle$in stage $m\rangle$ that is backordered.
λ_{ISD}	Average arrival rate of demand that is served immediately upon arrival.
$\lambda_{\text{SD}[,i]}^{\langle(m)\rangle}$	Average arrival rate of demand [for product i] $\langle$in stage $m\rangle$ that is served immediately upon arrival or after a stochastic waiting time.
M	Exponential distribution (distribution with the *M*arkov[ian]/*m*emoryless property).
M	Number of stages.
m	Stage index.
max	Largest value of a set.
min	Smallest value of a set.

μ_i' Transition rate for transition $(1,B) \rightarrow (0,I)$ in the CTMC for component C2 for product i.

μ_i' Transition rate for transitions $(1,y,B) \rightarrow (0,y,I)$, $y = 0,\ldots,Y_i$, and transitions $(n,0,B) \rightarrow (n-1,0,I)$, $n = 2,\ldots,K_i$, in the CTMC for component C3 for product i.

μ_i'' Transition rate for transition $(1,B) \rightarrow (0,V_{i+1})$ in the CTMC for component C2 for product i.

μ_i'' Transition rate for transitions $(1,y,B) \rightarrow (0,y,V_{i+1})$, $y = 0,\ldots,Y_i$, and transitions $(n,0,B) \rightarrow (n-1,0,V_{i+1})$, $n = 2,\ldots,K_i$, in the CTMC for component C3 for product i.

$\mu_{[i]}^{\langle(m)\rangle}$ Average container processing rate [for product i] $\langle$in stage $m\rangle$.

$\mu_{\mathrm{eff}[,i]}^{\langle(m)\rangle}$ Effective avg. container processing rate [for product i] $\langle$in stage $m\rangle$.

N Maximum number of customers in the system including server position (= system capacity).

$N^{\langle(m)\rangle}(t)$ Number of active kanbans and backorders $\langle$in stage $m\rangle$ at time t.

$\tilde{N}^{\langle(m)\rangle}(t)$ Number of active kanbans and backorders in the approximate model $\langle$in stage $m\rangle$ at time t.

$N_i(t)$ Number of active kanbans and backorders (in stage 2) in component C2 (C3) for product i at time t.

$\tilde{N}_i(t)$ Number of active kanbans and backorders (in stage 2) in the approximate model of component C2 (C3) for product i at time t.

$\mathcal{N}$ Set of index values of the neighbors of the current kanban configuration that satisfy given conditions.

$\mathcal{N}_i$ State space of $\{N_i(t), t \geq 0\}$ and $\{\tilde{N}_i(t), t \geq 0\}$.

n Number of active kanbans and backorders for product i.

$n^{(2)}, n$ Number of active kanbans and backorders for product i in stage 2 (component C3).

n_j Number of active kanbans and backorders for product j in stage 2 (component C3).

$\tilde{o}_i(n,y,z)$ Steady-state probability distribution of $\{[\tilde{N}_i(t), \tilde{Y}_i(t), \tilde{Z}_i(t)], t \geq 0\}$.

$P(A)$ Probability of event A.

