

MANUFACTURING PLANNING AND SHOP FLOOR
CONGESTION ANALYSIS IN MULTI-PRODUCT NETWORKS
USING A DATA-DRIVEN APPROACH

by

Rayan Saleem M Omar

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
May 2017

© Copyright by Rayan Saleem M Omar, 2017

*This work is dedicated to my parents, wife, daughter and son, and
parents-in-law.*

Table of Contents

List of Tables	vi
List of Figures	viii
Abstract	xi
List of Abbreviations Used	xii
Acknowledgements	xiv
Chapter 1: Introduction	1
1.1 Problem Statement and Thesis Objective	3
1.2 Thesis Framework	4
Chapter 2: Literature Review	6
2.1 Production Planning	9
2.1.1 The Iterative Concept	15
2.1.2 The Clearing Function	16
2.1.3 Integrating the CF with the Production Planning Formulation	21
2.1.4 CONWIP	29
2.2 CF Linearization	30
2.2.1 Fuzzy Clustering	30
2.2.2 Fixed Points Set	31
2.2.3 Piecewise Linearization	31
2.3 Summary of the Literature	32
Chapter 3: A Data Driven Approach to Estimate the Clearing Function for Multi-Product Production Network Planning Based on Mean-Value Analysis	34
3.1 Planning Model for Aggregate Production Planning	35
3.2 Bottleneck Machine CF Approach	36
3.3 MVA Algorithm Development for the Integrated CF Approach	37
3.4 Numerical Example	38
3.4.1 Analysis Using the $M/M/1$ Bottleneck Machine CF Approach	40
3.4.2 Analysis Using the MVA CF Approach	46

3.4.3	Comparison of Results	49
3.5	Conclusions and Future Work	53
Chapter 4:	Data Driven Aggregate Production Planning Model based on Queuing Network WIP/Throughput Estimates . . .	54
4.1	Fixed Points Release (FPR) Model	55
4.2	Illustrative Example of the Data-Driven CF Aggregate Planning Model	58
4.3	Results and Sensitivity Analysis	62
4.3.1	Effect of Introducing the CF on the Model	62
4.3.2	Increasing Backorder Costs	63
4.3.3	Effect of Increasing and Decreasing Demand	64
4.3.4	Effect of Step Size and the Planning Horizon Length	66
4.4	Supplementary Constraints and Variables to Decrease Computation Time	69
4.5	Conclusion	71
Chapter 5:	Cubic Approximation (CA) Approach	72
5.1	CA Model	72
5.2	Numerical Example	75
5.3	Effect of Step Size and the Planning Horizon Length	80
5.4	Supplementary Constraints and Variables to Decrease Computation Time	81
5.5	Comparing CA with the FPR Approach	81
5.6	Conclusion	83
Chapter 6:	Comparative Analysis of the ACF, the FPR and the CA Approaches	92
6.1	Numerical Implementation of ACF	94
6.1.1	Kacar and Uzsoy [37] ACF Model Formulation	101
6.2	Fixed Points Release Model Parameters	103
6.3	Cubic Approximation Approach Parameters	106
6.4	Numerical Experiments	106
6.4.1	Correlated Demand	106

6.4.2	Mixed Demand	115
6.4.3	Production Rate Comparison	115
6.4.4	Computational Time	126
6.5	Conclusion	127
Chapter 7:	Conclusion and Future Research Direction	128
7.1	Conclusions	128
7.2	Future Research Directions	131
Bibliography	133
Appendices	140
Appendix A: Chapter 3	141
A.1	Python Code for Computing MVA	141
Appendix B: Chapter 4	147
B.1	Python Code for the Optimization Model	147
B.1.1	Main Code	147
B.1.2	Calling Function for Closed_ISERC	164
B.2	MPL Code	167
Appendix C: Chapter 5	175
C.1	Python Code for Cubic Approximation	175
C.2	MPL Code	184
Appendix D: Chapter 6	191
D.1	FPR Grid for the Open Queuing Network	191
D.2	FPR Grid for the Closed Queuing Network	198

List of Tables

Table 3.1	Machine processing time	39
Table 3.2	Number of visits	40
Table 3.3	Interarrival time	40
Table 3.4	Effect of product 1 arrival rate at station 1 in M/M/1 model	41
Table 3.5	List of the product mix sets	43
Table 3.6	Effective arrival rates for product mix (0.33, 0.67, 0)	45
Table 3.7	Bottleneck machine for each product mix set	46
Table 3.8	Simulation WIP	47
Table 3.9	MVA throughput	48
Table 3.10	WIP analysis	50
Table 3.11	Throughput analysis	51
Table 4.1	Pattern, product mix, WIP, and throughput	59
Table 4.2	Generated demand	59
Table 4.3	FPR model output	61
Table 4.4	Effect of demand changes on objective function	65
Table 4.5	Generalizing the effect of decreasing and increasing demand on objective function and run time	66
Table 4.6	Step size and planning horizon effect on the objective function	67
Table 4.7	Objective function confidence limits	68
Table 4.8	Run time confidence limits	69

Table 5.1	CA throughput, slope, lower corner, and upper corner	77
Table 5.2	Generated demand	78
Table 5.3	CA model output	79
Table 5.4	Step size and planning horizon effect on the objective function	81
Table 6.1	Machine reliability parameters	93
Table 6.2	Processing time	94
Table 6.3	Release combinations, simulation WIP and throughput	104
Table 6.4	Release combinations, simulation WIP, and MVA throughput .	105
Table 6.5	Product WIP in ACF, FPR, and CA	110
Table 6.6	Demand for 70% utilization	118
Table 6.7	Demand for 90% utilization	119
Table 6.8	Demand for Poisson distribution	121
Table 6.9	Average value of the absolute difference between demand and production	125
Table 6.10	Production planning models computational time	126
Table D.1	All release combinations, simulation WIP and throughput . . .	191
Table D.2	All release combinations, simulation WIP, and MVA throughput	198

List of Figures

Figure 2.1	Advanced planning system matrix [49]	6
Figure 2.2	PPC hierarchical planning framework [20]	9
Figure 2.3	Karmarkar's CF [35]	18
Figure 2.4	Piecewise linearization	31
Figure 3.1	Production routings as taken directly from Kayton et al. [36] .	39
Figure 3.2	The CFs of $M/M/1$, MVA, and open queuing network simulation (upper and lower bounds)	42
Figure 3.3	Increasing product 1 while decreasing product 2 WIP level . .	49
Figure 3.4	Increasing product 1 while decreasing product 3 WIP level . .	52
Figure 3.5	Increasing product 2 while decreasing product 3 WIP level . .	52
Figure 4.1	CF effect on product 1	63
Figure 4.2	CF effect on product 2	63
Figure 4.3	CF effect on product 3	64
Figure 4.4	Effect of increasing backorder cost	65
Figure 4.5	Effect of decreasing and increasing the demand on the backorder and inventory level	66
Figure 4.6	Objective function value and the run time	70
Figure 5.1	Cubic representation of product mix with corresponding throughput	73
Figure 5.2	Slope of each products	75

Figure 5.3	Step size accuracy	80
Figure 5.4	Objective function value and the run time	82
Figure 5.5	Aggregate production planning comparison for production . . .	84
Figure 5.6	Aggregate production planning comparison for inventory . . .	85
Figure 5.7	Aggregate production planning comparison for backorder . . .	86
Figure 5.8	Total operational cost comparison	87
Figure 5.9	Computational time comparison for 10 periods	88
Figure 5.10	Computational time comparison for 20 periods	89
Figure 5.11	Computational time comparison for 30 periods	90
Figure 6.1	ACF for machine 1	96
Figure 6.2	ACF for machine 2	96
Figure 6.3	ACF for machine 3	97
Figure 6.4	ACF for machine 4	97
Figure 6.5	ACF for machine 5	98
Figure 6.6	ACF for machine 6	98
Figure 6.7	ACF for machine 7	99
Figure 6.8	ACF for machine 8	99
Figure 6.9	ACF for machine 9	100
Figure 6.10	ACF for machine 10	100
Figure 6.11	ACF for machine 11	101
Figure 6.12	Production rate comparison for 60-20-20 case	108

Figure 6.13	Production planning cost results for 60-20-20 demand	109
Figure 6.14	Product-WIP level for ACF, FPR, and CA	111
Figure 6.15	Production rate comparison for constant demand based on 70% utilization case	112
Figure 6.16	Production rate comparison for constant demand based on 90% utilization case	113
Figure 6.17	Production planning cost results for 70% utilization	114
Figure 6.18	Production planning cost results for 90% utilization	115
Figure 6.19	Production rate comparison for varying demand based on 70% utilization case	116
Figure 6.20	Production rate comparison for varying demand based on 90% utilization case	117
Figure 6.21	Production planning cost for 70% utilization for varying demand	120
Figure 6.22	Production planning cost for 90% utilization for varying demand	120
Figure 6.23	Production rate comparison for Poisson demand case	122
Figure 6.24	Production planning cost results for Poisson distribution	123

Abstract

The clearing function (CF) models the non-linear relationship between work-in-process (WIP) and throughput and has been proposed for production planning in environments with queuing (congestion) effects. However, incorporating the CF concept in multi-product multi-stage manufacturing networks is still a challenging problem. One approach proposed in literature is to model the CF at the bottleneck machines only. However, since the bottleneck may shift as the product mix changes and the queuing network effect is difficult to capture at the bottleneck, this approach has its limitations.

The dominant method in the literature to modelling congestion in multi-product multi-stage networks is the allocated clearing function (ACF) approach. In this approach, the CF is developed at the resource level using a numerical estimation method such as discrete-event simulation. Based on these estimates, the CF is fit using piecewise linear equations. The ACF linear program (LP) then partitions the CF for resource product combinations.

This thesis proposes an alternative methodology to ACF, where the release and WIP levels in each period are discrete (FPR). The CF is estimated at discrete load combinations using simulation, mean-value analysis (MVA), or queuing network analysis. A mixed-integer programming (MIP) formulation is developed to determine optimal material release. The approach is data-driven and does not require the use of a curve-fitting function. The CF estimates at the discrete lattice points in this approach are network based, as opposed to resource based as in ACF. The MIP behaviour is illustrated using the MVA approximation for a well-known multi-product re-entrant semiconductor manufacturing case in the literature.

The FPR-MIP formulation is extended to allow release quantities between the lattice points using a cubic approximation technique (CA). This approach allows the MIP to potentially obtain better solutions in continuous space and can be seen as a generalization of piecewise linearization for the single product case. A detailed comparison between FPR and CA is discussed using an example case.

Finally, FPR and CA are compared and contrasted with the ACF approach.

In summary, this thesis presents an alternative approach to the state-of-art in the modelling of congestion in manufacturing networks.

List of Abbreviations Used

ACF	Allocated Clearing Function
ANOVA	Analysis of Variance
APS	Advance Planning System
ATP	Available to Promise
BOM	Bill of Material
CA	Cubic Approximation
CF	Clearing Function
CONWIP	Constant WIP
CQN	Closed Queuing Network
CTP	Capable to Promise
ERP	Enterprise Resource Planning
FMS	Flexible Manufacturing System
FPR	Fixed Points Release
FPR-C	Fixed Points Release Using Closed Queuing Network
FPR-O	Fixed Points Release Using Open Queuing Network
LCL	Lower Confidence Limits
LP	Linear Programming
MDCF	Multi-Dimensional Clearing Functions
MILP	Mixed-Integer Linear Programming
MIP	Mixed Integer Programming
MPS	Master Production Schedule
MRP	Material Requirements Planning
MTBF	Mean Time Between Failures
MTO	Make to Order
MTS	Make to Stock
MTTR	Mean Time to Repair

MVA	Mean Value Analysis
MVAQ	Mean Value Analysis for Queues
PCF	Partitioned Clearing Function
PPC	Production Planning and Control
RCCP	Rough-Cut Capacity Planning
RN	Random Number
SCV	Squared Coefficients of Variation
UCL	Upper Confidence Limits
WIP	Work in Process

Acknowledgements

First, all praises and thanks are due to Allah (God), through whom all was made possible. Then, I would like to express my deepest appreciation to my supervisor, Dr. Uday Venkatadri, who has supported me throughout this work with his knowledge and patience. I am very indebted to him for all the guidance and encouragement that he gave me even during his private time. Also, I wish to acknowledge Dr. Claver Diallo, for his helpful advice and comments to improve this work.

