YAŞAR UNIVERSITY

GRADUATE SCHOOL

PhD THESIS

CONTROL AND PERFORMANCE ANALYSIS OF THREE STATION MAKE-TO-STOCK PRODUCTION LINES

ÖZGÜN YÜCEL

THESIS ADVISOR: ASSIST. PROF. (PhD) ÖNDER BULUT

DOCTOR OF PHILOSOPHY IN INDUSTRIAL ENGINEERING

PRESENTATION DATE: 06.09.2021

BORNOVA / İZMİR

SEPTEMBER 2021

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of the Doctor of Philosophy.

Jury Members:	**Signature:**
Prof. (PhD) Mehmet Cemali DİNÇER Yaşar University	....
Prof. (PhD) Refik GÜLLÜ Boğaziçi University	....
Assoc. Prof. (PhD) Ayhan Özgür TOY Yaşar University	....
Assist. Prof. (PhD) Önder BULUT Yaşar University	....
Assist. Prof. (PhD) Kamil Erkan KABAK İzmir University of Economics	....

Prof. (PhD) Yücel ÖZTÜRKOĞLU

Acting Director of the Graduate School

ABSTRACT

CONTROL AND PERFORMANCE ANALYSIS OF THREE STATION MAKE-TO-STOCK PRODUCTION LINES

YÜCEL, Özgün
PhD, Industrial Engineering
Advisor: Assist. Prof. (PhD) Önder BULUT
September 2021

For decades, studies on production systems cope with randomness, customer requirements, specific features of production processes, and system costs. Make-to-stock production enhances the level of customer service and facilitates balancing costs associated with production, inventory and shortages. This thesis considers production control and performance analysis of production systems consisting of three stations arranged in series in a make-to-stock environment. First, optimal control problems of production systems with single-machine stations, intermediate buffers and a finished goods buffer are studied. Demands arrive at the finished goods buffer according to a Poisson process, and those that cannot be immediately satisfied are lost. The system consisting of machines with Exponentially distributed processing times is defined as the basic model, while two-phase Coxian processing times are included to examine more complex systems with failure or rework occurrences in extended models. The objective is to find an optimal control policy that minimizes the long-run average system cost. The structure of optimal policies is revealed using the Markov decision process, and the study is enriched with various numerical examples. Secondly, an easy-to-apply alternative policy is introduced to overcome the challenges of finding optimal control policies. Computational results show that the proposed policy performs near-optimal in various instances. The settings with a relatively higher optimality gap are identified, and a modified version of the proposed approach is developed to improve the performance. This thesis lastly presents an exact Markovian analysis of production lines with two-phase Coxian processing times, parallel machines and finite buffers. Raw materials supply and finished goods demand are generated according to independent stationary Poisson processes. We model the line as a continuous-time Markov chain and propose recursive algorithms to generate the transition rate matrix.

Although the general recursive form is specific to 3-station 4-buffer lines, routines for calculating the number of states and generating the states work for any M-station (M+1)-buffer systems. The developed model allows us to obtain steady-state distribution and performance metrics such as throughput, the average number of items in the system, and average system cost. The proposed methodology can also be used as a decomposition block for the performance analysis of longer lines.

Key Words: serial production systems, optimal control, performance evaluation, two-phase Coxian distribution, Markovian analysis

ÖZ

ÜÇ İSTASYONLU STOĞA ÜRETİM HATLARININ KONTROLÜ VE PERFORMANS ANALİZİ

YÜCEL, Özgün
Doktora Tezi, Endüstri Mühendisliği
Danışman: Dr.Öğr.Üyesi Önder BULUT
Eylül 2021

Üretim sistemleri üzerine yapılan çalışmalar, on yıllardır rastgelelik, müşteri gereksinimleri, üretim süreçlerinin belirli özellikleri ve sistem maliyetleri ile başa çıkmaktadır. Stoğa üretim, müşteri hizmet düzeyini artırıp üretim, envanter ve kıtlıklarla ilişkili maliyetlerin dengelenmesini kolaylaştırır. Bu çalışma, stoğa üretim ortamında seri olarak düzenlenmiş üç istasyondan oluşan üretim sistemlerinin üretim kontrolü ve performans değerlendirmesini ele almaktadır. İlk olarak, tek makineli istasyonlar, istasyonlar arasında yer alan yarı mamül stokları ve bitmiş ürün stoğu içeren üretim sistemlerinin eniyi kontrol problemleri incelenmiştir. Taleplerin bir Poisson sürecine göre geldiği bu çalışmada, son ürün stoğundan anında karşılanamayan talepler için kayıp satış bedeli ödenir. Üstel olarak dağıtılmış işlem sürelerine sahip makinelerden oluşan sistem ana model olarak tanımlanırken, genişletilmiş modellerde arıza veya yeniden işleme oluşumları olan daha karmaşık sistemleri incelemek için iki fazlı Coxian işlem süreleri dikkate alınmıştır. Çalışmanın amacı, uzun vade ortalama sistem maliyetini en aza indiren eniyi kontrol politikasını bulmaktır. Markov karar süreci kullanılarak eniyi politikaların yapısı ortaya konmuş ve çalışma çeşitli sayısal örnekler ile zenginleştirilmiştir. İkinci olarak, eniyi kontrol politikaları bulmada karşılaşılan zorlukların üstesinden gelmek adına uygulaması kolay bir politika önerilmiştir. Önerilen politika birçok durumda eniyi politikaya yakın performans göstermektedir. Performansı eniyi politikadan uzak olan durumları iyileştirmek adına, önerilen yaklaşımın geliştirilmiş bir versiyonu da dikkate alınmıştır. Tez kapsamında yapılan son çalışma, iki-fazlı Coxian işlem süreleri, paralel makineler ve sonlu tamponlar içeren üretim hatlarının kesin bir Markov analizini sunar. Hammadde tedariği ve son ürün talebinin bağımsız Poisson süreçleri uyarınca geldiği bu problem, sürekli zamanlı bir Markov zinciri olarak modellenmiş ve geçiş hızı matrisini oluşturmak için özyinelemeli algoritmalar önerilmiştir. Genel özyinelemeli form 3-istasyon 4-tampon sistemlerine özgü olmasına rağmen, durum sayısını hesaplama ve durumları üretme rutinleri herhangi bir M-istasyon (M+1)-tampon sistemi için çalışmaktadır. Geliştirilen model, kararlı durum dağılımını ve verim, sistemdeki ürün sayısı ve ortalama sistem maliyeti gibi performans ölçütlerini hesaplamaya olanak sağlar. Önerilen metodoloji, daha uzun hatların performans analizi için bir ayrıştırma bloğu olarak da kullanılabilir.

Anahtar Kelimeler: seri üretim sistemleri, eniyi kontrol, performans değerlendirmesi, iki-fazlı Coxian dağılımı, Markov analizi

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Assist. Prof. Önder BULUT for his valuable insights, guidance, and support he has provided me from my undergraduate years to the completion of this thesis.

I want to express my sincere gratitude to my thesis committee members Prof. Mehmet Cemali DİNÇER and Prof. Refik GÜLLÜ, for their valuable insights, support and for guiding me through all these years. Without their advice, this thesis would not have been completed.

I want to thank jury members Assoc. Prof. Ayhan Özgür TOY and Assist. Prof. Kamil Erkan KABAK for their valuable comments and contributions.

Special thanks to my office mates in the Department of Industrial Engineering at Yaşar University for their friendship, moral support and encouragement.

Very special thanks to my dear family for their continuous love. My deep gratitude to my darling husband, Dr Çağatay Yücel, for his immense support. I am very grateful to him for helping me overcome many obstacles and tough times throughout my thesis.

Özgün Yücel
İzmir, 2021

TEXT OF OATH

I declare and honestly confirm that my study, titled “CONTROL AND PERFORMANCE ANALYSIS OF THREE STATION MAKE-TO-STOCK PRODUCTION LINES” and presented as a PhD Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Özgün Yücel
19/08/2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

Cox-2	Two-phase Coxian
CTMC	Continuous Time Markov Chain
DP	Dynamic Programming
FG	Finished Goods
MDP	Markov Decision Process
MTS	Make-to-Stock
NI	No Intentional Idleness
NI-j	No Intentional Idleness Alternative-j (j=1,2,3,4,5)
SS	State Space
NS_M	Number of recurrent states of M-station line
V_M	State vector of M-station line
T^M	Transition rate matrix of M-station line

SYMBOLS:

M	Number of stations of the production line
α	Discount rate
β	Visiting probability of two-phase Coxian random variable
ν	Uniform transition rate

CHAPTER 1
INTRODUCTION

Production systems have been challenged between excess inventory, shortages and production-related costs for decades. A production system may be composed of a single-machine or parallel-machine station or stations arranged in series. Different processing time distributions, rework/repair characteristics, demand structures, capacity, and cost components are the main characteristics of production control problems. If incoming customer demand is not directly satisfied by a production facility, backordering/lost sales costs, or service level constraints are likely to be encountered through the shortage. Moreover, capacity constraints, randomness, failure-prone characteristics of production lines cause line inefficiency. The system characteristics directly affect the production/inventory strategies, problem modeling and computational effort. Analytical and simulation models have been developed to understand, optimize and make decisions about different types of systems. A vast amount of research has been conducted in modeling production-inventory systems to either find and implement optimal policies or to develop alternative mechanisms and evaluate the performance in the last four decades, although modeling of those systems is started in the 1950s. Control and performance analysis of serial production systems have been studied in a broad field, from automobiles and computers to large appliances.

This thesis investigates production systems with three workstations arranged in series by considering various configurations of machines, processing time distributions, capacity constraints, and raw-material supply structure. The research of production systems in this thesis is revealed in threefold: (i) The first study consists of optimal control problems of three-station production lines. The general setting comprises ample raw-material supply, intermediate buffers between stations and a finished goods buffer with no capacity restrictions. Demands for finished goods follow a Poisson process, and those that cannot be immediately satisfied are assumed to be lost. The objective is to find the optimal control policy that minimizes long-run average system cost composed of holding and lost sales costs. Optimal production control poses a challenge due to the curse of dimensionality. Thus, alternative approaches have become crucial. (ii) Secondly, an easy-to-apply alternative policy called *no intentional idleness* policy, and its variants are studied. The proposed policies reduce the production control burden, and the problems turn into parameter optimization within these policies. They also constitute a near-optimal solution to the optimal control problems. (iii) The last study presents an exact Markovian analysis of make-to-stock production lines with parallel machines and limited supply. The developed model allows obtaining steady-state distribution and performance metrics such as throughput, the average number of items in the buffers, and average system cost consisting of production, holding, and shortage costs.

Production control is the main tool for optimizing the performance of the systems. The control mechanisms are developed based on '*what to/ how to/ when to/ how much to produce*' management decisions. In general, production control strategies are classified as push or pull types. Mechanisms that are run by future demand forecasts are known as push-type. Due to the forecasts, production control is handled with a schedule. On the other hand, if real demand occurrences are imposed upon production control, the corresponding mechanism is called pull-type. The idea behind the pull control mechanisms is a just-in-time manufacturing philosophy that aims to reduce the production cost by eliminating waste (T. Ohno, 1988).

In this thesis, we adopt '*when to produce*' and '*how much to produce*' decisions to solve the optimal control problems of three-station make-to-stock production systems. The system is built to produce a single item with ample raw material supply, single-machine workstations, intermediate buffers and a finished goods buffer. Demands arrive at the finished goods buffer according to a Poisson process. The objective is to minimize the long-run average system cost composed of holding and lost sales costs.

The basic model of the control problem consists of machines with Exponential processing times. In addition, to examine even more complex systems, we incorporate real-world features where rework operations, machines handling multi-stage operations, or failure-prone characteristics could be observed in production lines. From the modeling perspective, we consider two-phase Coxian (Cox-2) processing times. A two-phase Coxian random variable consists of independent Exponential phases and a certain visiting probability from phase one to phase two. The first phase is considered as the main service, while the second phase as inspection, rework or remanufacturing operation. Furthermore, the service completion time distribution of a machine with Exponential processing times, times to failure and repair times is shown to be equivalent to two-phase Coxian (Altiok & Stidham, 1983).

The first contribution of this research is investigating optimal control problems of three-station make-to-stock lines. Optimal policies have been studied in various single-stage production systems for decades. The first study that uses MDP techniques in make-to-stock queues shows that a single critical level policy, *base stock policy*, is optimal to control a single-machine single-stage make-to-stock system with lost sales (Ha, 1997a). Another important finding states that the optimal policy of a parallel machine system with Exponential processing times and lost sales is a state-dependent base-stock policy (Bulut & Fadiloğlu, 2011). Once having looked at the problems of serial production systems, the optimal control studies have been known to be limited to two-station systems.

Major progress is achieved with the findings of (Veatch & Wein, 1994), revealing that optimal control of two-station tandem production system is maintained by switching curves. The study assumes that each station has Exponential processing times, a finished goods inventory meets a Poisson demand, and the objective is to minimize the holding and backordering costs. It is acknowledged that control problems are computationally intense due to the curse of dimensionality. However, solutions of certain multi-stage production models are feasible to be obtained with a reasonable computational load. We study optimal control problems of three-station production-inventory systems considering Exponential and phase-type processing times in Chapter 3. A dynamic programming formulation is developed for each model to achieve the optimal policy and long-run average system cost. The structure of the production policies is characterized by propositions and numerical experiments.

Optimal production control requires taking actions at each system state, which obstructs implementing the optimal policies as state spaces grow. As the dimensionality problem has been encountered while finding the structure of optimal policies, alternative control mechanisms have become a concentrated research field of production-inventory systems. Analyses under given production policies would relatively reduce the complexity of the problems. Pull type control policies have been widely studied in multi-stage production (Karaesmen & Dallery, 2000; Khojasteh &

Sato, 2015). Structural properties of more complex mechanisms such as generalized Kanban (Buzacott, 1989), flexible Kanban (Gupta & Al-Turki, 1997), extended Kanban (Dallery & Liberopoulos, 2000) are adaptive to the systems with variability; however, they require a higher number of parameters.

In the second part of our study, we propose an alternative policy called *no intentional idleness* with the idea of eliminating production control actions. In the proposed approach, machines operate whenever possible. Buffer capacities are assumed to be finite; hence production is interrupted due to blocking and starvation. A station is blocked if its service is completed, but there is no room in its downstream buffer. A station is starved if it is idle because its upstream buffer is empty. The proposed *no intentional idleness* (NI) policy is defined with three parameters, each representing the capacity of a buffer. The objective is to identify buffer capacities to minimize the long-run average cost. A simulation model is developed for the proposed policy, and optimal buffer capacities are obtained via exhaustive search over the state space of the policy.

The second contribution of the thesis falls into quantifying how good the proposed approach is against the optimal policy. The proposed policy has been recognized in the performance evaluation of serial production systems (Diamantidis et al., 2020; H. T. Papadopoulos et al., 1989, 1990). The long-run behavior of systems is observed under the given policy to achieve certain metrics such as throughput and inventory held. However, a performance comparison of optimal and proposed policies has not been carried out for multi-stage make-to-stock systems. Chapter 4 presents the implementation of the *no intentional idleness* policy for the basic model with Exponential processing times and the extended models with Cox-2 processing times. The performance evaluation is conducted with an extensive numerical study revealing that it performs near-optimal in various cases. Additionally, a set of variants of the proposed policy is considered to improve the performance of the instances when the policy deteriorates.

Production systems have been analyzed for performance evaluation using several measures like throughput, machine utilization, the number of items in the system, or service level. Performance analysis has been an integral part of production/ manufacturing systems due to rapidly emerging industries. Analytical and simulation models have been developed with the main objectives of throughput maximization, buffer size determination, buffer allocation to optimize a reward function. Throughput, the average service completion rate of the system (Ross, 2014), is a well-accepted performance measure in practice. However, the general assumption in performance analysis indicates no finished goods buffer to meet the demand. Under this assumption, multi-stage systems with single machines (H. T. Papadopoulos et al., 1990; Vidalis & Papadopoulos, 1999) have been studied. In addition, there are a few studies considering parallel machines with exact (Diamantidis & Papadopoulos, 2009) and approximate (Van Vuuren et al., 2005) analyses.

In terms of performance evaluation, the third contribution involves an exact Markovian analysis of production lines with parallel machines having two-phase Coxian processing times. The setting is designed to scrutinize raw material replenishment and intermediate and finished goods buffers with capacity limits. Raw material supply and demand for finished goods are generated according to independent stationary Poisson processes. The system is modeled as a continuous-time Markov chain, and the transition rate matrix is recursively generated. The general recursive form is developed up to 3-station 4-buffer systems. However, the number of states calculation and state generation routines work for any M-station (M+1)-buffer systems. The steady-state distribution of the developed model is obtained using eigenvalue decomposition, which allows finding performance metrics such as throughput, the average number of items in the buffers, and average system cost consisting of production, holding, and shortage costs. Furthermore, an extensive numerical study demonstrates the impact of buffer capacities, parallel machines, and production rates on system performance. The exact analysis provided in Chapter 5 could also be used as the decomposition block for the performance analysis of longer lines.

Figure 1.1 presents the research structure of the thesis. Analytical and simulation models are developed for control problems and performance evaluation of serial production systems within the scope of our research. Due to the computational load, the literature presents the exact analysis of production systems covering a limited number of machines with serial/parallel structures (Diamantidis & Papadopoulos, 2009; Veatch & Wein, 1994). However, our study covers optimal control problems of three-station lines with both memoryless and phase-type processing times. The performance evaluation of three-station lines seeking exact analysis considers parallel-machine stations with two-phase Coxian processing times. Another important aspect of the thesis is considering the make-to-stock environments in which the objective function covers the finished goods-related costs. It should be noted that the parallel-machine and phase-type processing times assumptions increase the complexity of the models. Although our problems acknowledge the cost of exact techniques, our numerical studies reveal a range of tractable experiments.

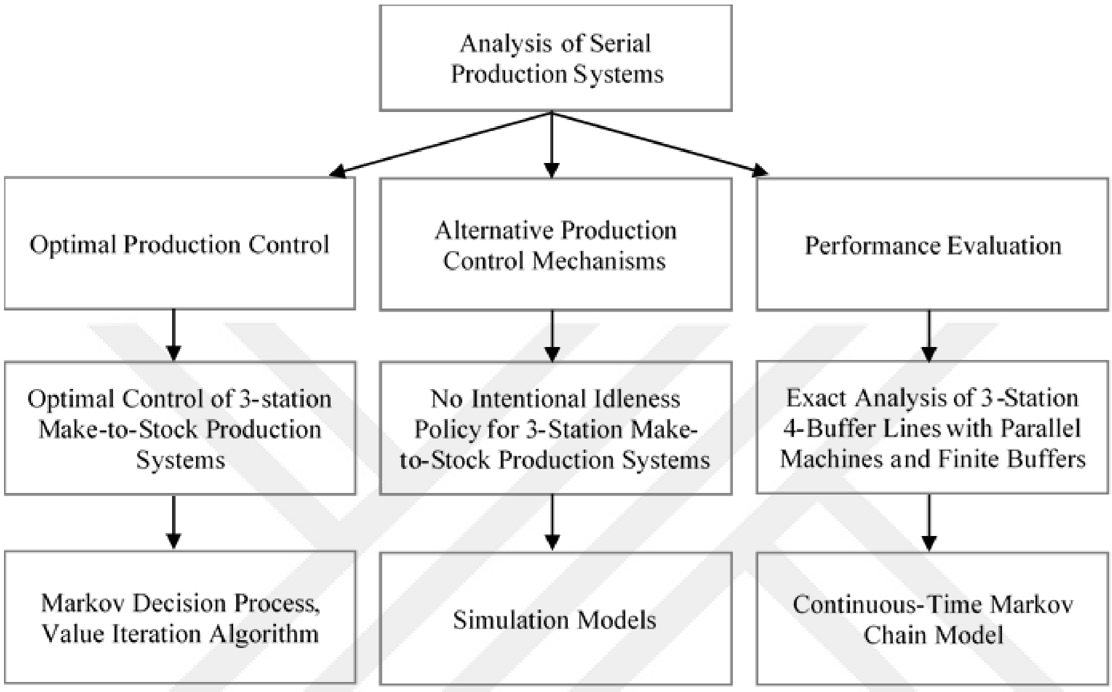

Figure 1.1. Research structure of the thesis

From the optimal control framework, we obtain a value iteration solution of a Markov decision process. Dynamic programming numerically finds an optimal policy, as we provide in Chapter 3. With the help of dynamic programming results, insights into the optimal policy structure can be revealed, and policy characterization can be done.

Nevertheless, simulation is still attractive due to advances in software development and the facilities that such programs provide for analyzing complex systems. In the event that finding optimal policies requires extreme effort or are not practically applicable, easy-to-apply approaches can be proposed and imposed on the production settings. Simulation tools are effective not only for modeling a problem but also for verification purposes. A simulation model can be developed to quantify how good a proposed strategy is, as is done in Chapter 4.

Scientific research is prevalently accompanied by simulation studies applicable for many systems, including production lines, with the help of artificial intelligence over a decade. However, we rely on that the exact analytical results of computationally tractable systems in this line of research nourish the simulation studies. We focus on the algorithmic construction of a matrix generation to conduct performance analysis for make-to-stock production lines with parallel machines, phase-type processing times, limited buffers and finite supply, as it is presented in Chapter 5.

The perspective in our study and the exact analysis we conduct would contribute to the literature in two ways. First, our study provides a theoretical background up to 3-station 4-buffer lines with failure-prone parallel machines. A secondary contribution is maintaining a longer decomposition block than those considered in the literature (Diamantidis et al., 2020; Van Vuuren et al., 2005). Time complexities of the proposed algorithms are also discussed.

The remainder of this thesis is organized as follows. Chapter 2 presents a literature review for the control mechanisms, performance analysis, and design problems of production systems. Optimal control problems of three station make-to-stock lines are defined in Chapter 3. Problem formulation and analysis of optimal control policy for the basic model with Exponential processing times are provided in Chapter 3.1. Chapter 3.2. presents the studies of the extended models with two-phase Coxian processing times. Chapter 4 proposes an alternative production control mechanism to the optimal policy. The policy implementation and a comprehensive numerical study for performance analysis are provided in Chapter 4.1. Chapter 4.2 presents the variants of the proposed policy with improvements. Markovian analysis of make-to-stock production lines with parallel machines, limited buffers and finite raw-material supply is presented in Chapter 5. Chapter 5.1 gives a system description. A recursive method to calculate the number of states is provided in Chapter 5.2 as the first routine of the general algorithm structure. The second routine for state generation is presented in Chapter 5.3. The main algorithm is developed in Chapter 5.4. Chapter 5.5 provides numerical experiments. Lastly, concluding remarks and suggestions for future research are provided in Chapter 6.

CHAPTER 2
LITERATURE REVIEW

This chapter provides a review of production control and performance evaluation problems of serial production systems. Production control approaches manage the tradeoff between production-related costs, costs of excess inventory and shortages. In a make-to-stock (MTS) environment, optimal production control requires starting production at the right time and producing with the optimum number of channels (servers, lines, or machines) to build up sufficient inventory. In this research channel, the majority of the analyses are based on queueing theory techniques and Markov Decision Process (MDP). Markovian structure of problems enables the development of MDP formulation to control MTS systems. The first studies using MDP techniques are (Veatch & Wein, 1994) for backorder and (Ha, 1997a) for lost sale cases. For the serial production lines, to the best of our knowledge, optimal control studies based on MDP formulations are limited to two-station production lines.

Moreover, pull-type control mechanisms have been studied where actual demand occurrences are used instead of demand forecasts. Just-in-Time manufacturing philosophy, the driving force behind pull mechanisms, has gained importance since Taiichi Ohno aimed at meeting customer demands with minimum delays (T. Ohno, 1988). Since then, several control policies have been proposed where Kanban, CONWIP and Base Stock are well-known pull mechanisms. The literature on optimal control and other control mechanisms for production lines are presented in Chapter 2.1.

Apart from optimal control, routines can be developed to control production, buffer sizes or production capacities of the machines. Buffer allocation problem, which has been attracted for years, is one of the main design problems of production systems. Several optimization techniques have been used for different configurations of production lines in the literature of buffer allocation problem. On the other hand, a vast amount of research has been conducted on optimizing throughput and average system cost composed of holding, shortage, and production costs. Performance evaluation of production systems has been undertaken mostly based on the throughput metric as it is one of the most acknowledged performance measures in practice (J. Li et al., 2009). Although performance evaluation of production systems has mostly been based on steady-state analysis, studies considering transient behavior have also been presented in the literature. Another assumption that has been commonly held is to have a single machine/server at each stage of the line. Only a few studies consider the exact analysis of multi-server lines (Diamantidis & Papadopoulos, 2009). Chapter 2.2 provides a review of the performance evaluation of production lines in several settings.

2.1. Optimal Control and Other Control Mechanisms in Serial Production Systems

Control mechanisms can be explained into two main streams as single-stage and multi-stage production systems. In early studies of optimal production control in single-stage systems, problems are mostly formulated as a queueing model, and the analyses are mostly based on queueing theory techniques (Gavish & Graves, 1980, 1981). In the studies of (Gavish & Graves, 1980, 1981), the system is modeled as M/D/1 and M/G/1 make-to-stock queues with backorders, and the optimal production policy is shown as a two-critical-number policy. The first study that uses MDP techniques in single-stage systems is based on $M/M/1$ make-to-stock queues with multiple demand classes and lost sales (Ha, 1997a), which the optimal control policy is proven as base-stock. The problem of (Ha, 1997a) is then analyzed in a backordering case (Ha, 1997b). Erlang processing time extensions of (Ha, 1997a) are also studied for both lost sales (Ha, 2000) and backordering (Gayon et al., 2009) cases. A model with multiple parallel servers, Exponential processing times and lost sales in single-stage make-to-stock systems is first presented in the work of (Bulut & Fadiloğlu, 2011). The problem is modeled as $M/M/s$ make-to-stock queue and the optimal production policy is shown as a state-dependent base-stock. A two-phase Coxian processing times extension of (Bulut & Fadiloğlu, 2011) is proposed in (Yücel & Bulut, 2019), the optimal production policy is numerically characterized and an easy-to-apply production policy is proposed.

In multi-stage production systems, both queueing models and MDP approaches continue to contribute to the literature. Tandem queueing systems with Poisson arrivals, Exponential service times and finite capacity waiting rooms are presented by

(K. Ohno & Ichiki, 1987a). A modified policy iteration algorithm is proposed to find an optimal control policy that minimizes the expected discounted cost of the system. However, it is stated that the form of optimal control policy of the system has no distinct structure. A two-station tandem queueing system with no intermediate buffer is studied by (Chao et al., 1997). The objective is to maximize expected discounted profit with control of arrival and departure processes. It is shown that the optimal policy has a threshold type structure. (Veatch & Wein, 1994) is the first study considering make-to-stock production systems with two stations in tandem having Exponential processing times, Poisson demands and backorders. The objective is to minimize long-run average system cost composed of holding and backordering costs, and a dynamical programming formulation is developed to obtain numerical results. It is shown that the optimal control policy is defined by switching curves where state space is divided into two as idle and busy sets. It is further established in (Veatch & Wein, 1994) that some policies are shown to be optimal under certain conditions, and the base stock policy is never optimal.

The recent studies of this line of research also follow two station systems. To the extent of our knowledge, the structure of the optimal control policies has been identified only for two stations in tandem in multi-stage production systems. The optimal production and rationing policy of a two-station tandem system is presented in (Xu et al., 2017), and optimal policy is characterized as dynamic switching curves. The system is built with Exponential processing times, partial-batch production, intermediate and finished goods buffers, bulk demands and lost sales, and the objective is to maximize the expected profit. Examples of real-life applications for production systems are provided. (Papachristos & Pandelis, 2020) study optimal server assignment in a two-stage Markovian tandem queueing system and present structural properties of optimal policies for discounted and average cost. An optimal pricing problem modeled as a tandem queueing system is studied in (Wang et al., 2020a). The structure of the optimal policy is characterized to maximize the expected revenue.

Another point is the evaluation of alternative control policies. Pull type production proposes various well-known control mechanisms. In pull type control, information of actual demand occurrences is considered instead of demand forecasts. The idea behind the pull control mechanisms is the Just-in-Time manufacturing philosophy proposed by (T. Ohno, 1988). Kanban, CONWIP and Base-stock have been well-known pull control policies. Kanban control policy sends demand requests from finished product inventory to the upstream stations using production authorization cards. The first studies of Kanban modeling are presented by (Tabe et al., 1980) and (Kimura & Terada, 1981). Then, various extensions of Kanban policy has been introduced to the literature, such as generalized Kanban (Buzacott, 1989), flexible Kanban (Gupta & Al-Turki, 1997), extended Kanban (Dallery & Liberopoulos, 2000). CONWIP, proposed by (Spearman et al., 1990), controls the amount of work-in-process in the system and aims to keep it constant. CONWIP policy can be defined as a single-stage Kanban, which also uses production authorization cards. Another control mechanism is a base-stock policy, which keeps the maximum number of finished products in a single production facility and has shown to be optimal in several settings (Ha, 1997a, 1997b). In multi-stage production systems, a base-stock level is defined for each production stage. It is likewise production authorization cards that are limited for each stage in Kanban. However, these policies differ while transferring demand requests to upstream stages. Performance comparisons of the pull policies have been widely studied in the literature (Bonvik et al., 1997; Duri et al., 2000; Karaesmen & Dallery, 2000; Khojasteh & Sato, 2015).

Since the Just-in-Time idea is suitable to be operated in environments where machines are synchronized/free from non-value-added operation times, its performance would be affected by variations in production times and demand process. Therefore, alternative Kanban policies, which dynamically change the number of cards in the system, have been developed. Although these policies are more likely to fit into a dynamic environment, more parameters are necessary to define the policies. An adaptive Kanban control is proposed in (Tardif & Maaseidvaag, 2001) for backordering cases. The proposed policy allows the system to release or capture kanbans cards depending on the system state. An adaptive Kanban mechanism for multi-stage systems is presented in (Sivakumar & Shahabudeen, 2008). (Xanthopoulos et al., 2018) define an adaptive Kanban production control for single-stage multi-server systems having different demand levels due to seasonality. A comparison of optimal, classical Kanban and extended Kanban policies is provided.

Furthermore, learning-based approaches have been studied in the literature. Reinforcement learning is a part of machine learning where an agent interacts with the environment and selects actions to maximize the future reward. (Xanthopoulos et al.,

2008) consider serial production lines with backorders. Control policies are derived with an average reward reinforcement learning algorithm, and it is shown that the derived algorithm outperforms Kanban, Base Stock and CONWIP. The study of (Paternina-Arboleda & Das, 2001) considers a four-station production line and solve the control problem to minimize the average WIP of the system. The setting is defined with random processing times, times to failures and repair times, and lost sales. The performance of the algorithm is compared with the pull control policies. In (Xanthopoulos et al., 2019), optimal adaptive control policies for CONWIP-type manufacturing systems are studied. The reinforcement learning approaches are compared to the special cases of CONWIP and Kanban systems.

(Lage Junior & Godinho Filho, 2010) present literature review and classification for the Kanban system. (Kumar & Panneerselvam, 2007) review the Just-in-Time Kanban system and discuss the performance measures in many cases. A review of the CONWIP production control system is presented in (Framinan et al., 2003).

2.2. Performance Evaluation of Serial Production Systems

Performance analysis of production lines has been essential due to investment of machinery, revenue/cost account, randomness, variability in production, among other motives. The behavior of systems is observed over some period to determine certain measures such as throughput, inventory held or service level. In the general framework, steady-state analysis has been widely considered. However, performance evaluation problems with transient analysis have also been studied in the literature (Gökçe et al., 2012; Meerkov et al., 2009; Meerkov & Zhang, 2008; Zhang et al., 2013). Analyses are mostly conducted while machines operate whenever possible, which we call *no intentional idleness* (NI) policy. This approach eliminates the production control and allows for performance analyses of production lines, and production is interrupted if there is blocking, starvation or failures at stations.

Throughput analysis has been vital in performance evaluation (J. Li et al., 2009). On the other hand, a vast amount of research has been conducted to optimize system cost, machine, and inventory in addition to throughput. There is a broad literature on the analysis of production lines. Recently, (C. T. Papadopoulos et al., 2019) provide a comprehensive review of Markov models of manufacturing systems. Earlier studies on performance analysis of throughput focus on two station tandem systems with intermediate buffers under different service time distributions: Exponential (Hatcher, 1969), Phase-type (Buzacott & Kostelski, 1987), Erlang (Berman, 1982), and General (Gershwin, 1987). Processing times as a mixture of Exponential and Erlang are also studied (Hillier & Boling, 1967; Rao, 1975).

Serial line studies are evolved into settings where there are more than two stations. However, the assumption 'single machine at each station' stayed. Reliable production lines with Exponential and Erlang processing times are studied by (H. T. Papadopoulos et al., 1989, 1990). (Perros & Altiok, 1986) propose an approximation of the system's steady-state distribution for Coxian processing times. For the same processing time distribution, (Vidalis & Papadopoulos, 1999) perform a Markovian analysis to obtain the transition matrix of two station lines where the first station is never starved and the last station is never blocked. (Heavey et al., 1993) provide an approximate steady-state analysis for unreliable multistation systems.

In the further studies of this line of research, researchers focus on the systems with parallel machine tandem stations. (Diamantidis et al., 2007) provide an approximate analysis based on decomposition for parallel machine stations with Exponential production times. An exact analysis of a two-station one-buffer system with unreliable parallel servers is conducted by (Diamantidis & Papadopoulos, 2009). (Van Vuuren et al., 2005) and (Diamantidis et al., 2020) further propose decomposition-based approximations for the performance analysis of multi-server production lines with general and unreliable exponential service times, respectively. Two-station tandem queues with multiple servers and phase-type service times are studied by (Baumann & Sandmann, 2017). Furthermore, modeling real manufacturing systems has also been studied where stations might consist of single or parallel machines (Patchong et al., 2003; Patchong & Willaeys, 2001).

Furthermore, the buffer allocation problem has been extensively researched. Reliable lines with Exponential processing times are studied by (Meester & Shanthikumar, 1990) and (Spinellis & Papadopoulos, 2000). However, the effect of breakdowns at stations are presented to the literature earlier (Buzacott, 1971; Conway et al., 1988; Freeman, 1964). An allocation of a periodic pull production system with M stages in tandem is studied to maximize throughput, and the system's dynamics is characterized in the study of (Kirkavak & Dinçer, 1999). A system with M operation stations with M-1 intermediate buffers with single servers is studied by (Altiok & Stidham, 1983).

Ample supply for raw materials and an infinite capacity of finished products buffer are assumed, and production continues whenever possible, i.e. production is controlled with no intentional idleness policy. However, the study defines a reward/cost-based objective function and optimization is done based on buffer capacities. Each station is subject to operation dependent breakdowns. It is proven using Laplace transform that a machine with Exponential processing times, times until failure and repair times are equivalent to one having Cox-2 processing times. For the given service completion time distribution, a steady-state analysis is performed to determine the optimal allocation of buffer capacities that maximizes long-run average profit. The result presented in (Altiok & Stidham, 1983) is then used to model flow lines with failure-prone machines (Helber, 2005; Hillier & So, 1991). Furthermore, flow lines with phase-type processing times (Yamashita & Altiok, 1998) and approximate analysis for general distributions (Altiok & Ranjan, 1989; So, 1997) are also studied. Over the recent years, throughput analysis (L. Li, 2018; Tan & Lagershausen, 2017) and buffer allocation problem (Shi & Gershwin, 2016; Weiss et al., 2018) still constitute the main research themes on production systems. Moreover, recent studies still focus on two-station single server lines (Gebennini et al., 2015; Matta & Simone, 2016; Tolio & Ratti, 2018) since two-station lines can be used as building blocks to analyze larger systems. Table 2.1 presents a classification of the related literature of production systems.