$p(n)$	Steady-state probability distribution of $\{N(t), t \geq 0\}$.
$\tilde{p}(n)$	Steady-state probability distribution of $\{\tilde{N}(t), t \geq 0\}$.
$\tilde{p}^{(m)}(n)$	Steady-state probability distribution of $\{\tilde{N}^{(m)}(t), t \geq 0\}$.
$\tilde{p}_i(n)$	Steady-state probability distribution of $\{\tilde{N}_i(t), t \geq 0\}$.
$\tilde{q}_i(n,z)$	Steady-state probability distribution of $\{[\tilde{N}_i(t), \tilde{Z}_i(t)], t \geq 0\}$.
$\mathrm{RI}_{[i]j}$	Relevant improvement of the average fill rate [for product i] with neighbor $\mathbf{K}_j$.
$\mathrm{RI}_j^{\max}$	The largest relevant improvement of the average fill rates with neighbor $\mathbf{K}_j$.
$\mathrm{RI}_{[k]}$	The kth largest value for the relevant improvement of neighbors $\mathbf{K}_j$ $(j \in \mathscr{N})$.
r	Number of different products in the system.
$\rho^{\langle(m)\rangle}$	Total traffic intensity (offered load) $\langle$in stage $m\rangle$.
$\rho_i^{\langle(m)\rangle}$	Traffic intensity (offered load) of product i $\langle$in stage $m\rangle$.
$S_{[i]}$	Setup [for product i].
SA	Subassembly.
SA1(1)	Subassembly SA1 for stage 1.
SA2$(m-1,m)$	Subassembly SA2 for stages $m-1$ and m.
$\mathscr{S}$	State space of a CTMC.
$\mathscr{S}_i$	State space of the CTMC for component C2 (C3) for product i.
$s_{[i]}^{\langle(m)\rangle}$	Average setup time [for product i] $\langle$in stage $m\rangle$.
T_i	Average amount of time from the end of a vacation period until the end of the next vacation period in component C2 (C3) for product i.
$\mathrm{TH}_{[i]}^{\langle(m)\rangle}$	Average production rate (= *average throughput*) $\langle$of stage $m\rangle$ [with respect to product i containers].
TRI_j	Total relevant improvement with neighbor $\mathbf{K}_j$.
$\mathrm{TRI}_{[k]}$	The kth largest value for the total relevant improvement of neighbors $\mathbf{K}_j$ $(j \in \mathscr{N})$.
t	Time index.

$t_B^{(i)}$ Average amount of time the manufacturing facility (in stage 2) in the model of component C2 (C3) for product i spends in state B between two vacation periods.

$t_I^{(i)}$ Average amount of time the manufacturing facility (in stage 2) in the model of component C2 (C3) for product i spends in state I between two vacation periods.

$t_S^{(i)}$ Average amount of time the manufacturing facility (in stage 2) in the model of component C2 (C3) for product i spends in state S between two vacation periods.

$t_{\mathrm{SBI}}^{(i)}$ Average amount of time between the end of a vacation period until the start of the next vacation period in the model of component C2 (C3) for product i.

$t_V^{(i)}$ Average length of a vacation period in the model of component C2 (C3) for product i.

t_{ji} Estimate for the average length of the time period from the end of the last busy period for product j until the end of the busy period for product i.

u Product index.

V Vacation period.

V_j Vacation phase for product j.

$w_{\mathrm{BD}[,i]}^{\langle(m)\rangle}$ Average waiting time of backordered demand [for product i] $\langle$in stage $m\rangle$.

$w_{\mathrm{BD},i}^{O:M}$ Average waiting time of backordered demand for product i in the last stage of the original system.

$w_{\mathrm{BD},i}^{S:M}$ Average waiting time of backordered demand for product i in the last stage of the substitute system.

$w_{\mathrm{IM}[,i]}^{\langle(m)\rangle}$ Average waiting time until input material is available [for product i] $\langle$in stage $m\rangle$.

$w_{\mathrm{SD}[,i]}^{\langle(m)\rangle}$ Average waiting time of served demand [for product i] $\langle$in stage $m\rangle$.

Y_i Maximum number of full containers in the output store of stage 1 in component C3 for product i.

$Y_i(t)$ Number of full containers in the output store of stage 1 in component C3 for product i at time t.

$\tilde{Y}_i(t)$ Number of full containers in the output store of stage 1 in the approximate model of component C3 for product i at time t.

$\bar{Y}_0$ Total average inventory in the system (average number of full containers) with the current kanban configuration, $\mathbf{K}_0$.

$\bar{Y}_j$ Total average inventory in the system with neighbor $\mathbf{K}_j$.

$\Delta\bar{Y}_j$ The increase or decrease in the total average inventory when changing from the current kanban configuration, $\mathbf{K}_0$, to neighbor $\mathbf{K}_j$.

$\mathscr{Y}_i$ State space of $\{Y_i(t), t \geq 0\}$ and $\{\tilde{Y}_i(t), t \geq 0\}$.

y Number of full containers in the output store.

$y^{(1)}, y$ Number of full containers in the output store of stage 1 in component C3 for product i.

y_j Number of full containers for product j in the output store of stage 1.

