DESIGN AND SCHEDULING OF PERIODIC REVIEW KANBAN SYSTEMS

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OFTHE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Thesis
TS
157
.E74
1997

By

Feryal Erhun

June, 1997

DESIGN AND SCHEDULING OF PERIODIC REVIEW KANBAN SYSTEMS

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Feryal Erhun

June, 1997

TS
157
.E74
1997

B 037980

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assis. Prof. M. Selim Aktürk(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Cemal Dinçer

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Jay B. Ghosh

Approved for the Institute of Engineering and Sciences:

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

DESIGN AND SCHEDULING OF PERIODIC REVIEW KANBAN SYSTEMS

Feryal Erhun
M.S. in Industrial Engineering
Supervisor: Assis. Prof. M. Selim Aktürk
June, 1997

In the last years, the term just-in-time (JIT) has become a common term in repetitive manufacturing systems. It can be defined as the ideal of having the necessary amount of material available where it is needed and when it is needed. One of the major elements of JIT philosophy and pull mechanism is the Kanban system. This system is the information processing and hence shop floor control system of JIT philosophy.

In this study, we propose an algorithm to determine the withdrawal cycle length, kanban size and number of kanbans simultaneously in a periodic review Kanban system under multi-item, multi-stage, multi-period modified flowline production setting. The proposed algorithm considers the impact of operating characteristics such as scheduling and actual lead times on design parameters.

Key words: Just-in-time, Kanban systems, Periodic review systems, Scheduling

ÖZET

PERİYODİK KONTROLLÜ KANBAN SİSTEMLERİNİN TASARIMI VE ÇİZELGELEMESİ

Feryal Erhun
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. M. Selim Aktürk
Haziran, 1997

Tam-zamanında üretim (TZÜ) son yıllarda tekrarlı üretim sistemlerinde çok sık kullanılan bir terimdir. TZÜ gereken malzemeye, gereken zaman ve gereken yerde ulaşma idealidir. TZÜ sistemlerinin ve çekme tipi üretim sistemlerinin en önemli elemanlarından biri Kanban sistemidir. Kanban sistemi, TZÜ sistemlerinin bilgi akışını da sağlayan atölye kontrol mekanizmasıdır.

Bu çalışmada amaç çok ürünlü, çok aşamalı, çok dönemli, akış hatlı, periyodik kontrollü Kanban sistemlerinde çekme süresinin, kanban sayılarının ve kanban büyüklüklerinin belirlenmesidir. Önerilen yöntem, çizelgeleme ve gerçek tedarik süreleri gibi işlem özelliklerinin tasarım üzerindeki etkilerini de gözönüne almaktadır.

Anahtar sözcükler. Tam-zamanında-üretim sistemleri, Kanban sistemleri, Periyodik kontrollü sistemler, Çizelgeleme

To my family

ACKNOWLEDGEMENT

I am mostly grateful to Selim Aktürk who has been supervising me with patience and everlasting interest and being helpful in any way during my graduate studies.

I am also indebted to Jay B. Ghosh and Cemal Dinçer for showing keen interest to the subject matter and accepting to read and review this thesis. Their remarks and recommendations have been invaluable.

I would like to thank to my friends Nilüfer Erkin, Ali Erkan, Fatma Gzara, Maher Lahmar, Muhittin Demir, Bahar Kara, and Nebahat Dönmez for their understanding and patience.

I cannot forget the friendship of Hülya Emir, Alev Kaya and Serkan Özkan who always supported me through my master's thesis.

I have to express my sincere gratitude to anyone who have helped me, which I have forgotten to mention here.

Finally, I would like to thank to Semih Oğuz, for his support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last years, the term just-in-time (JIT) has become a common term in repetitive manufacturing. It is used to describe a management philosophy that encourages change and improvement through inventory reduction. JIT can be defined as the ideal of having the necessary amount of material available where it is needed and when it is needed. It is an attempt to produce items in the smallest possible quantities, with minimal waste of human and natural resources, and only when they are needed. JIT systems have proven to be effective at meeting production goals in environments with high process reliability, low setup times and low demand variability [Groenevelt (1993)].

In general, JIT has a pull system of coordination between stages of production. In a pull system, a production activity at a stage is initiated to replace a part used by the succeeding stage, whereas in a push system, production takes place for future need. The push system is basically a planning system, whereas the pull system is an execution system [Olhager (1995)].

Some of the often cited benefits of JIT production include:

- reduced inventories,
- reduced lead times,

- higher quality,
- reduced scrap and rework rates,
- ability to keep schedules,
- increased flexibility,
- easier automation,
- better utilization of workers and equipment.

All of these ultimately translate into reduced costs, higher quality finished products, and the ability to compete better. However, there is a limit to the extent that JIT can be usefully applied in many industries. The major JIT successes are in repetitive manufacturing environment. If the demand cannot be predicted accurately and product variety cannot be constrained, it is not possible to implement JIT effectively. The final assembly schedule must also be very level and stable. Any major deviations will cause upstream stages to hold larger inventories which will destroy JIT nature of the system. In general, in pull approaches, information flow is tied to material flow. As a result valuable information (e.g. on the demand trend) is not sent to all stages of production as soon as it is available. Pull systems can therefore be characterized by large information lead times, especially where there are large material flow times.

Justification of JIT implementation can be supported from several angles. Most of the traditional systems (such as material requirements planning (MRP) or reorder point systems) are static systems emphasizing the status quo. In these systems, the emphasis is on achieving individual operation standards, and the aim is to avoid any deviation from the standard. If current values of manufacturing variables are met, then the system is regarded as successful. JIT, on the other hand, seeks to change the values of the manufacturing variables. It does this by organizing the production process so that small inventories are strategically placed throughout the process and then carefully reduces these inventories to expose production problems which when solved reduce costs and lead times, and improve quality. JIT uses an enforced problem solving approach. Because the inventory is reduced to a minimum, the system cannot tolerate any interruption; therefore extreme care is taken to find out and solve any production problems. In traditional systems, no such incentive to solve production problems is available.

One of the major elements of JIT philosophy and pull mechanism is the Kanban system. Kanban is the Japanese word for visual record or card. In Kanban system, cards are used to authorize production or transportation of a given amount of material. This system is the information processing and hence shop floor control system of JIT philosophy. While kanbans are being used to pull the parts, they are also used to visualize and control in-process inventories. The system effectively limits the amount of in-process inventories, and it coordinates production and transportation of consecutive stages of production in assembly-like fashion. Therefore, Kanban system is the manual method of harmoniously controlling production and inventory quantities within the plant. The Kanban system appears to be best suited for discrete part production feeding an assembly line.

There are a number of variants of Kanban system. The dual-card Kanban system employs two types of kanban cards: production kanban and transportation (also called conveyance or withdrawal) kanban. Transportation kanban defines the quantity that the succeeding stage should withdraw from the preceding stage. Each card circulates between two stages only; the user stage for the part in question and the stage which produces it. Production kanban, on the other hand, defines the quantity of the specific part that the producing stage should manufacture in order to replace those which have been removed.

For a Kanban system to operate effectively, very strict discipline is required. This discipline relates to the usage of the kanban cards. There are six rules for dual-card Kanban system on the usage of the kanbans [Browne et al. (1993)]:

Rule 1: A stage should withdraw the needed products from the preceding stage according to information provided on the transportation kanban.

Rule 2: A stage should produce products in quantities withdrawn by the succeeding stage according to the information provided by the production kanban.
Rule 3: If there is no kanban card, there will be no production and no transportation of parts.
Rule 4: Defective products should never be passed to the succeeding stage.
Rule 5: The number of kanbans can be gradually reduced in order to improve the processes and reduce waste.
Rule 6: Kanbans should be used to adapt to only small fluctuations in demand.

The first three rules tell that parts should be withdrawn and produced 'just-in-time' as they are needed, not before they are needed and not in larger quantities than needed. The fourth rule causes a rigorous quality control at each stage of production process. The fifth rule conveys the fact that inventory can be used as an independent control variable. The level of in-process inventory can be controlled by the number of kanbans and kanban sizes in the system at any time. The sixth rule is related with the limitations of the Kanban system. Kanban system can react effectively only to small fluctuations in demand. Monden (1984) states that the Kanban system should be able to adapt to daily changes in demand within 10 % deviations from the monthly master production schedule (MPS).

Kanban system can be either instantaneous or periodic review system. In instantaneous review systems, the kanbans are dispatched upstream as soon as an order occurs. In periodic review systems, the kanbans are collected and dispatched periodically. Periodic review systems may be either fixed quantity, nonconstant withdrawal cycle, or fixed withdrawal cycle, nonconstant quantity [Monden (1981)]. Under the fixed quantity, nonconstant withdrawal cycle system, kanbans are dispatched upstream when the number of kanbans posted on a withdrawal kanban post reaches a predetermined order point. Under the fixed withdrawal cycle, nonconstant quantity system, the period between material handling operations is fixed and the quantity ordered depends on the usage over the withdrawal cycle.

The use of a Kanban system without JIT philosophy makes little sense. The prerequisites of the system (design of the manufacturing system, smoothing of production, standardization of operations, etc.) must be implemented before an effective pull system can be implemented. For a Kanban system to work well, a number of requirements have to be fulfilled. Since each daily assembly schedule must be very similar to all other daily schedules, it is essential that it is possible to freeze the master production schedule for a fixed time period. The final schedule must be very level and stable. The manufacturing system should conform as closely as possible to the repetitive manufacturing system. Mixed model capability in all stages of the production process is required to run a mixed model system effectively.

The Kanban system performs best when:

- demand for parts is steady and has sufficient volume,
- setup times are small,
- the equipment is reliable, and
- defect rates are low.

A Kanban system is ineffective in achieving JIT production for items with low volume; i.e., less than a day's consumption per container. Since at each stage there is at least some inventory for a product, too much in-process inventory would accumulate for items that require only infrequent production.

Demand fluctuations have a tendency to become amplified from one stage to the next when timing and size of the replenishment orders are based on locally observed demands as in a pull system [Kimura and Terada (1981)]. Therefore, any major deviations will cause a ripple effect through the production system causing upstream stages to hold larger inventories.

Items with large setup times are more difficult to manage with the Kanban system. Large setup times require large lot sizes, and large lot sizes inflate lead times and in-process inventories. Unreliable systems, i.e. systems with unreliable machines and high defect rates, cause similar problems. In unreliable systems, the safety factor should be high enough to deal with the unexpected events, so the in-process inventories increase. The increase in lead times and in-process inventories destroy JIT nature of the system.

Even though the dual-card Kanban system provides strong control on the production system, due to its strict assumptions and prerequisites, it is difficult to implement it. Therefore, a variant of this system, called single-card Kanban system, is sometimes used as a first stage to develop a dual-card Kanban system. In single-card Kanban system, the transportation of materials is still controlled by transportation kanbans. However, the production kanbans to control the production within a cell is absent. Instead, a production schedule provided by the central production planning is used. Hence the system has a strong similarity to a conventional push system, with pull elements added to coordinate the transportation of the parts. One of the advantages of single-card push-pull system is its simplicity in implementation. Moreover, as the information on demand trend is sent to all stages of production as soon as its available, single-card Kanban system has shorter information lead times compared to dual-card Kanban systems.

In this study, we propose an algorithm to determine the withdrawal cycle length, number of kanbans and kanban sizes of a periodic review Kanban system simultaneously in a multi-item, multi-period, multi-stage capacitated modified flowline structure production setting with fixed withdrawal cycles. The proposed algorithm considers the impact of operational issues, such as kanban schedules and actual lead times, on the design parameters. The production setting is imperfect in which

- demand may be variable,
- setup times may be significant, and
- production line may be unbalanced.

In the following chapter, the related literature is reviewed with an emphasis on the limitations of the existing models. Chapter 3 is dedicated to problem definition. After stating the motivating points behind this study, the problem is defined, the underlying assumptions are stated, and the proposed algorithm is explained. In Chapters 4 and 5, the experimental results are given and a numerical example is presented, respectively. Finally, Chapter 6 concludes the study with suggestions for the future research.

Chapter 2

Literature Review

In this chapter, the related literature is reviewed. First, a review of Kanban systems is presented. To be able to compare the existing studies, we divide the models into two parts: models for determining the design parameters are reviewed in Section 2.1, and the models for determining the kanban sequences are reviewed in Section 2.2. In Section 2.3, brief reviews on due-date estimation and group technology are given. Section 2.4 summarizes this chapter.

2.1 Determining the Design Parameters

This section reviews the models for determining the design parameters in a Kanban system. To discuss and compare the models, first a tabular format is introduced. Then, the models are explained briefly. Finally, the limitations of the existing models are stated.

In the tabular format the below characteristics are considered:

1. Model Structure: Mathematical Programming, Simulation, Markov Chains, Others

2. Solution Approach:

2.1 Heuristic

2.2 Exact: Dynamic Programming, Integer Programming, Linear Programming, Mixed Integer Programming, Nonlinear Integer Programming.

3. **Decision Variables:**

3.1 number of kanbans

3.2 order interval

3.3 safety stock level

3.4 kanban size

4. **Performance Measures**

4.1 number of kanbans

4.2 utilization

4.3 measures: Inventory holding cost, Shortage cost, Fill rate

5. **Objective:**

5.1 Minimize cost: Inventory Holding Cost, Operating Costs, Shortage Cost, SeTup Cost.

5.2 Minimize inventory

6. **Setting:**

6.1 Layout : Assembly-tree, Serial, Network without backtracking

6.2 Time period: Multi-period, Single-period

6.3 Item: Multi-item, Single-item

6.4 Stage: Multi-stage, Single-stage

6.5 Capacity: Capacitated, Uncapacitated

7. Kanban type: Single-card Kanban, Dual-card Kanban

8. **Assumptions:**

8.1 Kanban Sizes: Known, Unit

8.2 Stochasticity: Demand, Lead time, Processing time

8.3 Withdrawal Cycle: Fixed withdrawal cycle, Instanteneous

8.4 No Shortages

8.5 System Reliability: Dynamic demand, Machine unreliability, Imbalance between stages, Rework

Most of the existing studies in the literature are modeled by using mathematical programming, simulation or Markov chains. There are a few exceptional studies that use other methods such as statistical analysis or Toyota formula. In the tabular format, we collect these models under the heading 'others'.

For the mathematical formulations a solution approach should also be stated. This approach can be either heuristic or exact. For exact solution, different methodologies such as dynamic programming, integer programming, linear programming, mixed integer programming, and nonlinear integer programming can be used.

For the analytical models the decision variables and for the simulation models the performance measures should be indicated. The decision variables are kanban sizes, number of kanbans, withdrawal cycle length for periodic review models, and safety stock levels. The performance measures are number of kanbans, machine utilizations, inventory holding cost, backorder cost, and fill rates. Fill rate can be defined as the probability that an order will be satisfied through inventory. Models can consider different combinations of the criteria stated above.

The objectives for the analytical models can be minimizing the costs or minimizing the inventories. For the cost minimization, the cost terms can be considered either independently as inventory holding cost, shortage cost, and setup cost, or the combination of these costs as operating cost.

The production setting for the models include the layout, number of time periods, number of items, number of stages, and the capacity. Layout can be serial (flowline), assembly-tree, or a general network without backtracking (modified flowline). An empty cell in the tables for any of these indicate that the characteristic is not considered in the corresponding study.

Kanban system can be either a single-card Kanban system or a dual-card Kanban system. The assumptions for the models are also stated in the tabular format. These assumptions are the ones that are commonly considered, more specific ones are indicated in the explanations of the models. The first assumption is on kanban size. An empty cell for this characteristic indicates that the kanban size is not a parameter, but is a decision variable for the system. Another assumption relates to the nature of the system such that the system can be either stochastic or deterministic. For the deterministic models, this cell is left empty. For the stochastic ones, the stochastic parameters are indicated. The withdrawal cycle length shows if the system is an instantaneous or a periodic review system. The fourth assumption is related with backorders. An empty cell indicates that backorder is allowed. And, the last assumption is on the system reliability. If the system is reliable, this cell is left empty, otherwise the unreliability of the system is stated.

In Figures 2.1 and 2.2, the models are presented by using the above scheme. Further explanations of the models are given below:

Kimura and Terada (1981) describe the operation of Kanban systems and examine the accompanying inventory fluctuations in a JIT environment. They provide several balance equations for Kanban systems in a single item, multi-stage, uncapacitated serial production setting. They use these equations to demonstrate how demand fluctuations of the final product influence the fluctuations of production and the fluctuations of inventory at the upstream

		Kimura&Terada	Huang et al.	Bitran&Chang	Rees et al.	Miyazaki et al.	Gupta&Gupta	Karmarkar&Kekre	Philipoom et al.	Bard&Golony
Year		81	83	87	87	88	89	89	90	91
Model Structure		M,S	S	M	O	O	S	MC	M,S	M
Solution Approach	Heuristic									
	D, I, L, M, N			N,M					I	M
Decision Variables	#of kanbans	X		X	X	X		X		X
	order interval					X				
	safety stock									
	kanban size							X	X	
Performance Measures	# of kanbans									
	utilization						X			
	I, S,F		I				I,S			
Objective	cost	O		O	H,S	H,T		H,S	H,T	S,H,T
	inventory								X	
Setting	layout	S	A	A			A			A
	horizon	M	M	M			M	S	S	M
	item	S	M	S			S	S	M	M
	stage	M	M	M			M	M	M	M
	capacity	U	C	C			C		C	C
Kanban Type	S, D	D	D	D	D	D	D	S,D	D	S
Assumptions	kanban size	K	K	K	U	K				K
	stochasticity	D	D,P		L		P	D,P		
	withdrawal cycle	I	I	I	I	F	I	I	F	I
	no shortages			X					X	
	system reliability		I		D		M,I			

Figure 2.1: Models for Kanban Systems: Choosing Design Parameters

		Li&Co	Mittal&Wang	Askin et al.	Mitwasi&Askin	Takahashi	Ohno	Philipoom et al.	Berkley	Proposed Algorithm
Year		91	92	93	94	94	95	96	96	97
Model Structure		M	S	MC	M	S	O	M	S	O
Solution Approach	**Heuristic**				X					X
	D, I, L, M, N	D			N,M			I		
Decision Variables	**#of kanbans**	X		X	X		X	X		X
	order interval									X
	safety stock			X			X			
	container size							X		X
Performance Measures	**# of kanbans**		X			X				
	utilization									
	I, S,F								I,F	
Objective	**cost**	H		H,S	O		H,S,T	H,T		H, S
	inventory									
Setting	**layout**	S,A	N	S		S		A	S	N
	horizon	M	M	S	M	M	M	M	M	M
	item	S	M	M	M	S	S	M	M	M
	stage	M	M	M	S	M	S	M	M	M
	capacity	U			C		C	C	C	C
S, D		D	D	S	D	D	D	D	D	D
Assumptions	**container size**	K	K	K	K	K	U			
	stochasticity		D,P	D,P		D,P	D		D,P	D
	withdrawal cycle	I	I	I	I	I	I	I	I	F
	no shortages		X		X			X		
	system reliability		M,R		D	I				I

Figure 2.2: Models for Kanban Systems: Choosing Design Parameters

stages. The authors use simulation to show that fluctuations are amplified when the size of order and/or lead time becomes large.

Huang et al. (1983) simulate the JIT (by using Q-Gert) with kanban for a multi-line, multi-stage production system in order to determine its adaptability to a U.S. production environment. The simulated production system includes variable processing times (normally distributed), variable master production scheduling (exponentially distributed demand), and imbalances between production stages. The authors conclude that the variability in processing times and demand rates are amplified in a multi-stage setting, and that excess capacity has to be available to avoid bottlenecks.

Monden (1984) comments on the conclusion drawn by Huang et al. (1983). He stated that the Kanban system should be able to adapt to daily changes in demand within 10 % deviations from the monthly MPS. Larger seasonal fluctuations in demand can be accommodated by setting up the monthly MPS appropriately.

Bitran and Chang (1987) formulate a nonlinear integer program to extend Kimura and Terada's (1981) serial model. They provide a formulation for the Kanban system in a deterministic single item, multi-stage capacitated assembly-tree structure production setting. The formulation assumes zero transportation lead time and planning periods of known length and finds the minimum feasible number of kanbans. The authors show that the initial nonlinear model can be transformed into an integer linear program with the same feasible and optimal solutions. The model does not incorporate uncertainties.

Rees et al. (1987) develop a methodology for dynamically adjusting the number of kanbans in an unstable production environment. They use time series methods and historical data to estimate the autocorrelation and distribution functions of lead time. They use Toyota equation with unit kanban capacities to determine the number of kanbans.

Miyazaki et al. (1988) modify the conventional economic order quantity

(EOQ) model to determine the average inventory for fixed interval withdrawal and supplier Kanban systems, give formulae to determine the minimum number of kanbans required, and derive an algorithm to obtain the optimal order interval. The objective is to minimize the average inventory holding and ordering costs in a deterministic setting.

Gupta and Gupta (1989) simulate (by using System Dynamics) a single item, multi-line, multi-stage, dual-card Kanban system. They investigate the impact of changing the number of kanbans and kanban sizes on the performance of the system. The performance measures are chosen to be in-process inventory, capacity utilization or production idle time and shortage of the final product. The authors conclude that determining the number of kanbans is essential to the performance of the system, and keeping the buffer size constant by increasing the kanban size and decreasing the number of kanbans accordingly increases the inventory. For the smooth operation in a JIT environment the stages should be balanced and the suppliers should be reliable. And finally, the system performance declines with an increase in demand variability.

Karmarkar and Kekre (1989) develop a continuous time Markov model to study the effect of batch sizing policy on production lead time and on inventory levels. Both single and dual card Kanban cells and two-stage systems are modeled. The effect of varying the number of kanbans in the cell is also examined. The primary intent of the investigation is to develop a qualitative analysis of Kanban systems that can provide insights to parametric behavior of Kanban systems. The results show that the kanban size has a significant effect on the performance of the Kanban systems.

Philipoom et al. (1990) describe the signal kanban technique and demonstrate two versions of an integer mathematical programming approach for determining the optimal lot sizes to signal kanbans in a multi-item multi-stage setting. A simulation model is employed to test the effectiveness of the programming models. The models assume no backorders, therefore the stages are decoupled and interdependencies are eliminated.

Bard and Golony (1991) develop a mixed integer linear program to determine the number of kanbans at each stage in a multi-item, multi-stage capacitated general assembly shop. The objective is to minimize inventory holding, shortage and setup costs for a given demand and planning horizon without violating the basic kanban principles. They show that the resulting solutions have total costs of approximately half those obtained using the Toyota equation. The model is most appropriate when the demand is steady and the lead times are short.

Li and Co (1991) develop bounds for an efficient kanban assignment and apply them to solve dynamic programming model in a deterministic, single item, multi-stage, serial/assembly-tree structure production setting. This model is an extension to Bitran and Chang's (1987) model. The authors assume infinite capacity. This assumption not only removes the capacity constraints but also eliminates the need to keep track of the number of units in partially filled kanbans. Therefore, the model is computationally very efficient, even for a complex non-serial system.

Mittal and Wang (1992) develop a database oriented simulator, CADOK (Computer Aided Determination of Kanbans) to determine the number of kanbans in a production setting where breakdowns, reworks, setup times, variable processing times (normally distributed), and variable demand (exponentially distributed) are modeled. Backorder costs are assumed to be prohibitively high. The model can handle both assembly and disassembly type operations, that is to say, several stages supply products to the same stage and one stage supplies parts to several stages, respectively. Also, delay can be induced for the information to travel from one stage to another. But, the system cannot handle backtracking so it is only applicable to flowlines or modified flowlines.

Askin et al. (1993) develop a continuous time, steady-state Markov model for determining the optimal number of production kanbans in a multi-item, multi-stage serial production setting. The objective is to minimize the inventory holding and shortage costs given external demand and the kanban sizes. The external demand assumption permits the modeling of a multi-stage system as independent stages. The model uses Toyota equation and finds the number of kanbans and safety factor. The performance of the model is sensitive to the accurate estimation of the queue lengths.

Mitwasi and Askin (1994) provide a nonlinear integer mathematical model for the multi-item, single stage, capacitated Kanban system. It is assumed that demand is external, dynamic and evenly distributed over period, the system is reliable, setups are small enough to allow batch sizes as small as a single kanban, and no shortages are permissible. The control periods are assumed to be small enough to ignore batch sequencing problem. The model is transformed to a simpler model with the same set of optimal and feasible kanban solutions. Lower and upper bounds for number of kanbans are developed. A heuristic solution is also presented.

