ANALYSIS OF A PULL TYPE REMANUFACTURING CONTROL SYSTEM

by

Özgür İnan Özel

B.S., Industrial Engineering, Yıldız Technical University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University

2006

ACKNOWLEDGEMENTS

With his invaluable help, my thesis supervisor Assist. Prof. Aybek Korugan was the main motivator of this study. I would like to thank him not only for his contributions as an instructor, but also for his great tolerance and help during the work for this thesis.

I would also like to thank Assoc. Prof. Necati Aras and Assist. Prof. Aslı Erdem for their important contributions which allowed this study to be a complete work.

I thank to my all friends especially to Güray and Fatih who are always with me throughout this study.

And my family; my mother, my father and my sisters with their endless help and support always encouraged me throughout my life.

ABSTRACT

ANALYSIS OF A PULL TYPE REMANUFACTURING CONTROL SYSTEM

The aim of this study is to evaluate multi-stage kanban controlled hybrid systems with arbitrary topologies. A hybrid system consists of manufacturing and remanufacturing subprocesses each dedicated to its respective operation. Both raw materials and returned parts are used as inputs in the production system and these components are processed to produce the final product.

A product form approximation technique is used to analyze the production system. The system is modeled as a multi-class closed queuing network. Average number of items in each queue, throughputs of each stage, WIP levels of each queue, average number of backordered demands and effective return rates are some of the performance measures obtained from the analytical method. The results of the analytical model are compared with the simulation model and the numerical examples are illustrated. Then a heuristic search method is given to search the minimum cost of the system. According to this method the number of kanbans is determined for each stage. Several numerical examples are given for possible different scenarios for a hybrid system.

ÖZET

ÇEKME SİSTEMİ İLE BİR YENİDEN ÜRETİM KONTROL SİSTEMİNİN İNCELENMESİ

Bu tezin amacı çekme sistemi ile kontrol edilen çok aşamalı rastgele dizilmiş hibrit sistemleri incelemektir. Hibrit sistem, üretim ve yeniden üretim yapan iki alt süreçten oluşmaktadır. Ham maddeler ve geri dönen parçalar bu sistemde girdi olarak kullanılmakta ve bu girdilerle son ürün oluşturulmaktadır.

Bu çok aşamalı sistemi incelemek için çarpım formlu bir yaklaşım yöntemi kullanılmaktadır. Uygulanan analitik modelle, her kuyruktaki ortalama miktar, ortalama ard ısmarlanan talep, etkin geri dönüş oranı ve diğer performans ölçüleri belirlenebilmektedir. Analitik modelin sonuçları ile benzetim modelinin sonuçları karşıl aştırılmış ve sayısal örnekler verilmiştir. Daha sonra en düşük maliyeti belirleyen arama yöntemi verilmiştir. Hibrit sistem için çok sayıda sayısal durum incelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

b	Backorder cost
B	Buffer size for returned items
C_{Disp}	Cost of disposal
C_M	Cost of manufacturing stage
C_R	Cost of remanufacturing stage
D	Diagonal matrix
D_r	Demand queue of stage-r
G_r	Normalization constant associated with the r-th single class network
h_n	Holding cost of returned items
h_R	Holding cost of remanufactured items
h_M	Holding cost of manufactured items
$h_{r,i}$	Holding cost of queue-i in stage-r
$J_{r,r+1}$	Synchronization station linking stage-r and $r+1$
K_r	Number of kanbans associated to stage-r
K_R	Number of kanbans remanufacturing kanbans
K_M	Number of kanbans manufacturing kanbans
L	Lower triangular matrix
m_r	Number of stations at stage-r
MP_r	Manufacturing process of stage-r
$\overline{n}$	State vector of a network
N_r	Total number of class-r customers
P_B	Percentage of backordered demands
P_r	Finished parts queue of stage-r
$P(n_r)$	Steady state probabilities for the r-th single class network
$\tilde{P}_{ri}(n_{ri})$	Marginal probabilities of station-i in stage-r in isolation
$P_{ri}(n_{ri})$	Marginal probabilities of station-i in the single class network for stage-r
$P_i(n)$	Steady state probabilities of station-i

Q	Transition rate matrix
Q_D	Average number of backordered demands
Q_{Rj}	Average number in returned items buffer of stage j, j=1,2
$Q_{r,i}$	Average number in queue of station-i in stage-r
r	Discount rate
$S(r)$	Set of stations visited by class-r customers
TH_r	Throughput of stage-r
U	Upper triangular matrix
V_{ri}	Visit ratios in the r-th single class network
v_{ri}	Conditional throughputs of station-i in stage-r in the original network
$\tilde{v}_{ri}$	Conditional throughputs of station-i in stage-r in isolation
w	Relaxation parameter
W_B	Average waiting time of backordered demands
W_R	Average waiting time of returned items
$WIP_{Qr,i}$	WIP cost of $Q_{r,i}$
Z	Total cost of the system
Z_{Opt}	Optimum cost of the system
Z_{Best}	The minimum cost found by the search method
Z_{Back}	Total backorder cost
Z_{Disp}	Total disposal cost
Z_{Q_R}	Total cost of holding returned items
Z_{TH}	Total production cost
Z_{gWIP}	Cost of WIP just in the production system
Z_{WIP}	Total holding cost
λ_D	The rate of external demands
$\lambda_{ri}(nri)$	State dependent arrical rate of class-r customers to station-i
$\lambda_M(i)$	Arrival rate to synchronization station from manufacturing process
$\lambda_R(i)$	Arrival rate to synchronization station from remanufacturing process

$\mu_{ri}(n_{ri})$	Load-dependent service rate of station-i in stage-r
π	Stationary probability vector
γ_j	Return rate for component-j
γ_{eff}	Effective return rate
Λ	Demand rate
KCS	Kanban control system
SSOR	Successive symmetric overrelaxation method

1. INTRODUCTION

Production systems managed with pull control policies have gained great importance after the success of Toyota Production System and they are widely used in the industry. In a pull type control, production is triggered by the occurrence of actual demands. These systems are motivated by the concept of just-in-time (JIT) production whose objective is that materials should be present just when they are needed, and in the quantities needed [1]. The concept of pull type control systems accommodates various kinds of pull policies and the kanban (Japanese word means card) system is the most known type. The kanban control system is simple to manage and implement. The information is transferred from downstream to upstream of the production system. In general, manufacturing systems are decomposed into stages and coordination of flow between these stages is controlled by kanbans. Stages of the manufacturing system may consist of a single machine, multiple machines, or a more complex system (e.g., a manufacturing flow line). In a system like flow lines, parts are pulled between stages while they are pushed through the machines of each stage. Such a multistage kanban system can be viewed as a hybrid system combining push and pull control mechanisms.

The simple kanban system and CONWIP (Constant Work in Process) system are two popular cases of the traditional kanban control system. In the simple kanban system each stage consists of a single machine. This kanban control system can thus be viewed as a pure pull system. The CONWIP system is a single stage kanban system. The entrance of the parts to the system is controlled at the input of the production system according to the demand for final products. The parts are then pushed through the whole production system up to the finished good inventory. At any time, the number of work in process in the production system is constant and limited with the number of kanbans associated with the system.

The second concept used in this study is remanufacturing. Remanufacturing of products is one of the highly discussed topics in today's world. Reuse of products becomes an important issue due to environmental concerns and legislative rules. Also economical benefits of reuse of products make companies to deal with this subject.

The manufacturing systems joint with remanufacturing processes are called hybrid systems. The analysis of hybrid systems differs from traditional systems in many aspects. The uncertainty in the behaviour of products return flows is major reason. These points are briefly given in Chapter 2.

In this study we deal with a kanban controlled hybrid system. The system is modeled as a queuing network and solved analytically. We try to figure out when and how the remanufacturing option is beneficial from the point of the manufacturer. More generally we try to develop a tool that is used for the evaluation of pull type hybrid systems. This thesis deals with two main problems. The first one is the evaluation of the parameters of the analytical model, while the other one is the determination of minimum costs and kanban allocation.

The rest of the thesis is organized as follows. Chapter 2 provides an overview of the kanban and remanufacturing literature. In Chapter 3 we describe the analytical method that we use in this study. Chapter 4 represents the main objectives of the thesis. In Chapter 5 the investigated problem is defined, the system is evaluated and validation of the system is given. The numerical examples are illustrated in Chapter 6. And the last chapter is the conclusion for the thesis.

2. LITERATURE SURVEY

This study is based on two main concepts. The first one is kanban control policy and the other one is remanufacturing of returned items. The research related with these topics are briefly given in the following sections. In each section the general view of the concepts and the work related to our study is presented.

2.1. Kanban Literature

In the basic kanban system, the parameters of the system are the number of kanbans for each stage. These parameters characterize the production system and play an important role in the efficiency and effectiveness of the system. For the design and the operational issues these parameters have to be determined efficiently. To analyze the kanban systems both simulation and analytical methods have been used in the literature. Simulation may be time consuming because many different variations of the system have to be tested. Analytical models on the other hand, give rapid results and save considerable amount of time. These models are based on stochastic modelling of the kanban-controlled manufacturing systems.

In the literature there is considerable research related to JIT philosophy and many researchers offer studies about the behaviour of kanban systems and develop models to evaluate their performances. Most of the researchers use simulation models to analyze the kanban systems but there are also analytical methods in the literature. In the last decade efficient analytical methods have been developed. For a comprehensive list of studies and evolution of analytical models we refer the reader to Uzsoy and Martin-Vega [2] and Di Mascolo et al. [1] and the references therein.

In general, the analytical models cannot be solved exactly because their structure is not simple enough and their state space is too large. Therefore, most analytical methods use an approximation.

Di Mascolo et.al.[1] propose an analytical method that can handle multi-stage systems where stages can consist of any number of machines. This method appears to be of special interest in the literature because most analytical methods that have been proposed for kanban systems assume that each stage consists of a single machine, i.e., simple kanban systems. They model the kanban system as a closed queuing network with synchronization mechanisms. The method is based on a general-purpose technique known as product-form approximation proposed by Baynat and Dallery [3]. The basic principle is to decompose the original system into a set of subsystems to use the proposed approximation method. Then the authors utilize an iterative procedure to determine the unknown parameters.

Di Mascolo and Dallery[4] study kanban controlled assembly systems. They define two different kanban-type control mechanisms, i.e. assembly with simultaneous release and assembly with independent release. The authors model these systems as queuing networks with synchronization mechanisms and use the analytical method proposed by the authors[1] for the performance evaluation of the model.

Baynat et.al.[5] propose a different way of deriving the analytical method proposed by Di Mascolo[1]. A closed queuing network with synchronization stations again represents the kanban system. They model the system as a multi-class queueing network in which each kanban loop represents a class of customers. Similarly they use the product form approximation technique proposed by Baynat and Dallery[6]. This method is equivalent to Di Mascolo's method, however its computational algorithm is more efficient and computational complexity is reduced. In Di Mascolos's algorithm two levels of iterations are required, in Baynat's algorithm it is reduced to a single level iteration. Another advantage of this method is that it can be used to analyze more general kanban systems such as assembly kanban systems, kanban systems with multiple consumers and multiple suppliers and generalized kanban systems.

In a recent paper, Matta et.al.[7] propose a new approximate analytical method for evaluating the performance of assembly systems with independent release of kanbans. This method is based on product-form approximation[8, 3] and multi-class aggregation techniques[5, 9] developed by Baynat and Dallery. The authors also compare the two different control policies for releasing kanbans in assembly systems for identifying the most suitable application areas of these control policies.

Tardif and Maaseidvaag[10] propose a new adaptive kanban-type pull control mechanism for single-stage kanban systems, which determines when to release or reorder raw parts based on customer demands, inventory and backorders. Here the number of kanbans is allowed to change with respect to the inventory and backorder levels. A single-stage single-product kanban system is considered. It is shown that under certain conditions the adaptive kanban control system can outperform the traditional kanban control system.

2.2. Remanufacturing Literature

Within the past two decades the reuse of products at the end of their useful lives has become an important issue due to environmental concerns and legislative rules. Also economical benefits gained by the reuse of products enable manufacturers to decrease their production costs. Hence increasingly many companies find this concept important. Collecting returned products and bringing them to a useful state via repair, refurbishing and rework activities is called remanufacturing. Some companies are specialized only in remanufacturing and they remanufacture other manufacturers' products. On the other hand, some companies choose to handle the remanufacturing issue by themselves in addition to their manufacturing processes. Companies that are involved in both manufacturing and remanufacturing activities are called hybrid companies. Some of the Original Equipment Manufacturer's (OEM) like IBM, Xerox and HP choose to undertake the remanufacturing activities by themselves.

Parallel to the development in the remanufacturing industry, the literature on this subject has grown in recent time. Guide, Jayaraman and Srivastava [11] provide a state-of-art survey in the production planning and control of remanufacturing systems. In this paper the characteristics of the remanufacturing systems are discussed to differentiate these systems from other manufacturing systems. The production planning and control perspective of remanufacturing companies are examined.

Several authors propose inventory control methods for remanufacturing systems. Generally the relationship between the returns and the optimal order quantity is investigated. Van der Laan et.al. analyze different inventory policies based on stochastic models to model the uncertainties in the return behaviour[12, 13]. In [14] the authors show that pull control policy is more cost effective than push control policy for the systems with return flows. Souza and Ketzenberg [15] study the two stage remanufacturing models, which also consider the operational disposals at the remanufacturing stage.

Teunter [16] studies a deterministic system with continuous review and zero lead times. In this paper the remanufactured products are assumed to be as good as new products. An EOQ control mechanism with fixed batch sizes is proposed.

Bayındır et.al.[17] investigate possible benefits of remanufacturing in inventory-related costs. The authors model their system as a queuing network and used the return ratio as a decision variable to determine whether the remanufacturing option is cost attractive or not.

Teunter et.al.[18] propose a method for setting the holding cost rates in average cost inventory models with reverse logistics. For the average cost (AC) inventory models, in general it is common to include an opportunity cost rate in the holding cost rate and the traditional way for calculating the opportunity cost rate is to multiply the interest rate (discount rate) by the marginal cost for producing/ordering an item. For single source inventory systems this method gives near optimal results from a discounted cash flow (DCF) point of view. However for the systems with reverse logistics this situation is not the same and the performance of the AC approach stands or fails with the right choice of the holding cost parameters. The authors propose a set of holding cost parameters that are suitable for reverse logistic system with a simulation study. Van der Laan [19] shows these results mathematically.

Aksoy and Gupta[20] study a buffer allocation plan for a remanufacturing cell. They model the system as an open queuing network and propose an algorithm that uses decomposition principle and expansion methodology to analyze the remanufacturing cell with finite buffers and unreliable servers. The buffer allocation algorithm distributes a predetermined number of available buffer slots among various stations to optimize the cell's performance.

Gungor and Gupta [21, 22] study the disassembly problem in remanufacturing. They propose a methodology that creates a disassembly precedence matrix and using this matrix they present a method for disassembly line balancing.

Kizilkaya and Gupta [23] study the disassembly problem in the remanufacturing systems controlled with pull systems. They present a dynamic kanban control specifically developed for disassembly lines. Unlike traditional manufacturing systems, where the external demand occurs only at the last station, in this system demand for disassembled parts occur at any of the intermediate stations. The authors construct a simulation model and show that for disassembly line balancing problem the dynamic kanban system is superior to the modified kanban system that was proposed by the authors in an earlier study [24].

Udomsawat et.al.[25] propose a multi-kanban mechanism for personal computer disassembly. They discuss some complications including product arrival, demand arrival, inventory fluctuations and production control mechanisms in the disassembly problem. The multi-kanban mechanism relies on dynamic routing of kanbans according to the state of the system. In this study also a simulation model is used to analyze the remanufacturing system.

Korugan and Gupta[26] develop an analytical model for a pull type hybrid production system with two discrete production lines where one manufactures new products while the other is reserved for remanufacturing activities. The problem is modeled as a stochastic system with exponentially distributed i.i.d. variables for demand occurrences and service completions. For this hybrid system, single-stage pull-type control mechanisms with state dependent and state independent routing are developed.

Korugan and Gupta[27] analyze a hybrid manufacturing system with an adaptive kanban control mechanism. The system is managed by a single stage pull type control mechanism with adaptive kanbans and state independent routing of demand information.

As stated above, in the literature there are a few studies that evaluate pull type hybrid systems with analytical methods but they are restricted to single-stage models. On the other hand, studies about multi-stage pull controlled hybrid systems are evaluated using simulation but not analytical methods. In this study, our aim is to analyze these multi-stage pull controlled hybrid systems with arbitrary topology using an analytical model.

3. THE METHODOLOGY

Di Mascolo et.al.[1] propose an analytical method that evaluates multi-stage kanban controlled systems, which can consist of any number of machines in stages. The method is based on an approximation method and the results are fairly accurate. Baynat et.al.[5] propose a new analytical method by improving Di Mascolo et.al's method. The new method reaches the same results as the previous one, but its contribution is to reduce the computational burden and it provides a general framework for the analysis of more general kanban systems. In this dissertation we adopt this new multiclass approximation technique for the analysis of kanban-like control systems and this technique is explained in the following sections.

3.1. Modelling a Kanban Control System

3.1.1. The Kanban Control System

As a general principle of the kanban control systems, the manufacturing system is decomposed into several stages as a manufacturing process and an output buffer for each stage. The manufacturing process consists of a set of machines and contains parts that are either waiting for or being processed at different machines [5]. A fixed number of kanbans are associated with each stage. Kanbans are used as production orders and coordinate the flow of materials between stages. A part can enter a stage if there is a corresponding kanban available, i.e. if there is a stage-r kanban available, it is attached to the part so that part can proceed to stage-r. After completion of its process, the part is placed into the output buffer of stage-r. As soon as, there is an available kanban of the next stage i.e., stage-r+1, the part is transferred to the next level. At that time, the stage-r kanban that was attached to the part is detached and stage-r+1 kanban is attached to the part. The stage-r kanban then returns back to the entrance of stage-r and becomes available for allowing a new part to enter this stage. The information of the consumption of a part is immediately transferred from downstream to upstream as long as there are finished parts waiting in each stage [5].

To explain the methodology, a kanban control system producing only one type of part is considered. Each manufacturing process-r, consists of arbitrarily linked m_r machines. The processing time distribution of each machine is characterized by a phase-type (PH) distribution (i.e. a mixture of exponential distributions). The routing of parts in the stages is probabilistic. The number of kanbans associated with any stage-r is K_r. The kanbans are associated with individual parts and their return from output of the stage to the input of the stage is instantaneous. Demands arrive to the output of the last stage and if they are not immediately satisfied they are backordered. The interarrival time distribution of demands is characterized by a Poisson distribution.

Percentage of demands that are not immediately satisfied, average number of backordered demands, average waiting time of backordered demands, average work in process (in each manufacturing stage and in each output buffer) are some of the important performance measures [5].

3.1.2. Queuing Network Model of The Kanban Control System

The kanban control system(KCS) is modeled by a queueing network with synchronization mechanisms [1]. For illustration the queuing network for a three-stage kanban control system is given in Figure 3.1. Each stage of the network is composed of a manufacturing process and synchronization stations, and each manufacturing process, MP_r, consists of m_r stations. In this case the number of stations in each manufacturing process is, $m_1 = 2, m_2 = 5$ and $m_3 = 3$. In subnetwork MP_2 a part can alternatively be sent via different paths. In these alternative paths the same operations are realized and the routing of parts are probabilistic.

The stages are linked via synchronization stations. The synchronization station linking stage-r and r+1 is denoted by $J_{r,r+1}$ and two upstream queues are denoted by P_r and D_{r+1}. Finished parts of stage-r to which a stage-r kanban is attached are accumulated in queue P_r and demands from stage-r+1 associated with a stage-r+1 kanban are accumulated in queue D_{r+1}.

The behaviour of synchronization station $J_{r,r+1}$ is the following: As soon as there are entities (at least one) in queues P_r and D_{r+1}, one entity is removed from each of them and an entity (representing a finished part of stage-r to which a stage-r+1 kanban is attached) joins the first queue of MP_{r+1}, and another entity (representing a demand together with the stage-r kanban) joins queue D_r.