$\bar{y}_{[i]}^{\langle(m)\rangle}$ Average inventory level [of product i] in the output store $\langle$of stage $m\rangle$ (average number of full [product-i] containers).

$\hat{y}_i^{(k)}$ Value of $\hat{y}_i$ at the end of the kth rotation.

$\bar{y}_i^{O:m}$ Average inventory level of product i in the output store of stage m of the original system (average number of full product-i containers).

$\bar{y}_i^{S:m}$ Average inventory level of product i in the output store of stage m of the substitute system (average number of full product-i containers).

$\bar{y}_i^{S:m^+}$ Average inventory level of product i in the output store of stage m^+ of the substitute system (average number of full product-i containers).

$Z_i(t)$ State of the manufacturing facility (in stage 2) in component C2 (C3) for product i at time t.

$\tilde{Z}_i(t)$ State of the manufacturing facility (in stage 2) in the approximate model of component C2 (C3) for product i at time t.

$\mathscr{Z}_i$ State space of $\{Z_i(t), t \geq 0\}$ and $\{\tilde{Z}_i(t), t \geq 0\}$.

z State of the manufacturing facility.

$z^{(2)}, z$ State of the manufacturing facility in stage 2 (component C3).

References

Akturk, M. S. and F. Erhun (1999). An overview of design and operational issues of kanban systems, *International Journal of Production Research* 37, 3859–3881.

Albino, V., M. Dassisti, and G. O. Okogbaa (1995). Approximation approach for the performance analysis of production lines under a kanban discipline, *International Journal of Production Economics* 40, 197–207.

Altiok, T. (1982). Approximate analysis of exponential tandem queues with blocking, *European Journal of Operational Research* 11, 390–398.

Altiok, T. (1985). On the phase-type approximations of general distributions, *IIE Transactions* 17, 110–116.

Altiok, T. (1997). *Performance Analysis of Manufacturing Systems*. New York: Springer.

Altiok, T. and H. G. Perros (1986). Open networks of queues with blocking: split and merge configurations, *IIE Transactions* 18, 251–261.

Altiok, T. and G. A. Shiue (1994). Single-stage, multi-product production/inventory systems with backorders, *IIE Transactions* 26, 52–61.

Altiok, T. and G. A. Shiue (1995). Single-stage, multi-product production/inventory systems with lost sales, *Naval Research Logistics* 42, 889–913.

Altiok, T. and G. A. Shiue (2000). Pull-type manufacturing systems with multiple product types, *IIE Transactions* 32, 115–124.

Amin, M. and T. Altiok (1997). Control policies for multi-product multi-stage manufacturing systems: an experimental approach, *International Journal of Production Research* 35, 201–223.

Anupindi, R. and S. Tayur (1998). Managing stochastic multiproduct systems: model, measures, and analysis, *Operations Research* 46, S98–S111.

Askin, R. G., M. G. Mitwasi, and J. B. Goldberg (1993). Determining the number of kanbans in multi-item just-in-time systems, *IIE Transactions* 25, 89–98.

Baynat, B., Y. Dallery, M. Di Mascolo, and Y. Frein (2001). A multi-class approximation technique for the analysis of kanban-like control systems, *International Journal of Production Research* 39, 307–328.

Berkley, B. J. (1990). Analysis and approximation of a JIT production line: a comment, *Decision Sciences* 21, 660–669.

Berkley, B. J. (1991). Tandem queues and kanban-controlled lines, *International Journal of Production Research* 29, 2057–2081.

Berkley, B. J. (1992a). A decomposition approximation for periodic kanban-controlled flow shops, *Decision Sciences* 23, 291–311.

Berkley, B. J. (1992b). A review of the kanban production control research literature, *Production and Operations Management* 1, 393–411.

Berkley, B. J. (1994). Testing minimum performance levels for kanban-controlled lines, *International Journal of Production Research* 32, 93–109.

Boxma, O. J. and A. G. Konheim (1981). Approximate analysis of exponential queueing systems with blocking, *Acta Informatica* 15, 19–66.