Takahashi (1994) provides a simulation model to determine the number of kanbans for single item, unbalanced serial production systems under stochastic conditions (demand with exponential distribution and processing time with gamma distribution). An algorithm that allows different numbers of production and withdrawal kanbans at an inventory point is proposed. It is assumed that the total number of kanbans are known, withdrawal lead time is negligible, and backorders are allowed.

Ohno et al. (1995) derive the stability condition of a JIT production system with the production and supplier kanbans under the stochastic demand and deterministic processing times. An algorithm for determining the optimal number of two kinds of kanbans that minimize an expected average cost per period is devised. In other words, this algorithm determines the optimal safety stocks in Toyota equation.

Philipoom et al. (1996) provide a nonlinear integer mathematical model for the multi-item, multi-stage, multi-period, capacitated Kanban system. It is assumed that demand is deterministic, the system is reliable, setups are sequence-independent, production costs are stable, lot sizes remain constant throughout the shop and no shortages are permissible. The model determines kanban sizes, number of kanbans, and final assembly sequence simultaneously by minimizing the setup and inventory holding cost.

Berkley (1996) investigates the effect of kanban size on system performance in a multi-item, multi-stage, dual-card Kanban system. The performance measures are in-process inventory and customer service level. The author varies the number of kanbans and kanban sizes in the tandem so that the total in-process inventory capacity remains constant. Simulation results show that smaller kanban sizes lead to smaller in-process inventories, and smaller kanban sizes can lead to better customer service when the cost of the greater setup times can be offset by the benefits of more frequent material handling. The study assumes that the kanban sizes and number of kanbans are same for all parts and the set of kanban size values are independent of the demand distribution.

The limitations of the analytical models can be stated as follows:

- Most of the models assume that the kanban sizes are known. The exceptions are Gupta and Gupta (1989), Karmarkar and Kekre (1989), and Philipoom et al. (1996). In the remaining studies, it is assumed that kanban sizes are known and the number of kanbans are determined by using these predetermined values. In fact, number of kanbans and kanban sizes should be determined simultaneously, as these two together affect the performance of the system. It is not known under what conditions large kanban sizes and small numbers of kanbans are preferred to small kanban sizes with large numbers of kanbans.

- Almost none of the models, except Philipoom et al. (1996), consider the impact of operating issues on design parameters. For example, it is usually assumed that the control periods are small enough to ignore batch sequencing problem. Even in the study of Philipoom et al., only the final assembly sequence is determined and it is assumed that it propagates back by the first-come-first-served (FCFS) rule.

- In general, instantaneous material handling is used. There are only two models that use noninstantaneous material handling, by Miyazaki et al.

(1988) and Philipoom et al. (1990). Philipoom et al. (1990) develop a model for signal kanbans, and Miyazaki et al. (1988) develop a model for supplier and withdrawal kanbans. The model structures are different, and the findings cannot be generalized to a dual-card Kanban system.

- Several models assume station independence. Askin et al. (1993) and Mitwasi and Askin (1994) assume external demand. The demand for each stage is externally generated and with the assumption of sufficient capacity, the multistage system can be modeled as independent stages linked by their proportional demand rates. Bitran and Chang (1987), Philipoom et al. (1990) and Philipoom et al. (1996) do not allow backorders. Under JIT system, all stages are integrally tied to each other and if one delays all the others may be affected. But, with the assumption of no backorders, this possibility is eliminated. Therefore, each stage can be modeled independently.

- Kimura and Terada (1981), Philipoom et al. (1990), and Li and Co (1991) assume that the capacity is unlimited. In that way, the formulation of the model becomes easier.

The limitations of the simulation models can be stated as follows:

- Even though simulation offers a number of advantages by restricting the number of assumptions of the system, it takes a great deal of time to develop simulation programs. Apart from reaching at optimality, one must test many alternative shop configurations. Almost all of the simulation models assume demand to be exponentially distributed. More realistic assumptions on the demand distribution are necessary.

- Simulation studies to determine the interaction of kanban sizes and number of kanbans are needed. Berkley (1996) study the effect of kanban sizes on system performance, but he assumes that the kanban sizes are set regardless of the demand distribution. Studies with more general assumptions on determination of the kanban sizes should be performed.

2.2 Determining the Kanban Sequences

In JIT systems, the final assembly schedule determines production schedules for all of the stages in the facility. Once the assembly line is scheduled, it is assumed that the sequence propagates back through the system. Kanbans in the rest of the shop are processed in order in which they are received, i.e. (FCFS). However, there are several studies in the literature that test this assumption.

Lee (1987) compares FCFS, shortest processing time (SPT), SPT/LATE, higher pull demand (HPF), and HPF/LATE in a flow shop using dual-card Kanban system with fixed order points. Simulation results show that SPT and SPT/LATE outperform in several performance measures considered such as production rate, utilization, queue time, and tardiness. The same system is simulated to see the effect of different job mixes, pull rates, and number of kanbans and kanban sizes on the system performance.

Berkley and Kiran (1991) compare the performance of SPT, FCFS, SPT/LATE, and FCFS/SPT in a dual-card Kanban system with constant withdrawal cycles. They find that contrary to the conventional results, SPT creates the largest average output material queues and in process-inventories, and FCFS and FCFS/SPT creates the least. FCFS and FCFS/SPT outperform other two rules.

Berkley (1993) compares the performance of FCFS and SPT in a single-card Kanban system with varying queue capacities and material handling frequencies by a simulation model. The results of the study are compared with the results of Berkley and Kiran (1991). It is shown that the results are due to material handling mechanism used in both of the models.

Lummus (1995) simulates a JIT system to investigate the effect of sequencing on the performance of the system in a multi-item, multi-stage assembly-tree structure setting. The author use three sequencing rules, which are Toyota's goal chasing algorithm (a detailed explanation of the algorithm can be found in Monden(1981)), demand-driven production and producing all the jobs of the same kind, and study their effects for different sequences given various setup and processing times. She concludes that the sequencing method selected affects the performance of the system.

The problem of production leveling through scheduling is crucial to Kanban systems. Sequencing in kanban-controlled shops are more complex when compared to conventional sequencing problems as kanbans do not have due dates and kanban-controlled shops have station blocking [Berkley(1992)]. Station blocking can be described as the idleness of a stage due to full outbound inventories. Although there are several studies on kanban scheduling, the rules used in these studies are simple dispatching rules. More sophisticated scheduling rules should be used to determine the effects of scheduling on the performance of the system.

Detailed reviews of JIT and Kanban systems can be found in Berkley (1992), Groenevelt (1993), and Huang and Kusiak (1996).

2.3 Related Literature

To avoid ambiguity throughout the study, we will give brief reviews of due date estimation models and group technology.

2.3.1 Due Date Estimation

Due dates are treated in two ways in the literature; they are either externally imposed or internally set. For the internally set due dates, a flow time is estimated for each job and a due date is set accordingly. There are several models in the literature for due date estimation in job shop or flowlines. In this section, we will briefly review the work of Ragatz and Mabert (1984). The authors compare eight different due date estimation rules in a job shop setting on a simulation study. They find out that both job characteristics and shop status information should be used to develop due date assignment rules.

Information about workcenter congestion along the routing of a job is more useful than the general shop information. Moreover, the use of more detailed information in predicting flow times provides only marginal improvement in performance over other rules that use more aggregate information.

More detailed analysis on due date estimation can be found in Vig and Dooley (1993), Russell and Philipoom(1991) and Mahmoodi et al. (1990).

2.3.2 Group Technology

One of the key elements of JIT is group technology (GT). GT is a manufacturing philosophy that exploits similarities in product design and production process. With the application of GT, a wide range of benefits can be possible, including variety reduction, reduced setup times, lead time and in-process inventory. GT provides the flexibility that support the design and implementation of JIT. JIT includes a simplified production line and standardized products. GT can be used to form families and machine cells which would lead to standardized products and a simplified production line. A family is a group of parts that share the same setup, processing, routing, and so on.

When parts are classified into a families, family scheduling is applicable. A family scheduling procedure incorporates information about family membership and generates solutions which build on the elimination of setups by combining jobs from the same families. There are several studies on family scheduling. In our study, we use the findings of Wemmerlöv and Vakharia (1991). The authors compare four family-based scheduling procedures with four corresponding item-based scheduling procedures on a flowline manufacturing cell. They find that first-come-first-served-family (FCFS-F) is the best performer among the eight rules investigated. The authors conclude that for the conditions used in the study, family-based scheduling approaches can generate significant improvements with respect to flow time and lateness-oriented measures. Moreover, when setup times increase, the advantage in using family-based rules over item-based rules increases.

Detailed reviews on GT can be found in Offodile et al. (1994) and Gunasekaran et al. (1994). Review on GT scheduling can be found in Wemmerlöv and Vakharia (1991) and Russell and Philipoom (1991).

2.4 Summary

In this chapter, the literature on Kanban systems is reviewed. First, the models on determining the design parameters are explained briefly with emphasis on their limitations. A tabular format is used to compare the models. Then, the models for sequencing the kanbans are introduced. The results of these studies can be summarized as follows:

- In the existing literature most of the studies determine the kanban sizes and number of kanbans separately. In fact, number of kanbans required depends on kanban sizes and these parameters together affect the system performance. Therefore these parameters should be set simultaneously, not sequentially.
- None of the studies consider the impact of operational issues on design parameters. The sequencing in Kanban systems need more elaboration.
- Even though dual-card Kanban systems are periodic in nature, there are a limited number of studies on periodic review systems.

In the next chapter, an algorithm is proposed to eliminate the above cited limitations of the existing models. The proposed algorithm determines withdrawal cycle length, kanban sizes, and number of kanbans simultaneously in a periodic review Kanban system. It provides a feedback mechanism to evaluate the impact of operational issues such as scheduling and actual lead times on the design parameters.

Chapter 3

Problem Statement

In this study, we propose an analytical model to determine the withdrawal cycle length, kanban sizes and number of kanbans simultaneously in a periodic review Kanban system under imperfect production settings. With the proposed algorithm, we try to eliminate the deficiencies of Kanban system due to its strict assumptions and prerequisites by incorporating flexibility to the system design. We use the impact of operating issues such as scheduling and actual lead times on the design parameters. Moreover, we analytically study the effects of system parameters on system performance.

The rest of this chapter is organized as follows: In Section 3.1 the problem is defined, the motivating points are highlighted, and the underlying assumptions are explained. In Section 3.2 the proposed algorithm is discussed. First, the notation used is introduced in Section 3.2.1, then the algorithm is explained in detail. The chapter finishes with the concluding remarks.

3.1 Problem Definition

In this study, an analytical model is proposed to determine the fixed withdrawal cycle length, number of kanbans and kanban sizes simultaneously in a multi-item, multi-stage, multi-horizon periodic review Kanban system on a minimum cost basis.

The motivating points behind this study are as follows:

Even though setting the kanban sizes is one of the primary decisions that the designers of a Kanban system must address [Berkley(1996)], there are only a limited number of studies that determine the number of kanbans and kanban sizes simultaneously. The recent research has shown that there is a significant relationship between kanban sizes and production lead times, and therefore the shop congestion [Karmarkar and Kekre(1989)]. The existing studies reflect the relation among the kanban sizes and the average inventory, but for the other performance measures no clear relations are present. One of the purposes of this study is to investigate analytically the effect of the kanban sizes on the system performance by using several performance measures such as average in-process inventory, total backorder cost, fill rate and total setup time.

Almost none of the models consider sequencing and determining the design parameters simultaneously. It is generally assumed that once the number of kanbans are determined, FCFS can be used to schedule the jobs. In fact, the problem of production leveling through scheduling is crucial in Kanban systems. Selecting the proper scheduling rules becomes even more important in the case of imperfect production settings, i.e. settings with high setup times, high product variety, and etc. [Huang and Kusiak(1996)]. In this study, we consider the impact of operating parameters on design parameters.

There are a limited number of studies on periodic review systems. In fact, the review frequency is an important factor in determining the operating parameters, as there is a trade-off between inventory holding cost and setup cost. This trade-off should be reflected in the design of a Kanban system.

In the design of this proposed algorithm, we consider several points to increase the flexibility and adaptability of the Kanban systems. We allocate different number of kanbans at each inventory point to decrease the possibility of blocking and backlogs. We use the idea of transfer batch and process batch. In that way, the kanban sizes can be decreased to low levels and the setup times can be justified [Browne(1993)]. We estimate the lead times for each stage in terms of periods so that the accuracy of the estimation increases and the problems due to lead time estimation are minimized. We determine the Kanban sizes according to demand distributions so that the amount of remnants can be decreased.

The following assumptions are made through out the study:

- The system is a periodic review system.
- Demand is discrete and stochastic.
- The processing times and setup times are deterministic.
- Kanban processing times are equal to the time required to process all parts in a kanban. Half full kanbans are not allowed.
- The setups are sequence dependent. A major setup is required when the family being processed at a stage changes. Minor setups are assumed to be zero.
- All the parts in a family follow the same routing on a modified flowline. Modified flowline is similar to flowline in that the material flow is unidirectional, but contrary to flowlines, in modified flowlines stage skippings are allowed.
- The withdrawal lead time is zero.
- Backorder is allowed, but the backorder costs are higher when compared to inventory holding costs.
- The system is reliable, i.e. there are no machine breakdowns. Processing at each stage is carried out without defects.

- The bill of materials quantity is assumed to be one for each component.

Under these assumptions, the withdrawal cycle length, kanban sizes and number of kanbans are found. The parameters are as follows:

1. Costs terms:

i. unit inventory holding cost

ii. unit backorder (unmet demand) cost

2. family routings

3. processing times

4. sequence dependent setup times

5. demand distributions

The proposed algorithm generates several alternatives for withdrawal cycle length and kanban sizes, and chooses the best combination among these by comparing the total inventory holding and backorder costs of each alternative.

To determine the maximum inventory level Toyota formula is used. In Toyota formula, the maximum inventory level is calculated by using the expression,

$$\text{maximum inventory level} = na = DL(1 + s)$$

where,

n is the number of kanbans,

a is the kanban size,

D is demand,

L is lead time, and

s is the safety factor.

An important problem that arise with the Toyota formula is the estimation of the lead times. Lead time is not an attribute of the part, rather it is a property of the shop floor. Lead times vary greatly depending on capacity, shop load, product mix and batch sizes. [Karmarkar (1987), Karmarkar (1993) and Karmarkar et al. (1985)]. Second, it is shown in Karmarkar and Kekre (1989) that the number of kanbans and kanban sizes have a significant effect on the performance of the Kanban system. So, it is not possible to think L independent of these two variables. Therefore, the problem becomes a difficult one to solve.

We try to eliminate the first problem through lead time estimation. We estimate the flow times for each stage by using the expected period demand. There are several studies in the literature for flow time estimation, but they are not directly applicable to our system because of its periodic nature. Therefore, we modify the work-in-queue (WIQ) rule which is indicated by Ragatz and Mabert (1984) as one of the most promising rules of the due date estimation literature.

For the second problem, various combinations of n and a are tried, and their effect on the system is investigated through cost terms.

3.2 The Proposed Algorithm

3.2.1 Notation

The following notation is used throughout the thesis.

I		number of families
i		family index, $i = 1, .., I$
j		item index, $j = 1, .., \text{size}[i]$
M	:	number of stages
m	:	machine index, $m = 1, .., M$

T	withdrawal cycle length
t	period index, $t = 1, .., (nop \cdot H)$
A_{ij}^{T}	the set of alternatives of kanban size for item j of family i for the withdrawal cycle length of T
a_{ij}^{T}	kanban size for item j of family i for withdrawal cycle length T
B_{m}^{t}	the set of number of backorders of items in period t at stage m
B_{ijm}^{t}	the number of backorders of item j of family i in period t at stage m, i.e. $(i, j)th$ component of S_{m}^{t}
b_{ijm}	unit backorder cost of item j of family i at stage m
D_{ijm}^{t}	demand for item j of family i at period t (in terms of number of kanbans)
d_{ijm}^{t}	demand for item j of family i at period t at stage m
d_{ijm}^{u}	updated demand for item j of family i at stage m
F_{ij}^{m}	expected flowtime of item j of family i at stage m
FG_{ij}^{t}	the remnants of item j of family i at period t
H	planning horizon
h_{ijm}	unit inventory holding cost of item j of family i at stage m
I_{m}^{t}	the set of in-process inventories of items at the beginning of period t at stage m
I_{ijm}^{t}	the in-process inventories of item j of family i at the beginning of period t at stage m, i.e $(i, j)th$ component of I_{m}^{t}
IS_{m}^{t}	index set at stage m at period t
k_{m}	lead time estimation constant for stage m
$MAXINV_{ij}^{m}$	the maximum inventory level of item j of family i at stage m
n_{ijm}^{T}	number of kanbans for item j of family i at stage m for withdrawal cycle length T
nop :	number of periods per shift
P	the set of possible withdrawal cycle lengths
p_{ijm}^{u}	the updated processing time of item j of family i at stage m

p_{ijm}	processing time of item j of family i at stage m
PS_m^t	production set at stage m at period t
R_i	routing of family i
ST_m	sequence dependent setup time matrix at stage m
size[i]	number of parts in family i
S_m^t	the set of in-process inventories of items at the end of period t at stage m
S_{ijm}^t	the set of in-process inventories of item j of family i at the end of period t at stage m, i.e. $(i,j)th$ component of S_m^t
SCH_{ijm}^t	the number of item j of family i scheduled at stage m at period t
SS_m^t	schedule set at stage m at period t
$UNSCH_m^t$	the set of items that remain unscheduled at stage m at period t
w_{ijm}	the probabilistic weight of item j of family i at stage m
WIQ_m	$\text{Work} - \text{in} - \text{Queue}$ of stage m

3.2.2 The Proposed Algorithm

The main steps of the proposed algorithm are as follows:

STEP 1: For all the possible values of T in the set P find the number of periods per shift:

$$nop = T_{max}/T$$

where,

T_{max} is the longest withdrawal cycle length value in set P

STEP 1.1: Call Procedure FEASIBILITY.

STEP 1.2: Call Procedure LEADTIME.

STEP 1.3: For each item generate the set of alternatives of kanban size,

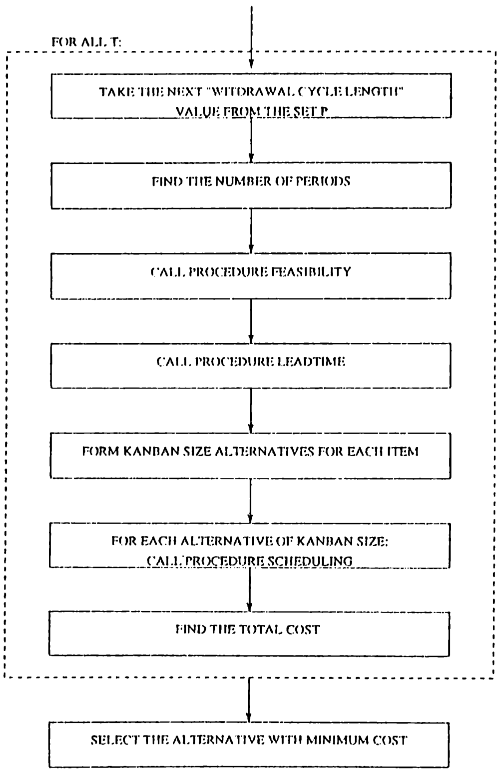

Figure 3.1: Flowchart of the Algorithm

A_{ij}^T by using the maximum inventory levels at the final stage:

$$A_{ij}^T = \{a_{ij}^T : a_{ij}^T = \lceil \frac{\text{MAXINV}_{ij}^1}{c} \rceil, \forall c = \{1, 2, 4, \cdots, 2^k\}\}$$

where,

k is a constant, and

$\lceil \cdot \rceil$ gives the smallest integer greater than or equal to the operand.

STEP 1.4: For each alternative of kanban size, $a_{ij}^T \in A_{ij}^T$, call Procedure SCHEDULING.

STEP 1.5: Calculate the total cost for T by adding up inventory holding and backorder costs at each stage.

STEP 2: Select the minimum cost alternative to find the withdrawal cycle length, kanban sizes and number of kanbans.

The flowchart of the proposed algorithm is given in Figure 3.1. The procedures used in the algorithm are explained in detail in next sections.

3.2.3 Procedure FEASIBILITY

This procedure checks if the selected withdrawal cycle length is operationally feasible or not. The steps of the procedure are as follows:

STEP 1: For all items, calculate a lower bound for period demand by using demand distribution and *nop*.

STEP 2: For all stages, $m = 1, ..., M$, do

> **STEP 2.1:** Form the schedule set with the lower bounds found in STEP 1 and the routing information.
>
> **STEP 2.2:** Call Procedure MAKESPAN to find the makespan of the schedule set.

STEP 3: Find the maximum makespan over all stages.

STEP 4: If the maximum makespan is longer than T, give a message that indicates a revision in daily plan.

> **STEP 4.1:** If T is equal to T_{max}, EXIT
>
> **STEP 4.2:** Else among the feasible set, select the alternative with minimum cost combination.

STEP 5: else T is feasible.

This procedure is used to quick check the feasibility of the withdrawal cycle length. If the cycle length is not long enough to produce even the lowest production amount possible, then using this withdrawal length will not be operationally feasible, as due to backorders the system will be blocked after some time. Therefore to avoid system blockage, if the makespan at any of the stages is longer than the withdrawal cycle length, we do not shorten the

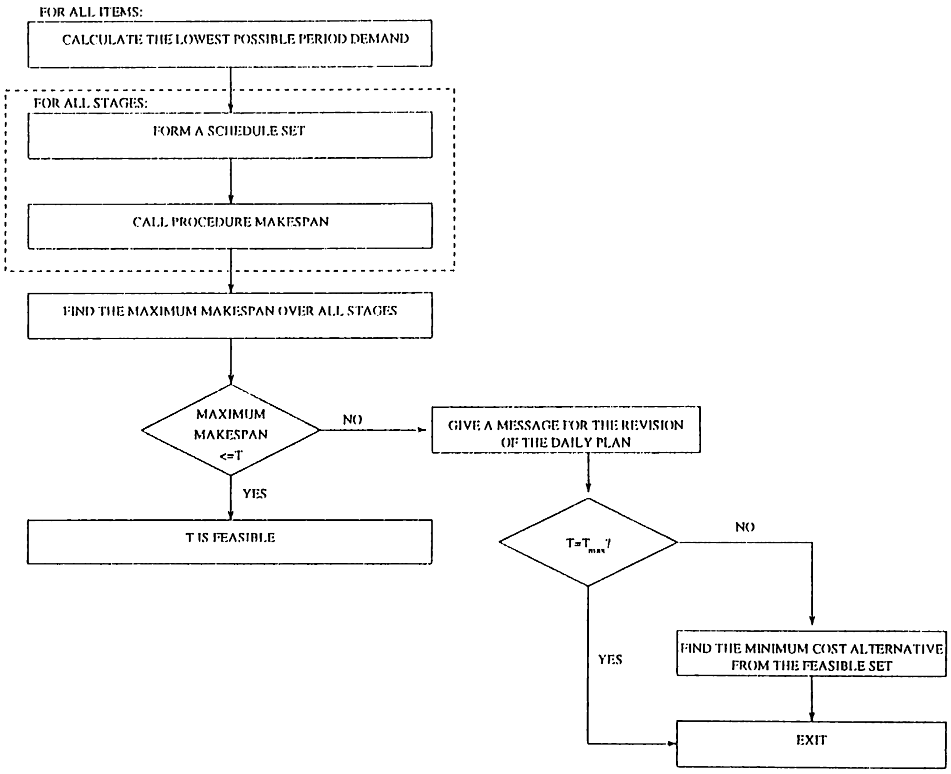

Figure 3.2: Flowchart of the Procedure FEASIBILITY

withdrawal cycle length any more and make the cost comparison among the withdrawal cycle lengths that are operationally feasible.

The flowchart of the Procedure FEASIBILITY is given in Figure 3.2.

3.2.4 Procedure MAKESPAN

This procedure finds the makespan of a schedule set by using nearest neighbor (NN) heuristic. The steps of the procedure are as follows:

STEP 1: Group all the jobs in the schedule set to their associated families.
STEP 2: Sort all the jobs in each family in nondecreasing order with respect to their slacks.
STEP 3: Find the order of families by nearest neighbor.
STEP 4: Use the order found at STEP 3 to find the makespan of the schedule

set by adding up the setup times and processing times.