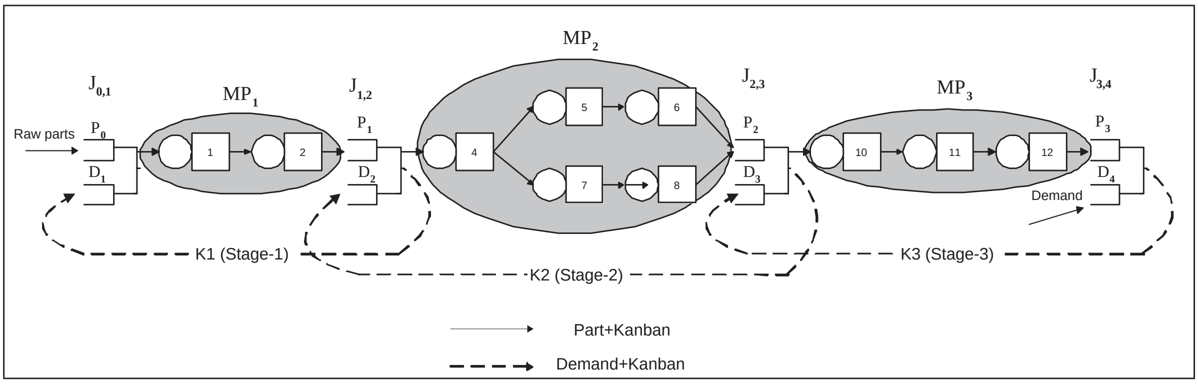

Figure 3.1. Queueing network model for a KCS with three stages in series

The last synchronization station of the last stage, $J_{N,N+1}$ synchronizes the finished products of the system and the external demands. The number in queue P_N represents the number of finished products currently available in the output buffer of the production system. The number in queue D_{N+1} represents the number of external demands currently backordered. External demands arrive to queue D_{N+1} and interarrival time of demands follow a Poisson process with rate λ_D. As soon as there is one entity in both queues P_N (representing a finished product with a stage-N kanban attached) and D_{N+1} (representing an external demand), a demand is satisfied and a stage-N kanban joins queue D_N. The first synchronization station, $J_{0,1}$ synchronizes raw materials/parts coming from outside and stage-1 kanbans. Raw materials are accumulated in queue P_0 and demands of stage-1 associated with stage-1 kanbans are accumulated in queue D_1. If we assume that there is infinite supply of raw materials/parts for the first stage, the queue P_0 will never be empty and the demands of stage-1 will never be backordered, and therefore the queue D_1 will be always empty. As a result, as soon as an entity (representing a demand associated with a stage-1 kanban) arrives in queue D_1, it is instantaneously synchronized with a raw part and joins the first queue of MP_1. Then in this case synchronization station $J_{0,1}$ can be removed from the queueing network model. The resulting model can be seen in Figure 3.2.

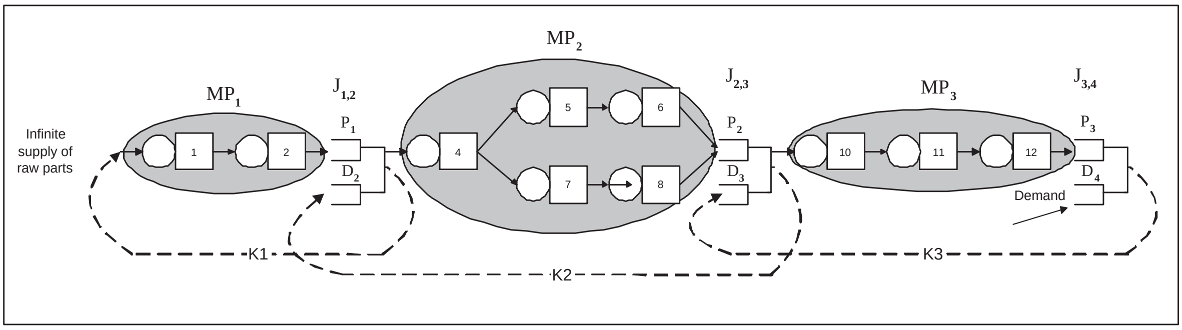

Figure 3.2. Queueing network model for a KCS with an infinite supply of raw materials

In this dissertation we will model the arrival pattern of returned products to the remanufacturing stages with the kind of synchronization stations given in Figure 3.1. Also we assume infinite supply of raw materials exists for manufacturing stages, so we omit synchronization stations for these stages. Similar to the removal of the first synchronization station we can remove the last synchronization stations in the presence of continuous demand. These kinds of systems are defined as saturated systems in the literature and the production capacity is the most important parameter of these systems.

3.1.3. Multiclass Model of the Kanban Control System

The kanbans are considered as the customers of the queueing network of the manufacturing system in Figure 3.2. In order to see the kanbans as customers of the network we can focus on the kanbans of a particular stage, r. A stage-r kanban constitutes a cycle while moving on its path; it waits in queue D_r for a part coming out of stage r-1 ; when the part is available, it is attached on the part and goes through stage-r; then it waits for a demand in queue P_r; finally when the part is synchronized with a stage r+1 kanban, stage-r kanban is detached from the part and returns to the queue D_r. Since the number of stage-r kanbans is constant, they form a closed population.

Thus, the queueing network of Figure 3.2 can equivalently be viewed as a multi-class closed queueing network with synchronization mechanisms. In each stage kanbans are viewed as customers, each kanban loop corresponds to a class of customers. There are N stages so there are N classes of customers and the number of class-r customers is equal to stage-r kanbans (K_r).

In the multiclass queueing network the interaction between different classes of customers takes place only at the synchronization stations. For example, class-r customers ($1 < r < N$) synchronize with class r-1 customers at synchronization station $J_{r-1,r}$ and with class r+1 customers at synchronization station $J_{r,r+1}$. Class-N customers synchronize with class N-1 customers at synchronization station $J_{N-1,N}$ and with external demands at synchronization station $J_{N,N+1}$. However the service stations are visited only by a single class of customer, namely service stations of subnetwork MP_r are only visited by class-r customers. The multi-class closed queueing network is shown in Figure 3.3.

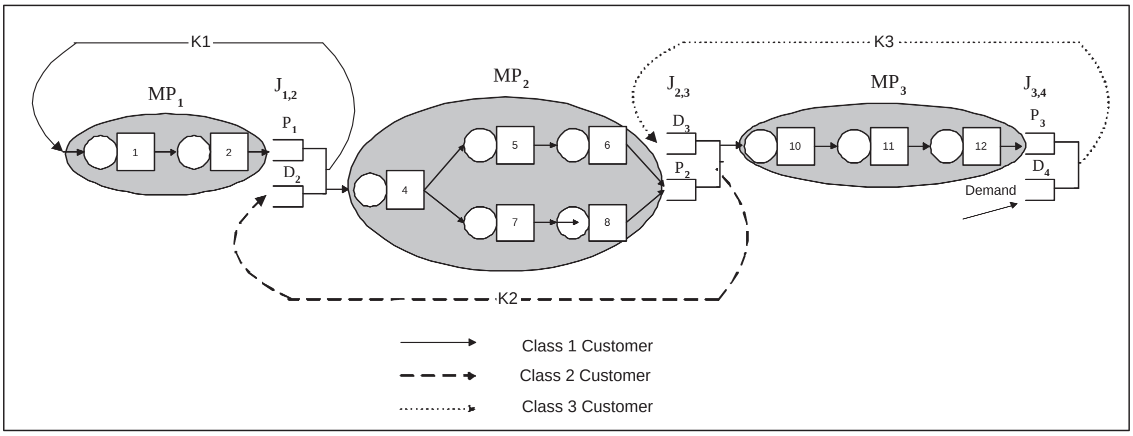

Figure 3.3. The multi-class closed queueing network model

3.2. Product-form Approximation Technique

The product form approximation technique proposed by Baynat and Dallery [6] is used to analyze the queueing network model of the kanban control system. This technique is devoted to the analysis of multi-class closed queueing networks with several non-classical mechanisms such as class dependent general service time distributions, priorities, synchronization mechanisms, resource sharing, fork/join mechanisms, etc.

[5]. Baynat et.al. present this technique for the multi-class closed queueing networks with three kinds of stations: stations with general service-time distributions visited by a single-class of customers; synchronization stations between two classes of customers; and synchronization stations between customers of a given class and external resources. In this dissertation besides these we analyze synchronization stations for assembly with three classes of customers, synchronization stations for assembly with three classes of customers with priority and synchronization stations between customers of a given class and external resources (returned products).

3.2.1. General Principle

Let the original multi-class closed queueing network be composed of M stations and R classes. Let N_r represent the total number of class-r customers and *S(r)* $\subset (1, ..., M)$ represents the set of indexes of the stations that are visited by class-r customers, r = 1,...,R. The main idea of the product-form approximation method is to represent the original multi-class network with R single-class product-form networks, which models the behaviour of a particular class of customer.

Consider a single-class closed queuing network of a particular class, i.e. r. For each station-i of the original network visited by class-r customers, i.e., $i \in S(r)$, a load-dependent exponential service station is associated. Thus, the single-class network for class-r consists of *S(r)* stations. For the r-th single-class network $\mu_{ri}(n_{ri})$, (n_{ri})= 1,...,N_r, denote the load-dependent service rates of station-i, $i \in S(r)$. V_{ri}, $i \in S(r)$ represents the visit ratios in the r-th single-class network and they are defined as the visit ratios for the r-th class in the original network. Since the single-class networks are Gordon-Newell networks , the steady-state probabilities P(n_r) for the r-th network, r= 1,..., R, have the following product-form solution.

$$P(n_r) = \frac{1}{G_r} \prod_{i \in S(r)} [\prod_{n=1}^{n_{ri}} \frac{V_{ri}}{\mu_{ri}(n)}] \tag{3.1}$$

where G_r is the normalization constant associated with the r-th network. This way, the performance of the single-class of customers in the r-th single-class network approximates the performance of the r-th class of customers in the original network.

This is the first stage of the approximation of the method. The second stage is to state the load dependent service rates $\mu_{ri}(n_{ri})$, (n_{ri})= 1,...,N_r, $i \in S(r)$. Ideally, the service rates $\mu_{ri}(n_{ri})$, of the flow equivalent service centers of the class-r product-form network should be equal to the conditional throughputs $v_{ri}(n_{ri})$ of class-r customers at the corresponding station (station-i) in the original network. The conditional throughput $v_{ri}(n_{ri})$ is defined as the average flow of class-r customers out of station-i, given that (n_{ri}) class-r customers are present at the station (regardless of the number of customers of the other classes simultaneously present at the station). However, obtaining the exact values of the conditional throughputs would, in general, require an exact solution of the original system. Therefore, the idea is to approximate the conditional throughputs. These approximations will be used instead of the exact values in order to obtain the service rates of the flow equivalent service centers.

In order to reflect the behavior of the customers in the r-th single-class network close to the class-r customers in the original network; the attitude of the other classes has to be reflected in the approximations of the conditional throughputs in an accurate manner. Thus approximation of the conditional throughputs of the different classes of customers at the different station of the system needs particular attention. These parameters are obtained by analyzing each station in isolation [5].

3.2.2. Analysis of a Station in Isolation

Analysis of a station in isolation to approximate the conditional throughputs of each class is depicted in Figure 3.4. Consider each station-i as a queue fed by external arrival processes, one for each class of customers, which are assumed to be a state-dependent Markovian process with rates $\lambda_{ri}(n_{ri})$, n_{ri} = 0,...,N_r-1, where n_{ri} is the number of class-r customers present at station-i.

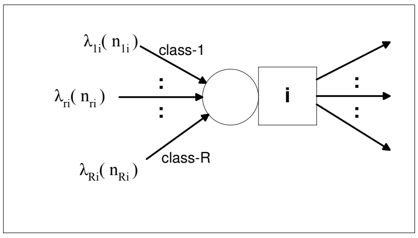

Figure 3.4. Analysis of station-i in isolation

In the beginning assume that the arrival rates for the different classes of customers at station-i is known; namely $\lambda_{ri}(n_{ri})$ for all r such that $i \in S(r)$, and n_{ri} = 0,...,N_r-1, are defined. Then using any appropriate technique this queue can be analyzed in isolation; steady state probability vector can be determined. The marginal probabilities $\tilde{P}_{ri}(n_{ri})$ that n_{ri} customers are present in the isolated station, n_{ri} = 0,...,N_r, can then be derived easily for each class of customers. By using marginal probabilities and arrivals rates the conditional throughputs can be estimated with the following equation;

$$\tilde{v}_{ri}(n_{ri}) = \lambda_{ri}\frac{\tilde{P}_{ri}(n_{ri}-1)}{\tilde{P}_{ri}(n_{ri})} \; for \; n_{ri} = 1, ..., N_r \tag{3.2}$$

The load dependent service rates of the associated station of the single-class product form networks are then set equal to the estimated conditional throughputs of the associated station, i.e.:

$$\mu_{ri}(n_{ri}) = \tilde{v}_{ri}(n_{ri}) \; for \; all \; r \; such \; that \; i \; \in S(r) \; and \; n_{ri} = 1, ..., N_r \tag{3.3}$$

The state dependent arrival rates $\lambda_{ri}(n_{ri})$ of each class of customers at the different stations have to be calculated to realize this approach. These quantities can be obtained from the product-form solutions of the single-class networks given by Equation 3.1. The arrival rates of a given class of customers, r such that $i \in S(r)$, at station-i, i.e., $\lambda_{ri}(n_{ri})$, n_{ri} = 0,...,N_r-1, only depend on the number of class-r customers currently present at this station. The state-dependent arrival rates can be obtained from the expression of the product-form solution of the class-r single-class network, as:

$$\lambda_{ri}(n_{ri}) = \mu_{ri}(n_{ri}+1)\frac{P_{ri}(n_{ri}-1)}{P_{ri}(n_{ri})} \; for \; n_{ri} = 0, ..., N_r - 1 \tag{3.4}$$

$P_{ri}(n_{ri})$ represents the marginal probabilities for the exponential server associated with station-i in the single-class network for class-r. These marginal probabilities can be calculated by using any computational algorithm for single-class closed queuing networks. We use the normalization constant approach [28] to determine these probabilities and this method is described in the next subsection. The marginal probabilities in the equivalent single-class networks are the same as those of the corresponding subsystem in isolation, i.e.:

$$P_{ri}(n_{ri}) = \tilde{P}_{ri}(n_{ri}) \; for \; all \; r \; such \; that \; i \; \in S(r) \; and \; n_{ri} = 0, ..., N_r$$

In order to finalize the procedure the algorithm in Section 3.2.3 can be used to determine the required parameters. The number of iterations to achieve convergence is usually very reasonable [5].

3.2.2.1. Normalization Constant Approach. We use the normalization constant approach to determine the marginal probabilities in Equation 3.4 for the exponential servers in the single class networks.

Normalization constant can be used for both single job class closed queuing networks and multiple job class closed queuing networks. We assume that the manufacturing system produces only one type of product and single job class closed queuing networks that we analyze have a finite number (M) of stations. We let n_i the number of parts present at station-i (n_i includes parts both waiting for and receiving services at station-i).N=n_1 +...+n_i +...+n_M denotes the total number of parts in the closed queuing network. The state of the network is determined by the number of parts at each station and the state vector can be given as:

$$\bar{n} = (n_1, n_2, ..., n_i, ...n_M)$$

Each station in the network is characterized by a service rate μ_i, which means the number of jobs departing station-i per unit of busy time of the station. In the algorithm that we use, the service rates are load-dependent namely depend on the length of the associated queue. We denote the load dependent service rate as $\mu_i(k)$ where k represents the number of jobs present at station-i. The proportion of transition from station i to station j is the transition frequency and denoted as p_{ij}.

The equilibrium (steady state) distribution of jobs in the network is given by:

$$P(\bar{n}) = P(n_1, n_2, ..., n_i, ...n_M) = \frac{1}{G(N)} \prod_{i=1}^{M} f_i(n_i) \tag{3.5}$$

where,

$$f_i(n_i) = \frac{{e_i}^{ni}}{\prod_{k=1}^{n_i} \mu_i(k)}$$

if station-i load dependent.

The e_i are the solutions of the M linear equations that equate the flows into and out of each station.

$$e_i = \sum_{i=1}^{M} e_i p_{ij} \text{ where } j\text{=1,2,....,}M$$

$G(N)$ is the normalization constant that makes all the $P(n)$ sum to one. Since each feasible system state is in the form of $(n_1, n_2, ...n_i...n_M)$ vector composed of non-negative integers that sum to N, we can denote the set of all possible states for a closed queueing network with M stations and N customers as;

$$S(N,M)= \left\{(n_1,n_2,...n_i...n_M) | \sum_{i=1}^{M} n_i = N, n_i \geq 0, i=1....M\right\}$$

Since the sum of all the state probabilities must equal to one the expression is:

$$1 = \sum_{\bar{n}\in S(N,M)} P(\bar{n}) = \frac{1}{G(N)} \sum_{\bar{n}\in S(N,M)} \prod_{i=1}^{M} f_i(n_i)$$

and thus the normalization constant is given by;

$$G(N) = \sum_{\bar{n}\in S(N,M)} \prod_{i=1}^{M} f_i(n_i) \tag{3.6}$$

The normalization constant approach is presented in detail in the book "Computational Algorithms for Closed Queueing Networks" of Bruell and Balbo [28].

<u>3.2.2.2. SSOR (Successive Symmetric Overrelaxation) Method.</u> When we cannot determine the marginal probabilities in Equation 3.2 in terms of equations, we use an iterative method to obtain these probabilities. For large state spaces of assembly synchronization stations (when dimension of the state structure is more than 2) we apply the SSOR method and find a matrix geometric solution.

If we have a transition probabilities matrix P, under certain conditions we can compute the long run probability vector with the power method. In terms of equations we mean:

$$\pi^{(k)} = \pi^{(k-1)}P = \pi^{(k-2)}P^2 = \pi^{(k-3)}P^3 = ... = \pi^{(0)}P^k$$

When the Markov chain is finite, aperiodic and irreducible, the vectors $\pi^{(k)}$ converge to the stationary probability vector π, independent of the choice of the initial vector.

$$\lim_{k \to \infty} \pi^{(k)} = \pi$$

For the discrete case the state probability vector may be computed by the power method. Similarly for the continuous case iterative methods may be applied to the infinitesimal generator matrix Q, namely the matrix of transition rates.

In the discrete case the power method is used to obtain the solution of an eigenproblem ($\pi P = \pi$) but in the continuous case iterative methods are used to obtain the solution of the homogeneous system of linear equation πQ =0.

The standard and well-known iterative methods for the solution of system of linear equations are the methods of Jacobi, Gauss-Seidell and SOR (successive overrelaxation).

These methods are derived from a nonhomogeneous systems of linear equations in the form of $Ax = b$ and the iterative form is

$$x^{(k+1)} = Hx^{(k)} + c, k = 0, 1, 2....$$

where H is the iteration matrix. To obtain the iteration matrix the coefficient matrix A is split as;

$$A = M - N, \text{ then we have } (M - N)x = b \text{ or } Mx = Nx + b$$

which leads to the iterative procedure

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b = Hx^{(k)} + c, \; k = 0, 1, 2....$$

We wish to solve the homogeneous system of equations $Q^T\pi^T = 0$. For notational convenience, say $x = \pi^T$.

The SOR method is applied to the homogeneous system $Q^Tx = (D-L-U)x$ =0, where D is as diagonal matrix and L and U are respectively strictly lower and strictly upper triangular matrices. (The matrices L and U are not the LU factors obtained from direct methods such as Gaussian elimination). The iteration form of SOR method in matrix form is;

$$x^{(k+1)} = (1-w)x^{(k)} + w[D^{-1}(Lx^{(k+1)} + Ux^{k})]$$

and the iteration matrix for the SOR method is

$$H_w = (D - wL)^{-1}[(1-w)D + wU]$$

where w is the relaxation parameter. For $w > 1$, the process is said to be one of overrelaxation; for $w < 1$ it is said to be underelaxation. The SOR method converges only if $0 < w < 2$.