I am indebted and grateful to my lovely wife, Mrs. M. Sabr, for her endless sacrifice and patience to stay beside me through all these years. I thank her for being an understanding and wonderful person. My thanks are extended to my daughter (Fatima) and son (Omar) for their patience.

Special thanks are due to my father Mr. Saleem Omar and mother Mrs. R. Tawfiq, father-in-law Mr. Sabri Sabr and mother-in-law Mrs. K. Salamah, for their sacrifice and unconditional support and continuous asking of Allah to help me in this study; my brother and sister, brothers-in-law and sisters-in-law for their love, enthusiasm and warm support.

Finally, I am grateful to Taibah University (located in Madinah, Saudi Arabia) for the financial support they provided to complete this work.

Rayan Saleem M Omar

Chapter 1

Introduction

Globalization has created a competitive dynamic market-place that has increased the demand on manufacturing companies to be responsive to their customers. Responsiveness includes the ability to manufacture on short notice and to customize products for individual customers. To retain their market share and gain a competitive edge, companies recognize the need to place a greater emphasis on cooperation and coordination among supply chain parties. This can be accomplished by applying production planning and control (PPC) concepts, which link the activities of each node in the supply chain with the rest of the chain.

PPC is a massive and complex task to achieve. Therefore, it is done in the different departments of each node in the supply chain, adding a coordination dimension, whereby nodes across the chain need to be connected. The coordination task is ensured by defining a higher managerial planning level within the PPC framework. Naturally, the aim of planning is to not only plan resources efficiently and responsively, but also to minimize conflicts between the production plans at the lower levels.

It is widely accepted that there are two levels of production planning: aggregate at the higher managerial planning level and disaggregate (detailed) at the shop floor or lower managerial planning level. The higher planning managerial level assigns available shop floor capacity (e.g. resource, material, personnel, etc.) to different products according to anticipated customer demand. Then, the master-production schedule (MPS) is generated to specify the order release size and timing to cater to customer demand. Finally, the lower managerial planning level uses material requirements planning (MRP) to translate the MPS into a detailed production plan of the required raw materials based on the product bill of material (BOM). Therefore, ignoring the effect of congestion at the higher planning level can make the disaggregate production plan infeasible and in such situations, shop floor managers are left with difficult choices such as under production or overtime staffing.

The essential coordination factor between aggregate planning, MPS, and MRP levels is the material lead time. Material lead time or flow time is the total time that the material spend in the shop floor to be processed, including queuing, batching, and processing, to constitute the finished product [25].

The popular production-planning tool MRP, which is used at the lower planning level, assumes that the lead-time is fixed and not load dependent. In addition, many of the classical formulations for production planning assume fixed lead times. Unfortunately, this assumption can lead to an overestimation of resource capacity leading to an infeasible production plan. Planning circularity is the problem where lead time is dependent on resource load and resource loading depends on the lead time. Asmundsson, Rardin, and Uzsoy [4] state that “many production systems, especially wafer fabs that must operate at high utilization for economic reasons, operate under regimes that cause them to have substantial lead times, which must be considered for a planning system to match supply to demand. However, per queuing models, lead time depends on the resource utilization in the system, which in turn is determined by the assignment of work to resources by the planning models. This constitutes a fundamental circularity in planning and a significant difficulty in this field for decades.” One work around is the linear clearing function introduced by Graves [19] which assumes that the system will be managed in a way that keeps the average cycle time constant, while the MRP style fixed lead time represents a time-shift of the input. This is illustrated in Orcun, Uzsoy, and Kempf [57].

Queuing theory has been used in the literature to capture the relationship between WIP level and resource throughput which takes the form of a non-decreasing concave function, termed the CF [35, 4]. The CF models the effect of the congestion on the aggregate plan and permits the production planner to decide on shop floor release quantities at the higher planning level. The CF concept can greatly enhance the feasibility of an aggregate production planning solution since it incorporates the congestion effect. Since the CF is non-linear, the literature presents several ways to linearize the CF and integrate it within the aggregate production planning optimization model.

Recently, the production planning literature has shown an increasing interest in deriving the CF either by using analytical [47] or simulation [3] approaches. The

analytical approach relies heavily upon deriving the CF for the bottleneck resource. Unfortunately, in the case of re-entrant manufacturing environments with multiple products, the bottleneck shifts with product mix, which varies from one period to another depending on the recommendation of the planning run (usually a linear program). Moreover, the analytical approach is feasible for only relatively small systems. The simulation approach suffers from the shortcoming that calibrating the CF numerically using simulation is often time-consuming.

1.1 Problem Statement and Thesis Objective

The top level planning system in traditional MPS does not integrate the congestion effect of queuing in manufacturing systems. Consequently, firms often fail to fulfill customer orders within the required time frame, which could lead to customer dissatisfaction and migration. Therefore, it is important for firms to have accurate information about shop-floor resource capacity availability encapsulating the congestion effect while generating production planning.

The main objective of this thesis is to contribute to the existing research in aggregate production planning dealing with congestion. One of the consequences of MRP is the tendency to overload the shop floor, which in turn causes system congestion and increases the cost of WIP, with marginal gain in productive capacity. Therefore, the supply chain's current and future capacity availability should be calculated with respect to shop floor load capacity. This implies that the BOM based MRP system should be setup for load-dependent lead time information. In order to achieve this objective, we propose constructing an aggregate production planning model for multi-product network manufacturing with load-dependent lead-times (ideally, the MRP system would also use load-dependent lead-times).

In particular, the following questions are relevant to the thesis:

1. What is the impact of product mix on the location of the network bottleneck machine?
2. What is the disadvantage of deriving the CF for the bottleneck machine in the case of multi-product networks?

3. How can the CF concept be introduced in queuing network models for complex manufacturing environments?
4. What are the computational implications of using the queuing network approach to estimate congestion effects in aggregate production planning?
5. How does the proposed approach compare with other approaches in the literature in dealing with congestion in multi-product network manufacturing?

1.2 Thesis Framework

In this thesis, we show the deficiency of using the CF only for the bottleneck machine for multi-product networks by using two alternative methods of comparison: simulation for the open queuing network case and MVA for the closed queuing network case. In practical terms, we look at how the congestion effect of a manufacturing facility can be introduced to aggregate production planning for closed queuing networks using MVA [73]. Then we look at the different approaches to incorporate the CF into an optimization model for aggregate production. The first approach estimates the multiple products CF at discrete points to avoid linearization complexity, as proposed in [54]. The second approach proposes a linearization methodology using the cubic approximation method. The remainder of this thesis is organized as follows:

- Chapter 2 provides a literature review of the relevant areas.
- Chapter 3 describes the methodological development of a data-driven approach to estimate the clearing function for a multi-product production network based on mean-value analysis.
- Chapter 4 presents an aggregate production planning formulation with discrete release choices.
- Chapter 5 illustrates a cubic approximation approach to linearize the multi-product CF that provides continuous space release options.
- Chapter 6 presents a comparison between the models in Chapter 4 and 5 with the ACF approach of [3].

- Chapter 7 concludes the thesis and outlines directions for future research.

Chapter 2

Literature Review

Supply chain management is the coordination of information and goods flowing through the nodes of the chain, from the suppliers, to the manufacturer, sometimes to the warehouse, and finally to the customers [66].

Since customers are the main drivers of supply chain operations, supply chain activities have to be designed to meet their needs. The advanced planning system (APS) can accomplish this by bridging the gap of hierarchical planning decisions made by various organizational functions such as procurement, production, distribution and sales departments [49]. Figure 2.1 illustrates APS, i.e., advanced planning and scheduling. APS modules are offered in most enterprise resource planning (ERP) packages.

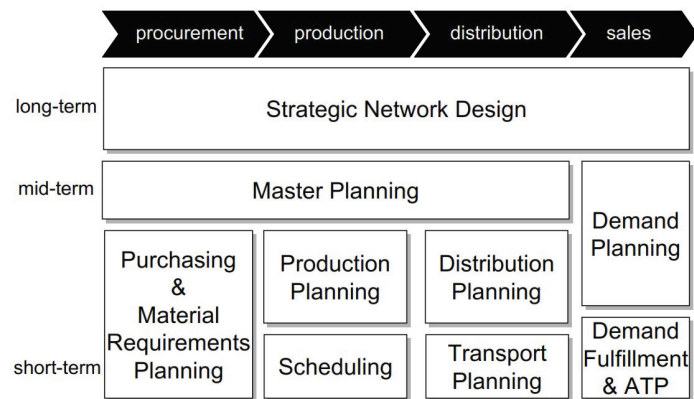


Figure 2.1: Advanced planning system matrix [49]

The vertical axis of Figure 2.1 shows the different hierarchical planning tasks in a supply chain. At the top are the long-term tasks which can be described as the strategic upper management level. At this level, decisions are made on the supply chain network configuration including the products to offer, location of nodes, choice of modes of transportation, and how to serve markets. Next is the master planning

tier, which is usually generated at the middle management level and is termed mid-term planning. Lastly, the short-term planning tier is often generated daily or weekly, and this set of decisions are at the lower management level.

To enhance an understanding of the big picture of supply chain management tools, we briefly discuss the topics shown above in the APS matrix, as they have a direct relationship to this research.

Strategic network design: This sets the long-term plan ranging from three or five to ten or 15 years, depending on industry. The supply chain configuration is determined at this stage. This level includes identifying the market needs and calibrating production processes and raw material sources. It also includes determining the best geographical location for facilities (plants, distribution centres, warehouses, etc.) to serve customer demand, and determining realistic capacity estimates to help the firm attain its strategic goals.

Master Planning: During this stage, plans are made based on the decisions of the upper strategic network design tier. The differences between these successive planning stages are the levels of aggregation. In the master planning stage, a detailed plan of the flow of goods throughout the entire supply chain is developed. Moreover, supply is matched with demand by allocating capacity (which is a decision of the strategic network design) to maintain efficient production, transportation, and capacity utilization for the raw material and seasonal stock-piles.

Purchasing and material requirements planning: At this stage, there is a disaggregation of the product families in the master plan, which enables MRP. The inputs to MRP are the BOM and the lead times for each item in the BOM. From this, the MRP run is able to develop a time-phased manufacturing and procurement plan for each item by offsetting the lead-time of the raw materials and subassemblies required for production. However, MRP does not consider production resource capacity or the shop floor congestion effect, and may result in infeasible production plans. While Rough-cut Capacity Planning (RCCP) is part of the MRP framework, the resource utilization calculation for RCCP is aggregate. Since the MRP is one of the pillars of this study, we elaborate on it

later in the thesis.

Production planning and scheduling: In this stage, the inputs are the planned targets in response to the customer order due-dates and customer demands forecast in the master plan. The outputs of this stage are the production plans and schedules including release, setup, and order sequencing. Production planning and scheduling usually covers the period of the demand forecasting horizon that can range from a day to a month. Planning is often done at a weekly or monthly level, while scheduling might take place at the hourly or daily level, depending on industry.

Distribution and transportation planning: Distribution systems link the nodes of the chain and in distribution planning, an attempt is made to reduce distribution costs through shipment planning and consolidation. In transportation planning, an attempt is made to identify truck loads that minimize traveling distance to the different customer locations considering delivery time.