Table 2.1. Classification of the literature

Optimal Control of Production-Inventory Systems	
Single-station optimal control problems	
(Gavish & Graves, 1980, 1981; Ha, 1997a, 1997b, 2000)	Control problems of a single machine systems
(Bulut & Fadiloğlu, 2011)	Control problems of parallel machine systems
(Yücel & Bulut, 2019)	Control problems with Cox-2 processing times
Multi-station optimal control problems	
(K. Ohno & Ichiki, 1987a)	Optimal control for a tandem queueing system
(Veatch & Wein, 1994)	Optimal Control of a two-station tandem system
(Xu et al., 2017)	Optimal policies of a two-stage tandem system
(Papachristos & Pandelis, 2020)	Optimal server assignment in a two-stage tandem queueing system
(Wang et al., 2020a)	Revenue maximization in two-station systems

Table 2.1 (cont'd). Classification of the literature

Pull control mechanisms	
(Spearman et al., 1990)	CONWIP: a pull alternative to Kanban
(Buzacott, 1989; Dallery & Liberopoulos, 2000; Gupta & Al-Turki, 1997)	Variants of Kanban systems
(Bonvik et al., 1997; Duri et al., 2000; Karaesmen & Dallery, 2000; Khojasteh & Sato, 2015)	Pull control mechanisms in multi-stage systems
(Sivakumar & Shahabudeen, 2008; Tardif & Maaseidvaag, 2001; Xanthopoulos et al., 2018)	Adaptive Kanban mechanisms
(Paternina-Arboleda & Das, 2001; Xanthopoulos et al., 2008, 2019)	Reinforcement learning-based control of pull production systems
Performance Analysis of Single-machine Multi-station Production Systems	
(Berman, 1982; Buzacott & Kostelski, 1987; Gershwin, 1987; Hatcher, 1969; Matta & Simone, 2016)	Throughput analysis of two-station lines with an intermediate buffer
(Hillier & Boling, 1967; Rao, 1975)	Throughput analysis of two-station lines with an intermediate buffer with mixture service times
(Gebennini et al., 2015, 2017)	Two-machine one-buffer models with restart policy
(Vidalis & Papadopoulos, 1999)	Markovian analysis of two-station lines with Cox-2 service times
(H. T. Papadopoulos et al., 1989, 1990)	Markovian analysis of multi-station production lines with intermediate buffers
(Heavey et al., 1993)	The throughput rate of multistation unreliable production lines
(Perros & Altiok, 1986)	Approximate analysis to steady-state distribution of tandem queues with Cox-2 service times
(Kirkavak & Dinçer, 1999)	Throughput maximization in pull systems
(Tan & Lagershausen, 2017)	Output dynamics of systems subject to blocking
(Gökçe et al., 2012; Meerkov et al., 2009; Meerkov & Zhang, 2008; Zhang et al., 2013)	Transient analysis of serial lines
(L. Li, 2018)	Detecting throughput bottlenecks
Buffer Allocation Problem in Single-machine Multi-station Production Systems	
(Buzacott, 1971; Conway et al., 1988; Freeman, 1964)	Analysis of the effect of breakdowns at stations
(Meester & Shanthikumar, 1990; Spinellis & Papadopoulos, 2000)	Analysis of reliable lines with Exponential processing times
(Altiok & Stidham, 1983; Helber, 2005; Hillier & So, 1991)	Modeling flow lines with failure-prone machines
(Altiok, 1989; Yamashita & Altiok, 1998)	Modeling flow-lines with phase-type distributions

Table 2.1 (cont'd). Classification of the literature

Buffer Allocation Problem in Single-machine Multi-station Production Systems	
(Altiok & Ranjan, 1989; Powell, 1994; So, 1997)	Approximate analysis of flow-lines with general service time distributions
(Liberopoulos, 2020)	Optimal buffer allocation under different production control policies
(Shi & Gershwin, 2016)	A segmentation approach in large systems
(Weiss et al., 2018)	Optimization in flow lines with limited supply
Analysis of Parallel-machine Multi-station Production Systems	
(Diamantidis et al., 2007)	Decomposition for large production systems
(Diamantidis & Papadopoulos, 2009)	Exact analysis of a two-workstation one-buffer flow line
(Diamantidis et al., 2020; Van Vuuren et al., 2005)	Decomposition-based approximate analysis with finite buffers
(Baumann & Sandmann, 2017)	Performance analysis of multi-server 2-station tandem queues with phase-type service times
(Patchong et al., 2003; Patchong & Willaeys, 2001)	Modeling real manufacturing systems with single or parallel structures
Review Papers	
(C. T. Papadopoulos et al., 2019)	Timed Markov models of manufacturing systems
(Weiss et al., 2019)	The buffer allocation problem in production lines
(J. Li et al., 2009)	Throughput analysis of production systems
(Framinan et al., 2003; Kumar & Panneerselvam, 2007; Lage Junior & Godinho Filho, 2010)	Pull production control systems

CHAPTER 3
OPTIMAL CONTROL PROBLEMS OF THREE-STATION MAKE-TO-STOCK PRODUCTION LINES

This chapter considers optimal control problems of production systems consisting of three stations arranged in series, intermediate buffers between stations and a finished goods buffer. The structure of optimal policies for single-station (Bulut & Fadiloğlu, 2011; Gayon et al., 2009; Ha, 1997a, 1997b, 2000) and two-station (K. Ohno & Ichiki, 1987b; Veatch & Wein, 1994; Wang et al., 2020b; Xu et al., 2017) systems has been discussed in the literature. However, to the extent of our knowledge, optimal control problems have not been studied for three stations in tandem. This chapter presents a characterization of optimal production control policies for three-station tandem lines in various settings. Further to that, the study is enriched with numerical experiments.

The setting contains single-machine stations, ample raw-material supply to produce a single item. It is assumed no capacity restrictions of intermediate buffers and the finished goods buffer. Since each station has a single machine, this chapter could use the terms *machine* and *station* interchangeably. Demands for finished goods are generated according to a Poisson process, and those who cannot immediately be satisfied from the finished goods buffer are lost. The objective is to find an optimal control policy that minimizes the long-run average system cost. The system cost is composed of holding and lost sales costs. Due to the Markovian structure of the problems, a dynamic programming formulation is developed, and minimum system cost is obtained via a value iteration algorithm.

We first model a system considering the Exponential processing times of machines, which is called the basic model. The effects of system parameters on optimal policies are examined by considering different production rates, demand rates, and lost sales costs, as shown in Chapter 3.1. In extended models, we incorporate real-world features where failure, rework, or repair can happen in production lines. Therefore, we consider two-phase Coxian (Cox-2) processing times. A Cox-2 random variable has two independent Exponential phases, and there is a certain visiting probability from phase one to phase two. A busy machine with Cox-2 processing times would be either at the first or second phases at a given time. In that sense, the first and second phases can be considered as the main operation and inspection/rework/remanufacturing operation, respectively. Also, it is known that the service completion time distribution of a machine having Exponential processing times, times to failure and repair times is equivalent to two-phase Coxian (Altiok & Stidham, 1983). One can also consider the first phase of Coxian as the service time and the second phase as the repair time of the unreliable exponential machine. The probability that the first event is a breakdown corresponds to the certain visiting probability of the Coxian-2 random variable. This result has been used to model flow lines with failure-prone machines (Helber, 2005; Hillier & So, 1991). Optimal policies are studied and presented for extended models in Chapter 3.2.

3.1. Basic Model with Exponential Processing Times

We consider a make-to-stock production line with Exponential processing times (see Figure 3.1). Each station has a single machine with an Exponential production rate of $\mu_j, j = 1,2,3$. It is assumed that the raw material supply is ample, and demand arrives according to a Poisson process with rate λ. A lost sales cost c is incurred for each demand that cannot be met from the finished goods (FG) buffer. Holding cost rate h_j is charged for items in j^{th} buffer (and the item being produced its downstream station). System parameters of the basic model are shown in Table 3.1.

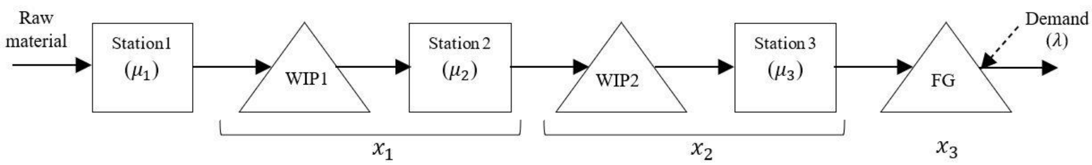

Figure 3.1. A make-to-stock production line with Exponential processing times

Table 3.1. System parameters of the basic model

j	Station index, j=1,2,3
λ	FG demand rate
c	Lost sales cost
h_j	Holding cost rate of buffer j (and the item being produced its downstream station)
μ_j	Production rate of station j

The system state is defined as a vector of three variables. Let $\vec{X}(t) = \big(X_1(t), X_2(t), X_3(t)\big)$ be the state vector where $X_j(t), j \in \{1,2\}$ is the number of elements in j^{th} buffer plus the item that is currently being produced its downstream station, and $X_3(t)$ is the number of items in the finished goods (FG) buffer at time t. The state vector does not include a variable for station-1 due to the single machine assumption and the memoryless property of the processing time distribution. The system state space is expressed as follows:

$$SS = \left\{ \big(X_1(t), X_2(t), X_3(t)\big) \mid X_j(t) \in Z^+ \cup \{0\}, j = 1,2,3 \right\} \tag{1}$$

The events are defined as production completions at stations and demand arrivals. Thus, the control policy requires whether or not to produce at each station. Let $a_j(t) \in \{0,1\}, j = 1,2,3$ be a control variable that keeps the status of machine-j at time t: the machine is busy if $a_j(t) = 1$, otherwise it is idle. The structure of the problem is Markovian due to Exponential processing times and inter-demand arrival times assumptions. The Markovian property refers to the memoryless property of the process. Due to the memoryless nature of the model, control decisions at time t will depend on only the current state of the system. Through the Markovian property, decisions can be made at a production completion or a demand arrival. For this reason, the system state definition can be used as independent from the time dimension.

Given a control policy π, the process $\{X_1^\pi(t), X_2^\pi(t), X_2^\pi(t) | t \geq 0\}$ constitutes a continuous-time Markov chain. The discrete-time equivalent of the problem is obtained using the uniformization technique as proposed by (Lippman, 1975). The uniform transition rate ν could be defined as greater than or equal to the summation of all transition rates in the system, which we define as $\nu = \lambda + \sum_{j=1}^{3} \mu_j$. As it is stated in (Bertsekas, 2000), an optimal control policy π^* exists and can be obtained through the solution of the below cost-to-go function:

$$\begin{aligned} V(x_1, x_2, x_3) = \frac{1}{\nu + \alpha} \min_{a_1, a_2, a_3} \{ & h_1 x_1 + h_2 x_2 + h_3 x_3 + \lambda L(x_1, x_2, x_3) \\ & + a_1 \mu_1 V(x_1 + 1, x_2, x_3) + a_2 \mu_2 T_1(x_1, x_2, x_3) + a_3 \mu_3 T_2(x_1, x_2, x_3) \\ & + \big(\textstyle\sum_{j=1}^{3} (1 - a_j)\, \mu_j\big) V(x_1, x_2, x_3) \} \end{aligned} \tag{2}$$

where α is a discount rate, and T_1, T_2 are production-related operators such that

$$T_1(x_1, x_2, x_3) = \begin{cases} V(x_1 - 1, x_2 + 1, x_3), & x_1 > 0 \\ V(x_1, x_2, x_3), & x_1 = 0 \end{cases}$$

$$T_2(x_1, x_2, x_3) = \begin{cases} V(x_1, x_2 - 1, x_3 + 1), & x_2 > 0 \\ V(x_1, x_2, x_3), & x_2 = 0 \end{cases}$$

L is a lost sales operator such that

$$L(x_1, x_2, x_3) = \begin{cases} V(x_1, x_2, x_3 - 1), & x_3 > 0 \\ c + V(x_1, x_2, 0), & x_3 = 0 \end{cases} \tag{3}$$

The optimal discounted cost V^* of the problem satisfies Bellman's equation given in (2). The minimization operator is based on production control variables a_1, a_2, a_3. Production completion at station-j occurs with probability $\frac{a_j \mu_j}{v+\alpha}$ and the term $\left(\sum_{j=1}^{3}(1 - a_j)\, \mu_j\right) V(x_1, x_2, x_3)$ is added for fictitious self-transition due to uniformization. The lost sales operator L is defined in (3): an incoming demand is immediately satisfied if there is an item in the finished goods buffer; otherwise, it is rejected, which causes a self-transition, and the lost sales cost c is charged. As stated in (1), state variables are not upper-bounded. Thus, the solutions are based on a truncated state space of x_1, x_2, x_3 such that the optimal policy and the optimal long-run average cost of the system are not affected.

Initialize $V^0(state)$ arbitrarily for each state

$k = 0$

$while\ (difference > \varepsilon)$

  $k = k + 1$

    Repeat for each state:

      Repeat for each possible control action:

        Calculate $V^k_{candidate}(state(action))$

      $V^k(state) = \min_{action} V^k_{candidate}(state(action))$

    $difference = max\ \left|\frac{V^k\,(state)}{k} - \frac{V^{k-1}\,(state)}{k-1}\right|$

$end\ while$

Figure 3.2. Pseudocode of the value iteration algorithm

The DP formulation is developed under the expected discounted cost criterion. However, system performance has been widely measured under the infinite horizon average cost criterion (Ha, 1997a; Karaesmen & Dallery, 2000; Veatch & Wein, 1994). We conduct numerical experiments under the average cost criterion, eliminating both the discount factor and the system's dependence on the initial state. We apply a value iteration algorithm to the formulation provided in (2)-(3) by setting discount rate α to zero (see Figure 3.2). Firstly, the value function at step 0 (V^0) is arbitrarily initialized. Then, for each state, the system cost $V^k_{candidate}$ is calculated considering control actions. The value function is updated based on the best next state with the minimization operator. The long-run average system cost is calculated as the convergent ratio of the value of the optimal cost-to-go function and the number of iterations. The value iteration algorithm is terminated when the absolute value of the difference between average costs in two consecutive steps is less than ε. The algorithm is developed in MATLAB 2018b program.

Table 3.2. Control action representation of the optimal policy

Control Action	Do not produce at all	Produce only at station 1	Produce only at station 2	Produce only at station 3	Produce at stations 1 and 2 but not 3	Produce at stations 2 and 3 but not 1	Produce at stations 1 and 3 but not 2	Produce at all
Label	0	1	2	3	4	5	6	7
Color								

Table 3.3. An example of the optimal policy with $\lambda = 4$

x2 \ x3	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1	0	4	4	4	4	4	4	2	0	4	4	4	4	4	2	2	0
1	6	6	6	6	6	6	0	0	7	7	7	7	7	5	0	0	7	7	7	7	5	5	0	0
2	6	6	6	6	6	3	0	0	7	7	7	7	5	3	0	0	7	7	7	5	5	3	0	0
3	6	6	6	6	3	3	0	0	7	7	7	3	3	3	0	0	7	5	5	5	3	3	0	0
4	6	6	6	3	3	3	0	0	7	5	3	3	3	3	0	0	5	5	3	3	3	3	0	0
5	6	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0
6	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0
7	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0
	x1=0								**x1=1**								**x1=2**							

x2 \ x3	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	4	4	4	2	2	2	2	0	4	2	2	2	2	2	2	0	2	2	2	2	2	2	2	0
1	7	7	5	5	5	5	0	0	5	5	5	5	5	5	0	0	5	5	5	5	5	5	0	0
2	5	5	5	5	5	3	0	0	5	5	5	5	5	3	0	0	5	5	5	5	5	3	0	0
3	5	5	5	5	3	3	0	0	5	5	5	5	3	3	0	0	5	5	5	5	3	3	0	0
4	5	5	3	3	3	3	0	0	5	5	3	3	3	3	0	0	5	5	3	3	3	3	0	0
5	3	3	3	3	3	3	0	0	5	3	3	3	3	3	0	0	5	3	3	3	3	3	0	0
6	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0
7	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0	3	3	3	3	3	3	0	0
	x1=3								**x1=4**								**x1=5**							

Let A be the action space for the production control actions. Since the actions to be taken are *produce* and *do not produce* for each station, A is a vector of size $2^3 = 8$. It is denoted that $A = \{0,1,..,7\}$ and the details of the control actions are shown in Table

3.2. In addition, a color is assigned to each action to represent the optimal policy. To the extent of our knowledge, the structure of the optimal control policy of three station make-to-stock lines with lost sales has not been studied yet. Thus, a series of computational experiments are designed and conducted to analyze the optimal policy of the basic model. The experiments are carried out on a Core i7, 2.80 GHz, 16 GB RAM computer. In the experiments, the error bound ε is set to 10^{-2}.

First, a base case scenario is defined with holding cost rates of $[h_1, h_2, h_3] = [1, 1.5, 2]$, lost sales cost of $c = 50$, identical production rates of $\mu_1 = \mu_2 = \mu_3 = 10$ and demand rate of $\lambda = \{1,2..,10\}$. An example of the optimal policy with $\lambda = 4$ is presented in Table 3.3. The three-dimensional state space is transformed into a two-dimension for representation. Some of the decisions are affected by the boundary conditions: production cannot be authorized (*i.*) at station-2 if there is no item in buffer-1 ($x_1 = 0$), (*ii.*) at station-3 if there is no item in buffer-2 ($x_2 = 0$).

We first express the general behavior of the optimal control policy, which also is observed in the example of Table 3.3. At a given demand rate $\lambda = \{1,2..,10\}$,

a) station-1 tends not to produce as x_1, x_2 or x_3 increases.
b) for any x_1 and x_2, it is optimal not to authorize production in the line (production control action is 0 for all states) after a certain level of x_3.
c) after a certain level of x_2, it is optimal to produce only at station-3 until a certain level of x_3, then not to produce at all.

d) after a certain level of x_1, a switching curve type structure is observed in the optimal policy. The switching curves (see Figure 3.3) are shown to be optimal for two-station make-to-stock flow lines with backorders by (Veatch & Wein, 1994). In our setting, there is a threshold that a certain number of items is accumulated in buffer-1. Beginning from the threshold, buffer-1 pretends to provide ample supply for station-2, and switching curves are observed. Consider Table 3.3 when x_1=5 and Figure 3.3: dark green area of Table 3.3 represents the busy set of station-2 that corresponds to the area of B_1 in Figure 3.3; the light green area represents the intersection of busy sets of stations 2 and 3, i.e. where both stations are busy, that corresponds to the area of $B_1 \cap B_2$ in Figure 3.3; the grey area represents the busy set of station-3 that corresponds to the area of B_2 in Figure 3.3.

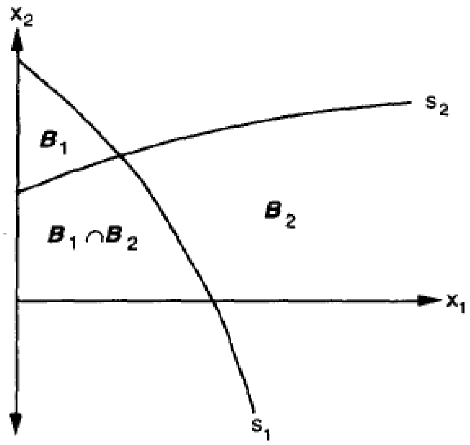

Figure 3.3. Switching curves (Veatch & Wein, 1994) where B_1 (B_2) is the busy set of station-1 (station-2), x_1 and x_2 are the number of items in buffer-1 and buffer-2

As demand rate λ increases, it is observed that

e) the number of states where all three machines are busy increases (control action 7).

f) production continuity lasts longer at each dimension of the state variables.

g) it is optimal to stop producing first at station-1 and then at station-2 as x_1 or x_2 increases until a certain value of λ (λ < 7 in base case scenarios). However, this order of operation is reversed at lower x_1 values (see x_1=1 of Table A1.1 in Appendix 1), then it is reversed again after a certain point of x_1 (see x_1=3 of Table A1.1 in Appendix 1). This behavior is interpreted as a transitional phase and is noticeably observed at higher λ values.

Proposition. There is a threshold T_{i} for $x_{\mathrm{i}} \geq 0, i \in \{1,2\}$ such that it is optimal not to produce at station-i when $x_{\mathrm{i}} > T_{\mathrm{i}}$ for all $x_{\mathrm{j}}, j \in \{1,2,3\} - \{i\}$.

Proof. For $i = 1$, Let Eq0 be the below equation representing the discounted DP formulation of the system:

$$
\begin{aligned}
V(x_1, x_2, x_3) = \frac{1}{v}\langle h_1x_1 + h_2x_2 + h_3x_3 \\
+ \lambda[c \cdot 1 \cdot (x_3 = 0) + V(x_1, x_2, \max\{x_3 - 1,0\})] \\
+\mu_1 min\{V(x_1 + 1, x_2, x_3), V(x_1, x_2, x_3)\} \\
+\mu_2 min\{V(x_1 - 1, x_2 + 1, x_3), V(x_1, x_2, x_3)\} \\
+\mu_3 min\{V(x_1, x_2 - 1, x_3 + 1), V(x_1, x_2, x_3)\}\rangle \qquad \text{(Eq0)}
\end{aligned}
$$

where $v = \lambda + \mu_1 + \mu_2 + \mu_3 + \alpha$ and $x_1, x_2, x_3 \geq 0$. α is defined as the discount rate.

Suppose for $k \geq 0$

$$\mu_1 min\{V^k(x_1+1,x_2,x_3), V^k(x_1,x_2,x_3)\} = \mu_1 V^k(x_1,x_2,x_3)$$, or similarly

$$V^k(x_1,x_2,x_3) \leq V^k(x_1+1,x_2,x_3)$$

We need to show that $V^{k+1}(x_1,x_2,x_3) \leq V^{k+1}(x_1+1,x_2,x_3)$.

Property is true for k=0: $V^0(x_1,x_2,x_3) = 0 \leq V^0(x_1+1,x_2,x_3)$ since $V^0(x_1,x_2,x_3) = 0$ for all $x_1,x_2,x_3 \geq 0$.

Let Eq1 be the following equation:

$$\begin{aligned} V^{k+1}(x_1,x_2,x_3) = {} & h_1x_1 + h_2x_2 + h_3x_3 \\ & +\lambda[c \cdot 1 \cdot (x_3 = 0) + V^k(x_1,x_2,\max\{x_3-1,0\})] \\ & +\mu_1 min\{V^k(x_1+1,x_2,x_3), V^k(x_1,x_2,x_3)\} \\ & +\mu_2 min\{V^k(x_1-1,x_2+1,x_3), V^k(x_1,x_2,x_3)\} \\ & +\mu_3 min\{V^k(x_1,x_2-1,x_3+1), V^k(x_1,x_2,x_3)\} \qquad \text{(Eq1)} \end{aligned}$$

Let Eq2 be the following equation:

$$\begin{aligned} V^{k+1}(x_1+1,x_2,x_3) = {} & h_1(x_1+1) + h_2x_2 + h_3x_3 \\ & +\lambda[c \cdot 1 \cdot (x_3 = 0) + V^k(x_1+1,x_2,\max\{x_3-1,0\})] \\ & +\mu_1 min\{V^k(x_1+2,x_2,x_3), V^k(x_1+1,x_2,x_3)\} \\ & +\mu_2 min\{V^k(x_1,x_2+1,x_3), V^k(x_1+1,x_2,x_3)\} \\ & +\mu_3 min\{V^k(x_1+1,x_2-1,x_3+1), V^k(x_1+1,x_2,x_3)\} \\ & \text{(Eq2)} \end{aligned}$$

So, each term should separately be considered:

[1]. $h_1x_1 + h_2x_2 + h_3x_3 \leq h_1(x_1+1) + h_2x_2 + h_3x_3$ holds due to positive holding cost rates

[2]. If $x_3 > 0$, then $\lambda V^k(x_1,x_2,x_3-1) \leq \lambda V^k(x_1+1,x_2,x_3-1)$ holds due to the supposition

If $x_3 = 0$, then $\lambda[c + V^k(x_1,x_2,0)] \leq \lambda[c + V^k(x_1+1,x_2,0)]$ holds due to the supposition

[3]. $\mu_1 V^k(x_1,x_2,x_3) \leq \mu_1 V^k(x_1+1,x_2,x_3)$ holds for Eq1 and

$\mu_1 V^k(x_1+1,x_2,x_3) \leq \mu_1 V^k(x_1+2,x_2,x_3)$ holds for Eq2 due to the supposition

[4]. For the decisions regarding μ_2 and μ_3, there are 4+4=8 possible combinations of control actions in Eq1 and Eq2.

Consider control actions regarding μ_2:

a. It is optimal for station-2 *to produce* in both Eq1 and Eq2.

   Eq1 returns $V^k(x_1-1, x_2+1, x_3)$ and Eq2 returns $V^k(x_1, x_2+1, x_3)$

   Hence, $V^k(x_1-1, x_2+1, x_3) \leq V^k(x_1, x_2+1, x_3)$ holds due to the supposition.

b. It is optimal for station-2 *not to produce* in both Eq1 and Eq2.

   Eq1 returns $V^k(x_1, x_2, x_3)$ and Eq2 returns $V^k(x_1+1, x_2, x_3)$

   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1+1, x_2, x_3)$ holds due to the supposition.

c. It is optimal for station-2 *to produce* in Eq1 but *not to produce* in Eq2.

   Eq1 states that $V^k(x_1-1, x_2+1, x_3) \leq V^k(x_1, x_2, x_3)$

   Eq2 states that $V^k(x_1+1, x_2, x_3) \leq V^k(x_1, x_2+1, x_3)$

   Hence, $V^k(x_1-1, x_2+1, x_3) \leq V^k(x_1+1, x_2, x_3)$ holds due to

$$V^k(x_1-1, x_2+1, x_3) \leq V^k(x_1, x_2, x_3) \leq V^k(x_1-1, x_2+1, x_3) \leq V^k(x_1+1, x_2, x_3)$$

d. It is optimal for station-2 *not to produce* in Eq1 but *to produce* in Eq2.

   Eq1 states that $V^k(x_1, x_2, x_3) \leq V^k(x_1-1, x_2+1, x_3)$

   Eq2 states that $V^k(x_1, x_2+1, x_3) \leq V^k(x_1+1, x_2, x_3)$

   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1, x_2+1, x_3)$ holds similarly to the part c.

Consider control actions regarding μ_3:

e. It is optimal for station-3 *to produce* in both Eq1 and Eq2.

   Eq1 returns $V^k(x_1, x_2-1, x_3+1)$ and Eq2 returns $V^k(x_1+1, x_2-1, x_3+1)$

   Hence, $V^k(x_1, x_2-1, x_3+1) \leq V^k(x_1+1, x_2-1, x_3+1)$ holds due to the supposition.

f. It is optimal for station-3 *not to produce* in both Eq1 and Eq2.

   Eq1 returns $V^k(x_1, x_2, x_3)$ and Eq2 returns $V^k(x_1+1, x_2, x_3)$

   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1+1, x_2, x_3)$ holds.

g. It is optimal for station-3 *to produce* in Eq1 but *not to produce* in Eq2.

   Eq1 states that $V^k(x_1, x_2-1, x_3+1) \leq V^k(x_1, x_2, x_3)$

   Eq2 states that $V^k(x_1+1, x_2, x_3) \leq V^k(x_1+1, x_2-1, x_3+1)$

   Hence, $V^k(x_1, x_2-1, x_3+1) \leq V^k(x_1, x_2-1, x_3+1)$ holds similarly to the part c.

h. It is optimal for station-3 *not to produce* in Eq1 but *to produce* in Eq2.

Eq1 states that $V^k(x_1, x_2, x_3) \leq V^k(x_1, x_2 - 1, x_3 + 1)$

Eq2 states that $V^k(x_1 + 1, x_2 - 1, x_3 + 1) \leq V^k(x_1 + 1, x_2, x_3)$

Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1 + 1, x_2 - 1, x_3 + 1)$ holds similarly to the part c.

We can conclude that the property holds for k+1. As $k \to \infty$, value function V converges with a given epsilon error, and the optimal value is found for the problem. Also, the average cost is obtained while setting α to 0 and dividing the value function by time steps. For $i = 2$, the proof is provided in Appendix 2.■

Figure 3.4 represents the optimal average cost and computation time (in minutes) of the base case scenario as λ increases. Since no upper boundaries are assigned to the state variables (x_1, x_2, x_3), a truncated state space of the variables is considered in computational studies. In Figure 3.4, the traffic intensity of the system increases with λ, which causes larger state spaces for convergence of the optimal control problem and sharp rises in the long-run average system cost. The number of states required for the basic model in the figure equals 125 when λ = 1. However, it reaches 2000 at λ =5 and exceeds 28,000 when λ = 10.

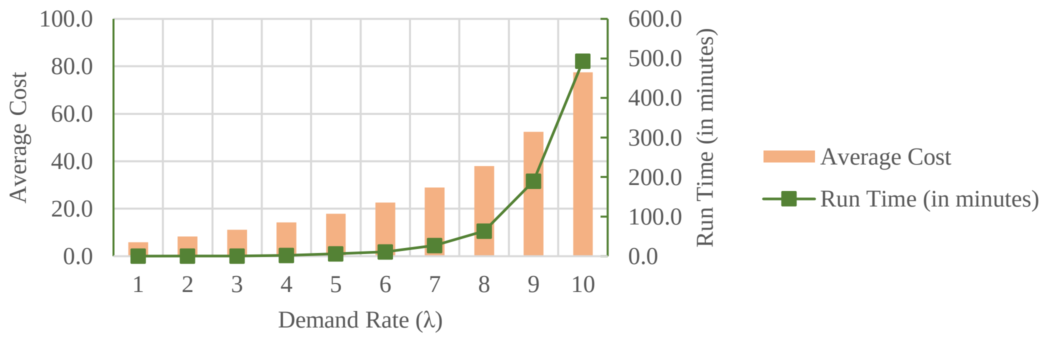

Figure 3.4. Average cost and run time versus demand rate

We carry out two experiments to analyze the effect of system parameters on the optimal behavior. The first experiment is to examine the impact of lost sales cost $c \in \{25, 50, 75\}$ on the optimal average cost, as shown in Figure 3.5. At any value of λ, the average cost is increasing in c. Although average cost responds slightly to changes in c at relatively lower λ values, it varies with c as λ increases.

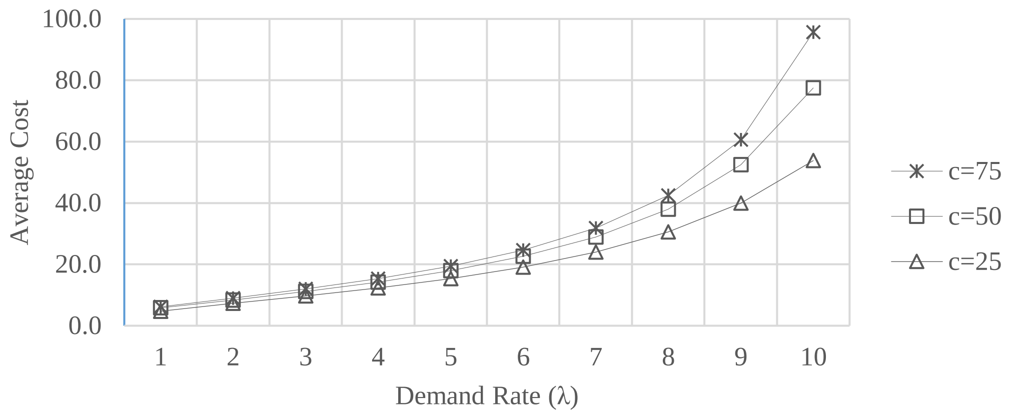

Figure 3.5. Average cost versus *c*

The second experiment is designed to study the effect of production rates on the optimal control problem. The base case scenario is defined as case 0, while six more cases are created, as shown in Table 3.4. Average system costs of the optimal control policies are presented in Figure 3.6 for all cases of Table 3.4, while λ varies from 1 to 10. For every value of λ in Figure 3.6, it is observed that case 6 produces the minimum system cost where a faster machine is located downstream ($\mu_3 = 15$). The second and the third minimum are cases 5 and 4, with the fastest machine is located at the intermediate station and upstream, respectively.

Table 3.4. Production rates of the cases

	Case 0	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
μ_1	10	5	10	10	15	10	10
μ_2	10	10	5	10	10	15	10
μ_3	10	10	10	5	10	10	15

On the other hand, case 3 produces the maximum system cost when a slower machine is assigned to the downstream station ($\mu_3 = 5$). The second and third maximum belong to cases 2 and 1, respectively. If the production rate of any machine in the system is increased, then the average cost is decreased. However, the location of the non-identical station affects the system cost, and case 0 is placed between partitions of cases 1,2,3 and 4,5,6. As it is shown in Figure 3.6, it is observed that the cost deterioration due to a slower machine is distinctively higher than the cost improvement due to a faster machine, when λ ≥ 5.

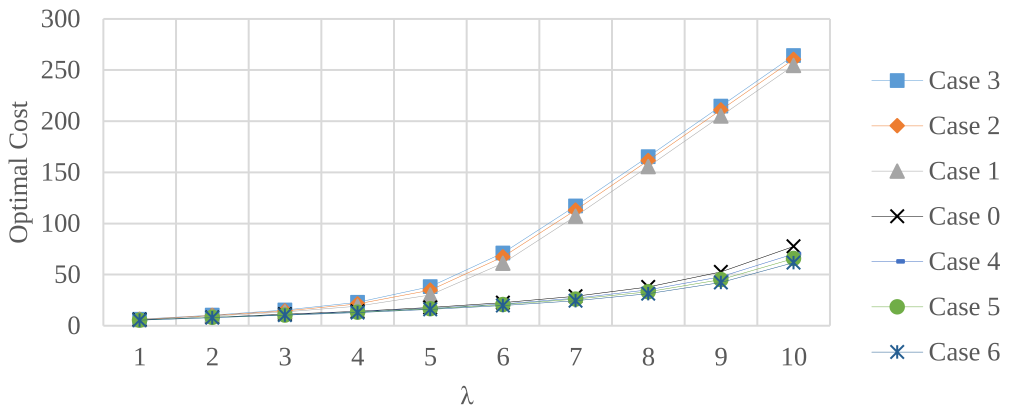

Figure 3.6. Average cost versus production rates

The optimal production policies of the cases are examined, and it is observed that the abovementioned observations *a* to *d* are still valid for all cases of the basic model at a given λ.

3.2. Extended Models with a Cox-2 Distributed Machine

This chapter considers extended models of three station make-to-stock systems to investigate the effect of failure-prone machines, rework/remanufacturing operations on a production line using two-phase Coxian (Cox-2) processing times. We examine three different models: while keeping the processing time distributions of remaining stations as Exponential, we assign Cox-2 processing times to upstream, intermediate and downstream stations. These models allow us to investigate the effect of system parameters and the location of the two-phase Coxian distributed machine on the optimal control problems.

A Cox-2 random variable has two independent Exponential phases, and there is a certain visiting probability from phase one to phase two (see Figure 3.7). A busy machine with Cox-2 processing times should be either at the first phase or the second phase at any given time in a production environment. Two-phase Coxian processing times allow us to

- consider machine breakdowns-repairs, rework or inspection operations: if processing times, times to failure and repair times of a machine are exponentially distributed, then the distribution of the total time spent processing at the machine is Cox-2 (Altiok & Stidham, 1983). For instance, frequent breakdowns with quick repairs or rare breakdowns with long repairs can be studied.

- model different system characteristics like incorporating customer requests in products: exclusive features of products that require additional stage before their production completion can be modeled, such as accessories of cars.
- approximate general service time distributions (Altiok, 1985; van der Heijden, 1988).

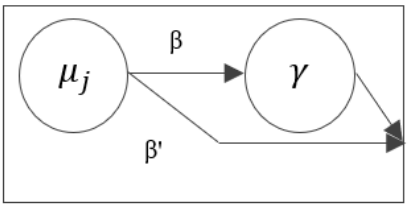

Figure 3.7. A representation of station-j with Cox-2 processing times

The system parameters are kept as they are defined for the basic model in Chapter 3.1. Since phases of a Cox-2 random variable are exponentially distributed, μ_j represents the production rates of the first Coxian phase and the machine with Exponential processing times, depending on the processing time distribution of station j. The additional parameters belong to Coxian processing times, where γ is defined as the production rate of the second phase of the Cox-2 distributed machine, and β is the visiting probability from phase one to phase two. The notation of the extended models is shown in Table 3.5.

Table 3.5. Notation of the extended models

j	Station index, j=1,2,3
λ	FG demand rate
c	Lost sales cost
h_j	Holding cost rate of buffer-j (and the item being produced its downstream station)
μ_j	Production rate of (i) the machine if its distribution is Exponential, (ii) phase-1 of the machine if its distribution is Cox-2, at station-j
γ	Production rate of the second phase of the Cox-2 distributed machine
β	Second phase visiting probability of the Cox-2 distributed machine

The extended models are presented in Figures 3.8, 3.9, 3.10, where the Cox-2 distributed machine is assigned to upstream, intermediate, and downstream locations. Cox-2 processing times do not have the memoryless property. However, they have

Exponentially distributed phases that are memoryless. In order to keep track of the phase information, additional state variables are defined for each model: let y_1 and y_2 be binary state variables defined to keep track of the production status of a Cox-2 distributed machine such that $y_i = 1$ if i^{th} phase of the machine is busy, otherwise $y_i = 0$, $i = 1,2$.

In the extended models, the system state is defined as a vector of five variables. Let $(y_1, y_2, x_1, x_2, x_3)$ be the state vector of model 1 (see Figure 3.8), where y_1 and y_2 are for the status of station 1, x_1 (x_2) is the number of items in buffer-1 (buffer-2) plus the item being processed its downstream station - due to Exponential processing times of stations 2 and 3, and x_3 is the number of items in the finished goods buffer. In model 2 (see Figure 3.9), $(x_1, y_1, y_2, x_2, x_3)$ is the state vector such that x_1 is the number of items in buffer-1, y_1 and y_2 are for the status of station 2, x_2 is the number of items in buffer-2 plus the item being processed its downstream station - due to the Exponential processing time assumption of station 2, x_3 is the number of items in the finished goods buffer. Model 3 has a state vector of $(x_1, x_2, y_1, y_2, x_3)$ as shown in Figure 3.10. The variable x_1 keeps the number of items in buffer-1 plus station-1, x_2 keeps the number of items in buffer-2, y_1 and y_2 are for the status of station-3 that has Cox-2 distributed processing times, and x_3 is defined for the finished goods buffer. Control variables $a_j \in \{0,1\}, j = 1,2,3$ are defined for each production stage such that $a_j = 1$ if station-j is busy, it is idle otherwise.