If the coefficient matrix A is symmetric, then the Symmetric Successive Overrelaxation method (SSOR), combines two SOR sweeps together in such a way that the resulting iteration matrix is similar to a symmetric matrix. This situation rarely occurs in markov chain models. In this case SSOR may help to reduce poor convergence behaviour that result from a badly ordered state space. The SSOR is a forward SOR sweep followed by a backward SOR sweep in which the unknowns are updated in the reverse order. The forward and backward sweeps can be carried out for the price of a single forward (or backward) sweep. The SSOR sweeps with relaxation parameter w are written in matrix form as

$$(D - wL)x^{k+1/2} = [(1-w)D + wU]x^{(k)}$$
$$(D - wU)x^{k+1} = [(1-w)D + wL]x^{(k+1/2)} \tag{3.7}$$

When the convergence is achieved, x^k represents the steady state probability vector and then desired marginal probabilities can be evaluated easily [29].

3.2.3. The Algorithm

In this section the algorithm that evaluates the parameters of a multi-class queuing network is given.

- Step 0. For $r = 1, ..., R$,
  Set $\mu_{ri}(n_{ri})$, for $i \in S(r)$ and n_{ri} = 1,...,N_r, to some initial values
- Step 1. For $r = 1, ..., R$,
  (a) Calculate the marginal probabilities $P_{ri}(n_{ri})$, for $i \in S(r)$ and n_{ri} = 1,...,N_r,in the r-th single class network
  (b) Derive arrival rates $\lambda_{ri}(n_{ri})$, for $i \in S(r)$ and n_{ri} = 0,...,N_r-1, from Equation (3.4)
- Step 2. For $r = 1, ..., M$,
  (a) Analyse station-i in isolation according to Figure 3.4
  (b) Derive the marginal probabilities $\tilde{P}_{ri}(n_{ri})$, for all r such that $i \in S(r)$ and n_{ri} = 0,...,N_r.
  (c) Calculate the conditional throughputs $\tilde{v}_{ri}(n_{ri})$, for all r such that $i \in S(r)$ and n_{ri} = 1,...,N_r, using equation (3.2)
- Step 3. For $r = 1, ..., R$,
  For $i \in S(r)$, set the load-dependent service rates of station-i in the r-th single-class network to $\mu_{ri}(n_{ri})$= $\tilde{v}_{ri}(n_{ri})$, for n_{ri} = 1,...,N_r
- Step 4. Go to step 1 until convergence of the parameters $\mu_{ri}(n_{ri})$ is achived for a

specified level of tolerance

- Step 5. Calculate the performance parameters of the different classes of customers from the product-form expression of the corresponding single-class networks.

After the convergence of the algorithm, we can calculate all performance parameters of interest. The throughput of the system, the average length of each queue, including the queues of the synchronization stations; and for each manufacturing stage the average work in process, the average number of finished parts as well as the average number of free kanbans can be determined. In the case of kanban systems with external demands we can also calculate the percentage of satisfied demands (the proportion of backordered demands), the average number of backordered demands and the mean waiting time of a backordered demand.

4. OBJECTIVES

In the previous chapter we described how a kanban controlled production system is analyzed as a queuing network. In this dissertation our main objective is to analyze hybrid production systems controlled using kanban control policy. A hybrid system consists of manufacturing and remanufacturing subprocesses each dedicated to its respective operation. The components obtained through these operations are combined to produce the final product.

Remanufactured components are assumed to be processed into as good as new condition. Thus the hybrid production system offers the same version of a component from two different subprocesses. There is a priority here, since remanufactured products are usually cheaper than their new counterparts, the system tends to pull as many remanufactured components as possible in the production sequence.

The main goal is to form a pull type remanufacturing control system joint with manufacturing. To begin with, we analyze any kanban controlled hybrid system analytically and evaluate the important performance measures. The behaviour of demand and return of products are distributed with independent Poisson random variables. The production system is modeled as a closed queuing network with synchronization stations. This queuing network is analyzed with the described methodology in Section 3. The evaluation of required parameters of each queue is done using stochastic calculations. The desired performance parameters are mainly the capacity of the system, throughput of stages, proportion of backordered demands, average number and waiting time of backordered demands, WIP levels of each queue, average number and waiting time of returned products. Our second aim is to calculate the total cost of the hybrid system that comprises inventory holding costs, production costs of stages, penalties for backordered demands, cost of keeping the returned items and disposal cost. For various combinations of kanban sizes of different stages or for a marked performance measure the total cost of the system can be calculated.

We are interested in determining the minimum cost of the hybrid manufacturing system. We propose a search algorithm to find out the optimum kanban combination that gives the minimum cost. According to our rigorous numerical analysis we conjecture that this search procedure gives optimal or near optimal results. Finally, we investigate the relationship among return rate, demand rate, disposal cost and the difference between manufacturing and remanufacturing costs. As a result, the effects of these factors on the design of the hybrid system is presented.

5. PROBLEM DEFINITION and MODEL

In this chapter we define the problem and explain how to evaluate it. The chapter is organized as follows. In Section 5.1 the system is introduced, in Section 5.2 we show how to analyze the model. Section 5.3 deals with the evaluation of the cost function and Section 5.4 describes the kanban assignment procedure. Finally in Section 5.5 numerical examples are illustrated for the validation of the system.

5.1. The System

The manufacturing system we analyze is controlled by kanbans and designed for both remanufacturing and manufacturing materials. The system produces a single product with multiple components and we model it as a multi-class queueing network with synchronization mechanisms (Figure 5.1). K_r number of kanban cards are associated to each stage-r, where r=1,...,9. In this case each stage consist of only a single machine, i.e. a simple kanban system. Machines are depicted as squares and their queues are shown with circles.

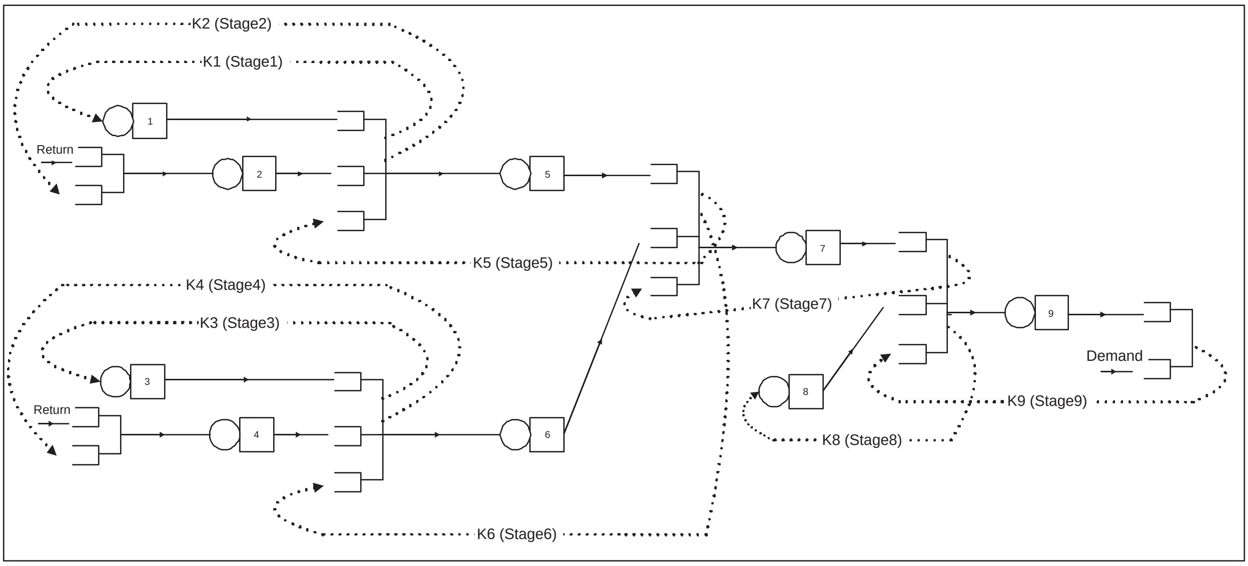

Figure 5.1. The hybrid manufacturing system

The final product is composed of 3 components. Stage-1 and stage-2 are processing component-1 and remanufactured component-1, respectively. Stage-3 and stage-4 are processing component-2 and remanufactured component-2 respectively. Both manufactured and remanufactured components are assumed to be in as good as new conditions. Finally stage-8 is processing component-3.

We assume that there is an infinite supply of raw materials for stages-1,3 and 8. Besides this for stages-2 and 4, the arrival pattern of returned products follow a Poisson distribution with rate γ_j, where $j = 1, 2$ (return rate for component 1 and 2).

When all the operations are completed, the final product is placed into the output buffer of the last station (output buffer of stage-9).

In the manufacturing sequence the components are synchronized at the synchronization stations. In this scenario there are 7 synchronization stations in 4 types.

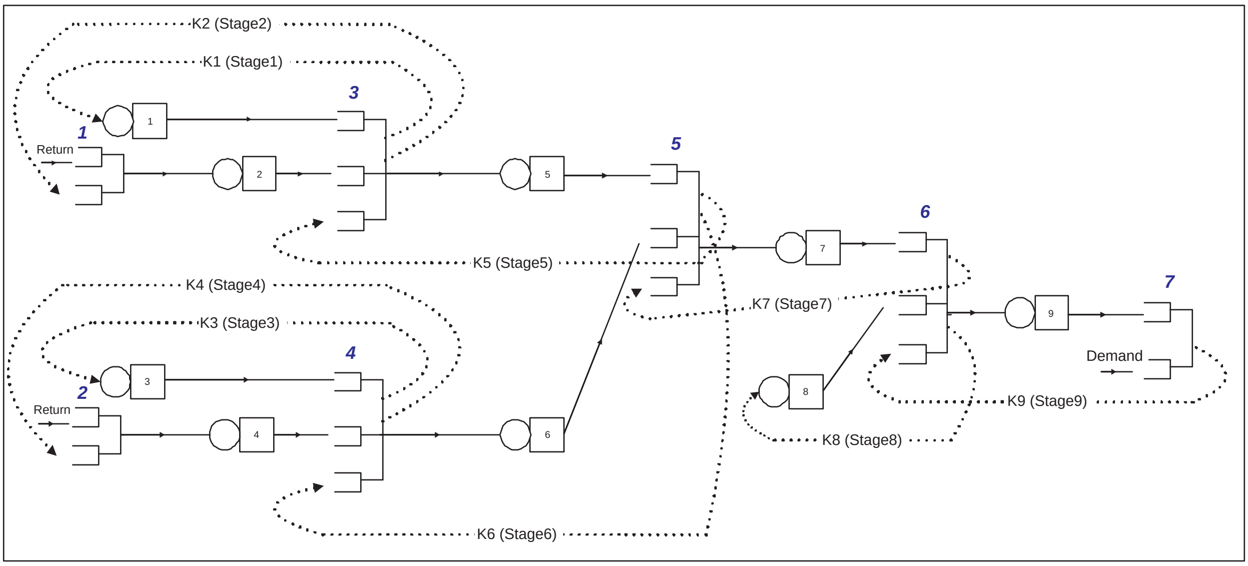

Figure 5.2. Synchronization stations of the system

First two synchronization stations represent the synchronization between the returned products and the demands of stage-2 and stage-4.

In the third synchronization station ($J_{(1,2),5}$) component-1 is synchronized with the corresponding demand. Different from the previous one we assign a priority rule here, that is when the demand of fifth stage arrives to the third synchronization station it is synchronized with the remanufactured component if it is available at that moment, if not the demand is synchronized with a manufactured component of stage-1. If both queues of components are empty the demand waits and is satisfied with the first coming component. Here remanufactured items have priority over manufactured ones and this makes no difference from the substitution point of view because both manufactured and remanufactured components are assumed to be equivalent. According to the studied environment priority can be given to manufactured or remanufactured components. The fourth synchronization station ($J_{(3,4),6}$) behaves in an identical way with the third one but here the operations are realized for component-2.

At the fifth synchronization station ($J_{(5,6),7}$) outputs of stage-5 and 6 (processed component-1 and component-2) are synchronized for assembly and they are processed at stage-7. Similarly at the sixth synchronization station the output of stage-7 (assembled component-1 and 2) and component-3 of stage-8 is synchronized and the product is finalized in stage-9.

The last synchronization station represents the synchronization between finished products of the manufacturing system with the external demands. Demands arrive to the last synchronization station according to a Poisson process with rate Λ. If there is enough supply of finished products, the demand is immediately satisfied, if not the demand is backordered.

This arbitrary topology enables us to analyze the behaviour of two different kinds of inputs, raw materials and returned items. Also the return process of components can be considered separately with different return rates. Besides this the routine assembly process of components and assembly with selection up to priority of parts can be investigated.

5.2. Analysis of the Kanban Control System Model

In the multi-class kanban system the classes are the kanban sets, e.g. kanbans of stage-1 refers class-1, so the number of classes of customers is equal to the number of stages, $R = N$. The total number of class-r customers is equal to the total number of stage-r kanbans, N_r=K_r. Class-r customers visit the set of stations of subnetwork MP_r plus the input and output synchronization stations of that subnetwork. For illustration we first number every station including synchronization stations as depicted in Figure 5.3.

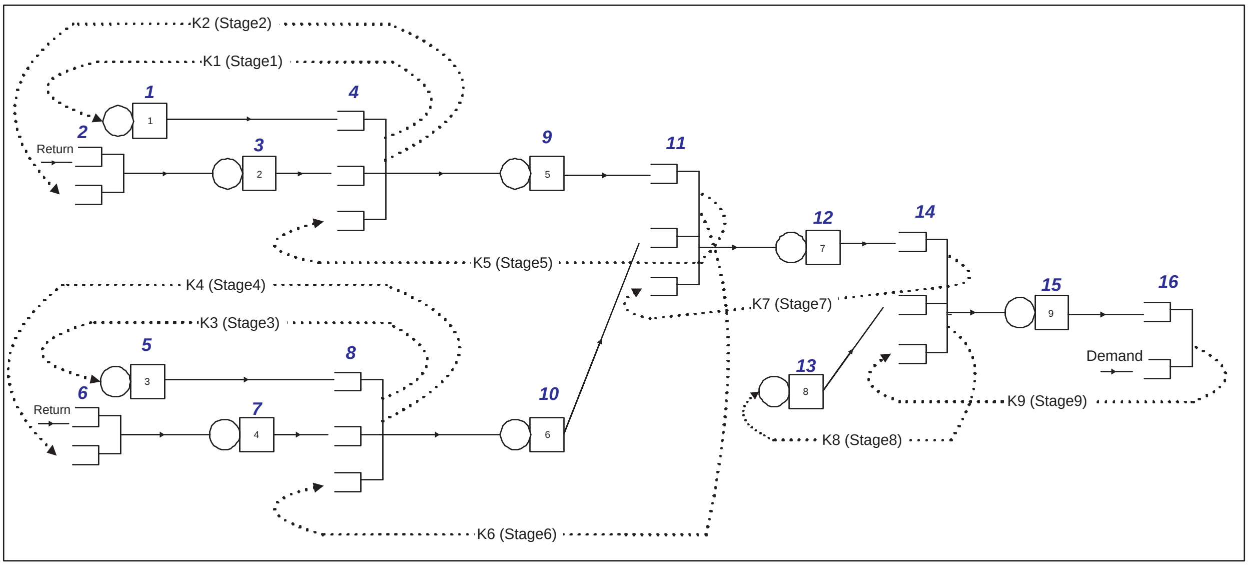

Figure 5.3. Numbering the stations

In this case the set of stations for each manufacturing stage are: $S(1) = \{1, 4\}$, $S(2) = \{2, 3, 4\}$, $S(3) = \{5, 8\}$, $S(4) = \{4, 6, 8\}$, $S(5) = \{4, 9, 11\}$, $S(6) = \{8, 10, 11\}$, $S(7) = \{11, 12, 14\}$, $S(8) = \{13, 14\}$, $S(9) = \{14, 15, 16\}$. The behaviour of each class of customer in the original network is approximated by a single-class product form network with the method described in Chapter 3. Since there are N classes of customers, there are N single class networks. For this model these single-class networks are illustrated in Figure 5.4.

In the next step we have to determine the parameters of these single-class net-

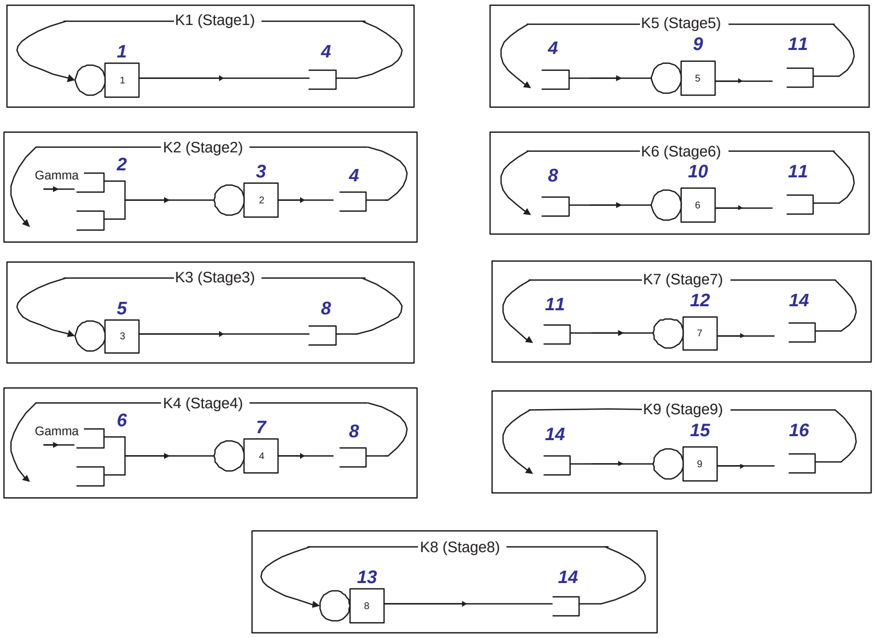

Figure 5.4. Single-class product-form networks

works. To this end, each station of the original network has to be analyzed in isolation with state dependent Markovian processes one for each class that visits the station.

For a single class service station (stations visited only by a single class of customers) the analysis in isolation corresponds to a $\lambda(n)/M/1/N$ queue, for stations 1,3,5,7,9,10,12,13 and 15. The classical methods can be used to analyze this queue and obtain marginal probabilities $\tilde{P}_{ri}(n_{ri})$ required by the overall method. The state structures of these queues are simple birth-death processes and the equations that calculate the steady state probabilities are given below.

$$P_i(0) = (1 + \sum_{n=1}^{K_i} \prod_{k=1}^{n} \frac{\lambda_i(k-1)}{\mu_i(k)})^{-1}$$

$$P_i(n) = (\prod_{k=1}^{n} \frac{\lambda_i(k-1)}{\mu_i(k)}))P_i(0)$$

The first two synchronization stations of Figure 5.2 are associated with the returned items, where they follow a Poisson Process with rate γ_j, $j = 1, 2$. The associated Markov chain is linear, namely its resolution is like a birth death process. Hence the probabilities can be derived without involving any numerical technique. Notice that if there is a capacity constraint for the returned items buffer, the Markov chain has a finite dimension; if not the MC has infinite dimension. The state diagram of this kind of synchronization station with finite capacity can be seen in Figure 5.5; the state of this Markov Chain is (n_r, n_k), where n_r is the number of returned items and the n_k is the number of kanbans currently present at the synchronization station.γ_j is shown as γ for simplicity. And also the equations that are used in the calculations of the stationary probabilities can be seen below. The structure of the last synchronizaiton is similar with these ones and the infinite case will be seen there.