Demand planning, demand fulfillment, and available-to-promise (ATP): The main driver of supply chain management is customer demand, which could be unknown or known. There are two main classifications of manufacturing systems based on demand. The first type is the make-to-stock (MTS) manufacturing environment, which deals with unknown demand by forecasting future customer needs and fulfillment through demand planning. This type of system is the push system. The second type of manufacturing environment is make-to-order (MTO), which deals with known demand through confirmed customer orders. This type of system is the pull system. Many systems are hybrid, and they have characteristics of both push and pull systems. The planning process of fulfilling customer orders is referred to as demand fulfillment. In ATP, this is done by assigning the projected inventory stockpiles to customer orders. In capable-to-promise (CTP), the demand fulfillment process commits capacity within a production plan to an individual order. In many cases, a part of a customer order can be fulfilled using ATP while the rest is fulfilled using CTP.

The themes discussed in this thesis include production planning, material requirements planning, and congestion in multi-stage production. We classify the literature

based on these topics.

2.1 Production Planning

PPC system links a firm's production objectives with the capacity of its available resources. A firm's production objective could manifest itself as a fill rate from customer demands. PPC synchronizes the flow of planning information throughout the firm's hierarchical planning levels, usually three. The top level is aggregate planning, where gross customer demand is matched with existing supply. In the second planning level, the MPS specifies the order release size and timing to cater to customer demand. Finally, the MRP translates the former plans into a detailed production plan of the required raw materials. Figure 2.2 illustrates a framework for PPC hierarchical planning.

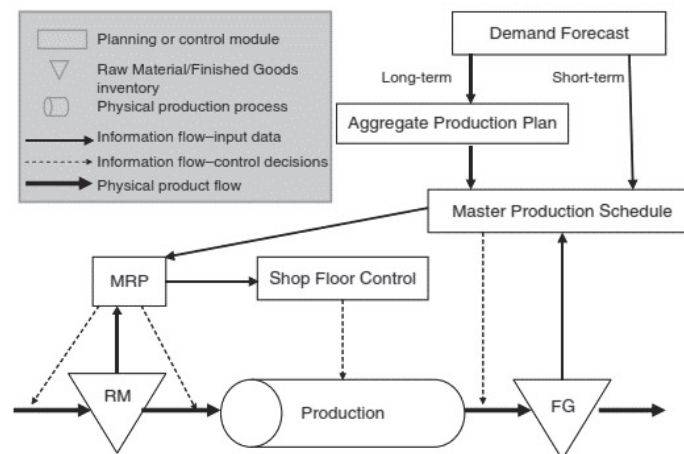


Figure 2.2: PPC hierarchical planning framework [20]

A mathematical representation of a manufacturing facility was first used in the literature to outline the relationship between a firm's available resources and the anticipated customer demand in the early work of Modigliani and Hohn [53]. Their model was formulated and solved by Manne [46] using an LP formulation equivalent to the transportation model. The main factor that links the top aggregate planning with the lower shop floor planning is the lead-time. Therefore, Hackman and Leachman [21] proposed a general production planning formulation based on the linear programming framework, where they address the lead-time as an exogenous parameter to the production resource load.

Kacar, Mönch and Uzsoy [31] compare the integer fixed lead time estimation method used in the formulation of Hackman and Leachman [21], a non-integer fixed lead time estimation, and the ACF formulation of Asmundsson, Rardin, Turkseven, and Uzsoy [3]. They conclude that the non-integer fixed lead time is superior to the integer fixed lead time and gives comparable results to the ACF. Therefore, they suggest considering the non-integer fixed lead time in the case where variability is low. This is to avoid the complexity of deriving and fitting the CF. The ACF performs better in the case of high variability.

A firm faces uncertainty (or variability) due to external or internal factors. External (or environmental) uncertainty is due to those factors which affect a firm's released production planning feasibility from outside its boundaries, such as unreliable suppliers and demand uncertainty. Conversely, internal (or operational) uncertainty involves factors such as machine failure, defective parts, and setup delays. These uncertainties have the same effects on production planning feasibility as external ones, but they come from within the manufacturing environment.

Koh and Simpson [44] studied the agility and responsiveness of ERP to uncertainty and system changes in small and medium enterprises. Their study analyzed 108 enterprises using analysis of variance (ANOVA) for different manufacturing environments. At the end of their analysis, they delineate 42 uncertainty factors that could cause late delivery. These factors are categorized into uncertainties in demand, suppliers, and operations.

As previously mentioned, uncertainty not only affects operational planning but also the planning and scheduling decisions of the top-level as well. Graves [20] investigated production planning uncertainty coming from the demand forecast, the external supply process, and the internal supply process. Graves also described several approaches for alleviating the effects of the above-mentioned uncertainties such as safety stock, frequent re-planning, inflating lead-time, using flexible capacity, and backlog management. However, because some of the aforementioned tactics are ad hoc approaches to mitigating the effect of external factors, they may cause an adverse effect on shop floor processes. This is due to inherent conflict between decisions made by the top and lower management levels.

Operational uncertainty can dramatically disrupt the due-date delivery schedule.

For that reason, we survey the literature concerning internal factors.

Roberts and Barrar [63] and Koh, Jones, Saad, Arunachalam, and Gunasekaran [42] both point out a key factor that can affect the performance of MRP, namely that it is heavily reliant on the working environment being deterministic. MRP is generated in response to the demand forecast of the top planning level, i.e., the MPS and aggregate production plan. The demand forecast always needs minor or even major modification as one gets closer to the execution date of the generated plans. Hence, there is a constant need to re-plan and re-schedule MRP plans in order to maintain a high service level.

Yeung, Wong, and Ma [79] provided an exhaustive review of factors that can affect the performance of MRP. They grouped these factors into six categories: demand forecast accuracy; product structure; safety stock; re-planning frequency of the production plans; length of the frozen period; and length of planning horizon.

Demand forecast accuracy: MTS manufacturing environments allocate most of the available capacity to forecasts of market demand. The accuracy of the demand forecast is a key factor in the stability of MRP. An unreliable demand forecast is likely to increase production costs because of planners in such a situation respond by either increasing inventory levels, increasing capacity in regular time, or by scheduling overtime. Sridharan and LaForge [70] used a simulation model to study the effects of integrating the variability of the demand forecast on the stability of production plans using safety stock. They concluded that while safety stock does enhance the stability of MPS and MRP, the buffer quantity should be small; otherwise, it could lead to schedule instability.

De Bodt and Wassenhove [7] examined the influence that forecast error has on the order quantity rule. Their simulation model shows how a small variation in the forecast will significantly affect the cost efficiency of the order quantity rule. Wemmerlöv [77] concluded that forecasting errors will drastically increase inventory levels and operational costs, which then also increase order costs. This is in alignment with the conclusion of Fildes and Kingsman [15] that there is an inverse relationship between forecast accuracy and inventory cost. Specifically, a high level of forecast accuracy reduces inventory costs and leads to stable production plans.

Product structure: BOM flexibility and component commonality improve PPC stability. Meixell [48] created a simulation model to examine the impact of product flexibility, capacity, and set-up costs on MRP stability in a precarious demand environment. The results demonstrated that designing component commonality into the products structure not only augments the stability of the manufacturer but also of the whole supply chain.

Lin, Krajewski, Leong, and Benton [45] studied the effect of BOM structure, cost structure, and other environmental factors on MPS-related decisions such as re-planning frequency and frozen period. The results showed that BOM structure and cost structure significantly impact MPS design.

Safety stock: Sustaining high service levels requires MTS to use safety stocks to hedge against demand and supplier uncertainty. However, excessive buffer quantity adversely influences the stability of the MRP [70]. Dolgui and Prodhon [14] provided a comprehensive review of MRP uncertainty in a supply chain environment. They suggested that safety stock strongly induces an increase in the production cost if it is not correctly set. Ho, Law, and Rampal [22] presented a framework for studying and classifying uncertainty related to MRP as well as provided an overview of how such a system's behavior could be dampened. Ho and co-authors concluded that investments in safety stock should be applied after a full analysis of the trade-off between costs and benefits.

Grasso and Taylor III [18] suggested using safety stock rather than safety lead-time to alleviate supplier delivery timing uncertainty. They used a multi-level manufacturing and product structure simulation model to examine the effects of four operating policies on the uncertainty of supplier delivery timing. These policies are: safety stock quantity and safety lead-time; lateness penalty and holding cost; lot-size rule; and amount of lead-time variability. Similarly, Whybark and Williams [78] conducted a simulation to study the uncertainty effect on customer satisfaction and inventory investment. They concluded that safety stock should be used for quantity uncertainty, while safety lead-time is favourable for timing uncertainty. Pujawan [61] concluded that companies should invest in safety stock to mitigate forecast error and oscillating demand.

Re-planning frequency of the production plan: It is common practice in MTS to track demand changes by frequently re-planning and re-scheduling the released PPC plans in the hopes of getting higher customer satisfaction. Chung and Krajewski [12] recommended frequent re-planning for firms with high set-up and work force costs. A rolling horizon aggregate plan and MPS were employed to examine the influence of re-planning frequency on the production plans cost structure in case of deterministic demand.

However, Sridharan, Berry, and Udayabhanu [68] concluded that more frequently re-planning the MPS can lead to adverse effects on the stability of production operations. They examined the impact of re-planning frequency on a single-level supply chain environment. Ganeshan, Boone, and Stenger [16] conducted a simulation to study the influence of three inventory planning factors (re-planning frequency, forecast error, and information sharing) among supply chain echelons. They revealed that high re-planning frequencies and unreliable forecasting could give rise to increased product cycle time, which would decrease customer satisfaction. Omar and Bennell [55] also concluded that the stability of MPS is affected by re-planning frequencies.

Length of the frozen period: MTS regularly use the strategy of freezing part of the production plan in order to diminish uncertainty from re-planning. Although this strategy enhances the stability of the production plan, it may decrease customer satisfaction if the plan is unable to meet new or revised customer demand. Furthermore, firms may incur high production and inventory costs as a result of freezing.

Sridharan, Berry, and Udayabhanu [69] constructed a simulation study to ascertain the influence of freezing interval length and the freezing method on operation and inventory costs. The results showed that freezing half of the planning horizon has a small effect on cost error. Cost error is the incremental cost caused by excessive production set-ups and holding inventory. Zhao and Lee [80] suggested increasing the interval of frozen periods in cases of deterministic demand, as this will improve the stability of production planning and reduce the total operational cost. On the other hand, stochastic demand conditions

require a short frozen period to maintain low operational costs and low production planning instability. In the paper, a multi-level MRP production system was simulated to understand the effect of the freezing interval proportion and methods on MPS stability and total operational cost.

Length of the planning horizon: The length of the planning horizon significantly affects MTS. Generally, a prolonged planning horizon is favorable in deterministic demand conditions. However, in the case of demand uncertainty, a shortened planning horizon should be considered, as otherwise the total production cost could increase [80].

Carlson, Beckman, and Kropp [10] indicated that the planning horizon should be as long as the order cycle, and that the demand forecast should be examined to prolong the planning horizon. Moreover, the planning horizon should be extended in instances where the planning horizon is less than the order cycle. Conversely, under demand uncertainty, this will increase inventory costs and make order cycle predictions difficult.

Recently, Sahin, Narayanan, and Robinson [64] surveyed the literature on rolling horizon planning in the supply chain context. This review looks at demand variability (deterministic vs. stochastic) and levels in planning (single vs. multiple). They state that the planning horizon affects cost and stability in all planning situations. They also conclude that the rolling horizon planning is relevant to virtually all planning systems where demand is stochastic in the long-term but there is a need to establish short-term replenishment schedules.