Bux, W. and U. Herzog (1977). The phase concept: approximation of measured data and performance analysis. In K. M. Chandy and M. Reiser (eds.), *Computer Performance*, pp. 23–38. Amsterdam: North-Holland.

Buzacott, J. A., X.-G. Liu, and J. G. Shanthikumar (1995). Multistage flow line analysis with the stopped arrival queue model, *IIE Transactions* 27, 444–455.

Buzacott, J. A. and J. G. Shanthikumar (1993). *Stochastic Models of Manufacturing Systems*. Englewood Cliffs, N.J.: Prentice-Hall.

Cox, D. R. (1955). A use of complex probabilities in the theory of stochastic processes, *Proceedings of the Cambridge Philosophical Society* 51, 313–319.

Dallery, Y. and Y. Frein (1993). On decomposition methods for tandem queueing networks with blocking, *Operations Research* 41, 386–399.

Dallery, Y. and S. B. Gershwin (1992). Manufacturing flow line systems: a review of models and analytical results, *Queueing Systems* 12 (special issue), 3–94.

De Araújo, S. L., M. Di Mascolo, and Y. Frein (1993). Performance evaluation of kanban controlled production systems with multiple consumers and multiple suppliers. In *International Conference on Industrial Engineering and Production Management (IEPM'93)*, Mons, Belgium, June 2–4, 1993.

De Araújo, S. L., Y. Frein, and M. Di Mascolo (1995). Efficient procedures for the design of kanban systems. In *International Conference on Industrial Engineering and Production Management (IEPM'95)*, Marrakech, Marocco, April 4–7, 1995, pp. 511–520.

Deleersnyder, J.-L., T. J. Hodgson, H. Muller (-Malek), and P. J. O'Grady (1989). Kanban controlled pull systems: an analytic approach, *Management Science* 35, 1079–1091.

Di Mascolo, M. (1996). Analysis of a synchronization station for the performance evaluation of a kanban system with a general arrival process of demands, *European Journal of Operational Research* 89, 147–163.

Di Mascolo, M. and Y. Dallery (1996). Performance evaluation of kanban controlled assembly systems. In *CESA'96 IMACS Multiconference on Computational Engineering in Systems Applications: Symposium on Discrete Events and Manufacturing Systems*, Lille, 9–12 July, 1996.

Di Mascolo, M., Y. Frein, and I. Dallery (1992). Queueing network modeling and analysis of kanban systems. In *Third International Conference on Computer Integrated Manufacturing*, New York, May 20–22, 1992, pp. 202–211. Rensselaer Institute: IEEE Computer Society Press, Troy, N.Y.

Di Mascolo, M., Y. Frein, and Y. Dallery (1996). An analytical method for performance evaluation of kanban controlled production systems, *Operations Research* 44, 50–64.

Duri, C., Y. Frein, and M. Di Mascolo (1995). Performance evaluation of kanban multiple-product production systems. In *INRIA/IEEE Conference on Emerging Technologies and Factory Automation (ETFA'95)*, Paris, October 1995, pp. 557–566.

Federgruen, A. and Z. Katalan (1996). The stochastic economic lot scheduling problem: cyclical base-stock policies with idle times, *Management Science* 42, 783–796.

Fujiwara, O., X. Yue, K. Sangaradas, and H. T. Luong (1998). Evaluation of performance measures for multi-part, single-product kanban controlled assembly systems with stochastic acquisition and production lead times, *International Journal of Production Research* 36, 1427–1444.

Fullerton, R. R. and C. S. McWatters (2001). The production performance benefits from JIT implementation, *Journal of Operations Management* 19, 81–96.

Groenevelt, H. (1993). The just-in-time system. In S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin (eds.), *Logistics of Production and Inventory*, Volume 4 of *Handbooks in Operations Research and Management Science*, Chapter 12, pp. 629–670. Amsterdam: Elsevier (North-Holland).

Gross, D. and C. M. Harris (1998). *Fundamentals of Queueing Theory*. New York: Wiley.

Gstettner, S. and H. Kuhn (1996). Analysis of production control systems kanban and CONWIP, *International Journal of Production Research* 34, 3253–3273.