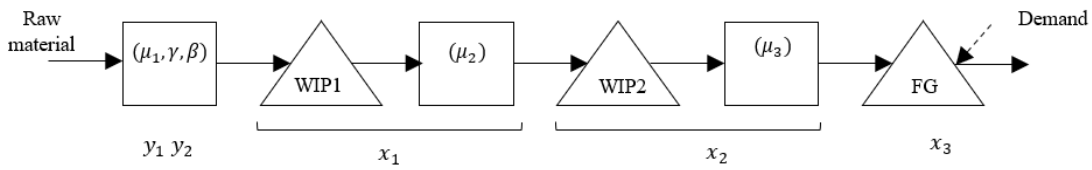

Figure 3.8. Extended model 1

The extended model 1 consists of a Cox-2 distributed machine at the upstream station (Figure 3.8), and the domain of the control variable a_1 depends on the state variables y_1, y_2 such that $a_1 \in \{0,1\}$ if $y_1 + y_2 = 0$, and $a_1 = y_1$ if $y_1 + y_2 = 1$. That is if both phases of Cox-2 are idle $(y_1 + y_2 = 0)$, then control actions 0 and 1 are feasible for the control variable a_1. If the first phase is busy $(y_1 = 1)$, then action $a_1 = 1$ has to be taken because we assume that ongoing production cannot be cancelled. If the second phase is busy $(y_2 = 1)$, then action $a_1 = 0$ is the only feasible solution because production cannot be authorized at the first phase while the second phase is busy. The DP formulation of the extended model-1 is given as follows:

$$V(y_1,y_2,x_1,x_2,x_3)=\frac{1}{\nu+\alpha}\min_{a_1,a_2,a_3}\{h_1x_1+h_2x_2+h_3x_3+\lambda L(y_1,y_2,x_1,x_2,x_3)$$
$$+a_1\mu_1T_1(y_1,y_2,x_1,x_2,x_3)+y_2\gamma T_2(y_1,y_2,x_1,x_2,x_3)$$
$$+a_2\mu_2T_3(y_1,y_2,x_1,x_2,x_3)+a_3\mu_3T_4(y_1,y_2,x_1,x_2,x_3)$$
$$+\left((1-y_2)\gamma+\textstyle\sum_{j=1}^{3}(1-a_j)\,\mu_j\right)V(y_1,y_2,x_1,x_2,x_3)\Big\} \tag{4}$$

where α is a discount rate, $\nu=\lambda+\gamma+\sum_{j=1}^{3}\mu_j$, and T_1,T_2,T_3,T_4 are production-related operators such that

$$T_1(y_1,y_2,x_1,x_2,x_3)=\begin{cases}\begin{matrix}\beta V(a_1-1,y_2+1,x_1,x_2,x_3)+\\(1-\beta)V(a_1-1,y_2,x_1+1,x_2,x_3)\end{matrix}, & a_1>0\\ \mathrm{V}(a_1,y_2,x_1,x_2,x_3), & a_1=0\end{cases}$$

$$T_2(y_1,y_2,x_1,x_2,x_3)=\begin{cases}V(a_1,y_2-1,x_1+1,x_2,x_3), & y_2>0\\ \mathrm{V}(a_1,y_2,x_1,x_2,x_3), & y_2=0\end{cases}$$

$$T_3(y_1,y_2,x_1,x_2,x_3)=\begin{cases}V(a_1,y_2,x_1-1,x_2+1,x_3), & x_1>0\\ \mathrm{V}(a_1,y_2,x_1,x_2,x_3), & x_1=0\end{cases}$$

$$T_4(y_1,y_2,x_1,x_2,x_3)=\begin{cases}V(a_1,y_2,x_1,x_2-1,x_3+1), & x_2>0\\ \mathrm{V}(a_1,y_2,x_1,x_2,x_3), & x_2=0\end{cases}$$

L is a lost sales operator such that

$$L(y_1,y_2,x_1,x_2,x_3)=\begin{cases}V(y_1,y_2,x_1,x_2,x_3-1), & x_3>0\\ c+V(y_1,y_2,x_1,x_2,0), & x_3=0\end{cases} \tag{5}$$

Whenever a control action a_1 is taken, the system state (y_1,y_2,x_1,x_2,x_3) makes an instantaneous transition to state (a_1,y_2,x_1,x_2,x_3). Due to the nature of Coxian processing times, there are three possible transitions regarding station-1 in the DP formulation, as shown in (4): (*i*) the term $a_1\mu_1\beta V(a_1-1,y_2+1,x_1,x_2,x_3)$ corresponds to a transition from phase-1 to phase-2 of the machine, (*ii*) the term $a_1\mu_1(1-\beta)V(a_1-1,y_2,x_1+1,x_2,x_3)$ is for a transition from phase-1 of the machine to buffer-1, (*iii*) the term $y_2\gamma V(a_1,y_2-1,x_1+1,x_2,x_3)$ represents a transition from phase-2 of the machine to buffer-1. The remaining terms are for production completion at stations 2 and 3, and the term $\left((1-y_2)\gamma+\sum_{j=1}^{3}(1-a_j)\,\mu_j\right)V(y_1,y_2,x_1,x_2,x_3)$ is self-transition due to uniformization.

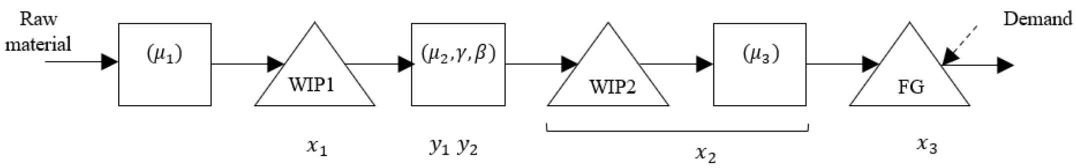

Figure 3.9. Extended model 2

In the extended model 2, a Cox-2 distributed machine is assigned to intermediate while remaining machines operate with Exponential processing times (Figure 3.9). Control variable a_2 depends on the state variables y_1, y_2 such that $a_2 \in \{0,1\}$ if $y_1 + y_2 = 0$, and $a_2 = y_1$ if $y_1 + y_2 = 1$, while other control variables are defined as $a_1, a_3 \in \{0,1\}$. The DP formulation of model 2 is given as

$$\begin{aligned}
V(x_1, y_1, y_2, x_2, x_3) = \frac{1}{v+\alpha} \min_{a_1,a_2,a_3} \{ & h_1(x_1 + y_1 + y_2) + h_2 x_2 + h_3 x_3 \\
& +\lambda L(x_1, y_1, y_2, x_2, x_3) \\
& +a_1 \mu_1 V(x_1 + 1, a_2, y_2, x_2, x_3) + a_2 \mu_2 T_1(x_1, y_1, y_2, x_2, x_3) \\
& +y_2 \gamma T_2(x_1, y_1, y_2, x_2, x_3) + a_3 \mu_3 T_3(x_1, y_1, y_2, x_2, x_3) \\
& + \left((1 - y_2)\gamma + \textstyle\sum_{j=1}^{3} (1 - a_j)\, \mu_j \right) V(x_1, y_1, y_2, x_2, x_3) \Big\}
\end{aligned} \tag{6}$$

where α is a discount rate, $v = \lambda + \gamma + \sum_{j=1}^{3} \mu_j$ and T_1, T_2, T_3 are production-related operators such that

$$T_1(x_1, y_1, y_2, x_2, x_3) = \begin{cases} \begin{array}{c} \beta V(x_1 - 1, a_2 - 1, y_2 + 1, x_2, x_3) + \\ (1-\beta) V(x_1 - 1, a_2 - 1, y_2, x_2 + 1, x_3) \end{array}, & x_1, a_2 > 0 \\ \mathrm{V}(x_1, a_2, y_2, x_2, x_3), & o.w. \end{cases}$$

$$T_2(x_1, y_1, y_2, x_2, x_3) = \begin{cases} V(x_1, a_2, y_2 - 1, x_2 + 1, x_3), & y_2 > 0 \\ \mathrm{V}(x_1, a_2, y_2, x_2, x_3), & y_2 = 0 \end{cases}$$

$$T_3(x_1, y_1, y_2, x_2, x_3) = \begin{cases} (x_1, a_2, y_2, x_2 - 1, x_3 + 1), & x_2 > 0 \\ \mathrm{V}(x_1, a_2, y_2, x_2, x_3), & x_2 = 0 \end{cases}$$

L is a lost sales operator such that

$$L(x_1, y_1, y_2, x_2, x_3) = \begin{cases} V(x_1, y_1, y_2, x_2, x_3 - 1), & x_3 > 0 \\ c + V(x_1, y_1, y_2, x_2, 0), & x_3 = 0 \end{cases} \tag{7}$$

The system state $(x_1, y_1, y_2, x_2, x_3)$ makes an instantaneous transition to state $(x_1, a_2, y_2, x_2, x_3)$ when a control action a_2 is taken. The operators T_1 and T_2 represent the transitions from the first and second phases of the Coxian distributed machine. The operator T_3 is defined for the production completion event at station-3.

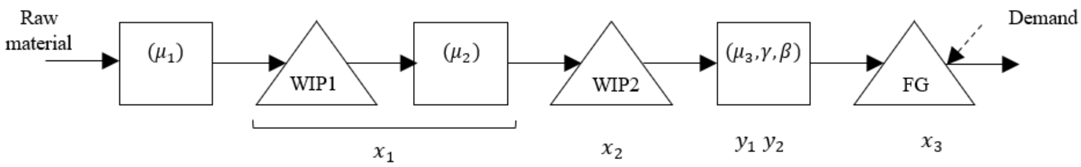

Figure 3.10. Extended model 3

Model 3 consists of Cox-2 processing times at the downstream station (Figure 3.10). Exponentially distributed stations 1 and 2 are controlled with variables $a_1, a_2 \in \{0,1\}$. The domain of control variable a_3 is defined as $a_3 \in \{0,1\}$ if $y_1 + y_2 = 0$, and $a_3 = y_1$ if $y_1 + y_2 = 1$, which causes an instant transition from state $(x_1, x_2, y_1, y_2, x_3)$ to $(x_1, x_2, a_3, y_2, x_3)$. The DP formulation of model 3 is developed as follows:

$$
\begin{aligned}
V(x_1, x_2, y_1, y_2, x_3) = \frac{1}{\nu + \alpha} \min_{a_1, a_2, a_3} \{ & h_1 x_1 + h_2 (x_2 + y_1 + y_2) + h_3 x_3 \\
& + \lambda L(x_1, x_2, y_1, y_2, x_3) \\
& + a_1 \mu_1 V(x_1 + 1, x_2, a_3, y_2, x_3) + a_2 \mu_2 V T_1(x_1, x_2, y_1, y_2, x_3) \\
& + a_3 \mu_3 T_2(x_1, x_2, y_1, y_2, x_3) + y_2 \gamma T_3(x_1, x_2, y_1, y_2, x_3) \\
& + \left((1 - y_2)\gamma + \textstyle\sum_{j=1}^{3} (1 - a_j)\, \mu_j \right) V(x_1, x_2, y_1, y_2, x_3) \Big\}
\end{aligned}
\tag{8}
$$

where α is a discount rate, $\nu = \lambda + \gamma + \sum_{j=1}^{3} \mu_j$ and and T_1, T_2, T_3 are production-related operators such that

$$
T_1(x_1, x_2, y_1, y_2, x_3) = \begin{cases} V(x_1 - 1, x_2 + 1, a_3, y_2, x_3), & x_1 > 0 \\ V(x_1, x_2, a_3, y_2, x_3), & x_1 = 0 \end{cases}
$$

$$
T_2(x_1, x_2, y_1, y_2, x_3) = \begin{cases} \beta V(x_1, x_2 - 1, a_3 - 1, y_2 + 1, x_3) + \\ (1 - \beta) V(x_1, x_2 - 1, a_3 - 1, y_2, x_3 + 1)', & x_2, a_3 > 0 \\ V(x_1, x_2, a_3, y_2, x_3), & o.w. \end{cases}
$$

$$
T_3(x_1, x_2, y_1, y_2, x_3) = \begin{cases} V(x_1, x_2, a_3, y_2 - 1, x_3 + 1), & y_2 > 0 \\ V(x_1, x_2, a_3, y_2, x_3), & y_2 = 0 \end{cases}
$$

L is a lost sales operator such that

$$
L(x_1, x_2, y_1, y_2, x_3) = \begin{cases} V(x_1, x_2, y_1, y_2, x_3 - 1), & x_3 > 0 \\ c + V(x_1, x_2, y_1, y_2, 0), & x_3 = 0 \end{cases}
\tag{9}
$$

In the extended model 3, the operator T_1 represents the production completion event at station-2. The operators T_2 and T_3 are defined for the transitions from the first and second phases of the Coxian distributed machine.

The dynamic programming formulations developed for the extended models are based on a discounted cost criterion with a discount rate of α as in the basic model. The average system cost is obtained via the value iteration algorithm given in Figure 3.2 of Chapter 3.1.

In the extended models, the optimal policy decides whether *to produce* or *not to produce* at each machine, similar to the basic model developed in Chapter 3.1. Thus, there are eight different production control actions. Let $A = \{0,1,..,7\}$ be the action space for the production control actions. The details of the control actions are presented in Table 3.2 of Chapter 3.1. For numerical experiments, we define a set of parameters as given in Table 3.6 below. Lost sales cost c is set to 50, holding cost rates are defined as $[h_1, h_2, h_3]$ = [1, 1.5, 2], and demand rate is $\lambda \in \{3,4,5,8\}$. Production rates of Exponentially distributed stations and the first phase of Cox-2 distributed station are set to 10. The production rate of the second phase of Cox-2 is set as $\gamma \in \{5,10,20\}$. Visiting probability β has a range of {0, 0.1, …, 0.9, 1}. For simplicity, β = 0 corresponds to an Exponential random variable where its parameter is the phase-1 parameter of Cox-2, β = 1 corresponds to a generalized Erlang random variable. For each extended model, 132 instances are created, and the optimal control problem is solved. In numerical studies, the termination criterion for the value iteration algorithm is set to $\varepsilon = 10^{-2}$.

Table 3.6. Set of parameters defined for the numerical studies

$[h_1, h_2, h_3]$	[1, 1.5, 2]
λ	{3, 4, 5, 8}
c	50
$[\mu_1, \mu_2, \mu_3]$	[10, 10, 10]
γ	{5, 10, 20}
β	{0.0, 0.1, …, 0.9, 1.0}

The optimal policies of the extended models depend on the Cox-2 phases. The results are represented for the states when (*i.*) both phases are idle, $(y_1, y_2) = (0,0)$; (ii.) the first phase is busy, $(y_1, y_2) = (1,0)$; (*iii.*) the second phase is busy, $(y_1, y_2) = (0,1)$. Table 3.7 presents an example of the optimal policy for model-2 with parameters of $\lambda = 4, \gamma = 5, \beta = 0.6$.

Table 3.7. An example of the optimal policy for model 2, $\lambda = 4, \gamma = 5, \beta = 0.6$

		$(y_1, y_2) = (0,0)$																	$(y_1, y_2) = (1,0)$														$(y_1, y_2) = (0,1)$												
	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12
$x_1 = 1$	0	4	4	4	4	4	4	4	4	4	4	4	2	2	2	0	0	0	4	4	4	4	4	4	4	4	4	4	4	2	2	0	1	1	1	1	1	1	1	1	1	1	1	0	0
	1	7	7	7	7	4	4	4	4	4	4	2	2	2	0	0	0	1	7	7	7	7	7	4	4	4	4	4	2	2	2	1	6	6	6	6	6	1	1	1	1	1	0	0	0
	2	7	7	7	7	7	4	4	4	4	2	2	2	0	0	0	0	2	7	7	7	7	7	4	4	4	4	2	2	2	2	2	6	6	6	6	6	1	1	1	1	0	0	0	0
	3	7	7	7	7	7	4	4	4	2	2	2	0	0	0	0	0	3	7	7	7	7	7	4	4	4	2	2	2	2	2	3	6	6	6	6	6	1	1	1	0	0	0	0	0
	4	7	7	7	7	7	4	4	2	2	2	0	0	0	0	0	0	4	7	7	7	7	7	7	4	2	2	2	2	2	2	4	6	6	6	6	6	6	1	0	0	0	0	0	0
	5	7	7	7	7	7	4	2	2	2	0	0	0	0	0	0	0	5	7	7	7	7	7	7	2	2	2	2	2	2	2	5	6	6	6	6	6	6	0	0	0	0	0	0	0
	6	7	7	7	7	7	2	2	2	0	0	0	0	0	0	0	0	6	7	7	7	7	7	5	2	2	2	2	2	2	2	6	6	6	6	6	6	3	0	0	0	0	0	0	0
	7	7	7	7	7	5	2	2	0	0	0	0	0	0	0	0	0	7	7	7	7	7	5	5	2	2	2	2	2	2	2	7	6	6	6	6	3	3	0	0	0	0	0	0	0
	8	7	7	7	5	5	3	0	0	0	0	0	0	0	0	0	0	8	7	7	7	5	5	5	2	2	2	2	2	2	2	8	6	6	6	3	3	3	0	0	0	0	0	0	0
	9	7	5	5	5	3	3	0	0	0	0	0	0	0	0	0	0	9	7	7	5	5	5	5	2	2	2	2	2	2	2	9	6	6	3	3	3	3	0	0	0	0	0	0	0
	10	5	5	5	3	3	3	0	0	0	0	0	0	0	0	0	0	10	5	5	5	5	5	5	2	2	2	2	2	2	2	10	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	5	5	3	3	3	3	0	0	0	0	0	0	0	0	0	0	11	5	5	5	5	5	5	2	2	2	2	2	2	2	11	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	5	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	12	5	5	5	5	5	5	2	2	2	2	2	2	2	12	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	13	5	5	5	5	5	5	2	2	2	2	2	2	2	13	3	3	3	3	3	3	0	0	0	0	0	0	0
	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12
$x_1 = 2$	0	4	4	4	4	4	4	4	4	2	2	2	2	2	2	0	0	0	4	4	4	4	4	4	4	4	4	2	2	2	2	0	1	1	1	1	1	1	1	1	1	0	0	0	0
	1	7	7	7	7	4	4	4	2	2	2	2	2	2	0	0	0	1	7	7	7	7	7	4	4	4	2	2	2	2	2	1	6	6	6	6	6	1	1	1	0	0	0	0	0
	2	7	7	7	7	7	4	2	2	2	2	2	2	0	0	0	0	2	7	7	7	7	7	4	4	2	2	2	2	2	2	2	6	6	6	6	6	1	1	0	0	0	0	0	0
	3	7	7	7	7	7	2	2	2	2	2	2	0	0	0	0	0	3	7	7	7	7	7	4	2	2	2	2	2	2	2	3	6	6	6	6	6	6	0	0	0	0	0	0	0
	4	7	7	7	7	5	2	2	2	2	2	0	0	0	0	0	0	4	7	7	7	7	7	5	2	2	2	2	2	2	2	4	6	6	6	6	6	3	0	0	0	0	0	0	0
	5	7	7	7	5	5	2	2	2	2	0	0	0	0	0	0	0	5	7	7	7	7	5	5	2	2	2	2	2	2	2	5	6	6	6	6	3	3	0	0	0	0	0	0	0
	6	7	7	5	5	5	2	2	2	0	0	0	0	0	0	0	0	6	7	7	7	5	5	5	2	2	2	2	2	2	2	6	6	6	6	3	3	3	0	0	0	0	0	0	0
	7	7	5	5	5	5	2	2	0	0	0	0	0	0	0	0	0	7	7	5	5	5	5	5	2	2	2	2	2	2	2	7	6	6	3	3	3	3	0	0	0	0	0	0	0
	8	5	5	5	5	5	2	0	0	0	0	0	0	0	0	0	0	8	5	5	5	5	5	5	2	2	2	2	2	2	2	8	3	3	3	3	3	3	0	0	0	0	0	0	0
	9	5	5	5	5	5	3	0	0	0	0	0	0	0	0	0	0	9	5	5	5	5	5	5	2	2	2	2	2	2	2	9	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	5	5	5	5	3	3	0	0	0	0	0	0	0	0	0	0	10	5	5	5	5	5	5	2	2	2	2	2	2	2	10	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	5	5	5	3	3	3	0	0	0	0	0	0	0	0	0	0	11	5	5	5	5	5	5	2	2	2	2	2	2	2	11	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	5	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	12	5	5	5	5	5	5	2	2	2	2	2	2	2	12	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	13	5	5	5	5	5	5	2	2	2	2	2	2	2	13	3	3	3	3	3	3	0	0	0	0	0	0	0
	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12
$x_1 = 3$	0	4	4	4	4	4	2	2	2	2	2	2	2	2	2	0	0	0	4	4	4	4	4	4	2	2	2	2	2	2	2	0	1	1	1	1	1	1	0	0	0	0	0	0	0
	1	7	7	7	7	2	2	2	2	2	2	2	2	2	0	0	0	1	7	7	7	7	7	2	2	2	2	2	2	2	2	1	6	6	6	6	6	0	0	0	0	0	0	0	0
	2	7	7	7	5	5	2	2	2	2	2	2	2	0	0	0	0	2	7	7	7	7	5	2	2	2	2	2	2	2	2	2	6	6	6	6	3	0	0	0	0	0	0	0	0
	3	7	7	5	5	5	2	2	2	2	2	2	0	0	0	0	0	3	7	7	7	5	5	5	2	2	2	2	2	2	2	3	6	6	6	3	3	3	0	0	0	0	0	0	0
	4	5	5	5	5	5	2	2	2	2	2	0	0	0	0	0	0	4	7	7	5	5	5	5	2	2	2	2	2	2	2	4	6	6	3	3	3	3	0	0	0	0	0	0	0
	5	5	5	5	5	5	2	2	2	2	0	0	0	0	0	0	0	5	5	5	5	5	5	5	2	2	2	2	2	2	2	5	3	3	3	3	3	3	0	0	0	0	0	0	0
	6	5	5	5	5	5	2	2	2	0	0	0	0	0	0	0	0	6	5	5	5	5	5	5	2	2	2	2	2	2	2	6	3	3	3	3	3	3	0	0	0	0	0	0	0
	7	5	5	5	5	5	2	2	0	0	0	0	0	0	0	0	0	7	5	5	5	5	5	5	2	2	2	2	2	2	2	7	3	3	3	3	3	3	0	0	0	0	0	0	0
	8	5	5	5	5	5	2	2	0	0	0	0	0	0	0	0	0	8	5	5	5	5	5	5	2	2	2	2	2	2	2	8	3	3	3	3	3	3	0	0	0	0	0	0	0
	9	5	5	5	5	5	3	0	0	0	0	0	0	0	0	0	0	9	5	5	5	5	5	5	2	2	2	2	2	2	2	9	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	5	5	5	5	3	3	0	0	0	0	0	0	0	0	0	0	10	5	5	5	5	5	5	2	2	2	2	2	2	2	10	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	5	5	5	3	3	3	0	0	0	0	0	0	0	0	0	0	11	5	5	5	5	5	5	2	2	2	2	2	2	2	11	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	5	5	3	3	3	3	0	0	0	0	0	0	0	0	0	0	12	5	5	5	5	5	5	2	2	2	2	2	2	2	12	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	13	5	5	5	5	5	5	2	2	2	2	2	2	2	13	3	3	3	3	3	3	0	0	0	0	0	0	0
	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12
$x_1 = 4$	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	4	2	2	2	2	2	2	2	2	2	2	2	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	1	5	5	5	5	2	2	2	2	2	2	2	2	2	2	0	0	1	5	5	5	5	5	2	2	2	2	2	2	2	2	1	6	3	3	3	3	0	0	0	0	0	0	0	0
	2	5	5	5	5	5	2	2	2	2	2	2	2	2	0	0	0	2	5	5	5	5	5	2	2	2	2	2	2	2	2	2	3	3	3	3	3	0	0	0	0	0	0	0	0
	3	5	5	5	5	5	2	2	2	2	2	2	0	0	0	0	0	3	5	5	5	5	5	5	2	2	2	2	2	2	2	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	4	5	5	5	5	5	2	2	2	2	2	0	0	0	0	0	0	4	5	5	5	5	5	5	2	2	2	2	2	2	2	4	3	3	3	3	3	3	0	0	0	0	0	0	0
	5	5	5	5	5	5	2	2	2	2	0	0	0	0	0	0	0	5	5	5	5	5	5	5	2	2	2	2	2	2	2	5	3	3	3	3	3	3	0	0	0	0	0	0	0
	6	5	5	5	5	5	2	2	2	0	0	0	0	0	0	0	0	6	5	5	5	5	5	5	2	2	2	2	2	2	2	6	3	3	3	3	3	3	0	0	0	0	0	0	0
	7	5	5	5	5	5	2	2	0	0	0	0	0	0	0	0	0	7	5	5	5	5	5	5	2	2	2	2	2	2	2	7	3	3	3	3	3	3	0	0	0	0	0	0	0
	8	5	5	5	5	5	2	2	0	0	0	0	0	0	0	0	0	8	5	5	5	5	5	5	2	2	2	2	2	2	2	8	3	3	3	3	3	3	0	0	0	0	0	0	0
	9	5	5	5	5	5	3	0	0	0	0	0	0	0	0	0	0	9	5	5	5	5	5	5	2	2	2	2	2	2	2	9	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	5	5	5	5	3	3	0	0	0	0	0	0	0	0	0	0	10	5	5	5	5	5	5	2	2	2	2	2	2	2	10	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	5	5	5	3	3	3	0	0	0	0	0	0	0	0	0	0	11	5	5	5	5	5	5	2	2	2	2	2	2	2	11	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	5	5	3	3	3	3	0	0	0	0	0	0	0	0	0	0	12	5	5	5	5	5	5	2	2	2	2	2	2	2	12	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	13	5	5	5	5	5	5	2	2	2	2	2	2	2	13	3	3	3	3	3	3	0	0	0	0	0	0	0
	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12	$x_2 \backslash x_3$	0	1	2	3	4	5	6	7	8	9	10	11	12
$x_1 = 5$	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	5	5	5	5	2	2	2	2	2	2	2	2	2	2	0	0	1	5	5	5	5	5	2	2	2	2	2	2	2	2	1	3	3	3	3	3	0	0	0	0	0	0	0	0
	2	5	5	5	5	5	2	2	2	2	2	2	2	2	0	0	0	2	5	5	5	5	5	2	2	2	2	2	2	2	2	2	3	3	3	3	3	0	0	0	0	0	0	0	0
	3	5	5	5	5	5	2	2	2	2	2	2	2	0	0	0	0	3	5	5	5	5	5	5	2	2	2	2	2	2	2	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	4	5	5	5	5	5	2	2	2	2	2	2	0	0	0	0	0	4	5	5	5	5	5	5	2	2	2	2	2	2	2	4	3	3	3	3	3	3	0	0	0	0	0	0	0
	5	5	5	5	5	5	2	2	2	2	2	0	0	0	0	0	0	5	5	5	5	5	5	5	2	2	2	2	2	2	2	5	3	3	3	3	3	3	0	0	0	0	0	0	0
	6	5	5	5	5	5	2	2	2	2	0	0	0	0	0	0	0	6	5	5	5	5	5	5	2	2	2	2	2	2	2	6	3	3	3	3	3	3	0	0	0	0	0	0	0
	7	5	5	5	5	5	2	2	2	0	0	0	0	0	0	0	0	7	5	5	5	5	5	5	2	2	2	2	2	2	2	7	3	3	3	3	3	3	0	0	0	0	0	0	0
	8	5	5	5	5	5	2	2	0	0	0	0	0	0	0	0	0	8	5	5	5	5	5	5	2	2	2	2	2	2	2	8	3	3	3	3	3	3	0	0	0	0	0	0	0
	9	5	5	5	5	5	3	0	0	0	0	0	0	0	0	0	0	9	5	5	5	5	5	5	2	2	2	2	2	2	2	9	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	5	5	5	5	3	3	0	0	0	0	0	0	0	0	0	0	10	5	5	5	5	5	5	2	2	2	2	2	2	2	10	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	5	5	5	3	3	3	0	0	0	0	0	0	0	0	0	0	11	5	5	5	5	5	5	2	2	2	2	2	2	2	11	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	5	5	3	3	3	3	0	0	0	0	0	0	0	0	0	0	12	5	5	5	5	5	5	2	2	2	2	2	2	2	12	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	13	5	5	5	5	5	5	2	2	2	2	2	2	2	13	3	3	3	3	3	3	0	0	0	0	0	0	0

Control actions are observed to change under Cox-2 stage information. Consider the states where it is optimal to produce at all stations (action 7). It is observed for every x_1 in Table 3.7, the area of action (7) under the information that the first or second Coxian phase is busy is greater than when both Coxian phases are idle. In addition, the policy has slight changes depending on the busy stages of Cox-2. $action\{(x_1, x_2, x_3|y_1, y_2) = (2,3,5|1,0)\} = 4$ states that it is optimal for station 3 to remain idle and stations 1 and 2 to produce if the first Coxian phase is busy. On the other hand, it is optimal to activate stations 1 and 3 if the second Coxian phase is operating, i.e. $action\{(x_1, x_2, x_3|y_1, y_2) = (2,3,5|0,1)\} = 6$. Such observations are general to any extended models depending on the values of Cox-2 parameters β, γ and demand rate λ. The effect of stage information on the optimal policy is prevalent at higher λ values.

The observation leading to the proposition of Chapter 3.1 " *There is a threshold* T_i *for* $x_i \geq 0, i \in \{1,2\}$ *such that it is optimal not to produce at station-*i *when* $x_i > T_i$ *for all* $x_j, j \in \{1,2,3\} - \{i\}$." for the basic model is also identified in the extended models. The threshold for the example in Table 3.7 is shown as $T_1 = 4$ when $(y_1, y_2) = (0,0)$, it is $T_1 = 5$ when $(y_1, y_2) = (1,0)$ and $(y_1, y_2) = (0,1)$.

The numerical results for the extended models are presented in Figures 3.11 – 3.19. For the sake of interpretation of the Coxian parameters, the results are shown as optimal average cost versus visiting probability β considering three different values of γ in each model. In addition, four different demand levels are considered. The parameters used in numerical studies are shown in Table 3.6.