In addition to the uncertainty factors mentioned above, conventional MRP is constructed assuming a fixed lead-time, which in turn implies infinite production capacity. From queuing theory, there is a non-linear relationship between lead-time and utilization, which is affected by the capacity loading decision. Despite the significant drawback of the capacity loading decision on system performance, it was not captured in the MRP [42, 59]. Thus, based on what already mentioned about the uncertainty sources and their impact on the released plans, variability is an integral part of the production planning.

The majority of the research in the literature on production planning does not

consider the effect of the shop floor load on the system throughput rate, as the shop floor load affecting the resource utilization too [59, 52]. This is an important research gap since these factors affect the feasibility of the released production plans, as the production lead time may increase along with heavily load the shop floor, which in turn increases production costs dramatically [20, 5, 42, 63].

To overcome this shortcoming of the MRP, three approaches have been highlighted in the literature. The first approach is to use the iterative concept to derive the lead time by imitating the system dynamics using a simulation model. The second approach is to derive the CF by using queuing theory, to capture production resource capacity. The third approach is to use the constant work-in-process (CONWIP) approach of the closed queuing network to avoid excessive WIP levels in the system that could lead to shop floor congestion.

2.1.1 The Iterative Concept

As discussed above, the early modeling of production planning was deficient in capturing shop floor dynamics. Hung and Leachman [26] proposed first using production planning optimization model to generate a release schedule, and then introduce this schedule to the simulation model to collect the flow time, where the shop floor dynamics is embedded. They then use the simulation output to reformulate the optimization model. This iteration of the optimization model and simulation continue until some satisfactory criteria are met. Byrne and Bakir [9] used the same approach to get a real estimation of the shop floor capacity. Kim and Kim [39] combined and further extended the idea of Hung and Leachman [26] and the approach of Byrne and Bakir [9], where they modify the right hand side of the capacity constraint in the Hung and Leachman [26] LP model. This modification is based on the simulation model. The simulation model is also used to directly estimate the loading factors for each machine in each period. Finally, they apply the iteration approach of Byrne and Bakir [9] to find a feasible production plan. This iterative procedure is based on constantly updating the shop floor capacity and the production lead-time.

Irdem, Kacar, and Uzsoy [27] provided an exploratory computational experiment comparing the iterative approaches of Hung and Leachman [26] and Kim and Kim [39],

and concluded that the iterative approach of [39] converges in a few number of iterations. They notice cycling across a few solutions in this methodology. However, they note that the iterative process of [26] does not converge easily. Missbauer and Uzsoy [52] indicate that the convergence of the iterative approach needs further research.

Kacar, Irden, and Uzsoy [28] presented an experimental comparison of the iterative approaches of Hung and Leachman [26] and Kim and Kim [39] with the ACF of Asmundsson, Rardin, Turkseven, and Uzsoy [3]. They point out that ACF outperforms the other approaches by giving a realistic representation of the production planning environment. Their model uses simulation optimization, which is different from the iterated multi-model approach in Hung and Leachman [26].

2.1.2 The Clearing Function

Planning and coordinating material flow throughout the supply chain is a complex task. Therefore, planning tasks are held at two different hierarchical planning levels in a firm. At the higher level, the aggregate plan breaks down the planning horizon into discrete time buckets and assigns the aggregated demand, in the form of time units, to the available capacity. The focus at the lower level is on decisions related to capacity planning and order-release timing. In fact, there is a discrepancy between the plans of the two hierarchical planning levels because the managers at the higher level do not consider uncertainty in their aggregate plan. This uncertainty can be triggered from external and/or internal factors.

As mentioned above, external factors affect the firm's released production planning feasibility. Internal factors such as machine failure, defective parts, and order-release decisions have the same effect on production planning. In fact, order-release decisions can dramatically affect lead-time. Lead-time or cycle time [3] is defined as the time required for the transformation process of raw materials into a product that is ready to be used by customers. This time is measured from the time the work is released into the factory, and consists of queue, production, and batch time.

The lead-time has a direct relationship to product routing, set-up, and WIP levels. These operational factors are dependent on shop floor planning decisions such as lot-size, batch and product mix, which in turn have a significant impact on production

costs.

Therefore, the release to a production resource is related to its utilization using the CF concept. The CF can be derived from the steady-state of queuing system or transient state of a production resource, or estimated using simulation model for the production system [29]. For a steady state queue, Little's law shows the relationship between WIP and throughput. In other words, for a given resource, it provides the output based on input WIP. Though the steady state queue is often used to illustrate the CF concept, Missbauer [51] and Selçuk et al. [65] both derive transient clearing functions that require neither steady-state nor Little's Law.

For the $M/M/1$ single product single machine case, the CF can be easily derived. However, it gets more complicated for both multiple products and a network of machines. Another way of looking at the CF is as a means to achieve consistency between planning at the aggregate and shop floor levels.

The first attempt to derive the CF was done by Graves [19], who assumed that the production system could be managed by speeding up or slowing down the production process to keep the average cycle time constant. This is done using a linear CF that relates the throughput rate X of a work center i to a constant portion α of the WIP level W at the beginning of the production period (t), as follows:

$$X_{i,t} = \alpha_i W_{i,t} \quad (2.1)$$

The Graves' linear CF, which represents a managerial policy, implies a fixed lead-time, which is independent of the WIP level at that period and leads to unlimited resource capacity. This assumption is different from the one considered in the most popular production-planning tool, viz., the MRP, as shown in Figure 2.3. MRP represents the fixed lead-time as a time-shift of the input. Thus, $X_t = R_{t-L}$, where X_t is the output of a resource in period t , R_{t-L} is the release in period $t - L$, and L is the exogenous lead-time [57]

Karmarkar [35] re-defined the CF by relating the congestion at the shop floor for a single resource to the resource output X_t , using a nonlinear concave non-decreasing function, as follows:

$$X_t = \min \left[\frac{C_t(W_{t-1} + R_t)}{W_{t-1} + R_t + k}; W_{t-1} + R_t \right] \quad (2.2)$$

Where: C_t and R_t are the maximum production capacity and the work release in period t , respectively, W_{t-1} is the WIP level at the end of period $t - 1$, and k is the curvature shape parameter. Figure 2.3 presents the CFs derived by Graves [19] and Karmarkar [35], represented by a constant proportion line and an effective curve line, respectively.

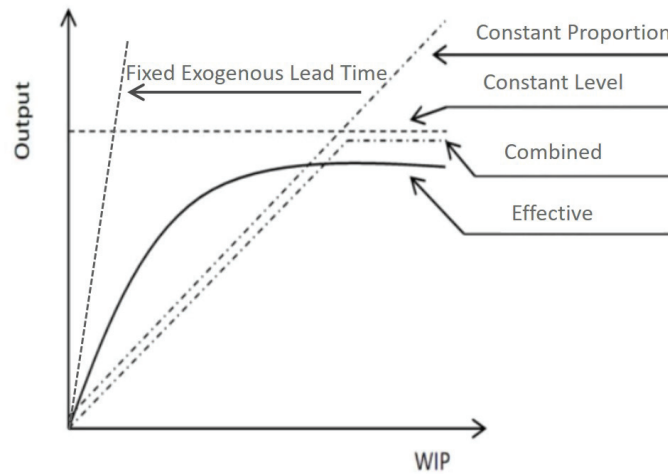


Figure 2.3: Karmarkar's CF [35]

Due to the importance of integrating the CF in the aggregate plan, there is emerging research in the literature dealing with this topic, as will be seen through the rest of this chapter. Spearman [67] derived the CF for a closed production system with high processing time and failure rate. Hopp and Spearman [23] derived a number of CFs to relate the WIP level to the practical worst-case analysis of throughput. Srinivasan, Carey, and Morton [71] constructed a non-linear CF which gives a good estimation for the due-date, since it reflects the continuous lead-time changes captured at each workstation.

Missbauer [50] constructed a continuous time based optimization model to balance order release with capacity. He proposed deriving the CF for $M/G/1$ steady-state system by relating the relationship between the expected workload $E[L(t)]$ and the

expected output $E[X(t)]$ as follows:

$$E[X(t)] = \left[C + k + E[L(t)] - (C^2 + 2Ck + k^2 - 2CE[L(t)] + 2kE[L(t)] + E[L(t)]^2)^{1/2} \right] / 2 \quad (2.3)$$

Where:

$$k = \frac{\mu\sigma^2}{2} + \frac{1}{2\mu} \quad (2.4)$$

Here, σ^2 and μ^{-1} are the variance and mean of the processing time of a resource with the maximum capacity of C in hours of work per period. The parameter k represents the curvature shape.

Medhi [47] provides an equation to measure the expected number of entities in a system characterized by the $G/G/1$ queuing behavior. This is the well-known Kingman approximation for the expected time in the system for a $G/G/1$ queue [41, 24]:

$$W = \frac{c_a^2 + c_s^2}{2} \frac{\rho^2}{1 - \rho} + \rho \quad (2.5)$$

Where: W is the queue length, c_a^2 , c_s^2 are the squared coefficients of variation (SCV) of the arrival and service time, respectively, and ρ is the resource utilization.

Because utilization in this expression is a surrogate for throughput, Asmundsson, Rardin, Turkseven, and Uzsoy [3] solved Equation 2.5 for the positive root of utilization, yielding the following CF:

$$\rho = \frac{W + 1(W^2 + (2c^2W) + 1)^{\frac{1}{2}}}{1 - c^2} \quad (2.6)$$

Where:

$$c^2 = \frac{c_a^2 + c_s^2}{2} \quad (2.7)$$

Kefeli, Uzsoy, Fathi, and Kay [38] compared the CF-based production planning of Karmarkar [35] with the classical linear programming model based on dual prices. They concluded that the classical linear programming model underestimates the capacity constraint dual price since it fails to capture the congestion effect. On the other hand, the non-linear model shows a positive shadow price as resource utilization approaches but is still less than one. Therefore, this approach provides a more

accurate representation of the system and implies that carrying inventory may be necessary even if recourse utilization less than one. The authors pointed out that the approach is difficult to apply in a multi-product and multi-stage manufacturing environment.

Kim and Uzsoy [40] introduced the effect of engineering process improvement stage into the production planning generated for a single resource re-entrant manufacturing environment. The CF concept employed to allocate capacity between the production stage and the engineering improvement stage. The marginal costs are analyzed using the Karush-Kuhn-Tucker (KKT) optimality conditions to derive insights on the effective management of resources.

For the case of multiple products, Asmundsson, Rardin, Turkseven, and Uzsoy [3] derived the ACF to allocate a portion of the resource capacity to produce different products according to customer demand. This was done with the aim of introducing the effect of accumulating different product WIP levels to estimate resource utilization for a time period in aggregate planning. Therefore, the capacity allocated to each resource in the aggregate planning model is segregated by product. Their model can handle CFs at multiple resources. In this approach, clearing functions are developed for each station and included in a planning LP independently, though they are a reflection of the effect of correlated flows between stations through a queuing network. This can be seen in [3] as well as [28, 29, 33] where open queuing network simulation is used to build the piecewise linear clearing functions at each workstation for the ACF model.

Orcun, Uzsoy, and Kempf [58] extended the ACF for stochastic demand for the single product case. The safety stock is integrated into the CF to hedge for demand uncertainty. Kacar, Mönch, and Uzsoy [30] empirically derived the CF for two products requiring hundreds of operations using multiple runs of a simulation model of the system. After that, the CF is derived by fitting WIP to the estimated simulation throughput in order to establish a release schedule for both products in each time period. Finally, the derived CF is integrated to the ACF production planning formulation (presented in the next subsection).

Kacar and Uzsoy [32] presented a new way to fit the CF to a simulation model using linear regression. In addition, they compared the load-based CF with the

product-based CF, where they showed that the load-based CF performs better than the product-based ones in most cases. Kacar and Uzsoy [33] presented an improved way for fitting the CF to empirical data using the simulation optimization tool to optimize the decision variables related to parameterize the CF.