Hall, R. W. (1983). *Zero Inventories*. Homewood, Ill.: Dow Jones-Irwin.

Hillier, F. S. and R. W. Boling (1967). Finite queues in series with exponential or Erlang service times: a numerical approach, *Operations Research* 15, 286–303.

Huang, C.-C. and A. Kusiak (1996). Overview of kanban systems, *International Journal of Computer Integrated Manufacturing* 9, 169–189.

Jordan, S. (1988). Analysis and approximation of a JIT production line, *Decision Sciences* 19, 672–681.

Karmarkar, U. S. and S. Kekre (1989). Batching policy in Kanban systems, *Journal of Manufacturing Systems* 8, 317–328.

Kim, I. and C. S. Tang (1997). Lead time and response time in a pull production control system, *European Journal of Operational Research* 101, 474–485.

Kimura, O. and H. Terada (1981). Design and analysis of pull system: a method of multi-stage production control, *International Journal of Production Research* 19, 241–253.

Kleinrock, L. (1975). *Queueing Systems: Volume I: Theory*. New York: Wiley.

Krieg, G. N. and H. Kuhn (2001). Production planning of multi-product kanban systems with significant setup times. In B. Fleischmann, R. Lasch, U. Derigs, W. Domschke, and U. Rieder (eds.), *Operations Research Proceedings 2000*, pp. 316–321. Berlin: Springer.

Krieg, G. N. and H. Kuhn (2002a). A decomposition method for multi-product kanban systems with setup times and lost sales, *IIE Transactions* 34, 613–625.

Krieg, G. N. and H. Kuhn (2002b). Performance evaluation of two-stage multi-product kanban systems. Submitted for publication.

Krieg, G. N. and H. Kuhn (2004). Analysis of multi-product kanban systems with state-dependent setups and lost sales, *Annals of Operations Research* 125, 141–166.

Kulkarni, V. G. (1999). *Modeling, Analysis, Design, and Control of Stochastic Systems*. New York: Springer.

Law, A. M. and W. D. Kelton (2000). *Simulation Modeling and Analysis* (3rd ed.). New York: McGraw-Hill.

Little, J. D. C. (1961). A proof for the queuing formula: $L = \lambda W$, *Operations Research* 9, 383–387.

Markowitz, D. M., M. I. Reiman, and L. M. Wein (2000). The stochastic economic lot scheduling problem: heavy traffic analysis of dynamic cyclic policies, *Operations Research* 48, 136–154.

Mitra, D. and I. Mitrani (1990). Analysis of a kanban discipline for cell coordination in production lines. I, *Management Science* 36, 1548–1566.

Mitra, D. and I. Mitrani (1991). Analysis of a kanban discipline for cell coordination in production lines, II: stochastic demands, *Operations Research* 39, 807–823.

Monden, Y. (1981a). Adaptable kanban system helps Toyota maintain just-in-time production, *The Journal of Industrial Engineering* 15, 29–46.

Monden, Y. (1981b). How Toyota shortened supply lot production time, waiting time and conveyance time, *The Journal of Industrial Engineering* 15, 22–30.

Monden, Y. (1981c). Smoothed production lets Toyota adapt to demand changes and reduce inventory, *The Journal of Industrial Engineering* 15, 42–51.

Monden, Y. (1981d). What makes the Toyota production system really tick?, *The Journal of Industrial Engineering* 15, 36–45.

Monden, Y. (1998). *Toyota Production System: An Integrated Approach to Just-In-Time* (3 ed.). Norcross, Ga.: Engineering & Management Press.

Neuts, M. F. (1994). *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach.* New York: Dover. Unabridged and corrected republication.

Nori, V. S. and B. R. Sarker (1998). Optimum number of kanbans between two adjacent stations, *Production Planning & Control* 9, 60–65.

O'Cinneide, C. A. (1999). Phase-type distributions: open problems and a few properties, *Communications in Statistics: Stochastic Models* 15, 731–757.

Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production.* Cambridge, Mass.: Productivity Press.