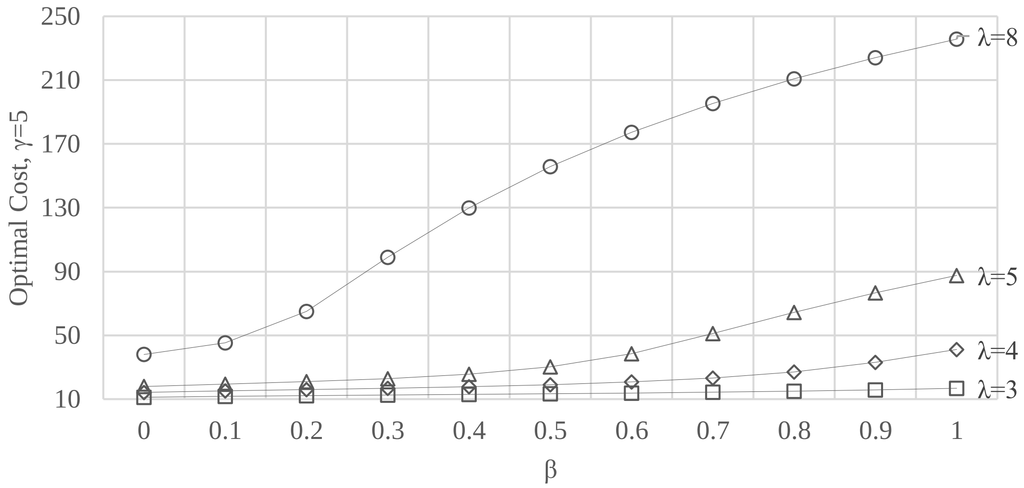

Figure 3.11. Optimal cost versus β (γ=5), Model-1

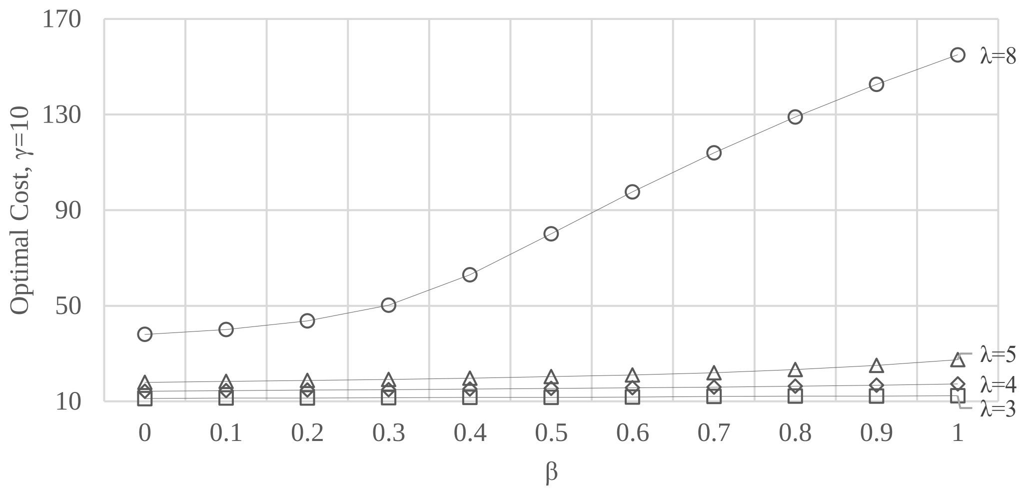

Figure 3.12. Optimal cost versus β (γ=10), Model-1

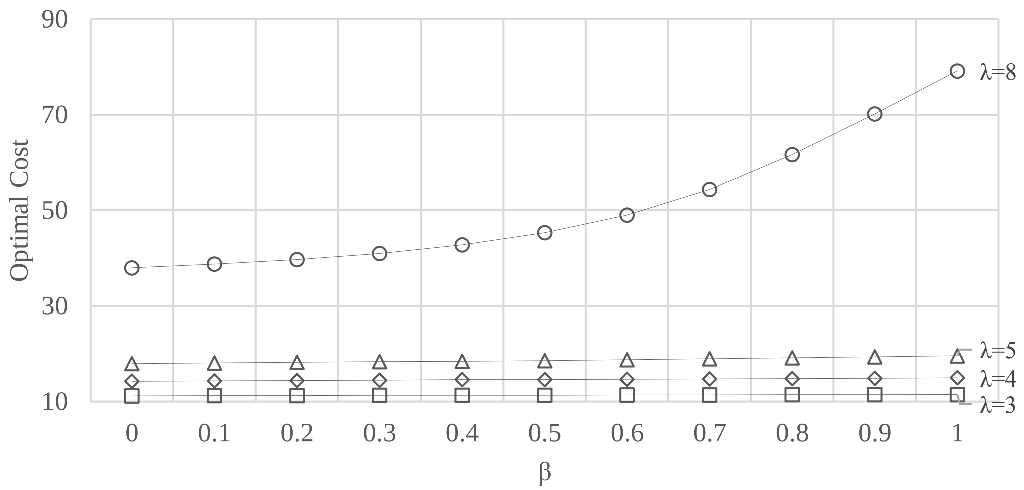

Figure 3.13. Optimal cost versus β (γ=20), Model-1

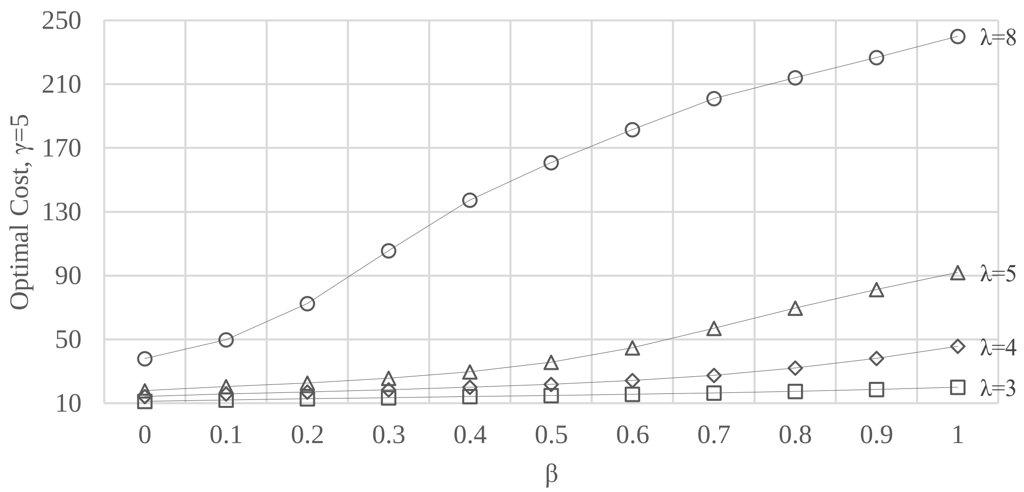

Figure 3.14. Optimal cost versus β (γ=5), Model-2

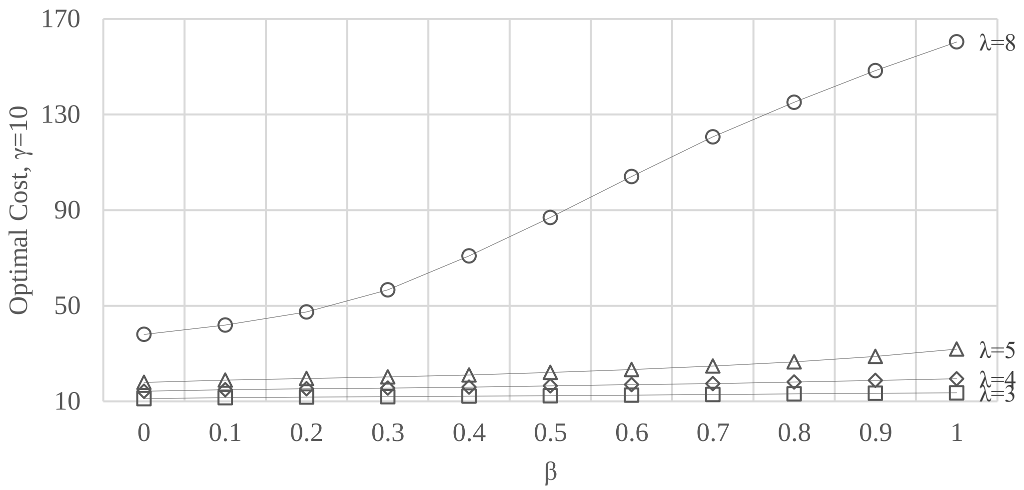

Figure 3.15. Optimal cost versus β (γ=10), Model-2

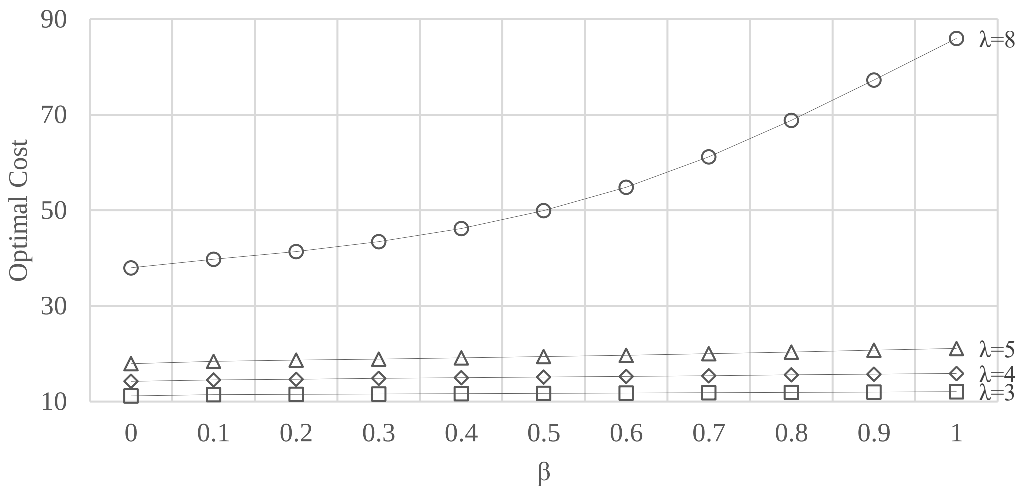

Figure 3.16. Optimal cost versus β (γ=20), Model-2

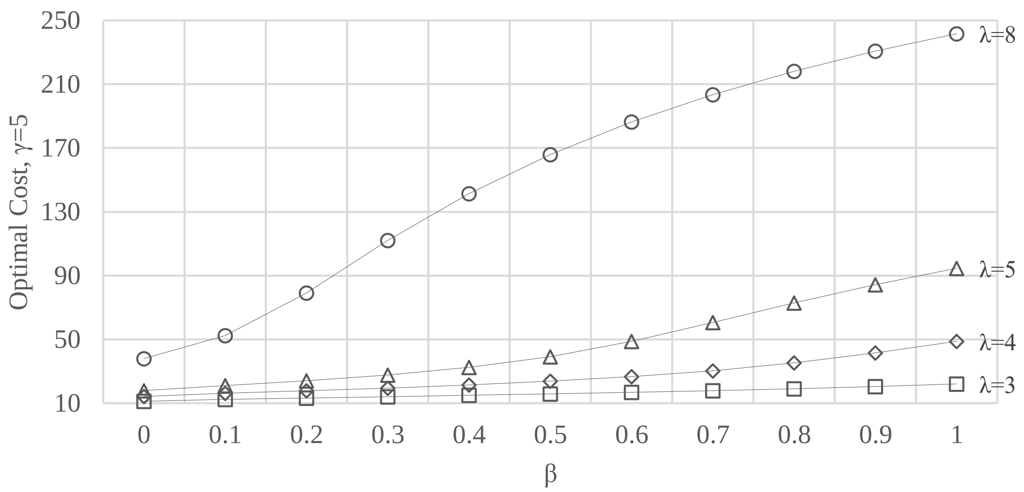

Figure 3.17. Optimal cost versus β (γ=5), Model-3

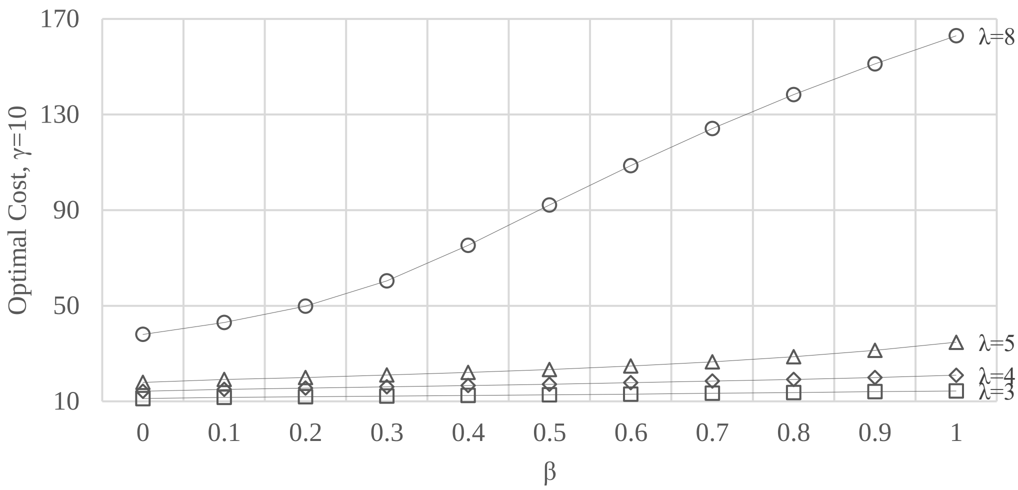

Figure 3.18. Optimal cost versus β (γ=10), Model-3

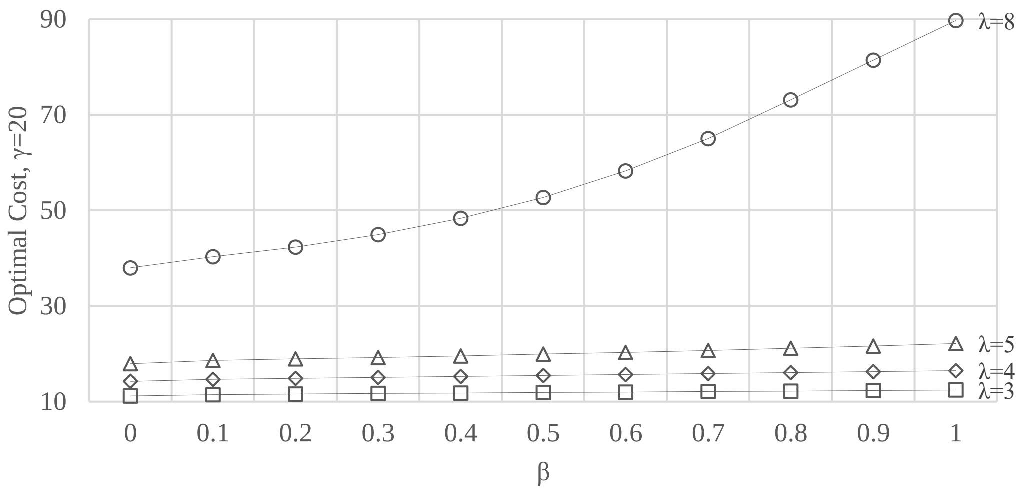

Figure 3.19. Optimal cost versus β (γ=20), Model-3

It is observed for all models that

- At a given λ, the average cost is an increasing function of β for each γ, and it is a decreasing function of γ for each β. The effect of β on the average cost is noticeable at relatively higher values of λ and γ. As λ decreases and γ increases, the cost curve flattens.
- For any γ – β pair, the average cost is an increasing function of λ. The percentage cost increment due to λ increases with β and decreases with γ.
- Although the cost structure follows a similar pattern between models, at any intersection of λ, β and γ, locating a Cox-2 distributed machine at the downstream stage (model- 3) produces the highest system cost. In contrast, the

minimum system cost consists of a system with a Cox-2 distributed machine at the upstream stage (model-1).

The optimal policy is observed to be highly sensitive to visiting probability β when $\gamma < 20$. For those instances, state space, hence the action space increases with β. On the other hand, it is observed at $\gamma = 20$ that the optimal policy does not change after a certain value of β. As $\gamma \rightarrow \infty$, it is expected that the effect of β on both optimal policy and optimal cost would be diminished.

CHAPTER 4
PROPOSED PRODUCTION POLICIES

This chapter proposes alternative approaches to the optimal control problems of three station make-to-stock systems. Optimal production control requires action at each state of the system. Thus, it poses a challenge on computation time of production lines as state spaces enlarge. The fact that the curse of dimensionality in finding optimal policies remains, alternative approaches have become prominent. Alternative policies would be expected to be easy to apply and perform near-optimal. In this chapter, an alternative policy called *no intentional idleness*, which corresponds to an approach where machines operate whenever possible, is considered. Production is merely interrupted with the occurrence of blocking and starvation, and failures if exist. A station is blocked if its service is completed, but its downstream buffer has no room. If a station is idle because its upstream buffer has no item, then it is starved.

The proposed policy eliminates production control decisions (control actions) and constitutes an approximate solution to the optimal control problems. A substantial amount of research has been conducted to analyse production lines under the *no intentional idleness* approach. However, throughput is one of the most studied metrics for systems considering several configurations of machines and buffers (J. Li et al., 2009).

Another important aspect is the -no finished goods buffer to meet the demand- assumption (Diamantidis et al., 2020; H. T. Papadopoulos et al., 1989, 1990) in the performance analysis under *no intentional idleness*. This chapter presents an important contribution to the production systems literature by considering three-station make-to-stock flow lines under the proposed approach and its performance comparison to the optimal policy. The proposed approach and the optimal policy have not been compared in three-station make-to-stock systems to the extent of our knowledge.

The proposed policy is presented for the basic model and extended models in Chapter 4.1. The basic model consists of Exponential processing times, while extended models include Exponential and two-phase Coxian processing times to incorporate real-world features into our problems. It is observed that the proposed policy performs near-optimal for models with Cox-2 processing times. The average optimality gap is calculated as less than 3% in numerical experiments conducted with 396 instances. For the basic model with Exponential processing times, the performance of the proposed policy alternates depending on the demand rate and production rates of machines. The performance of the proposed policy deteriorates in cases with lower demand rates. A modified version of the proposed approach is developed for such cases and presented in Chapter 4.2.

4.1. No Intentional Idleness Policy

The proposed *no intentional idleness* (NI) policy relies on eliminating production control decisions and letting machines produce whenever possible. Production could be temporarily suspended due to blocking and starvation. In the optimal control problem described in Chapter 3, buffer capacities are assumed to be infinite, i.e. blocking does not occur. However, NI policy may let infinitely many finished goods be produced under this assumption. While imposing the policy, we set finite buffer capacities for the make-to-stock flow line model to indirectly control the production. Blocking and starvation interrupt production; although a station is operational, it could not operate on an item. The cost structure is defined as follows: Holding cost rate h_j is charged for the items in j^{th} buffer. The same cost is incurred for the item in j^{th} station if the station is blocked. For each demand that cannot be met from the finished goods buffer, lost sales cost c is paid.

Let $m_j \geq 0$ be the capacity of buffer j, j=1,2,3. In case of $m_j = 0$, a completed item at station j causes blocking at the station until the machine at its downstream becomes idle. Our objective is to find optimum buffer capacities m_1^*, m_2^*, m_3^* that minimize the long-run average system cost composed of holding and lost sales costs.

The proposed policy is modeled using ARENA Simulation Software 2019. Our problem is a steady-state simulation model which has no natural termination criterion. However, we define a stopping criterion based on a certain number of events occurring in the system, revealing that the statistics stabilize. Then, the long-run average system cost and throughput is collected for given values of m_1, m_2 and m_3. A single replication is considered; a counter variable is defined to keep a count of the number of times station-3 has seized an entity, which terminates the replication after reaching a specified number. Figure 4.1 depicts an example of the system throughput and average cost as functions of the counter variable (which keeps a count of the number of times station-3 has seized an entity). It is noted that the statistics approach steady-state after the counter reaches 2.75 million, two-digit accuracy after the decimal separator is obtained in both statistics.

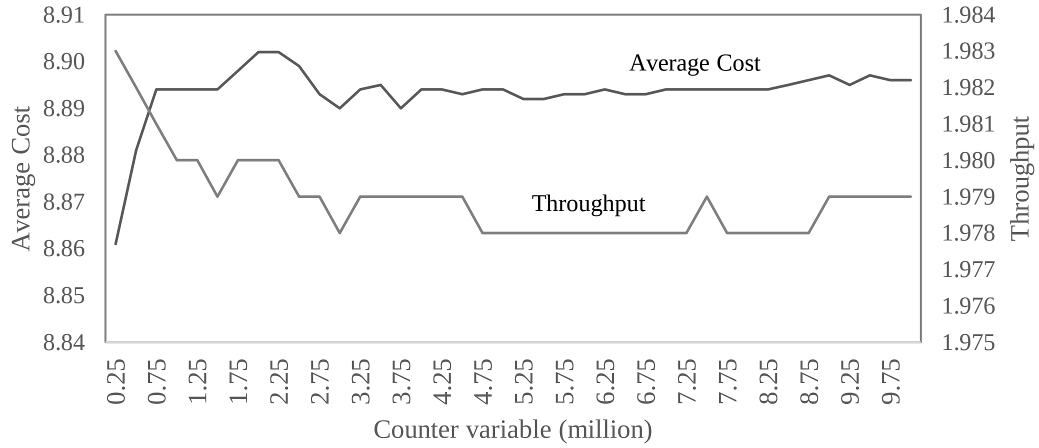

Figure 4.1. System throughput and average cost as functions of event occurrence

The cost structure is observed by inspection: a wide range of elements are considered for variables m_1, m_2, m_3 in several settings and the buffer capacities m_1^*, m_2^*, m_3^* that minimize the average system cost is obtained. Assuming the cost structure is convex based on numerical inspection, optimal buffer capacities are obtained with an exhaustive search over the state space of the proposed policy. The experiments are carried out on a Core i7, 2.80 GHz, 16 GB RAM computer. The no intentional idleness policy results are then compared to the optimal policy studied in Chapter 3. In addition to the time-average cost criterion, simulation allows us to obtain the system's throughput.

We first consider the basic model with Exponential processing times as described in Chapter 3.1. The system parameters are set as $\lambda = \{1, .., 10\}, c = 50, h_1 = 1, h_2 = 1.5, h_3 = 2$. In total, seven cases are created with different production rates, as shown in Table 3.4 of Chapter 3.1. The base case scenario is defined as case 0 with production rates of (μ_1, μ_2, μ_3) = (10, 10, 10) while the remaining cases are designed as follows: (5, 10, 10) in case 1, (10, 5, 10) in case 2, (10, 10, 5) in case 3, (15, 10, 10) in case 4, (10, 15, 10) in case 5 and (10, 10, 15) in case 6, as shown in Table 3.4 of Chapter 3.1.

The optimality gap (%) of the proposed policy is presented in Figures 4.2 and 4.3 as λ increases. In both figures, holding costs constitute most of the system cost of the proposed policy at lower λ values. However, the optimality gaps of cases 1-2-3 converge to zero as λ increases, as shown in Figure 4.2. In Figure 4.3, optimality gaps of remaining cases 0-4-5-6 follow the same pattern, while case 4 presents the maximum gap. On the other hand, gaps are obtained as 1.55% (case 0), 2.24% (case 4), 1.17% (case 5) and 1.01% (case 6) at λ = 10.

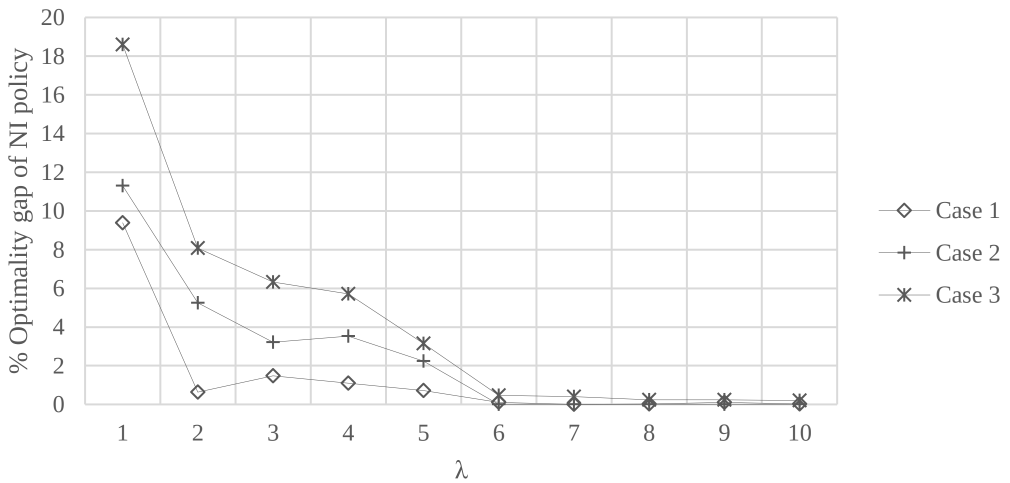

Figure 4.2. Optimality gap (%) of cases 1-2-3 versus demand rate

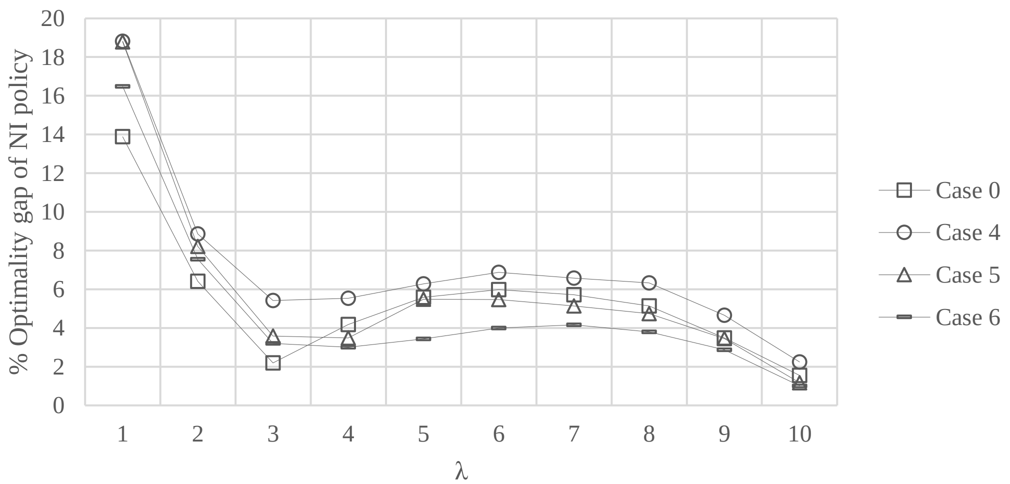

Figure 4.3. Optimality gap (%) of cases 0-4-5-6 versus demand rate

Figure 4.4 presents throughput of the cases 0-6 under given that the optimal buffer capacities m_1^*, m_2^*, m_3^* that minimize the long-run average system cost. The first three cases consist of a machine with a production rate of 5, reaching their maximum throughput values at λ = 6. In addition, those cases produce higher system costs than case 0 in optimal production control policy, as shown in Chapter 3.1. The throughput values of cases 0-4-5-6 increase with λ, while case 0 provides a slightly lower throughput than cases with a faster machine. Case 6, in which a faster machine is assigned to the downstream location, has the maximum throughput, with the second maximum being case 5 having a faster machine at the intermediate stage. In optimal control problems of the basic model in Chapter 3.1, it is observed that cases 6 and 5 produce the minimum and the second minimum system costs.

It is shown in (Buzacott & Shanthikumar, 1993) that the throughput of a three-station flow line is maximized if the fastest station is located in the middle stage and the slowest stations are located in the first and third stages. The study assumes finite intermediate buffers but no finished goods buffer. On the contrary, our study consists of a make-to-stock production setting with a well-defined cost objective. In numerical experiments, the maximum throughput is obtained in case 6, where the faster machine is assigned to station-3, and case 5 provides a slightly lower throughput.

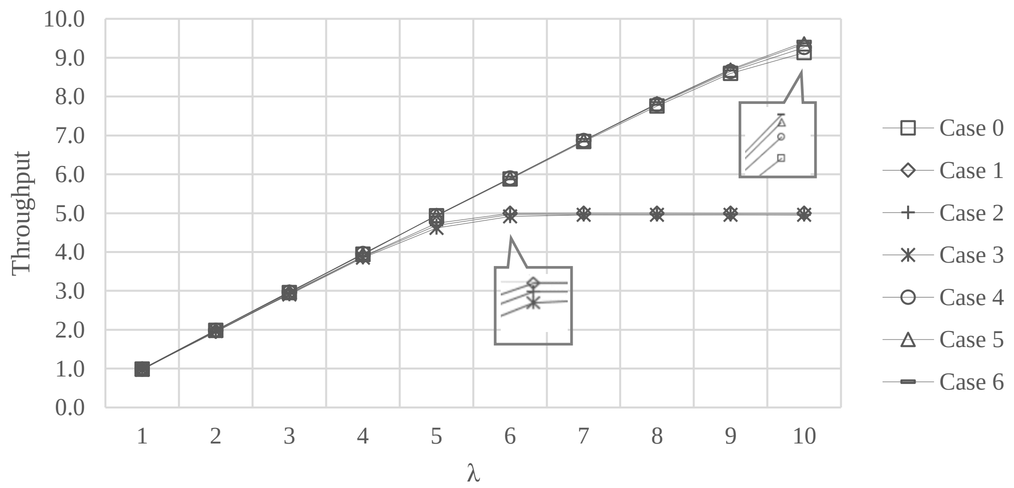

Figure 4.4. Throughput of the cases under m_1^*, m_2^*, m_3^*

The optimal buffer capacities (m_1^*, m_2^*, m_3^*) of the cases under the proposed policy are presented in Table 4.1. For any given λ, it is observed in cases 1-2-3 that while the location of the slower machine moves downstream, m_1^* tends to decrease and m_3^* tends to increase. On the contrary, while the location of the faster machine moves downstream (cases 4-5-6), m_1^* tends to increase and m_3^* tends to decrease.

Moreover, for the systems having a slower machine, a non-monotone trend is observed in the optimal capacity of its downstream buffer as λ increases. Consider the results of

m_1^* in case 1, as shown in Table 4.1: m_1^* increases until λ = 6, and then it decreases, recalling that case 1 represents the model with a slower machine at the first station. The same structure is preserved in the results of m_2^* in case 2 and m_3^* in case 3, the same breaking point at λ = 6 represents their convergence point of throughput, as shown in Figure 4.4. On the other hand, for cases 4 to 6 and 0, optimal buffer capacities increase with λ.

Table 4.1. Optimal buffer capacities of the cases

	Case 0			Case 1			Case 2			Case 3			Case 4			Case 5			Case 6		
λ	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
2	0	0	2	0	0	2	0	0	2	0	0	3	0	0	2	0	0	2	0	0	2
3	0	0	3	1	1	3	0	2	3	0	0	5	0	0	3	0	0	3	0	0	3
4	0	1	4	4	1	4	1	4	5	0	1	9	0	1	4	0	0	4	0	1	3
5	1	1	6	12	2	5	3	10	6	0	3	15	0	1	6	1	1	5	1	2	4
6	2	2	7	27	2	5	4	24	6	1	4	26	1	2	7	1	1	7	2	2	5
7	3	4	9	24	2	5	4	19	5	1	5	23	1	4	9	2	2	9	3	4	6
8	5	6	12	19	2	5	4	17	5	1	5	19	2	6	12	4	3	12	6	6	7
9	8	10	17	15	3	5	4	16	5	1	5	17	3	9	17	6	5	16	9	10	9
10	11	17	16	13	3	6	4	13	6	1	5	16	4	15	24	11	7	24	17	18	10

In addition to the basic model with Exponential processing times, the performance of extended models with Cox-2 processing times under the proposed approach is compared with the optimal policy. Extended models consist of three different designs, and their optimal control problems are presented in Chapter 3.2. While processing times of remaining machines are Exponentially distributed, Model-1 consists of a Cox-2 distributed machine at the upstream stage (Figure 3.8), processing times are Cox-2 at the intermediate station in Model-2 (Figure 3.9), and Cox-2 processing times are assigned to station at the downstream stage in Model-3 (Figure 3.10).

Numerical experiments of extended models are based on the set of parameters given in Table 3.6 of Chapter 3.2. Holding cost rates are set to $[h_1, h_2, h_3] = [1,1.5,2]$, lost sales cost c is set to 50. Production rates of Exponentially distributed machines and the first phase of Cox-2 distributed machine are equal to 10. The production rate of the second Coxian phase is defined as $\gamma \in \{5,10,20\}$. The last Coxian parameter, visiting probability β, varies from 0 to 1. Four different demand rates are considered such that $\lambda = \{3,4,5,8\}$. The set of parameters selected for the experiments creates different processing time moments for Coxian and analyzes different rework/failure characteristics considering moderate and higher demand rates.

Figures 4.5 – 4.13 present the optimality gap (%) of the proposed policy for extended models. In total, 360 instances are created and solved, and the maximum optimality gap of a single instance is recorded as less than 6%. It is important to note that the performance of the extended models against the optimal policy is quite similar to each other. However, when $\gamma = 5$, it is observed that Model-3 produces the highest average optimality gap at every demand rate. Model-1 delivers a slightly higher optimality gap considering all of the instances. The overall gap (%) of the models are calculated as 2.33 (Model-1), 1.89 (Model-2), and 2.14 (Model-3). At $\lambda = 8$, the average cost of the proposed policy converges to the optimal policy as β increases. The expected time to production completion increases with visiting probability. The expected value of a Cox-2 random variable X denoting the processing time of a machine with parameters (μ, γ, β) is calculated as $\frac{1}{\mu} + \frac{\beta}{\gamma}$. In general, lower γ and higher β constitute the worst-case scenario in terms of production completion time. Production control policy tends to produce whenever possible, according to the observed optimality gap results. On the other hand, the performance of the proposed policy slightly deteriorates at higher γ values.