As the aim is to find the minimum makespan of a set of items with sequence-dependent setup times, intra-family splits are not allowed. Once the stage is set up for a family, all the items of the family are processed (STEP 1). Even though the sequence within a family does not affect the makespan as minor setups are assumed to be zero, in STEP 2 the jobs in each family are sorted in nondecreasing order of their slacks. As our system is assumed to be a reliable one this step seems needless, but for the unreliable systems it will be necessary to implement this step to deal with uncertainties. In STEP 3, NN heuristic is used to find the family sequences. NN is a procedure in which the sequence is constructed by the greedy approach of always selecting the shortest setup time not yet visited. NN is a myopic rule, but the tests on randomly-generated problems suggest that it produces solutions within 10 % of the optimum solutions for $n \leq 20$, but the performance deteriorates if there is a considerable variability in the setup matrix [Baker(1994)].

3.2.5 Procedure LEADTIME

This procedure finds the maximum inventory level of each item at each stage by using the expected period demands. The procedure has two levels. In the first level (STEP 1-STEP 4), lead times are estimated in terms of periods for each stage. In the second level (STEP 5-STEP 6), by using these estimates and Toyota formula, the maximum inventory levels are determined. The steps of the procedure are as follows:

STEP 1: For each stage, form a schedule set by using the expected daily demands of each item produced at the stage.

STEP 2: Call Procedure MAKESPAN to find the makespan of the schedule set.

STEP 3: Let

$$F_{ij}^{m} = k_m * WIQ_m$$

where,

WIQ_m is the sum of the setup and processing times of the items in the queue of the stage.

STEP 4: Determine the lead time of stage m:

STEP 4.1: If F_{ij}^m is longer than the longest possible withdrawal cycle length value, T_{max}, in set P then

$$L_m = \frac{F_{ij}^m}{T_{max}}$$

STEP 4.2: else

$$L_m = 1$$

STEP 5: For all items:

$$\text{MAXINV}_{ij}^1 = \lceil \text{expected period demand} * L_1 * (1 + s) \rceil$$

where,

$$\text{expected period demand} = \lceil \frac{\text{expected daily demand}}{nop} \rceil$$

STEP 6: For each stage, find the maximum inventory levels by propagating the value found at the last stage backwards.

For $m = 2, \cdots, M$

$$\text{MAXINV}_{ij}^m = \lceil L_m * \text{MAXINV}_{ij}^p \rceil$$

where,

p is the succeeding stage in the routing of family i

As one of the problems that arise with the Toyota formula is the estimation of the lead times, instead of taking the lead time as a parameter we develop a lead time estimation procedure that determines the lead time depending on the shop information. Ragatz and Mabert (1984) show that the rules that utilize shop information generally perform better than rules that utilize only job information. As our system is a periodic review system, in the literature there are no models that can be directly applied to our study. Therefore, we select one of the most promising rules of the due date estimation literature, WIQ, and modify it to adapt to periodic review systems.

Due to the periodic nature of our system each stage can be thought independent from each other. Therefore, we can estimate the lead times at each stage separately. In that way we increase the quality of the estimation. Also, as the lead times are estimated for each stage, the maximum inventory level for an item at each stage will change. In that way, we can allocate different number of kanbans at each stage for the same item and this will increase the flexibility of the system.

Through STEP 1 and STEP 4, a flow time is calculated for each stage by using WIQ method. For the proper functioning of the JIT system, a certain minimum amount of inventory must always be in the system. We use the expected demand values to calculate that amount. Once the flow times are estimated, they are converted into periods in STEP 4. In STEP 5 and STEP 6, by using the Toyota formula and the expected period demands the maximum inventory levels are calculated. We propagate the inventory backward in the system so that the upstream stages hold larger inventories than the downstream stages. As the information lead time of the Kanban systems are long, the upstream stages cannot react to changes in demand easily and the system becomes erratic. To minimize this erratic behavior, more inventories can be allocated to upstream stages.

3.2.6 Procedure SCHEDULING

The scheduling module finds the schedules for each stage in a period. There are three main levels in this module. In the first level, we try to schedule all the items in the schedule board. If this is not possible, we limit the schedule set to form a new set which is called production set. The production set determines the number of kanbans of each item that should be scheduled to prevent backorders in the next period. Then, we try to schedule all the items in the production set. If this is possible, for the remaining time we include items from the schedule board that are not yet scheduled to our production set. But if it is not possible to schedule all the items in the production set, we further limit the production set by means of a proposed index and form a new set which is called the index set. At this level, the aim is to decrease the backorders as much as possible. If backorders are inevitable, we try to minimize the backorder costs.

The levels of the procedure are further explained below:

LEVEL 1: Find the makespan for the complete schedule set SS^t_m, $MT(SS^t_m)$. If $MT(SS^t_m) \leq T$, then the schedule set is feasible. Fine tune this schedule (STEPS 1.2.2-1.2.3).

LEVEL 2: Else if $MT(SS^t_m) > T$ then find a subset of the schedule set, PS^t_m. If $MT(PS^t_m) \leq T$, schedule all the items in production set PS^t_m. For the remaining time, which is given by $(T - MT(PS^t_m))$, solve a set of knapsack problems for the items in $SS^t_m - PS^t_m$, i.e. items in set SS^t_m but not in set PS^t_m (STEPS 1.2.4-1.2.8).

LEVEL 3: If $MT(PS^t_m) > T$, then calculate the index for each item and form a new set IS^t_m according to this index. IS^t_m should be a subset of PS^t_m. For the remaining time solve a set of knapsack problems for the items in $PS^t_m - IS^t_m$ (STEPS 1.2.9-1.2.12). If no more items from the set $PS^t_m - IS^t_m$ can be scheduled for the remaining time, then solve a set of knapsacks for the items $SS^t_m - IS^t_m$.

The detailed steps of the procedure are as follows:

STEP 1: For all the periods do

STEP 1.1: Find the number of kanbans demanded at period t for each item.

STEP 1.2: For each stage

STEP 1.2.1: Update schedule set

$$SS^t_{ijm} = UNSCH^{t-1}_{ijm} + D^t_{ijm}$$

STEP 1.2.2: Call Procedure MAKESPAN with the current schedule set, SS^t_m.

STEP 1.2.3: If MT(SS^t_m) $\leq T$, finetune the schedule and goto STEP 1.2.

STEP 1.2.4: Else find an updated demand for the next period for each item by using the most recent demand values (in terms of number of kanbans):

$$\begin{aligned} d^u_{ij} &= w * \frac{\sum_{i=1}^{n} (\text{demand for period } t-i)}{n} \\ &+ (1-w) * (\text{demand for period } t) \end{aligned}$$

where

w is a constant (weight), and

n is the number of past periods used in forecasting

STEP 1.2.5: Find the minimum amount of each item that should be scheduled to avoid future backorders by using the updated demands, on-hand inventories, demands for the period, and backorders.

$$\begin{aligned} \text{minimum amount to be scheduled} &= \text{updated demand} \\ &- (\text{inventory from period } t-1) \\ &+ (\text{demand at period } t) \\ &+ (\text{backorders from period } t-1) \end{aligned}$$

Mathematically,

$$PA_{ij} = d^u_{ijm} - S^{t-1}_{ijm} + D^t_{ijm} + B^{t-1}_{ijm}$$

STEP 1.2.6: Form a production set, PS^t_m. Include all the items with positive production amounts.

STEP 1.2.7: Call Procedure MAKESPAN with the new set, PS^t_m.

STEP 1.2.8: If MT$(PS^t_m) \leq T$

STEP 1.2.8.1: From the unscheduled jobs in $SS^t_m - PS^t_m$, select jobs for the remaining time.

STEP 1.2.8.1.1: Find the probabilistic weight for each unscheduled item, w_{ijm}.

The probabilistic weight is the difference between the probability that an item will be backordered in the next period multiplied with its backorder cost and the probability that the item will remain unused multiplied with its inventory holding cost.

$$\begin{aligned} w_{ijm} &= P(D^{t+1}_{ijm} > PS^t_{ijm} + I^t_{ijm}) \cdot b_{ijm} \\ &- P(D^{t+1}_{ijm} < PS^t_{ijm} + I^t_{ijm}) \cdot h_{ijm} \end{aligned}$$

STEP 1.2.8.1.2: For all unscheduled jobs with positive probabilistic weights find the updated processing times, p^u_{ijm}:

The updated processing time of item j of family i will be either the processing time of the item if at least one item of family i is already scheduled in the set PS^t_m, or the sum of processing time and family setup time (with respect to final family in the schedule) otherwise.

STEP 1.2.8.1.3: Solve knapsack problem with the updated processing times and probabilistic weights to find the next item to be scheduled.

MAX $\sum_{i=1}^{I} \sum_{j=1}^{size[i]} (X_{ijm} \cdot w_{ijm})$

subject to

$$\sum_{i=1}^{I} \sum_{j=1}^{size[i]} (X_{ijm} \cdot p^u_{ijm}) \leq (T - MK(PS^t_m))$$

$$X_{ijm} \leq RM_{ijm}$$

$$X_{ijm} \text{ is integer}$$

where,

RM$_{ijm}$ is the amount of item j of family i in the set $SS^t_m - PS^t_m$,

RM$_m$ is the set $SS^t_m - PS^t_m$, and

X_{ijm} is the number of items j of family i at stage m that should be included to PS^t_m from the set RM_m

The right hand side of the first equation gives the remaining time. It is the difference between the withdrawal cycle length and the makespan of the schedule set PS^t_m. Repeat STEP 1.2.8.1 until the remaining time is less than the minimum updated processing time.

STEP 1.2.8.1.4: If an item with a new setup is selected in knapsack problem, include only this item to PS^t_m and goto STEP 1.2.7. That is, if a new setup can be justified, then the updated processing times of items will change and the analysis should be repeated.

STEP 1.2.8.2: Update the slack values of the unscheduled jobs.
STEP 1.2.8.3: Goto STEP 1.2.

STEP 1.2.9: Else update PS^t_m. If the production amount of an item is bigger than its expected demand, set the production amount to the expected demand.

STEP 1.2.10: Call Procedure MAKESPAN with the updated PS^t_m.

STEP 1.2.11: If $MT(PS^t_m) \leq T$, repeat STEP 1.2.8.1-STEP 1.2.8.3 for the updated PS^t_m.

STEP 1.2.12: else for all $i = 1, ..., I$ and $j = 1, ..., J$ do

$$\alpha_{ijm} = \frac{\text{sequence-dependent setup time of family } i}{\text{total backorder cost of the family } i} + \frac{p_{ijm}}{(1 + L_{ijm}) \cdot b_{ijm}}$$

where,

total backorder cost of the family $i = \sum_{j=1}^{size[i]} \sum_{s=0}^{L_{ijm}} (s+1) \cdot Q_{ijm}[s] \cdot b_{ijm}$,

L_{ijm} be the maximum slack of item (i, j) at stage m, and

$Q_{ijm}[s]$ is the number of kanbans for item (i, j) that is backlogged for s periods at stage m

In the Kanban systems, the in-process inventories should always be full. Even if there is no demand, there may be an order for an item which has lower in-process inventory than the maximum inventory level. For these items, there is no backorder, so their slacks are expressed as zero and $Q_{ijm}[0]$ is the amount needed to fill up the in-process inventories.

STEP 1.2.12.1: Select the item with the smallest index and include it to the index set, IS_m^t. Add the processing time and set-up time of the selected item to the total time. Repeat STEP 1.2.12.1 while the total time is less than β% of T. Recalculate index at each repetition.
STEP 1.2.12.2: Call Procedure MAKESPAN with the new schedule set, IS_m^t.
STEP 1.2.12.3: For the remaining time, select jobs from the unscheduled job set, $PS_m^t - IS_m^t$.

STEP 1.2.12.3.1: Find the probabilistic weight for each unscheduled item.
STEP 1.2.12.3.2: For all unscheduled jobs find the updated processing times.
STEP 1.2.12.3.3: Solve Knapsack problem with the updated processing times and probabilistic weight to determine the next item to be scheduled.

MAX $\sum_{i=1}^{I}\sum_{j=1}^{size[i]}(X_{ijm}\cdot w_{ijm})$
subject to

$$\sum_{i=1}^{I}\sum_{j=1}^{size[i]}(X_{ijm}\cdot p^{u}_{ijm})\leq(T-MT(IS^{t}_{m}))$$

$$X_{ijm}\leq RM_{ijm}$$

$$X_{ijm}\text{ is integer}$$

where,

RM_{ijm} is the amount of item j of family i in the set $PS^{t}_{m}-IS^{t}_{m}$,

RM_{m} is the set $PS^{t}_{m}-IS^{t}_{m}$, and

X_{ijm} is the number of items j of family i at stage m that should be included to PS^{t}_{m} from the set RM_{m}

Repeat STEPs 1.2.12.2- 1.2.12.3 until the remaining time is less than the minimum updated processing time.

STEP 1.2.12.3.5: If an item with a new setup is selected in knapsack problem, include only this item to IS^{t}_{m} and goto STEP 1.2.12.2. That is, if a new setup can be justified, then the updated processing times of items will change and the analysis should be repeated.

STEP 1.2.12.4: If time remains, select jobs from the unscheduled job set, $SS^{t}_{m}-IS^{t}_{m}$.

STEP 1.2.12.4.1: Find the probabilistic weight for each unscheduled item:

$$w_{ijm}=b_{ijm}-h_{ijm}$$

STEP 1.2.12.4.2: For all unscheduled jobs find the updated processing times.

STEP 1.2.12.34.3: Solve Knapsack problem with the updated processing times and probabilistic weight to determine the next item to be scheduled.

Repeat STEP 1.2.12.4 until the remaining time is less than the minimum updated processing time.

STEP 1.2.12.5: For the unscheduled items update the slack values.

STEP 1.3: Find the backorder and inventory holding cost over all stages.

STEP 1.4: Find the total setup time.

In the scheduling module, the main goal is to complete as many items as possible from the schedule board in the given period. If it is possible to complete the whole set, the second and third levels of the module are not used. But if it is not possible to complete all the items, then a subset of this schedule set is chosen at the second and third levels. The flowchart of the scheduling module is given in Figure 3.3.

In Level 2, an updated demand is calculated. There are two reasons for this:

- To lower the probability of backorder. As the whole schedule set SS_m^t cannot be completed within the given period, a subset of it should be selected. This subset should reflect the demand trend, therefore the demands for the current period and last n periods are used.
- To incorporate a global perspective into the schedule. Till now, the stages are thought independent of each other. This is a reasonable assumption, as there were no backorders. But, as the whole schedule set cannot be completed, backorders will be inevitable. Therefore, to deal with this interdependency among the stages, a production amount that considers the expected demand in the future is used.

So, a production amount is found for each item by using the demand estimate, current inventory, and backorders. This production amount is the minimum amount that should be produced in this period. A new schedule set is formed in STEP 1.2.6 by using this production amount. If this set which is a subset of the former one can be completed within the period, a further analysis is done to include jobs from unscheduled set into schedule set (STEP 1.2.8). In this analysis, the aim is to select the most profitable items among the unscheduled ones. For each item, by using its probability distribution, the backorder probability and inventory holding probability are calculated. These probabilities are multiplied with their associated costs. The probabilistic weight together with the updated processing time is used to decide if it will be profitable to produce the item in the remaining time slot. In case the production set cannot be completed within the period, the third level of the procedure is used. In this level, we propose a new index that considers a possible trade off between a setup time lost for a family with the urgency of an item. Prior to the usage of the index the production set is updated. If any of the items have a lower expected demand than its production amount, the production amount for the item is set to the the expected demand. In the index, backorder costs are assumed to be the weights for each item. For the families, the total backorder cost over all items is considered as the weights. First term of the index allocates the family setup to each item in the family. According to this index, an initial schedule set is formed. We use the index to form the initial set till $\beta\%$ of the period length is occupied and for the remaining time a probabilistic analysis similar to the one in Level 2 is used. The only difference between the probabilistic analysis used in Level 2 and Level 3 is that, in Level 2 gain can be negative or positive, while in Level 3 it is strictly positive due to updating the schedule set prior to index calculation.

β is a number between 0 and 100, and it determines the amount of knapsack problems that will be solved. Even though the index is dynamic, it is myopic in nature. So to decrease the problems caused by the myopicity of the index, knapsack formulations are used. There is a trade-off between using the index and knapsack problems. The index is faster but the knapsack problems eliminate the myopicity. When β is small, we solve more knapsacks so the computation time increases but the solution is less myopic. When β is big, a bigger portion of the period is filled by using the index so the computation time decreases.

3.3 Summary

In this study an analytical model is proposed to determine the fixed withdrawal cycle length, number of kanbans and kanban sizes, and kanban schedules simultaneously in a multi-item, multi-stage, multi-horizon periodic review Kanban system under an imperfect production setting.

The proposed algorithm is designed to increase the flexibility of the system. With the proposed algorithm:

- Different number of kanbans can be allocated at each inventory point to decrease the possibility of blocking and backlog.
- The idea of transfer batch and process batch are introduced.
- Lead times are estimated for each stage in terms of periods so the accuracy of the estimation increases and the problems due to lead time estimation are minimized.
- Kanban sizes are determined according to demand distributions so that the amount of remnants can be decreased.

With the index that we propose, we consider the possible trade-off between the setup times and backorders. Even though the stages are scheduled separately, they are not independent due to the look ahead in the proposed algorithm. Therefore, the proposed algorithm eliminates the possibility of total blockage of the system.

In the next chapter, the efficiency of the proposed algorithm will be tested through an experimental design.

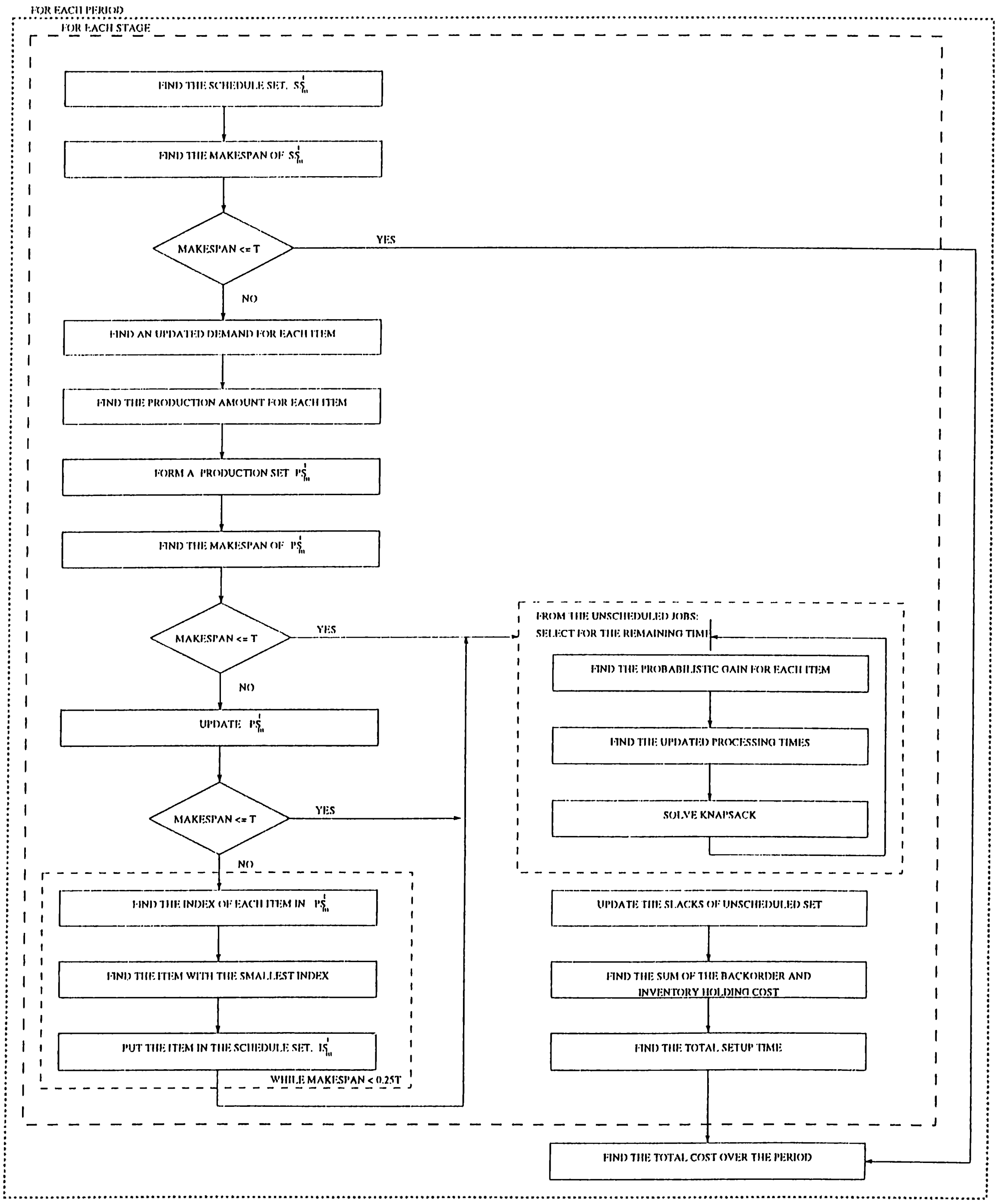

Figure 3.3: Flowchart of the Procedure SCHEDULING

Chapter 4

Experimental Design

In this chapter the efficiency of the proposed algorithm is tested. We compare the performance measure values found by the proposed algorithm with the values found by methods in the existing literature. All the algorithms are coded in C language and compiled with Gnu C compiler. The IP formulations in the proposed algorithm are solved by using callable library routines of CPLEX MIP solver on a Sparc station 10 under SunOS 5.4.

In the next section, the experimental setting is explained and the methods that we compare the proposed algorithm with are described. In Section 5.2, the results are presented and discussed. The final section summarizes this chapter.

4.1 Experimental Setting

To test the efficiency of the proposed algorithm an experimental design with the factors given in Table 4.1 is performed. The experimental design is a 2^7 full-factorial design as there are seven factors with two levels each. Five replications are taken for each combination. Therefore, 640 different randomly generated runs are obtained.

Five performance measures are used. The inventory holding cost and

Factors	Definition	Low	High
A	Number of Families	4	7
B	Demand Mean, μ	25	40
C	Demand Variability, σ	6.3	8.8
D	Number of Parts in each family	UN $\sim$ [4,8]	UN $\sim$ [8,12]
E	Imbalance	balanced	unbalanced
F	S/P ratio	0.9	1.75
G	B/I ratio	2	4

Table 4.1: Experimental Factors

backorder cost are the sum of the inventory holding and backorder costs over all stages, respectively. Fill rate is the probability of an order being satisfied through the inventories and it is calculated only for the final stage. Setup time is the sum of the setup times at each stage. Setup time is a surrogate performance measure, i.e. as long as the system reacts timely, it is not important if the setup times are high or low. Finally, the run time is the computation time in seconds.

Brief explanations for the experimental factors are as follows:

- The number of families and the number of parts in each family affect the product mix and congestion of the shop floor. As the number of families increase, the setup requirement increases and the scheduling decision becomes more important.

- The demand mean and demand variability specify the mean and the standard deviation of the demand distribution. The probability mass function (pmf) of the demand distribution for low variability case is defined as:

$$f_D(d) = \begin{cases} 0.1 & , \quad D = UN \sim [\mu - 10, \mu - 7] \\ 0.2 & , \quad D = UN \sim [\mu - 6, \mu - 3] \\ 0.4 & , \quad D = UN \sim [\mu - 2, \mu + 1] \\ 0.2 & , \quad D = UN \sim [\mu + 2, \mu + 5] \\ 0.1 & , \quad D = UN \sim [\mu + 6, \mu + 9] \\ 0 & , \quad \text{otherwise} \end{cases}$$

where,

μ is the mean of the demand distribution

UN stands for the uniform distribution

This density states that 10% of the time demand will be distributed uniformly between $[\mu - 10, \mu - 7]$, and for another 20 % of the time it will be distributed uniformly between $[\mu - 6, \mu - 3]$ and so on.