Perros, H. G. (1990). Approximation algorithms for open queueing networks with blocking. In H. Takagi (ed.), *Stochastic Analysis of Computer and Communication Systems*, pp. 451–498. Amsterdam: Elsevier (North-Holland).

Sauer, C. H. and K. M. Chandy (1975). Approximate analysis of central server models. Technical report, IBM T. J. Watson Research Center, Yorktown Heights, N.Y.

Schmidbauer, H. and A. Rösch (1994). A stochastic model of an unreliable kanban production system, *OR Spektrum* 16, 155–160.

Schömig, A. (1997). *Performance Analysis of WIP-Controlled Manufacturing Systems.* Ph.D. diss., University of Würzburg, Germany.

Schonberger, R. J. (1982). *Japanese Manufacturing Techniques: Nine Hidden Lessons in Simplicity.* New York: The Free Press.

Seki, Y. and N. Hoshino (1999). Transient behavior of a single-stage kanban system based on the queueing model, *International Journal of Production Economics* 60-61, 369–374.

Shingo, S. (1989). *A Study of the Toyota Production System from an Industrial Engineering Viewpoint* (revised ed.). Cambridge, Mass.: Productivity Press.

Siha, S. (1994). The pull production system: modelling and characteristics, *International Journal of Production Research* 32, 933–949.

Singh, N. and J. K. Brar (1992). Modelling and analysis of just-in-time manufacturing systems: a review, *International Journal of Operations & Production Management* 12, 3–14.

So, K. C. and S. C. Pinault (1988). Allocating buffer storages in a pull system, *International Journal of Production Research* 26, 1959–1980.

Sox, C. R., P. L. Jackson, A. Bowman, and J. A. Muckstadt (1999). A review of the stochastic lot scheduling problem, *International Journal of Production Economics* 62, 181–200.

Stewart, W. J. (1991). MARCA: Markov chain analyzer, a software package for Markov modeling. In W. J. Stewart (ed.), *Numerical Solutions of Markov Chains*, pp. 37–61. New York: Marcel Dekker.

Stewart, W. J. (1994). *Introduction to the Numerical Solution of Markov Chains.* Princeton, N.J.: Princeton University Press.

Stewart, W. J. (1996). *MARCA: Markov chain analyzer, a software package for Markov modelling, Version 3.0.* Raleigh, N.C. 27965-8206: Department of Computer Science, North Carolina State University.

Sugimori, Y., K. Kusunoki, F. Cho, and S. Uchikawa (1977). Toyota production system and Kanban system: materialization of just-in-time and respect-for-human system, *International Journal of Production Research* 15, 553–564.

Takahashi, Y., H. Miyahara, and T. Hasegawa (1980). An approximation method for open restricted queueing networks, *Operations Research* 28, 594–602.

Tempelmeier, H. (2000). Inventory service-levels in the customer supply chain, *OR Spektrum* 22, 361–380.

Uzsoy, R. and L. A. Martin-Vega (1990). Modelling kanban-based demand-pull systems: a survey and critique, *Manufacturing Review* 3, 155–160.

Wang, H. and H.-P. B. Wang (1990). Determining the number of kanbans: a step toward non-stock-production, *International Journal of Production Research* 28, 2101–2115.

Wang, H. and H.-P. B. Wang (1991a). Decomposition and optimal design of Kanban systems. In A. Satir (ed.), *Just in Time Manufacturing Systems: Operational Planning and Control Issues*, pp. 165–174. Amsterdam: Elsevier.

Wang, H. and H.-P. B. Wang (1991b). Optimum number of kanbans between two adjacent workstations in a JIT system, *International Journal of Production Economics* 22, 179–188.

White, R. E., J. N. Pearson, and J. R. Wilson (1999). JIT manufacturing: a survey of implementations in small and large U.S. manufacturers, *Management Science* 45, 1–15.

White, R. E. and V. Prybutok (2001). The relationship between JIT practices and type of production system, *Omega* 29, 113–124.

Xiaobo, Z., Q. Gong, and J. Wang (2002). On control strategies in a multi-stage production system, *International Journal of Production Research* 40, 1155–1171.