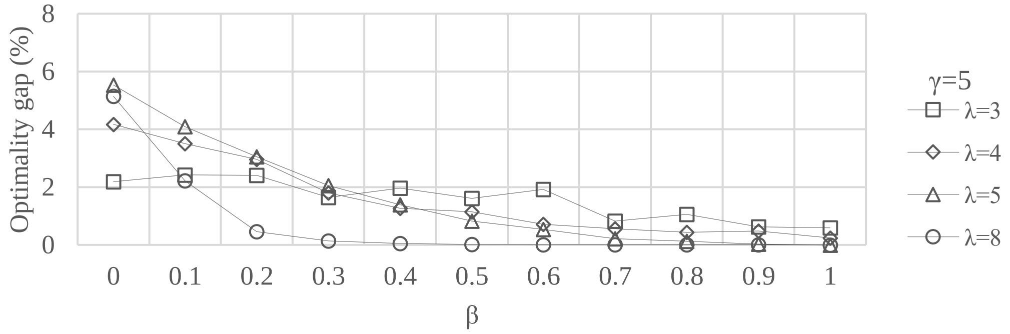

Figure 4.5. Optimality gap (%) of Model-1, γ=5

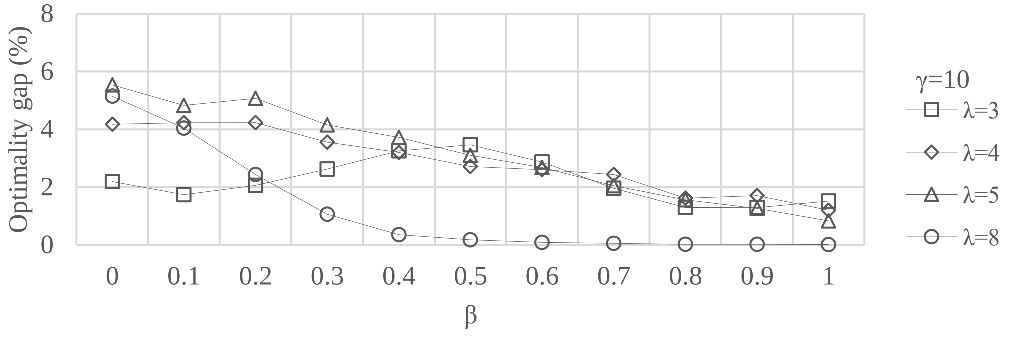

Figure 4.6. Optimality gap (%) of Model-1, γ=10

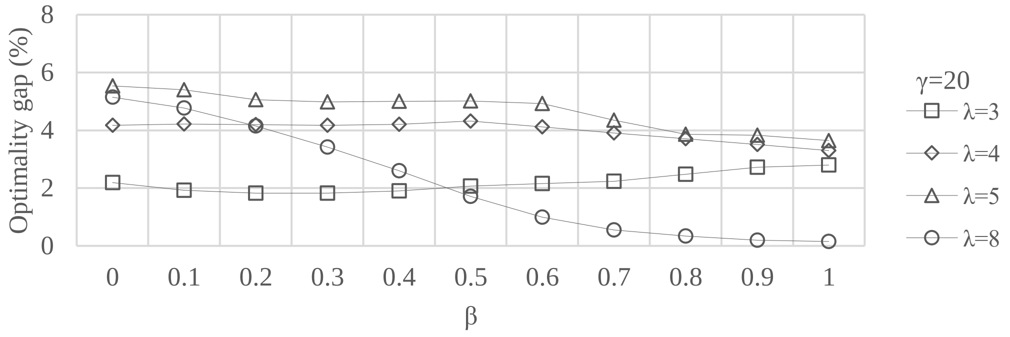

Figure 4.7. Optimality gap (%) of Model-1, γ=20

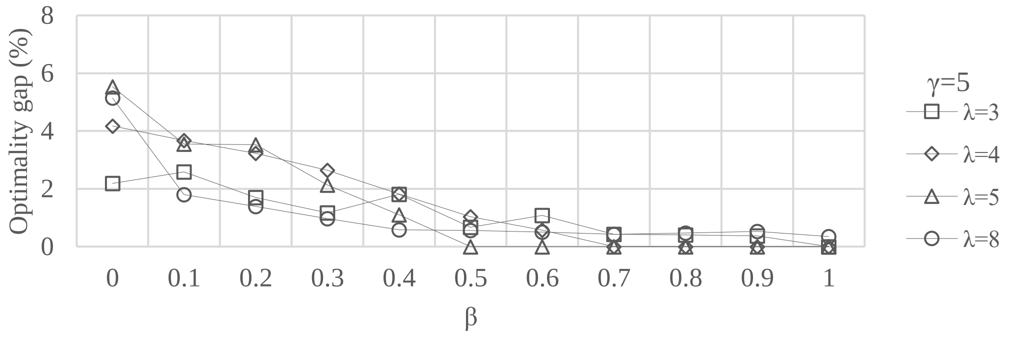

Figure 4.8. Optimality gap (%) of Model-2, γ=5

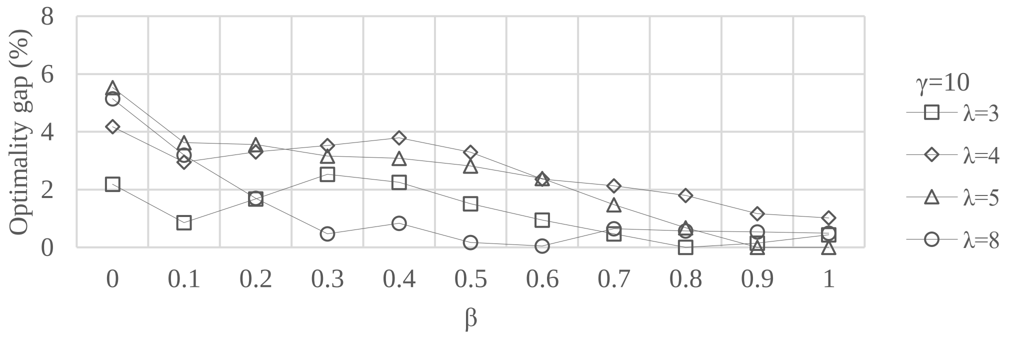

Figure 4.9. Optimality gap (%) of Model-2, γ=10

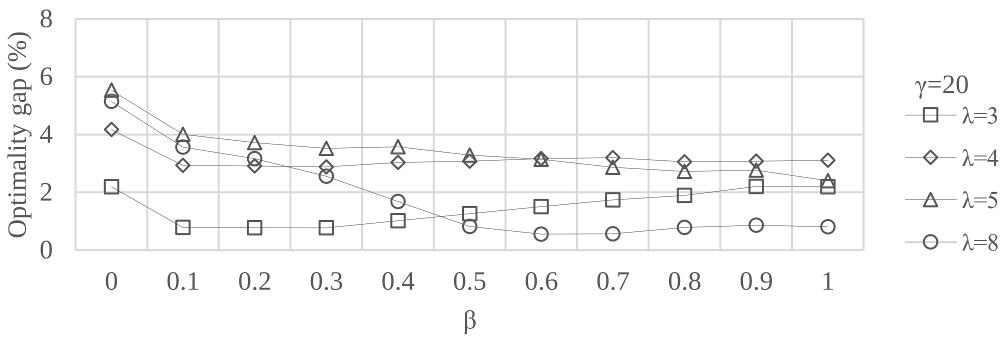

Figure 4.10. Optimality gap (%) of Model-2, γ=20

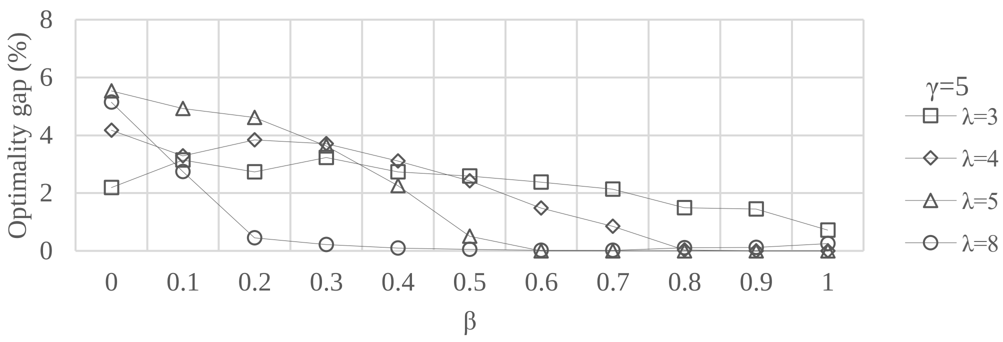

Figure 4.11. Optimality gap (%) of Model-3, γ=5

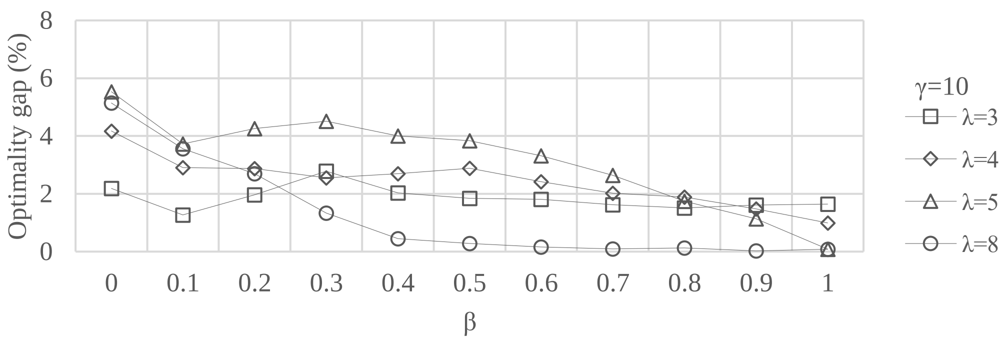

Figure 4.12. Optimality gap (%) of Model-3, γ=10

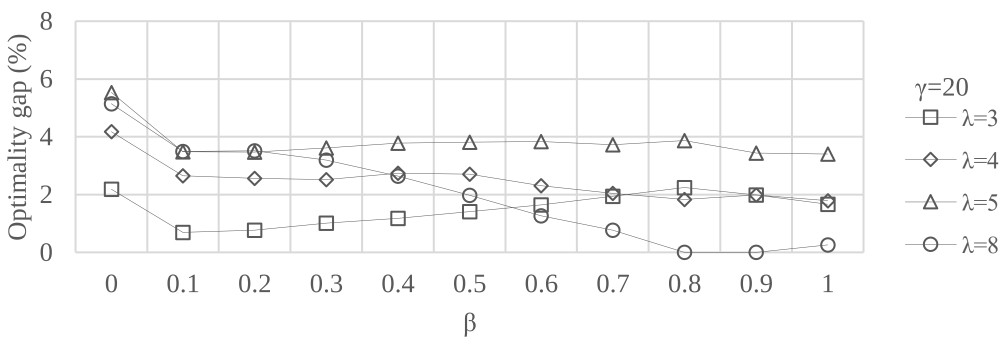

Figure 4.13. Optimality gap (%) of Model-3, γ=20

Table 4.2. Throughput under m_1^*, m_2^*, m_3^*

		λ=3			λ=4			λ=5			λ=8		
	β	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20
Model-1	0	2.96	2.96	2.96	3.94	3.94	3.94	4.93	4.93	4.93	7.76	7.76	7.76
	0.1	2.94	2.95	2.95	3.94	3.93	3.93	4.91	4.91	4.92	7.66	7.73	7.75
	0.2	2.95	2.95	2.95	3.92	3.95	3.93	4.89	4.90	4.92	7.12	7.67	7.74
	0.3	2.94	2.94	2.95	3.91	3.94	3.93	4.88	4.91	4.91	6.25	7.51	7.72
	0.4	2.93	2.94	2.95	3.91	3.94	3.92	4.85	4.89	4.91	5.56	7.13	7.69
	0.5	2.94	2.93	2.95	3.89	3.93	3.95	4.74	4.90	4.90	5.00	6.67	7.63
	0.6	2.92	2.95	2.95	3.87	3.92	3.95	4.51	4.91	4.93	4.55	6.25	7.53
	0.7	2.92	2.95	2.95	3.82	3.93	3.95	4.16	4.86	4.92	4.17	5.88	7.36
	0.8	2.92	2.95	2.94	3.73	3.92	3.94	3.85	4.87	4.91	3.85	5.56	7.14
	0.9	2.90	2.94	2.94	3.56	3.91	3.94	3.57	4.82	4.90	3.57	5.26	6.90
	1	2.88	2.94	2.94	3.33	3.91	3.94	3.33	4.76	4.89	3.33	5.00	6.67
		λ=3			λ=4			λ=5			λ=8		
	β	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20
Model-2	0	2.96	2.96	2.96	3.94	3.94	3.94	4.93	4.93	4.93	7.76	7.76	7.76
	0.1	2.93	2.95	2.95	3.93	3.93	3.93	4.90	4.91	4.92	7.60	7.71	7.74
	0.2	2.95	2.94	2.95	3.92	3.91	3.93	4.88	4.92	4.91	7.03	7.63	7.71
	0.3	2.93	2.94	2.95	3.92	3.93	3.93	4.86	4.90	4.90	6.20	7.42	7.69
	0.4	2.95	2.96	2.95	3.90	3.92	3.92	4.78	4.90	4.90	5.51	7.05	7.65
	0.5	2.92	2.96	2.94	3.88	3.93	3.92	4.69	4.87	4.92	4.96	6.60	7.56
	0.6	2.93	2.95	2.94	3.82	3.92	3.94	4.47	4.87	4.92	4.53	6.22	7.46
	0.7	2.90	2.95	2.94	3.76	3.91	3.93	4.14	4.85	4.91	4.15	5.85	7.29
	0.8	2.89	2.94	2.94	3.68	3.93	3.93	3.82	4.82	4.90	3.84	5.52	7.05
	0.9	2.86	2.93	2.93	3.53	3.91	3.92	3.56	4.80	4.88	3.56	5.24	6.84
	1	2.87	2.93	2.96	3.32	3.88	3.92	3.33	4.72	4.90	3.33	4.97	6.63
		λ=3			λ=4			λ=5			λ=8		
	β	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20	γ=5	γ=10	γ=20
Model-3	0	2.96	2.96	2.96	3.94	3.94	3.94	4.93	4.93	4.93	7.76	7.76	7.76
	0.1	2.96	2.95	2.95	3.93	3.92	3.93	4.88	4.90	4.92	7.56	7.70	7.73
	0.2	2.94	2.94	2.95	3.92	3.95	3.93	4.87	4.91	4.91	6.97	7.57	7.70
	0.3	2.92	2.97	2.95	3.91	3.93	3.92	4.83	4.88	4.90	6.17	7.35	7.67
	0.4	2.93	2.96	2.94	3.88	3.92	3.92	4.75	4.89	4.89	5.47	7.00	7.60
	0.5	2.91	2.95	2.94	3.85	3.90	3.95	4.62	4.89	4.92	4.95	6.55	7.51
	0.6	2.91	2.95	2.94	3.79	3.92	3.95	4.43	4.85	4.90	4.51	6.17	7.39
	0.7	2.88	2.94	2.93	3.72	3.90	3.94	4.10	4.83	4.89	4.13	5.82	7.23
	0.8	2.88	2.93	2.97	3.61	3.88	3.93	3.82	4.80	4.88	3.81	5.51	7.01
	0.9	2.84	2.92	2.96	3.45	3.90	3.93	3.54	4.75	4.90	3.55	5.22	6.79
	1	2.82	2.95	2.96	3.27	3.87	3.92	3.32	4.69	4.89	3.32	4.95	6.60

Throughput rates of the instances under the proposed policy with optimal buffer capacities (m_1^*, m_2^*, m_3^*) are presented in Table 4.2 for every instance of the extended models. According to the results,

- Throughput is an increasing function of demand rate λ at any values of visiting probability β.
- Throughput sharply decreases in instances when $\gamma = 5$ as β increases. However, the effect of β on throughput is reduced as γ increases.
- Throughput is less sensitive in changes of parameters at lower demand rates.
- Although effects of system parameters on throughput pursue a similar pattern in every model, Model-1, which constitutes a design with Cox-2 processing times at the upstream stage, produces a slightly higher throughput rate.

Table 4.3. An example of optimal buffer capacities of the extended models

	λ=5, γ=5									λ=8, γ=5								
	Model-1			Model-2			Model-3			Model-1			Model-2			Model-3		
β	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*
0	1	1	6	1	1	6	1	1	6	5	6	12	5	6	12	5	6	12
0.1	2	1	6	1	2	6	1	1	7	11	7	12	7	11	13	4	8	19
0.2	3	2	5	1	4	6	1	2	8	26	8	11	8	24	12	4	8	32
0.3	4	2	6	2	5	6	1	2	10	25	5	8	7	25	8	3	7	27
0.4	7	2	6	2	7	6	1	2	12	25	3	6	5	19	6	2	5	21
0.5	12	2	5	3	10	6	1	3	14	19	2	5	4	18	5	1	5	20
0.6	19	1	5	3	17	5	1	3	19	16	2	4	4	15	5	1	4	14
0.7	24	1	4	3	22	5	1	3	24	14	1	4	3	12	4	1	3	11
0.8	22	1	3	2	24	4	1	2	24	9	1	3	3	11	4	1	2	10
0.9	16	0	3	2	23	3	1	2	24	7	1	3	2	8	3	1	2	8
1	14	0	3	2	18	3	1	2	22	9	1	3	2	7	3	1	2	8

Optimal buffer capacities (m_1^*, m_2^*, m_3^*) of the instances are provided in Appendix 3. The effect of β on optimal buffer capacities m_1^*, m_2^*, m_3^* is significant at $\gamma = 5$. Table 4.3 presents an example of the buffer capacities for two different demand rates $\lambda = 5$ and $\lambda = 8$ when $\gamma = 5$. It is observed in all instances that, at γ = 5, the optimal capacity of the downstream buffer of Cox-2 distributed machine rapidly increases until a certain threshold of β, and then it starts to diminish. For instance, the maximum of m_1^* of Model-1 is observed when $\beta = 0.7$ at $\lambda = 5$, and it is $\beta = 0.2$ at $\lambda = 8$, as seen in Table 4.3. However, the instances with higher values of γ and lower values of λ,

m_1^*, m_2^*, m_3^* slightly change with β. For example, when λ=3 and γ = 20, optimal capacities are obtained as $(m_1^*, m_2^*, m_3^*) = (1,1,3)$ for every instance of Model-3 (see Appendix 3). Furthermore, the total space required in buffers decreases in every model as γ increases.

In the analysis of the basic model with Exponential processing times, it is observed that holding cost constitutes the major part of the system cost of the proposed policy at lower λ values. Due to the nature of the proposed approach, the first station is either busy or blocked, but it is never starved because the raw material supply is ample. For this reason, Chapter 4.2. proposes an easy-to-apply control to the supply process of the basic model to improve the performance of the proposed approach.

4.2. Extended No Intentional Idleness Policies

The proposed *no intentional idleness* policy authorizes production for every machine whenever possible. The long-run average system cost consists of holding and lost sales costs, and the objective is to obtain buffer capacities (m_1^*, m_2^*, m_3^*) that minimize the system cost. Even though an optimal value for a buffer is zero, holding cost is incurred for each blocked machine. For the basic model with Exponential processing times, the performance of the proposed policy depends on the demand and production rates. The performance of the proposed approach deteriorates in cases with lower demand rates because holding cost causes a majority of the system cost. For such cases, this chapter presents a modified version of the alternative approach to improve the performance.

Due to the ample raw material supply assumption, station-1 is never idle; however, stations 2 and 3 could be starved. The main idea is to propose an easy-to-apply mechanism for the raw-material supply process. Initially, five different stopping criteria are defined as follows:

NI Alternative-1 (NI -1): raw material is released when station-2 or station-3 is idle.

NI Alternative-2 (NI -2): raw material is released when station-2 is idle.

NI Alternative-3 (NI -3): raw material is released when station-3 is idle.

NI Alternative-4 (NI -4): raw material is released when both station-2 and station-3 is idle.

NI Alternative-5 (NI -5): raw material is released when either station-2 or station-3 is idle.

In each alternative, station-1 remains idle until the corresponding criterion is met since the raw material is not released. Thus, the aim is to prevent excessive holding costs due to ample supply assumption.

A set of test runs are carried out to evaluate the performance of each alternative policy, and results are presented in Tables 4.4 and 4.5. Cases 2, 3 and 5 are selected for the test runs while the demand rate varies from 1 to 4. It is observed that every proposed alternative outperforms the NI policy at λ =1. It also noted that every alternative approach requires more buffer spaces than the proposed policy at every instance (see Table 4.5). However, Alternative-1 (NI -1) results in the best performance among other rules, noting that NI -5 performs the same as NI -1 except λ =2 of case 3. Table 4.6 presents a performance comparison between the proposed policy, NI-1 and the optimal policy for all seven cases of the basic model.

According to the average optimality gaps (%) of the proposed policies in Table 4.6, significant improvement is achieved with NI-1 comparing to NI in every instance except case 1. In case 1, the cost improvement is observed only at λ =1 in NI -1, and the policy deteriorates as λ increases. On the other hand, NI policy provides a minimum average optimality gap (%) with case 1, as shown in Table 4.6.

Table 4.4. Test results – (1)

	λ	Optimal Cost	Optimality Gap (%)					
			NI -1	NI -2	NI -3	NI -4	NI -5	NI
Case 2	1	6.20	0.00	0.32	0.00	4.02	0.00	11.30
	2	9.92	0.40	4.98	2.07	20.83	0.40	5.25
	3	14.47	2.49	19.17	9.34	55.91	2.49	3.21
	4	21.31	7.95	49.85	14.18	72.78	7.95	3.53
Case 3	1	6.39	7.79	7.79	7.79	1.84	7.79	18.60
	2	10.36	2.54	3.27	2.63	17.97	2.63	8.07
	3	15.25	5.40	7.36	7.52	53.75	5.40	6.33
	4	22.96	6.21	9.78	19.30	70.73	6.21	5.71
Case 5	1	5.49	5.51	5.34	3.17	1.96	5.51	18.79
	2	8.04	0.25	1.11	0.25	7.69	0.25	8.22
	3	10.49	2.87	6.92	2.87	25.76	2.87	3.58
	4	13.30	3.90	11.10	6.14	52.90	3.90	3.48

Table 4.5. Test results – (2)

	λ	Optimal Buffer Capacities (m_1^*, m_2^*, m_3^*) NI-1	NI-2	NI-3	NI-4	NI-5	NI
Case 2	1	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,2)	(0,0,1)	(0,0,1)
	2	(1,1,2)	(0,1,2)	(0,0,3)	(0,0,5)	(1,1,2)	(0,0,2)
	3	(1,2,4)	(0,5,3)	(2,1,5)	(0,0,15)	(1,2,4)	(0,2,3)
	4	(2,5,7)	(0,7,6)	(4,1,9)	(0,0,10)	(2,5,7)	(1,4,5)
Case 3	1	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,2)	(0,0,1)	(0,0,1)
	2	(0,0,3)	(0,0,3)	(0,0,3)	(0,0,5)	(0,0,3)	(0,0,3)
	3	(1,1,5)	(0,1,5)	(1,0,6)	(0,0,15)	(1,1,5)	(0,0,5)
	4	(1,2,9)	(0,3,9)	(1,0,12)	(0,0,11)	(1,2,9)	(0,1,9)
Case 5	1	(0,0,1)	(0,0,1)	(0,1,1)	(0,1,1)	(0,0,1)	(0,0,1)
	2	(0,0,2)	(0,0,2)	(0,0,2)	(0,0,3)	(0,0,2)	(0,0,2)
	3	(1,0,3)	(0,1,3)	(1,0,3)	(0,0,6)	(1,0,3)	(0,0,3)
	4	(2,1,4)	(1,2,4)	(1,0,5)	(0,0,13)	(2,1,4)	(0,0,4)

Table 4.6. Performance Comparison: Optimal Policy vs NI vs NI -1

	Parameters				Optimal	NI -1					NI				
	λ	μ_1	μ_2	μ_3	Cost	Gap%	Avg Gap%	m_1^*	m_2^*	m_3^*	Gap%	Avg Gap%	m_1^*	m_2^*	m_3^*
Case 0	1	10	10	10	5.84	0.51	2.1	0	0	1	13.88	6.7	0	0	1
	2	10	10	10	8.32	0.60		0	0	2	6.41		0	0	2
	3	10	10	10	11.19	2.19		1	1	3	2.19		0	0	3
	4	10	10	10	14.23	5.20		1	1	5	4.18		0	1	4
Case 1	1	5	10	10	6.08	0.00	6.3	0	0	1	9.39	3.1	0	0	1
	2	5	10	10	9.41	2.89		1	1	2	0.63		0	0	2
	3	5	10	10	13.42	6.42		2	2	4	1.47		1	1	3
	4	5	10	10	19.05	16.08		3	2	8	1.09		4	1	4
Case 2	1	10	5	10	6.20	0.00	2.7	0	0	1	11.30	5.8	0	0	1
	2	10	5	10	9.92	0.40		1	1	2	5.25		0	0	2
	3	10	5	10	14.47	2.49		1	2	4	3.21		0	2	3
	4	10	5	10	21.31	7.95		2	5	7	3.53		1	4	5
Case 3	1	10	10	5	6.39	7.79	5.5	0	0	1	18.60	9.7	0	0	1
	2	10	10	5	10.36	2.54		0	0	3	8.07		0	0	3
	3	10	10	5	15.25	5.40		1	1	5	6.33		0	0	5
	4	10	10	5	22.96	6.21		1	2	9	5.71		0	1	9

Table 4.6 (cont'd). Performance Comparison: Optimal Policy vs NI vs NI -1

	Parameters				Optimal	NI -1					NI				
	λ	μ_1	μ_2	μ_3	Cost	Gap%	Avg Gap%	m_1^*	m_2^*	m_3^*	Gap%	Avg Gap%	m_1^*	m_2^*	m_3^*
Case 4	1	15	10	10	5.53	5.64	3.5	0	0	1	18.82	9.0	0	0	1
	2	15	10	10	8.13	0.37		0	0	2	8.87		0	0	2
	3	15	10	10	10.65	3.18		0	0	3	5.42		0	0	3
	4	15	10	10	13.66	2.78		1	1	4	5.54		0	1	4
	5	15	10	10	16.99	5.72		3	2	5	6.29		0	1	6
Case 5	1	10	15	10	5.49	5.51	4.0	0	0	1	18.79	7.9	0	0	1
	2	10	15	10	8.04	0.25		0	0	2	8.22		0	0	2
	3	10	15	10	10.49	2.87		1	0	3	3.58		0	0	3
	4	10	15	10	13.30	3.90		2	1	4	3.48		0	0	4
	5	10	15	10	16.53	7.24		3	2	6	5.49		1	1	5
Case 6	1	10	10	15	5.47	2.84	3.1	0	0	1	16.49	6.7	0	0	1
	2	10	10	15	7.83	0.25		0	0	2	7.56		0	0	2
	3	10	10	15	10.25	2.38		1	0	3	3.21		0	0	3
	4	10	10	15	12.93	2.86		2	1	4	3.00		0	1	3
	5	10	10	15	15.99	7.30		3	2	6	3.44		1	2	4

CHAPTER 5

A MARKOVIAN ANALYSIS OF MAKE-TO-STOCK PRODUCTION LINES WITH LIMITED SUPPLY

This chapter considers production lines with parallel-machine stations and two-phase Coxian processing times. The setting in this study is designed specifically for scrutiny of the replenishment of raw materials and finished goods inventories with intermediate buffers in-between stations. Each buffer has a capacity limit. Raw material supply and demand for finished goods are generated according to independent stationary Poisson processes. Two-phase Coxian (Cox-2) processing times can be utilized to model failure-prone machines with exponential service times, times to failure, and repair times (Altiok & Stidham, 1983). The second phase of Coxian-2 can also be considered as a rework operation visited with a certain probability. We model the production system as a continuous-time Markov chain and propose recursive algorithms to generate the transition rate matrix. Although the general recursive form is limited to 3-station 4-buffer lines, routines for calculating the number of states and generating the states work for any M-station (M+1)-buffer systems. The developed model allows obtaining steady-state distribution and performance metrics such as throughput, the average number of items in the buffers, and average system cost consisting of production, holding, and shortage costs. Furthermore, we enrich our study with numerical experiments and analyze the impacts of buffer capacities, the number of parallel machines, and the processing rates of machines on the system performance. Moreover, the exact analysis provided in this study can be used as a decomposition block for the performance analysis of longer lines.

This study focuses on serial production systems where each station consists of failure-prone parallel machines modeled using two-phase Coxian distribution. After completing the first operation, the second stage/phase of Cox-2 is visited at each station with a certain probability. This structure corresponds to the case where processing times, times to failure and repair times are all independent exponential random variables. The second stage of the operations can also be considered rework operations frequently encountered in real-life practices. Furthermore, Cox-2, a phase type-family member, can be used to approximate general processing time distributions (Altiok, 1985; van der Heijden, 1988).

To the best of our knowledge, neither 2-station 3-buffer nor 3-station 4-buffer lines with failure-prone parallel machines have been studied. Exact analyses of multi-server stations in the literature are limited to 2-station lines where raw material replenishment/finished goods buffers are not considered. It is assumed that the raw material is always available or finished goods are immediately sent to the customers. However, our study distinguishes itself from the literature by contributing a novel model where supply and demand are presented as Poisson processes, and inventory levels in raw material replenishment and end product buffers are tracked.

We aim to model more implementable and realistic systems incurring holding costs for the finished goods and raw materials, variability in inter-demand and supply lead times, and shortages in raw material and finished goods. These are the factors that affect the variability of production lines (Romero-Silva et al., 2019). Although there is a tradeoff between modeling a realistic system and its computational load, our methodology is tractable in computation while being more realistic than the literature. Moreover, in our modeling approach, it is possible to eliminate the effects of buffers related to supply and demand by selecting large values for the rates of supply and demand processes.

The most relevant works in the literature are (Altiok & Stidham, 1983) and (Diamantidis & Papadopoulos, 2009). However, these studies do not consider replenishment and finished goods buffers, and our work distinguishes them by keeping track of the supply and demand processes. Besides, our study extends the performance analysis of (Altiok & Stidham, 1983) by considering parallel machines. On the other hand, our recursive approach derives exact solutions of both 2-station 3-buffer and 3-station 4-buffer lines extending the 2-station 1-buffer study of (Diamantidis & Papadopoulos, 2009).

The remainder of the chapter is organized as follows: Chapter 5.1 describes the considered production system, the state vector and the general structure of the matrix generation algorithm composed of three sub-routines. Chapter 5.2 presents the recursive method to calculate the number of states, which is the first routine of the general algorithm structure. After that, Chapter 5.3 presents the routine for state generation. The final routine, the matrix generation method, is described in Chapter 5.4. Lastly, Chapter 5.5 provides numerical studies.

5.1. Description of the System and the State Definition

A production system with M stations arranged in series where each station has parallel machines with two-phase Coxian processing times and (M+1) buffers with their capacities is considered in this study. Buffer-1 is defined for the raw materials, Buffer-(M+1) is for the finished goods, and the remaining ones are for the work-in-process inventories (see Figure 5.1). Cox-2 random variables have two independent Exponential phases, and phase-2 is visited with a prespecified probability. Hence, each random variable defined for Exponential phases corresponds to the length of a particular production stage of the process at each machine. The supply of the raw materials and the demand for finished goods are generated according to independent Poisson processes. The notation describing the system parameters is provided in Table 5.1.

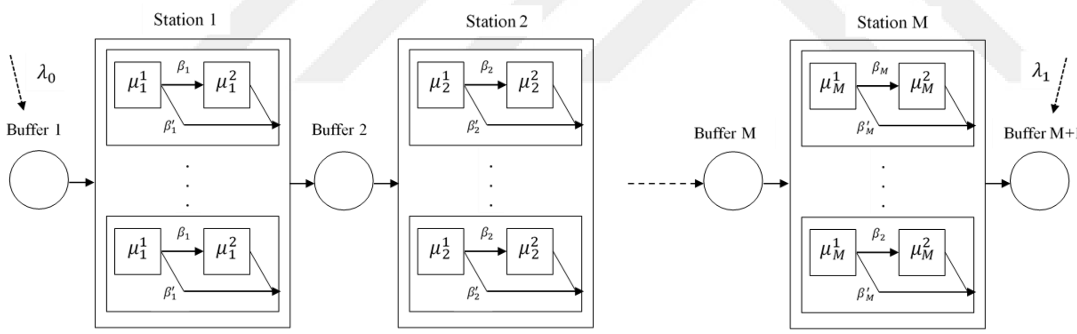

Figure 5.1. A production line with M-stations (M+1)-buffers and multiple Coxian-2 servers at stations

During production, machines are not intentionally idle, i.e., items are produced whenever it is possible. However, the production is interrupted if a machine is starved or blocked: after completion of its production, a station is starved if its upstream buffer is empty, and it is blocked if there is no room left in its downstream buffer. Within this context, if there is at least one room in buffer-j, then machines at station-j cannot be idle, and if there exists an idle machine at the station-(j+1), then blocking cannot be occurred at station-j, j=1, …, M.

For any item departing from station-j, j=1,.., M-1, one of the following could happen: (i) if there is an idle server at the station-(j+1) then the item is immediately transferred to the next station, (ii) if there is no idle server at the station-(j+1), but buffer (j+1) is not full, then the item is held in the buffer until a server becomes available at station-(j+1); (iii) otherwise, the item has to wait in the server, and the server remains blocked until an item is released from buffer-(j+1). Thus, in our case, all the stations can be idle (starved) or blocked, which is not a typical supposition in the literature.

Table 5.1. System parameters

$SI = \{1, \ldots, M\}$	: set of station indices
$BI = \{1, .., M+1\}$	: set of buffer indices
s_j	: number of available parallel servers at station j, $j \in SI$
m_j	: capacity of buffer j, $j \in BI$
μ_j^i	: production rate at phase-i of station j, $i \in \{1,2\}, j \in SI$
β_j	: visiting probability of phase-2 (from phase-1) at station j, $j \in SI$
β_j'	:$(1-\beta_j), j \in SI$
λ_0	: raw material replenishment rate
λ_1	: demand rate

The events of the system can be grouped into three: raw material replenishment, demand arrival to the finished goods, and production-related events. Due to the phase-type nature of production times, production-related events are also three types: an item departing from phase-1 of the server either (*i*) visits phase-2 or (*ii*) release the server without visiting phase-2 if there is room in the downstream buffer; (*iii*) an item departing from phase-2 releases the server providing that there is a room in the downstream. The events defined in (*ii*) and (*iii*) lead to production completion at the server. However, (*i*) is a phase completion event that changes the status of the production server, but the total number of busy servers and the number of items in the buffers remain the same within the occurrence of this event.

State vector should keep track of the status of the servers/machines at each station and the number of items in each buffer. Hence, the state vector of an M-station system can be described as $V_M = [n_1, k_1^1, k_1^2, b_1, n_2, k_2^1, k_2^2, b_2, \ldots, n_M, k_M^1, k_M^2, b_M, n_{M+1}]$ with the following state variables:

n_j = the number of items in buffer j, $j \in BI$

k_j^i = the number of busy servers being processed at phase-i of station j, $i = 1,2$, $j \in SI$

b_j = the number of blocked servers at station j, $j \in SI$

In total, there are $(1 + 4M)$ state variables for an M-station system, and the recurrent state space is defined by (10) – (17) with no intentional idleness in production.

$$n_1 \in \{0,1, \dots, m_1\} \tag{10}$$

$$k_j^1 \in \{0,1, \dots, s_j\}, j \in SI \tag{11}$$

$$k_j^2 \in \{0,1, \dots, s_j - k_j^1\}, \qquad j \in SI \tag{12}$$

$$b_j \in \begin{cases} \{s_j - k_j^1 - k_j^2\}, & if\ n_j > 0 \\ \{0, \dots, s_j - k_j^1 - k_j^2\} & if\ n_j = 0 \end{cases}, \qquad j \in SI \tag{13}$$

$$n_j \in \begin{cases} \{m_j\}, & if\ b_j > 0 \\ \{0, \dots, m_j\} & if\ b_j = 0 \end{cases}, \qquad j = 2, \dots, M+1 \tag{14}$$

$$n_j \geq 0, \qquad j \in BI \tag{15}$$

$$k_j^i, b_j \geq 0, \qquad j \in SI, i = 1,2 \tag{16}$$

$$k_j^1 + k_j^2 + b_j \leq s_j, \qquad j \in SI \tag{17}$$

One of the main contributions of this study is presenting a novel methodology to generate the transition matrix for Cox-2 distributed parallel-machine production lines having Poisson supply and demand processes. Figure 5.2 illustrates the general structure of our method consisting of three algorithms. Algorithm 1 calculates NS_j, which is the number of recurrent states of a j-station line. Then, Algorithm 2 uses NS_M to generate the system's state vector, and Algorithm 3 generates the transition rate matrix for a given state vector. Algorithms 1 and 2 work for any M, whereas Algorithm 3 is developed up to M=3 due to the computation limitations.

Algorithm 1 *Number of states calculation*
Input: $M, s_j, j \in SI, m_j, j \in BI$
Output: NS_M, the number of states of M-station

Algorithm 2 *State generation*
Input: $NS_j, j \in SI$
Output: V_M, the ordered state vector of M-station

Algorithm 3 *Transition rate matrix generation*
Input: V_M
Output: T^M, the state transition rate matrix

Figure 5.2. The routines of the algorithm with their hierarchy

5.2. Number of States Calculation Method

In this chapter, we propose a recursive algorithm to calculate the number of states for any M-station (M+1)-buffer system where station-$j, j \in SI$, has s_j parallel servers and buffer-$j, j \in BI$, has a capacity of m_j. In Figure 5.3, the procedure of NS_M with inputs $M, s_j, j \in SI, m_j, j \in BI$ is given as Algorithm 1. Basis step provides a base case solution for a 1-station 2-buffer system. Solution for any M >1 can then be obtained.

Algorithm 1

Basis Step	**Recursion Step**
station index j=1 $X_1 = m_1(s_1+1) + \sum_{i=0}^{s_1}(i+1)$ $Y_1 = m_1 \frac{s_1(s_1+1)}{2} + \sum_{i=0}^{s_1} \frac{i(i+1)}{2}$ $NS_1 = (m_2+1)X_1 + Y_1$ return NS_1	for j=2 to M do $X_j = (NS_{j-1} - X_{j-1})(s_j+1) +$ $X_{j-1} \sum_{i=0}^{s_j}(i+1)$ $Y_j = (NS_{j-1} - X_{j-1}) \frac{s_{j-1}(1+s_{j-1})}{2} +$ $X_{j-1} \sum_{i=0}^{s_j} \frac{i(i+1)}{2}$ $NS_j = (m_{j+1}+1)X_j + Y_j$ return NS_M

Figure 5.3. Number of states calculation algorithm with complexity $O(M * max(s_1, \dots, s_M))$

NS_j is defined as a linear function of X_j and Y_j that are partitions of the state space. X_j calculates the number of non-blocked states, whereas Y_j counts the states when there is a partial or full blocking. X_j has the multiplier $(m_{j+1}+1)$ because the state variable n_j has a range of $\{0,1,\dots,m_j\}$ for the non-blocked states as it is defined in Equation (14).

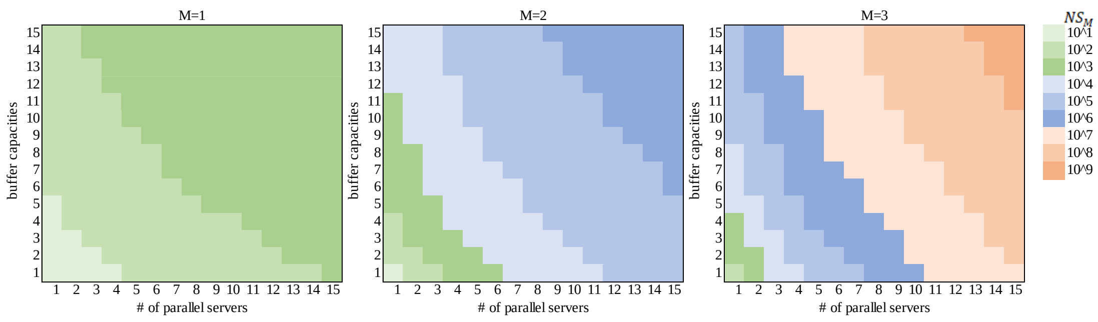

Figure 5.4. The number of states growing for M=1,2,3

Figure 5.4 shows how the number of states changes with the buffer capacities and the number of parallel servers for M=1,2,3. The figure assumes an equal number of servers at each station and identical buffer capacities for a given M. In Figure 5.4, the number of states reaches up to 10^3 (M=1), 10^6 (M=2) and 10^9 (M=3), which reveals that the dimensionality problem is the main limitation of the studies aiming at exact analysis. As M increases, the effect of parallel servers on NS_M seems to be prominent compared to buffer sizes.

5.3. State Generation Method

The next step of the study is to generate and order the states. Ordering of the states could determine the structure of the transition matrix. Thus, the transition structure explained in Chapter 5.4 depends on the order of states that we propose in this chapter. State vector $V_j, j \in SI$ is a dynamic array that keeps and orders the states. The state generation algorithm with inputs $M, s_j, NS_j, j \in SI, m_j, j \in BI$ is given as Algorithm 2 in Figure 5.5.

States are generated based on a recursion where the basis step returns $V_1 = [n_1, k_1^1, k_1^2, b_1, n_2]$ with all recurrent states. After that, in the recursion step, we make insertions for each additional state variable that is required for j = 2, and V_1 automatically grows to obtain $V_2 = [n_1, k_1^1, k_1^2, b_1, n_2, k_2^1, k_2^2, b_2, n_3]$.

We increase the number of dimensions of the vector V_1 for the state variables k_2^1, k_2^2, b_2, n_3 for the recursion step. For every state defined through V_1, only the recurrent set of k_2^1, k_2^2, b_2, n_3 are generated. That is while generating the states of V_2 every state in V_1 repeats itself as many times as the number of feasible values of k_2^1, k_2^2, b_2, n_3. Then, the procedure continues until the states are generated for any $M \geq 1$. Each time a new state is generated, a row number (index) is assigned to the state, starting from 1 up to NS_M and the transition matrix is created by relying on the labels of the states.

The complexity of Algorithm 2 is bounded by the number of states calculated in Algorithm 1, which is $O(NS_M)$, as Algorithm 2 is defined in such a way that there is no redundant state generation takes place during the execution.

Algorithm 2	
Basis Step	**Recursion Step**

station index j=1
$x = 0;$
for $v_0^1 = 0$ to m_1
  for $v_1^1 = 0$ to s_1
    for $v_2^1 = 0$ to $s_1 - v_1^1$
      if $v_0^1 > 0$ then $A = s_1 - v_1^1 - v_2^1$
      else $A = 0$
      for $v_3^1 = A$ to $s_1 - v_1^1 - v_2^1$
        if $v_3^1 > 0$ then $B = m_2$
        else $B = 0$
        for $v_4^1 = B$ to m_2
          $x = x + 1;$
          $V_1(x) = [v_0^1, v_1^1, v_2^1, v_3^1, v_4^1];$
return V_1
	for j=2 to M
	  $V_{j-1} = \tilde{V}_{j-1}; x = 0;$
	  for $i = 1$ to NS_{j-1}
	    for $v_1^j = 0$ to s_j
	      for $v_2^j = 0$ to $s_j - v_1^j$
	        if $v_4^{j-1} > 0$ then $A = s_j - v_1^j - v_2^j$
	        else $A = 0$
	        for $v_3^j = A$ to $s_j - v_1^j - v_2^j$
	          if $v_3^j > 0$ then $B = m_{j+1}$
	          else $B = 0$
	          for $v_4^j = B$ to m_{j+1}
	            $x = x + 1;$
	            $V_j(x) = [\tilde{V}_{j-1}(i), v_1^j, v_2^j, v_3^j, v_4^j];$
	return V_M

Figure 5.5. State generation algorithm with complexity $O(NS_M)$

To better understand the routine, consider an example for a 3-station line with the following number of servers and buffer capacities: $\vec{s} = [1,1,1]$ and $\vec{m} = [1,1,2,1]$. According to Algorithm 1, the number of states is calculated as NS_1= 12 for j=1, NS_2= 99 for j=2, and NS_3= 553 for j=M=3. After that, states are generated and ordered by Algorithm 2. Figure 5.6 illustrates the structure of the state generation algorithm with an example. In the basis step, states are generated and ordered for j=1. Then, recursion starts with the expansion of the state vector V_1 to create V_2 and then V_3.

In Figure 5.6, the states shown in bold are the ones generated for j=1 and then repeatedly used in V_2 and V_3. Similarly, italic elements are first generated for j=2 and then repeatedly used in V_3. The repetition pattern of states depends on the last element of a state vector. Let $C1_j^M$ and $C2_j^M$ be repetition coefficients of j^{th} station for an M station system where $C1_j^M$ counts the number of repetitions of each state with "$n_{j+1} = 0$" and $C2_j^M$ counts the number of repetitions of each state with "$n_{j+1} > 0$". These coefficients are recursively calculated over stations (see Appendix 4) and carried in transition rate matrix generation explained in the next chapter. In the example stated in Figure 5.6, they are calculated as $C1_1^2 = 7, C2_1^2 = 10$ for M=2, and $C1_1^2 = 5$, $C2_1^2 = 7$ for M=3.