The pmf of the demand distribution for high variability case is defined as:

$$f_D(d) = \begin{cases} 0.1 & , \quad D = UN \sim [\mu - 15, \mu - 11] \\ 0.2 & , \quad D = UN \sim [\mu - 10, \mu - 6] \\ 0.4 & , \quad D = UN \sim [\mu - 5, \mu + 4] \\ 0.2 & , \quad D = UN \sim [\mu + 5, \mu + 9] \\ 0.1 & , \quad D = UN \sim [\mu + 10, \mu + 14] \\ 0 & , \quad \text{otherwise} \end{cases}$$

- The fifth factor relates with the balance of the system. In the balanced case, the processing times of items have the same uniform distribution at each stage. In the unbalanced case, the processing times at the fourth stage (stage D in the routing) has a uniform distribution with a higher mean. Therefore, the fourth stage becomes a bottleneck stage and consequently the smooth material flow is disturbed.

- The sixth factor is used to determine the sequence-dependent setup times at each stage. The setup time has a uniform distribution. The lower bound, SL_m, and the upper bound, SH_m, of the uniform distribution

Family	Operation Sequence
1	A B C D E
2	A D
3	A C D E
4	A B D
5	A D E
6	A D
7	A C D

Table 4.2: Family Routings

are calculated by using the S/P ratio and the processing times at each stage. First, the average processing time of each family at each stage is calculated as follows:

$$\text{average processing time of family } i \text{ at stage } m = \frac{\sum_{j=1}^{size[i]} P_{ijm} * K}{size[i]}$$

where,

$$K \text{ is an estimated kanban size}$$

K is selected according to Factor D. When Factor D is low, this estimate is 25 and when it is high, it is 50. This different values are used to keep the ratio of the setup time to total time constant. Then, SL_m and SH_m values are calculated as follows:

$$SL_m = \text{S/P} \cdot \text{average processing time of family } i \text{ at stage } m \cdot 0.50$$

$$SH_m = \text{S/P} \cdot \text{average processing time of family } i \text{ at stage } m \cdot 1.50$$

- The seventh factor is B/I ratio. The backorder cost of an item is equal to the inventory holding cost times the B/I ratio:

$$b_{ijm} = \text{B/I} \cdot h_{ijm}$$

The parameters are generated as follows:

- There are five stages, denoted as A, B, C, D and E.

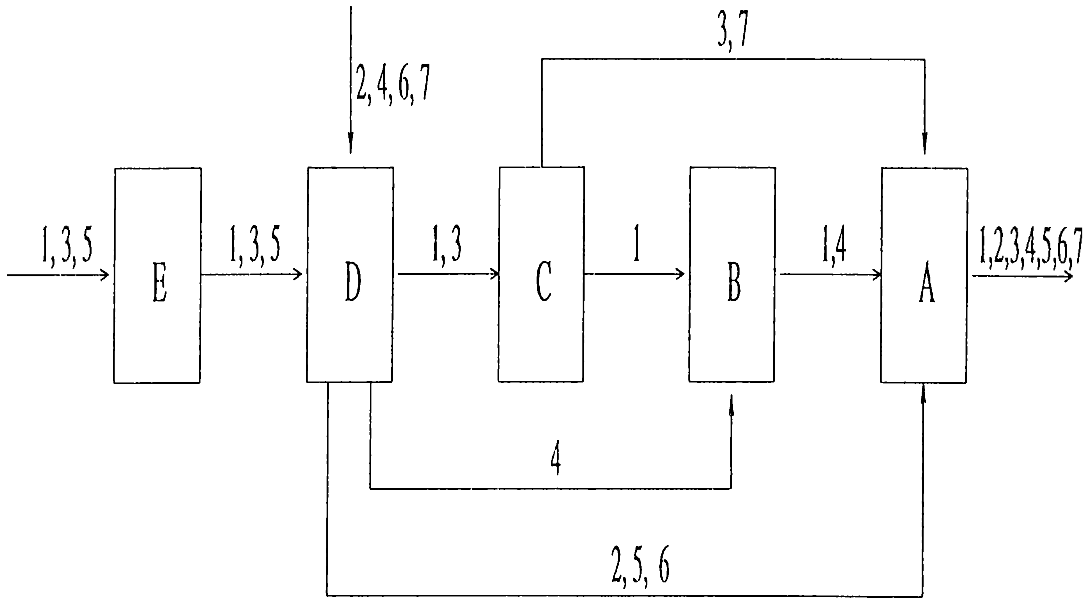

Figure 4.1: Layout

- The longest possible withdrawal cycle length is 480 minutes, i.e. a shift is equal to 480 minutes.
- The routings for families are fixed and given in Table 4.2 and Figure 4.1. When Factor A is at the low level, the first four families are used. The letters and the numbers in Table 4.2 and Figure 4.1 stand for the stages and jobs, respectively.
- The safety factor, s, is 0.05.
- The forecasting weight is 0.5 and three past periods are used for estimation.
- The lead time coefficient is same for all stages and equal to 1.01.
- The inventory holding costs are generated randomly from the interval UN $\sim$ [5,10].
- The processing times for balanced case are selected randomly from the interval UN $\sim$ [0.1,0.3]. For unbalanced case, the processing times at stage 4 are selected randomly from the interval UN $\sim$ [0.3,0.5] when the number of parts in each family are low, and from the interval UN $\sim$

[0.2,0.4] when it is high. Two different distributions are selected to keep the setup to processing time ratio in the system constant.

- As all the demand distributions are assumed to be identical, the kanban sizes for all items are the same. There are six alternatives for kanban sizes, i.e. $k = 5$ in STEP 1.3 of the proposed algorithm:

$$A_{ij}^T = \{a_{ij}^T : a_{ij}^T = \lceil \frac{\text{MAXINV}_{ij}^l}{c} \rceil, \forall c = \{1, 2, 4, 8, 16, 32\}\}$$

- For the withdrawal cycle length, six alternatives are generated such as $\{8, 4, 2, 1, 0.5, 0.25\}$ hours or $\{480, 240, 120, 60, 30, 15\}$ minutes. Therefore, to determine the values of the decision variables, the algorithms evaluate 36 alternatives and select the one with minimum inventory holding and backorder cost.

The experimental design is also applied to the commonly used sequencing rules in the literature. Four sequencing rules are considered, which are SPT, SPT-F, FCFS, and FCFS-F. It is shown by Berkley and Kiran (1991) that under periodic review Kanban systems, FCFS and FCFS/SPT perform better than SPT or SPT/LATE. Lee (1987) and Lee and Seah (1988), on the other hand, show that SPT and SPT/LATE perform better than FCFS. Therefore, we select FCFS and SPT/LATE as in the earlier studies their performance are justified. Wemmerlöv and Vakharia (1991) show that the family-based rules perform better than their corresponding item-based rules when the performance measure is flow time. Therefore, even though the family-based rules have not been used in JIT literature before, we select the corresponding family-based rules of FCFS and SPT/LATE to test the validness of this finding under kanban setting. For each rule, the selection is made among the items on the schedule board that have a corresponding full kanban in in-bound storage.

Even though FCFS ignores shop status and job characteristics, it is included as it is the most commonly used method in the literature. According to FCFS, the item that has the maximum slack, i.e. the item that arrives the scheduling board first, is processed first. The items in the same family that have the same slack are grouped and processed together. SPT is used as the tie-breaking rule.

That is, among the groups that have the same slack, the one with the shortest processing time is processed first.

FCFS-F schedules all families, and all items within each family according to FCFS. For each family, a slack value is calculated by summing up the slack values of the items in the family. Among the families, the one with the maximum slack is chosen first. If ties exist, SPT-F is used as the tie-breaking rule, i.e. the family with the minimum average processing time is processed first.

SPT/LATE is a modified version of the SPT which is used to identify the late items. It uses SPT rule to choose among the items. When the maximum slack of an item reaches to a level, FCFS replaces SPT and the items that are late are processed first. This SPT/LATE is a modified version of the rule used by Berkley and Kiran (1991). In the rest of the thesis, SPT/LATE is referred as SPT.

SPT-F schedules all families, and all items within each family according to SPT. For each family, an average processing time is calculated by taking the average of the processing times of the items in the family. The family with the minimum average processing time is processed first. FCFS-F is the tie-breaking rule.

4.2 Experimental Results

The overall results of the algorithms are summarized through Table 4.3 to Table 4.9. The tables show the minimum, average, and the maximum values for the performance measures for all of the algorithms. For the ease of explanation, first a representative graph for the cost terms is given in Figure 4.2. The x-axis corresponds to withdrawal cycle length alternatives. Alternative 1 corresponds to 480 minutes and alternative 6 corresponds to 15 minutes, i.e. alternatives are in decreasing order of withdrawal cycle lengths. For all the withdrawal cycle length alternatives, the kanban sizes are the same and equal to one.

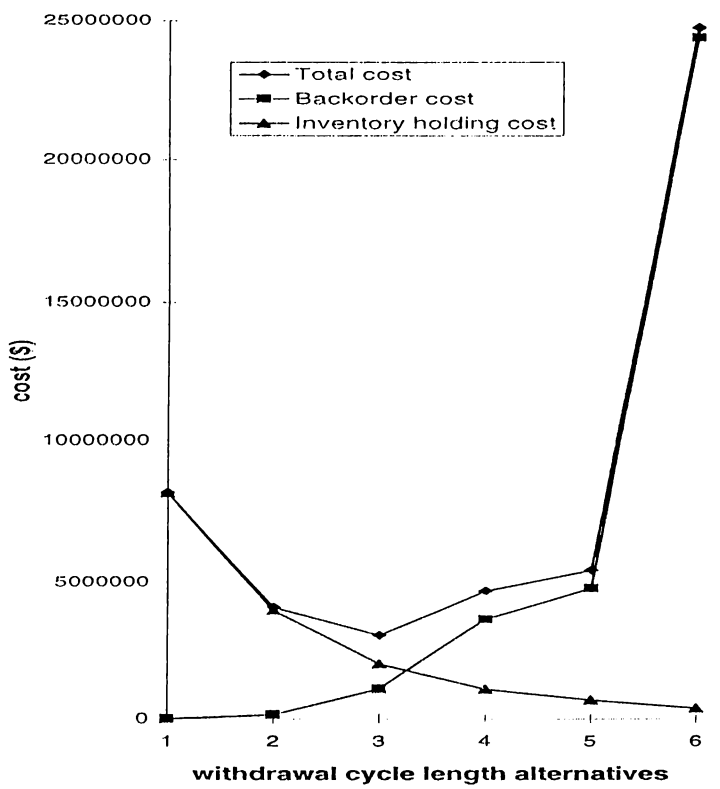

Figure 4.2: The detailed analysis of cost components

From Figure 4.2, we see that as the withdrawal cycle lengths get shorter, the inventory holding costs decrease and the backorder costs increase.

In Table 4.3, the minimum inventory levels of the algorithms are summarized. The minimum inventory levels are calculated by summing up the inventory holding costs when inventories are full over all stages over the planning horizon. For the Kanban system to work properly a minimun level of inventory should be kept in the system. Once this level is determined, the inventories should be kept full, and the deviations reflect the nervousness of the system that should be interpreted as lost production. Minimum inventory level is not a performance measure and it should not be interpreted alone.

Min. Inv. Level	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
Minimum	141552	94368	141552	94368	141552
Average	1361891	636928	1347406	**517241**	1537542
Maximum	10621760	5981832	10621760	4834472	12537344

Table 4.3: Comparison of the minimum inventory levels of algorithms

As long as the system performs good, the lower the minimum inventory level, the better the system performance. When we compare the algorithms we see that the minimum inventory level is maximum for the proposed algorithm. The minimum value is achieved by SPT-F. The family-based approaches hold less inventory than the item-based methods. This result can be interpreted as follows: as can be seen from Figure 4.2, for the same kanban sizes as the withdrawal cycle lengths get shorter, the inventory holding costs decrease. The backorder costs increase as setup to processing time ratio increases. For the family-based methods, the setup to processing time ratio is smaller, therefore family-based methods can force the withdrawal cycle lengths to shorter values. But for the item-based methods this ratio is higher, so longer withdrawal cycle lengths are chosen, and the minimum inventory level increases. This result is also shown in Table 4.5 where the number of instances of best withdrawal cycle lengths for the algorithms are given. When we analyze the Table 4.5, we see that the withdrawal cycle length selected is not robust to scheduling rule used. This result shows the impact of operating parameters on design parameters in decision making. In the existing literature, instantaneous material handling and FCFS is used, but when we consider the table, we see that this combination is not effective. In fact, item-based rules perform well when the withdrawal cycle lengths are long enough so that the setup times can be justified and the items can be highlighted. Family-based rules, on the other hand, prefer shorter withdrawal cycle lengths. The proposed algorithm selects a different cycle length with respect to the system parameters.

In Table 4.4, the inventory holding costs are given. When we interpret

Total cost	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
Minimum	121822.9	74493.8	121822.9	74383.6	102888.4
Average	1111238	491503.4	1103024	407045.2	1226052
Maximum	9051021	4794965	9051021	3785738	10254816

Table 4.4: Comparison of the inventory holding costs of algorithms

Withdrawal Cycle Length (minutes)	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
480	14	0	14	0	54
240	156	11	146	5	62
120	260	98	266	55	164
60	170	133	174	73	219
30	40	277	40	245	137
15	0	121	0	261	4

Table 4.5: Comparison of the number of instances of withdrawal cycle lengths of algorithms

these values in conjunction with the minimum inventory levels, we see that even though the family-based methods hold a lower level of inventory, they cannot react timely as on the average 77.1% and 78.7% of inventories are full for FCFS-F and SPT-F, respectively. The best performance is achieved by FCFS, as on the average 81.6% of inventories are full. For SPT this ratio is 77.2% and for the proposed algorithm it is 79.7%. These figures are obtained by dividing the average inventory holding cost to average minimum inventory level of each algorithm.

This finding is consistent with the findings of Berkley and Kiran (1991) and Wemmerlöv and Vakharia (1991). In a simulation study, Berkley and Kiran conclude that '...the use of SPT to be disruptive to the coordination of the kanban-production control system. ... Further, SPT has the greatest average finished-goods withdrawal kanban waiting times.'. Wemmerlöv and

Backorder cost	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
Minimum	0	0	0	0	0
Average	405924.4	273012.2	394840.4	280428.2	**85198.32**
Maximum	8909065	4503146	8578615	4864888	7630009

Table 4.6: Comparison of the backorder costs of algorithms

Fill rate	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
Minimum	0.597	0.751	0.618	0.711	0.520
Average	0.9743	0.9687	0.9736	0.9704	**0.9906**
Maximum	1	1	1	1	1

Table 4.7: Comparison of the fill rates of algorithms

Vakharia show that the family-based scheduling rules can generate significant improvements with respect to flow time and lateness-oriented performance measures over item-based rules.

Table 4.6 shows the backorder costs for the algorithms. The proposed algorithm has the minimum average backorder cost. In terms of backorder costs, the family-based methods work better then the item-based methods. The second best average performance is achieved by FCFS-F.

In Table 4.7, the comparison of the fill rates of algorithms is given. Again, the best average performance is achieved by the proposed algorithm. Among the other algorithms, item-based methods perform better, and among them FCFS performs better than SPT.

Table 4.8 compares the total setup times of algorithms. On the average, the proposed algorithm uses the minimum setup time. For the rest, as expected, family-based methods have lower average setup times. Even though setup time is a surrogate measure, Wemmerlöv and Vakharia (1991) have shown that the ability to avoid setups is related with the performance of scheduling

Setup time	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
Minimum	690.7	476.3	690.7	682.3	0
Average	2232.47	2158.68	2252.68	1970.37	**1837.90**
Maximum	3854.80	3451.61	3854.80	3401.99	4065.20

Table 4.8: Comparison of the total setup times of algorithms

Run time	FCFS	FCFS-F	SPT	SPT-F	Proposed Algorithm
Minimum	7.48	4.84	6.64	4.19	19.05
Average	19.80	23.48	20.20	**16.20**	154.98
Maximum	40.64	66.58	44.41	38.46	1850.34

Table 4.9: Comparison of the run times of algorithms

procedures in general. The proposed algorithm decreases the total setup time, and therefore increases the total available production time and the performance of the system in terms of other measures improve.

And finally, in Table 4.9 the comparison of the run times of algorithms is presented. The average run time of the proposed algorithm is the highest as expected but still well within the acceptable limits for such a decision making problem.

As a summary, the average performance of the proposed algorithm in terms of backorder cost, fill rate and setup times is better than the methods commonly used in the literature, but the average run time is considerably high. When we compare the existing methods we see that the average performance of FCFS is not bad as it is expected by Lee (1987) and Lee and Seah (1988). In both of these studies, the authors use global lateness information to sequence jobs on a flow line with a fixed quantity, nonconstant withdrawal cycle Kanban system and find that SPT/LATE performs significantly better the FCFS. In fact, our findings are consistent with the findings of Berkley and Kiran (1991) and Berkley (1993) who perform simulation studies on fixed withdrawal cycle, nonconstant quantity Kanban system.

To see the effects of the experimental factors on the system performance, for each factor we prepare the tables that show the performance of the system for low and high values of each factor. The results are summarized in the rest of this section.

- **Demand Mean and Demand Variability**

  When demand mean increases, the minimum inventory level increases. This result is obvious, as the minimum inventory level is calculated directly by using the demand mean. The backorder costs increase and the fill rates decrease as the system load increases.

  According to the existing literature when the demand variability in the system increase, the system performance should decline. When we investigate the tables in Appendix A we see that, for low demand mean this result is also achieved in our experimental design. But, surprisingly, for high demand mean some contradicting results are found. When we consider the high demand mean, low demand variability combination we see that for 16 out of 160 runs, all the algorithms behave erratic and give solutions worse than that found in the same combinations of high demand mean, high demand variability. For the rest of the runs, the results are as expected. All these 16 runs correspond to the same replication. As for the high variability case the lower bound for the demand distribution is smaller, there is chance that the high variability case might create lower demand values. In the corresponding replication, the demands are generated lower than the ones for the high variability case which seems as if the performance of this combination is better. In fact, if the runs are repeated with more replications, i.e. the effect of biased random numbers can be eliminated, this effect should diminish, and findings consistent with the existing literature should be found.

  When we consider the total setup times, at the first sight it seem as if contradictory results are found. But, total setup times should not be interpreted alone. Total setup times can be low due to system blockage which is in fact a highly undesirable situation. So, when we interpret all the tables together we can conclude that in some replications for high demand mean, low demand variability combination, system blockages might have occurred which causes unpredictable results in terms of performance measures.

- **The number of families and the number of parts in each family**

  When the number of families and/or number of parts in each family change(s), the system performance changes. By using the tables in Appendix B, we examine the effect of the number of families and/or number of parts in each family on the system performance. The results are given below.

  As the number of families and/or number of parts in each family increase(s), the average inventory holding cost and backorder cost increase. This result is obvious as cost terms are linear. But when we compare the percent increases, we see that the percent increase in cost terms are higher than the percent increase in the number of items in the system. Therefore, the increase in cost terms is also related with a decline in system performance. When the number of families increase, the product variety and the total setup times increase. When the number of parts in each family increase, the kanban variety increases. In a simulation study, Krajewski et al. (1987) show that Kanban systems perform well only under certain operating conditions. Under this set of conditions other approaches like MRP also perform well. The authors conclude that the operating conditions are key to improvements in system performance. Kanban system performs best when the setup times are small and the products are standardized. When the setup times and product/kanban variety increase, the operating conditions are disturbed so does the performance of the Kanban system.

  The fill rate decreases as a result of increase in any one of the above factors. This result is also related with the disturbance of the kanban conditions.

  When the number of families and the number of parts in each family increase, the setup times decrease. As the system congestion increases, some of the parts or families may be blocked. Therefore, due to blockages setup times can decrease. In fact when both factors are high, the average setup time is minimum.

The above results are applicable to all of the algorithms. Therefore, we can conclude that when the system is congested, the performance of the Kanban system declines. In fact, this result is consistent with Huang et al. (1983). Huang et al. conclude that for the proper functioning of Kanban system, excess capacity is necessary.

Congestion can create bottleneck stages which will affect the performance of Kanban systems. The detailed analysis for bottleneck stages are given under the subheading of imbalance.

If we compare the average performances of the algorithms, we see that in terms of fill rates and backorder costs the average performance of the proposed algorithm is better than the average performances of the other algorithms at any load level. Among the other algorithms, for highly loaded systems, the family-based approaches perform better, while in loose cases item-based methods perform better. This result is due to the effect of setup times on system performance.

- **Imbalance**

  The tables for the performance analysis of algorithms for imbalance in the system can be seen in Appendix C.

  When we consider the imbalance in the system it is seen that the balanced systems work better than the imbalanced ones under the kanban setting. The minimum inventory levels increase for all algorithms when there is imbalance in the system. For the proposed algorithm, the backorder cost and fill rate stay constant in unbalanced case. For the other algorithms, in spite of an increase in minimum inventory level, the system performance decline. We can conclude that:

  - Although we allocate different numbers of kanbans to each stage to increase flexibility, the performance of the Kanban system declines with an imbalance in the system.

- In an unbalanced setting, the proposed algorithm outperforms the other methods.
- The item-based approaches react to imbalance in the system better than the family-based approaches.

The first result is consistent with the findings of Huang et al. (1983) and Gupta and Gupta (1989). Gupta and Gupta show that in order to achieve the highest efficiency, all the stages of the Kanban system should be balanced. Huang et al. show that if bottlenecks occur regularly, the system performance declines. They conclude that additional kanbans at each stage are no help at all when there is a bottleneck in the system. Contrary to this finding, in the second result we show that by using a proper scheduling module the imbalance problem can be solved by adding kanbans as the performance of the proposed algorithm does not decline.

- **S/P ratio**

  When we analyze the systems with low and high S/P ratios we see that when S/P ratio increases, the minimum inventory levels and backorder costs increase, and fill rates decrease. That is to say, system performance declines when setup times become considerable. The tables for S/P ratio are summarized in Appendix D. This finding is consistent with the study of Mittal and Wang (1992). In a simulation study the authors show that after a threshold value for the setup times, the number of kanbans required for smooth flow tends towards infinite.

- **B/I ratio**

  When the B/I ratio increases, the minimum inventory levels and backorder costs increase for all algorithms. The lower the withdrawal cycle length, the better it is in terms of backorder costs. The increase in backorder costs will force the system to longer withdrawal cycle lengths and the minimum inventory levels will increase. The total backorder costs increase as the backorder costs of the items increase. But, when we consider the percent changes, the percent increase in backorder costs is

	Min. Inv. Level		Total Cost		Backorder Cost	
Factors	F	p	F	p	F	p
A	2654.033	0.000	1014.804	0.000	21.284	0.000
B	1780.115	0.000	675.024	0.000	21.034	0.000
C	94.517	0.000	37.912	0.000	15.918	0.000
D	2634.061	0.000	1012.479	0.000	21.089	0.000
E	52.430	0.000	15.243	0.000	0.000	0.999
F	172.280	0.000	102.721	0.000	10.536	0.001
G	0.002	0.963	0.940	0.333	2.383	0.123

Table 4.10: F values and Significance Levels (p) for ANOVA results of th proposed algorithm-I

equal to or smaller than the percent increase in B/I ratio. That means, on the average, in terms of backorders the system performance remains the same or increase.

For the proposed algorithm the minimum inventory level is almost the same in two cases. So the average performances are not affected too much. But for the other algorithms, this change is considerable, therefore the system performances in terms of fill rates and backorders increase.

The tables for the performance analysis of algorithms for B/I ratio can be seen in Appendix E.

4.3 ANOVA Results

We applied a two-way analysis of variance (ANOVA) test on the minimum inventory level and the performance measures of total cost, backorder cost, fill rate, setup time, and run time. The significance levels (p) and F values for these performance measures for the seven factors are given in Tables 4.10 and 4.11.