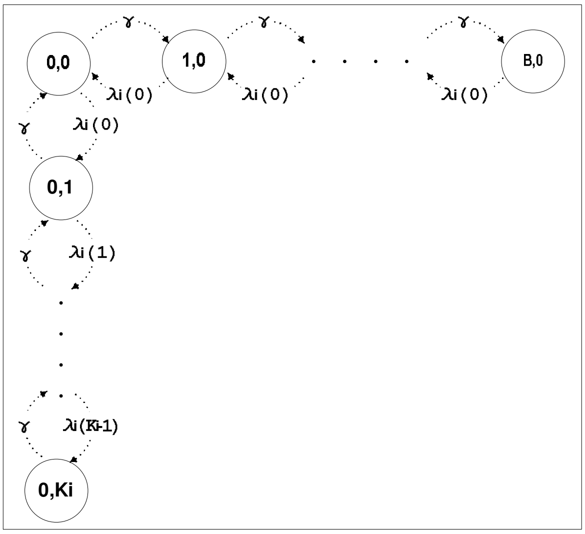

Figure 5.5. State diagram of the first synchronization station

$$
\begin{aligned}
P00_i &= (\sum_{n=1}^{Ki} \frac{\prod_{j=0}^{n-1} \lambda_i[j]}{\gamma^n} + \frac{(\frac{\gamma}{\lambda_i[0]})^{B+1} - 1}{\frac{\gamma}{\lambda_i[0]} - 1})^{-1} \\
P_i(0) &= P00_i(\frac{(\frac{\gamma}{\lambda_i[0]})^{B+1} - 1}{\frac{\gamma}{\lambda_i[0]} - 1}) \\
P_i(n) &= P00_i \frac{\prod_{j=0}^{n-1} \lambda_i(j)}{\gamma^n}
\end{aligned}
$$

We can denote the third synchronization station in Figure (5.2), namely station-4 as $J_{(1,2),5}$, which is synchronizing parts out of first two stages with the stage-5 kanban. It has 3 upstream queues; P_1 represents the manufactured component1, P_2 represents the remanufactured component1 and D_5 represents a stage-5 kanban. In this case remanufactured components have a priority, namely if both manufactured and remanufactured components are available, remanufactured components are consumed first. For example when the demand arrives to queue D_5, assume there is at least one entity in each of the queues P_1 and P_2, then the remanufactured component-1 in queue P_2 is synchronized with the demand and only stage-2 is released. The figure and the state diagram of the synchronization station can be seen in the Figures 5.6 and 5.7 respectively.

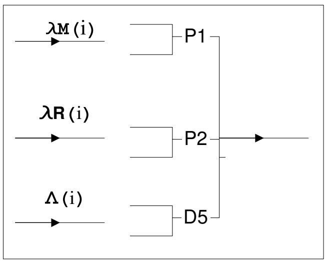

Figure 5.6. The third synchronization station, $J_{(1,2),5}$

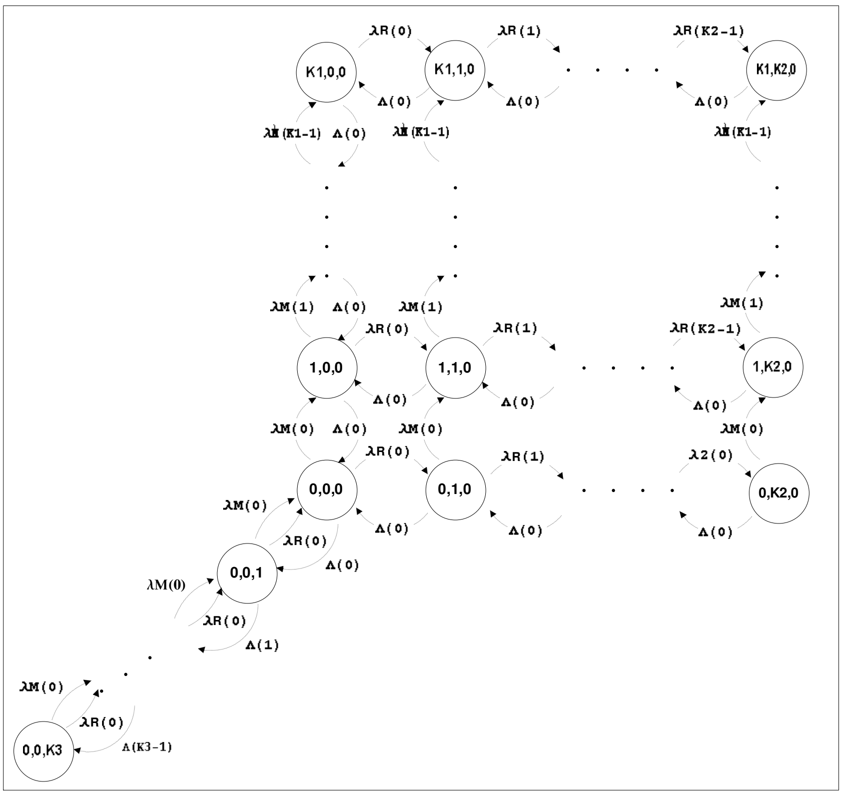

Figure 5.7. State diagram of the third synchronization station

The state structure of this synchronization station is not simple to solve this Markov chain in terms of equations; so we have to use numerical methods. We first write the transition rate matrix Q. Then we use the SSOR algorithm described in section 3.2.2.2 to obtain the steady state probability vector and then marginal probabilities of each queue of this synchronization station easily. The fourth synchronization station is identical of the third synchronization station but in this case operations are realized for component-2.

The fifth synchronization station in Figure 5.2 is $J_{(5,6),7}$. Parts coming from stage-5 and stage-6 are synchronized here and then assembled at the seventh machine. Similar with the previous one we use the same numerical method to solve the underlying markov chain. SSOR algorithm is again used to obtain the matrix geometric solution. The behavior of the sixth synchronization station is the same.

The last synchronization station is associated with external resources $J_{N,N+1}$. Demand arrivals follow a Poisson Process with rate Λ. The state structure is similar with the first two synchronization stations, but here the number of backorders is not limited. The state diagram of the last synchronization station can be seen in Figure 5.8; the state of this Markov Chain is (n_p, n_b), where n_p is the number of finished products and the n_b is the number of backordered demands currently present at the synchronization station. The equations for the steady state probabilities are given below.

$$\begin{aligned}
P00_i &= ((1/(1-(\frac{\Lambda}{\lambda_i(0)}))) + \sum_{n=1}^{K}(\frac{\prod_{m=0}^{n-1}\lambda_i[m]}{\Lambda^n})) \\
P_i(0) &= P00_i(1/(1-(\frac{\Lambda}{\lambda_i(0)}))) \\
P_i(n) &= P00_i\frac{\prod_{m=0}^{n-1}\lambda_i(m)}{\Lambda^n}
\end{aligned}$$

These computed probabilities are used in the algorithm and after the convergence of the algorithm the desired performance measures can be obtained.

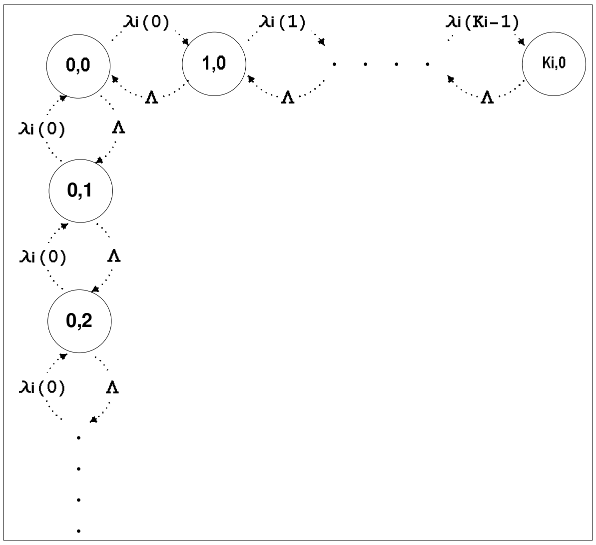

Figure 5.8. State diagram of the last synchronization station

5.3. The Cost Function

The total cost function of the system is composed of three main parts. The first part is associated with the production costs of each stage that is obtained by multiplying the total operation cost of a particular stage by its throughput. The production cost of a particular stage involves whole processing costs of machines and all other operation costs i.e. cost of consumable parts/liquids etc.

The second part is the holding cost, which is the associated cost to keep the inventory i.e. WIP, returned items, finished products. It is calculated by multiplying the average number of items in each queue (buffer) in the steady state with the corresponding holding cost rate.

Finally the third part is the backorder cost, which is associated with the cost of demands that are waiting for products at the output buffer of the system. It is simply obtained by multiplying the average number of backorders at steady state with the backorder cost for a unit.

Thus the total cost is simply given as,

Total Cost= Production Cost + Holding Cost + Backorder Cost

This total cost is for the case with no disposals of returned items. If the disposal is allowed a disposal cost has to be added. The disposal cost is determined in a similar way with the production cost, based on the throughput of the disposed items. For the case with disposal when the rate of returned items is γ, the rate of the returned parts that enters the system is defined as $\gamma_{effective}$. Clearly the rate of disposed items is $(\gamma - \gamma_{effective})$ and the total disposal cost is obtained by multiplying this rate by the cost of disposed items.

The determination of holding cost rates requires special attention for the manufacturing systems joint with remanufacturing. For the production systems working only with forward logistics; using the traditional methods for determining the holding cost rates used in the average cost (AC) models gives near optimal results from a discounted cash flow (DCF) point of view. However for the systems with remanufacturing the situation is not the same and the AC approach stands or fails with the right choice of the holding cost parameters. When the choice of the holding cost parameter for returned items, h_n, holding cost rate for remanufactured items h_R and holding cost rate for manufactured items h_M are as;

$$h_n = r * (C_M - C_R) \tag{5.1}$$

$$h_R = h_M = r * C_M \tag{5.2}$$

where r is the discount rate, C_M and C_R are cost for manufacturing and remanufacturing respectively, the average cost model gives near optimal results [18, 19]. For the determination of holding cost parameters for manufacturing and remanufacturing stages we adopt this approach. For the rest of the stages we use the traditional approach to determine the holding cost parameters, namely multiplying the interest rate (or discount rate) by the marginal cost for a particular stage for processing a part.

In the kanban system the control parameters are the number of kanbans. All the performance measures like cost, average WIP levels, average backorder etc. are functions of the number of kanbans. The cost function of the model in Figure 5.1 is stated below and this relation can be seen explicitly.

$$\begin{aligned} Z_{(K1,K2,K3,K4,K5,K6,K7,K8,K9)} &= WIP_{Qr,i(K1,K2,K3,K4,K5,K6,K7,K8,K9)} * h_{r,i} \\ &+ TH_{r(K1,K2,K3,K4,K5,K6,K7,K8,K9)} * CS_r \\ &+ Q_{D(K1,K2,K3,K4,K5,K6,K7,K8,K9)} * b \\ &+ QR_{1(K1,K2,K3,K4,K5,K6,K7,K8,K9)} * h_n \\ &+ QR_{2(K1,K2,K3,K4,K5,K6,K7,K8,K9)}) * h_n \end{aligned} \quad (5.3)$$

where r=1,...,9, $Q_{r,i}$ represents the queue of station i in stage r, $WIP_{Qr,i}$ denotes the average work in process level of each queue and $h_{r,i}$ is the related holding cost for that queue, TH_r is the throughput of stage-r, CS_r is the total production cost of stage-r. Q_D and QR_j represents the average number of backorders and average number in the returned items buffer (j=1 for stage-2, j=2 for stage-4) respectively; b is the backorder cost and h_n is the holding cost for returned items.

To give a clear sight the cost of stage-2 and stage-9 are given below. Again all the terms are functions of K_r, r=1,...,9 but for notational simplicity this is not shown here.

$$\begin{aligned} Z_{STAGE2} &= (QR_1 + NQ_{2,3}) * h_n + NQ_{2,4} * h_R + TH_2 * C_{S2} \\ Z_{STAGE9} &= NQ_{9,15} * h_{9,15} + NQ_{9,16} * h_{9,16} + TH_9 * C_{S9} + Q_D * b \end{aligned} \quad (5.4)$$

In the next section we state a heuristic search method that gives the minimum cost under the given situation and convexity of the cost function is discussed there.

5.4. Kanban Assignment

After writing the cost function including WIP, production and backorder costs, we can calculate the total cost (Z) for various combinations of kanbans. The number of combinations is naturally excessive so we propose the following search method to obtain the minimum cost. The general view of the method is given below and the details are given throughout this section.

The Search Method:

Step 1: Find a $K_{initial}$ value, which is same for all stages that minimizes the total cost

Step 2: For stages 5 to 9 keep $K_{initial}$ constant and determine the remanufacturing and manufacturing kanbans, namely K_M and K_R.

Step 3: Using K_M and K_R find the best combination that gives the minimum cost.

Step 4: Check K_M and K_R

In the following the search method is explained in detail and the convexity of the cost function is discussed. In the first step to find an initial K, we keep the number of kanbans for each stage equal and the cost is calculated beginning from K^0, i.e. $Z(K^0)$ where K^0 is the minimum value that satisfies the condition $(\Lambda/\lambda(K^0) < 1)$ and $\lambda P(K)$ is the throughput of the whole system when all the stages have K kanbans. Then K is increased by one at every iteration. When $Z(K+1) - Z(K) > 0$, we set K as $K_{initial}$ for each stage and go to step 2.

In the second step, the aim is to determine the manufacturing and remanufacturing kanbans, K_M and K_R. In this system stage-1 and stage-2 produces manufactured and remanufactured component-1 respectively and we assume that both manufactured and remanufactured products are in an as good as new condition. Due to this fact we may see stage-1 and stage-2 as a cell producing component-1. So we assign K kanbans to this cell and then share these kanbans as manufacturing and remanufacturing kanbans, i.e. $K = K_M + K_R$. We start with the possible smallest K (like K^0) and then we begin to increase it. At some point we reach the minimum cost, and optimal values of

K_M and K_R are determined. After the determination of K_M and K_R, set $K_R^* = K_R$ and $K_M^* = K_M$ and go to step 3. After step-2 the cost function is denoted with these parameters as $Z(K_M^*, K_R^*, K_M^*, K_R^*, K_{initial}, ..., K_{initial})$.

In the third step, after determining $K_{initial}$, K_R^* and K_M^* we keep K_R^* and K_M^* constant and we check the neighboring values of $K_{initial}$ for each stage. To determine the number of kanbans for stage-r, we calculate the total cost for K_r= $K_{initial} - 1$ and K_i= $K_{initial} + 1$ while keeping the number of kanbans for the rest of the stages constant. Three situations may occur where $Z_r(K)$ denotes the cost when there is only change in stage-r kanbans while keeping the rest of them constant;

- If $Z_r(K_{initial})$ is smaller than both $Z_r(K_{initial} - 1)$ and $Z_r(K_{initial} + 1)$ then set $K_r^* = K_{initial}$.

- If $Z_r(K_{initial} + 1)$ gives a smaller value we check $Z_r({}_{Kinitial}{+}2)$ and if the total cost is still getting smaller we check $Z_r(K_{initial} + 3)$ and goes on. At some point $Z_r(K_{initial} + x) < Z_r(K_{initial} + x + 1)$ so; $K_{initial} + x$ gives the minimum total cost.

- If $Z_r(K_{initial} - 1)$ gives a smaller value, similarly at some point $Z_r(K_{initial} - x) < Z_r(K_{initial} - (x + 1))$, so $K_{initial} - x$ gives the minimum total cost.

After this search we set $K_r^* = K_{initial} - x$ or $K_{initial} + x$. Step three is followed for every stage to get the optimum values for K_r^*, r=5...9.

After the third step with newly determined kanbans for stage-r, r=5...9, we check K_M^* and K_R^* values whether if there is a change or not. And finally minimum cost is obtained with these parameters.

$$Z(K_{M1}^*, K_{R1}^*, K_{M2}^*, K_{R2}^*, K_5^*, K_6^*, K_7^*, K_8^*, K_9^*) = Z^*$$

For the validation of the search procedure we compiled an experiment set using the L(3^4) orthogonal array [30] as given in Table 5.1. Since we are interested in determining the minimum cost, the parameters that have an important effect on the cost function are selected. Buffer size for returned items, return rate of used items, backorder cost and processing time of machines are chosen in three levels. The processing time of all machines are equal. The demand rate is fixed at 0.8 in all of the experiments.

Table 5.1. Experiment set for the validaton of search method

EX	B	γ	b	μ_i
1	inf	0.1	3.2	1.5
2	inf	0.4	6.4	1
3	inf	0.7	12.8	0.5
4	4	0.1	6.4	0.5
5	4	0.4	12.8	1.5
6	4	0.7	3.2	1
7	8	0.1	12.8	1
8	8	0.4	3.2	0.5
9	8	0.7	6.4	1.5

We solve this experiment set according to the proposed search procedure. Afterwards we make an exhaustive search which covers all the possible kanban combinations and compare these results. Out of 9 experiments the search procedure finds 5 exact results and 4 near optimal results with a slight difference. The results and the relevant kanban combinations are summarized in Table 5.2 and in Figure 5.9.

After describing the search method the convexity of the cost function is investigated. In the first step of the algorithm, we want to optimize the system by assigning the same $K_{initial}$ value for every stage that minimizes the long term average cost associated with production cost, the inventories for WIP, returned items and finished goods, and backordered demands. Our parameters like cost and WIP are functions of all $K_{initial}$'s but as they are equal for simplicity we denote them with single parameter as $Z(K_{initial})$.

Table 5.2. Total costs

	Search Method		Exhaustive Search		Relative
	Z_{Best}	Kanban Set	Z_{Opt}	Kanban Set	Error (%)
EX1	43.388	(3,1,3,1,7,7,9,9,7)	43.130	(2,1,2,1,7,7,10,10,6)	0.6
EX2	23.583	(1,2,1,2,3,3,4,4,4)	23.583	(1,2,1,2,3,3,4,4,4)	0
EX3	19.762	(1,4,1,4,2,2,2,2,3)	19.575	(1,4,1,4,1,1,3,3,2)	0.9
EX4	16.916	(1,1,1,1,2,2,2,2,2)	16.916	(1,1,1,1,2,2,2,2,2)	0
EX5	52.020	(1,10,1,10,8,8,10,10,12)	51.994	(1,10,1,10,7,7,11,11,12)	0.05
EX6	22.639	(1,8,1,8,2,2,4,4,3)	22.639	(1,8,1,8,2,2,4,4,3)	0
EX7	28.529	(1,1,1,1,4,4,4,4,5)	28.529	(1,1,1,1,4,4,4,4,5)	0
EX8	14.476	(1,1,1,1,1,1,2,2,1)	14.476	(1,1,1,1,1,1,2,2,1)	0
EX9	45.781	(1,8,1,8,7,7,10,10,8)	45.381	(1,9,1,9,6,6,10,10,9)	0.9

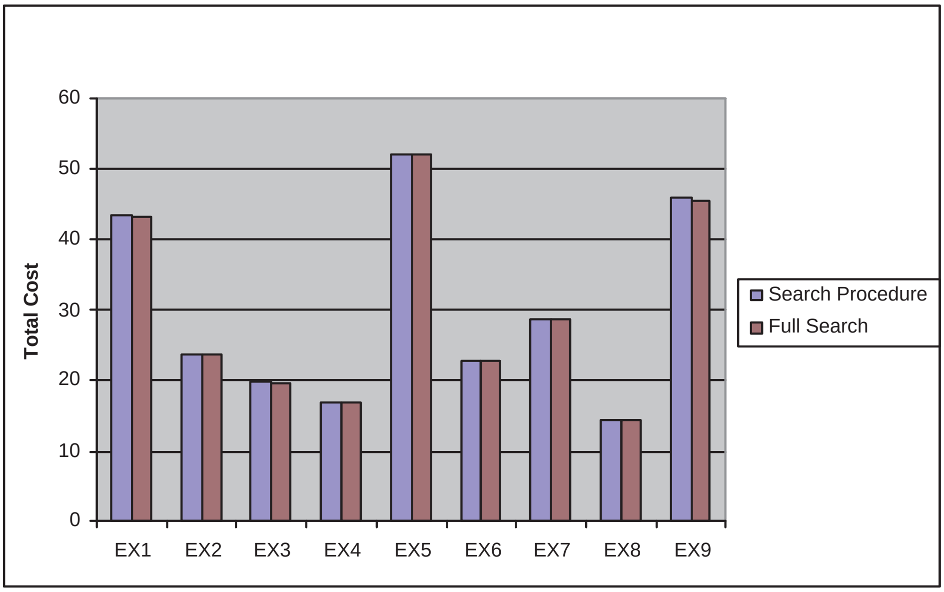

Figure 5.9. Search procedure vs Exhaustive search

$$
\begin{aligned}
Z(K_{initial}) &= WIP_{Sr} * h_r + TH_r(K_{initial}) * C_{Sr} + Q_D(K_{initial}) * b + Q_{R1}(K_{initial}) * h_n \\
&+ Q_{R2}(K_{initial}) * h_n \qquad (5.5)
\end{aligned}
$$

Notice that here we write the WIP levels of stages, not queues separately, where WIP_{Sr} represents the WIP level of stage-r. The relation is given below.

$$WIP_{Sr} = \sum_{i=1}^{X} WIP_{Qr,i}$$

where X is the total number of queues in stage-r.