Label	V_1		Label	V_2		Label	V_3
1	[**0, 0, 0, 0, 0**]		1	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*]		1	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 0, 0, 0, 0]
2	[**0, 0, 0, 0, 1**]		2	[**0, 0, 0, 0, 0**, *0, 0, 0, 1*]		2	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 0, 0, 0, 1]
3	[**0, 0, 0, 1, 1**]		3	[**0, 0, 0, 0, 0**, *0, 0, 0, 2*]		3	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 0, 0, 1, 1]
4	[**0, 0, 1, 0, 0**]		4	[**0, 0, 0, 0, 0**, *0, 0, 1, 2*]		4	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 0, 1, 0, 0]
5	[**0, 0, 1, 0, 1**]		5	[**0, 0, 0, 0, 0**, *0, 1, 0, 0*]		5	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 0, 1, 0, 1]
6	[**0, 1, 0, 0, 0**]		6	[**0, 0, 0, 0, 0**, *0, 1, 0, 1*]	$C2_2^3$	6	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 1, 0, 0, 0]
7	[**0, 1, 0, 0, 1**]		7	[**0, 0, 0, 0, 0**, *0, 1, 0, 2*]		7	[**0, 0, 0, 0, 0**, *0, 0, 0, 0*, 1, 0, 0, 1]
8	[**1, 0, 0, 1, 1**]		8	[**0, 0, 0, 0, 0**, *1, 0, 0, 0*]		8	[**0, 0, 0, 0, 0**, *0, 0, 0, 1*, 0, 0, 1, 1]
9	[**1, 0, 1, 0, 0**]	$C2_1^2$	9	[**0, 0, 0, 0, 0**, *1, 0, 0, 1*]		9	[**0, 0, 0, 0, 0**, *0, 0, 0, 1*, 0, 1, 0, 0]
10	[**1, 0, 1, 0, 1**]		10	[**0, 0, 0, 0, 0**, *1, 0, 0, 2*]		10	[**0, 0, 0, 0, 0**, *0, 0, 0, 1*, 0, 1, 0, 1]
11	[**1, 1, 0, 0, 0**]		11	[**0, 0, 0, 0, 1**, *0, 0, 1, 2*]	$C1_2^3$	11	[**0, 0, 0, 0, 0**, *0, 0, 0, 1*, 1, 0, 0, 0]
12	[**1, 1, 0, 0, 1**]		12	[**0, 0, 0, 0, 1**, *0, 1, 0, 0*]		12	[**0, 0, 0, 0, 0**, *0, 0, 0, 1*, 1, 0, 0, 1]
			13	[**0, 0, 0, 0, 1**, *0, 1, 0, 1*]		.	.
			14	[**0, 0, 0, 0, 1**, *0, 1, 0, 2*]		.	.
			15	[**0, 0, 0, 0, 1**, *1, 0, 0, 0*]		.	.
		$C1_1^2$	16	[**0, 0, 0, 0, 1**, *1, 0, 0, 1*]		.	.
			17	[**0, 0, 0, 0, 1**, *1, 0, 0, 2*]		.	.
			.	.		.	.
			99	[**1, 1, 0, 0, 1**, *1, 0, 0, 2*]		553	[**1, 1, 0, 0, 1**, *1, 0, 0, 2*, 1, 0, 0, 1]

Figure 5.6. The state generation algorithm: An explanatory example

The first two routines of the main algorithm, which are explained previously, apply to any M-station (M+1) buffer lines.

5.4. Transition Rate Matrix Generation Method

The final routine of the general algorithm structure defined in Figure 5.2 is the matrix generation method. This chapter proposes a novel matrix generation methodology for failure-prone multi-server production lines with finite buffers. The procedure is based on recursions over stations. Five different event types drive transitions among the states. Three of the transitions belong to production-related events, and the other two are defined for raw material replenishment (supply) and demand arrival events. For $j = 1, .., M$, events are defined as follows:

P1(j): An item departing from phase-1 of the server at station j visits phase-2 with rate $\beta_j k_j^1 \mu_j^1$.

P2(j): An item departing from phase-1 of the server at station j leaves the server without visiting phase-2 with rate $\beta_j' k_j^1 \mu_j^1$.

P3(j): An item departing from phase-2 of the server at station j leaves the server with rate $k_j^2 \mu_j^2$.

S: Raw materials are replenished to buffer-1 with a rate λ_0.

D: Demands arrive at finished goods inventory with a rate λ_1.

For an M-station system, let T^M be the transition rate matrix of size $NS_M \times NS_M$ where NS_M is defined as the number of recurrent states. Then, T^M can be written as the summation of the matrices corresponding to the transitions described above:

$$T^M = T_S^M + T_D^M + \sum_{j=1}^{M} \left(T_{P1(j)}^M + T_{P2(j)}^M + T_{P3(j)}^M \right) \tag{18}$$

The matrix generation method is developed up to 3-station 4-buffer lines. While generating the matrix, states are partitioned into classes considering the status of state variables: they are classified as blocked ($b_j > 0$) and non-blocked ($b_j = 0$). The non-blocked class consists of states where all the servers at stations are either busy or idle. On the other hand, buffers are classified as empty ($n_j = 0$) and non-empty ($n_j > 0$).

Table 5.2 represents the partition of states for all event types. While j represents the station index, M represents the total number of stations in the line. For the transitions at station 1, there are two partitions, shown as (1) and (2), based on the status of the upstream buffer. However, for the transitions at stations j=2 and j=3, partitions are based on the status of first j buffers and (j-1) stations.

Table 5.2. Partition of states

$M = 1,2,3, j = 1$		$M = 2,3, j = 2$		$M = 3, j = 3$	
(1)	$n_1 = 0$	(3)	$n_1 = 0, n_2 = 0$	(8)	$n_1 = 0, n_2 = 0, n_3 = 0$
(2)	$n_1 > 0$	(4)	$n_1 = 0, n_2 > 0$	(9)	$n_1 = 0, n_2 = 0, n_3 > 0$
		(5)	$n_1 > 0, n_2 = 0$	(10)	$n_1 = 0, n_2 > 0, n_3 = 0$
		(6)	$n_1 > 0, b_1 > 0, n_2 > 0$	(11)	$n_1 = 0, n_2 > 0, b_2 > 0, n_3 > 0$
		(7)	$n_1 > 0, b_1 = 0, n_2 > 0$	(12)	$n_1 = 0, n_2 > 0, b_2 = 0, n_3 > 0$
				(13)	$n_1 > 0, n_2 = 0, n_3 = 0$
				(14)	$n_1 > 0, n_2 = 0, n_3 > 0$
				(15)	$n_1 > 0, b_1 > 0, n_2 > 0, n_3 = 0$
				(16)	$n_1 > 0, b_1 > 0, n_2 > 0, b_2 > 0, n_3 > 0$
				(17)	$n_1 > 0, b_1 > 0, n_2 > 0, b_2 = 0, n_3 > 0$
				(18)	$n_1 > 0, b_1 = 0, n_2 > 0, n_3 = 0$
				(19)	$n_1 > 0, b_1 = 0, n_2 > 0, b_2 > 0, n_3 > 0$
				(20)	$n_1 > 0, b_1 = 0, n_2 > 0, b_2 = 0, n_3 > 0$

Labels (ordering numbers) of the actual states are used while generating the matrix. Let T_{Event}^{M} be a transition rate matrix of an M station line for a specific *Event* $\in \{P1, P2, P3, S, D\}$. Then the property " $T_{Event}^{M}[F][T] = rate$ " represents the transition from a particular state labeled as *F* to another state labeled as *T* by an *Event* with a certain *rate*.

For each partition defined in Table 5.2, the calculation of state labels F and T has different structures. Thus, several loop blocks are identified for all event types. There are 22 loop blocks used in a nested manner (see Appendix 5). Let L_i^k be the i^{th} outer loop block with type k, $i = 1,2,3, k = 0, \ldots, 21$. The i^{th} block uses the parameters of the station at the i^{th} position. Each block has its structure to determine the state labels. The main idea is to use the blocks recursively for the sake of effective computation.

A particular sequence of loop blocks for each event type defines a specific algorithm. Let $A_{Event}^{i}(I_j^M)$ symbolize the algorithm-*i* that corresponds to an *Event* $\in \{P1, P2, P3, S, D\}$ occurred at station-j of an M-station line with an input vector *I*. Vector *I* carries the station related parameters and the repetition coefficients (see the example given in Chapter 5.3).

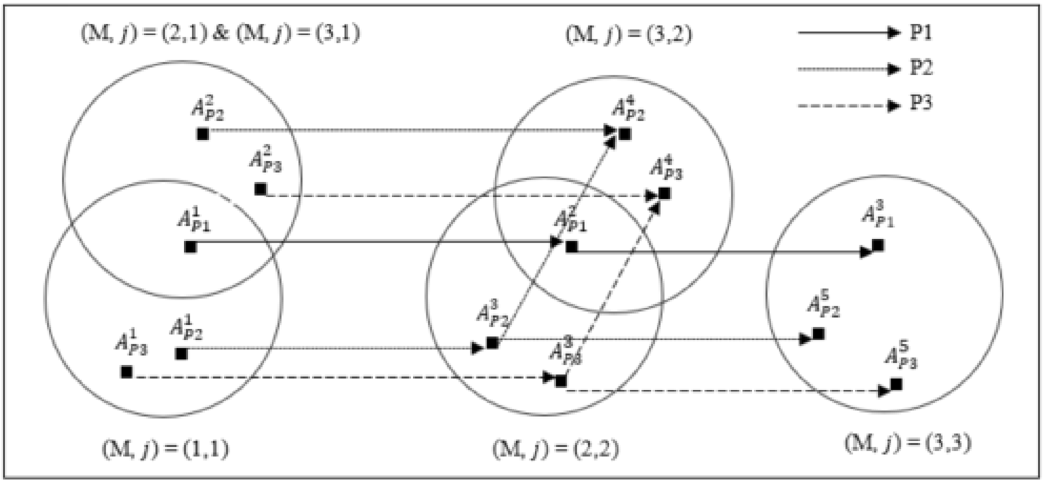

Figure 5.7. Mapping diagram of the production-related algorithms

We first focus on production-related events. Figure 5.7 provides a set representation of the algorithms where the mappings are defined for P1, P2 and P3. For every feasible (M, j) pair where M=1,2,3 and j=1,.., M, a set is defined to cover the algorithms. An algorithm in a set labeled by (M, j) is used for the transitions at the j^{th} station of an M station line. A specific algorithm can also be used for different (M, j) pairs with other input vectors I_j^M. For instance, transitions at station-1 corresponding to the event P1 can be generated by the algorithm A_{P1}^{1} for M=1,2,3 as shown in Figure 5.7.

For each production-related event, first, the base algorithms A_{P1}^{1}, A_{P2}^{1} and A_{P3}^{1} are defined for M=1. Then, the rest of the algorithms are developed by expanding the structure of the base algorithms. A mapping from algorithms A to B determines the loop structure of A is a part of B. There are three types of mapping: solid for P1, dotted for P2 and dashed for P3. For instance, the mapping expressed by solid arrows represents that A_{P1}^{1} is defined first, then A_{P1}^{2} is developed by expanding A_{P1}^{1}, after that A_{P1}^{3} is developed by expanding A_{P1}^{2}. However, the mapping diagrams of P2 and P3 are more complicated than P1 because P1 represents a phase completion rather than a production completion. Considering the partition of states defined in Table 5.2, the structure of the algorithms of event P1 that are represented as loop blocks are given in Table 5.3. The symbol " ↳ " represents the nested loop blocks.

Table 5.3. Algorithm structure of event P1

	$A_{P1}^{1}(I_{j=1}^{M=1,2,3})$		$A_{P1}^{2}(I_{j=2}^{M=2,3})$		$A_{P1}^{3}(I_{j=3}^{M=3})$		
(1)	L_1^{19}	(3)	L_1^1 ↳ L_2^{19}	(8)	L_1^1 ↳ L_2^1 ↳ L_3^{19}	(14)	L_1^0 ↳ L_1^3 ↳ L_2^2 ↳ L_3^{20}
(2)	L_1^0 ↳ L_1^{20}	(4)	L_1^2 ↳ L_2^{20}	(9)	L_1^1 ↳ L_2^2 ↳ L_3^{20}	(15)	L_1^0 ↳ L_1^4 ↳ L_2^3 ↳ L_3^{19}
		(5)	L_1^0 ↳ L_1^3 ↳ L_2^{19}	(10)	L_1^2 ↳ L_2^3 ↳ L_3^{19}	(16)	L_1^0 ↳ L_1^4 ↳ L_2^4 ↳ L_3^{20}
		(6)	L_1^0 ↳ L_1^4 ↳ L_2^{20}	(11)	L_1^2 ↳ L_2^4 ↳ L_3^{20}	(17)	L_1^0 ↳ L_1^4 ↳ L_2^5 ↳ L_3^{20}
		(7)	L_1^0 ↳ L_1^5 ↳ L_2^{20}	(12)	L_1^2 ↳ L_2^5 ↳ L_3^{20}	(18)	L_1^0 ↳ L_1^5 ↳ L_2^3 ↳ L_3^{19}
				(13)	L_1^0 ↳ L_1^3 ↳ L_2^1 ↳ L_3^{19}	(19)	L_1^0 ↳ L_1^5 ↳ L_2^4 ↳ L_3^{20}
						(20)	L_1^0 ↳ L_1^5 ↳ L_2^5 ↳ L_3^{20}

Note that every loop block defined in $A_{P1}^{i}, i = 1,2$ are used in A_{P1}^{i+1} with additional blocks. Figure 5.8 represents the base algorithm A_{P1}^{1}. Equation-(j) that is used in kth loop block and recall the parameters of the station in position-i, i=1,2,3, is denoted as $eq_{i(j)}^{k}$. The detailed equations that are used in base algorithms are in Appendix 6.

$$A_{P1}^{1}\left(I_{j=1}^{M=1} = C1_{j=1}^{M=1}, C2_{j=1}^{M=1}, s_j, m_j, m_{j+1}\right)$$

(1) L_1^{19}

$$T_{P1(1)}^{1}\left[eq_{1(1)}^{19}\right]\left[eq_{1(2)}^{19}\right] = k_1\beta_1\mu_1^1$$

(2) L_1^0 ↳ L_1^{20}

$$T_{P1(1)}^{1}\left[eq_{1(1)}^{0} + eq_{1(1)}^{20}\right]\left[eq_{1(1)}^{0} + eq_{1(2)}^{20}\right] = k_1\beta_1\mu_1^1$$

Figure 5.8. The base algorithm of event P1: A_{P1}^{1}

P2 and P3 define production completion events for the items departing directly from phase-1 and phase-2 of a Coxian server, respectively. Since both events result in releasing a server, they have the same mapping structure in Figure 5.7. Moreover, we observe that if the event P2 triggers a transition to a specific state with a rate $\beta_j' k_j^1 \mu_j^1$ at a station-j, then the event P3 would certainly trigger a transition to that specific state with a rate $k_j^2 \mu_j^2$. The base algorithms A_{P2}^1 and A_{P3}^1 are defined in Figure 5.9, where both algorithms have precisely the same loop blocks in each partition. The complete algorithm structures of P2 and P3 are given in Appendix 7.

$A_{P2}^1\left(I_{j=1}^{M=1} = s_1, m_1, m_2\right)$ and $A_{P3}^1\left(I_{j=1}^{M=1} = s_1, m_1, m_2\right)$

(1) $L_1^8 : T_{P2(1)}^1\left[eq_{1(1)}^8\right]\left[eq_{1(2)}^8\right] = k_1\beta_1'\mu_1^1$

$T_{P3(1)}^1\left[eq_{1(7)}^8\right]\left[eq_{1(8)}^8\right] = k_1\mu_1^2$

(2) $L_1^0 \Lsh L_1^{18}$

$T_{P2(1)}^1\left[eq_{1(1)}^0 + eq_{1(1)}^{18}\right]\left[eq_{1(1)}^0 + eq_{1(2)}^{18}\right] = k_1\beta_1'\mu_1^1$

$T_{P3(1)}^1\left[eq_{1(1)}^0 + eq_{1(3)}^{18}\right]\left[eq_{1(1)}^0 + eq_{1(2)}^{18}\right] = l_1\mu_1^2$

$L_1^0 \Lsh L_1^{11}$

If $p == 0$ then

$T_{P2(1)}^1\left[eq_{1(1)}^0 + eq_{1(1)}^{11}\right]\left[eq_{1(3)}^{11}\right] = k_1\beta_1'\mu_1^1$

$T_{P3(1)}^1\left[eq_{1(1)}^0 + eq_{1(5)}^{11}\right]\left[eq_{1(3)}^{11}\right] = (s_1 - k_1 + 1)\mu_1^2$

Else

$T_{P2(1)}^1\left[eq_{1(1)}^0 + eq_{1(1)}^{11}\right]\left[eq_{1(2)}^0 + eq_{1(2)}^{11}\right] = k_1\beta_1'\mu_1^1$

$T_{P3(1)}^1\left[eq_{1(1)}^0 + eq_{1(5)}^{11}\right]\left[eq_{1(2)}^0 + eq_{1(2)}^{11}\right] = (s_1 - k_1 + 1)\mu_1^2$

$L_1^0 \Lsh L_1^{12}$

If $p == 0$ then

$T_{P2(1)}^1\left[eq_{1(1)}^0 + eq_{1(1)}^{12}\right]\left[eq_{1(3)}^{12}\right] = k_1\beta_1'\mu_1^1$

$T_{P3(1)}^1\left[eq_{1(1)}^0 + eq_{1(4)}^{12}\right]\left[eq_{1(3)}^{12}\right] = (s_1 - k_1 + 1)\mu_1^2$

Else

$T_{P2(1)}^1\left[eq_{1(1)}^0 + eq_{1(1)}^{12}\right]\left[eq_{1(2)}^0 + eq_{1(2)}^{12}\right] = k_1\beta_1'\mu_1^1$

$T_{P3(1)}^1\left[eq_{1(1)}^0 + eq_{1(4)}^{12}\right]\left[eq_{1(2)}^0 + eq_{1(2)}^{12}\right] = (s_1 - k_1 + 1)\mu_1^2$

Figure 5.9. The base algorithms of events P2 and P3: A_{P2}^1 and A_{P3}^1

In practice, it is reasonable to expect that raw materials may not be directly received when needed. Thus, we assume that raw materials are replenished to the first buffer at an Exponential rate λ_0 (event S) which corresponds to the Exponential lead time of the supply. The occurrence of event S in the line is twofold. If there is at least one idle server at station 1 (i), which implies that buffer 1 is empty, then one of the idle servers switches to the busy state. If there is at least one room at buffer-1 (ii), then the raw material is held in buffer-1 until a server at station-1 is available.

$A_S^M(I^M = C1_1^M, C2_1^M, s_M, m_M, m_{M+1})$

(1) $L_1^8 : T_S^1[eq_{1(6)}^8][eq_{1(1)}^8] = \lambda_0$

$L_1^{16}: T_S^1[eq_{1(1)}^{16}][eq_{1(2)}^{16}] = \lambda_0$

$L_1^{17} : T_S^1[eq_{1(1)}^{17}][eq_{1(2)}^{17}] = \lambda_0$

(2) $L_1^{21} : T_S^1[eq_{1(1)}^{21}][eq_{1(2)}^{21}] = \lambda_0$

Figure 5.10. The algorithm for supply event: A_S^M

$A_D^1(I^{M=1} = s_M, m_M, m_{M+1})$

(1) $L_1^2: T_D^1[eq_{1(3)}^2][eq_{1(4)}^2] = \lambda_1$

(2) $L_1^0 \Rsh L_1^5$

$T_D^1[eq_{1(1)}^0 + eq_{1(2)}^5][eq_{1(1)}^0 + eq_{1(3)}^5] = \lambda_1$

If $p == 0$ then

$L_1^0 \Rsh L_1^{18}$

$T_D^1[eq_{1(1)}^0 + eq_{1(2)}^{18}][eq_{1(4)}^{18}] = \lambda_1$

If $p > 0$ then

$L_1^0 \Rsh L_1^9$

$T_D^1[eq_{1(1)}^0 + eq_{1(2)}^9][eq_{1(2)}^0 + eq_{1(1)}^9] = \lambda_1$

$L_1^0 \Rsh L_1^{15}$

$T_D^1[eq_{1(1)}^0 + eq_{1(6)}^{15}][eq_{1(2)}^0 + eq_{1(7)}^{15}] = \lambda_1$

Figure 5.11. The base algorithm for demand arrival: A_D^1

The effects (i) and (ii) apply regardless of how long the production line is. For this reason, a base algorithm A_S^1 is defined, then it is recalled with the input vector $I^M = (C1_1^M, C2_1^M, s_M, m_M, m_{M+1})$ to obtain T_S^M for any M. Event S is the only part of

Algorithm 3 that is general to any number of stations, and the complete algorithm is given in Figure 5.10. In total, four different loop blocks are used for generating the matrix related to event S.

Lastly, event D is defined for demand arrivals to the finished goods buffer. Demands are generated according to a Poisson process with a rate of λ_1. The base algorithm A_D^1 is given in Figure 5.11. A_D^1 is expanded to develop A_D^2 and A_D^3 recursively (see Appendix 8).

The set of nested loop blocks used for matrix generation are represented in Appendix 5 with Big-O complexity. Depending on event type and the partition of states, loop blocks consist of different statements. As M increases, the number of loop blocks and statements and their combinations increase, hence complexity calculation becomes more tedious. Therefore, to provide a good complexity approximation in terms of s_j and m_j without drowning in details, the execution of the statements in loop blocks is assumed to be constant. Complexity can be calculated for 1-station 2-buffer systems with the algorithms given in Figures 5.8 to 5.11, and bounded by $max\ (m_1 s_1^2 + m_1 m_2 s_1, s_1^3 + m_2 s_1^2)$. For M=2 and M=3, corresponding loop blocks (in Table 5.3, Figure 5.10, Appendices 6-7) can be revisited, and complexities can be calculated by determining the term with the highest power in loop blocks. The runtime to generate matrix would be a function of $s_{1,} s_2, m_{1,} m_2, m_3$ for M=2, and a function of $s_{1,} s_2, s_3, m_{1,} m_2, m_3, m_4$ for M=3.

5.5. Numerical Study

The transition rate matrix is solved to obtain steady-state probabilities. Let P be the steady-state probability vector, then transition rate matrix of a continuous-time Markov chain (CTMC) model T satisfies the below balance equation where T^t is the transpose of T

$$T^t P = 0.$$

While computing the steady-state probabilities, a discrete-time equivalent of a CTMC solution of T is obtained using the uniformization technique (Lippman, 1975). For an M-station system, the uniform transition rate v is defined as $v = \lambda_0 + \lambda_1 + \sum_{j=1}^{M} s_j \sum_{i=1}^{2} \mu_j^i$, and the discrete-time equivalent problem is solved using eigenvalue decomposition. Our approach is general enough to be executed for the production lines up to three stations and four buffers with an arbitrary number of parallel machines and buffer capacities. The steady-state solution allows obtaining throughput, average WIPs, stock-out probabilities and average system cost. Furthermore, the effects of processing times, parallel machines and buffer capacities on these metrics can be identified. A MATLAB 2018b program is developed for all the routines to compute the number of states, generate the states and create the transition matrix. A series of computational experiments are carried out on a Core i7, 2.80 GHz, 16 GB RAM computer.

The proposed method is first tested with the existing results obtained by (Diamantidis & Papadopoulos, 2009). Raw material replenishment and finished goods buffers are not considered in their study. Thus it is assumed that (i) the first station is never starved, and (ii) the last station is never blocked. In verification studies, we set the capacities of replenishment and finished goods buffers to 1 and select large-enough values for λ_0 and λ_1 to guarantee the almost sure convergence of probabilities $P(n_1 > 0) = 1$ and $P(n_{M+1} < m_{M+1}) = 1$ corresponding to (i) and (ii), respectively. The throughput results of (Diamantidis & Papadopoulos, 2009) for identical parallel machines are aligned with ours with an average of 0.02% deviation.

Table 5.4. Numerical results: $\lambda_0 = 4, \lambda_1 = 3, s_1 = s_2 = 3, m_1 = m_3 = 1$

m_2	# of States	Throughput	Run Time (seconds)
10	2,604	1.857	14.6
20	4,564	1.922	48.4
30	6,524	1.947	107.3
40	8,484	1.959	208.5
50	10,444	1.965	350.7
60	12,404	1.968	560.3
70	14,364	1.969	826.2
80	16,324	1.970	1537.4
90	18,284	1.971	1663.0
100	20,244	1.971	2075.9

First, we focus on 2-station 3-buffer lines and examine how throughput changes with the capacity of the intermediate buffer. Table 5.4 shows the results for the settings considered in Table 6 of (Diamantidis & Papadopoulos, 2009) with additional parameters for the first and last buffers. We set the other parameters as $m_1 = m_3 = 1, \lambda_0 = 4, \lambda_1 = 3$, and present the throughput and the computation time (in seconds). We obtain reasonable run times in Table 5.4 despite our setting is general than the one shown in Table 6 of (Diamantidis & Papadopoulos, 2009). Throughput increases up to a certain level of m_2, after that, it becomes constant at 1.971. The level with no improvement in throughput represents the practical infinite for buffer-2. Hence the capacity of buffer-2 cease to be a bottleneck in the system.

For the numerical experiments presented below, in addition to the throughput and computation time (in seconds), we also provide $\overline{n_j}$: the long-run average WIP in buffer-j, and $P(S)$: the probability of stock-out. $P(S)$ can also be interpreted as the probability of losing an arriving customer due to the PASTA property.

Table 5.5. Numerical results: $\lambda_0 = 3, \lambda_1 = 2, s_1 = 2, m_1 = 3, m_2 = 6, m_3 = 4$

s_2	# of States	Throughput	$\overline{n_1}$	$\overline{n_2}$	$\overline{n_3}$	$P(S)$	Run Time (secs)
1	1,373	0.962	2.504	5.924	1.037	0.519	3.5
2	2,364	1.621	2.027	5.665	2.464	0.189	11.8
3	3,578	1.897	1.777	5.374	3.410	0.051	30.1
4	5,030	1.975	1.688	5.181	3.804	0.012	66.8
5	6,735	1.993	1.663	5.093	3.933	0.003	127.1
6	8,708	1.997	1.656	5.059	3.972	0.001	227.8
7	10,964	1.998	1.655	5.047	3.985	0.001	383.4
8	13,518	1.999	1.654	5.042	3.991	0.001	688.7
9	16,385	1.999	1.653	5.039	3.994	0.000	1193.4
10	19,580	1.999	1.653	5.037	3.996	0.000	1667.3

In Table 5.5, we assess the performance of a 2-station 3-buffer system where $s_1 = 2$ and s_2 varies from 1 to 10. Coxian parameters are set to $(\mu_1^1, \mu_1^2, \beta_1) = (2, 1.2, 0.05)$ for station 1 and $(\mu_2^1, \mu_2^2, \beta_2) = (2, 0.4, 0.2)$ for station 2. From raw materials to finished goods, buffer capacities are defined as $m_1 = 3, m_2 = 6$, and $m_3 = 4$. Raw materials replenishment and demand arrival rates are $\lambda_0 = 3$ and $\lambda_1 = 2$. Increasing s_2 implies higher production capacity at station-2, hence the long-run average WIPs in buffer-1 and buffer-2 decrease while WIP in buffer-3 increases. Moreover, the overall production capacity increases up to a level, and then throughput converges to 1.999 and $P(S)$ tends to zero, i.e., all incoming demands are satisfied.

Table 5.6 presents the numerical results of a 2-station 3-buffer system while the rate of the first stage at station-2, μ_2^1, is changing. Other Coxian parameters are given as $(\mu_1^1, \mu_1^2, \beta_1) = (2.5, 1, 0.06)$ and $(\mu_2^2, \beta_2) = (1.5, 0.4)$. Performance metrics respond to changes in μ_2^1 in the same direction as they do in Table 5.5.

Table 5.6. Numerical results: $\lambda_0 = 6, \lambda_1 = 3, s_1 = 2, s_2 = 1, m_1 = 4, m_2 = 7, m_3 = 3$ with # of states of 1,512

μ_2^1	Throughput	$\overline{n_1}$	$\overline{n_2}$	$\overline{n_3}$	$P(S)$	Run Time (secs)
1	0.788	3.855	6.991	0.335	0.737	4.9
3	1.587	3.660	6.894	0.935	0.471	5.2
5	1.901	3.556	6.781	1.294	0.366	4.5
7	2.039	3.501	6.699	1.488	0.320	4.6
9	2.112	3.468	6.646	1.600	0.296	4.6
11	2.154	3.448	6.610	1.670	0.282	4.5
13	2.182	3.435	6.586	1.718	0.273	4.7
15	2.201	3.425	6.567	1.752	0.266	4.7

Tables 5.7 and 5.8 present the results for 3-station 4-buffer systems. Table 5.7 assumes a single machine (server) at each station with parameters $(\mu_1^1, \mu_1^2, \beta_1) = (2.5, 1, 0.06)$, $(\mu_2^2, \beta_2) = (1.5, 0.4)$ with $\mu_2^1 \in \{1,2, \dots ,10\}$, and $(\mu_3^1, \mu_3^2, \beta_3) = (6, 2.5, 0.5)$. In this experiment, it is observed that the changes in μ_2^1 mostly affect the immediate downstream buffer. On the other side, the average number in the immediate upstream buffer, $\overline{n_2}$, decreases with μ_2^1 but with a slower rate when compared to the increase in $\overline{n_3}$. It is because of having a bottleneck at the upstream stage, which slows down the production line. For the raw materials and finished goods inventories, we observe similar monotone behaviors with $\overline{n_2}$ and $\overline{n_3}$, respectively, however, it is in much slower paces. In addition to the effects on average inventories, an increase in μ_2^1 first results in sharp increases in throughput (and thus sharp decreases in stock-out probability). However, after certain rate values, the system reaches saturation, and no more significant improvement in throughput is observed.

Table 5.8 examines the effect of the number of parallel servers at the stations for $(\mu_1^1, \mu_1^2, \beta_1) = (2, 0.7, 0.05)$, $(\mu_2^1, \mu_2^2, \beta_2) = (2.7, 0.9, 0.4)$ and $(\mu_3^1, \mu_3^2, \beta_3) = (5, 2.5, 0.5)$. Starting from station-1, the number of parallel servers at each station is changed from 1 to 5, while the number of servers at the remaining stations is kept at 1.

Since station-2 has the lowest production rate among all (see the Coxian parameters), the throughput of the system is maximized as s_2 increases.

Table 5.7. Numerical results: $\lambda_0 = 5, \lambda_1 = 2, s_j = 1, j = 1,2,3, m_1 = 3,$ $m_2 = 5, m_3 = 10, m_4 = 2$ with # of states of 10,406

μ_2^1	Throughput	$\overline{n_1}$	$\overline{n_2}$	$\overline{n_3}$	$\overline{n_4}$	$P(S)$	Run Time (secs)
1	0.788	2.824	4.799	0.158	0.536	0.606	152.8
2	1.280	2.677	4.220	1.266	1.022	0.360	155.5
3	1.530	2.585	3.641	3.690	1.304	0.235	158.2
4	1.613	2.550	3.356	5.500	1.404	0.192	158.0
5	1.642	2.537	3.231	6.440	1.439	0.177	158.5
6	1.655	2.532	3.167	6.947	1.453	0.171	155.7
7	1.662	2.529	3.129	7.251	1.460	0.168	159.4
8	1.666	2.527	3.104	7.449	1.464	0.166	160.1
9	1.669	2.526	3.087	7.587	1.466	0.165	158.6
10	1.672	2.526	3.075	7.688	1.467	0.165	158.1

Table 5.8. Numerical results: $\lambda_0 = 5, \lambda_1 = 2, m_1 = 3, m_2 = 2, m_3 = 5, m_4 = 2$

$[s_1, s_2, s_3]$	# of States	Throughput	$\overline{n_1}$	$\overline{n_2}$	$\overline{n_3}$	$\overline{n_4}$	$P(S)$	Run Time
[1,1,1]	3,412	1.078	2.737	1.373	0.687	0.824	0.461	20.4
[2,1,1]	6,301	1.204	2.693	1.904	1.142	0.990	0.380	61.9
[3,1,1]	10,108	1.214	2.682	1.985	1.181	1.011	0.370	155.9
[4,1,1]	14,930	1.214	2.678	1.996	1.179	1.011	0.370	335.7
[5,1,1]	20,864	1.214	2.676	1.997	1.178	1.011	0.370	660.3
[1,1,1]	3,412	1.078	2.737	1.373	0.687	0.824	0.461	20.4
[1,2,1]	6,114	1.456	2.593	0.710	2.263	1.205	0.275	63.2
[1,3,1]	9,564	1.476	2.538	0.364	2.407	1.229	0.264	144.0
[1,4,1]	13,825	1.482	2.517	0.211	2.450	1.240	0.259	313.3
[1,5,1]	18,960	1.492	2.510	0.155	2.572	1.252	0.254	579.6
[1,1,1]	3,412	1.078	2.737	1.373	0.687	0.824	0.461	20.4
[1,1,2]	6,194	1.082	2.736	1.364	0.161	0.845	0.459	57.6
[1,1,3]	9,788	1.083	2.737	1.364	0.072	0.849	0.458	152.9
[1,1,4]	14,265	1.085	2.737	1.364	0.037	0.852	0.457	318.9
[1,1,5]	19,696	1.086	2.737	1.364	0.021	0.856	0.456	611.6

Although the computation time of our algorithm increases with the number of stations, the number of servers and buffer capacities, we believe that the results are reasonable and acceptable for such exact analyses, which are required for mid and long-term planning purposes. Production engineers and decision-makers do not conduct such analyses for daily or weekly operational activities. Furthermore, with our computational capacity, the steady-state solution of the transition matrix is obtained up to 64,000 states for M=1, it is up to 32,000 states for M=2 and M=3. All the statistics provided in this study were obtained using a standard personal computer and can further be improved with more powerful workstations.

CHAPTER 6
CONCLUSIONS AND FUTURE RESEARCH

In this thesis, we focus on control problems and performance analysis of three-station make-to-stock tandem lines. From the production control perspective, the main contribution of the studies considered in this thesis is twofold. Firstly, a characterization of optimal production control policies for three-station tandem lines is presented in various settings. To the extent of our knowledge, the structure of the optimal policies has not been investigated for three-station tandem production systems. Veatch and Wein are the first to study two-station tandem make-to-stock production systems having Exponential processing times, demands occurring according to a Poisson process and backorders to minimize long-run average system cost (Veatch & Wein, 1994). Optimal control policies are obtained using dynamic programming, and the results are compared with the well-known control mechanisms. It is shown in (Veatch & Wein, 1994) that the optimal control policy is defined by switching curves which defines busy sets of machines.

Our considered optimal control study extends the work of (Veatch & Wein, 1994) to three stations in a make-to-stock environment. Our setting consists of lost sales cases of three-station systems with ample raw material supply, intermediate buffers, finished goods buffer and demand occurrences with a Poisson process to minimize long-run average system cost. The structure of the optimal production policies is characterized by the results of the value iteration algorithm. In the basic model presented in Chapter 3.1 of the thesis, the processing times of machines are Exponentially distributed. Numerical studies reveal the effect of different production and demand rates on optimal policies. Moreover, in a limiting case of the basic model, a switching curve-type structure is observed, according to the propositions stated and proven in Chapter 3.1. The basic model is observed and shown with the numerical results that T_1 represents the threshold of optimal control actions to form a switching curve.

From the optimal control framework, additionally, we integrate complex features of the production systems. We consider two-phase Coxian processing times that can be utilized to model failure-prone machines with exponential service times, times to failure, and repair times (Altiok & Stidham, 1983). A two-phase Coxian random variable has independent Exponential phases with a certain visiting probability from phase one to phase two. The second phase of Cox-2 can also be maintained as a rework operation occurring with a predefined probability. Three different design problems of Cox-2 processing times are presented as extended models in Chapter 3.2. The models are built assigning Cox-2 times to upstream, intermediate and downstream stations while remaining stations are Exponentially distributed. The optimal policy structure is observed to be dependent on the Coxian phases. Furthermore, the analyses conducted with various system parameters show that locating a Cox-2 distributed machine at the downstream stage produces the highest system cost. On the contrary, the minimum system cost is observed with a Cox-2 distributed machine at the upstream stage.

The second contribution of control problems involves an alternative mechanism called the *no intentional idleness* policy. The proposed policy presented in Chapter 4 relies on easing production control decisions and letting machines produce as much as possible. To the best of our knowledge, the performance analysis of the proposed approach has not been compared to the optimal control policies of three-station make-to-stock flow lines. Defining finite buffer capacities for our considered make-to-stock flow line model allows us to sustain production with blocking and starvation. The policy aims to obtain the buffer capacities that minimize the average system cost. The proposed approach is modeled using ARENA Simulation Software 2019, and optimal buffer capacities are obtained with exhaustive search. Then the results are compared with the optimal control policy. It is observed that the proposed policy performs near-optimal for extended models with Cox-2 processing times. The optimality gap is calculated as less than 3% in numerical experiments conducted with 396 instances. For the basic model with Exponential processing times, the performance of the proposed policy alternates depending on the demand rate and production rates of machines. The policy deteriorates in the cases with a lower demand rate due to the holding cost accumulation. Then, it is modified to improve its performance for the cases of the basic model with lower demand rates. Presented results identify the improvement in every case defined for the basic model except Case 1 (see Table 4.6). The best alternative (NI -1) worsens in case 1, which belongs to a case with a lower production rate at the upstream station than the existing proposed policy.

Further to the results of the basic model, the maximum throughput with buffer capacities that minimize the system cost is observed in case 6, where the faster machine is assigned to station-3. Case 5 (with a faster machine in the middle) provides a slightly lower throughput. This result contradicts the findings of (Buzacott & Shanthikumar, 1993), stating that the throughput of a three-station flow line is maximized if the fastest station is located in the middle stage and the slowest stations are located in the first and third stages. However, (Buzacott & Shanthikumar, 1993) assumes no finished goods buffer and no cost-related objective function. Our study consists of a make-to-stock production setting with a well-defined cost objective.

In addition to the optimal and alternative control mechanisms, we present an exact long-run analysis up to 3-station 4-buffer production lines in Chapter 5. The setting consists of parallel-machine stations with Coxian-2 processing times. Raw materials, intermediate and finished goods buffers of finite capacities are considered. The production line is modeled as a continuous-time Markov chain, and a novel recursive method is developed to generate the transition rate matrix. The method developed as a MATLAB program consists of three routines calculating the number of states, generating the states and then the transition matrix. Steady-state distribution is obtained via Eigenvalue decomposition. Although the last routine is limited to 3-station 4-buffer lines, the first two routines of the algorithm are general to be executed with an arbitrary number of stations and parallel machines. Considering parallel machines, multi-stage operations and raw material and finished goods inventories, we succeed to obtain the steady-state distribution for quite a general setting. Before this study, the exact analysis of the lines having parallel machines was specific to 2-station lines to the best of our knowledge.