For the minimum inventory level, all the factors except the B/I ratio are significant with $p \leq 0.000$. As the minimum inventory levels are determined

	Fill rate		Setup time		Run time	
Factors	F	p	F	p	F	p
A	31.092	0.000	690.891	0.000	250.476	0.000
B	28.258	0.000	795.929	0.000	17.772	0.000
C	14.775	0.000	205.917	0.000	12.722	0.000
D	28.298	0.000	280.974	0.000	101.126	0.000
E	0.000	0.999	8.055	0.005	0.008	0.927
F	7.815	0.005	98.600	0.000	1.207	0.272
G	0.016	0.898	0.174	0.677	0.02	0.888

Table 4.11: F values and Significance Levels (p) for ANOVA results of the proposed algorithm-II

by using the demand and makespan values, the factors that affect these values also affect the minimum inventory level. Factors B and C directly affect the demand, and the others affect the makespan. Factors A and D affect the load on the system, hence the makespan. Factor E introduces imbalance to the system by increasing the processing times and factor F affects the setup times which in turn increase the flow time. Even though the selection of the minimum inventory levels are done based on cost criteria, the B/I ratio does not affect the minimum inventory level. The proposed algorithm uses the backorder costs only in the index. When we consider the index, we see that when B/I ratio increases, the index for all the items will decrease by the same ratio and the relative rankings of the index will not change. Therefore the selection criteria for the proposed algorithm does not change.

For the total costs, all the factors except the last is significant with $p \leq 0.000$. In fact, this analysis shows that in the proposed algorithm the ratio of the total backorder cost to total inventory holding cost is small. Backorder cost and fill rate are closely related with the reactiveness of the system. When the load in the system changes, the reactiveness of the system is also influenced. So, the factors that affect the load on the system also affect the backorder cost and fill rate. That is why the first four factors are significant with $p \leq 0.000$ for both of the measures, and the sixth factor is significant with $p \leq 0.001$

	Min. Inv. Level		Total Cost		Backorder Cost	
Factors	F	p	F	p	F	p
A	1248.393	0.000	2530.556	0.000	602.053	0.000
B	785.457	0.000	1893.015	0.000	525.738	0.000
C	0.254	0.615	167.103	0.000	146.010	0.000
D	1371.516	0.000	2638.183	0.000	595.738	0.000
E	8.387	0.004	16.776	0.000	3.940	0.048
F	27.251	0.000	64.882	0.000	17.839	0.000
G	43.115	0.000	101.468	0.000	27.628	0.000

Table 4.12: F values and Significance Levels (p) for ANOVA results of FCFS-I

and $p \leq 0.005$ for backorder cost and fill rate, respectively. Even though the imbalance in the system affects the system load, it does not affect the backorder cost and fill rate as the imbalance in the system is introduced in a middle stage and is compensated by introducing more in-process inventories at that stage.

For the setup times, the factors A, B, C, D, and F are significant with $p \leq 0.000$ and factor E is significant with $p \leq 0.005$. As expected, the S/P ratio affects the setup times. The processing time to setup time ratio becomes important when the system load changes. The remaining five factors directly affect the system load, therefore they are significant.

The ANOVA results for the run times show that the factors that affect the size of the problem are the first four factors with $p \leq 0.000$. Factors A and D affect the number of items in the system, and factors B and C affect the load of the system. In both cases scheduling decisions become critical, and the computation times are affected.

In the above analysis the B/I ratio does not affect any of the performance measures. One possibility is that the B/I ratio for both of the levels could be a low value so that its effect on the costs are negligible. To test it, we also applied the ANOVA to the most commonly used method in the literature, FCFS. The run times are not included as the computation times of the FCFS are so small.

	Fill rate		Setup time	
Factors	F	p	F	p
A	923.183	0.000	450.068	0.000
B	567.756	0.000	672.231	0.000
C	200.509	0.000	76.407	0.000
D	871.873	0.000	598.918	0.000
E	3.354	0.068	0.016	0.900
F	121.972	0.000	27.588	0.000
G	20.621	0.000	11.643	0.001

Table 4.13: F values and Significance Levels (p) for ANOVA results of FCFS-II

When we consider the results, we see that the B/I ratio is significant for all of the performance measures. Therefore, the results for the proposed algorithm are not due to the low B/I ratio levels. The proposed algorithm can react to changes more easily than the existing methods. We can conclude that, the proposed algorithm is more robust under the imperfect production settings.

4.4 Conclusion

In this chapter, the experimental design is presented. First, the experimental setting is explained. Then, the results are summarized and discussed, and the ANOVA tables for the proposed algorithm and FCFS are given and interpreted.

The results can be summarized as follows:

- The impact of operating issues on the design parameters is shown. We observed that the withdrawal cycle lengths are not robust to scheduling decisions.
- As the product variety increases, the product standardization decreases and the performance of the Kanban system declines. When the product variety increases, the repetitive nature of the system is disturbed. One

of the main assumptions of JIT is repetitive manufacturing. Therefore, the factors that adversely affect the repetitive nature of the system also affect the system performance.

- It is analytically observed that perfectly balanced lines outperforms the imbalanced ones even when the number of kanbans at each stage are different. But, contrary to the existing literature, we observed that by using a good scheduling rule, the performance of system can be improved even under the imbalanced setting.

- The factors that affect the system congestion (such as demand mean, number of families, number of parts in each family, and etc.) also affect the system performance.

- When setup times become considerable the system performance declines. For high setup cases algorithms that can decrease the total setup time perform better than the others, i.e family-based methods outperforms item-based methods.

Chapter 5

Numerical Example

In this chapter, we will discuss the detailed execution of the algorithms over a simple example. In the example, only the execution of the scheduling modules will be given. It is assumed that the withdrawal cycle length, kanban sizes and the number of kanbans are known.

The data for the example are given in Tables 5.1- 5.3. For this example the withdrawal cycle length is $T_{max}/2 = 480/2 = 240$ minutes and there is only one stage. So, the stage index for the notation given in Chapter 3 is eliminated. In Table 5.3, the sequence-dependent setup times are given. The first row in the table corresponds to the case where the stage is not initially setup to any of the families.

Before executing the algorithms, the definitions and formulae will be given. Moreover, some of common terms will be calculated.

The demand for each item should be converted to number of kanbans. It is calculated as:

$$D_{ij}^t = \lceil \frac{d_{ij}^t - FG_{ij}^{t-1}}{a_{ij}} \rceil$$

where,

d_{ij}^t is the period demand of item (i, j),

Family	1		2	3	
Part	1	2	1	1	2
Kanban size, a_{ij} (units)	10	10	10	10	10
Minimum inventory level (units/period)	20	20	20	20	20
Backorder cost, b_{ij} (\$/unit/period)	1	2	10	2	3
Processing times, p_{ij} (minutes/unit)	1.5	1	4	1	2

Table 5.1: Data for numerical example

D_{ij}^t is the period demand of item (i,j) in terms of number of kanbans

a_{ij} is the kanban size of item (i,j), and

FG_{ij}^t is the remnants of item (i,j)

The notation (i,j) is used to indicate the item j of family i. FG_{ij}^t is calculated as follows:

$$FG_{ij}^t = D_{ij}^t \cdot a_{ij} - (d_{ij}^t - FG_{ij}^{t-1})$$

Therefore, the number of kanbans for each item in the first period:

$$D_{11}^1 = \lceil \frac{30-0}{10} \rceil = 3 \ , \ FG_{11}^1 = 3 \cdot 10 - 30 = 0$$

$$D_{12}^1 = \lceil \frac{27-0}{10} \rceil = 3 \ , \ FG_{12}^1 = 3 \cdot 10 - 27 = 3$$

$$D_{21}^1 = \lceil \frac{23-0}{10} \rceil = 3 \ , \ FG_{21}^1 = 3 \cdot 10 - 23 = 7$$

$$D_{31}^1 = \lceil \frac{25-0}{10} \rceil = 3 \ , \ FG_{31}^1 = 3 \cdot 10 - 25 = 5$$

$$D_{32}^1 = \lceil \frac{30-0}{10} \rceil = 3 \ , \ FG_{32}^1 = 3 \cdot 10 - 30 = 0$$

and for the second period:

$$D_{11}^2 = \lceil \frac{7-0}{10} \rceil = 1 \ , \ FG_{11}^2 = 1 \cdot 10 - 7 = 3$$

$$D_{12}^2 = \lceil \frac{13-3}{10} \rceil = 1 \ , \ FG_{12}^2 = 1 \cdot 10 - 10 = 0$$

Period Demand, d_{ij}^t	Family 1		Family 2	Family 3	
	1	2	1	1	2
1	30	27	23	25	30
2	7	13	32	27	20
3	15	8	20	17	11

Table 5.2: The period demands for the items

	1	2	3
0	5	5	5
1	0	40	15
2	5	0	5
3	10	20	0

Table 5.3: The sequence-dependent setup times (in minutes)

$$D_{21}^2 = \lceil\frac{32-7}{10}\rceil = 3\ ,\ FG_{21}^2 = 3\cdot 10 - 25 = 5$$

$$D_{31}^2 = \lceil\frac{27-5}{10}\rceil = 3\ ,\ FG_{31}^2 = 3\cdot 10 - 22 = 8$$

$$D_{32}^2 = \lceil\frac{20-0}{10}\rceil = 2\ ,\ FG_{32}^2 = 2\cdot 10 - 20 = 0$$

and for the third period:

$$D_{11}^3 = \lceil\frac{15-3}{10}\rceil = 2\ ,\ FG_{11}^3 = 2\cdot 10 - 12 = 8$$

$$D_{12}^3 = \lceil\frac{8-0}{10}\rceil = 1\ ,\ FG_{12}^3 = 1\cdot 10 - 8 = 2$$

$$D_{21}^3 = \lceil\frac{20-5}{10}\rceil = 2\ ,\ FG_{21}^3 = 2\cdot 10 - 15 = 5$$

$$D_{31}^3 = \lceil\frac{17-8}{10}\rceil = 1\ ,\ FG_{31}^3 = 1\cdot 10 - 9 = 1$$

$$D_{32}^3 = \lceil\frac{11-0}{10}\rceil = 2\ ,\ FG_{11}^3 = 2\cdot 10 - 11 = 9$$

Let I^t be the inventory set at the beginning of the period t, S^t be the inventory set at the end of the period t and B^t be the backorder set at period

t. All these sets are expressed in terms of number of kanbans. Initially:

$$I^0 = \{2, 2; 2; 2; 2, 2\}$$

as the minimum inventory levels are 20 and the kanban sizes are 10 (minimum inventory level = kanban size × number of kanbans), and

$$B^0 = \{0, 0; 0; 0; 0, 0\}$$

where (;) is used to separate the families and (,) is used to separate the items within families.

Let I_{ij}^t be the inventory level of item (i, j) at the beginning of the period t, S_{ij}^t be the inventory level of item (i, j) at the end of the period t, and B_{ij}^t be the backorder of item (i, j) at period t. These will be calculated as follows:

$$I_{ij}^t = max(0, S_{ij}^{t-1} - D_{ij}^t - B_{ij}^{t-1})$$

$$S_{ij}^t = I_{ij}^t + SCH_{ij}^t$$

$$B_{ij}^t = max(0, D_{ij}^t + B_{ij}^{t-1} - S_{ij}^{t-1})$$

where,

SCH_{ij}^t is the number of kanbans of item (i, j) scheduled in period t

Backorder cost is calculated as follows:

$$\text{backorder cost} = \sum_{i=1}^{3} \sum_{j=1}^{size[i]} B_{ij}^t \cdot a_{ij} \cdot b_{ij}$$

And, the schedule set is calculated as:

$$SS_{ij}^t = UNSCH_{ij}^{t-1} + D_{ij}^t$$

where,

$UNSCH_{ij}^t$ is the number of unscheduled kanbans of item (i, j) in period t

Initially:

$$UNSCH^0 = \{0, 0; 0; 0, 0\}$$

The executions of the algorithms are given below:

EXECUTION OF SPT:

For SPT, the sequence will be determined as follows:

STEP 1: Calculate the updated processing time for each item:

$$P_{ij} = \text{sequence-dependent setup time of family } i + p_{ij} \cdot D_{ij}^t \cdot a_{ij}$$

STEP 2: Among the unscheduled items, select the item with minimum updated processing time. If ties exist, use FCFS as the tie-breaking rule.

The execution of the algorithm is as follows:

Period 1:

First, we should update the inventory levels, backorders, and schedule set, and calculate the backorder cost of the period:

$$I_{11}^1 = max(0, 2-3-0) = 0 \ , \ B_{11}^1 = max(0, 3+0-2) = 1$$

$$I_{12}^1 = max(0, 2-3-0) = 0 \ , \ B_{12}^1 = max(0, 3+0-2) = 1$$

$$I_{21}^1 = max(0, 2-3-0) = 0 \ , \ B_{21}^1 = max(0, 3+0-2) = 1$$

$$I_{31}^1 = max(0, 2-3-0) = 0 \ , \ B_{31}^1 = max(0, 3+0-2) = 1$$

$$I_{32}^1 = max(0, 2-3-0) = 0 \ , \ B_{32}^1 = max(0, 3+0-2) = 1$$

$$SS_{11}^1 = UNSCH_{11}^0 + D_{11}^1 = 0 + 3 = 3$$

$$SS_{12}^1 = UNSCH_{12}^0 + D_{12}^1 = 0 + 3 = 3$$

$$SS_{21}^1 = UNSCH_{21}^0 + D_{21}^1 = 0 + 3 = 3$$

$$SS_{31}^1 = UNSCH_{31}^0 + D_{31}^1 = 0 + 3 = 3$$

$$SS_{32}^1 = UNSCH_{32}^0 + D_{32}^1 = 0 + 3 = 3$$

$$\text{backorder cost} = 1 \cdot 10 \cdot 1 + 1 \cdot 10 \cdot 2 + 1 \cdot 10 \cdot 10 + 1 \cdot 10 \cdot 2 + 1 \cdot 10 \cdot 3 = 180$$

The schedule set will be:

$$SS^1 = \{3, 3; 3; 3, 3\}$$

As initially the stage is not set to any of the families, the updated processing times of the items will be:

$$P_{11} = 5 + 1.5 \cdot 3 \cdot 10 = 50$$

$$P_{12} = 5 + 1 \cdot 3 \cdot 10 = 35$$

$$P_{21} = 5 + 4 \cdot 3 \cdot 10 = 125$$

$$P_{31} = 5 + 1 \cdot 3 \cdot 10 = 35$$

$$P_{32} = 5 + 2 \cdot 3 \cdot 10 = 65$$

P_{12} and P_{31} are equal, therefore we should use FCFS to break the ties. As the slacks are also equal for these items, one of them will be selected randomly. Select the item with higher family index. Then item $(3,1)$ will be scheduled. The total processing time will be 35 minutes, and the updated processing times will become:

$$P_{11} = 10 + 1.5 \cdot 3 \cdot 10 = 55$$

$$P_{12} = 10 + 1 \cdot 3 \cdot 10 = 40$$

$$P_{21} = 20 + 4 \cdot 3 \cdot 10 = 140$$

$$P_{32} = 2 \cdot 3 \cdot 10 = 60$$

Therefore, item (1,2) will be scheduled second. The total production time will be 75 minutes and the updated processing times will be:

$$P_{11} = 1.5 \cdot 3 \cdot 10 = 45$$

$$P_{21} = 40 + 4 \cdot 3 \cdot 10 = 160$$

$$P_{32} = 15 + 2 \cdot 3 \cdot 10 = 75$$

The minimum updated processing time is for item $(1,1)$, so select this item for the third place. The total production time will be 120 minutes. If we continue in the same way, the fourth place will be occupied by item $(3,2)$, and the total production time will be 195 minutes. As only 45 minutes left, we cannot schedule the item $(2,1)$ as the updated processing time for even one kanban is 60 minutes. Therefore, all three kanbans will remain unscheduled.

So, at the end of the first period, the inventory will be:

$$S^1 = \{3, 3; 0; 3, 3\}$$

such that:

$$S^1_{11} = I^1_{11} + SCH^1_{11} = 0 + 3 = 3$$

$$S^1_{12} = I^1_{12} + SCH^1_{12} = 0 + 3 = 3$$

$$S^1_{21} = I^1_{21} + SCH^1_{21} = 0 + 0 = 0$$

$$S^1_{31} = I^1_{31} + SCH^1_{31} = 0 + 3 = 3$$

$$S^1_{32} = I^1_{32} + SCH^1_{32} = 0 + 3 = 3$$

and the set of unscheduled items will be:

$$UNSCH^1 = \{0, 0; 3; 0, 0\}$$

The stage is currently setup to family 3.

Period 2:

First, let us update the inventory levels, backorders, and schedule set, and calculate the backorder cost of the period:

$$I^2_{11} = max(0, 3 - 1 - 1) = 1 \ , \ B^2_{11} = max(0, 1 + 1 - 3) = 0$$

$$I^2_{12} = max(0, 3 - 1 - 1) = 1 \ , \ B^2_{12} = max(0, 1 + 1 - 3) = 0$$

$$I^2_{21} = max(0, 0 - 3 - 1) = 0 \ , \ B^2_{21} = max(0, 3 + 1 - 0) = 4$$

$$I^2_{31} = max(0, 3 - 3 - 1) = 0 \ , \ B^2_{31} = max(0, 3 + 1 - 3) = 1$$

$$I^2_{32} = max(0, 3 - 2 - 1) = 0 \ , \ B^2_{32} = max(0, 2 + 1 - 3) = 0$$

$$SS^2_{11} = UNSCH^1_{11} + D^2_{11} = 0 + 1 = 1$$

$$SS^2_{12} = UNSCH^1_{12} + D^2_{12} = 0 + 1 = 1$$

$$SS^2_{21} = UNSCH^1_{21} + D^2_{21} = 3 + 3 = 6$$

$$SS^2_{31} = UNSCH^1_{31} + D^2_{31} = 0 + 3 = 3$$

$$SS^2_{32} = UNSCH^1_{32} + D^2_{32} = 0 + 2 = 2$$

backorder cost $= 0 \cdot 10 \cdot 1 + 0 \cdot 10 \cdot 2 + 4 \cdot 10 \cdot 10 + 1 \cdot 10 \cdot 2 + 0 \cdot 10 \cdot 3 = 420$

The schedule set will be:

$$SS^2 = \{1, 1; \mathbf{3} + 3; 3; 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period.

The rest of the steps will not be shown explicitly. In the second period items $(1, 2)$, $(1, 1)$, $(3, 1)$, $(3, 2)$, and two kanbans of item $(2, 1)$ will be scheduled in the given order.

So, at the end of the second period, the inventories for the items will be:

$$S^2 = \{2, 2; 2; 3, 2\}$$

such that:

$$S_{11}^2 = I_{11}^2 + SCH_{11}^2 = 1 + 1 = 2$$

$$S_{12}^2 = I_{12}^2 + SCH_{12}^2 = 1 + 1 = 2$$

$$S_{21}^2 = I_{21}^2 + SCH_{21}^2 = 0 + 2 = 2$$

$$S_{31}^2 = I_{31}^2 + SCH_{31}^2 = 0 + 3 = 3$$

$$S_{32}^2 = I_{32}^2 + SCH_{32}^2 = 0 + 2 = 2$$

The unscheduled items will be:

$$UNSCH^2 = \{0, 0; 4; 0, 0\}$$

The stage is currently setup to family 2.

The backorder cost for two periods will be 600.

Period 3:

If we update the inventory levels, backorders and slacks, and calculate the backorder cost of third period:

$$I_{11}^3 = max(0, 2 - 2 - 0) = 0 \ , \ B_{11}^3 = max(0, 2 + 0 - 2) = 0$$

$$I_{12}^{3} = max(0, 2-1-0) = 1 \ , \ B_{12}^{3} = max(0, 1+0-2) = 0$$
$$I_{21}^{3} = max(0, 2-2-4) = 0 \ , \ B_{21}^{3} = max(0, 2+4-2) = 4$$
$$I_{31}^{3} = max(0, 3-1-1) = 1 \ , \ B_{31}^{3} = max(0, 1+1-3) = 0$$
$$I_{32}^{3} = max(0, 2-2-0) = 0 \ , \ B_{32}^{3} = max(0, 2+0-2) = 0$$
$$\text{backorder cost} = 4 \cdot 10 \cdot 10 = 400$$

In the rest of the chapter, we will not show the calculations of the schedule set. The schedule set will be:

$$SS^{3} = \{2, 1; \mathbf{4} + 2; 1; 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period. The total backorder cost over the three periods for SPT is 1000.

The schedules for the third periods will not calculated due to space limitations. The calculations are very similar to the previous two periods.

EXECUTION OF FCFS:

For FCFS, the sequence will be determined as follows:

STEP 1: Update the slack values for each item.
STEP 2: Among the unscheduled items select the item with maximum slack. If ties exist, use SPT as the tie-breaking rule.

Period 1:

As the slacks of all the items are the same for the first period, FCFS will not be able to differentiate among the items and the tie-breaking rule, SPT, will be used. Therefore, for the first period the same schedule as SPT will be achieved.

Period 2:

As the schedule set, and the updated inventory levels and backorders are the same for SPT and FCFS, the calculations will not be shown again. Only the slack calculations will be performed:

Let L_{ij} be the maximum slack of item (i, j). As there are no backorders for items $(1, 1)$, $(1, 2)$, and $(3, 2)$, the maximum slacks for these items will be zero. For item $(3, 1)$, the maximum slack will be one as only the orders for the current period are backlogged. All the previous backorders are satisfied at the beginning of the period. For item $(2, 1)$, the maximum slack is two as the backorder from the previous period could not be satisfied at the beginning of the second period. So:

$$L_{11} = L_{12} = L_{32} = 0$$

$$L_{21} = 2$$

$$L_{31} = 1$$

Therefore, item $(2, 1)$ will be selected first and a kanban will be scheduled. The total processing time will be 60 minutes as the stage is setup to family 3 at the beginning of the period. If we update the slacks:

$$L_{11} = L_{12} = L_{32} = 0$$

$$L_{21} = L_{31} = 1$$

As the slacks of items $(2, 1)$ and $(3, 1)$ are the same, the tie-breaking rule SPT will be used. As the stage is currently setup to family 2, the updated processing times of the items will be:

$$P_{21} = 4 \cdot 3 \cdot 10 = 120$$

$$P_{31} = 5 + 1 \cdot 1 \cdot 10 = 15$$

First, item $(3, 1)$ and then item $(2, 1)$ will be scheduled and the total processing time will be 215 minutes. As the slacks are the same for all the unscheduled items, SPT will be used. As only 25 minutes left, the updated processing times will be calculated for one kanban for each item:

$$P_{11} = 5 + 1.5 \cdot 1 \cdot 10 = 20$$

$$P_{12} = 5 + 1 \cdot 1 \cdot 10 = 15$$

$$P_{21} = 4 \cdot 1 \cdot 10 = 40$$

$$P_{31} = 5 + 1 \cdot 1 \cdot 10 = 15$$

$$P_{32} = 5 + 2 \cdot 1 \cdot 10 = 25$$

The processing times for item (1,2) and (3,1) are the same. One of them will be selected randomly. Select one with higher family index. So, item (3, 1) will be selected. The total processing time will be 230 minutes. If we calculate the updated processing times again:

$$P_{11} = 10 + 1.5 \cdot 1 \cdot 10 = 25$$

$$P_{12} = 10 + 1 \cdot 1 \cdot 10 = 20$$

$$P_{21} = 20 + 4 \cdot 1 \cdot 10 = 60$$

$$P_{31} = 1 \cdot 1 \cdot 10 = 10$$

$$P_{32} = 2 \cdot 1 \cdot 10 = 20$$

Then, one more kanban of item (3, 1) will be scheduled and at the end of the second period, the inventories will be:

$$S^2 = \{1, 1; 4; 3, 0\}$$

One kanban of items (1, 1) and (1, 2), two kanbans of items (2, 1) and (3, 2) will remain unscheduled.

$$UNSCH^2 = \{1, 1; 2; 0; 2\}$$

The backorder cost for two periods will be 600.

Period 3:

If we update the inventory levels and backorders for the third period:

$$I_{11}^3 = max(0, 1 - 2 - 0) = 0 \ , \ B_{11}^3 = max(0, 2 + 0 - 1) = 1$$

$$I_{12}^3 = max(0, 1 - 1 - 0) = 0 \ , \ B_{12}^3 = max(0, 1 + 0 - 1) = 0$$

$$I_{21}^3 = max(0, 4 - 2 - 4) = 0 \ , \ B_{21}^3 = max(0, 2 + 4 - 4) = 2$$

$$I_{31}^3 = max(0, 3 - 1 - 1) = 1 \ , \ B_{31}^3 = max(0, 1 + 1 - 3) = 0$$

$$I_{32}^3 = max(0, 0 - 2 - 0) = 0 \ , \ B_{32}^3 = max(0, 2 + 0 - 0) = 2$$

$$\text{backorder cost} = 1 \cdot 10 \cdot 1 + 0 \cdot 10 \cdot 2 + 2 \cdot 10 \cdot 10 + 0 \cdot 10 \cdot 2 + 2 \cdot 10 \cdot 3 = 270$$

The schedule set will be:

$$SS^3 = \{\mathbf{1} + 2, \mathbf{1} + 1; \mathbf{2} + 2; 1; \mathbf{2} + 2\}$$

where the bold numbers correspond to unscheduled kanbans from the previous period.