When the number of kanban cards is K_{inital}, we know that $WIP_{Sr}(K_{initial}) = K_{initial}$ as a property of the kanban system. As we have N stages, when we rearrange the cost function, holding cost and penalty parameters, the optimization problem turns into;

$$
\begin{aligned}
Z(K_{initial}) &= N * K_{initial} + TH_r(K_{initial}) * C'_{Sr} + Q_D(K_{initial}) * b' + Q_{R1}(K_{initial}) * h'_n \\
&+ Q_{R2}(K_{initial}) * h'_n \qquad (5.6)
\end{aligned}
$$

The expected number of backorders $Q_D(K_{initial})$ must be greater or equal to zero. Besides this $Q_D(K_{initial})$ is proportional to the average waiting time that a demand waits for a finished product, $W_B(K_{initial})$, with respect to Little's Law, $W_B(K_{initial}) = \Lambda{*}Q_D(K_{initial})$. When we assign $(K_{initial} + 1)$ kanbans more finished products accumulate in the output buffer and more demands are satisfied from inventory so $W_B(K_{initial} + 1)$ is smaller than $W_B(K_{initial})$ obviously. Consequently $Q_D(K_{initial} + 1)$ is smaller than $Q_D(K_{initial})$ and thus $Q_D(K_{initial})$ is a decreasing function of $K_{initial}$.

Similarly $Q_R(K_{initial})$ is a decreasing function of $K_{initial}$ too. Increasing the number of kanbans decreases the waiting time of returned items, i.e.$W_R(K_{initial} + 1) < W_R(K_{initial})$. So we can both see $Q_R(K_{initial}))$ and $Q_D(K_{initial})$ as decreasing functions of $K_{initial}$.

Production cost of the system is directly related with throughputs and throughputs are increasing functions of number of kanbans. When there are no kanban cards the throughput is zero, it increases with the increase of kanban cards and finally converges to the system's maximum throughput.

Tardif and Maaseidvaag deduced a property for a single stage kanban system [10]. With the statements given above, the first step of our algorithm is based on this property given below.

Property 5.1 *There exists a* $K^*_{initial}$ *such that* $Z(K^*_{initial})$ *is minimized where*

- $Z(K_{initial}) \geq Z(K_{initial} + 1)$ *for* $K^0_{initial} \leq K_{initial} < K^*_{initial}$ *where* $K^0_{initial}$ *is the smallest* $K_{initial}$ *such that* $\Lambda/\lambda(K_{initial}) < 1$ *and*

- $Z(K_{initial}) \leq Z(K_{initial} + 1)$ *for* $K_{initial} \geq K^*_{initial}$

Proof:

$$\begin{aligned} Z(K_{initial}) &= \Big[N * K_{initial} + TH_r(K_{initial}) * C'_{Si}\Big] \\ &+ \Big[Q_D(K_{initial}) * b' + Q_{R1}(K_{initial}) * h'_n + Q_{R2}(K_{initial}) * h'_n\Big] \end{aligned}$$

Recall that the second part of the sum is a decreasing function of $K_{initial}$, which approaches 0 as $K_{initial}$ increases. Therefore $Z(K_{initial})$ must increase at some point. The minimum value of $Z(K_{initial})$ is found by increasing $K_{initial}$ until $Z(K_{initial} + 1) - Z(K_{initial}) \geq 0.\square$

Throughout this section we give numerical examples to support the properties according to the experiment set given in Table 5.1. All numerical examples hold the properties without exception. For Property 5.1, beginning from K^0 to $K = 15$ the total costs of experiments are given in the Figure 5.10 and 5.11 . From the graphics it can be seen that there exists only one $K_{initial}$ value that minimizes the total cost.

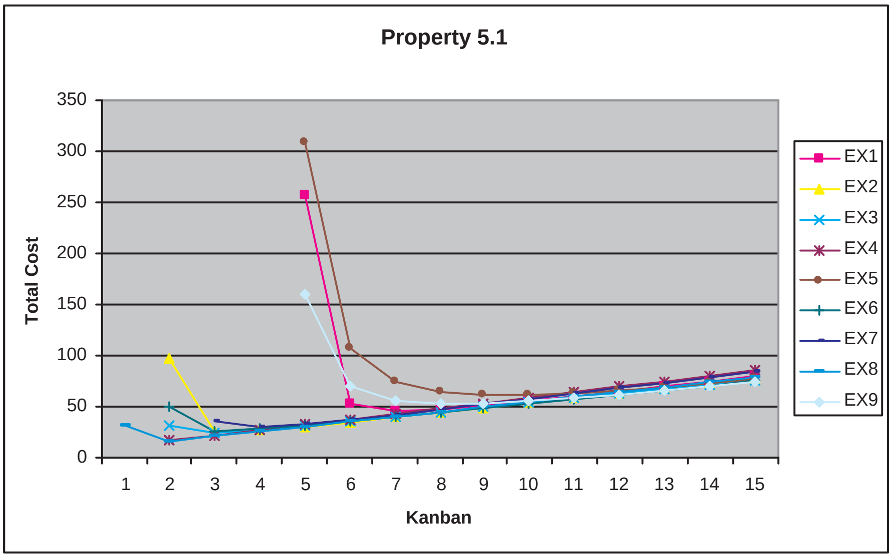

Figure 5.10. Numerical examples for Property 5.1

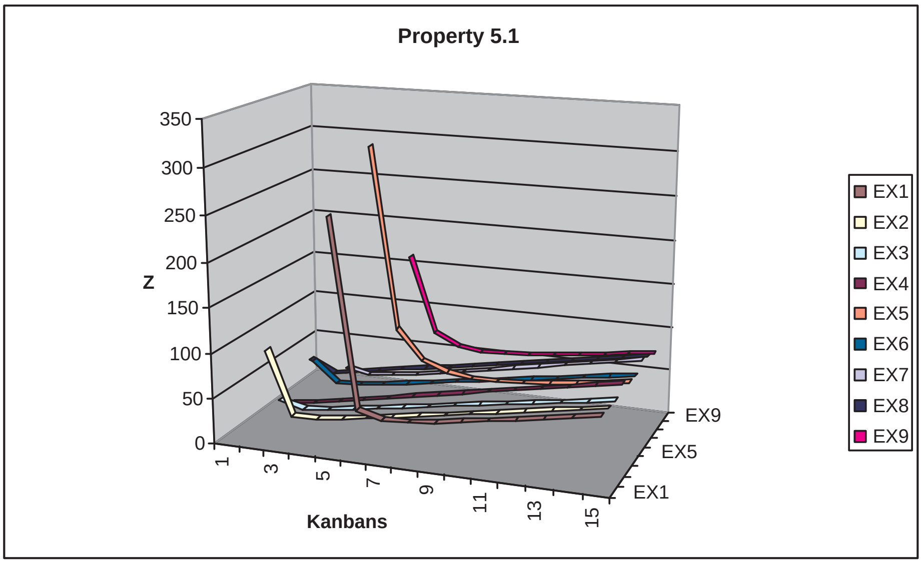

Figure 5.11. Numerical examples for Property 5.1

In the second step of the algorithm we want to show that by changing the number of manufacturing K_M (in the example K_1, K_3) and remanufacturing kanbans K_R

(K_2, K_4) and keeping the number of kanbans for the rest of the stages constant $(K^*_{initial})$, there exists a minimum value for Z for the combination of K_1, K_2, K_4, and K_4.

We know that Z is a function of parameters K_r, where r=1,...,9. As mentioned above, we keep the number of kanbans for stages S_5 to S_9 constant at $K^*_{initial}$ and only deal with manufacturing and remanufacturing kanbans. So we can analyze Z as a function of K_1, K_2, K_4, and K_4. Besides this in this system we have two stages for manufacturing and two stages for remanufacturing, stages 1 and 2 operates in the same way with stages 3 and 4. For simplicity we analyze the cost function, as a function of parameters K_1 and K_2 because if we can show the convexity of the cost function as parameters of K_1 and K_2 the result is identical for the parameters K_3 and K_4. So we denote the cost function and the other performance measures as functions of K_1 and K_2 as $Z(K_1, K_2), WIP_{S1}(K_1, K_2), WIP_{S2}(K_1, K_2), Q_D(K_1, K_2)$ and $Q_R(K_1, K_2)$. Since manufacturing is done at stage 1 and remanufacturing is done at stage 2, we denote K_1 as K_M and K_2 as K_R.

In our system, since manufacturing and remanufacturing stages take out same output, they are alternative stages of each other. Taking this into consideration we treat stage 1 and 2 as a single stage that produces component 1. And we assign K kanbans for these two stages and then we share these kanbans as remanufacturing and manufacturing kanbans. K=$K_M + K_R$. With this sharing we can use the economical opportunities of remanufacturing while keeping the manufacturing stage alive. So we avoid starvation of system due to the unreliable nature of return process.

Now our objective is to minimize Z with respect to the control parameter K and K_R, namely $Z(K, K_R), (K_M = K - K_R)$. The steady state probabilities of the system depend on demand distribution, return distribution and service distribution. In order to simplify our approach, we first assume that K_R is fixed to its local optimal value K^*_R for each K considered. Then the optimization problem reduces to a problem similar to that discussed by Tardif and Maaseidvaag[10], where we can look at each performance measure individually to analyze the behavior of the expected total cost function.

We first look at the expected backorders, $Q_D(K, K_R^*)$. Here as the number of total kanbans increase, the systems maintains higher average on hand inventory for component-1 so the waiting time of an arriving demand for component-1 decreases. The higher on hand inventory for component-1 keeps the quicker response to the next stage and as a result of this $Q_D(K, K_R^*)$ decreases as K increases; $Q_D(K, K_R^*)$ is a decreasing function of K. If the number of kanbans is lower than K^0, where $\Lambda/\lambda P(K^0) < 1$, the backorders accumulate faster than the output of the system so in the long run $Q_D(K, K_R^*)$ converges to infinity. If the number of kanbans is K^0, $Q_D(K, K_R^*)$ has a finite value. And when the number of kanbans increase significantly $Q_D(K, K_R^*)$ converges to its minimum value correlated with the speed of the fallowing stages when they have $K_{initial}^*$ kanbans. Therefore we can argue that $Q_D(K, K_R^*)$ is an asymptotic function of K.

When we deal with the average WIP levels $WIP_M(K, K_R^*)$ and $WIP_R(K, K_R^*)$, we see that each addition to the total kanban size K, increases either K_R^* or K_M^* (K_M^*=K - K_R^*) by one. This results in an increase in $WIP_M(K, K_R^*)$ or $WIP_R(K, K_R^*)$, respectively. Hence $WIP_M(K, K_R^*)$ and $WIP_R(K, K_R^*)$ are monotonically increasing functions of K. Therefore $WIP_{S1} + WIP_{S2}$ sum is a monotonically increasing function of K. And clearly $TH_M(K, K_R^*)$ and $TH_R(K, K_R^*)$ are increasing functions of K and converges to the maximum throughput of the corresponding stages with the increase of K.

Finally we deal with the term $Q_R(K, K_R^*)$ which represents the average number in the buffer for returned items. If there is an increase in K_R^* it is obvious that remanufacturing stage pulls more parts from returned items buffer and $Q_R(K, K_R^*)$ decreases. Increase of K_R^* is dependent to K, we know that increasing K increases either K_R^* or K_M^*. Therefore $Q_R(K, K_R^*))$ is a monotonically decreasing function of K.

Hence, in the light of these arguments we can state the following property.

Property 5.2 *Let $K_R = K_R^*$, where K_R^* is the local optimal value for each K value. Then there exist a K^* such that $Z(K, K_R^*)$ is minimized where*

- $Z(K, K_R^*) \geq Z(K+1, K_R^*)$, *for* $max\{K^0, K_R^*\} \leq K < K^*$ *where* K^0 *is the smallest* K *that satisfies* $\Lambda/\lambda(K_{initial}) < 1$

- $Z(K, K_R^*) \leq Z(K+1, K_R^*)$ *for* $K \geq K^*$

Numerical examples for Property 5.2 are given for two different values of remanufacturing kanbans; for K_R^* the results are shown at Figure 5.12 and for an arbitrary K_R they are given at Figure 5.13. In both cases there exists a K^* that gives the minimum total cost. In general, when K_M is equal to one, the local minimum costs are incurred so the first points in the graphics show the minimum cost values.

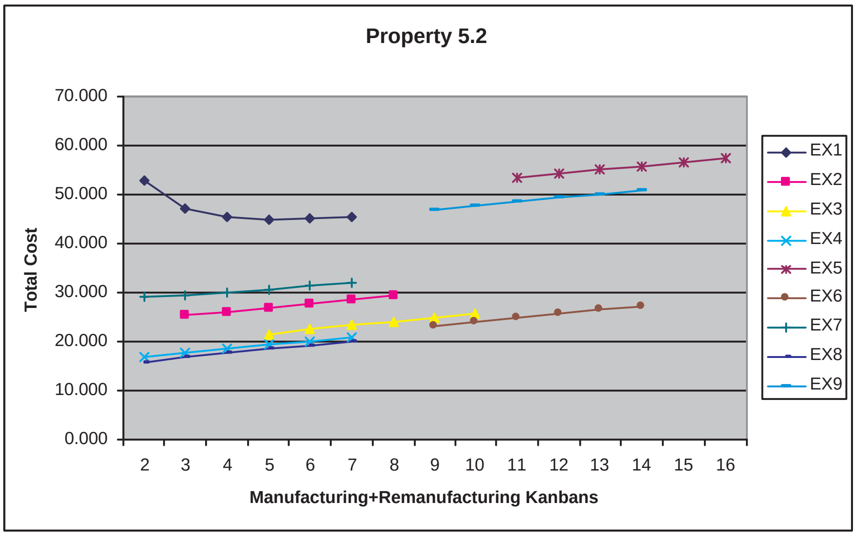

Figure 5.12. Property 5.2 for K_R^*

Similarly when K is fixed to a value $\kappa, \kappa \geq K^0$, $WIP_M(\kappa, K_R)$ is a monotonically increasing function of $K_M = K - K_R$ and $WIP_R(\kappa, K_R)$ is a monotonically increasing function of K_R. The relationship between the choice of K_R and the on hand inventory for component-1 is highly dependent on the production throughputs of the manufacturing and remanufacturing stages. As the throughput of the production process for component-1 is the combination of manufacturing and remanufacturing processes, by increasing their throughputs we increase the total throughput for component-1. As the sum of manufacturing and remanufacturing throughputs increase when the demand is constant average on hand inventory for component 1 increases so a demand's waiting

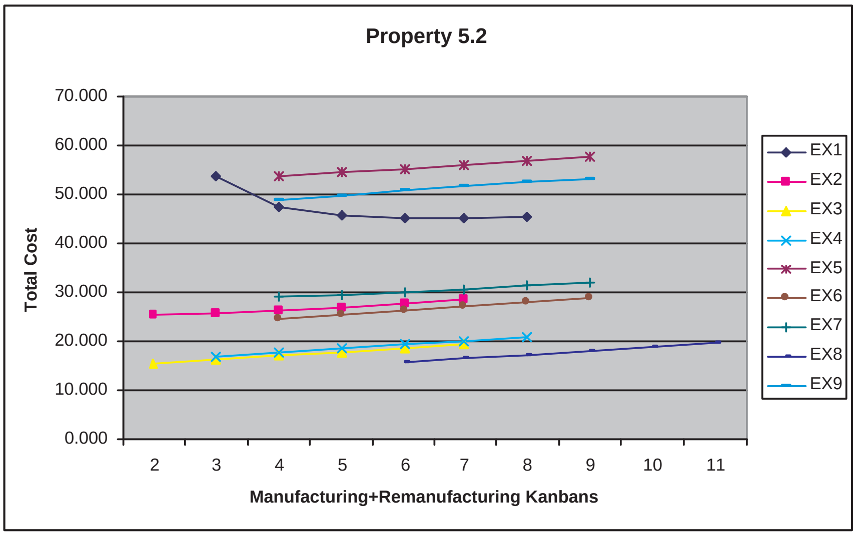

Figure 5.13. Property 5.2 for an arbitrary K_R

time for component 1 decreases. For fixed κ when we increase the throughput of remanufacturing process by increasing K_R, we decrease the throughput of manufacturing process. Hence there exists at least one $K_R, 0 \leq K_R < \kappa$ value that provides the best combination of manufacturing and remanufacturing throughputs that maximizes on hand inventory level for component-1. Since an increase in on hand inventory results in a decrease $Q_D(\kappa, K_R)$, we can state that for fixed κ there exists a K_R that minimizes the backorder level.

Similarly production costs of stages are determined up to their throughputs as mentioned above and increasing K_R increases operation costs for remanufacturing and decreases operation costs for manufacturing and decreasing K_R results vice versa.

In additon we know that $Q_R(\kappa, K_R)$ is a decreasing function of K_R, as K_R increases $Q_R(\kappa, K_R)$ decreases. By rearranging the cost function according to (κ, K_R) we get,

$$\begin{aligned} Z(\kappa, K_R) &= WIP_M(\kappa, K_R) * h_M + WIP_R(\kappa, K_R) * h_R + TH_M(\kappa, K_R) * C_{SM} \\ &+ TH_R(\kappa, K_R) * C_{SR} + WIP_{Si} * h_i + TH_i * C_{Si} + Q_D(\kappa, K_R) * b \\ &+ Q_R(\kappa, K_R) * h_n \end{aligned} \quad (5.7)$$

The results of our computations comply with the arguments above. Taking these points into consideration, for fixed κ, we can say that there is an optimal K_R that minimizes $Z(\kappa, K_R)$, and based on this argument we conjecture the following property.

Property 5.3 *Let* $K = \kappa$. *Then there exists a* K_R^* *such that* $Z(\kappa, K_R^*)$ *is the local minimum for* $K_R \in \{0, 1, ..., \kappa\}$

Numerical examples for Property 5.3 are summarized in Figures 5.14 and 5.15 where total number of kanbans (κ) is 5 and 10 respectively. In both cases there exists a $K_R{*}^*$ that minimizes the total cost.

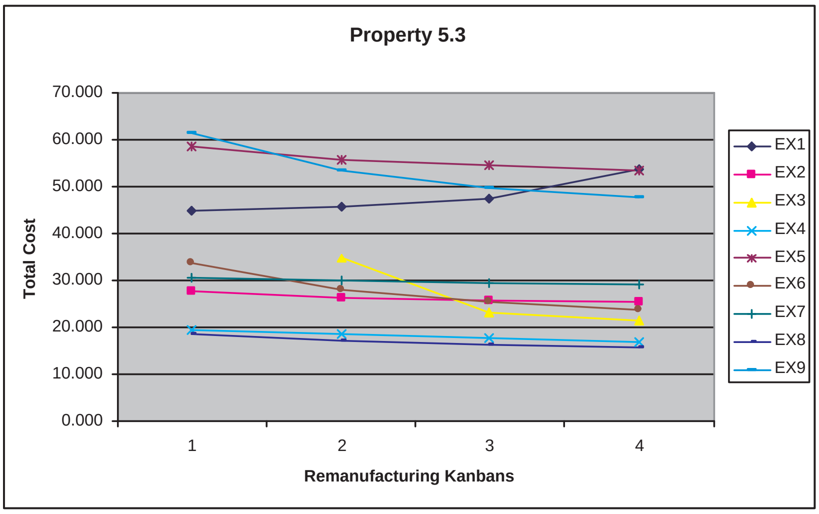

Figure 5.14. Property 5.3 for κ=5

5.5. Validation of the System

For the validation of the system described in section 5.1 we compiled an experiment set using $L(2^1 3^7)$ orthogonal array [30] as given in Table 5.3 below. Using this experiment set we evaluate the system both with the analytical model and simulation model. Then we compare the results of these models. Simulation results are obtained from Arena 7.0. Here, the values of the parameters are selected in such a way that they reflect the behaviour of the system in normal and extreme cases, such as high and low values for return rate, processing times of machines, infinite and capacitated buffer

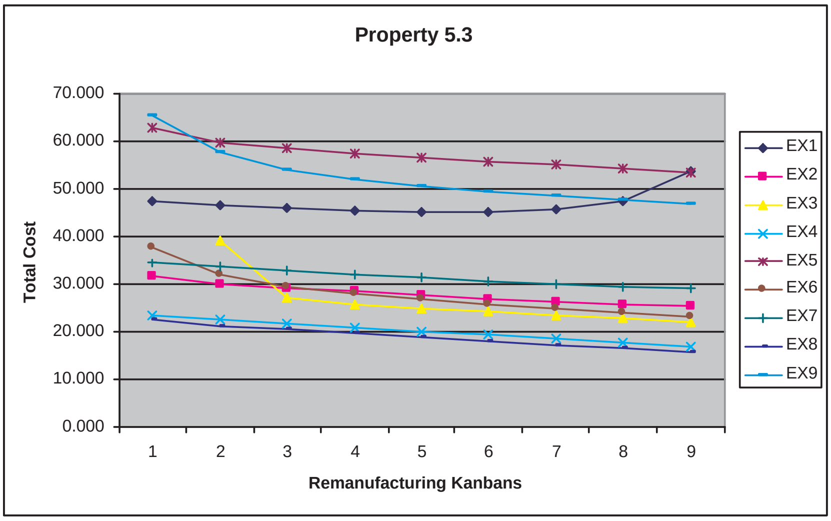

Figure 5.15. Property 5.3 for κ=10

sizes for returned items and various number of kanbans. The demand rate is fixed to 0.8 in all of the experiments.