Moreover, we enrich the study with numerical experiments that allow us to observe the effects of processing times, parallel machines and the buffer capacities on throughput, average number of items in buffers and stock-out probabilities. Although two of our sub-routines are already designed for any number of stations, the general structure of the proposed method can be extended to longer lines. However, this extension could lead to the dimensionality problem of the state space. Instead, the provided exact analysis can be used for approximate analyses of longer lines, benefiting from the advantages of longer decomposition blocks than those considered in the literature (Diamantidis et al., 2020; Van Vuuren et al., 2005). In addition, some optimization routines, which maximize the throughput or minimize the system cost, can also be structured over the developed long-run analysis.

Dynamic programming and Markov decision process enable us to solve a wide range of optimal control problems. Alternatively, approximate solutions would be tractable since optimal control is costly due to the curse of dimensionality. As a future research direction, *reinforcement learning* (RL), one of the machine learning approaches, can be applied to solve control problems. The key aspect of reinforcement learning is to find an optimal way to make decisions while interacting with the environment (R. Sutton, 2018). The goal is that an agent selects actions to maximize the total expected future reward.

Q-learning is one of the RL approaches introduced by (Watkins, 1989), and its convergence is proven by (Watkins & Dayan, 1992). While dynamic programming uses state information to find the optimal action that maximizes the expected reward, Q-learning uses state-action pairs. Q-learning is a form of model-free reinforcement learning that constitutes an alternative to solve control problems when the model is unavailable. In Q-learning, a Q function describes the expected reward of a system in the long run for each state-action pair and determines the policy since a policy can be defined as a function from state to action. On the other hand, dynamic programming is used to solve optimal control problems if the system has a model. A model represents how an agent's actions change the environment. In our environment, a model can be represented by transition matrices, and for any state and action, immediate reward and the next state can be obtained by the model.

In reinforcement learning, optimization of a system can be conducted by considering the discounted (Watkins, 1989) or average (Bertsekas & Tsitsiklis, 1996; Das et al., 1999; Sridhar Mahadevan, 1996; Schwartz, 1993) reward. The studies considering RL techniques have been presented to the literature of production systems. (Paternina-Arboleda & Das, 2001) apply a reinforcement learning technique for a four-station production line to solve the control problem, assuming random processing times, times to failures and repair times and lost sales. (Xanthopoulos et al., 2008) derive control policies for serial production lines with backorders using an average reward RL algorithm and show that the derived algorithm outperforms the existing approaches such as Kanban, Base Stock and CONWIP. Additionally, scheduling problems of production lines (Arviv et al., 2016; Chen et al., 2015; Shiue et al., 2018) have been studied using RL techniques.

Q-Learning can learn optimal decisions for a Markovian problem. However, function approximation techniques are developed for the systems having larger state-action spaces. Q-learning constitutes a base to solve the control problem for larger systems using function approximations. Q-Learning visits certain state-action pairs and updates their values. Then function approximation techniques benefit information due to Q-learning to estimate the values of the unvisited state-action pairs. Function approximations are advantageous in many ways: it provides compact state-action value representation, reduces memory and computation requirements, can handle continuous state spaces, and uses generalization to unvisited states.

Many function approximators are presented in the literature (Buşoniu et al., 2010). For example, linear value function approximation is based on obtaining system features and then representing states with a function of the features. Features are mainly functions of states and actions. In linear value function approximation, the state-action value function (Q-function) is represented with a weighted linear combination of features:

$$\hat{Q}(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + .. + w_n f_n(s,a) \tag{19}$$

where w_i represents the weight of the feature $f_i(s,a)$, $i = 1, .., n$,when the current state is s and action is a.

The exact Q-values are obtained by the Q-function given in (20), and the approximate Q-values are obtained by updating the weights of features as defined in (21).

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[\mathcal{R} + \gamma \max_{a'}\{Q(s',a')\} - Q(s,a)\right] \tag{20}$$

$$w_i \leftarrow w_i + \alpha \left[\mathcal{R} + \gamma \max_{a'}\{Q(s',a')\} - Q(s,a)\right] f_i(s,a) \tag{21}$$

In equation (6), $\alpha \in (0,1]$ is the learning rate, $\mathcal{R}$ is immediate reward and $\gamma \in [0,1)$ is the discount rate. Q-learning uses the temporal difference (R. S. Sutton, 1988), which is defined as the difference between the updated Q value of the current state-action pair $\mathcal{R} + \gamma \max_{a'}\{Q_k(s',a')\}$ and the current value of the state-action pair $Q_k(s,a)$. At a given step, suppose that the current state is s and the action is a. First, the action a is taken for state s and the next state s' is observed. Then, the algorithm tries to choose optimal action from state s' from a set of possible actions a'.

The goal is to find the parameter vector $\vec{w}$ that minimizes the error between a true state-action value function Q and its approximation $\hat{Q}$. In order to evaluate the error, least-squares algorithms can be used (Bradtke & Barto, 1996). We foresee that function approximation techniques can be used to solve control problems of longer production lines.

The proposed policy *no intentional idleness* that we present in Chapter 4 relies on obtaining optimal values of control parameters with exhaustive search. Another solution approach to the study can be finding near-optimal results with a simulation optimization technique. Simulation optimization seeks to find the best values of variables without explicitly assessing every possibility (Carson & Maria, 1997). The basic components of a simulation optimization model consist of a set of decision variables, an objective function with constraints. (Olafsson & Kim, 2002) classify the simulation optimization methods based on continuous and discrete decision variables. For the models with continuous variables, stochastic approximation iterates solutions based on gradient estimation, in which early studies of the method falls in the 1950s (Robbins & Monro, 1951). Other methods include the simple path method (Gurkan et al., 1994) and the response surface method (Allen & Yu, 2000). Statistical selection (Chick & Inoue, 2000), random search and metaheuristic methods (Haddock & Mittenhall, 1992) are proposed for the models with discrete variables.

In the production environment, simulation optimization applications include production control mechanisms such as Kanban, Base Stock and CONWIP (Hall & Bowden, 1996; Xanthopoulos & Koulouriotis, 2014) based on optimizing control parameters to obtain near-optimal solutions. Moreover, reinforcement learning approaches consider simulation optimization techniques for optimizing production environments with Markov Decision Processes (Barde et al., 2019; S. Mahadevan & Theocharous, 1998; Paternina-Arboleda & Das, 2001). As production lines become longer, it would be harder to obtain optimal values of control variables with exhaustive search. Instead, near-optimal results can be achieved with simulation optimization techniques.

REFERENCES

Allen, T., & Yu, L. (2000). Low Cost Response Surface Methods for and from Simulation Optimization. *Proceedings of the Winter Simulation Conference*, 704–714.

Altiok, T. (1985). On the Phase-Type Approximations of General Distributions. *IIE Transactions*, *17*(2), 110–116. https://doi.org/10.1080/07408178508975280

Altiok, T. (1989). Approximate analysis of queues in series with phase-type service times and blocking. *Operations Research*, *37*(4), 601–610. https://doi.org/10.1287/opre.37.4.601

Altiok, T., & Ranjan, R. (1989). Analysis of production lines with general service times and finite buffers: A two-node decomposition approach. *Engineering Costs and Production Economics*, *17*(1–4), 155–165. https://doi.org/10.1016/0167-188X(89)90065-7

Altiok, T., & Stidham, S. (1983). The allocation of interstage buffer capacities in production lines. *IIE Transactions (Institute of Industrial Engineers)*, *15*(4), 292–299. https://doi.org/10.1080/05695558308974650

Arviv, K., Stern, H., & Edan, Y. (2016). Collaborative reinforcement learning for a two-robot job transfer flow-shop scheduling problem. *International Journal of Production Research*, *54*(4), 1196–1209. https://doi.org/10.1080/00207543.2015.1057297

Barde, S. R. A., Yacout, S., & Shin, H. (2019). Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. *Journal of Intelligent Manufacturing*, *30*(1), 147–161. https://doi.org/10.1007/s10845-016-1237-7

Baumann, H., & Sandmann, W. (2017). Multi-server tandem queue with Markovian arrival process, phase-type service times, and finite buffers. *European Journal of Operational Research*, *256*(1), 187–195. https://doi.org/10.1016/j.ejor.2016.07.035

Berman, O. (1982). Efficiency and production rate of a transfer line with two machines and a finite storage buffer. *European Journal of Operational Research*, *9*(3), 295–308.

Bertsekas, D. P. (2000). *Dynamic programming and optimal control: Vol. 1*. Belmont: Athena scientific.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Athena Scientific.

Bonvik, A. M., Couch, C. E., & Gershwin, S. B. (1997). A comparison of production-line control mechanisms. *International Journal of Production Research*, *35*(3), 789–804. https://doi.org/10.1080/002075497195713

Bradtke, S. J., & Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, *22*(1), 33–57.

Bulut, Ö., & Fadiloğlu, M. M. (2011). Production control and stock rationing for a make-to-stock system with parallel production channels. *IIE Transactions (Institute of Industrial Engineers)*, *43*(6), 432–450.

https://doi.org/10.1080/0740817X.2010.532853

Buşoniu, L., Babuška, R., De Schutter, B., & Ernst, D. (2010). Reinforcement learning and dynamic programming using function approximators. *Reinforcement Learning and Dynamic Programming Using Function Approximators*, 1–271. https://doi.org/10.1201/9781439821091

Buzacott, J. A. (1971). The role of inventory banks in flow-line production systems. *International Journal of Production Research*, *9*(4), 425–436. https://doi.org/10.1080/00207547108929891

Buzacott, J. A. (1989). Queueing models of Kanban and MRP controlled production systems. *Engineering Costs and Production Economics*, *17*(1–4), 3–20. https://doi.org/10.1016/0167-188X(89)90050-5

Buzacott, J. A., & Kostelski, D. (1987). Matrix-Geometric and Recursive Algorithm Solution of a Two-stage Unreliable Flow Line. *IIE Transactions (Institute of Industrial Engineers)*, *19*(4), 429–438. https://doi.org/10.1080/07408178708975416

Buzacott, J. A., & Shanthikumar, J. G. (1993). *Stochastic models of manufacturing systems* (4th ed.). Prentice Hall.

Carson, Y., & Maria, A. (1997). Simulation optimization: methods and applications. *Proceedings of the 29th Conference on Winter Simulation*, 118–126.

Chao, X., Chen, H., & Li, W. (1997). Optimal control for a tandem network of queues with blocking. *Acta Mathematicae Applicatae Sinica*, *13*(4), 425–437. https://doi.org/10.1007/BF02009552

Chen, C., Xia, B., Zhou, B., & Xi, L. (2015). A reinforcement learning based approach for a multiple-load carrier scheduling problem. *Journal of Intelligent Manufacturing*, *26*(6), 1233–1245. https://doi.org/10.1007/s10845-013-0852-9

Chick, S. E., & Inoue, K. (2000). New Results on Procedures that Select the Best System using CRN. *Proceedings of the Winter Simulation Conference*, 603–610.

Conway, R., Maxwell, W., McClain, J. O., & Thomas, L. J. (1988). Role of Work-in-Process Inventory in Serial Production Lines. *Operations Research*, *36*(2), 229–241. https://doi.org/10.1287/opre.36.2.229

Dallery, W., & Liberopoulos, G. (2000). Extended kanban control system: Combining kanban and base stock. In *IIE Transactions (Institute of Industrial Engineers)* (Vol. 32, Issue 4, pp. 369–386). https://doi.org/10.1080/07408170008963914

Das, T. K., Gosavi, A., Mahadevan, S., & Marchalleck, N. (1999). Solving semi-Markov decision problems using average reward reinforcement learning. *Management Science*, *45*(4), 560–574. https://doi.org/10.1287/mnsc.45.4.560

Diamantidis, A., Lee, J. H., Papadopoulos, C. T., Li, J., & Heavey, C. (2020). Performance evaluation of flow lines with non-identical and unreliable parallel machines and finite buffers. *International Journal of Production Research*, *58*(13), 3881–3904. https://doi.org/10.1080/00207543.2019.1636322

Diamantidis, A., & Papadopoulos, C. T. (2009). Exact analysis of a two-workstation one-buffer flow line with parallel unreliable machines. *European Journal of Operational Research*, *197*(2), 572–580.

https://doi.org/10.1016/j.ejor.2008.07.004

Diamantidis, A., Papadopoulos, C. T., & Heavey, C. (2007). Approximate analysis of serial flow lines with multiple parallel-machine stations. *IIE Transactions (Institute of Industrial Engineers)*, *39*(4), 361–375. https://doi.org/10.1080/07408170600838423

Duri, C., Frein, Y., & Di Mascolo, M. (2000). Comparison among three pull control policies: Kanban, base stock, and generalized kanban. *Annals of Operations Research*, *93*(1–4), 41–69. https://doi.org/10.1023/a:1018919806139

Framinan, J. M., González, P. L., & Ruiz-Usano, R. (2003). The CONWIP production control system: Review and research issues. *Production Planning and Control*, *14*(3), 255–265. https://doi.org/10.1080/0953728031000102595

Freeman, M. C. (1964). The effects of breakdowns and interstage storage on production line capacity. *Journal of Industrial Engineering*, *15*(4), 194–200.

Gavish, B., & Graves, S. C. (1980). One-Product Production/Inventory Problem Under Continuous Review Policy. *Operations Research*, *28*(5), 1228–1236. https://doi.org/10.1287/opre.28.5.1228

Gavish, B., & Graves, S. C. (1981). Production/inventory systems with a stochastic production rate under a continuous review policy. *Computers and Operations Research*, *8*(3), 169–183. https://doi.org/10.1016/0305-0548(81)90006-X

Gayon, J. P., de Véricourt, F., & Karaesmen, F. (2009). Stock rationing in an M/Er/1 multi-class make-to-stock queue with backorders. *IIE Transactions (Institute of Industrial Engineers)*, *41*(12), 1096–1109. https://doi.org/10.1080/07408170902800279

Gebennini, E., Grassi, A., & Fantuzzi, C. (2015). The two-machine one-buffer continuous time model with restart policy. In *Annals of Operations Research* (Vol. 231, Issue 1). Springer Science+Business Media New York. https://doi.org/10.1007/s10479-013-1373-9

Gebennini, E., Grassi, A., Fantuzzi, C., & Rimini, B. (2017). Discrete time model of a two-station one-buffer serial system with inventory level-dependent operation. *Computers and Industrial Engineering*, *113*, 46–63. https://doi.org/10.1016/j.cie.2017.09.007

Gershwin, S. B. (1987). Representation and analysis of transfer lines with machines that have different processing rates. *Annals of Operations Research*, *9*(1), 511–530. https://doi.org/10.1007/BF02054752

Gökçe, M. A., Cemali Dinçer, M., & Arslan Örnek, M. (2012). Analysis of transient throughput rates of transfer lines with pull systems. *Springer Optimization and Its Applications*, *60*, 287–304. https://doi.org/10.1007/978-1-4614-1123-9_13

Gupta, S. M., & Al-Turki, Y. A. Y. (1997). An algorithm to dynamically adjust the number of Kanbans in stochastic processing times and variable demand environment. *Production Planning & Control*, *8*(2), 133–141. https://doi.org/10.1080/095372897235398

Gurkan, G., Ozge, A. Y., & Robinson, T. M. (1994). Sample-Path Optimization in Simulation. *Proceedings of the 1994 Winter Simulation Conference*, 247–254.

Ha, A. Y. (1997a). Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Science*, *43*(8), 1093–1103. https://doi.org/10.1287/mnsc.43.8.1093

Ha, A. Y. (1997b). Stock-rationing policy for a make-to-stock production system with two priority classes and backordering. *Naval Research Logistics*, *44*(5), 457–472. https://doi.org/10.1002/(SICI)1520-6750(199708)44:5<457::AID-NAV4>3.0.CO;2-3

Ha, A. Y. (2000). Stock rationing in an M/Ek/1 make-to-stock queue. *Management Science*, *46*(1), 77–87. https://doi.org/10.1287/mnsc.46.1.77.15135

Haddock, J., & Mittenhall, J. (1992). Simulation Optimization using Simulated Annealing. *Computers and Industrial Engineering*, *22*(4), 387–395.

Hall, J. D., & Bowden, R. O. (1996). Simulation optimization for a manufacturing problem. *Proceedings of the 1996 Southeastern Simulation Conference*, 135–140.

Hatcher, J. M. (1969). The effect of internal storage on the production rate of a series of stages having exponential service times. *AIIE Transactions*, *1*(2), 150–156. https://doi.org/10.1080/05695556908974427

Heavey, C., Papadopoulos, H. T., & Browne, J. (1993). The throughput rate of multistation unreliable production lines. *European Journal of Operational Research*, *68*(1), 69–89. https://doi.org/10.1016/0377-2217(93)90077-Z

Helber, S. (2005). Analysis of flow lines with Cox-2-distributed processing times and limited buffer capacity. *OR Spectrum*, *27*(2–3), 221–242. https://doi.org/10.1007/s00291-005-0198-6

Hillier, F. S., & Boling, R. W. (1967). Finite queues in series with exponential or Erlang service times—a numerical approach. *Operations Research*, *15*(2), 286–303.

Hillier, F. S., & So, K. C. (1991). The effect of the coefficient of variation of operation times on the allocation of storage space in production line systems. *IIE Transactions (Institute of Industrial Engineers)*, *23*(2), 198–206. https://doi.org/10.1080/07408179108963854

Karaesmen, F., & Dallery, Y. (2000). Performance comparison of pull type control mechanisms for multi-stage manufacturing. *International Journal of Production Economics*, *68*(1), 59–71. https://doi.org/10.1016/S0925-5273(98)00246-1

Khojasteh, Y., & Sato, R. (2015). Selection of a pull production control system in multi-stage production processes. *International Journal of Production Research*, *53*(14), 4363–4379. https://doi.org/10.1080/00207543.2014.1001530

Kimura, O., & Terada, H. (1981). Desiǵn and analysis of Pull System, a method of multi-staǵe production control. *International Journal of Production Research*, *19*(3), 241–253. https://doi.org/10.1080/00207548108956651

Kirkavak, N., & Dinçer, C. (1999). General behavior of pull production systems: the allocation problems. *European Journal of Operational Research*, *119*(2), 479–494. https://doi.org/10.1016/S0377-2217(99)00148-4

Kumar, C. S., & Panneerselvam, R. (2007). Literature review of JIT-KANBAN system. *International Journal of Advanced Manufacturing Technology*, *32*(3–4), 393–408. https://doi.org/10.1007/s00170-005-0340-2

Lage Junior, M., & Godinho Filho, M. (2010). Variations of the kanban system: Literature review and classification. In *International Journal of Production Economics* (Vol. 125, Issue 1, pp. 13–21). https://doi.org/10.1016/j.ijpe.2010.01.009

Li, J., E. Blumenfeld, D., Huang, N., & M. Alden, J. (2009). Throughput analysis of production systems: Recent advances and future topics. *International Journal of Production Research*, *47*(14), 3823–3851. https://doi.org/10.1080/00207540701829752

Li, L. (2018). A systematic-theoretic analysis of data-driven throughput bottleneck detection of production systems. *Journal of Manufacturing Systems*, *47*, 43–52. https://doi.org/10.1016/j.jmsy.2018.03.001

Liberopoulos, G. (2020). Comparison of optimal buffer allocation in flow lines under installation buffer, echelon buffer, and CONWIP policies. *Flexible Services and Manufacturing Journal*, *32*(2), 297–365. https://doi.org/10.1007/s10696-019-09341-y

Lippman, S. A. (1975). Applying a New Device in the Optimization of Exponential Queuing Systems. *Operations Research*, *23*(4), 687–710. https://doi.org/10.1287/opre.23.4.687

Mahadevan, S., & Theocharous, G. (1998). Optimizing Production Manufacturing using Reinforcement Learning. *Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference*, *Gershwin 1994*, 372–377. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Optimizing+Production+Manufacturing+using+Reinforcement+Learning#0

Mahadevan, Sridhar. (1996). Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. *Machine Learning*, *22*, 159–195. https://doi.org/10.1007/978-0-585-33656-5_8

Matta, A., & Simone, F. (2016). Analysis of two-machine lines with finite buffer, operation-dependent and time-dependent failure modes. *International Journal of Production Research*, *54*(6), 1850–1862. https://doi.org/10.1080/00207543.2015.1085654

Meerkov, S. M., Shimkin, N., & Zhang, L. (2009). Transient behavior of two-machine geometric production lines. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *13*(PART 1), 498–503. https://doi.org/10.3182/20090603-3-RU-2001.0116

Meerkov, S. M., & Zhang, L. (2008). Transient behavior of serial production lines with Bernoulli machines. *IIE Transactions (Institute of Industrial Engineers)*, *40*(3), 297–312. https://doi.org/10.1080/07408170701488037

Meester, L. E., & Shanthikumar, J. G. (1990). Concavity of the throughput of tandem queueing systems with finite buffer storage space. *Advances in Applied Probability*, *22*(3), 764–767. https://doi.org/10.2307/1427472

Ohno, K., & Ichiki, K. (1987a). Computing Optimal Policies for Controlled Tandem Queueing Systems. *Operations Research*, *35*(1), 121–126. https://doi.org/10.1287/opre.35.1.121

Ohno, K., & Ichiki, K. (1987b). Computing Optimal Policies for Controlled Tandem Queueing Systems. *Operations Research*, *35*(1), 121–126. https://doi.org/10.1287/opre.35.1.121

Ohno, T. (1988). *Toyota production system: beyond large-scale production*. crc Press.

Olafsson, S., & Kim, J. (2002). Simulation Optimization. *Proceedings of the Winter Simulation Conference*, *1*, 79–84.

Papachristos, I., & Pandelis, D. G. (2020). Optimal server assignment in a two-stage tandem queueing system. *Operations Research Letters*, *48*(1), 71–77. https://doi.org/10.1016/j.orl.2019.12.001

Papadopoulos, C. T., Li, J., & O'Kelly, M. E. J. (2019). A classification and review of timed Markov models of manufacturing systems. *Computers and Industrial Engineering*, *128*(December 2018), 219–244. https://doi.org/10.1016/j.cie.2018.12.019

Papadopoulos, H. T., Heavey, C., & O'Kelly, M. E. J. (1989). Throughput rate of multistation reliable production lines with inter station buffers: (I) Exponential Case. *Computers in Industry*, *13*(3), 229–244. https://doi.org/10.1016/0166-3615(89)90113-9

Papadopoulos, H. T., Heavey, C., & O'Kelly, M. E. J. (1990). Throughput rate of multistation reliable production lines with inter station buffers (II) Erlang case. *Computers in Industry*, *13*(4), 317–335. https://doi.org/10.1016/0166-3615(90)90004-9

Patchong, A., Lemoine, T., & Kern, G. (2003). Improving car body production at PSA Peugeot Citroën. *Interfaces*, *33*(1). https://doi.org/10.1287/inte.33.1.36.12723

Patchong, A., & Willaeys, D. (2001). Modeling and analysis of an unreliable flow line composed of parallel-machine stages. *IIE Transactions (Institute of Industrial Engineers)*, *33*(7), 559–568. https://doi.org/10.1080/07408170108936854

Paternina-Arboleda, C. D., & Das, T. K. (2001). Intelligent dynamic control policies for serial production lines. *IIE Transactions (Institute of Industrial Engineers)*, *33*(1), 65–77. https://doi.org/10.1080/07408170108936807

Perros, H. G., & Altiok, T. (1986). Approximate Analysis of Open Networks of Queues With Blocking: Tandem Configurations. *IEEE Transactions on Software Engineering*, *SE-12*(3), 450–461. https://doi.org/10.1109/TSE.1986.6312886

Powell, S. G. (1994). Buffer allocation in unbalanced three-station serial lines. In *International Journal of Production Research* (Vol. 32, Issue 9, pp. 2201–2217).

Rao, N. P. (1975). Two-stage production systems with intermediate storage. *AIIE Transactions*, *7*(4), 414–421. https://doi.org/10.1080/05695557508975025

Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *Annals of Mathematical Statistics*, *22*, 400–407.

Romero-Silva, R., Marsillac, E., Shaaban, S., & Hurtado-Hernández, M. (2019). Serial production line performance under random variation: Dealing with the 'Law of Variability.' *Journal of Manufacturing Systems*, *50*(December 2018), 278–289. https://doi.org/10.1016/j.jmsy.2019.01.005

Ross, S. M. (2014). Introduction to Probability Models. In *Academic press*.

Schwartz, A. (1993). A reinforcement learning method for maximizing undiscounted rewards. *Proceedings of the Tenth International Conference on Machine Learning*, *298*, 298–305.

Shi, C., & Gershwin, S. B. (2016). A segmentation approach for solving buffer allocation problems in large production systems. *International Journal of Production Research*, *54*(20), 6121–6141. https://doi.org/10.1080/00207543.2014.991842

Shiue, Y.-R., Lee, K.-C., & Su, C.-T. (2018). Real-time scheduling for a smart factory using a reinforcement learning approach. *Computers & Industrial Engineering*, *125*, 604–614. https://doi.org/10.1016/j.cie.2018.03.039

Sivakumar, G. D., & Shahabudeen, P. (2008). Design of multi-stage adaptive kanban system. In *International Journal of Advanced Manufacturing Technology* (Vol. 38, Issues 3–4, pp. 321–336). https://doi.org/10.1007/s00170-007-1093-x

So, K. C. (1997). Optimal buffer allocation strategy for minimizing work-in-process inventory in unpaced production lines. *IIE Transactions (Institute of Industrial Engineers)*, *29*(1), 81–88. https://doi.org/10.1080/07408179708966314

Spearman, M. L., Woodruff, D. L., & Hopp, W. J. (1990). CONWIP: a pull alternative to kanban. *International Journal of Production Research*, *28*(5), 879–894. https://doi.org/10.1080/00207549008942761

Spinellis, D. D., & Papadopoulos, C. T. (2000). A simulated annealing approach for buffer allocation in reliable production lines. *Annals of Operations Research*, *93*(1–4), 373–384. https://doi.org/10.1023/a:1018984125703

Sutton, R. (2018). *Introduction to Reinforcement Learning (2nd Ed)*.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*(1), 9–44.

Tabe, T., Muramatsu, R., & Tanaka, Y. (1980). Analysis of production ordering quantities and inventory variations in a multi-stage production ordering system. In *International Journal of Production Research* (Vol. 18, Issue 2, pp. 245–257). https://doi.org/10.1080/00207548008919664

Tan, B., & Lagershausen, S. (2017). On the output dynamics of production systems subject to blocking. In *IISE Transactions* (Vol. 49, Issue 3, pp. 268–284). https://doi.org/10.1080/0740817X.2016.1222470

Tardif, V., & Maaseidvaag, L. (2001). An adaptive approach to controlling kanban systems. In *European Journal of Operational Research* (Vol. 132, Issue 2, pp. 411–424). https://doi.org/10.1016/S0377-2217(00)00119-3

Tolio, T. A. M., & Ratti, A. (2018). Performance evaluation of two-machine lines with generalized thresholds. *International Journal of Production Research*, *56*(1–2), 926–949. https://doi.org/10.1080/00207543.2017.1420922

van der Heijden, M. C. (1988). On the three-moment approximation of a general distribution by a coxian distribution. *Probability in the Engineering and Informational Sciences*, *2*(2), 257–261. https://doi.org/10.1017/S0269964800000772

Van Vuuren, M., Adan, I. J. B. F., & Resing-Sassen, S. A. E. (2005). Performance analysis of multi-server tandem queues with finite buffers and blocking. *OR Spectrum*, *27*(2–3), 315–338. https://doi.org/10.1007/s00291-004-0189-z

Veatch, M. H., & Wein, L. M. (1994). Optimal Control of a Two-Station Tandem Production/Inventory System. *Operations Research*, *42*(2), 337–350. https://doi.org/10.1287/opre.42.2.337

Vidalis, M. I., & Papadopoulos, H. T. (1999). Markovian analysis of production lines with Coxian-2 service times. *International Transactions in Operational Research*, *6*(5), 495–524. https://doi.org/10.1111/j.1475-3995.1999.tb00170.x

Wang, X., Andradóttir, S., Ayhan, H., & Suk, T. (2020a). Revenue maximization in two-station tandem queueing systems. In *Naval Research Logistics* (Vol. 67, Issue 2, pp. 77–107). https://doi.org/10.1002/nav.21887

Wang, X., Andradóttir, S., Ayhan, H., & Suk, T. (2020b). Revenue maximization in two-station tandem queueing systems. *Naval Research Logistics*, *67*(2), 77–107. https://doi.org/10.1002/nav.21887

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-Learning. *Machine Learning*, *8*, 279–292. https://doi.org/10.4018/978-1-59140-993-9.ch026

Weiss, S., Matta, A., & Stolletz, R. (2018). Optimization of buffer allocations in flow lines with limited supply. *IISE Transactions*, *50*(3), 191–202. https://doi.org/10.1080/24725854.2017.1328751

Weiss, S., Schwarz, J. A., & Stolletz, R. (2019). The buffer allocation problem in production lines: Formulations, solution methods, and instances. *IISE Transactions*, *51*(5), 456–485. https://doi.org/10.1080/24725854.2018.1442031

Xanthopoulos, A. S., Chnitidis, G., & Koulouriotis, D. E. (2019). Reinforcement learning-based adaptive production control of pull manufacturing systems. In *Journal of Industrial and Production Engineering* (Vol. 36, Issue 5, pp. 313–323). https://doi.org/10.1080/21681015.2019.1647301

Xanthopoulos, A. S., Ioannidis, S., & Koulouriotis, D. E. (2018). Optimal adaptive Kanban-type production control. *The International Journal of Advanced Manufacturing Technology*, *97*(5–8), 2887–2905. https://doi.org/10.1007/s00170-018-2110-y

Xanthopoulos, A. S., & Koulouriotis, D. E. (2014). Multi-objective optimization of production control mechanisms for multi-stage serial manufacturing-inventory systems. In *International Journal of Advanced Manufacturing Technology* (Vol. 74, Issues 9–12, pp. 1507–1519). https://doi.org/10.1007/s00170-014-6052-8

Xanthopoulos, A. S., Koulouriotis, D. E., & Gasteratos, A. (2008). A Reinforcement Learning Approach for Production Control in Manufacturing Systems. *International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems (ERLARS)*.

Xu, J., Serrano, A., & Lin, B. (2017). Optimal production and rationing policy of two-stage tandem production system. *International Journal of Production Economics*, *185*, 100–112. https://doi.org/10.1016/j.ijpe.2016.12.025

Yamashita, H., & Altiok, T. (1998). Buffer capacity allocation for a desired throughput in production lines. *IIE Transactions (Institute of Industrial Engineers)*, *30*(10), 883–891. https://doi.org/10.1080/07408179808966542

Yücel, Ö., & Bulut, Ö. (2019). Control of M/Cox-2/s make-to-stock systems. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, *10*(1), 26–36. https://doi.org/10.11121/ijocta.01.2020.00801

Zhang, L., Wang, C., Arinez, J., & Biller, S. (2013). Transient analysis of Bernoulli serial lines: Performance evaluation and system-theoretic properties. *IIE Transactions (Institute of Industrial Engineers)*, *45*(5), 528–543. https://doi.org/10.1080/0740817X.2012.721946

APPENDIX 1 – An Example of Optimal Policy of the Basic Model

Table A1.1. An optimal policy of the base case (Case 0) with λ=8

x3

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	1	6	6	6	6	6	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	0	0
	2	6	6	6	6	6	6	6	6	6	6	6	6	6	6	1	1	1	1	0	0	0	0
	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	1	1	1	0	0	0	0	0
	4	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	1	0	0	0	0	0	0
	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0
	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	3	0	0	0	0	0	0	0
	7	6	6	6	6	6	6	6	6	6	6	6	6	6	3	3	0	0	0	0	0	0	0
x2	8	6	6	6	6	6	6	6	6	6	6	6	6	3	3	3	0	0	0	0	0	0	0
	9	6	6	6	6	6	6	6	6	6	6	6	3	3	3	3	0	0	0	0	0	0	0
	10	6	6	6	6	6	6	6	6	6	6	3	3	3	3	3	0	0	0	0	0	0	0
	11	6	6	6	6	6	6	6	6	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	6	6	6	6	6	6	6	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	6	6	6	6	6	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	14	6	6	6	6	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	15	6	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0

x1=0

x3

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	1	0	0	0
	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	4	4	1	0	0	0	0
	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	4	1	0	0	0	0	0
	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	1	0	0	0	0	0	0
	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	6	0	0	0	0	0	0	0
	5	7	7	7	7	7	7	7	7	7	7	7	7	7	6	3	0	0	0	0	0	0	0
	6	7	7	7	7	7	7	7	7	7	7	7	7	6	3	3	0	0	0	0	0	0	0
	7	7	7	7	7	7	7	7	7	7	7	7	6	3	3	3	0	0	0	0	0	0	0
x2	8	7	7	7	7	7	7	7	7	7	7	6	3	3	3	3	0	0	0	0	0	0	0
	9	7	7	7	7	7	7	7	7	6	6	3	3	3	3	3	0	0	0	0	0	0	0
	10	7	7	7	7	7	7	7	6	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	7	7	7	7	7	6	6	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	7	7	7	6	6	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	13	6	6	6	6	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	14	6	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	15	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0

x1=1

x3

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	0	0	0
	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	4	4	2	0	0	0	0
	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	2	0	0	0	0	0
	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	2	0	0	0	0	0	0
	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	3	0	0	0	0	0	0	0
	5	7	7	7	7	7	7	7	7	7	7	7	7	7	3	3	0	0	0	0	0	0	0
	6	7	7	7	7	7	7	7	7	7	7	7	7	3	3	3	0	0	0	0	0	0	0
	7	7	7	7	7	7	7	7	7	7	7	7	3	3	3	3	0	0	0	0	0	0	0
x2	8	7	7	7	7	7	7	7	7	7	7	3	3	3	3	3	0	0	0	0	0	0	0
	9	7	7	7	7	7	7	7	7	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	7	7	7	7	7	7	7	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	7	7	7	7	7	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	7	7	7	7	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	13	7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	15	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0

x1=2

x3

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	0	0	0
	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	4	2	2	0	0	0	0
	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	2	2	0	0	0	0	0
	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	5	2	0	0	0	0	0	0
	4	7	7	7	7	7	7	7	7	7	7	7	7	7	5	5	0	0	0	0	0	0	0
	5	7	7	7	7	7	7	7	7	7	7	7	7	5	5	3	0	0	0	0	0	0	0
	6	7	7	7	7	7	7	7	7	7	7	7	5	3	3	3	0	0	0	0	0	0	0
	7	7	7	7	7	7	7	7	7	7	5	5	3	3	3	3	0	0	0	0	0	0	0
x2	8	7	7	7	7	7	7	7	7	5	5	3	3	3	3	3	0	0	0	0	0	0	0
	9	7	7	7	7	7	7	7	5	5	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	7	7	7	7	7	5	5	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	7	7	7	5	5	5	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	12	7	5	5	5	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	13	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	15	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0

x1=3

x3

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	0	0	0
	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	2	2	2	0	0	0	0
	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	5	2	2	0	0	0	0	0
	3	7	7	7	7	7	7	7	7	7	7	7	7	7	5	5	2	0	0	0	0	0	0
	4	7	7	7	7	7	7	7	7	7	7	7	7	5	5	5	0	0	0	0	0	0	0
	5	7	7	7	7	7	7	7	7	7	7	7	5	5	5	3	0	0	0	0	0	0	0
	6	7	7	7	7	7	7	7	7	7	5	5	5	3	3	3	0	0	0	0	0	0	0
	7	7	7	7	7	7	7	7	7	5	5	5	3	3	3	3	0	0	0	0	0	0	0
x2	8	7	7	7	7	7	7	7	5	5	5	3	3	3	3	3	0	0	0	0	0	0	0
	9	7	7	7	7	7	5	5	5	5	3	3	3	3	3	3	0	0	0	0	0	0	0
	10	7	7	7	5	5	5	5	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	11	5	5	5	5	5	5	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
	12	5	5	5	5	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	13	5	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	15	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0

x1=4

x3

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0
	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	2	2	2	0	0	0	0
	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	2	2	0	0	0	0	0
	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	2	0	0	0	0	0	0
	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0
	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	0	0	0	0	0	0	0
	6	5	5	5	5	5	5	5	5	5	5	5	5	5	3	3	0	0	0	0	0	0	0
	7	5	5	5	5	5	5	5	5	5	5	5	5	3	3	3	3	0	0	0	0	0	0
x2	8	5	5	5	5	5	5	5	5	5	5	3	3	3	3	3	3	0	0	0	0	0	0
	9	5	5	5	5	5	5	5	5	5	3	3	3	3	3	3	3	0	0	0	0	0	0
	10	5	5	5	5	5	5	5	5	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	11	5	5	5	5	5	5	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	12	5	5	5	5	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	13	5	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	15	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0
	16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0

x1=11

APPENDIX 2 – The Proof of Proposition for i=2

Proposition. There is a threshold T_i for $x_\mathrm{i} \geq 0, i \in \{1,2\}$ such that it is optimal not to produce at station-i when $x_\mathrm{i} > T_\mathrm{i}$ for all $x_\mathrm{j}, j \in \{1,2,3\} - \{i\}$.