The total backorder cost over the three periods for FCFS is 870.

EXECUTION OF SPT-F:

For SPT-F, the sequence will be determined as follows:

STEP 1: Calculate the average processing time of each family:

$$P_i = \frac{\text{sequence-dependent setup time of family } i + \sum_{j=1}^{size[i]} p_{ij} \cdot D_{ij}^t \cdot a_{ij}}{size[i]}$$

STEP 2: Among the unscheduled families, select the family with minimum average processing time. In case ties exist, use FCFS-F as the tie-breaking rule.

Period 1:

For SPT-F, the average processing times for the families are as follows:

$$P_1 = \frac{5 + 1.5 \cdot 3 \cdot 10 + 1 \cdot 3 \cdot 10}{2} = 40$$

$$P_2 = \frac{5 + 4 \cdot 3 \cdot 10}{1} = 125$$

$$P_3 = \frac{5 + 1 \cdot 3 \cdot 10 + 2 \cdot 3 \cdot 10}{2} = 47.5$$

Therefore, the first family will be scheduled first. Then, the average processing times will become:

$$P_2 = \frac{40 + 4 \cdot 3 \cdot 10}{1} = 160$$

$$P_3 = \frac{15 + 1 \cdot 3 \cdot 10 + 2 \cdot 3 \cdot 10}{2} = 52.5$$

So, the third family will be scheduled second, and the total processing time will be 185, as the remaining time is 55 minutes, the second family cannot be scheduled. Therefore, all three kanbans will remain unscheduled.

So, at the end of the first period, the inventory will be:

$$S^1 = \{3, 3; 0; 3, 3\}$$

The set of unscheduled items will be:

$$UNSCH^1 = \{0, 0; 3; 0, 0\}$$

The backorder cost for the first period is 180. The stage is currently setup to family 3.

Period 2:

For period 2, we should update the inventory levels and backorders, and calculate the backorder cost of the period:

$$I_{11}^2 = max(0, 3 - 1 - 1) = 1 \ , \ B_{11}^2 = max(0, 1 + 1 - 3) = 0$$

$$I_{12}^2 = max(0, 3 - 1 - 1) = 1 \ , \ B_{12}^2 = max(0, 1 + 1 - 3) = 0$$

$$I_{21}^2 = max(0, 0 - 3 - 1) = 0 \ , \ B_{21}^2 = max(0, 3 + 1 - 0) = 4$$

$$I_{31}^2 = max(0, 3 - 3 - 1) = 0 \ , \ B_{31}^2 = max(0, 3 + 1 - 3) = 1$$

$$I_{32}^2 = max(0, 3 - 2 - 1) = 0 \ , \ B_{32}^2 = max(0, 2 + 1 - 3) = 0$$

backorder cost $= 0 \cdot 10 \cdot 1 + 0 \cdot 10 \cdot 2 + 4 \cdot 10 \cdot 10 + 1 \cdot 10 \cdot 2 + 0 \cdot 10 \cdot 3 = 420$

The schedule set will be:

$$SS^2 = \{1, 1; \mathbf{3} + 3; 3; 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period.

The rest of the steps will not be shown explicitly. In the second period family 1 will be scheduled first, family 3 will be scheduled second, and family 2 will be scheduled third. Only two kanbans of item $(2, 1)$ can be scheduled.

So, at the end of the second period, the inventory will be:

$$S^2 = \{2, 2; 2; 3, 2\}$$

The set of unscheduled items will be:

$$UNSCH^2 = \{0, 0; 4; 0, 0\}$$

The backorder cost for two periods will be 600.

Period 3:

If we update the inventory levels, backorders and slacks, and calculate the backorder cost of third period:

$$I_{11}^3 = max(0, 2 - 2 - 0) = 0 \ , \ B_{11}^3 = max(0, 2 + 0 - 2) = 0$$

$$I_{12}^3 = max(0, 2 - 1 - 0) = 1 \ , \ B_{12}^3 = max(0, 1 + 0 - 2) = 0$$

$$I_{21}^3 = max(0, 2 - 2 - 4) = 0 \ , \ B_{21}^3 = max(0, 2 + 4 - 2) = 4$$

$$I_{31}^3 = max(0, 3 - 1 - 1) = 1 \ , \ B_{31}^3 = max(0, 1 + 1 - 3) = 0$$

$$I_{32}^3 = max(0, 2 - 2 - 0) = 0 \ , \ B_{32}^3 = max(0, 2 + 0 - 2) = 0$$

$$\text{backorder cost} = 4 \cdot 10 \cdot 10 = 400$$

The schedule set will be:

$$SS^3 = \{2, 1; \mathbf{4} + 2; 1; 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period.

The total backorder cost over the three periods for SPT-F is 1000.

EXECUTION OF FCFS-F:

For FCFS-F, the sequence will be determined as follows:

STEP 1: Calculate the total backorders of each family:

$$\text{total backorder of family } i = \sum_{j=1}^{size[i]} \sum_{s=1}^{L_{ij}} s \cdot Q_{ij}[s] \cdot a_{ij}$$

where,

$Q_{ij}[s]$ is the number of kanbans for item (i,j) that is backlogged for s

periods, $s = 0, \cdots, L_{ij}$

In the Kanban systems, the in-process inventories should always be full. Even if there is no demand, there may be an order for an item which has lower in-process inventory than the maximum inventory level. For these items, there is no backorder, so their slacks are expressed as zero and $Q_{ij}[0]$ is the amount needed to fill up the in-process inventories.

STEP 2: Among the unscheduled families, select the family with maximum total backorder. Schedule all the items in the family. In case ties exist, use SPT-F as the tie-breaking rule.

Period 1:

For the first period, the L_{ij} and $Q_{ij}[s]$ values are as follows:

$$L_{11} = L_{12} = L_{21} = L_{31} = L_{32} = 1 \text{ since } D_{ij}^1 = 3 > I^0 = 2 \ \forall i, j$$

$$Q_{11}[1] = Q_{12}[1] = Q_{21}[1] = Q_{31}[1] = Q_{32}[1] = 1$$

$$Q_{11}[0] = Q_{12}[0] = Q_{21}[0] = Q_{31}[0] = Q_{32}[0] = 2$$

Therefore, the total backorders for the families are:

$$\text{total backorder of family } 1 = 1 \cdot 1 \cdot 10 + 1 \cdot 1 \cdot 10 = 20$$

$$\text{total backorder of family } 2 = 1 \cdot 1 \cdot 10 = 10$$

$$\text{total backorder of family } 3 = 1 \cdot 1 \cdot 10 + 1 \cdot 1 \cdot 10 = 20$$

As the total backorders for family 1 and 3 are equal, the tie-breaking rule, SPT-F, will be used. The average processing times of the families are as follows:

$$P_1 = \frac{5 + 1.5 \cdot 3 \cdot 10 + 1 \cdot 3 \cdot 10}{2} = 40$$

$$P_3 = \frac{5 + 1 \cdot 3 \cdot 10 + 2 \cdot 3 \cdot 10}{2} = 47.5$$

So, family 1 will be scheduled first and family 3 will be scheduled second. The total processing time will be 185 minutes. Then, the L_{ij} and $Q_{ij}[s]$ values should be updated:

$$L_{11} = L_{12} = L_{31} = L_{32} = 0$$

$$Q_{11}[0] = Q_{12}[0] = Q_{31}[0] = Q_{32}[0] = 0$$

$$L_{21} = 1$$

$$Q_{21}[1] = 1$$

$$Q_{21}[0] = 2$$

The total backorder of the family 2 will be 10 and family 2 will be scheduled next. But, in the remaining time, even one kanban of the item $(2,1)$ cannot be scheduled. Therefore, three kanbans of item $(2,1)$ will remain unscheduled.

So, at the end of the first period, the inventory will be:

$$S^1 = \{3, 3; 0; 3, 3\}$$

The set of unscheduled items will be:

$$UNSCH^1 = \{0, 0; 3; 0, 0\}$$

The stage is currently setup to family 3.

Period 2:

We should update the inventory levels, backorders and slacks, and calculate the backorder cost of the period:

$$I_{11}^2 = max(0, 3 - 1 - 1) = 1 \ , \ B_{11}^2 = max(0, 1 + 1 - 3) = 0$$

$$I_{12}^2 = max(0, 3 - 1 - 1) = 1 \ , \ B_{12}^2 = max(0, 1 + 1 - 3) = 0$$

$$I_{21}^2 = max(0, 0 - 3 - 1) = 0 \ , \ B_{21}^2 = max(0, 3 + 1 - 0) = 4$$

$$I_{31}^2 = max(0, 3 - 3 - 1) = 0 \ , \ B_{31}^2 = max(0, 3 + 1 - 3) = 1$$

$$I_{32}^2 = max(0, 3 - 2 - 1) = 0 \ , \ B_{32}^2 = max(0, 2 + 1 - 3) = 0$$

backorder cost $= 0 \cdot 10 \cdot 1 + 0 \cdot 10 \cdot 2 + 4 \cdot 10 \cdot 10 + 1 \cdot 10 \cdot 2 + 0 \cdot 10 \cdot 3 = 420$

The schedule set will be:

$$SS^2 = \{1, 1; \mathbf{3} + 3; 3; 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period.

For the second period, the L_{ij} and $Q_{ij}[s]$ values are as follows:

$$L_{11} = L_{12} = L_{32} = 0$$

$$L_{21} = 2$$

$$L_{31} = 1$$

$$Q_{11}[0] = Q_{12}[0] = 1$$

$$Q_{21}[0] = Q_{31}[0] = Q_{32}[0] = 2$$

$$Q_{21}[1] = 3$$

$$Q_{31}[1] = 1$$

$$Q_{21}[2] = 1$$

Therefore, the total backorders for the families are:

$$\text{total backorder of family } 1 = 0$$

$$\text{total backorder of family } 2 = 1 \cdot 3 \cdot 10 + 2 \cdot 1 \cdot 10 = 50$$

$$\text{total backorder of family } 3 = 1 \cdot 1 \cdot 10 = 10$$

So, family 2 will be scheduled first. There are 6 kanbans to schedule, but only 5 can be scheduled as the total processing time will be 220 minutes as the stage is currently setup to family 3. As no time remains, all the other items will stay unscheduled.

So, at the end of the second period, the inventory will be:

$$S^2 = \{1, 1; 5; 0, 0\}$$

and the backorder cost for two periods will be 600. Three kanbans of item (3,1), two kanbans of item (3,2), and one kanban of items (1,1), (1,2), and (2,1) will remain unscheduled. So:

$$UNSCH^2 = \{1,1;1;3,2\}$$

The stage is currently setup to family 2.

Period 3:

The updated inventory levels, backorders and slacks for the third period are as follows:

$$I_{11}^3 = max(0,1-2-0) = 0 \ , \ B_{11}^3 = max(0,2+0-1) = 1$$

$$I_{12}^3 = max(0,1-1-0) = 0 \ , \ B_{12}^3 = max(0,1+0-1) = 0$$

$$I_{21}^3 = max(0,5-2-4) = 0 \ , \ B_{21}^3 = max(0,2+4-5) = 1$$

$$I_{31}^3 = max(0,0-1-1) = 0 \ , \ B_{31}^3 = max(0,1+1-0) = 2$$

$$I_{32}^3 = max(0,0-2-0) = 0 \ , \ B_{32}^3 = max(0,2+0-0) = 2$$

backorder cost $= 1 \cdot 10 \cdot 1 + 0 \cdot 10 \cdot 2 + 1 \cdot 10 \cdot 10 + 2 \cdot 10 \cdot 2 + 2 \cdot 10 \cdot 3 = 210$

The schedule set will be:

$$SS^2 = \{\mathbf{1}+2, \mathbf{1}+1; \mathbf{1}+2; \mathbf{3}+1; \mathbf{2}+2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period. The total backorder cost over the three periods for FCFS-F is 810.

EXECUTION OF THE PROPOSED ALGORITHM:

The steps of the proposed algorithm are explained in detail in Chapter 3. The execution of these steps will be shown below. Unless otherwise stated, $\beta = 100$ for this example problem.

Period 1:

For the first period the schedule set will be:

$$SS^1 = \{3,3;3;3,3\}$$

and the slacks of the items will be:

$$L_{11} = L_{12} = L_{21} = L_{31} = L_{32} = 1$$

$$Q_{11}[1] = Q_{12}[1] = Q_{21}[1] = Q_{31}[1] = Q_{32}[1] = 1$$
$$Q_{11}[0] = Q_{12}[0] = Q_{21}[0] = Q_{31}[0] = Q_{32}[0] = 2$$

In the first level, we will find the makespan for the complete schedule set SS^1. The sequence according to NN will be family 2, family 3, and family 1:

$$\text{makespan} = \mathbf{5} + 4 \cdot 3 \cdot 10 + \mathbf{5} + 1 \cdot 3 \cdot 10 + 2 \cdot 3 \cdot 10 + \mathbf{10} + 1.5 \cdot 3 \cdot 10 + 1 \cdot 3 \cdot 10 = 305$$

where the bold numbers correspond to setup times and the products correspond to the total production time for the kanbans demanded.

As the makespan of SS^1 is longer than T which is 240 units, it is not possible to produce all the items in this set. We should proceed to Level 2 of the proposed algorithm.

In the second level, we will find a subset of the schedule set, PS^1. To find the production set, first the updated demand for the next period for each item should be calculated. As we have only one period, the updated demand will be equal to the period demand. So updated demands of the items will be 3. As the inventories are empty, the production amounts will be equal to the updated demand. Therefore,

$$PS^1 = \{3, 3; 3; 3, 3\}$$

and it will not be possible to schedule all of the items. Then we should proceed to Level 3.

In the third level, an index for each item is calculated and a new schedule set IS^1 will be formed. The index will be calculated as follows:

$$\begin{aligned}\alpha_{ij} &= \frac{\text{sequence-dependent setup time of family } i}{\text{total backorder cost of the family } i} \\ &+ \frac{\text{processing time of the item } (i,j)}{(1 + L_{ij}) \cdot \text{backorder cost of item } (i,j)}\end{aligned}$$

where

$$\text{total backorder cost of the family } i = \sum_{j=1}^{size[i]} \sum_{s=0}^{L_{ij}} (s+1) \cdot Q_{ij}[s] \cdot b_{ij}$$

Let f_i be a binary variable that indicates if any of the items from family i has already been scheduled or not, i.e.,

$$f_i = \begin{cases} 1 & , \text{ if at least one item from family } i \text{ is scheduled} \\ 0 & , \text{ otherwise} \end{cases}$$

Initially, $f_i = 0 \ \forall i$.

So,

$$\text{total backorder cost of the family } 1 = 120$$

$$\text{total backorder cost of the family } 2 = 400$$

$$\text{total backorder cost of the family } 3 = 200$$

and, the indexes will be:

$$\alpha_{11} = \frac{5}{120} + \frac{1.5}{2} = 0.79$$

$$\alpha_{12} = \frac{5}{120} + \frac{1}{4} = 0.29$$

$$\alpha_{21} = \frac{5}{400} + \frac{4}{20} = 0.21$$

$$\alpha_{31} = \frac{5}{200} + \frac{1}{4} = 0.28$$

$$\alpha_{32} = \frac{5}{200} + \frac{2}{6} = 0.36$$

Item $(2,1)$ will be scheduled first and $f_2 = 1$. The total processing time will be 45 minutes. The total backorder cost of the family 2 will change and the other costs will remain unchanged.

$$\text{total backorder cost of the family } 2 = 200$$

We schedule one kanban at a time. The reason for this is to assign dynamic backorder costs to items. There can be kanbans with different slack values for the same item, so these kanbans will have different backorder costs. For example, the backorder cost of item (2, 1) is 20 in the first index calculation. When we recalculate the index, as we have already scheduled the kanban of item (2, 1) that is one period late, we do not have any backorders for this item. If we go on scheduling item (2, 1), this will be to avoid the future backorders that may occur due to empty in-process inventories. So the item (2, 1) will have less priority in the next calculations of the index. The indexes will be:

$$\alpha_{11} = \frac{5}{120} + \frac{1.5}{2} = 0.79$$

$$\alpha_{12} = \frac{5}{120} + \frac{1}{4} = 0.29$$

$$\alpha_{21} = 0 + \frac{4}{10} = 0.40$$

$$\alpha_{31} = \frac{5}{200} + \frac{1}{4} = 0.28$$

$$\alpha_{32} = \frac{5}{200} + \frac{2}{6} = 0.36$$

Item (3, 1) will be selected and $f_3 = 1$. The total processing time will be 60 minutes. The total backorder cost for family 3 will be 160. As $f_2 = f_3 = 1$, no setup time will be incurred for the items in family 2 and 3. Therefore, the updated indexes will be:

$$\alpha_{11} = \frac{10}{120} + \frac{1.5}{2} = 0.83$$

$$\alpha_{12} = \frac{10}{120} + \frac{1}{4} = 0.33$$

$$\alpha_{21} = 0 + \frac{4}{10} = 0.4$$

$$\alpha_{31} = 0 + \frac{1}{2} = 0.5$$

$$\alpha_{32} = 0 + \frac{2}{6} = 0.33$$

Items (1, 2) and (3, 2) have the same index. One of them will be selected. Select the item with minimum setup time. Therefore, item (3,2) will be selected. The total processing time will be 80 minutes and the total backorder cost for family 3 will be 100. The indexes will be:

$$\alpha_{11} = \frac{10}{120} + \frac{1.5}{2} = 0.83$$

$$\alpha_{12} = \frac{10}{120} + \frac{1}{4} = 0.33$$
$$\alpha_{21} = 0 + \frac{4}{10} = 0.4$$
$$\alpha_{31} = 0 + \frac{1}{2} = 0.5$$
$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

Item $(1,2)$ will be selected and $f_1 = 1$. So the total backorder cost of family 1 will be 80, and the total processing time will be 100 minutes. The updated indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{2} = 0.75$$
$$\alpha_{12} = 0 + \frac{1}{2} = 0.5$$
$$\alpha_{21} = 0 + \frac{4}{10} = 0.4$$
$$\alpha_{31} = 0 + \frac{1}{2} = 0.5$$
$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

Item $(2,1)$ will be selected again and as the indexes will not change, this will continue till all the kanbans of item $(2,1)$ are completed. So the total processing time will become 180 minutes and the total backorder cost of the second family will be 0. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{2} = 0.75$$
$$\alpha_{12} = 0 + \frac{1}{2} = 0.5$$
$$\alpha_{31} = 0 + \frac{1}{2} = 0.5$$
$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

The indexes for item $(1,2)$ and $(3,1)$ are the same. As their setup requirements are also the same, one of them will be selected randomly, say item $(1,2)$. As the indexes will not change, all the kanbans of item $(1,2)$ will be scheduled.

The total processing time will be 200 and the total backorder cost of family 1 will be 40. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{2} = 0.75$$

$$\alpha_{31} = 0 + \frac{1}{2} = 0.5$$

$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

Item (3,1) will be selected and as the indexes will not change all the kanbans of item (3, 1) will be scheduled. The total processing time will be 220 minutes and the total backorder cost of family 3 will be 60. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{2} = 0.75$$

$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

So, item (3, 2) will be selected. One kanban of item (3, 2) can be scheduled as the total time will be 240 minutes. As no time remains, we should stop calculating indexes. The index set will be:

$$IS^1 = \{0, 3; 3; 3, 2\}$$

The sequence will be family 2, family 3 and family 1. This sequence is the same as the sequence found by NN. Therefore, the makespan will be 240 minutes. The inventories at the end of first period will be:

$$S^1 = \{0, 3; 3; 3, 2\}$$

The set of unscheduled items will be:

$$UNSCH^1 = \{3, 0; 0; 0, 1\}$$

The stage is currently setup to family 1.

The backorder cost for the first period is 180.

Period 2:

For the second period, first we should update the inventory levels and backorders:

$$I_{11}^2 = max(0, 0 - 1 - 1) = 0 \ , \ B_{11}^2 = max(0, 1 + 1 - 0) = 2$$

$$I_{12}^2 = max(0, 3 - 1 - 1) = 1 \ , \ B_{12}^2 = max(0, 1 + 1 - 3) = 0$$

$$I_{21}^2 = max(0, 3 - 3 - 1) = 0 \ , \ B_{21}^2 = max(0, 3 + 1 - 3) = 1$$

$$I_{31}^2 = max(0, 3 - 3 - 1) = 0 \ , \ B_{31}^2 = max(0, 3 + 1 - 3) = 1$$

$$I_{32}^2 = max(0, 2 - 2 - 1) = 0 \ , \ B_{32}^2 = max(0, 2 + 1 - 2) = 1$$

$$\text{backorder cost} = 2 \cdot 10 \cdot 1 + 0 \cdot 10 \cdot 2 + 1 \cdot 10 \cdot 10 + 1 \cdot 10 \cdot 2 + 1 \cdot 10 \cdot 3 = 170$$

The schedule set will be:

$$SS^2 = \{\mathbf{3} + 1, 1; 3; 3; \mathbf{1} + 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period.

The slacks will be:

$$L_{12} = 0$$

$$L_{21} = L_{31} = L_{32} = 1$$

$$L_{11} = 2$$

$$Q_{11}[2] = 1$$

$$Q_{11}[1] = Q_{21}[1] = Q_{31}[1] = Q_{32}[1] = 1$$

$$Q_{11}[0] = Q_{21}[0] = Q_{31}[0] = Q_{32}[0] = 2$$

$$Q_{12}[0] = 1$$

Find the makespan for the complete schedule set SS^2. The sequence according to NN will be family 1, family 3, and family 2:

$$\text{makespan} = 1.5 \cdot 4 \cdot 10 + 1 \cdot 1 \cdot 10 + \mathbf{15} + 1 \cdot 3 \cdot 10 + 2 \cdot 3 \cdot 10 + \mathbf{20} + 4 \cdot 3 \cdot 10 = 315$$

where the bold numbers correspond to setup times and the products correspond to the total production time for the kanbans demanded.

As the makespan of SS^2 is longer than T, it is not possible to produce all the items in this set. We should proceed to Level 2 of the proposed algorithm.