Proportion of backordered demands (P_B), average number of backordered demands (Q_D), effective return rate (γ_{eff}) and one arbitrary queue ($NQ_{5,11}$) is selected as response variable to compare the results. According to the results represented in Tables 5.4 and 5.5 the analytical model finds fairly accurate results. At first glance the relative errors can be seen a little high but the lowest values of the response variables cause this. The average relative error of the analytical model compared with simulation is 3.76 per cent. The only situation is in the first example the approximation results are not good enough. This kind of large errors maybe encountered when K is very low, especially K=1.The reason for this situation is that the arrival processes involved in the approximation technique are assumed to be state-dependent Markovian processes, which is not a good approximation when the number of kanbans is too small. Indeed, with a small number of kanbans, there are strong dependencies among stations that are not well captured by the state-dependent Markovian processes[1].

Table 5.3. Parameter set for the validation of the analytical method

Ex	B	γ	μ_i	μ_R	μ_M	K_R	K_M	K_i
1	inf	0.2	1	1	1	1	1	6
2	inf	0.6	1	0.5	1	2	2	8
3	inf	0.4	1	0.25	0.25	3	2	6
4	inf	0.4	0.75	1	0.5	1	3	8
5	inf	0.2	0.75	0.5	0.25	2	3	7
6	inf	0.6	0.75	0.25	0.5	3	1	7
7	4	0.4	1	1	0.5	2	2	7
8	4	0.2	1	0.5	0.5	3	3	6
9	4	0.6	1	0.25	1	1	3	7
10	4	0.6	0.75	1	0.25	2	1	6
11	4	0.4	0.75	0.5	1	3	1	8
12	4	0.2	0.75	0.25	0.25	1	2	8
13	8	0.6	1	1	0.25	3	3	8
14	8	0.4	1	0.5	0.25	1	1	7
15	8	0.2	1	0.25	0.5	2	1	8
16	8	0.2	0.75	1	1	3	2	7
17	8	0.6	0.75	0.5	0.5	1	2	6
18	8	0.4	0.75	0.25	1	2	3	6

Table 5.4. Results for P_B and Q_D

	P_B				Q_D		
Ex	Sim	Analytic	Error	Ex	Sim	Analytic	Error
1	0.654	0.743	-13.6	1	10.535	13.945	-32.4
2	0.289	0.310	-7.2	2	1.821	1.940	-6.5
3	0.554	0.586	-5.8	3	5.906	6.057	-2.5
4	0.018	0.019	-6.4	4	0.027	0.029	-7.8
5	0.031	0.033	-4.2	5	0.049	0.052	-6.1
6	0.031	0.033	-7.2	6	0.046	0.052	-12.2
7	0.402	0.427	-6.1	7	3.137	3.220	-2.7
8	0.552	0.587	-6.4	8	5.841	6.094	-4.3
9	0.406	0.423	-4.3	9	3.329	3.160	5.1
10	0.057	0.059	-3.2	10	0.094	0.097	-3.4
11	0.018	0.019	-6.1	11	0.030	0.030	-0.2
12	0.018	0.019	-3.1	12	0.027	0.029	-7.0
13	0.294	0.310	-5.1	13	1.922	1.930	-0.4
14	0.405	0.423	-4.7	14	3.298	3.163	4.1
15	0.286	0.313	-9.3	15	1.807	1.967	-8.9
16	0.033	0.035	-6.4	16	0.054	0.057	-7.0
17	0.056	0.059	-4.4	17	0.091	0.097	-6.7
18	0.057	0.059	-3.7	18	0.096	0.098	-2.0

Table 5.5. Results for γ_{eff} and $NQ_{5,11}$

	γ_{eff}				$NQ_{5,11}$		
Ex	Sim	Analytic	Error	Ex	Sim	Analytic	Error
1	0.200	0.200	0.2	1	2.643	2.490	5.8
2	0.600	0.600	0.0	2	4.870	4.776	1.9
3	0.400	0.400	0.1	3	3.283	3.210	2.2
4	0.400	0.400	0.0	4	6.555	6.528	0.4
5	0.200	0.200	0.0	5	5.578	5.554	0.4
6	0.599	0.600	-0.1	6	5.559	5.519	0.7
7	0.391	0.387	1.1	7	4.061	3.918	3.5
8	0.200	0.200	-0.2	8	3.275	3.197	2.4
9	0.543	0.528	2.8	9	4.046	3.985	1.5
10	0.522	0.508	2.6	10	4.610	4.577	0.7
11	0.398	0.398	0.0	11	6.177	6.086	1.5
12	0.199	0.200	-0.1	12	6.553	6.528	0.4
13	0.581	0.570	1.8	13	4.866	4.810	1.1
14	0.399	0.396	0.8	14	4.059	3.979	2.0
15	0.199	0.200	-0.3	15	4.819	4.677	3.0
16	0.200	0.200	-0.1	16	5.040	4.927	2.2
17	0.541	0.529	2.1	17	4.619	4.584	0.8
18	0.400	0.400	0.1	18	4.536	4.477	1.3

6. NUMERICAL ANALYSIS

For the numerical examples the model given in Figure 5.1 is analyzed. In each stage, the processing time of machines are exponentially distributed with mean μ_i=1, i=1...9. The arrival processes of returned products follow Poisson distributions at stages 2 and 4 with rates γ_1 and γ_2 respectively. And the demand process again follows a Poisson distribution with rate Λ.

In this chapter the consequences of change in backorder cost, change in manufacturing-remanufacturing costs, change in return rate of used items and change in disposal cost is examined where for each of these changes, the systems with disposal and without disposal are considered. The notation used in this chapter is given below.

Z_{Best}: *The minimum cost found by the search method*

P_B: *Percentage of backordered demands*

Q_D: *Average number of backordered demands*

Q_{R1}: *Average number of returned items in* $Queue_{2,2}$ *(Returned item buffer of stage-2)*

Q_{R2}: *Average number of returned items in* $Queue_{4,6}$ *(Returned item buffer of stage-4)*

TH_r: *Throughput of stage-r*

Z_{Q_R}: *Cost of holding returned parts in the returned items buffers(both stage-2 and stage-4)*

Z_{gWIP}: *Cost of WIP just in the production system(*Z_{Q_R} *is excluded)*

Z_{WIP}: Z_{Q_R}+Z_{gWIP} *; Total holding cost*

Z_{TH}: *Total production cost of the system*

Z_{Back}: *Total backorder cost*

Z_{Disp}: *Total disposal cost of the system*

C_{Disp}: *Cost of disposal*

B: Buffer size for returned items

b: Backorder cost

r: Discount rate

Table 6.1. Change in the backorder cost without disposal, $\gamma_{1,2} = 0.35$

b	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	Kanban Set
3.2	26.27	0.39	1.45	0.25	0.10	15.94	5.60	4.63	0.35	0.35	(1,3,1,3,4,4,6,6,4)
6.4	30.12	0.24	0.89	0.59	0.24	18.59	5.60	5.69	0.35	0.35	(1,2,1,2,4,4,6,6,6)
12.8	34.25	0.12	0.42	0.25	0.10	23.19	5.60	5.36	0.35	0.35	(1,3,1,3,5,5,6,6,8)
25.6	38.40	0.08	0.25	0.25	0.10	26.30	5.60	6.40	0.35	0.35	(1,3,1,3,5,5,7,7,9)
51.2	42.45	0.04	0.12	0.26	0.10	30.75	5.60	6.00	0.35	0.35	(1,3,1,3,6,6,6,6,12)

The default system variables are given below, they vary in sections for numerical examples.

Discount Rate (r)=0.2

Cost Parameters=(2,1,2,1,1,1,1,1,1), Processing cost of each machine-i respectively, i=1,...,9

Demand Rate (Λ) =0.7

Return Rate ($\gamma_1 = \gamma_2$) = 0.35

Backorder Cost= 12.8 (8 times of $h_{9,16}$)

The examples in this section are categorized according to the allowance of disposal. In the cases with no disposal the buffer size for returned items is infinite and in the rest of the cases it is limited to some finite value.

6.1. Change in Backorder Cost

The backorder cost is estimated proportional to the holding cost of the output buffer of the system, namely $h_{9,16}$ (holding cost of items in $Queue_{9,16}$). We analyze an experiment set that consists of five experiments where the backorder cost is 2,4,8,16, and 32 times of $h_{9,16}$ in these examples. The minimum cost of the system is searched.

6.1.1. Without Disposal

In this case disposal is not allowed. The return rates for components are both equal to 0.35 ($\gamma_1 = \gamma_2$). The results according to the change in backorder cost are summarized in Table 6.1 below.

Table 6.2. Change in the backorder cost without disposal, $\gamma_{1,2} = 0.65$

b	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	Kanban Set
3.2	31.45	0.41	1.64	7.13	2.85	18.35	5.00	5.24	0.05	0.65	(1,11,1,11,4,4,5,5,4)
6.4	34.74	0.22	0.74	7.14	2.86	22.17	5.00	4.71	0.05	0.65	(1,11,1,11,4,4,6,6,6)
12.8	38.54	0.11	0.36	7.22	2.89	26.00	5.00	4.65	0.05	0.65	(1,11,1,11,5,5,6,6,8)
25.6	42.19	0.07	0.22	7.23	2.89	28.76	5.00	5.53	0.05	0.65	(1,11,1,11,5,5,7,7,9)
51.2	45.79	0.03	0.10	7.26	2.91	32.58	5.00	5.31	0.05	0.65	(1,11,1,11,6,6,6,6,12)

From Table 6.1 it can be seen that when the backorder cost increases the proportion of backordered demands decreases. The system does not allow backordered demands by assigning more kanbans to stages when the penalty is increased. Especially the number of kanbans at the last stage increases and the system holds more products at the output buffer of the system. Hence when a demand arrives, the probability of finding the finished goods buffer empty, decreases. And it is clear that when the system assigns more kanbans, the WIP costs increases and this is the reason of increase in the total cost of the system. When the backorder cost is low ($2^*h_{9,16}$) the percentage of backordered demands is %38,7 and when it is increased ($32^*h_{9,16}$), the percentage of backordered demands decreases to %3,7. From remanufacturing point of view there is not a significant change; the system uses all the returned products to reduce the total cost.

When the return rates are increased to 0.65 the results are similar and summarized in Table 6.2

6.1.2. With Disposal

In this case disposal is allowed and the disposal cost is fixed to twice of the total production cost. Here we analyze two different scenarios to examine the behaviour of disposal action. In the first one the return rates are 0.35 ($\gamma_1 = \gamma_2$) as in the case without disposal. In the second one the return rate are increased to 0.65.

In the first case the buffer size for returned items (B) is fixed to five. The results are nearly similar with the previous one; when the backorder cost is increased, proportional to this cost, the total cost increases and the percentage of backordered demands decreases. Here since the return rate is not high the system uses all the returned items to decrease the total cost. For this example if we did not fix the buffer size to five, the system will increase B and the results will be identical with the previous case. That is to say, for small return rates disposal action is not necessary and using all the returned products is more feasible from the cost point of view. The results are summarized in Table 6.3

In the second case the return rates are increased to 0.65 to enable disposal of returned items. Here again the disposal cost is fixed to twice the value of the total production cost. But this time the buffer size for returned items is not constant and the buffer size that gives the minimum cost is searched. The results are summarized in Table 6.4.

It is again clear that when the backorder cost increases the percentage of backordered demands decreases. But here when we compare the Tables 6.2 and 6.4 we can say that for high return rates disposal is a necessary action to obtain the minimum total cost. The optimum buffer size slightly increases with the increase in backorder cost. In the Figures 6.1 and 6.2, for b is equal to 3.2 and 25.6, the change of total cost with respect to buffer sizes and the determination of the optimal buffer sizes is given.

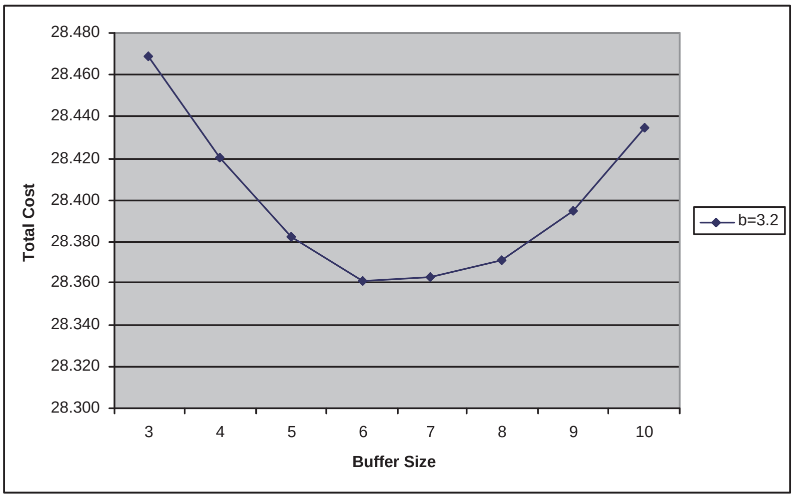

Figure 6.1. Change of total cost according to buffer size, b =3.2

Table 6.3. Change in the backorder cost with disposal, $\gamma_{1,2} = 0.35$

B=5												
b	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
3.2	26.30	0.39	1.46	0.02	0.01	16.03	5.60	4.66	0.01	0.35	0.35	(1,6,1,6,4,4,6,6,4)
6.4	30.40	0.19	0.73	0.02	0.01	20.11	5.60	4.67	0.01	0.35	0.35	(1,6,1,6,4,4,6,6,7)
12.8	34.29	0.12	0.42	0.02	0.01	23.28	5.60	5.39	0.01	0.35	0.35	(1,6,1,6,5,5,6,6,8)
25.6	38.44	0.08	0.25	0.02	0.01	26.39	5.60	6.43	0.01	0.35	0.35	(1,61,6,5,5,7,7,9)
51.2	42.48	0.04	0.11	0.02	0.01	31.03	5.60	5.84	0.01	0.35	0.35	(1,6,1,6,6,6,7,7,12)

Table 6.4. Change in the backorder cost with disposal, $\gamma_{1,2} = 0.65$

B^*	b	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
6	3.2	28.361	0.404	1.595	1.279	0.512	16.084	5.072	5.105	1.588	0.086	0.614	(1,7,1,7,3,3,6,6,4)
7	6.4	31.982	0.216	0.743	1.577	0.631	20.083	5.066	4.757	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,6)
7	12.8	35.617	0.131	0.451	1.577	0.631	22.705	5.066	5.769	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,8)
7	25.6	40.238	0.067	0.258	1.576	0.630	26.497	5.066	6.599	1.445	0.083	0.617	(1,7,1,7,4,4,5,5,12)
8	51.2	43.161	0.041	0.123	1.873	0.749	29.720	5.060	6.320	1.312	0.080	0.620	(1,7,1,7,4,4,8,8,11)

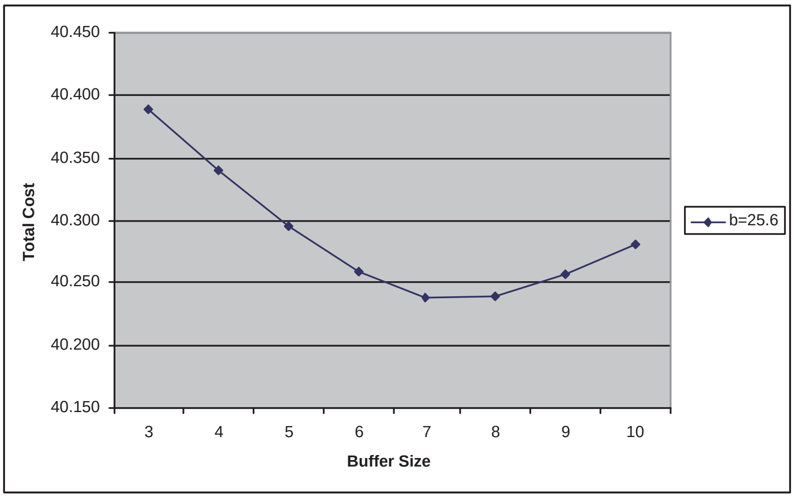

Figure 6.2. Change of total cost according to buffer size, b =25.6

6.2. Change in Manufacturing-Remanufacturing Costs

In this section the consequences of the change in remanufacturing/manufacturing cost is investigated. The manufacturing cost is proportional to remanufacturing cost. We use an experiment set that consists of five experiments and in each of them manufacturing cost is x times of remanufacturing cost. And all the numerical analysis is done to design a system whose percentage of backordered demands has to be a given threshold, which is 15 per cent here.

6.2.1. Without Disposal

The results of the change in manufacturing/remanufacturing cost with no disposal are given in the Table 6.5. Only costs of manufacturing and remanufacturing stages are shown and for the rest of the stages costs are fixed to 1. The return rates are 0.35 ($\gamma_1 = \gamma_2$).

When the difference between the manufacturing and remanufacturing cost increases, the system tends to use more returned items and assigns more kanbans to remanufacturing stages. Another reason to the increase in remanufacturing kanbans is the increase in unit holding cost(h_n) for returned items.In this example the kanbans for

Table 6.5. Change in manufacturing/remanufacturing cost without disposal, $\gamma_1 = \gamma_2 = 0.35$

Cost	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	Kanban Set
(1.1),1	29.57	0.12	0.39	2.13	0.09	19.47	4.97	5.04	0.35	0.35	(1,1,1,1,5,5,6,6,8)
(1.5),1	31.72	0.12	0.41	0.59	0.12	21.07	5.25	5.29	0.35	0.35	(1,2,1,2,5,5,6,6,8)
2,1	34.25	0.12	0.46	0.25	0.10	23.19	5.60	5.92	0.35	0.35	(1,3,1,3,5,5,6,6,8)
3,1	39.24	0.12	0.42	0.12	0.09	27.46	6.30	5.38	0.35	0.35	(1,4,1,4,5,5,6,6,8)
4,1	44.20	0.12	0.42	0.06	0.07	31.75	7.00	5.39	0.35	0.35	(1,5,1,5,5,5,6,6,8)

remanufacturing stages increases from 1 to 5 with the increase in manufacturing cost. For the systems where the difference between manufacturing and remanufacturing cost is high, remanufactured parts should be highly used to reduce the total cost.

For the same experiment set if the orders are satisfied with only manufacturing the results are given in Table 6.6 below. Naturally pure manufacturing option is expensive

Table 6.6. Change in manufacturing cost with pure manufacturing option

Cost	Z_{Best}	P_B	Q_D	Z_{WIP}	Z_{TH}	Z_{Back}	TH	Kanban Set
(1.1),1	31.066	0.136	0.485	19.812	5.040	6.214	0.700	(4,4,5,5,6,6,8)
(1.5),1	34.749	0.136	0.485	22.891	5.600	6.214	0.700	(4,4,5,5,6,6,8)
2,1	38.886	0.136	0.485	26.739	6.300	6.214	0.700	(4,4,5,5,6,6,8)
3,1	47.160	0.136	0.485	34.435	7.700	6.214	0.700	(4,4,5,5,6,6,8)
4,1	55.330	0.141	0.522	41.480	9.100	6.682	0.700	(4,4,4,4,7,7,8)

than hybrid option. As expected the gap between the cost of pure manufacturing and hybrid option increases when the difference in manufacturing/remanufacturing cost increases.