Kefeli and Uzsoy [37] used the ACF of [3] to extend their previous work [38] to identify the bottleneck machine for multiple products multiple machines presented in the well known case study presented in [36].

Albey, Bilge, and Uzsoy [1] constructed a multi-dimensional CF simulation model to show the effect of the product mix on the aggregate plan. They disaggregated the total workload into individual products by using different combinations of product ratios to derive the relationship between utilization and lead-time. Kang, Albey, Hwang, and Uzsoy [34] addressed the effects of lot-size for multiple products produced on a single machine on the deriving the CF analytically, using queuing theory.

The works of Asmundsson, Rardin, Turkseven, and Uzsoy [3] and Albey, Bilge, and Uzsoy [1] emphasize the importance of integrating the product mix into the aggregate capacity planning models; in both cases, the CF is based on an output analysis of a production facility simulation. This approach suffers from two shortcomings. Firstly, simulating a real production facility requires a lot of detailed information about the shop floor performance, which is sometimes hard to get. Secondly, even if it is possible to gather this information, the task is time-consuming.

2.1.3 Integrating the CF with the Production Planning Formulation

In this section, we begin by looking at how the CF may be integrated into the capacity constraint of an aggregate planning formulation. The CFs in Missbauer [50] and Asmundsson, Rardin, Turkseven, and Uzsoy [3] are discussed in detail because they deal with multiple products. The paper by Albey, Bilge, and Uzsoy [1] came later and is also covered.

2.1.3.1 Missbauer's [50] Mathematical Model

The approach of Missbauer [50] in dealing with the order-release problem considers the transient behavior of queuing networks by deriving a CF for the bottleneck machines independently. Non-bottleneck machines are assumed to have a fixed lead-time. A

linear programming model is constructed to optimize the order-release decision in a multi-product situation. The multiple products are aggregated into product families, which require a number of processing operations on different machines.

The CF is derived analytically for the bottleneck machines to reduce the size of the model. Due to this fact, there is a different CF for the different bottleneck machines. The CF is derived considering Poisson arrivals and general processing or service time distribution times for the bottleneck machine $M/G/1$. The proposed CF relates the expected output $E(X_t)$ to the capacity C and the expected load $E(L_t) = E(W_t) + E(X_t)$, where $E(W_t)$ is the expected WIP, for time period t . Essentially, the output increase rate can be controlled by the curvature parameter k , which encompasses the mean and variance of the machine processing time $k = \left((\mu\sigma^2)/2 \right) + \left(1/(2\mu) \right)$.

Missbauer proposed applying the CF shown in Equation 2.3 for the multi-product case. This equation is reproduced here for convenience:

$$E[X(t)] = \frac{C + k + E[L(t)] - \left(C^2 + 2Ck + k^2 - 2CE[L(t)] + 2kE[L(t)] + E[L(t)]^2 \right)^{1/2}}{2}$$

In this formulation the variance of the processing time comes from the combination of product mix and different processing time distributions of each product, aggregated into an effective processing time distribution (Hopp and Spearman [24], Chapter 8).

This model considers product flow between machines in continuous time, while the planning horizon itself is divided into discrete time buckets in the linear programming model presented below. Therefore, flow through non-bottleneck machines happens in discrete time, while flow through bottleneck machines is in continuous time.

Linear programming formulation

The aggregate order-release planning model uses the following notation:

Parameters :

- $h_{\hat{j}mt}$ The costs of holding WIP for aggregate product \hat{j} ($1..J$) at machine m ($1..M$) in a discrete period of time t ($1..T$),
- $l_{\hat{j}t}$ The cost of holding finished goods for aggregate product \hat{j} ,
- $D_{\hat{j}t}$ Demand of the finished goods for aggregate product \hat{j} in a discrete period of time t ,

- $\tilde{P}_{\hat{j}im}$ Average load of aggregated product \hat{j} transported to machine m after completion at machine i ,
- $z_{\hat{j}im\tau}$ The output ratio of aggregated product \hat{j} from machine i to machine m that reach at m in τ periods of time (continuous) after completion at i ,

Decision variables :

- $I_{\hat{j}t}$ Level of inventory of aggregated product \hat{j} at the end of period t ,
- $\hat{I}_{\hat{j}t}$ Amount of work of product group \hat{j} at the end of period t , with no more bottleneck operations to be performed ($I_{\hat{j}t}$ plus work at non-bottlenecks after the last bottleneck operation)
- $\hat{W}_{\hat{j}mt}$ WIP level in front of machine m (non-bottleneck) after completion at machine i (bottleneck) of aggregated product \hat{j} at the end of period t ,
- $W_{\hat{j}mt}$ WIP level in front of machine m of aggregated product \hat{j} at the end of period t ,
- $R_{\hat{j}t}$ Work released of aggregated product \hat{j} in period t ,
- $X_{\hat{j}mt}$ The output from machine m of the aggregated product \hat{j} in period t ,

Objective function :

$$\min \sum_{\hat{j}=1}^{\hat{J}} \sum_{m=1}^M \sum_{t=1}^T \hat{W}_{\hat{j}mt} \cdot h_{\hat{j}mt} + \sum_{\hat{j}=1}^{\hat{J}} \sum_{t=1}^T \hat{I}_{\hat{j}t} \cdot l_{\hat{j}t} \quad (2.8)$$

Subject to:

$$\begin{aligned} W_{\hat{j}mt} = & W_{\hat{j},m,t-1} + \sum_{i=1}^M \sum_{\tau=0}^{\infty} X_{\hat{j},i,t-\tau} \cdot \tilde{P}_{\hat{j}im} \cdot z_{\hat{j}im\tau} \\ & + \sum_{\tau=0}^{\infty} R_{\hat{j},t-\tau} \cdot \tilde{P}_{\hat{j}0m} \cdot z_{\hat{j}0m\tau} - X_{\hat{j}mt} \quad \forall \hat{j}, m, t \end{aligned} \quad (2.9)$$

$$\hat{W}_{\hat{j}mt} = \hat{W}_{\hat{j},m,t-1} + \sum_{i=1}^M X_{\hat{j}it} \cdot \tilde{P}_{\hat{j}im} + R_{\hat{j}t} \cdot \tilde{P}_{\hat{j}0m} - X_{\hat{j}mt} \quad \forall \hat{j}, m, t \quad (2.10)$$

$$I_{\hat{j}t} = I_{\hat{j},t-1} + \sum_{m=1}^M \sum_{\tau=0}^{\infty} X_{\hat{j},m,t-\tau} \cdot \tilde{P}_{\hat{j}m0} \cdot z_{\hat{j}m0\tau} - D_{\hat{j}t} \quad \forall \hat{j}, t \quad (2.11)$$

$$\hat{I}_{\hat{j}t} = \hat{I}_{\hat{j},t-1} + \sum_{m=1}^M X_{\hat{j}mt} \cdot \tilde{P}_{\hat{j}m0} - D_{\hat{j}t} \quad \forall \hat{j}, t \quad (2.12)$$

$$\sum_{\hat{j}=1}^{\hat{j}} X_{\hat{j}mt} \leq \alpha_{mn} + \beta_{mn} \cdot \sum_{\hat{j}=1}^{\hat{j}} (W_{\hat{j}mt} + X_{\hat{j}mt}) \quad \forall n, m, t \quad (2.13)$$

$$R_{\hat{j}t} \leq R_{\hat{j}t}^{max} \quad \forall \hat{j}, t \quad (2.14)$$

$$\hat{W}_{\hat{j}mt}, W_{\hat{j}mt}, I_{\hat{j}t}, \hat{I}_{\hat{j}t}, R_{\hat{j}t}, X_{\hat{j}mt} \geq 0 \quad (2.15)$$

The objective function in the model is to minimize the total cost of holding the WIP and finished goods inventory, subject to restrictions on material flow (2.9), (2.10), inventory balance (2.11), (2.12), and machine capacities (2.13). Equation (2.13) represents the concave CF through piecewise linear constraints aggregated by products \hat{j} . The CF is derived with respect to the machine's expected processing time. Equation (2.14) prevents overloading the non-bottleneck machines; otherwise, they will become the bottleneck. This is another limitation of this formulation, since identifying the bottleneck machines themselves is not obvious. The non-negativity constraint is shown in (2.15).

In the following section, the partitioned clearing function (PCF) is developed and compared with Missbauer's [50] work.

2.1.3.2 Asmundsson, Rardin, and Uzsoy [4] Model

The effect of the product mix on resource utilization and the lead time is captured by the partitioned CF developed by Asmundsson, Rardin, and Uzsoy [4], in which the correlation between the utilization and lead-time is exemplified by a concave curve.

The partitioned CF $f_t(W)$ is derived empirically using a simulation model of a downsized semi-conductor wafer fabrication plant with three products. The plant consists of eleven individual workstations, one of which is identified as a bottleneck and two of which are identified as unreliable. The product mix is introduced to the partitioned CF by ξ_{it} , which is the consumption of the resource time to produce product i at period t .

Formulation of the partitioned clearing function

The formulation uses the following notation:

Indices :

- t Time index,
- n Stage index,
- i Product index.

Parameters :

- ϕ_{it}^n The cost of producing product i in period t at stage n ,
- ω_{it}^n The cost of holding WIP of product i in period t at stage n ,
- ρ_{it}^n The cost of releasing product i in period t at stage n ,
- π_{it}^n The cost of holding finished goods inventory of product i in period t at stage n ,
- δ_{it}^{nk} The cost of transporting product i in period t from stage n to stage k ,
- D_{it}^n The demand of product i in period t at stage n ,
- Z_{it}^n The fraction of the maximum production at stage n in period t allocated to produce product i .

Decision variables :

- X_{it}^n The total production quantity of product i in period t at stage n ,
- R_{it}^n The released quantity of product i
in period t at stage n ,
- W_{it}^n The WIP level of product i in period t at stage n ,
- I_{it}^n The inventory level of the final product i
in period t at stage n ,
- ξ_{it}^n The required time to produce product i
in period t at stage n ,
- Y_{it}^{nk} The quantity transported from stage n
to stage k of product i in period t .

Objective function :

$$\min \sum_t \left[\sum_n \sum_i (\phi_{it}^n X_{it}^n + \omega_{it}^n W_{it}^n + \pi_{it}^n I_{it}^n + \rho_{it}^n R_{it}^n) + \sum_k \delta_{it}^{nk} Y_{it}^{nk} \right] \quad (2.16)$$

Subject to:

$$W_{it}^n = W_{i,t-1}^n - \frac{1}{2} (X_{it}^n + X_{i,t+1}^n) + R_{it}^n + \sum_k Y_{it}^{nk} \quad \forall i, n, t \quad (2.17)$$

$$I_{it}^n = I_{i,t-1}^n - \frac{1}{2} (X_{it}^n + X_{i,t+1}^n) - \sum_k Y_{it}^{nk} - D_{it}^n \quad \forall i, n, t \quad (2.18)$$

$$\alpha_{ct}^n \xi_{it}^n W_{it}^n + \beta_{ct}^n Z_{it}^n - \xi_{it}^n X_{it}^n \geq 0 \quad \forall i, n, t, c \in C(n) \quad (2.19)$$

$$\sum_i Z_{it}^n = 1 \quad \forall n, t \quad (2.20)$$

$$Z_{it}^n, W_{it}^n, X_{it}^n, I_{it}^n, R_{it}^n, Y_{it}^{nk} \geq 0 \quad \forall i, n, t, k \quad (2.21)$$

The objective function of the model is to minimize the total cost of production, holding WIP and finished goods inventory, order release, and transportation. This is subject to restrictions on the flow conservation of WIP (2.17) and finished goods inventory (2.18). The CF is used in (2.19) to model the machine capacity at only the bottleneck machine n . It is estimated using simulation runs from which usually three lines (i.e., $C(n) \in 3$) are constructed, each with slope (α) and intercept (β). Since there are multiple products i , the clearing function (capacity) is partitioned using the variable Z in (2.20). The non-negativity constraint is shown in (2.21).