The updated demands will be:

$$d^u_{11} = \lceil 0.5 \cdot 3 + 0.5 \cdot 1 \rceil = 2$$

$$d^u_{12} = \lceil 0.5 \cdot 3 + 0.5 \cdot 1 \rceil = 2$$

$$d^u_{21} = \lceil 0.5 \cdot 3 + 0.5 \cdot 3 \rceil = 3$$

$$d^u_{31} = \lceil 0.5 \cdot 3 + 0.5 \cdot 3 \rceil = 3$$

$$d^u_{32} = \lceil 0.5 \cdot 3 + 0.5 \cdot 2 \rceil = 3$$

As the inventories are empty for items other than $(1,2)$, the production amounts will be equal to the updated demands for these items. For item $(1,2)$, the production amount will be 1, i.e. the difference between updated demand and in-process inventory. So:

$$PS^2 = \{2,1;3;3,3\}$$

The makespan of this set is:

$$\text{makespan} = 1.5 \cdot 2 \cdot 10 + 1 \cdot 1 \cdot 10 + \mathbf{15} + 1 \cdot 3 \cdot 10 + 2 \cdot 3 \cdot 10 + \mathbf{20} + 4 \cdot 3 \cdot 10 = 285$$

So, it will not be possible to produce all of the items in the set. The index will be calculated for each item:

$$\text{total backorder cost of the family } 1 = 90$$

$$\text{total backorder cost of the family } 2 = 400$$

$$\text{total backorder cost of the family } 3 = 200$$

and, as the stage is currently set to family 1, $f_1 = 1$ and the indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$

$$\alpha_{12} = 0 + \frac{1}{2} = 0.50$$

$$\alpha_{21} = \frac{40}{400} + \frac{4}{20} = 0.30$$

$$\alpha_{31} = \frac{15}{200} + \frac{1}{4} = 0.32$$
$$\alpha_{32} = \frac{15}{200} + \frac{2}{6} = 0.40$$

Item $(2,1)$ will be scheduled first and $f_2 = 1$. The total processing time will be 80 minutes. The total backorder cost of the family 2 will be 200, and the indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$
$$\alpha_{12} = 0 + \frac{1}{2} = 0.50$$
$$\alpha_{21} = 0 + \frac{4}{10} = 0.40$$
$$\alpha_{31} = \frac{5}{200} + \frac{1}{4} = 0.28$$
$$\alpha_{32} = \frac{5}{200} + \frac{2}{6} = 0.36$$

So, item $(3,1)$ will be selected, $f_3 = 1$. The total processing time will be 95 minutes. The total backorder cost of the family 3 will be 160, and the indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$
$$\alpha_{12} = 0 + \frac{1}{2} = 0.50$$
$$\alpha_{21} = 0 + \frac{4}{10} = 0.40$$
$$\alpha_{31} = 0 + \frac{1}{2} = 0.50$$
$$\alpha_{32} = 0 + \frac{2}{6} = 0.33$$

Item $(3,2)$ will be scheduled third. The total processing time will be 115 minutes. And, the total backorder cost of family 3 will be 100. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$
$$\alpha_{12} = 0 + \frac{1}{2} = 0.50$$
$$\alpha_{21} = 0 + \frac{4}{10} = 0.40$$

$$\alpha_{31} = 0 + \frac{1}{2} = 0.50$$

$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

$(2,1)$ will be selected next. The total processing time will be 155 minutes and the total backorder cost of family 2 will be 100. As the indexes will not change, item $(2,1)$ will be selected again. The total processing time will be 195 minutes and the total backorder cost of family 2 will be zero. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$

$$\alpha_{12} = 0 + \frac{1}{2} = 0.50$$

$$\alpha_{31} = 0 + \frac{1}{2} = 0.50$$

$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

As the indexes and the setup requirements of the items $(1,1)$, $(1,2)$ and $(3,1)$ are the same, one of them will be selected randomly. Select item $(3,1)$. The total processing time will be 205 minutes and the total backorder cost of family 3 will be 80. As the indexes will not change, item $(3,1)$ will be selected again. The total processing time will be 215 minutes and the total backorder cost of family 3 will be 60. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$

$$\alpha_{12} = 0 + \frac{1}{2} = 0.50$$

$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

The indexes for item $(1,1)$ and $(1,2)$ are the same. So we will select one of them randomly, select item $(1,2)$. The total processing time will be 225, and the total backorder cost for family 1 will be 70. As in PS^2 there is only one kanban of item $(1,2)$, we cannot schedule item $(1,2)$ any more. The indexes will be:

$$\alpha_{11} = 0 + \frac{1.5}{3} = 0.50$$

$$\alpha_{32} = 0 + \frac{2}{3} = 0.66$$

So, item $(1, 1)$ will be selected. The total processing time will be 240 minutes. As no time remains, we should stop calculating the indexes. The index set will be:

$$IS^2 = \{1, 1; 3; 3, 1\}$$

The makespan of this set by using NN will be:

$$\text{makespan} = 1.5 \cdot 1 \cdot 10 + 1 \cdot 1 \cdot 10 + \mathbf{15} + 1 \cdot 3 \cdot 10 + 2 \cdot 1 \cdot 10 + \mathbf{20} + 4 \cdot 3 \cdot 10 = 230$$

To fill the remaining 10 minutes, knapsack problems will be solved. First the updated processing times should be calculated.

If there is at least one item of family i in IS^t:

$$p^u_{ij} = p_{ij} \cdot a_{ij}$$

otherwise:

$$p^u_{ij} = \text{sequence-dependent setup time of family } i + p_{ij} \cdot a_{ij}$$

As at least one item is scheduled from each family, we will use the first equation to calculate the updated processing times:

$$p^u_{11} = 1.5 \cdot 10 = 15$$

$$p^u_{12} = 1 \cdot 10 = 10$$

$$p^u_{21} = 4 \cdot 10 = 40$$

$$p^u_{31} = 1 \cdot 10 = 10$$

$$p^u_{32} = 2 \cdot 10 = 20$$

As the remaining time is 10, only item $(1, 2)$ and $(3, 1)$ are feasible. We should check the remaining set only for these items. Let RM_{ij} be the amount of item (i, j) in the set $PS^t - IS^t$. Then:

$$RM_{12} = 1 - 1 = 0$$

$$RM_{31} = 3 - 3 = 0$$

As the remaining set is empty for both of these items, there is no need to solve a knapsack problem. We cannot include any item from the set PS^t. As we have enough remaining time, we now consider the set SS^t. The updated processing times will not change. Then, the set of remaining items will be:

$$RM_{12} = 1 - 1 = 0$$

$$RM_{31} = 3 - 3 = 0$$

So, no items can be included to the index set. The final index set will be:

$$IS^2 = \{1, 1; 3; 3, 1\}$$

The inventories will be:

$$S^2 = \{1, 2; 3; 3, 1\}$$

And, the set of unscheduled items will be:

$$UNSCH^2 = \{3, 0; 0; 0, 2\}$$

The total backorder cost for two periods will be 350.

We also schedule the second period with $\beta = 50$ in order to demonstrate how the β parameter might affect the final schedule as well as its total cost. We will use the index to form the index set, IS^2, till the total processing time is less than or equal to $\frac{\beta \times 240}{100} = 120$. So, the previous index calculations will be valid till total processing time reaches 120. Therefore, we will begin to use knapsack after the third calculation of index. At that time the index set was:

$$IS^2 = \{0, 0; 1; 1, 1\}$$

And, the total processing time was 115. The makespan of this set by using NN was:

$$\text{makespan} = \mathbf{15} + 1 \cdot 1 \cdot 10 + 2 \cdot 1 \cdot 10 + \mathbf{20} + 4 \cdot 1 \cdot 10 = 105$$

For the remaining time, we will solve knapsack problems. The updated processing times will be:

$$p_{11}^u = 5 + 1.5 \cdot 10 = 20$$

$$p^u_{12} = 5 + 1 \cdot 10 = 15$$

$$p^u_{21} = 4 \cdot 10 = 40$$

$$p^u_{31} = 1 \cdot 10 = 10$$

$$p^u_{32} = 2 \cdot 10 = 20$$

Then, the probabilistic weights should be calculated. For this example, lets assume that probability mass function of the demand distribution is given as:

$$f_D(d) = \begin{cases} 0.3 & , \quad D^t_{ij} = 1 \\ 0.4 & , \quad D^t_{ij} = 2 \\ 0.3 & , \quad D^t_{ij} = 3 \\ 0 & , \quad \text{otherwise} \end{cases}$$

Then, the probabilistic weight will be

$$w_{ij} = P(D^{t+1}_{ij} \geq IS^t_{ij} + I^t_{ij}) \cdot b_{ij} \cdot a_{ij}$$

Then,

$$w_{11} = 1 \cdot 10 = 10$$

$$w_{12} = 0.7 \cdot 20 = 14$$

$$w_{21} = 0.7 \cdot 100 = 70$$

$$w_{31} = 0.7 \cdot 20 = 14$$

$$w_{32} = 0.7 \cdot 30 = 21$$

The set of remaining items will be:

$$RM_{11} = 2 - 0 = 2$$

$$RM_{12} = 1 - 0 = 1$$

$$RM_{21} = 3 - 1 = 2$$

$$RM_{31} = 3 - 1 = 2$$

$$RM_{32} = 3 - 1 = 2$$

So, the knapsack formulation will be:

$$\text{MAX } 10X_{11} + 14X_{12} + 70X_{21} + 14X_{31} + 21X_{32}$$

st

$$20X_{11} + 15X_{12} + 40X_{21} + 10X_{31} + 20X_{32} \leq 135$$

$$X_{11} \leq 2$$

$$X_{12} \leq 1$$

$$X_{21} \leq 2$$

$$X_{31} \leq 2$$

$$X_{32} \leq 2$$

The solution will be $X_{11} = 0$, $X_{21} = X_{31} = 2$ and $X_{12} = X_{32} = 1$. As item $(1,2)$, which requires a new setup, is selected, we should include only this item into the index set. Now, each family will have at least one item in the schedule set. Therefore, we should update the processing times and solve a new knapsack problem. So, the index set will be:

$$IS^2 = \{0, 1; 1; 1, 1\}$$

The updated processing times will be:

$$p_{11}^u = 1.5 \cdot 10 = 15$$

$$p_{12}^u = 1 \cdot 10 = 10$$

$$p_{21}^u = 4 \cdot 10 = 40$$

$$p_{31}^u = 1 \cdot 10 = 10$$

$$p_{32}^u = 2 \cdot 10 = 20$$

The makespan found by NN will be:

$$\text{makespan} = 10 + \mathbf{15} + 1 \cdot 3 \cdot 10 + 2 \cdot 2 \cdot 10 + \mathbf{20} + 4 \cdot 3 \cdot 10 = 115$$

The probabilistic weights will be:

$$w_{11} = 1 \cdot 10 = 10$$

$$w_{12} = 0.3 \cdot 20 = 6$$

$$w_{21} = 0.7 \cdot 100 = 70$$

$$w_{31} = 0.7 \cdot 20 = 14$$

$$w_{32} = 0.7 \cdot 30 = 21$$

And, the set of remaining items will be:

$$RM_{11} = 2 - 0 = 2$$

$$RM_{12} = 1 - 1 = 0$$

$$RM_{21} = 3 - 1 = 2$$

$$RM_{31} = 3 - 1 = 2$$

$$RM_{32} = 3 - 1 = 2$$

So, the knapsack formulation will be:

$$\text{MAX } 10X_{11} + 70X_{21} + 14X_{31} + 21X_{32}$$

st

$$15X_{11} + 40X_{21} + 10X_{31} + 20X_{32} \leq 125$$

$$X_{11} \leq 2$$

$$X_{21} \leq 2$$

$$X_{31} \leq 2$$

$$X_{32} \leq 2$$

The solution will be $X_{11} = 0$, $X_{21} = X_{31} = 2$ and $X_{32} = 1$. So, the index set will become:

$$IS^2 = \{0, 1; 3; 3, 2\}$$

The remaining time is 5 minutes and this value is smaller than the minimum processing time. Therefore, we will not solve knapsack problems for the set $SS^2 - IS^2$.

So, when $\beta = 50$, the index set will be:

$$IS^2 = \{0, 1; 3; 3, 2\}$$

The inventories will be:

$$S^2 = \{0, 2; 3; 3, 2\}$$

And, the set of unscheduled items will be:

$$UNSCH^2 = \{4, 0; 0; 0, 1\}$$

The total backorder cost for two periods will be 350.

Period 3:

If we update the inventory levels, backorders and slacks, and calculate the backorder cost of third period:

If $\beta = 100$ for second period:

$$I_{11}^3 = max(0, 1 - 2 - 2) = 0 \ , \ B_{11}^3 = max(0, 2 + 2 - 1) = 3$$

$$I_{12}^3 = max(0, 2 - 1 - 0) = 1 \ , \ B_{12}^3 = max(0, 1 + 0 - 2) = 0$$

$$I_{21}^3 = max(0, 3 - 2 - 1) = 0 \ , \ B_{21}^3 = max(0, 2 + 1 - 3) = 0$$

$$I_{31}^3 = max(0, 3 - 1 - 2) = 0 \ , \ B_{31}^3 = max(0, 1 + 2 - 3) = 0$$

$$I_{32}^3 = max(0, 1 - 2 - 2) = 0 \ , \ B_{32}^3 = max(0, 2 + 2 - 1) = 3$$

$$\text{backorder cost} = 3 \cdot 10 \cdot 1 + 3 \cdot 10 \cdot 3 = 120$$

The schedule set will be:

$$SS^3 = \{\mathbf{3} + 2, 1; 2; 1, \mathbf{2} + 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period. The total backorder cost over three periods is 470.

If $\beta = 50$ for second period, the updated the inventory levels, backorders, and backorder cost of third period will be:

$$I_{11}^3 = max(0, 0 - 2 - 2) = 0 \ , \ B_{11}^3 = max(0, 2 + 2 - 0) = 4$$

$$I^3_{12} = max(0, 2-1-0) = 1 \ , \ B^3_{12} = max(0, 1+0-2) = 0$$

$$I^3_{21} = max(0, 3-2-1) = 0 \ , \ B^3_{21} = max(0, 2+1-3) = 0$$

$$I^3_{31} = max(0, 3-1-2) = 0 \ , \ B^3_{31} = max(0, 1+2-3) = 0$$

$$I^3_{32} = max(0, 2-2-2) = 0 \ , \ B^3_{32} = max(0, 2+2-2) = 2$$

$$\text{backorder cost} = 4 \cdot 10 \cdot 1 + 2 \cdot 10 \cdot 3 = 100$$

The schedule set will be:

$$SS^3 = \{\mathbf{4} + 2, 1; 2; 1, \mathbf{1} + 2\}$$

where the bold number corresponds to unscheduled kanbans from the previous period. The total backorder cost over three periods is 450.

5.1 Summary

In this chapter, the detailed executions of the algorithms over a simple example problem are presented. In the example, only the execution of the scheduling modules are given. It is assumed that the withdrawal cycle lengths, kanban sizes and the number of kanbans are known. As no decisions are made, the inventory holding costs are treated as sunk costs (as the withdrawal cycle lengths and kanban sizes are the same for all the algorithms, the maximum inventory levels should also be the same). The total backorder costs over two periods are 870, 1000, 810, and 1000 for FCFS, SPT, FCFS-F, SPT-F, respectively. For the proposed algorithm, when $\beta = 100$, the backorder cost is 470, and when $\beta = 50$ in second period, the backorder cost is 450. As can be seen from the results, the proposed algorithm finds the minimum total backorder costs. This simple example highlights the effectiveness of the proposed algorithm and the proposed index. As expected, when β decreases, the total backorder costs also decrease.

Chapter 6

Conclusion

In this chapter, a brief summary of the findings of this study is presented and the possible extensions for future research are stated. In this study, we analytically studied the Kanban systems in a multi-item, multi-stage, multi-period modified flowline production setting. We proposed an algorithm that provides a feedback mechanism to evaluate the impact of operating parameters to determine the design parameters of a Kanban system such as withdrawal cycle length, kanban size and number of kanbans. Some commonly used algorithms from the kanban literature were also investigated under different experimental settings. Furthermore, we used two family-based scheduling rules that we adapted from GT scheduling literature and examined their performance under kanban setting. The results are summarized in the following section.

6.1 Results and Contribution

To our knowledge, this study is the first analytical study that uses operating characteristics of a Kanban system to determine the design parameters. Most of the existing models determine the number of kanbans independent of the kanban sizes and scheduling decisions. We determined the number of kanbans and kanban sizes simultaneously by using the scheduling decisions.

Very few researchers have studied the effects of material-handling frequency on the system performance. In fact, the withdrawal cycle length have a significant effect on average in-process inventories and production rates. We showed that as the withdrawal cycle lengths decrease, the in-process inventories decrease significantly, but the backorders increase. In fact, when the kanban sizes are constant, the total cost curve is convex over withdrawal cycle lengths.

Even though there are analytical and simulation models on the effect of kanban sizes on the system performance, the findings are not so clear and are limited to the in-process inventories. We considered several performance measures and kanban sizes to show the effect of kanban size on the system performance.

The findings of the study can be summarized as follows:

- When the variety of the products increase, the performance of the system declines. In fact, this result is quite obvious as one of the main assumptions of Kanban system is repetitive manufacturing. When the product variety increases, the standardization in product design decreases, and the repetitive nature of the system disappears.

- As long as transportation is not constrained, the smaller the kanban sizes, the better the system performance. Decreasing the kanban sizes not only decreases the remnants but also brings the advantages of production and transportation lot size. The customer service increases.

- The shorter the withdrawal cycle lengths, the smaller the in-process inventories. On the contrary, the shorter the withdrawal cycle length, the larger the backorders. Therefore, when the kanban sizes are the same, the cost curve is convex and the decision is due to the relation between backorder and inventory holding costs.

- When the system is balanced, the system performance is better. When we introduced imbalance to the system, the minimum inventory levels increased and in spite of the high inventory level, the system performance got worse. Allocating different number of kanbans at each stage and using a proper scheduling rule can help to cope with the imbalance in the system.

- The lower the setup times, the better the system performance. As the setup times increase, the JIT characteristic of the system disappears.

Therefore, Kanban system performs best when the setup times are low, the congestion is low, the product variety is low and there are no imbalances.

If we compare the item-based algorithms with the family-based algorithms, we see that:

- On the average, the item-based scheduling rules have longer withdrawal cycle lengths than the other algorithms. As the average setup times are high for item-based rules, the system cannot respond when the withdrawal cycle length falls below a level. Therefore, if the setup times are considerable family-based rules are better than the item-based rules.
- When the system load is loose, the item-based rules perform better than the family-based rules in terms of backorder cost and fill rates. As the system load increases, the vice versa is true.
- The family-based rules are more robust to setup time increases.
- The average fill rates of the item-based algorithms are better than their family-based versions.

If we compare the item-based rules with the proposed algorithm, we can say that the proposed algorithm performs better than the item-based rules in terms of withdrawal cycle length values, backorder cost, setup times and fill rates. Even when the system load is loose, the performance of the proposed algorithm is better than the performances of the item-based approaches. When we compare the proposed algorithm with the family-based rules, in terms of backorder costs, setup times and fill rates, the proposed algorithm performs better. The maximum inventory level is lower for the family-based rules but there is a high variability in the inventory levels hence the fill rates are low. Therefore, family-based rules hold less in-process inventory but create nervousness in the system.

In fact, the performances of FCFS-F and SPT-F are quite similar because of the periodic review system. And, as the processing times of the items are quite alike, they both perform quite well. To be able to compare the family-based approaches and the proposed method further studies should be performed under more variable settings.

6.2 Future Research Directions

There are several future research directions for this study:

- In this study, the processing times come from the same distribution for all items and all stages. In fact, some of the rules, especially SPT and SPT-F, can perform poorly under variable processing times. The study can be enlarged to include variable processing times.
- As the proposed algorithm is for system design, the periods are strict, i.e. if a kanban cannot be processed in the remaining time, preemption between periods is not allowed, and the remaining time stays idle. In case the algorithm is used for operational purposes, this assumption can be relaxed. Instead of keeping the remaining time idle, a kanban can be preempted between periods. The algorithm can be updated to include this preemption between periods.
- In the proposed algorithm, NN heuristic is used to order the items. In fact, the performance of the NN heuristic declines when the setup matrix become variable. Therefore, NN can be replaced with a better rule.
- A very simple forecasting method is used in the proposed algorithm. An advanced technique can be used to increase the predicting power of the

algorithm.

- In the lead time estimation module, we selected a due-date estimation rule from the literature, and updated it for the periodic review models. In fact, a better due-date rule can be developed for the periodic systems.
- And finally, the most promising research direction is to add transportation lead times to the existing model. In that way, a trade-off will appear among different kanban sizes and withdrawal cycle lengths.

Bibliography

[1] Aktürk, M.S., and Dağhoğlugil, 1995, Dynamic Lot-Sizing and Scheduling in Cellular Manufacturing Systems Using Hybrid Simulation/Analytical Model, *International Journal of Industrial Engineering,* **2**, 4, 251-260.

[2] Askin, R.G., Mitwasi, M.G., and Goldberg, J.B., 1993, Determining the Number of Kanbans in Multi-Item Just-in-Time Systems, *IIE Transactions,* **25**, 1, 89-97.

[3] Baker, K.R., 1994, *Elements of Sequencing and Scheduling,* Manuscript.

[4] Bard, J.F., and Golony, B., 1991, Determining the Number of Kanbans in a Multiproduct, Multistage Production System, *International Journal of Production Research,* **29**, 5, 881-895.

[5] Berkley, B.J., 1992, A Review of the Kanban Production Control Research Literature, *Production and Operations Management,* **1**, 393-411.

[6] Berkley, B.J., 1993, Simulation Tests of FCFS and SPT Sequencing in Kanban Systems, *Decision Sciences,* **24**, 1, 218-227.

[7] Berkley, B.J., 1996, A Simulation Study of Container Size in Two-Card Kanban System, *International Journal of Production Research,* **34**, 12, 3417-3445.

[8] Berkley, B.J., and Kıran, S., 1991, A Simulation Study of Sequencing Rules in a Kanban-Controlled Flow Shop, *Decision Sciences,* **22**, 3, 559-582.

[9] Bitran, G.R., and Chang, L., 1987, A Mathematical Programming Approach To a Deterministic Kanban System, *Management Science,* **33**, 4, 427-441.

[10] Browne, J., Harhen, J., and Shivnan, J., 1996, *Production Management Systems, An Integrated Perspective, Second Edition,* Addison-Wesley.

[11] Groenevelt, H., 1993, 'The Just-in-Time System', *Handbooks in OR & MS,* Vol.4, edited by S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, Elsevier Science Publishers.

[12] Gunasekaran, A., Goyal, S.K., Virtanen, I., and Yli-Olli, P., 1994, An Investigation into the Application of Group Technology in Advanced Manufacturing Systems, *International Journal of Computer Integrated Manufacturing,* **7**, 4, 215-228.

[13] Gupta, Y.P., and Gupta, M.C., 1989, A System Dynamics Model for a Multi-Stage, Multi-Line Dual-Card JIT-Kanban System, *International Journal of Production Research,* **27**, 2, 309-352.

[14] Huang, C.C., and Kusiak, A., 1996, Overview of Kanban systems, *International Journal of Computer Integrated Manufacturing,* **9**, 3, 169-189.

[15] Huang, P.Y., Rees, P.L., and Taylor III, B.W., 1983, Simulation Analysis of the Japanese Just-in-Time Technique (with Kanbans) for a Multiline, Multistage Production System, *Decision Sciences,* **14**, 3, 326-344.

[16] Karmarkar, U., 1987, Lot Sizes, Lead Times, and In-Process Inventories, *Management Science,* **33**, 3, 409-418.

[17] Karmarkar, U., 1993, 'Manufacturing Lead Times, Order Release and Capacity Loading', *Handbooks in OR & MS,* Vol.4, edited by S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, Elsevier Science Publishers.

[18] Karmarkar, U., and Kekre, S., 1989, Batching Policy in Kanban Systems, *Journal of Manufacturing Systems,* **8**, 4, 317-328.

[19] Karmarkar, U., Kekre, S., Kekre, S., and Freeman, S., 1985, Lot-Sizing and Lead-Time Performance in a Manufacturing Cell, *Interfaces,* **15**, 1-9.

[20] Kırkavak, N., 1995, Modelling and Analysis Analysis of Pull Production Systems, Ph.D. Dissertation, Bilkent University, Ankara, Turkey.

[21] Kimura, O., and Terada, H., 1981, Design and Analysis of Pull System: A Method of Multi-Stage Production Control, *International Journal of Production Research,* **19**, 241-253.

[22] Krajewski, L.J., King, B.E., Ritzman, L.P., and Wong, D.S., 1987, Kanban, MRP, and Shaping the Manufacturing Environment, *Management Science,* **33**, 1, 39-57.

[23] Lee, L.C., 1987, Parametric Appraisal of the JIT System, *International Journal of Production Research,* **25**, 1415-1429.

[24] Lee, L.C., and Seah, K.H.W., 1988, JIT and the Effects of varying Process and Set-up Times *International Journal of Operations and Production Management,* **6**, 3, 186-190.

[25] Li, A., and Co, H.C., 1991, A Dynamic Programming Model for The Kanban Assignment Problem In a Multistage Multiperiod Production System, *International Journal of Production Research,* **29**, 1, 1-16.

[26] Lummus, R.R., 1995, A Simulation Analysis of Sequencing Alternatives for JIT Lines using Kanbans, *Journal of Operations Management,* **13**, 183-191.

[27] Mahmoodi, F., Dooley, K.J., and Starr, P.J., 1990, An Evaluation of Order Releasing and Due Date Assignment Heuristics in a Cellular Manufacturing System, *Journal of Operations Management,* **9**, 4, 548-573.

[28] Mitra, D., and Mitrani, I., 1990, Analysis of a Kanban Discipline For Cell Coordination in Production Lines. I, *Management Science,* **36**, 12, 1548-1566.

[29] Mitra, D., and Mitrani, I., 1991, Analysis of a Kanban Discipline For Cell Coordination in Production Lines. II: Stochastic Demands, *Operations Research,* **39**, 5, 807-823.

[30] Mittal, S., and Wang, H.P., 1992, Simulation of JIT Production to Determine Number of Kanbans, *International Journal of Advanced Manufacturing Technology,* **7**, 292-305.