When the holding cost for returned items (h_n) is fixed to a value for all examples, which is the holding cost of third example here i.e. h_n=0.2, the results are given in Table 6.7.

Unit holding cost for a remanufactured part increaes with the increase in manufacturing cost. When manufacturing cost is low the system takes the returned parts in by assigning more remanufacturing kanbans. When the manufacturing cost increases

Table 6.7. Change in manufacturing/remanufacturing cost without disposal, fixed h_n=0.2, $\gamma_1 = \gamma_2 = 0.35$

Cost	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	Kanban Set
(1.1),1	30.021	0.125	0.421	0.028	0.011	19.653	4.970	5.387	0.350	0.350	(1,6,1,6,5,5,6,6,8)
(1.5),1	31.909	0.125	0.419	0.254	0.101	21.200	5.250	5.358	0.350	0.350	(1,3,1,3,5,5,6,6,8)
2,1	34.254	0.125	0.419	0.254	0.101	23.194	5.600	5.358	0.350	0.350	(1,3,1,3,5,5,6,6,8)
3,1	38.911	0.124	0.413	0.594	0.238	27.088	6.300	5.285	0.350	0.350	(1,2,1,2,5,5,6,6,8)
4,1	43.565	0.124	0.413	0.594	0.238	31.042	7.000	5.285	0.350	0.350	(1,2,1,2,5,5,6,6,8)

the system prefers to hold returned parts in the returned items buffer by decreasing the number of kanbans rather than taking them in. For infinite capacity of returned items buffer the system uses all the returned parts.

6.2.2. With Disposal

In this case the effect of disposal is investigated where the return rate is 0.65 and buffer size is fixed to 5. The results are given in Table 6.8. The reason for selecting a fixed buffer size is to observe the holding cost of returned items with respect to change in manufacturing/remanufacturing costs. Disposal cost is again fixed to twice the value of the total production cost. As we mention in Section 5.3, the holding cost for returned items is determined as; manufacturing cost minus remanufacturing cost times the discount rate.

$$h_{Returneditems} = r * (C_M - C_R)$$

So when the difference between manufacturing and remanufacturing cost increases the holding cost for returned items increases. From the table the increase in Z_{Q_R} can be seen. And notice that the γ_{eff} decreases with the increase in manufacturing cost because the system prefers to dispose returned items rather than keeping them due to high holding cost.

Secondly we analyze the same experiment set again but this time buffer sizes are not fixed, optimum buffer size is searched and the results are given in Table 6.9. The results are naturally better than the fixed one of Table 6.8. And as mentioned above due to high holding costs, the procedure assigns low buffer sizes and dispose returned items to reduce the total cost. When the manufacturing cost is 1.1, the optimum buffer size is 41 and the γ_{eff} is 0.645 that means the system nearly uses all the returned products. With the increase in manufacturing cost γ_{eff} decreases to 0.592.

Table 6.8. Change in manufacturing/remanufacturing cost for fixed B, $\gamma_1 = \gamma_2 = 0.65$

B=5												
Cost	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
(1.1),1	32.206	0.131	0.451	0.776	0.031	20.136	4.916	5.770	1.353	0.081	0.619	(1,9,1,9,4,4,6,6,8)
(1.5),1	33.758	0.131	0.452	0.881	0.176	21.271	4.985	5.779	1.546	0.085	0.615	(1,8,1,8,4,4,6,6,8)
2,1	35.660	0.131	0.452	1.007	0.403	22.598	5.081	5.790	1.788	0.091	0.609	(1,7,1,7,4,4,6,6,8)
3,1	39.392	0.131	0.452	1.007	0.806	25.746	5.263	5.790	1.788	0.091	0.609	(1,6,1,6,4,4,6,6,8)
4,1	43.066	0.131	0.453	1.163	1.396	28.283	5.486	5.803	2.098	0.098	0.602	(1,6,1,6,4,4,6,6,8)

Table 6.9. Change in manufacturing/remanufacturing cost with disposal for B^*, $\gamma_1 = \gamma_2 = 0.65$

Cost	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
(1.1),1	41	31.229	0.129	0.444	11.779	0.471	19.946	4.911	5.687	0.214	0.055	0.645	(1,6,1,6,4,4,6,6,8)
(1.5),1	13	33.491	0.130	0.448	3.855	0.771	20.988	4.973	5.736	1.024	0.073	0.627	(1,6,1,6,4,4,6,6,8)
2,1	7	35.617	0.131	0.451	1.577	0.631	22.705	5.066	5.769	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,8)
3,1	4	39.349	0.131	0.453	0.740	0.592	25.664	5.282	5.804	2.007	0.096	0.604	(1,7,1,7,4,4,6,6,8)
4,1	1	42.860	0.132	0.456	0.090	0.108	28.797	5.550	5.841	2.565	0.108	0.592	(1,8,1,8,4,4,6,6,8)

When unit holding cost for returned items (h_n) is fixed to 0.2 as in the without disposal case the results are given for optimum and fixed buffer sizes in Tables 6.10 and 6.11 respectively. When unit holding cost for returned items (h_n) is fixed to 0.1 the results are given for optimum and fixed buffer sizes in Tables 6.12 and 6.13 respectively.

The behaviour of the system is similar with the case where disposal is not allowed. When unit holding cost for a remanufactured part is low by assigning more remanufacturing kanbans the system pulls returned parts into the manufacturing system. When the manufacturing cost increases the system prefers to hold the returned parts in the returned items buffer by increasing the buffer capacity and decreasing the number of kanbans. The situation is similar for the fixed buffer capacity case, the number of kanbans decreases with the increase in manufacturing cost.

6.3. Change in Return Rate

In this section the consequences of the change in the rate of returned items is investigated. Again all the numerical analysis is done to design a system whose percentage of backordered demands has to be a given threshold that is 15 per cent.

6.3.1. Without Disposal

In the first case of changes in return rate, disposal is not allowed and γ_1 and γ_2 is equal to each other($\gamma_1 = \gamma_2 = \gamma$) and the results are given in Table 6.14.

We first begin with pure manufacturing option where the return rate is 0 and step-by-step we increase the return rate. Firstly the total cost begins to decrease and at some point it reaches its minimum cost. For this case we obtain the minimum cost when the return rate is 0.5. And after this point increasing return rate increases the total cost. Because the system has to remanufacture all the returned products, it assigns more kanban to remanufacturing stages, which increases the WIP cost. And of course the cost of holding the returned items increases. So when there is a necessity to remanufacture all the returns there exists a return rate that gives the minimum cost

Table 6.10. Change in manufacturing/remanufacturing cost with disposal for B^*, fixed (h_n)=0.2, $\gamma_1 = \gamma_2 = 0.65$

Cost	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
(1.1),1	2	33.015	0.131	0.451	0.141	0.057	20.956	4.916	5.770	1.316	0.080	0.620	(1,12,1,12,4,4,6,6,8)
(1.5),1	6	34.224	0.131	0.451	1.129	0.452	21.633	4.982	5.770	1.387	0.082	0.618	(1,8,1,8,4,4,6,6,8)
2,1	7	35.617	0.131	0.451	1.577	0.631	22.705	5.066	5.769	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,8)
3,1	9	38.240	0.130	0.450	2.892	1.157	24.332	5.257	5.766	1.727	0.089	0.611	(1,5,1,5,4,4,6,6,8)
4,1	10	40.712	0.130	0.450	3.290	1.316	26.625	5.418	5.758	1.596	0.086	0.614	(1,5,1,5,4,4,6,6,8)

Table 6.11. Change in manufacturing/remanufacturing cost for fixed B, fixed (h_n)=0.2, $\gamma_1 = \gamma_2 = 0.65$

B=5												
Cost	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
(1.1),1	33.041	0.131	0.451	0.776	0.310	20.691	4.916	5.770	1.353	0.081	0.619	(1,9,1,9,4,4,6,6,8)
(1.5),1	34.232	0.131	0.451	0.776	0.310	21.817	4.981	5.770	1.353	0.081	0.619	(1,9,1,9,4,4,6,6,8)
2,1	35.660	0.131	0.452	1.007	0.403	22.598	5.081	5.790	1.788	0.091	0.609	(1,7,1,7,4,4,6,6,8)
3,1	38.393	0.131	0.453	1.163	0.465	24.736	5.291	5.803	2.098	0.098	0.602	(1,6,1,6,4,4,6,6,8)
4,1	41.052	0.131	0.453	1.163	0.465	27.199	5.486	5.803	2.098	0.098	0.602	(1,6,1,6,4,4,6,6,8)

Table 6.12. Change in manufacturing/remanufacturing cost with disposal for B^*, fixed (h_n)=0.1, $\gamma_1 = \gamma_2 = 0.65$

Cost	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
(1.1),1	2	33.015	0.131	0.451	0.141	0.057	20.956	4.916	5.770	1.316	0.080	0.620	(1,12,1,12,4,4,6,6,8)
(1.5),1	13	33.491	0.130	0.448	3.855	0.771	20.988	4.973	5.736	1.024	0.073	0.627	(1,6,1,6,4,4,6,6,8)
2,1	13	34.774	0.130	0.448	3.855	0.771	22.197	5.047	5.736	1.024	0.073	0.627	(1,6,1,6,4,4,6,6,8)
3,1	15	37.198	0.130	0.448	5.303	1.061	24.071	5.203	5.728	1.135	0.076	0.624	(1,5,1,5,4,4,6,6,8)
4,1	16	39.596	0.130	0.447	5.705	1.141	26.315	5.346	5.724	1.070	0.074	0.626	(1,5,1,5,4,4,6,6,8)

Table 6.13. Change in manufacturing/remanufacturing cost for fixed B, fixed (h_n)=0.1, $\gamma_1 = \gamma_2 = 0.65$

B=5												
Cost	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
(1.1),1	33.026	0.130	0.450	0.688	0.275	20.879	4.915	5.762	1.194	0.077	0.623	(1,10,1,10,4,4,6,6,8)
(1.5),1	33.758	0.131	0.452	0.881	0.176	21.271	4.985	5.779	1.546	0.085	0.615	(1,8,1,8,4,4,6,6,8)
2,1	35.172	0.131	0.452	1.007	0.201	22.312	5.081	5.790	1.788	0.091	0.609	(1,7,1,7,4,4,6,6,8)
3,1	37.889	0.131	0.453	1.163	0.233	24.465	5.291	5.803	2.098	0.098	0.602	(1,6,1,6,4,4,6,6,8)
4,1	40.548	0.131	0.453	1.163	0.233	26.928	5.486	5.803	2.098	0.098	0.602	(1,6,1,6,4,4,6,6,8)

Table 6.14. Change in return rate without disposal

γ	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	Kanban Set
γ=0	38.886	0.136	0.485				6.300	6.214	0.700		(4,4,5,5,6,6,8)
γ=0.05	38.199	0.130	0.447	0.011	0.004	26.275	6.200	5.719	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ=0.2	35.744	0.138	0.496	0.251	0.100	23.395	5.900	6.348	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	34.254	0.125	0.419	0.254	0.101	23.194	5.600	5.358	0.350	0.350	(1,3,1,3,5,5,6,6,8)
γ=0.5	33.698	0.117	0.381	0.850	0.340	23.182	5.300	4.875	0.200	0.500	(1,5,1,5,5,5,6,6,8)
γ=0.65	38.540	0.114	0.364	7.225	2.890	25.997	5.000	4.653	0.050	0.650	(1,11,1,11,5,5,6,6,8)
γ=0.7	61.267	0.114	0.361	30.086	12.034	39.644	4.968	4.621	0.017	0.700	(1,39,1,39,5,5,6,6,8)

Table 6.15. Change in return rate without disposal, $\gamma_1 \neq \gamma_2$

γ_1=0.35	Z_{Best}	P_B	Q_D	Q_{R_1}	Q_{R_1}	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	TH_3	TH_4	Kanban Set
γ_2=0.05	36.20	0.14	0.50	0.59	0.01	0.12	23.80	5.90	6.38	0.35	0.35	0.65	0.05	(1,2,2,1,5,5,6,6,8)
γ_2=0.2	34.98	0.13	0.45	0.59	0.25	0.17	23.28	5.75	5.78	0.35	0.35	0.50	0.20	(1,2,1,1,5,5,6,6,8)
γ_2=0.35	34.25	0.12	0.42	0.25	0.25	0.10	23.19	5.60	5.36	0.35	0.35	0.35	0.35	(1,3,1,3,5,5,6,6,8)
γ_2=0.5	33.97	0.12	0.40	0.25	0.85	0.22	23.19	5.45	5.11	0.35	0.35	0.20	0.50	(1,3,1,5,5,5,6,6,8)
γ_2=0.65	36.39	0.12	0.39	0.25	7.22	1.49	24.61	5.30	4.99	0.35	0.35	0.05	0.65	(1,3,1,11,5,5,6,6,8)
γ_2=0.7	47.75	0.12	0.39	0.25	30.07	6.06	31.43	5.28	4.97	0.35	0.35	0.02	0.70	(1,3,1,39,5,5,6,6,8)

of the system. The results in Table 6.14 is shown in Figure 6.3.

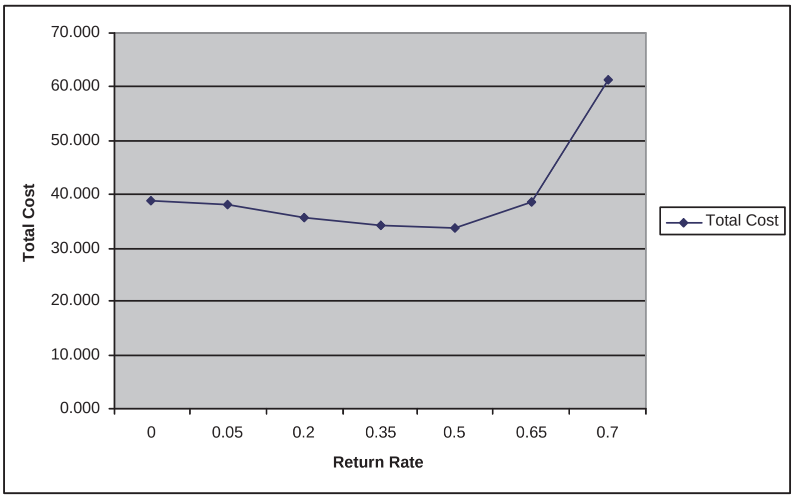

Figure 6.3. Total cost up to change in return rate without disposal

The return rate of components may not be symmetric. To analyze this situation γ_1 is fixed to some value and γ_2 changes. The results are and given in Table 6.15. They are similar with the Table 6.14 and are somewhere between the cases when $\gamma_1 = \gamma_2$, viz when $\gamma_1 = 0.35$ and $\gamma_2 = 0.2$ the total cost is between the cases when $\gamma_1 = \gamma_2 = 0.2$ and $\gamma_1 = \gamma_2 = 0.35$.

In all of the experiments we assume that service rates of all machines are equal to each other. But in real life remanufacturing usually takes less time than manufacturing. In Table 6.16 the results are belong to a situation where remanufacturing is faster than manufacturing, i.e. $\mu_R = 0.5$, $\mu_M = 1$. When we compare Tables 6.14 and 6.16, we see that the total cost of the system decreases with the increase in the speed of remanufacturing stages. The system assigns less remanufacturing kanbans. The average number waiting in the returned items buffer decreases and with respect to this Z_{Q_R} decreases. The minimum cost of the system is still obtained when the return rate is 0.5.

Table 6.16. Change in return rate without disposal, $\mu_R = 0.5$

γ	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	TH_1	TH_2	Kanban Set
γ=0	38.886	0.136	0.485				6.300	6.214	0.700		(4,4,5,5,6,6,8)
γ=0.05	38.187	0.130	0.447	0.006	0.002	26.265	6.200	5.719	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ=0.2	35.651	0.138	0.496	0.120	0.048	23.355	5.900	6.348	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	33.981	0.120	0.394	0.742	0.297	23.042	5.600	5.042	0.350	0.350	(1,1,1,1,5,5,6,6,8)
γ=0.5	33.217	0.116	0.375	1.699	0.679	22.441	5.300	4.797	0.200	0.500	(1,2,1,2,5,5,6,6,8)
γ=0.65	36.740	0.114	0.363	9.403	3.761	23.328	5.000	4.650	0.050	0.650	(1,5,1,5,5,5,6,6,8)
γ=0.7	56.856	0.114	0.361	25.694	10.278	36.974	4.981	4.624	0.020	0.700	(1,38,1,38,5,5,6,6,8)

6.3.2. With Disposal

For the case with disposal the minimum costs and optimum buffer sizes with respect to change in return rates are given in Table 6.17.

When we compare Table 6.14 and 6.17, again we see that disposal decreases the total cost of the system for high return rates. In Table 6.17, when the return rate increases, the optimum buffer sizes increases in the beginning, then it decreases. The reason for this is, for low return rates low buffer sizes are adequate but at high return rates after some point carrying WIP cost of returned items becomes infeasible. Due to high return rates the system mostly finds returned items whenever needed. Therefore for high return rates there is no need to assign high buffer sizes.

In this experiment set disposal begins nearly at rate $\gamma = 0.5$. And after this value an increase in return rate decreases the buffer size. Since the system can find returned items whenever needed due to high return, holding them increase the cost so disposing them is chosen.

Another observation from Table 6.17 is that an increase in return rate decrease the production cost (Z_{TH}) since remanufacturing operation are cheaper.

When buffer size is fixed to 5 the results are similar with the previous one and they are given in Table 6.18. Here notice that for the cases where γ is 0.35 and 0.5 the systems assign more remanufacturing kanbans than Table 6.17. When optimum buffer size is higher than the fixed one, by increasing the remanufacturing kanbans the system pulls more returned items to decrease the total cost.

In Table 6.19 the results are given for the different return rates for fixed buffer size. γ_1 is fixed to 0.35 and γ_2 changes. The minimum cost is obtained when $\gamma_1 = 0.35$ and $\gamma_2 = 0.5$.

We again look at the case where remanufacturing is faster than manufacturing

Table 6.17. Change in return rate with disposal for B^*

γ	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
γ=0		38.886	0.136	0.485				6.300	6.214		0.700		(4,4,5,5,6,6,8)
γ=0.05	3	38.199	0.130	0.447	0.011	0.004	26.275	6.200	5.719	0.000	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ=0.2	9	35.744	0.138	0.496	0.251	0.100	23.395	5.900	6.348	0.001	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	14	34.254	0.125	0.419	0.253	0.101	23.194	5.600	5.358	0.001	0.350	0.350	(1,3,1,3,5,5,6,6,8)
γ=0.5	17	33.612	0.137	0.487	0.807	0.323	21.741	5.301	6.237	0.011	0.200	0.500	(1,5,1,5,5,6,6,8,)
γ=0.65	7	35.617	0.131	0.451	1.577	0.631	22.705	5.066	5.769	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,8)
γ=0.7	5	37.088	0.130	0.447	1.393	0.557	22.829	5.034	5.720	2.948	0.067	0.633	(1,7,1,7,4,4,6,6,8)

Table 6.18. Change in return rate with disposal for fixed B

γ	B	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
γ=0		38.886	0.136	0.485				6.300	6.214		0.700		(4,4,5,5,6,6,8)
γ=0.05	5	38.199	0.130	0.447	0.011	0.004	26.275	6.200	5.719	0.000	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ=0.2	5	35.767	0.138	0.496	0.242	0.097	23.395	5.901	6.355	0.019	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	5	34.288	0.125	0.421	0.024	0.010	23.283	5.600	5.387	0.007	0.350	0.350	(1,6,1,6,5,5,6,6,8)
γ=0.5	5	33.728	0.137	0.489	0.099	0.040	22.035	5.304	6.265	0.085	0.202	0.498	(1,9,1,9,5,5,6,6,8)
γ=0.65	5	35.660	0.131	0.452	1.007	0.403	22.598	5.081	5.790	1.788	0.091	0.609	(1,7,1,7,4,4,6,6,8)
γ=0.7	5	37.088	0.130	0.447	1.393	0.557	22.829	5.034	5.720	2.948	0.067	0.633	(1,7,1,7,4,4,6,6,8)

Table 6.19. Change in return rate with disposal, $\gamma_1 \neq \gamma_2$

$B_1 = B_2 = 5$													
$\gamma_1 = 0.35$	Z_{Best}	P_B	Q_D	Q_{R_1}	Q_{R_2}	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_3	TH_4	Kanban Set
γ_2=0.05	36.35	0.13	0.44	0.02	0.01	0.01	24.79	5.90	5.64	0.00	0.65	0.05	(1,6,1,1,6,6,6,6,8)
γ_2=0.2	35.01	0.13	0.46	0.02	0.24	0.05	23.35	5.75	5.84	0.01	0.50	0.20	(1,6,1,1,5,5,6,6,8)
γ_2=0.35	34.29	0.12	0.42	0.02	0.02	0.01	23.28	5.60	5.39	0.01	0.35	0.35	(1,6,1,6,5,5,6,6,8)
γ_2=0.5	34.04	0.12	0.40	0.02	0.10	0.03	23.39	5.45	5.13	0.05	0.20	0.50	(1,6,1,9,5,5,6,6,8)
γ_2=0.65	35.06	0.12	0.39	0.05	1.01	0.21	23.60	5.34	5.00	0.90	0.09	0.61	(1,5,1,7,4,4,7,7,8)
γ_2=0.7	35.82	0.14	0.51	0.02	1.39	0.28	22.21	5.32	6.53	1.48	0.07	0.63	(1,6,1,7,4,4,6,6,8)

for the without disposal case. The results are summarized in the Tables 6.20 and 6.21 for optimum and fixed buffer sized respectively.