2.1.3.3 The Albey, Bilge, and Uzsoy [1] Mathematical Model

The next development in the literature on clearing functions is the approach by Albey, Bilge, and Uzsoy [1] who derived a multi-dimensional CF (*MDCF*s) to predict resource performance and then disaggregate it for different products.

The *MDCF*s are derived empirically by constructing a simulation model to study a manufacturing plant according to the $M/G/1$ queue. The plant is assumed to consist of a single machine manufacturing four distinct products. The processing time for these distinct products follows log-normal distribution, and the demand is assumed to follow Poisson distribution.

The simulation output fitted for any of the following *MDCF*s, the one that best fits the simulation output is used as the CF:

$$MDCF1 : TH_i = \frac{(C - \sum_{j \neq i} TH_j) WIP_i^{avg}}{M_i + WIP_i^{avg}} \quad \forall i \quad (2.22)$$

$$MDCF2 : TH_i = \frac{(C - \sum_{j \neq i} TH_j) WIP_i^{avg}}{M_i - b_i(\sum_{j \neq i} TH_j) + WIP_i^{avg}} \quad \forall i \quad (2.23)$$

$$MDCF3 : TH_i = \frac{\left(C - a_i \sum_{j \neq i} TH_j\right) WIP_i^{avg}}{M_i - b_i \left(\sum_{j \neq i} TH_j\right) + WIP_i^{avg}} \quad \forall i \quad (2.24)$$

$$MDCF4 : TH_i = \frac{\left(C - \sum_{j \neq i} a_j TH_j\right) WIP_i^{avg}}{M_i - \left(\sum_{j \neq i} b_j TH_j\right) + WIP_i^{avg}} \quad \forall i \quad (2.25)$$

$$MDCF5 : TH_i = \frac{\left(C - \sum_{j \neq i} a_j WIP_j^{avg}\right) WIP_i^{avg}}{M_i - \left(\sum_{j \neq i} b_j WIP_j^{avg}\right) + WIP_i^{avg}} \quad \forall i \quad (2.26)$$

$$MDCF6 : TH_i = \frac{a_i WIP_i^{avg} + b_i \sum_{j \neq i} WIP_j^{avg}}{M_i + c_i WIP_i^{avg} + d_i \sum_{j \neq i} WIP_j^{avg}} \quad \forall i \quad (2.27)$$

$$MDCF7 : TH_i = \frac{\sum_j a_j WIP_i^{avg}}{M_i + \sum_j b_j WIP_j^{avg}} \quad \forall i \quad (2.28)$$

where TH_i and WIP_i^{avg} are the throughput and average WIP level of product i in time units, C is the maximum resource capacity, M_i is the curvature factor, and a_i , a_j , b_i , b_j , c_i , and d_i are capacity scaling factors for product i , and j (where $j \neq i$).

The first four *MDCF*s are throughput-based CFs, while the remaining are WIP-based CFs. All of these *MDCF*s are derived based on the CF developed by Karmarkar [35] for the $M/M/1$ queuing system, whereas the queuing behavior of the manufacturing plant simulated in this work considers the $M/G/1$ queue.

Once the clearing function is developed using simulation, the *MDCF*s are fit to the data using least squares regression. The conflict between the resource queuing behavior of the simulation and the CF results in an underestimation of the throughput at high WIP levels, and an overestimation of throughput at low WIP levels. The best *MDCF* then enters the planning model described in the next section.

Aggregate production planning based *MDCF*s

The formulation uses the following notation:

Indices :

- t Time index,
- i Product index.

Parameters :

- φ_i The cost of producing product i ,
- ω_i The cost of holding WIP of product i ,

- ρ_i The cost of releasing product i ,
 π_i The cost of holding finished goods inventory of product i ,
 β_i The cost of backorder for product i ,
 d_{it} The demand of product i in period t ,
 ε_i The required time to produce product i .

Variables :

- X_{it} The total production quantity of product i in period t ,
 R_{it} The released quantity of product i in period t ,
 W_{it} The WIP level of product i in period t ,
 I_{it} The inventory level of the final product i in period t ,
 B_{it} The backorder quantity of product i in period t .

Objective function :

$$\min z \sum_{i,t} [\varphi_i X_{it} + \omega_i W_{it} + \pi_i I_{it} + \rho_i R_{it} + \beta_i B_{it}] \quad (2.29)$$

Subject to:

$$I_{i,t-1} + X_{it} + B_{i,t} - B_{i,t-1} - I_{i,t} = d_{it} \quad \forall i, t \quad (2.30)$$

$$W_{it} = W_{i,t-1} - X_{it} + R_{it} \quad \forall i, t \quad (2.31)$$

$$\varepsilon_i X_{it} \leq MDCF_i \quad \forall i, t \quad (2.32)$$

$$X_{it}, W_{it}, I_{it}, R_{it}, B_{it} \geq 0 \quad \forall i, t \quad (2.33)$$

The objective function is to minimize the total costs of production, WIP and finished goods inventory, release, and backorders costs, subject to material flow balance constraints for the finished goods (2.30) and WIP inventory (2.31). Production is limited by the CF for machine capacity (2.32). (2.33) presents the nonnegative constraint.

It should be noted that the method developed in [1] is for a single resource only. While [3] can handle multiple resources, both approaches involve complex curve-fitting from empirical data.

A research question investigated in this thesis is whether the manufacturing system shop floor resource utilization could be represented in aggregate form (across

resources). To answer this question, we consider the MVA, which is an approximation approach for closed-queuing networks. MVA has been successfully used in the area of flexible manufacturing system (FMS) to capture the congestion of different parts competing to be processed at the same machine [72]. The CONWIP system is designed to operate as a closed queuing network, as we shall see in the following section.

2.1.4 CONWIP

CONWIP is based on the closed queuing network concept and is a production control system designed to overcome the shortcomings of the MRP and Kanban systems [67]. Kanban is a Japanese term for managing the movement of parts on the shop floor through the production process. Unfortunately, Kanban only works in a repetitive manufacturing environment with limited product variety. In CONWIP, the total WIP level in the system is kept constant and is chosen to support the required production rate. CONWIP is easier to implement, unlike Kanban that has a high degree of complexity.

In a closed queuing network (CQN) such as CONWIP, the WIP level is controlled to achieve a target production rate. In contrast, the consequence of releasing WIP in an open queuing network sets the production rate. The CQN is more stable since the production planning matches workload release decisions with actual resource capacity. To maintain high service levels with a low WIP, the lead-time should match the committed due date. Therefore, the CQN achieves the required production rate with the lowest WIP level needed taking into account the congestion effect at the production resources.

In addition to the aforementioned, CQN can also be used to model the open queuing network. Dallery, David, and Xie [13] used the performance measure of a closed queuing network to derive an approximation for an open queuing network. Park, Kim, and Jun [60] proposed an approximation method based on MVA to estimate the mean cycle time and the mean queue length for an open queuing network system.

MVA gets its name from the fact that it was first developed to study the first moment of queue length for the exponential service time. It is based on a simple

computation that relates the WIP level in the system to the production rate and lead-time. Reiser and Lavenberg [62] studied a multi-chain closed queuing network with multiple servers by developing an MVA method to calculate performance measures such as production rate, mean waiting time, and queue length. Spearman [67] derived an analytical approach based on MVA to compute the throughput of a closed queuing network as a function of the WIP level in the system. While useful to an extent, Spearman's approach has limitations for high congestion stations.

Suri and Hildebrandt [73] developed the mean value analysis for queues (MVAQ) algorithm to predict the performance measure of closed flexible manufacturing systems producing distinct multiple products. An open question is whether this method can be integrated into a production planning optimization model to capture the effect of product mix in a multiple machines supply chain environment.

As discussed earlier, the CF is a concave non-linear mathematical representation for the relationship between the shop floor load (WIP+release) and the production system throughput. For that, the CF has to be linearized to be integrated into the production planning optimization model. The next section will present the approaches in the literature to linearizing such mathematical functions.

2.2 CF Linearization

The attempts in the literature to linearize a non-linear function, to make it solvable in linear programming, can be classified into four major areas: fuzzy clustering, fixed set of points, piecewise linearization, and outer approximation or outer linearization using a cutting plane algorithm.

2.2.1 Fuzzy Clustering

Fuzzy clustering can be employed to linearize the concave CF into a set of linear constraints using the method developed by Chiu [11]. This approach is applicable for the single product case, or in the case of multiple products, for the partitioned CF approach in Asmundsson, Rardin, and Uzsoy [4]. Omar [56] applied the fuzzy clustering approach to linearize the CF constraint for the single product case through a set of 10 linear equations to solve a production planning model.

In this method, the center of a set of clustered portions is found. Then, the radius of the cluster determines the correlation of the points spread around the center with the cluster center in order to determine the members of the cluster. Finally, the equation line corresponding to each cluster center is derived.

2.2.2 Fixed Points Set

The CF derived for multiple products produced in multiple machines can be represented by a set of fixed points [54]. These points are considered to be the WIP release values to the shop floor. Unfortunately, the effects of the product mix on the CF has been ignored.

2.2.3 Piecewise Linearization

Piecewise linearization is a way to represent the nonlinear function by a set of arbitrarily lines [17]. This is done by introducing these lines to subdivide a particular interval (WIP level) that the nonlinear function lies on, as shown in Figure 2.4. Then, values of the function at the line break-points can be calculated. The essential challenge of this approach is the appropriate number of lines that should be introduced for the linearization. It should not be too few to minimize the approximation error and not too many to keep the problem tractable.

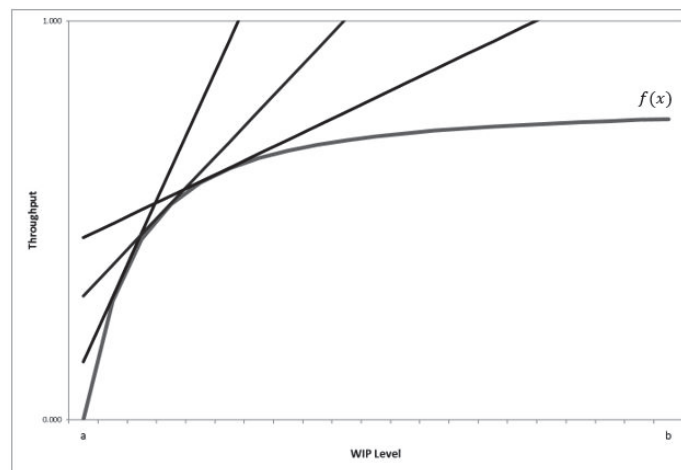


Figure 2.4: Piecewise linearization

Vielma and Nemhauser [75] constructed a non-separable piecewise linear function based on triangulating the domain of the nonlinear variable. This is done by adding

extra constraints and a logarithmic number of binary variable to the formulation. Thus, an optimal solution for the non-separable objective function may be found by applying an independent branching and bound technique based on a special order set for two variables. The branching is applied to the logarithmic binary variables to approximate the value of the objective function by the piecewise linearization on each triangle. Unfortunately, their methodology is applicable to a nonlinear objective function but not for the nonlinear constraint that we have in the case of the integrated CF. While constraints may be moved to the objective function using the Lagrangian method, we prefer in this research to leave this technique for future research.

On the other hand, The CF can be linearized using outer approximation. This function is a constraint in the production planning optimization model to ensure that the production does not exceed available resource capacity. In this approach, successive cuts are introduced until the problem converges to a limit solution with acceptable error [50, 3, 29, 37].