[31] Mitwasi, M.G., and Askin, R.G., 1994, Production Planning for a Multi-item, Multi-stage Kanban System, *International Journal of Production Research,* **32**, 5, 1173-1195.

[32] Miyazaki, S., Ohta, H., and Nishiyama, N., 1988, The Optimal Operation Planning of Kanban to Minimize the Total Operation Cost, *International Journal of Production Research,* **26**, 10, 1605-1611.

[33] Monden, Y., 1981, *Toyota Production System,* Industrial Engineering and Management Press, Norcross.

[34] Monden, Y., 1984, A Simulation Analysis of the Japanese Just-in-Time Technique (with Kanbans) for a Multiline, Multistage Production System: A Comment, *Decision Sciences,* **15**, 3, 445-447.

[35] Muckstadt, J.A., and Tayur, S.R., 1995, A Comparison of Alternative Kanban Control Mechanisms. I. Background and Structural Results, *IIE Transactions,* **27**, 2, 140-150.

[36] Muckstadt, J.A., and Tayur, S.R., 1995, A Comparison of Alternative Kanban Control Mechanisms. II. Experimental Results, *IIE Transactions,* **27**, 2, 151-161.

[37] Offodile, F.O., Mehrez, A., and Grzner, J., 1994, Cellular Manufacturing: A Taxonomic Review Framework, *Journal of Manufacturing Systems,* **13**, 3, 196-220.

[38] Oğuz, C., 1988, Design and Analysis of Just-In-Time Production Systems, Master Thesis, Bilkent University, Ankara, Turkey.

[39] Ohno, K., Nakashima, K., and Kojima, M., 1995, Optimal Numbers of Two Kinds of Kanbans in a JIT Production System, *International Journal of Production Research,* **33**, 5, 1387-1401.

[40] Olhager, J., 1995, Technical note: Safety mechanisms in just-in-time systems, *International Journal of Operations and Production Management,* **15**, 9, 289-292.

[41] Philipoom, P.R., Rees, L.P., Taylor III, B.W., 1996, Simultaneously Determining the Number of Kanbans, Container Sizes, and the Final-Assembly Sequence of Products in a Just-in-Time Shop, *International Journal of Production Research,* **34**, 1, 51-69.

[42] Philipoom, P.R., Rees, L.P., Taylor III, B.W., and Huang, P.Y., 1990, A Mathematical Programming Approach for Determining Workcentre Lotsizes in a Just-in-Time System with Signal Kanbans, *International Journal of Production Research,* **28**, 1, 1-15.

[43] Ragatz, G.L., and Mabert, V.A., 1984, A Simulation Analysis of Due Date Assignment Rules, *Journal of Operations Management,* **5**, 1, 27-39.

[44] Rees, L.P., Philipoom, P.R., Huang, P.Y., and Taylor III, B.W., 1989, A Comparative Analysis of an MRP Lot-for-Lot System and a Kanban System for a Multistage Production Operation, *International Journal of Production Research,* **27**, 8, 1427-1443.

[45] Rees, L.P., Philipoom, P.R., Taylor III, B.W., and Huang, P.Y., 1987, Dynamically Adjusting the Number of Kanbans in a Just-in-Time Production System Using Estimated Values of Leadtime, *IIE Transactions,* **19**, 2, 199-207.

[46] Russell, G.R., and Philipoom, P.R., 1991, Sequencing Rules and Due Date Setting Procedures in Flow Line Cells with Family Setups, *Journal of Operations Management,* **10**, 4, 524-545.

[47] Sugimori, Y., Kusunoki, K., Cho, F., and Uchikawa, S., 1977, Toyota Production System and Kanban System: Materialization of Just-in-time and Respect-for-human-system, *International Journal of Production Research,* **28**, 1, 1-15.

[48] Takahashi, K., 1994, Determining the Number of Kanbans for Unbalanced Serial Production Systems, *Computers in Industrial Engineering,* **27**, 213-216.

[49] Wemmerlöv, U., and Vakharia, A.J., 1991, Job and Family Scheduling of a Flow-Line Manufacturing Cell: A Simulation Study, *IEE Transactions,* **23**, 4, 383-394.

[50] Vig, M.M., and Dooley, K.J., 1993, Mixing Static and Dynamic Flowtime Estimates for Due-date Assignment, *Journal of Operations Management,* **11**, 67-79.

Appendix A

Analysis of Demand Mean and Variability

Min. Inventory Level		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(141552, 665593, 2250672)	(141552, 1828192, 12537344)
	High	(188736, 2831691, 12537344)	(160920, 2075651, 11963664)

Table A.1: The minimum inventory levels for the proposed algorithm

Min. Inventory Level		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(141552, 835274, 4199200)	(141552, 1537430, 10621760)
	High	(283104, 2068358, 10621760)	(283104, 1833281, 9326688)

Table A.2: The minimum inventory levels for FCFS

Min. Inventory Level		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(94368, 394736, 1280328)	(94368, 717658, 5981832)
	High	(141552, 989227, 5981832)	(141552, 861334, 4834472)

Table A.3: The minimum inventory levels for FCFS-F

Min. Inventory Level		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(141552, 808909, 4199200)	(141552, 1526905, 10621760)
	High	(283104, 2037168, 10621760)	(283104, 1832896, 9326688)

Table A.4: The minimum inventory levels for SPT

Min. Inventory Level		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(94368, 310030, 1108224)	(94368, 586312, 4834472)
	High	(141552, 746289, 4834472)	(141552, 761594, 3019600)

Table A.5: The minimum inventory levels for SPT-F

Inv. Holding Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(102888.4, 549214.5, 1836351)	(102888.4, 499497.1, 1874222)
	High	(155615.8, 2128279, 9619708)	(135737.8, 1727219, 10254816)

Table A.6: The inventory holding costs for the proposed algorithm

Inv. Holding Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(121822.9, 669482.9, 3379578)	(128664.6, 605527.3, 1937132)
	High	(205775.9, 1658785, 9051021)	(205775.9, 1511157, 8069022)

Table A.7: The inventory holding costs for FCFS

Inv. Holding Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(74493.8, 302684, 1006001)	(74493.8, 243450.5, 1069536)
	High	(113272.7, 754449.5, 4794965)	(111866.1, 665429.5, 3789106)

Table A.8: The inventory holding costs for FCFS-F

Inv. Holding Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(121822.9, 648999.5, 3379578)	(128664.6, 606023.8, 1937132)
	High	(205775.9, 1639564, 9051021)	(205775.9, 1517508, 8152027)

Table A.9: The inventory holding costs for SPT

Inv. Holding Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(74383.6, 240575.2, 864347)	(74383.6, 201259.3, 645752.4)
	High	(113420.5, 580206, 3785738)	(117648.4, 606140.5, 2438244)

Table A.10: The inventory holding costs for SPT-F

Backorder Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0, 167.5, 1843.2)	(0, 863.6, 49708.8)
	High	(0, 317566.4, 7630009)	(0, 22195.6, 1160519)

Table A.11: The backorder costs of the proposed algorithm

Backorder Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0, 56117, 1019855)	(0, 10287.8, 424980.8)
	High	(0, 967487.5, 8909065)	(0, 589805.2, 6267150)

Table A.12: The backorder costs of FCFS

Backorder Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1153.2, 84770.2, 564458.8)	(0, 52808, 335049.3)
	High	(0, 612267, 4503146)	(0, 342203.5, 3471774)

Table A.13: The backorder costs of FCFS-F

Backorder Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0, 64258.4, 1382847)	(0, 8660.1, 349270.4)
	High	(0, 945516.3, 8578615)	(0, 560926.7, 6139369)

Table A.14: The backorder costs of SPT

Backorder Cost		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0, 71518.5, 453303.3)	(0, 37336, 292406.8)
	High	(1655.4, 727540.7, 4864888)	(0, 285317.4, 2100244)

Table A.15: The backorder costs of SPT-F

Fill Rate		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1, 1, 1)	(0.931, 0.9991, 1)
	High	(0.53, 0.9683, 1)	(0.803, 0.9950, 1)

Table A.16: The fill rates for the proposed algorithm

Fill Rate		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0.849, 0.9932, 1)	(0.946, 0.9986, 1)
	High	(0.597, 0.9443, 1)	(0.696, 0.9610, 1)

Table A.17: The fill rates for FCFS

Fill Rate		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0.9, 0.9784, 0.999)	(0.936, 0.9872, 1)
	High	(0.751, 0.9460, 1)	(0.806, 0.9630, 1)

Table A.18: The fill rates for FCFS-F

Fill Rate		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0.849, 0.9916, 1)	(0.955, 0.9988, 1)
	High	(0.618, 0.9423, 1)	(0.687, 0.9612, 1)

Table A.19: The fill rates for SPT

Fill Rate		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(0.937, 0.9916, 1)	(0.943, 0.9904, 1)
	High	(0.711, 0.9413, 0.999)	(0.842, 0.9685, 1)

Table A.20: The fill rates for SPT-F

Setup Time		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1426.8, 2096.6, 3082.5)	(1470.2, 2397.8, 4065.2)
	High	(0, 1162.7, 2609.8)	(693.2, 1694.4, 3465.4)

Table A.21: The total setup times for the proposed algorithm

Setup Time		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1069.3, 2597, 3531.1)	(2118, 2865.5, 3854.8)
	High	(690.7, 1645.6, 3186.6)	(1129.6, 1821.7, 3186.6)

Table A.22: The total setup times for FCFS

Setup Time		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1662.8, 2443.9, 3273.5)	(1662.8, 2663.7, 3451.6)
	High	(476.3, 1662.4, 2808.2)	(723.7, 1864.7, 3128.6)

Table A.23: The total setup times for FCFS-F

Setup Time		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1069.3, 2623.8, 3531.1)	(2118, 2865.5, 3854.8)
	High	(690.7, 1684.3, 3186.6)	(1129.6, 1837, 3186.6)

Table A.24: The total setup times for SPT

Setup Time		Demand Variability	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
Demand Mean	Low	(1238.3, 2121.4, 2989.8)	(1456.7, 2335.3, 3402)
	High	(682.3, 1645.2, 2358.8)	(850.7, 1779.4, 2989.8)

Table A.25: The total setup times for SPT-F

Appendix B

Analysis of Number of Families and Parts

Min. Inv. Level		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(141552, 200588, 347696)	(264864, 1983194, 12537344)
	High	(228696, 645675, 2295048)	(719800, 4666666, 12537344)

Table B.1: The minimum inventory levels for the proposed algorithm

Min. Inv. Level		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(141552, 287951, 643680)	(400600, 1719871, 10621760)
	High	(381160, 842486, 2295048)	(1031808, 3394379, 10621760)

Table B.2: The minimum inventory levels for FCFS

Min. Inv. Level		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(94368, 146495, 325872)	(240360, 800405, 5981832)
	High	(152464, 348743, 1228568)	(386928, 1645014, 5981832)

Table B.3: The minimum inventory levels for FCFS-F

Min. Inv. Level		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(141552, 287951, 643680)	(400600, 1700558, 10621760)
	High	(381160, 812340, 2295048)	(1031808, 3378293, 10621760)

Table B.4: The minimum inventory levels for SPT

Min. Inv. Level		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(94368, 128876, 217248)	(160240, 646696, 4834472)
	High	(152464, 277346, 676848)	(386928, 1361460, 4834472)

Table B.5: The minimum inventory levels for SPT-F

Inv. Holding Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(102888.4, 168737, 283796.8)	(221069, 525354.9, 922535.1)
	High	(193218.7, 526780.9, 1830112)	(727558.5, 3683337, 10254816)

Table B.6: The inventory holding costs for the proposed algorithm

Inv. Holding Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(121822.9, 224704.8, 462528)	(338684.1, 735675.5, 1727987)
	High	(321268.1, 668164.4, 1890969)	(850212.8, 2816407, 9051021)

Table B.7: The inventory holding costs for FCFS

Inv. Holding Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(74493.8, 117357.7, 255987.8)	(174194.2, 319573.1, 922214.6)
	High	(109199, 268125.8, 980069.2)	(283238.9, 1260957, 4794965)

Table B.8: The inventory holding costs for FCFS-F

Inv. Holding Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(121822.2, 224738.7, 462528)	(338684.1, 726622.8, 1727987)
	High	(321268.1, 647011.8, 1890969)	(824072.8, 2813721, 9051021)

Table B.9: The inventory holding costs for SPT

Inv. Holding Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(74383.6, 106172.3, 171404.9)	(118349.9, 238612.1, 489939.7)
	High	(113435.3, 216013.4, 533852)	(281219.4, 1067383, 3785738)

Table B.10: The inventory holding costs for SPT-F

Backorder Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0, 0, 0)	(0, 24.8, 601.6)
	High	(0, 808.6, 49708.8)	(0, 339959.7, 7630009)

Table B.11: The backorder costs for the proposed algorithm

Backorder Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0, 2188.7, 169618.6)	(0, 18154.2, 394759.5)
	High	(0, 18359.2, 572901.6)	(0, 1584995, 8909065)

Table B.12: The backorder costs for FCFS

Backorder Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0, 17789.3, 88382.2)	(0, 77421.7, 392233.4)
	High	(0, 96210, 517927)	(23236.4, 900626.7, 4503146)

Table B.13: The backorder costs for FCFS-F

Backorder Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0, 1851.9, 142377.6)	(0, 25158.5, 725491.7)
	High	(0, 28519.6, 440664)	(0, 1523832, 8578615)

Table B.14: The backorder costs for SPT

Backorder Cost		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0, 10632.3, 57258.9)	(0, 66527.8, 362851.3)
	High	(0, 69071.4, 364618.5)	(230, 975481.1, 4864888)

Table B.15: The backorder costs for SPT-F

Fill Rate		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(1, 1, 1)	(1, 1, 1)
	High	(0.931, 0.9991, 1)	(0.53, 0.9634, 1)

Table B.16: The fill rates for the proposed algorithm

Fill Rate		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0.958, 0.9994, 1)	(0.931, 0.9980, 1)
	High	(0.906, 0.9979, 1)	(0.597, 0.9018, 1)

Table B.17: The fill rates for FCFS

Fill Rate		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0.975, 0.9930, 1)	(0.955, 0.9840, 1)
	High	(0.933, 0.9814, 1)	(0.751, 0.9162, 0.987)

Table B.18: The fill rates for FCFS-F

Fill Rate		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0.953, 0.9993, 1)	(0.895, 0.9964, 1)
	High	(0.898, 0.9948, 1)	(0.618, 0.9034, 1)

Table B.19: The fill rates for SPT

Fill Rate		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(0.978, 0.9942, 1)	(0.955, 0.9856, 1)
	High	(0.959, 0.9845, 1)	(0.711, 0.9168, 1)

Table B.20: The fill rates for SPT-F

Setup Time		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(1556.8, 2378.1, 4065.2)	(1128.5, 2060.4, 3351)
	High	(0, 1784, 3327.8)	(0, 1129, 2550.6)

Table B.21: The total setup times for the proposed algorithm

Setup Time		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(1563, 2635.4, 3854.8)	(1288.4, 2299.3, 3470.4)
	High	(1149.4, 2132.2, 3004.6)	(690.7, 1862.9, 3021.6)

Table B.22: The total setup times for FCFS

Setup Time		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(1845.9, 2622.8, 3377)	(1402.4, 2267.1, 3161.6)
	High	(1473.4, 2280.4, 3451.6)	(476.3, 1464.4, 2794.3)

Table B.23: The total setup times for FCFC-F

Setup Time		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(1563, 2635.4, 3854.8)	(1288.4, 2313.9, 3470.4)
	High	(1203.8, 2172, 3004.6)	(690.7, 1889.2, 3021.6)

Table B.24: The total setup times for SPT

Setup Time		number of parts	
		Low(Min., Avg., Max.)	High(Min., Avg., Max.)
number of families	Low	(1998, 2391.8, 2989.8)	(1356.8, 1971.2, 3010.8)
	High	(1205.9, 2036, 3402)	(682.3, 1482.4, 2490.3)

Table B.25: The total setup times for SPT-F

Appendix C

Analysis of Imbalance

Imbalance	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1537542, 12537344)	(141552, 1694768, 12537344)
Inv. hold. cost	(102888.4, 1107860, 7960096)	(102888.4, 1344244, 10254816)
Backorder cost	(0, 85220.6, 7630009)	(0, 85176, 7630009)
Fill rate	(0.53, 0.9906, 1)	(0.53, 0.9906, 1)
Setup time	(0, 1879, 4061)	(0, 1796, 4065.2)

Table C.1: Performance analysis of proposed algorithm for imbalanced systems

Imbalance	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1361891, 10621760)	(141552, 1430420, 10621760)
Inv. hold. cost	(121822.9, 1034627, 7186731)	(121822.9, 1187849, 9051021)
Backorder cost	(0, 387132.8, 8909065)	(0, 424715.9, 8909065)
Fill rate	(0.632, 0.9754, 1)	(0.597, 0.9732, 1)
Setup time	(690.7, 2225.8, 3854.8)	(1041.7, 2239, 3854.8)

Table C.2: Performance analysis of FCFS for imbalanced systems

Imbalance	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(94368, 636928, 5981832)	(94368, 651025, 5981832)
Inv. hold. cost	(74493.8, 471403.2, 3789106)	(74493.8, 511603.5, 4794965)
Backorder cost	(0, 246440.7, 4503146)	(0, 299583.6, 4503146)
Fill rate	(0.751, 0.9700, 1)	(0.751, 0.9674, 1)
Setup time	(476.3, 2163.6, 3451.6)	(547.7, 2153.7, 3451.6)

Table C.3: Performance analysis of FCFS-F for imbalanced systems

Imbalance	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1347406, 10621760)	(141552, 1400134, 10621760)
Inv. hold. cost	(121822.9, 1041118, 7308926)	(121822.9, 1164929, 9051021)
Backorder cost	(0, 363124.4, 8578615)	(0, 426556.4, 8578615)
Fill rate	(0.632, 0.9755, 1)	(0.618, 0.9715, 1)
Setup time	(690.7, 2240, 3854.8)	(1041.7, 2265.4, 3854.8)

Table C.4: Performance analysis of SPT for imbalanced systems

Imbalance	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(94368, 517241, 4834472)	(94368, 532212, 4714624)
Inv. hold. cost	(74383.6, 387811.8, 3724296)	(74383.65, 426278.7, 3785738)
Backorder cost	(0, 260864.5, 4169937)	(0, 299991.9, 4864888)
Fill rate	(0.711, 0.9715, 1)	(0.711, 0.9692, 1)
Setup time	(682.3, 1971.6, 3402)	(707, 1969.1, 3402)

Table C.5: Performance analysis of SPT-F for imbalanced systems

Appendix D

Analysis of S/P ratio

S/P ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1537542, 12537344)	(141552, 1822546, 12537344)
Inv. hold. cost	(102888.4, 979223.9, 8548413)	(125695.2, 1472881, 10254816)
Backorder cost	(0, 25263.8, 1524557)	(0, 145132.8, 7630009)
Fill rate	(0.803, 0.9953, 1)	(0.53, 0.9859, 1)
Setup time	(0, 1693.8, 3185.5)	(0, 1981.9, 4065.2)

Table D.1: Performance analysis of the proposed algorithm for S/P ratio

S/P ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1361891, 10621760)	(235920, 1609707, 10621760)
Inv. hold. cost	(121822.9, 912624.5, 5561502)	(171480, 1309851, 9051021)
Backorder cost	(0, 285885.3, 6266745)	(0, 525963.4, 8909065)
Fill rate	(0.597, 0.9834, 1)	(0.632, 0.9652, 1)
Setup time	(1041.7, 2239, 3854.8)	(690.7, 2102.2, 3531.1)

Table D.2: Performance analysis of FCFS for S/P ratio

S/P ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(94368, 636928, 5981832)	(98144, 726551, 5981832)
Inv. hold. cost	(74493.8, 422791.1, 3296552)	(75184.8, 560215.6, 4794965)
Backorder cost	(0, 199758.6, 3977269)	(0, 346265.7, 4503146)
Fill rate	(0.808, 0.9764, 1)	(0.751, 0.9610, 1)
Setup time	(547.7, 2153.7, 3451.6)	(476.3, 2099.4, 3451.6)

Table D.3: Performance analysis of FCFS-F for S/P ratio

S/P ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1347406, 10621760)	(235920, 1601508, 10621760)
Inv. hold. cost	(121822.9, 896519.5, 5626648)	(171480, 1309528, 9051021)
Backorder cost	(0, 283335.1, 6074489)	(0, 506345.7, 8578615)
Fill rate	(0.618, 0.9822, 1)	(0.632, 0.9647, 1)
Setup time	(1041.7, 2265.4, 3854.8)	(690.7, 2120.8, 3531.1)

Table D.4: Performance analysis of SPT for S/P ratio

S/P ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(94368, 517241, 4834472)	(94368, 564809, 4834472)
Inv. hold. cost	(78619.2, 372376.4, 2944384)	(74383.6, 441714.1, 3785738)
Backorder cost	(0, 238798.1, 3899153)	(0, 322058.2, 4864888)
Fill rate	(0.761, 0.9774, 1)	(0.711, 0.9632, 1)
Setup time	(707, 1969.1, 3402)	(756.3, 1918.1, 3010.8)

Table D.5: Performance analysis of SPT-F for S/P ratio

Appendix E

Analysis of B/I ratio

B/I ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1537542, 12537344)	(141552, 1538562, 12537344)
Inv. hold. cost	(102888.4, 1225217, 10254816)	(102888.4, 1226888, 10254816)
Backorder cost	(0, 56693.37, 3815005)	(0, 113703.3, 7630009)
Fill rate	(0.53, 0.9908, 1)	(0.53, 0.9904, 1)
Setup time	(0, 1843.9, 4065.2)	(0, 1831.8, 4065.2)

Table E.1: Performance analysis of proposed algorithm for B/I ratio

B/I ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1361891, 10621760)	(141552, 1458537, 10621760)
Inv. hold. cost	(121822.9, 1029867, 5105244)	(121822.9, 1192609, 9051021)
Backorder cost	(0, 341761.8, 5324165)	(0, 470087, 8909065)
Fill rate	(0.597, 0.9694, 1)	(0.635, 0.9791, 1)
Setup time	(1041.7, 2267.1, 3854.8)	(690.7, 2197.8, 3854.8)

Table E.2: Performance analysis of FCFS for B/I ratio

B/I ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(94368, 636928, 5981832)	(94368, 705374, 5981832)
Inv. hold. cost	(74493.8, 434692.8, 2384454)	(74493.8, 548313.9, 4794965)
Backorder cost	(0, 230980.6, 2883205)	(0, 315043.7, 4503146)
Fill rate	(0.751, 0.9645, 1)	(0.778, 0.9728, 1)
Setup time	(547.7, 2116.8, 3377)	(476.3, 2200.5, 3451.6)

Table E.3: Performance analysis of FCFS-F for B/I ratio

B/I ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(141552, 1347406, 10621760)	(141552, 1455771, 10621760)
Inv. hold. cost	(121822.9, 1010669, 5145280)	(121822.9, 1195378, 9051021)
Backorder cost	(0, 344193.8, 5189504)	(0, 445487, 8578615)
Fill rate	(0.618, 0.9676, 1)	(0.642, 0.9794, 1)
Setup time	(1041.7, 2290.3, 3854.8)	(690.7, 2215, 3854.8)

Table E.4: Performance analysis of SPT for B/I ratio

B/I ratio	Low (Min., Avg., Max.)	High (Min., Avg., Max.)
Min. inv. level	(94368, 517241, 4834472)	(94368, 570600, 4834472)
Inv. hold. cost	(74383.6, 363437.3, 2438244)	(74383.6, 450653.1, 3785738)
Backorder cost	(0, 56693.37, 3815005)	(0, 113703.3, 7630009)
Fill rate	(0.711, 0.9674, 1)	(0.776, 0.9732, 1)
Setup time	(836.7, 1936.4, 3231.7)	(682.3, 2004.3, 3402)

Table E.5: Performance analysis of SPT-F for B/I ratio

VITA

Feryal Erhun was born in Ankara, Turkey, in 1974. She attended to the Department of Industrial Engineering, Bilkent University, in 1991 and graduated with high honors in July 1995. In September 1995, she joined to the Department of Industrial Engineering at Bilkent University as a research assistant. From that time to the present, she worked with Dr. Selim Aktürk for her graduate study at the same department.