If the Tables 6.17 and 6.20 are compared the decrease in total cost can be seen. With the increase in the speed of remanufacturing, effective return rate $\gamma_{effective}$ of the system increases. The system assigns less kanbans to remanufacturing stages when service times of remanufacturing machines decreases and according to change in remanufacturing kanbans, optimum buffer sizes are different from the Table 6.17.

When the buffer size is fixed to five, faster remanufacturing decreases the total cost and the system assigns less kanbans compared to Table 6.18

For the sixth example in the experiment set where γ=0.65 we look at the consequences of the change in μ_R/μ_M ratio, the results for optimum and fixed buffer cases are given in Tables 6.22 and 6.23 respectively. When the speed of remanufacturing stages increase the total cost of the system decreases. Since the system can remanufacture more returned parts the capacity of returned items buffer increases hence the effective return rate($\gamma_{effective}$) increases.

6.4. Change in Disposal Cost

The effect of disposal cost on the system is investigated for two different values. In the first one disposal cost is set as equal to the value of total production cost (C_{Disp}=11) and in the latter one set as twice the value of the total production cost (C_{Disp}=22). To analyze the systems behaviour the return rate changes in the examples and the results are summarized in the Tables 6.24 and 6.25.

When the disposal cost increases, as expected the system increases the capacity of buffer sizes. The situation is depicted in Figure 6.4. For low return rates the capacity of return buffers is not affected since they carry low average number of returned items.

Table 6.20. Change in return rate with disposal, $\mu_R = 0.5$

γ	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
γ=0		38.886	0.136	0.485				6.300	6.214		0.700		(4,4,5,5,6,6,8)
γ=0.05	3	38.187	0.130	0.447	0.006	0.002	26.265	6.200	5.719	0.000	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ=0.2	8	35.651	0.138	0.496	0.120	0.048	23.355	5.900	6.348	0.000	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	13	33.982	0.120	0.394	0.739	0.295	23.042	5.600	5.042	0.003	0.350	0.350	(1,1,1,1,5,5,6,6,8)
γ=0.5	19	33.132	0.136	0.484	0.995	0.398	21.229	5.300	6.201	0.005	0.200	0.500	(1,3,1,3,4,4,6,6,8,)
γ=0.65	11	34.539	0.130	0.446	3.591	1.436	21.060	5.058	5.710	1.275	0.079	0.621	(1,3,1,3,4,4,6,6,8,)
γ=0.7	8	35.874	0.129	0.444	2.882	1.153	21.526	5.013	5.685	2.496	0.057	0.643	(1,4,1,4,4,4,6,6,8)

Table 6.21. Change in return rate with disposal for fixed B, $\mu_R = 0.5$

γ	B	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
γ =0		38.886	0.136	0.485				6.300	6.214		0.700		(4,4,5,5,6,6,8)
γ =0.05	5	38.187	0.130	0.447	0.006	0.002	26.265	6.200	5.719	0.000	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ =0.2	5	35.656	0.138	0.496	0.118	0.047	23.355	5.900	6.349	0.004	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ =0.35	5	34.123	0.124	0.416	0.247	0.099	23.032	5.603	5.319	0.070	0.352	0.348	(1,2,1,2,5,5,6,6,8)
γ =0.5	5	33.448	0.137	0.490	0.190	0.076	21.633	5.307	6.271	0.160	0.204	0.496	(1,6,1,6,5,5,6,6,8)
γ =0.65	5	34.913	0.131	0.453	1.052	0.421	21.810	5.082	5.792	1.809	0.091	0.609	(1,5,1,5,4,4,6,6,8)
γ =0.7	5	36.103	0.130	0.447	1.564	0.625	21.476	5.047	5.724	3.231	0.073	0.627	(1,4,1,4,4,4,6,6,8)

Table 6.22. Change in remanufacturing/manufacturing rate for B^*, $\gamma_1 = \gamma_2 = 0.65$

μ_R/μ_M	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
0.1	13	33.779	0.129	0.444	4.126	1.650	20.497	5.041	5.680	0.910	0.071	0.629	(1,2,1,2,4,4,6,6,8)
0.25	12	34.076	0.129	0.442	4.127	1.651	20.541	5.053	5.660	1.171	0.077	0.623	(1,2,1,2,4,4,6,6,8)
0.5	11	34.539	0.130	0.446	3.591	1.436	21.060	5.058	5.710	1.275	0.079	0.621	(1,3,1,3,4,4,6,6,8,)
0.75	9	34.999	0.130	0.449	2.393	0.957	21.948	5.058	5.751	1.284	0.079	0.621	(1,5,1,5,4,4,6,6,8)
1	7	35.617	0.131	0.451	1.577	0.631	22.705	5.066	5.769	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,8)

Table 6.23. Change in remanufacturing/manufacturing rate for fixed B, $\gamma_1 = \gamma_2 = 0.65$

μ_R/μ_M	B	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
0.1	5	34.709	0.131	0.453	1.131	0.453	21.425	5.088	5.804	1.939	0.094	0.606	(1,4,1,4,4,4,6,6,8)
0.25	5	34.757	0.131	0.453	1.153	0.461	21.427	5.090	5.803	1.976	0.095	0.605	(1,4,1,4,4,4,6,6,8)
0.5	5	34.913	0.131	0.453	1.052	0.421	21.810	5.082	5.792	1.809	0.091	0.609	(1,5,1,5,4,4,6,6,8)
0.75	5	35.183	0.131	0.452	0.996	0.398	22.192	5.079	5.787	1.728	0.089	0.611	(1,6,1,6,4,4,6,6,8)
1	5	35.660	0.131	0.452	1.007	0.403	22.598	5.081	5.790	1.788	0.091	0.609	(1,7,1,7,4,4,6,6,8)

Table 6.24. Change in disposal cost, C_{Disp}=11

	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
γ=0		38.886	0.136	0.485				6.300	6.214		0.700		(4,4,5,5,6,6,8)
γ=0.05	3	38.250	0.137	0.491	0.013	0.005	25.756	6.200	6.289	0.000	0.650	0.050	(1,1,1,1,5,5,6,6,8)
γ=0.2	9	35.744	0.138	0.496	0.251	0.100	23.395	5.900	6.348	0.000	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	12	34.254	0.125	0.419	0.252	0.101	23.194	5.600	5.358	0.001	0.350	0.350	(1,3,1,3,5,5,6,6,8)
γ=0.5	10	33.682	0.117	0.381	0.692	0.277	23.178	5.304	4.879	0.044	0.202	0.498	(1,5,1,5,5,5,6,6,8)
γ=0.65	4	34.851	0.115	0.367	0.861	0.345	23.514	5.107	4.703	1.182	0.104	0.596	(1,6,1,6,5,5,6,6,8)
γ=0.7	3	35.575	0.114	0.366	0.748	0.299	23.551	5.079	4.682	1.964	0.089	0.611	(1,6,1,6,5,5,6,6,8)

Table 6.25. Change in disposal cost, C_{Disp}=22

	B^*	Z_{Best}	P_B	Q_D	$Q_{R_{1,2}}$	Z_{Q_R}	Z_{gWIP}	Z_{TH}	Z_{Back}	Z_{Disp}	TH_1	TH_2	Kanban Set
γ=0		38.886	0.136	0.485				6.300	6.214		0.700		(4,4,5,5,6,6,8)
γ=0.05	3	38.199	0.130	0.447	0.011	0.004	26.275	6.200	5.719	0.000	0.650	0.050	(2,1,2,1,6,6,6,6,8)
γ=0.2	9	35.744	0.138	0.496	0.251	0.100	23.395	5.900	6.348	0.001	0.500	0.200	(1,1,1,1,5,5,6,6,8)
γ=0.35	14	34.254	0.125	0.419	0.253	0.101	23.194	5.600	5.358	0.001	0.350	0.350	(1,3,1,3,5,5,6,6,8)
γ=0.5	17	33.612	0.137	0.487	0.807	0.323	21.741	5.301	6.237	0.011	0.200	0.500	(1,5,1,5,5,6,6,8,)
γ=0.65	7	35.617	0.131	0.451	1.577	0.631	22.705	5.066	5.769	1.446	0.083	0.617	(1,7,1,7,4,4,6,6,8)
γ=0.7	5	37.088	0.130	0.447	1.393	0.557	22.829	5.034	5.720	2.948	0.067	0.633	(1,7,1,7,4,4,6,6,8)

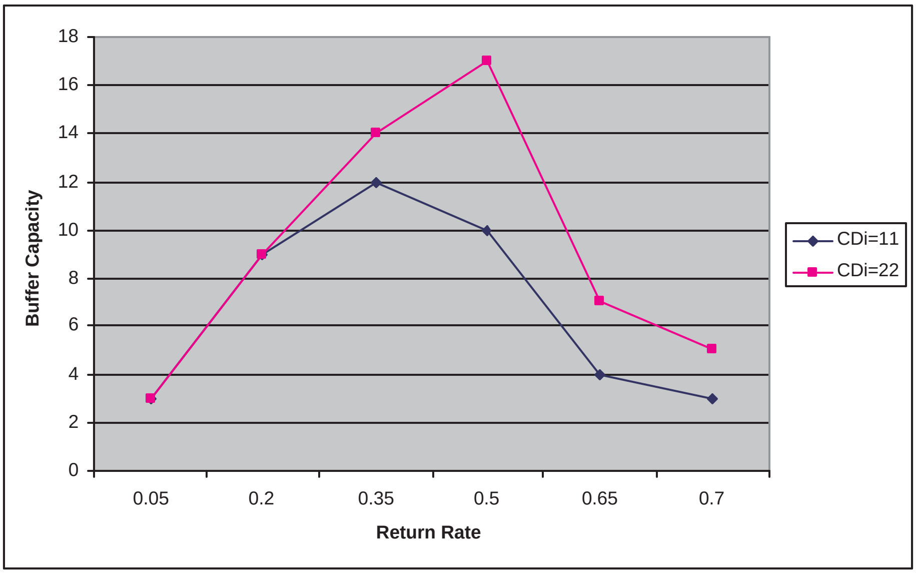

Figure 6.4. Optimum buffer sizes up to different disposal costs

In the beginning buffer sizes increase, after some point they decrease due to high returns. For high return rates increasing the capacity of returned items buffer increases the total cost and assigned capacities are adequate to use the advantage of cheaper returned parts. In the last two experiments the effective return rates are reasonably high.

7. CONCLUSIONS

In this dissertation we studied an analytical model for kanban controlled hybrid systems. The model can be applied to any hybrid system with arbitrary topology. A product-form approximation technique is used to analyze the system. With this analytical model, systems with multiple stages can be investigated where the stages may consist of multiple-machines. The system is modeled as a multi-class closed queuing network. The determination of system parameters is done with stochastic calculations. Average number of queues, average number of backordered demands, average waiting time of backordered demand, throughput of stages, effective return rates are some of the typical performance measures that can be obtained with the analytical model.

As a problem we investigate a kanban-controlled system that is producing a single part type with multiple components, where for some components the system takes returned items. Two different kinds of assembly processes occur in the problem. The first one is the routine assembly process of two components with the associated demand. In the second assembly operation, remanufactured parts have priority over manufactured parts and they are chosen for the operation if they are available at the moment. Arrangement of components for an assembly processes is done with synchronization stations. The behaviour of these synchronization stations is defined with transition rate matrices. For a given parameter set we calculate the minimum cost of the system with the search procedure. After describing the analytical method, we conjecture a heuristic search method that searches for minimum cost of the system or the minimum cost for a marked performance measure. Kanban allocation is realized in light of this procedure.

For the numerical examples we investigate the hybrid systems behaviour under various scenarios such as different return rates, manufacturing/remanufacturing costs, disposal costs, etc. As expected with the increase in backorders cost the system assigns more kanbans to stages, holds more finished products at the output buffer and consequently the percentage of backordered demands decreases.

When the manufacturing cost increases, the unit holding cost for returned items, remanufactured parts and manufactured parts increases also, therefore the system pulls more returned items into the system to reduce the total cost. The capacity of returned items buffer decreases due to the increase in unit holding cost for a returned item and more remanufacturing kanbans are assigned. When the unit holding cost for a returned item is fixed to some value this time with the increase in manufacturing cost the system prefers to hold the returned items in the returned items buffer, increases the capacity of buffers and assigns less remanufacturing kanbans. When different return rates are analyzed we can say that there is and optimum return rate that gives the minimum cost. Beginning from zero return with the increase in return rate total cost decreases down to a certain point than it increases. For high return rates disposal is necessary to decrease the total cost. When return rate increases, the capacity of returned items buffer increases up to a certain point, which gives the minimum cost, then it decreases. In some situations the return rate of components may not be symmetric and in these cases the results are similar and somewhere between symmetric cases.

When the speed of remanufacturing stages increase the system becomes capable of remanufacturing more return. Buffer capacities for returned items increase and total cost of the system decreases.

The increase in disposal cost increases the buffer capacities for returned items. The system remanufactures more retuned items and effective return rate increases. Therefore the total cost of the system decreases.

For future research, disassembly process of returned products may be added to the model. Analysis of a hybrid system producing multiple parts may be another topic for future studies.

In conclusion, the study presents a useful tool for a decision maker in a hybrid production process, using kanban control. With this tool any hybrid model can be observed for various scenarios.

REFERENCES

1. Mascolo, M. D., Y. Frein, and Y. Dallery, “An analytical method for performance evaluation of kanban controlled production systems”, *Operations Research*, Vol. 44, No. 1, pp. 50–64, 1996.

2. Uzsoy, R. and L. A. Martin-Vega, “Modelling kanban based demand pull systems: A survey and critique”, *Manufacturing Review*, Vol. 3, 1990.

3. Baynat, B. and Y. Dallery, “A unified view of product-form approximation techniques for general closed queuing networks”, *Performance Evaluation*, Vol. 18, p. 205, 224 1993.

4. Mascolo, M. D. and Y. Dallery, “Performance evaluation of kanban controlled assembly systems”, Multiconference on Computational Engineering in Systems Applications, Symposium on Discrete Events and Manufacturing Systems,Lille, 9-12 July 1996.

5. Baynat, B., Y. Dallery, M. D. Mascolo, and Y. Frein, “A multi-class approximation technique for the analysis of kanban-like control systems”, *International Journal of Production Research*, Vol. 39, No. 2, pp. 307–328, 2001.

6. Baynat, B. and Y. Dallery, “A product-form approximation method for general closed queuing networks with several classes of customers”, *Performance Evaluation*, Vol. 24, pp. 165–188, 1996.

7. Matta, A., Y. Dallery, and M. D. Mascolo, “Analysis of assembly controlled systems with kanbans”, *European Journal of Operational Research*, Vol. 166, pp. 310–336, 2005.

8. Baynat, B. and Y. Dallery, “Approximate techniques for general closed queuing networks with subnetworks having population constraints”, *European Journal of*

Operational Research, Vol. 69, pp. 250–264, 1993.

9. Baynat, B. and Dallery, Y. 1995 MASCOTS, *Approximate analysis of multi-class synchronized closed queuing networks*, Proceedings of the Third International Workshop on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, 18-20 January 1995.

10. Tardif, V. and L. Maaseidvaag, “An adaptive approach to controlling kanban systems”, *European Journal of Operational Research*, Vol. 132, pp. 411–424, 2001.

11. Guide, V., V. Jayaraman, and R. Srivastava, “Production planning and control for remanufacturing:A state-of-the-art survey”, *Robotics and Computer Integrated Manufacturing*, Vol. 15, pp. 221–230, 1999.

12. van der Laan, E., R. Dekker, and M. Salomon, “Product remanufacuring and disposal”, *International Journal of Production Economics*, Vol. 45, pp. 489–498, 1996.

13. van der Laan, E., R. Dekker, M. Salomon, and A. Ridder, “An (s,Q) inventory model with remanufacturing and disposal”, *International Journal of Production Economics*, Vol. 46-47, pp. 339–350, 1996.

14. der Laan, E. V., M.Salomon, R. Dekker, and L. Wassenhove, “Inventory control in hybrid systems with remanufacturing”, *Management Sicence*, Vol. 45, No. 5, pp. 733–747, 1999.

15. Souza, G. and M. E. Ketzenberg, “Two-stage make-to-order remanufacturing with service-level constraints”, *International Journal of Production Research*, Vol. 40, pp. 477–493, 2002.

16. Teunter, R., “Economic ordering quantities for remanufacturable item inventory systems”, *Naval Research Logistics*, Vol. 48, No. 6, pp. 484–495, 2001.

17. Bayindir, Z., N. Erkip, and R. Gullu, “A model to evaluate inventory costs in a

remanufacturing environment", *International Journal of Production Economics*, Vol. 81-82, pp. 597–607, 2003.

18. Teunter, R., E. van der Laan, and K. Inderfurth, "How to set the holding cost rates in average cost inventory model with reverse logistics?", *Omega*, Vol. 28, pp. 409–415, 2000.

19. van der Laan, E., "An NPV and AC analysis of stochastic inventory system with joint manufacturing and remanufacturing", *International Journal of Production Economics*, Vol. 81-82, pp. 317–331, 2003.

20. Aksoy, H. K. and S. M. Gupta, "Buffer allocation plan for a remanufacturing cell", *Computers and Industrial Engineering*, Vol. 48, pp. 657–677, 2005.

21. Gungor, A. and S. M. Gupta, "Disassembly sequence planning for products, with defective parts in product recovery", *Computers&Industrial Engineering*, Vol. 35, No. (1-2), pp. 161–164, 1998.

22. Gungor, A. and S. M. Gupta, "Disassembly line in product recovery", *International Journal of Production Research*, Vol. 40, No. 11, pp. 2569–2589, 2002.

23. Gupta, S. M. (editor), *Modelling operational behavior of a disassembly line*, Vol. 5583 of *Proceedings of SPIE*, Elif A. Kizilkaya and S. M. Gupta, 2004.

24. Kizilkaya, E. and Gupta, S., *A Modified kanban system for disassembly*, Proceedings of the Northeast Decision Sciences Institute, 2000, 239-241.

25. Gupta, S. M. (editor), *Multi-kanban mechanism for personal computer disassembly*, Vol. 5583 of *Proceedings of SPIE*, G. Udomsawat and S. M. Gupta and S. V. Kamarthi, 2004.

26. Korugan, A. and S. Gupta, "Pull Type Control Mechanisms for Single Stage Hybrid Systems", *Woking Paper (Submitted to Computers and Operations Research)*, 2001.

27. Korugan A. and Gupta S.,Proceedings of the International Conference on Environmentally Conscious Manufacturing II, Newton, MA, USA, *An Adaptive Kanban Control Mechanism for a Single Stage Hybrid System*, 2001.

28. Bruell, S. C. and G. Balbo, *Computational Algorithms for Closed Queuing Networks*, Elsevier North Holland Inc., 1980.

29. Stewart, W. J., *Introduction to the Numerical Solutions of Markov Chains*, Princeton University Press, 1994.

30. *Design Expert 7.0, www.statease.com.*