2.3 Summary of the Literature

Firms failing to fulfill customer needs on time can lose their goodwill, which could then lead to customer migration as the competition has been increased with the globalization. Therefore, there is need for a reliable production plan which links the customers demand with the firm resources and with the firm's suppliers too. The key factor for the success of this process is synchronizing the information and goods flow between the supply chain parties. This can be done by estimating a realistic production lead-time by addressing the firm's resources utilization. As ignoring the resource capacity limitation in production planning leads to shop floor congestion, this then increases production lead-time.

The CF is applied to capture the non-linear relationship between production resource capacity loading and lead-time. The literature shows that research on incorporating analytical approximations for the CF for multi-stage supply chain environments is beginning to gather interest. The capacity load at the upstream decision affects the arrival of the lower stream and thus affects the shape of the CF [52].

In addition, various competing products will affect the shape of the CF, so the

product mix should be included in the formulation of the production planning. Kar-markar [35] proved this effect, and more recently for the case of multiple products, Asmundsson, Rardin, Turkseven, and Uzsoy [3] and Albey, Bilge, and Uzsoy [1] present a way to tackle this problem. Asmundsson, Rardin, Turkseven, and Uzsoy [3] constructed a multi-stage CF that they refer to as the allocated clearing function. Albey, Bilge, and Uzsoy [1] derive a multi-dimensional disaggregation CF for a single stage to show the effect of the product mix on the aggregate plan. To the best of our knowledge, these papers represent the state-of-the-art in addressing the interaction between products in a queuing environment. Moreover, the authors in [1] have shown that in the absence of setup times, ACF performs quite well relative to multi-dimensional disaggregation CFs.

The closed queuing literature shows that the MVA methodology can successfully capture the effect of product mix on the performance of the network. Therefore, for closed queuing networks, the relationship between throughput and available WIP level may be derived based on the MVA approach, taking into account the effect of congestion through the network.

Chapter 3

A Data Driven Approach to Estimate the Clearing Function for Multi-Product Production Network Planning Based on Mean-Value Analysis

It is widely accepted that there are two levels of production planning: aggregate at the higher managerial planning level and disaggregate (detailed) at the shop floor or lower managerial planning level. Ignoring the effect of congestion at the higher planning level can make the disaggregate production plan infeasible and in such situations, shop floor managers are left with difficult choices. The popular production-planning tool MRP, which is used at the lower planning level assumes a fixed lead-time [57]. Unfortunately, this assumption leads to an overestimation of resource capacity leading to an infeasible production plan.

To address this issue, queuing theory has been used in the literature to capture the relationship between WIP level and resource throughput which takes the form of a non-decreasing concave function, termed the CF. The CF models the effect of the congestion on the aggregate plan and permits the production planner to decide on shop floor release quantities at the higher planning level.

Recently, the production planning literature shows an increasing interest in different ways of deriving the CF using analytical, simulation, or mixed approaches [52]. In addition, since the resource utilization depends on product mix, which may vary from one period to another depending on the recommendation of the planning run (which is usually a linear program), this needs to be taken into account.

One method suggested in the literature to model the CF in an aggregate planning LP is to incorporate it for bottleneck machines only. In this chapter, we first show the shortcomings of this approach. It has been pointed out in the scheduling literature (e.g. Lawrence and Buss [43], Uzsoy and Wang [74]) that the representation of a queuing network by a bottleneck resource alone can be misleading. We then suggest an MIP formulation to estimate the clearing function at certain discrete WIP points using

the MVA algorithm applicable in closed queuing network settings such as CONWIP and show how the behavior of an entire queuing network may be encapsulated within an aggregate planning model.

In this chapter, we show the deficiency of using the CF for the bottleneck machine only for multi-product networks using two alternative methods of comparison: simulation for the open queuing network case and MVA for the closed queuing network case. Section 3.1 presents the aggregate model and shows how congestion is modeled in production planning LPs using the CF. Section 3.2 presents the basic queuing formulas required for the bottleneck machine approach while section 3.3 shows the MVA algorithm for closed queuing networks, such as CONWIP. Section 3.4 uses a numerical example to demonstrate the deficiency of using the bottleneck approach. Section 3.5 concludes the chapter by summarizing its main contributions.

3.1 Planning Model for Aggregate Production Planning

The aggregate planning problem formulation in Asmundsson, Rardin, Uzsoy [4] is as follows:

Parameters :

- π_{it} Inventory holding cost of product i in time t ,
- w_{it} WIP holding cost of product i in time t ,
- ρ_{it} Raw material release cost of product i in time t ,
- ϕ_{it} Production cost of product i in time t ,
- D_{it} Demand of product i in time t ,
- f_t Clearing function in time period t .

Decision variables :

- X_{it} Production quantity of product i in time t ,
- R_{it} Amount of raw material released of product i in time t ,
- W_{it} WIP level of product i in time t ,
- I_{it} Finished goods inventory level of product i in time t ,
- ξ_{it} Time required to produce unit product i in time t at the resource.

Objective function :

$$\min \sum_t (\rho_{it}R_{it} + w_{it}W_{it} + \phi_{it}X_{it} + \pi_{it}I_{it}) \quad (3.1)$$

Subject to:

$$I_{it} = I_{i,t-1} + X_{it} - D_{it} \quad \forall i, t \quad (3.2)$$

$$W_{it} = W_{i,t-1} + R_{it} - X_{it} \quad \forall i, t \quad (3.3)$$

$$\sum_i \xi_{it}X_{it} \leq f_t(\sum_i \xi_{it}W_{it}) \quad \forall t \quad (3.4)$$

$$R_{it}, W_{it}, X_{it}, I_{it} \geq 0 \quad \forall i, t \quad (3.5)$$

The objective function of the model is to minimize the total raw material release, WIP level, production, and finished good inventory cost (3.1). Material flow conservation for the finished goods and WIP inventory is shown in (3.2) and (3.3), respectively. Production should not exceed machine capacity, which is represented by the CF (3.4). Finally, the non-negativity constraints are shown in (3.5).

3.2 Bottleneck Machine CF Approach

The simplest version of the bottleneck machine CF approach is to consider the $M/M/1$ model, which is characterized by Poisson arrivals and exponential service times. It is possible to use other models such as the $M/G/1$ or the $G/G/1$, but in any case, the approach relies on developing the CF for each product on its bottleneck machine.

Using the notation in Buzacott and Shanthikumar [8]:

$E[N]$ The expected number of entities in the system (WIP level),

$E[T]$ The expected flow time of the entities in the system,

λ_m Arrival rate at machine m ,

μ_m Service rate of machine m ,

The first step in the analysis of the $M/M/1$ system is to calculate server utilization ρ . In the case of a manufacturing network, the bottleneck machine is identified by calculating the utilization of each server in the network using Equation 3.6 and

selecting the machine with the highest utilization rate:

$$\rho = \max_m \left(\frac{\lambda_m}{\mu_m} \right) \quad (3.6)$$

Let the bottleneck machine be b , i.e.,

$$b = \operatorname{argmax}_m \left(\frac{\lambda_m}{\mu_m} \right) \quad (3.7)$$

Secondly, the expected flow time of the entities in the system can be calculated by the following equation:

$$E[T] = \frac{\rho}{\lambda_b(1 - \rho)} = \frac{1}{\mu_b - \lambda_b} \quad (3.8)$$

Thirdly, system throughput, which also may be represented by the arrival rate (λ_b) in steady-state, can be found from Equation 3.6, Equation 3.8 and Little's law, from which $E[N] = \lambda_b E[T]$. Thus, the clearing function is given by:

$$\lambda_b = \frac{E[N]\mu_b}{E[N] + 1} \quad (3.9)$$

3.3 MVA Algorithm Development for the Integrated CF Approach

The MVA algorithm can capture the non-linear relationship between the workload and the throughput for closed queuing networks for multiple products. In this approach, products are allowed to have different processing times and routings. It is also possible for some products to revisit a machine more than once along their production route, which is common in semiconductor manufacturing.

The MVA algorithm is presented in Suri and Hildebrant [73]. This procedure can be used to model the behavior of a production facility based on the following parameters, using the notation in Askin and Standridge [2]:

- Number of jobs in process N_p , where $p = 1, \dots, P$ are the product types.
- Number of servers c_j at workstation j .
- Mean processing time μ_{jp}^{-1} of product p at workstation j .

- Number of visits v_{jp} of product p to workstation j .

The objective of the MVA algorithm is to find the expected lead-time L_{jp} , production rate X_p , and WIP level W_{jp} , as follows:

Step 1 : Initialize the iteration counter $\tau = 0$, and compute the initial WIP level in the system $W_{jp}^0 = \frac{N_p}{Z_p}$, where Z_p is the number of workstations visited by product p .

Step 2 : Update L_{jp}^τ ,

$$\tau = \tau + 1,$$

$$\text{Calculate } L_{jp}^\tau = \mu_{jp}^{-1} + \frac{N_p - 1}{N_p} W_{jp}^\tau \mu_{jp}^{-1} + \sum_{r \neq p} (W_{jr}^\tau \mu_{jr}^{-1}) \quad \forall j, p,$$

or $L_{jp}^\tau = \frac{\mu_{jp}^{-1}}{c_j} + \frac{N_p - 1}{N_p} W_{jp}^\tau \frac{\mu_{jp}^{-1}}{c_j} + \sum_{r \neq p} (W_{jr}^\tau \frac{\mu_{jr}^{-1}}{c_j})$ in case the number of servers at the workstations is more than one.

Step 3 : Update production rate X_p^τ ,

$$\text{Calculate } X_p^\tau = \frac{N_p}{\sum_{j=1}^M v_{jp} L_{jp}^\tau} \quad \forall p.$$

Step 4 : Update W_{jp}^τ ,

$$\text{Calculate } W_{jp}^\tau = X_p^\tau (v_{jp} L_{jp}^\tau),$$

If $(\frac{W_{jp}^\tau - W_{jp}^{\tau-1}}{W_{jp}^\tau}) > \epsilon$ go to Step 2; else stop. Here, ϵ is an arbitrarily small number.

This algorithm can provide the inputs to an aggregate planning model to model multiple resources multiple products CFs. However, a critical step in this approach is to choose the WIP level N_p of product p , as will be discussed later in Section 3.4.2.

3.4 Numerical Example

We now look at a numerical example to evaluate the difference between the bottleneck machine CF and the integrated CF approach. The example presented is taken from the wafer fabrication facility described in Kayton, Teyner, Schwartz, and Uzsoy [36] with three differences. First, the case presented in the original paper has batching in machines 1 and 2 and also unreliable machines. For simplicity, we ignore batching and machine reliability. Second, the lognormal distribution is proposed for processing

times in Kacar, Irdem, and Uzsoy [29], who also study this case. Third, the shop floor runs 8 hours per day, 7 days per week. In this chapter, we assume exponential processing times. The assumption of exponential processing times and no failures makes the problem of getting a good fit via MVA a lot easier.

The shop floor comprises eleven stations (see Figure 3.1) and produces three different products with the same processing time on the stations when a product visits a station. However, the routings are different for each product. The product routings are shown in Figure 3.1. The average machine processing time for the different products and the number of visits to each machine are shown in Tables 3.1 and 3.2 respectively. Kacar, Irdem, and Uzsoy [29] identified station 4 as the bottleneck for the products when the product ratio is 3:1:1. The bottleneck machine could in fact change based on the product mix recommended by the aggregate planning LP.