Proof. For $i = 2$, let Eq0 be the below equation representing the discounted DP formulation of the system:

$$V(x_1, x_2, x_3) = \frac{1}{\nu}\langle h_1x_1 + h_2x_2 + h_3x_3$$
$$+ \lambda[c \cdot 1 \cdot (x_3 = 0) + V(x_1, x_2, \max\{x_3 - 1,0\})]$$
$$+\mu_1 min\{V(x_1 + 1, x_2, x_3), V(x_1, x_2, x_3)\}$$
$$+\mu_2 min\{V(x_1 - 1, x_2 + 1, x_3), V(x_1, x_2, x_3)\}$$
$$+\mu_3 min\{V(x_1, x_2 - 1, x_3 + 1), V(x_1, x_2, x_3)\}\rangle \qquad \text{(Eq0)}$$

where $\nu = \lambda + \mu_1 + \mu_2 + \mu_3 + \alpha$ and $x_1, x_2, x_3 \geq 0$. α is defined as the discount rate.

Suppose for $k \geq 0$

$$V^k(x_1, x_2, x_3) \leq V^k(x_1 - 1, x_2 + 1, x_3)$$

We need to show that $V^{k+1}(x_1, x_2, x_3) \leq V^{k+1}(x_1 - 1, x_2 + 1, x_3)$.

Property is true for k=0: $V^0(x_1, x_2, x_3) = 0 \leq V^0(x_1 - 1, x_2 + 1, x_3)$ since $V^0(x_1, x_2, x_3) = 0$ for all $x_1, x_2, x_3 \geq 0$.

Let Eq1 be the following equation:

$$V^{k+1}(x_1, x_2, x_3) = h_1x_1 + h_2x_2 + h_3x_3$$
$$+\lambda[c \cdot 1 \cdot (x_3 = 0) + V^k(x_1, x_2, \max\{x_3 - 1,0\})]$$
$$+\mu_1 min\{V^k(x_1 + 1, x_2, x_3), V^k(x_1, x_2, x_3)\}$$
$$+\mu_2 min\{V^k(x_1 - 1, x_2 + 1, x_3), V^k(x_1, x_2, x_3)\}$$
$$+\mu_3 min\{V^k(x_1, x_2 - 1, x_3 + 1), V^k(x_1, x_2, x_3)\} \qquad \text{(Eq1)}$$

Let Eq3 be the following equation:

$$V^{k+1}(x_1 - 1, x_2 + 1, x_3) = h_1(x_1 - 1) + h_2(x_2 + 1) + h_3x_3$$
$$+\lambda[c \cdot 1 \cdot (x_3 = 0) + V^k(x_1 - 1, x_2 + 1, \max\{x_3 - 1,0\})]$$
$$+\mu_1 min\{V^k(x_1, x_2 + 1, x_3), V^k(x_1 - 1, x_2 + 1, x_3)\}$$
$$+\mu_2 min\{V^k(x_1 - 2, x_2 + 2, x_3), V^k(x_1 - 1, x_2 + 1, x_3)\}$$
$$+\mu_3 min\{V^k(x_1 - 1, x_2, x_3 + 1), V^k(x_1 - 1, x_2 + 1, x_3)\} \qquad \text{(Eq3)}$$

So, each term should separately be considered:

[5]. $h_1x_1 + h_2x_2 + h_3x_3 \leq h_1(x_1 - 1) + h_2(x_2 + 1) + h_3x_3$ holds due to positive holding cost rates

[6]. If $x_3 > 0$, then $\lambda V^k(x_1 - 1, x_2, x_3 - 1) \leq \lambda V^k(x_1 - 1, x_2 + 1, x_3 - 1)$ holds due to the supposition

If $x_3 = 0$, then $\lambda[c + V^k(x_1 - 1, x_2, 0)] \leq \lambda[c + V^k(x_1 - 1, x_2 + 1, 0)]$ holds due to the supposition

[7]. $\mu_2 V^k(x_1, x_2, x_3) \leq \mu_2 V^k(x_1 - 1, x_2 + 1, x_3)$ holds for Eq1 and

$\mu_2 V^k(x_1 - 1, x_2 + 1, x_3) \leq \mu_2 V^k(x_1 - 2, x_2 + 2, x_3)$ holds for Eq3 due to the supposition

[8]. For the decisions regarding μ_1 and μ_3, there are 4+4=8 possible combinations of control actions in Eq1 and Eq3.

Consider control actions regarding μ_1:

a. It is optimal for station-1 *to produce* in both Eq1 and Eq3.
   Eq1 returns $V^k(x_1 + 1, x_2, x_3)$ and Eq3 returns $V^k(x_1, x_2 + 1, x_3)$
   Hence, $V^k(x_1 + 1, x_2, x_3) \leq V^k(x_1, x_2 + 1, x_3)$ holds due to the supposition.

b. It is optimal for station-1 *not to produce* in both Eq1 and Eq3.
   Eq1 returns $V^k(x_1, x_2, x_3)$ and Eq3 returns $V^k(x_1 - 1, x_2 + 1, x_3)$
   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1 - 1, x_2 + 1, x_3)$ holds due to the supposition.

c. It is optimal for station-1 *to produce* in Eq1 but *not to produce* in Eq3.
   Eq1 states that $V^k(x_1 + 1, x_2, x_3) \leq V^k(x_1, x_2, x_3)$
   Eq3 states that $V^k(x_1 - 1, x_2 + 1, x_3) \leq V^k(x_1 - 1, x_2, x_3 + 3)$
   Hence, $V^k(x_1 + 1, x_2, x_3) \leq V^k(x_1 - 1, x_2 + 1, x_3)$ holds due to

$$V^k(x_1 + 1, x_2, x_3) \leq V^k(x_1, x_2, x_3) \leq V^k(x_1 - 1, x_2 + 1, x_3) \leq V^k(x_1 - 1, x_2, x_3 + 3)$$

d. It is optimal for station-1 *not to produce* in Eq1 but *to produce* in Eq3.
   Eq1 states that $V^k(x_1, x_2, x_3) \leq V^k(x_1 + 1, x_2, x_3)$
   Eq3 states that $V^k(x_1, x_2 + 1, x_3) \leq V^k(x_1 - 1, x_2 + 1, x_3)$
   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1, x_2 + 1, x_3)$ holds similarly to the part c.

Consider control actions regarding μ_3:

e. It is optimal for station-3 *to produce* in both Eq1 and Eq3.
   Eq1 returns $V^k(x_1, x_2 - 1, x_3 + 1)$ and Eq3 returns $V^k(x_1 - 1, x_2, x_3 + 1)$
   Hence, $V^k(x_1, x_2 - 1, x_3 + 1) \leq V^k(x_1 - 1, x_2, x_3 + 1)$ holds due to the supposition.

f. It is optimal for station-3 *not to produce* in both Eq1 and Eq3.
   Eq1 returns $V^k(x_1, x_2, x_3)$ and Eq3 returns $V^k(x_1 - 1, x_2 + 1, x_3)$
   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1 - 1, x_2 + 1, x_3)$ holds.

g. It is optimal for station-3 *to produce* in Eq1 but *not to produce* in Eq3.
   Eq1 states that $V^k(x_1, x_2 - 1, x_3 + 1) \leq V^k(x_1, x_2, x_3)$
   Eq3 states that $V^k(x_1 - 1, x_2 + 1, x_3) \leq V^k(x_1 - 1, x_2, x_3 + 1)$
   Hence, $V^k(x_1, x_2 - 1, x_3 + 1) \leq V^k(x_1 - 1, x_2 + 1, x_3)$ holds similarly to the part c.

h. It is optimal for station-3 *not to produce* in Eq1 but *to produce* in Eq3.
   Eq1 states that $V^k(x_1, x_2, x_3) \leq V^k(x_1, x_2 - 1, x_3 + 1)$
   Eq3 states that $V^k(x_1 - 1, x_2, x_3 + 1) \leq V^k(x_1 - 1, x_2 + 1, x_3)$
   Hence, $V^k(x_1, x_2, x_3) \leq V^k(x_1 - 1, x_2, x_3 + 1)$ holds similarly to the part c.

We can conclude that the property holds for k+1. As $k \to \infty$, value function V converges with a given epsilon error, and the optimal value is found for the problem. Also, the average cost is obtained while setting α to 0 and dividing the value function by time steps.■

APPENDIX 3 – Buffer Capacities of the Extended Models

The below table represents the optimal buffer capacities of the proposed *no intentional idleness* policy for the extended models.

Table A3.1. Optimal buffer capacities of extended models

λ=3	Model 1									Model 2									Model 3								
	γ=5			γ=10			γ=20			γ=5			γ=10			γ=20			γ=5			γ=10			γ=20		
β	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*
0.0	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	1	1	3	1	1	3	1	1	3
0.1	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	1	1	3	1	1	3	1	1	3
0.2	1	0	3	0	0	3	0	0	3	0	1	3	0	0	3	0	0	3	1	1	4	1	1	3	1	1	3
0.3	1	0	3	0	0	3	0	0	3	0	1	3	0	0	3	0	0	3	1	1	4	1	1	3	1	1	3
0.4	1	0	3	0	0	3	0	0	3	0	1	4	0	1	3	0	0	3	1	1	4	1	1	3	1	1	3
0.5	1	1	3	0	0	3	0	0	3	0	2	3	0	1	3	0	0	3	1	1	5	1	1	3	1	1	3
0.6	1	1	3	1	0	3	0	0	3	0	3	3	0	1	3	0	0	3	1	1	5	1	1	4	1	1	3
0.7	3	0	3	1	0	3	0	0	3	0	3	3	0	1	3	0	0	3	1	1	6	1	1	4	1	1	3
0.8	4	0	3	1	0	3	0	0	3	0	4	3	0	1	3	0	0	3	1	1	6	1	1	4	1	1	3
0.9	4	0	3	1	0	3	0	0	3	0	4	3	0	1	3	0	0	3	1	1	7	1	1	4	1	1	3
1.0	5	0	3	1	0	3	0	0	3	1	5	3	0	1	3	0	1	3	1	1	8	1	1	4	1	1	3
λ=4	Model 1									Model 2									Model 3								
	γ=5			γ=10			γ=20			γ=5			γ=10			γ=20			γ=5			γ=10			γ=20		
β	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*
0.0	0	1	4	0	1	4	0	1	4	0	1	4	0	1	4	0	1	4	1	1	4	1	1	4	1	1	4
0.1	1	1	4	0	1	4	0	1	4	0	1	5	0	1	4	0	1	4	1	1	5	1	1	4	1	1	4
0.2	1	1	4	1	1	4	0	1	4	1	2	4	0	1	4	0	1	4	1	1	5	1	1	4	1	1	4
0.3	2	1	4	1	1	4	0	1	4	1	2	5	1	1	4	0	1	4	1	1	6	1	1	5	1	1	4
0.4	3	1	4	1	1	4	0	1	4	1	3	5	1	1	4	0	1	4	1	1	7	1	1	5	1	1	4
0.5	4	1	4	1	1	4	1	1	4	1	4	5	1	2	4	0	1	4	1	1	8	1	1	5	1	1	4
0.6	6	1	4	1	1	4	1	1	4	1	5	5	1	2	4	1	1	4	1	1	10	1	1	6	1	1	4
0.7	8	1	4	2	1	4	1	1	4	1	8	4	1	2	4	1	1	4	1	1	11	1	1	6	1	1	5
0.8	12	1	4	2	1	4	1	1	4	2	10	4	1	2	5	1	1	4	1	1	14	1	1	6	1	1	5
0.9	18	1	4	2	1	4	1	1	4	2	14	4	1	2	5	1	1	4	1	1	17	1	1	7	1	1	5
1.0	26	0	3	3	1	4	1	1	4	2	19	4	1	3	4	1	1	4	1	1	20	1	1	7	1	1	5

Table A3.1 (cont'd). Optimal buffer capacities of extended models

λ=5	Model 1									Model 2									Model 3								
	γ=5			γ=10			γ=20			γ=5			γ=10			γ=20			γ=5			γ=10			γ=20		
β	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*
0.0	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6
0.1	2	1	6	1	2	5	1	1	6	1	2	6	1	2	5	1	2	5	1	1	7	1	1	6	1	1	6
0.2	3	2	5	1	2	5	1	2	5	1	4	6	1	2	6	1	2	5	1	2	8	1	2	6	1	1	6
0.3	4	2	6	2	1	6	1	2	5	2	5	6	1	2	6	1	2	5	1	2	10	1	1	7	1	1	6
0.4	7	2	6	2	2	5	1	2	5	2	7	6	1	3	6	1	2	5	1	2	12	1	2	7	1	1	6
0.5	12	2	5	3	1	6	1	2	5	3	10	6	1	3	6	1	2	6	0	3	15	1	2	8	1	2	6
0.6	19	1	5	3	2	6	2	1	6	3	17	5	2	3	6	1	2	6	1	3	19	1	2	8	1	1	7
0.7	24	1	4	4	2	5	2	1	6	3	22	5	2	4	6	1	2	6	0	3	23	1	2	9	1	1	7
0.8	22	1	3	5	2	6	2	1	6	2	24	4	2	5	6	1	2	6	0	3	24	1	2	10	1	1	7
0.9	16	0	3	7	1	6	2	2	5	2	23	3	2	7	6	1	2	6	0	2	23	1	2	11	1	2	7
1.0	14	0	3	9	2	5	2	2	5	2	18	3	2	8	6	1	3	6	0	2	22	1	2	13	1	2	7

λ=8	Model 1									Model 2									Model 3								
	γ=5			γ=10			γ=20			γ=5			γ=10			γ=20			γ=5			γ=10			γ=20		
β	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*	m_1^*	m_2^*	m_3^*
0.0	5	6	12	5	6	12	5	6	12	5	6	12	5	6	12	5	6	12	5	6	12	5	6	12	5	6	12
0.1	11	7	12	7	6	12	6	6	12	7	11	13	5	8	13	5	7	12	4	8	19	4	7	15	5	6	13
0.2	26	8	11	9	7	12	6	7	12	8	24	12	6	11	13	5	8	12	4	8	32	4	7	18	4	7	14
0.3	25	5	8	14	8	12	7	7	12	7	25	8	7	14	13	6	8	13	3	7	27	4	7	23	4	7	16
0.4	25	3	6	23	6	11	9	7	12	5	19	6	7	23	12	6	10	13	2	5	21	3	8	29	4	7	17
0.5	19	2	5	26	5	9	11	7	12	4	18	5	6	23	10	6	12	13	1	5	20	3	6	25	4	7	19
0.6	16	2	4	25	4	7	14	7	12	4	15	5	6	21	8	7	14	13	1	4	14	2	6	23	4	7	22
0.7	14	1	4	24	3	6	19	7	12	3	12	4	5	17	7	7	18	12	1	3	11	2	5	22	4	7	25
0.8	9	1	3	20	3	5	25	6	11	3	11	4	4	15	6	6	21	12	1	2	10	1	5	17	3	7	27
0.9	7	1	3	20	2	5	29	5	10	2	8	3	4	15	5	6	22	11	1	2	8	1	4	16	3	6	27
1.0	9	1	3	16	2	4	26	5	8	2	7	3	3	13	5	6	25	9	1	2	8	1	3	15	3	6	27

APPENDIX 4 – The Algorithm of Repetition Coefficients

Algorithm A recursively calculates the repetition coefficients $C1_j^k$ and $C2_j^k$, which are used in j^{th} station transition calculation for the system which has k many stations.

Algorithm A	
Basis Step	**Recursion Step**
$k = 1;$ $j = 1;$ $C1_j^k = 1;$ $C2_j^k = 1;$	For $k = 2$ to M $C1_2^k = \frac{s_k(1+s_k)}{2} + (1 + s_k)(1 + m_{k+1})$ $C2_2^k = \sum_{x=0}^{s_k} \left((x + 1)(1 + m_{k+1}) + \frac{x(1+x)}{2}\right)$ For $j = 3$ to k $C1_j^k = C1_{j-1}^k \frac{(s_{k-j+2})(1+s_{k-j+2})}{2} + (1 + s_{k-j+2})\left(C1_{j-1}^k m_{k-j+3} + C2_{j-1}^k\right)$ $C2_j^k = \sum_{x=0}^{s_{k-j+2}} \left(C1_{j-1}^k \frac{x(1+x)}{2} + (x + 1)\left(C1_{j-1}^k m_{k-j+3} + C2_{j-1}^k\right)\right)$

Figure A4.1. The algorithm of repetition coefficients - The upper bound of the runtime of this algorithm is given by $O(M^2 * max(s_1, \ldots, s_M))$

APPENDIX 5 – Nested Loop Blocks with Big-O Complexity

Table A5.1. Nested loop blocks

Name	Nested Loop Block	Big-O Complexity
L_i^0	For $p = 0$ to $m_1 - 1$	$O(m_1)$
L_i^1	For $k_i = 0$ to s_i For $j_i = k_i$ to s_i	$O(s_i^2)$

L_i^2	For $k_i = 0$ to s_i
	    For $j_i = k_i$ to s_i
	      For $l_i = 1$ to $s_i + m_{i+1} - j_i$
		$O(s_i^2 * (m_{i+1} + s_i))$

L_i^3	For $k_i = 0$ to s_i	$O(s_i)$
L_i^4	For $k_i = 0$ to s_i For $l_i = 1$ to $s_i - k_i$	$O(s_i^2)$
L_i^5	For $k_i = 0$ to s_i For $l_i = 1$ to m_{i+1}	$O(s_i * m_{i+1})$

L_i^6	For $k_i = 0$ to s_i
	    For $j_i = k_i$ to s_i
	      For $l_i = 1$ to $s_i + m_{i+1} - j_i - 1$
		$O(s_i^2 * (m_{i+1} + s_i))$

L_i^7	For $k_i = 0$ to s_i For $l_i = 1$ to $m_{i+1} - 1$	$O(s_i * m_{i+1})$

L_i^8	For $k_i = 1$ to s_i
	    For $j_i = k_i$ to s_i
	      For $l_i = 1$ to $s_i + m_{i+1} - j_i + 1$
		$O(s_i^2 * (m_{i+1} + s_i))$

L_i^9	For $k_i = 1$ to s_i For $l_i = 1$ to $s_i - k_i$	$O(s_i^2)$
L_i^{10}	For $k_i = 1$ to s_i For $l_i = 1$ to $C1_{M-i+1}^M$	$O(s_i)$
L_i^{11}	For $k_i = 1$ to s_i For $l_i = 1$ to $m_{i+1} - 1$	$O(s_i * m_{i+1})$
L_i^{12}	For $k_i = 1$ to s_i For $l_i = 1$ to $C2_{M-i+1}^M$	$O(s_i)$
L_i^{13}	For $k_i = 1$ to s_i For $j_i = k_i$ to s_i	$O(s_i^2)$

L_i^{14}	For $k_i = 1$ to s_i
	    For $j_i = k_i$ to s_i
	      For $l_i = 1$ to $s_i + m_{i+1} - j_i$
		$O(s_i^2 * (m_{i+1} + s_i))$

L_i^{15}	For $k_i = 1$ to s_i	$O(s_i)$

Table A5.1 (cont'd). Nested loop blocks

L_i^{16}	For $k_i = 1$ to s_i For $j_i = k_i$ to s_i For $l_i = 1$ to $C1_{M-i+1}^{M}$	$O(s_i^2)$
L_i^{17}	For $k_i = 0$ to s_i For $l_i = 1$ to $m_{i+1} + 1$	$O(s_i * m_{i+1})$
L_i^{18}	For $k_i = 1$ to s_i For $l_i = 1$ to $C1_{M-i+1}^{M}(s_i - k_i + 1)$	$O(s_i^2)$
L_i^{19}	For $k_i = 1$ to s_i For $l_i = 1$ to $\left(C1_{M-i+1}^{M}\frac{(s_i-k_i)(s_i+1-k_i)}{2} + (s_i + 1 -\right.$ $\left.k_i)(C1_{M-i+1}^{M}m_{i+1} + C2_{M-i+1}^{M})\right)$	$O(s_i^2 *$ $\max(m_{i+1}, s_i))$
L_i^{20}	For $k_i = 1$ to s_i For $l_i = 1$ to $(C1_{M-i+1}^{M}(s_i - k_i + m_{i+1}) + C2_{M-i+1}^{M})$	$O(s_i * (m_{i+1} + s_i))$
L_i^{21}	For $l_i = 1$ to $\left(C1_M^M\frac{s_1(s_1+1)}{2} + (s_1 + 1)(C1_M^M m_2 + C2_M^M)\right)(m_1 - 1)$	$O(s_1^2 * m_1)$

APPENDIX 6 – Equation Details of Base Case Algorithms

Table A6.1. Equation details

$eq^{0}_{i(1)} = p\left(C1^{M}_{M}\frac{s_1(s_1+1)}{2} + (s_1+1)(C1^{M}_{M}m_2 + C2^{M}_{M})\right) + \sum_{x=0}^{s_1}\left((x+1)(m_2+1) + \frac{x(x+1)}{2}\right)$
$eq^{0}_{i(2)} = (p-1)\left(C1^{M}_{M}\frac{s_1(s_1+1)}{2} + (s_1+1)(C1^{M}_{M}m_2 + C2^{M}_{M})\right) + \sum_{x=0}^{s_1}\left((x+1)(m_2+1) + \frac{x(x+1)}{2}\right)$
$eq^{2}_{i(3)} = l_i + C2^{M}_{M-i+1} + \sum_{x=0}^{k_i-1}\left(C1^{M}_{M-i+1}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i+1-x)(C1^{M}_{M-i+1}m_{i+1} + C2^{M}_{M-i+1})\right)$ $+ \sum_{x=k_i}^{j_i-1}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{2}_{i(4)} = l_i + eq^{2}_{i(3)} - C2^{M}_{M-i+1}$
$eq^{5}_{i(2)} = l_i - C1^{M}_{M-i+1}(m_{i+1}) + \sum_{x=1}^{k_i+1}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x + 1) + C2^{M}_{M-i+1})$
$eq^{5}_{i(3)} = eq^{5}_{i(2)} - C2^{M}_{M-i+1}$
$eq^{8}_{i(1)} = l_i + \sum_{x=0}^{k_i-1}\left(C1^{M}_{M-i+1}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i+1-x)(C1^{M}_{M-i+1}m_{i+1} + C2^{M}_{M-i+1})\right)$ $+ \sum_{x=k_i}^{j_i-1}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{8}_{i(2)} = l_i + C1^{M}_{M-i+1} + \sum_{x=0}^{k_i-2}\left(C1^{M}_{M-i+1}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i+1-x)(C1^{M}_{M-i+1}m_{i+1} + C2^{M}_{M-i+1})\right)$ $+ \sum_{x=k_i}^{j_i-1}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x + 1) + C2^{M}_{M-i+1})$
$eq^{8}_{i(6)} = l_i + eq^{8}_{i(2)} - C1^{M}_{M-i+1}$
$eq^{8}_{i(7)} = l_i + \sum_{x=k_i}^{j_i-1}\left(C1^{M}_{M-i+1}\frac{(s_i-x+k_i)(s_i+1-x+k_i)}{2} + (s_i+1-x+k_i)(C1^{M}_{M-i+1}m_{i+1} + C2^{M}_{M-i+1})\right)$ $+ \sum_{x=0}^{k_i-1}(C1^{M}_{M-i+1}(s_i - j_i + m_{i+1} + x + 2) + C2^{M}_{M-i+1})$
$eq^{8}_{i(8)} = l_i + \sum_{x=k_i}^{j_i-1}\left(C1^{M}_{M-i+1}\frac{(s_i-x+k_i)(s_i+1-x+k_i)}{2} + (s_i+1-x+k_i)(C1^{M}_{M-i+1}m_{i+1} + C2^{M}_{M-i+1})\right)$ $+ C1^{M}_{M-i+1} + \sum_{x=1}^{k_i-1}(C1^{M}_{M-i+1}(s_i - j_i + m_{i+1} + x + 2) + C2^{M}_{M-i+1})$
$eq^{9}_{i(1)} = l_i + \sum_{x=0}^{k_i-1}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{9}_{i(2)} = l_i + \sum_{x=0}^{k_i-2}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{11}_{i(1)} = l_i - C1^{M}_{M-i+1}m_{i+1} + \sum_{x=0}^{k_i}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{11}_{i(2)} = l_i - C1^{M}_{M-i+1}(-1 + m_{i+1}) + \sum_{x=0}^{k_i}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{11}_{i(3)} = l_i - C1^{M}_{M-i+1}(-1 + m_{i+1}) + \sum_{x=0}^{k_i}\left(C1^{M}_{M-i+1}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i+1-x)(C1^{M}_{M-i+1}m_{i+1} +\right.$ $\left.C2^{M}_{M-i+1})\right)$
$eq^{11}_{i(5)} = l_i - C1^{M}_{M-i+1}m_{i+1} + \sum_{x=0}^{k_i-1}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{12}_{i(1)} = l_i - C1^{M}_{M-i+1}m_{i+1} - C2^{M}_{M-i+1} + \sum_{x=0}^{k_i}(C1^{M}_{M-i+1}(s_i + m_{i+1} - x) + C2^{M}_{M-i+1})$
$eq^{12}_{i(3)} = l_i - C1^{M}_{M-i+1}(-1 + m_{i+1}) - C2^{M}_{M-i+1}$ $+ \sum_{x=0}^{k_i}\left(C1^{M}_{M-i+1}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i+1-x)(C1^{M}_{M-i+1}m_{i+1} + C2^{M}_{M-i+1})\right)$

Table A6.1 (con't). Equation details

$eq_{i(2)}^{12} = l_i - C1_{M-i+1}^{M}(-1 + m_{i+1}) - C2_{M-i+1}^{M} + \sum_{x=0}^{k_i}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(4)}^{12} = l_i - C1_{M-i+1}^{M}m_{i+1} - C2_{M-i+1}^{M} + \sum_{x=0}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(6)}^{15} = C1_{M-i+1}^{M}(s_i - k_i + 1) + \sum_{x=1}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x + 1) + C2_{M-i+1}^{M})$
$eq_{i(7)}^{15} = \sum_{x=0}^{k_i}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(1)}^{16} = l_i + \sum_{x=0}^{k_i-2}\left(C1_{M-i+1}^{M}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i + 1 - x)(C1_{M-i+1}^{M}m_{i+1} + C2_{M-i+1}^{M})\right)$ $+ \sum_{x=k_i}^{j_i}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x + 1) + C2_{M-i+1}^{M}) - C1_{M-i+1}^{M}$
$eq_{i(2)}^{16} = l_i + C1_{M-i+1}^{M}(j_i - k_i) + \sum_{x=0}^{k_i-2}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(1)}^{17} = l_i + \sum_{x=0}^{k_i-1}\left(C1_{M-i+1}^{M}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i + 1 - x)(C1_{M-i+1}^{M}m_{i+1} + C2_{M-i+1}^{M})\right)$ $+ \sum_{x=k_i}^{s_i}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(2)}^{17} = l_i + C1_{M-i+1}^{M}(s_i - k_i) + \sum_{x=0}^{k_i-2}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(1)}^{18} = l_i + \left(l_i > C1_{M-i+1}^{M}(s_i - k_i)\right)(C1_{M-i+1}^{M}(-1 + m_{i+1}) + C2_{M-i+1}^{M})$ $+ \sum_{x=0}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(2)}^{18} = l_i + \sum_{x=1}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x + 1) + C2_{M-i+1}^{M})$
$eq_{i(3)}^{18} = l_i + C1_{M-i+1}^{M} + \left(i > C1_{M-i+1}^{M}(s_i - k_i)\right)(C1_{M-i+1}^{M}(-1 + m_{i+1}) + C2_{M-i+1}^{M})$ $+ \sum_{x=0}^{k_i-2}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(4)}^{18} = \sum_{x=0}^{k_i-1}\left(C1_{M-i+1}^{M}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i + 1 - x)(C1_{M-i+1}^{M}m_{i+1} + C2_{M-i+1}^{M})\right)$ $+ \sum_{x=k_i}^{k_i+l_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(1)}^{19} = l_i + \sum_{x=0}^{k_i-1}\left(C1_{M-i+1}^{M}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i + 1 - x)(C1_{M-i+1}^{M}m_{i+1} + C2_{M-i+1}^{M})\right)$
$eq_{i(2)}^{19} = l_i + C2_{M-i+1}^{M} + \sum_{x=1}^{k_i-1}\left(C1_{M-i+1}^{M}\frac{(s_i-x)(s_i+1-x)}{2} + (s_i + 1 - x)(C1_{M-i+1}^{M}m_{i+1} + C2_{M-i+1}^{M})\right)$ $+ \sum_{x=0}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(1)}^{20} = l_i + \sum_{x=0}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(2)}^{20} = l_i + C1_{M-i+1}^{M}k_i + C2_{M-i+1}^{M} + \sum_{x=0}^{k_i-1}(C1_{M-i+1}^{M}(s_i + m_{i+1} - x) + C2_{M-i+1}^{M})$
$eq_{i(1)}^{21} = l_i + \sum_{x=0}^{s_1}\left((x + 1)(m_2 + 1) + \frac{x(x+1)}{2}\right)$
$eq_{i(2)}^{21} = l_i + \left(C1_{M}^{M}\frac{s_1(s_1+1)}{2} + (s_1 + 1)(C1_{M}^{M}m_2 + C2_{M}^{M})\right) + \sum_{x=0}^{s_1}\left((x + 1)(m_2 + 1) + \frac{x(x+1)}{2}\right)$

APPENDIX 7 – Algorithm Structures of Events P2, P3

Table A7.1. Algorithm structures of events P2 and P3

	$A^1_{P2,P3}\left(I^{M=1}_{j=1}\right)$	$A^2_{P2,P3}\left(I^{M=2,3}_{j=1}\right)$
(1)	L_1^8	$L_1^{13} \rightarrow L_2^{k=8,16,17}$ L_1^{14}
(2)	$L_1^0 \rightarrow L_1^{18}$ $L_1^0 \rightarrow L_1^{11}$ $L_1^0 \rightarrow L_1^{12}$	$L_1^0 \rightarrow L_1^{15} \rightarrow L_2^{k=8,16}$ $L_1^0 \rightarrow L_1^{k=11,18}$

	$A^3_{P2,P3}\left(I^{M=2}_{j=2}\right)$	$A^4_{P2,P3}\left(I^{M=3}_{j=2}\right)$
(3)	$L_1^1 \rightarrow L_2^8$	$L_1^{13} \rightarrow L_2^{k=8,16,17}$ $L_1^1 \rightarrow L_2^{14}$
(4)	$L_1^2 \rightarrow L_2^{k=9,10}$ $L_1^1 \rightarrow L_2^{k=11,12}$ $L_1^6 \rightarrow L_2^{k=11,12}$	$L_1^2 \rightarrow L_2^{k=9,10}$ $L_1^1 \rightarrow L_2^{k=11,12}$ $L_1^6 \rightarrow L_2^{k=11,12}$
(5)	$L_1^0 \rightarrow L_1^3 \rightarrow L_2^8$	$L_1^0 \rightarrow L_1^3 \rightarrow L_2^{13} \rightarrow L_3^{k=8,16,17}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{14}$
(6)	$L_1^0 \rightarrow L_1^4 \rightarrow L_2^{k=9,10,11,12}$	$L_1^0 \rightarrow L_1^4 \rightarrow L_2^{15} \rightarrow L_3^{k=8,16,17}$ $L_1^0 \rightarrow L_1^4 \rightarrow L_2^{k=9,10,11}$
(7)	$L_1^0 \rightarrow L_1^5 \rightarrow L_2^{k=9,10}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{k=11,12}$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^{k=11,12}$	$L_1^0 \rightarrow L_1^5 \rightarrow L_2^{k=9,10}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{k=11}$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^{k=11}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{15} \rightarrow L_3^{k=8,16,17}$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^{15} \rightarrow L_3^{k=8,16,17}$

	$A^3_D\left(I^{M=3}\right)$
(8)	$L_1^1 \rightarrow L_2^1 \rightarrow L_3^2$
(9)	$L_1^1 \rightarrow L_2^2 \rightarrow L_3^5$ $L_1^1 \rightarrow L_2^1 \rightarrow L_3^{18}$ $L_1^1 \rightarrow L_2^6 \rightarrow L_3^{k=9,15}$
(10)	$L_1^2 \rightarrow L_2^3 \rightarrow L_3^2$
(11)	$L_1^2 \rightarrow L_2^4 \rightarrow L_3^5$ $L_1^1 \rightarrow L_2^{18} \rightarrow L_3^{k=9,15}$ $L_1^6 \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^6 \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$
(12)	$L_1^2 \rightarrow L_2^5 \rightarrow L_3^5$ $L_1^2 \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^2 \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$
(13)	$L_1^0 \rightarrow L_1^3 \rightarrow L_2^1 \rightarrow L_3^2$
(14)	$L_1^0 \rightarrow L_1^3 \rightarrow L_2^2 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^1 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^6 \rightarrow L_3^{k=9,15}$
(15)	$L_1^0 \rightarrow L_1^4 \rightarrow L_2^3 \rightarrow L_3^2$
(16)	$L_1^0 \rightarrow L_1^4 \rightarrow L_2^4 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$
(17)	$L_1^0 \rightarrow L_1^4 \rightarrow L_2^5 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$
(18)	$L_1^0 \rightarrow L_1^5 \rightarrow L_2^3 \rightarrow L_3^2$
(19)	$L_1^0 \rightarrow L_1^5 \rightarrow L_2^4 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{18} \rightarrow L_3^{k=9,15}$
(20)	$L_1^0 \rightarrow L_1^5 \rightarrow L_2^5 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^5 \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^5 \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$

APPENDIX 8 – Algorithm Structure of Event D

Table A8.1. Algorithm structure of event D

$A_D^1(I^{M=1})$	$A_D^2(I^{M=2})$	$A_D^3(I^{M=3})$
(1) L_1^2	(3) $L_1^1 \rightarrow L_2^2$	(8) $L_1^1 \rightarrow L_2^1 \rightarrow L_3^2$
(2) $L_1^0 \rightarrow L_1^5$ $L_1^0 \rightarrow L_1^{18}$ $L_1^0 \rightarrow L_1^9$ $L_1^0 \rightarrow L_1^{15}$	(4) $L_1^2 \rightarrow L_2^5$ $L_1^6 \rightarrow L_2^{9,15}$ $L_1^1 \rightarrow L_2^{18}$	(9) $L_1^1 \rightarrow L_2^2 \rightarrow L_3^5$ $L_1^1 \rightarrow L_2^1 \rightarrow L_3^{18}$ $L_1^1 \rightarrow L_2^6 \rightarrow L_3^{k=9,15}$
	(5) $L_1^0 \rightarrow L_1^3 \rightarrow L_2^2$	(10) $L_1^2 \rightarrow L_2^3 \rightarrow L_3^2$
	(6) $L_1^0 \rightarrow L_1^4 \rightarrow L_2^5$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^{k=9,15}$	(11) $L_1^2 \rightarrow L_2^4 \rightarrow L_3^5$ $L_1^1 \rightarrow L_2^{18} \rightarrow L_3^{k=9,15}$ $L_1^6 \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^6 \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$
	(7) $L_1^0 \rightarrow L_1^5 \rightarrow L_2^5$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{18}$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^{k=9,15}$	(12) $L_1^2 \rightarrow L_2^5 \rightarrow L_3^5$ $L_1^2 \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^2 \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$
		(13) $L_1^0 \rightarrow L_1^3 \rightarrow L_2^1 \rightarrow L_3^2$
		(14) $L_1^0 \rightarrow L_1^3 \rightarrow L_2^2 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^1 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^6 \rightarrow L_3^{k=9,15}$
		(15) $L_1^0 \rightarrow L_1^4 \rightarrow L_2^3 \rightarrow L_3^2$
		(16) $L_1^0 \rightarrow L_1^4 \rightarrow L_2^4 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$
		(17) $L_1^0 \rightarrow L_1^4 \rightarrow L_2^5 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^9 \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^{15} \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$
		(18) $L_1^0 \rightarrow L_1^5 \rightarrow L_2^3 \rightarrow L_3^2$
		(19) $L_1^0 \rightarrow L_1^5 \rightarrow L_2^4 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^9 \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^7 \rightarrow L_2^{15} \rightarrow L_3^{k=9,15}$ $L_1^0 \rightarrow L_1^3 \rightarrow L_2^{18} \rightarrow L_3^{k=9,15}$
		(20) $L_1^0 \rightarrow L_1^5 \rightarrow L_2^5 \rightarrow L_3^5$ $L_1^0 \rightarrow L_1^5 \rightarrow L_2^3 \rightarrow L_3^{18}$ $L_1^0 \rightarrow L_1^5 \rightarrow L_2^7 \rightarrow L_3^{k=9,15}$