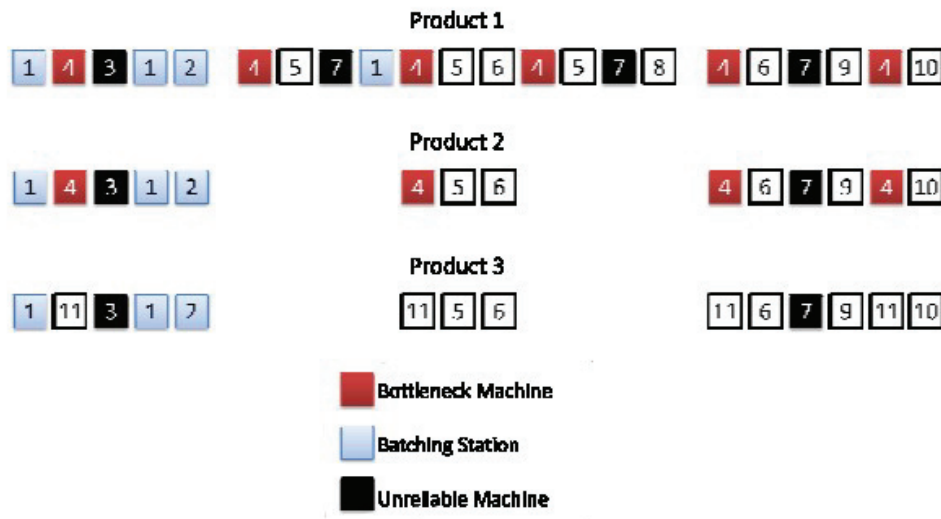


Figure 3.1: Production routings as taken directly from Kayton et al. [36]

Table 3.1: Machine processing time

	Machines										
	1	2	3	4	5	6	7	8	9	10	11
Processing Time (min)	80	220	45	40	25	22	20	100	50	50	70

Table 3.2: Number of visits

	Product	Machine											Total
		1	2	3	4	5	6	7	8	9	10	11	
Number of Visit	1	3	1	1	6	3	2	3	1	1	1	0	22
	2	2	1	1	4	1	2	1	0	1	1	0	14
	3	2	1	1	0	1	2	1	0	1	1	4	14

3.4.1 Analysis Using the $M/M/1$ Bottleneck Machine CF Approach

The arrival rate for the three products are shown in Table 3.3.

Table 3.3: Interarrival time

	Product 1	Product 2	Product 3
Interarrival Time (min)	300	350	330

The CF for the $M/M/1$ requires identifying the bottleneck machine, defined as the machine with the highest effective utilization rate. The effective arrival rate at each station for a re-entrant system can be calculated using Equation 3.10. In this equation, N_{im} is the number of visits by product i to machine m , PM_i is the product mix proportion and λ_i represents the product i arrival rate. PM_i is a multiplication factor applied to the nominal arrival rate λ_i ; it will be used later in this chapter to model a range of product combinations typical in manufacturing systems by enforcing $\sum_i PM_i = 1$. The number of visits N_{im} is considered because in a re-entrant production environment, products may need to visit a machine more than once on their production process route.

$$\lambda_m = \sum_{i=1} (N_{im} PM_i \lambda_i) \quad (3.10)$$

Then, the bottleneck machine is identified by using Equation 3.11, with the highest value of effective utilization.

$$\rho_e = \max_m \left(\frac{\lambda_m}{\mu_m} \right) \quad (3.11)$$

In the first numerical experiment, the arrival rate of products 2 and 3 in the

network are set to zero, and the arrival rate of product 1 (λ_1) is varied. For this case, both stations 1 and 4 are bottlenecks. From Table 3.3, $\lambda_1 = 60/300 = 0.2$ in units per hour. This value is multiplied by a factor between 0.1 to 1.2, as illustrated in Table 3.4. For product 1 on station 1, since the processing time is 80 minutes, the station service rate is $\mu_1 = 60/80 = 0.75$. It can be seen in Table 3.4 that the arrival rate of product 1 cannot be multiplied by 1.25 or more in order to ensure that $\rho_e < 1$ and the system is stable. The $\rho_e < 1$ values for station 4 are also identical because the processing time on station 4 is double, as is the number of visits.

Table 3.4: Effect of product 1 arrival rate at station 1 in M/M/1 model

Multiplier Factor	N_1	λ_1 (h)	Effective $\lambda_{m=1}$	$\mu_{m=1}$ (h)	Effective ρ_e
0.1	3	0.2	0.06	0.75	0.08
0.2	3	0.2	0.12	0.75	0.16
0.3	3	0.2	0.18	0.75	0.24
0.4	3	0.2	0.24	0.75	0.32
0.5	3	0.2	0.30	0.75	0.40
0.6	3	0.2	0.36	0.75	0.48
0.7	3	0.2	0.42	0.75	0.56
0.8	3	0.2	0.48	0.75	0.64
0.9	3	0.2	0.54	0.75	0.72
1.0	3	0.2	0.60	0.75	0.80
1.1	3	0.2	0.66	0.75	0.88
1.2	3	0.2	0.72	0.75	0.96
1.25	3	0.2	0.75	0.75	1.00
1.3	3	0.2	0.78	0.75	1.04

The $M/M/1$ WIP levels are calculated for effective utilization (ρ_e) values using the following relation in [6]: $WIP = \rho/(1 - \rho)$. These values are contrasted in Figure 3.2 against the WIP obtained from an open queuing simulation model of the network as well as the MVA method for the closed queuing network case. This figure also shows the 95% confidence lower and upper bounds of WIP observed from the simulation model (these values are very close and sometimes not visible on the subgraph). Also, the WIP levels for the bottleneck $M/M/1$ are multiplied by a scale factor of 10 in the subgraph on top. It should be noted that the WIP level is very low because in the bottleneck $M/M/1$ case, it is the WIP at machine 1. On the other hand, in the MVA and simulation cases, the WIP value is the total WIP across all 11 machines.

The bottom subgraph in Figure 3.2 shows the WIP-Throughput relationship (CF) for the $M/M/1$ case and the shape of the non-linear curve can be seen clearly.

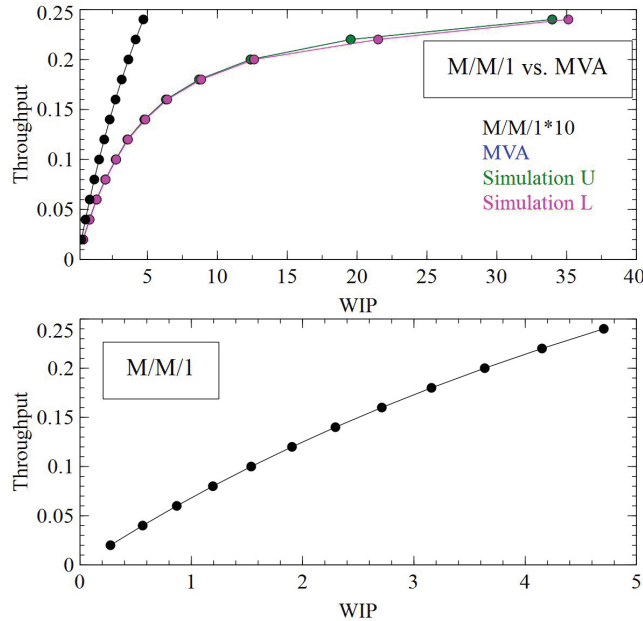


Figure 3.2: The CFs of $M/M/1$, MVA, and open queuing network simulation (upper and lower bounds)

In the next set of experiments, we look at different proportions of the three products summing up to 1, as shown in Table 3.5. The sets shown in Table 3.5 are such that there are two active and one inactive product in any row. These are just representative combinations, it is possible to have all three active products so long as the queuing network is stable for the open queuing network case. Note that the product proportions are based on 6 steps, i.e., the step size is $1/6 \approx 0.17$. It is also worth noting that several product mix sets are considered in Table 3.5, as opposed to one product mix (3:1:1) in Asmundsson, Rardin, Turkseven, and Uzsoy [3], which correspond to (0.6, 0.2, 0.2) in our product mix construction. The corresponding arrival rates for products 1, 2, and 3 turn out to be 0.2, 0.17, and 0.18 units per hour respectively.

For each product mix, the bottleneck machine is identified by first calculating the effective arrival rate at each machine using Equation 3.10. Consider the third product mix (0.34, 0.66, and 0) in the first set of Table 3.5. Table 3.6 shows the effective arrival rate for the 11 stations. For this product mix, station 2 is the bottleneck,

Table 3.5: List of the product mix sets

Set	Product Mix		
	P1	P2	P3
1	0	1	0
	0.17	0.83	0
	0.34	0.66	0
	0.50	0.50	0
	0.67	0.33	0
	0.83	0.17	0
	1	0	0
2	0	0	1
	0.17	0	0.83
	0.34	0	0.66
	0.50	0	0.50
	0.67	0	0.33
	0.83	0	0.17
	1	0	0
3	0	0	1
	0	0.17	0.83
	0	0.34	0.66
	0	0.50	0.50
	0	0.67	0.33
	0	0.83	0.17
	0	1	0

as indicated by the (*) symbol since the value of λ_m/μ_m is the highest for this row in the table. Table 3.7 shows the bottleneck station for each of the product mixes of Table 3.5. It can be clearly seen that the bottleneck machine changes its location based on the product mix.

Table 3.6: Effective arrival rates for product mix (0.33, 0.67, 0)

Stn(m)	μ_m (*)	Number of Visits			Product Mix			Arrival Rate (units\h)			Effect. λ_m	ρ_m
		$N1$	$N2$	$N3$	$P1$	$P2$	$P3$	λ_1	λ_2	λ_3		
1	0.75	3	2	2	0.33	0.67	0	0.2	0.17	0.18	0.43	0.57
2*	0.27	1	1	1	0.33	0.67	0	0.2	0.17	0.18	0.18	0.66
3	1.33	1	1	1	0.33	0.67	0	0.2	0.17	0.18	0.18	0.14
4	1.50	6	4	0	0.33	0.67	0	0.2	0.17	0.18	0.86	0.57
5	2.40	3	1	1	0.33	0.67	0	0.2	0.17	0.18	0.31	0.13
6	2.73	2	2	2	0.33	0.67	0	0.2	0.17	0.18	0.36	0.13
7	3.00	3	1	1	0.33	0.67	0	0.2	0.17	0.18	0.31	0.10
8	0.60	1	0	0	0.33	0.67	0	0.2	0.17	0.18	0.07	0.11
9	1.20	1	1	1	0.33	0.67	0	0.2	0.17	0.18	0.18	0.15
10	1.20	1	1	1	0.33	0.67	0	0.2	0.17	0.18	0.18	0.15
11	0.86	0	0	4	0.33	0.67	0	0.2	0.17	0.18	0.00	0.00

Table 3.7: Bottleneck machine for each product mix set

Set	Product Mix			Bottleneck Machine
	P1	P2	P3	
1	0	1	0	2
	0.17	0.83	0	2
	0.34	0.66	0	2
	0.50	0.50	0	2
	0.67	0.33	0	2
	0.83	0.17	0	1
	1	0	0	1
2	0	0	1	11
	0.17	0	0.83	11
	0.34	0	0.66	2
	0.50	0	0.50	2
	0.67	0	0.33	2
	0.83	0	0.17	1
	1	0	0	1
3	0	0	1	11
	0	0.17	0.83	11
	0	0.34	0.66	2
	0	0.50	0.50	2
	0	0.67	0.33	2
	0	0.83	0.17	2
	0	1	0	2
4	0.6	0.2	0.2	2

3.4.2 Analysis Using the MVA CF Approach

A critical input in developing the CF for the MVA case is the WIP level of the system. In order to estimate a reasonable value of WIP, an open queuing simulation model of the network is used. For each product mix, the open queuing network is simulated and the average WIP is estimated for a 95% confidence interval. The simulation model is run for 400 replications with a run length of 100 days per replication. Table 3.8 shows the resulting WIP and half-width of the WIP. Using the WIP levels from the simulation output of the open queuing network, the MVA method is used to find the throughput of an equivalent closed queuing system, as can be seen in Table 3.9.

Table 3.8: Simulation WIP

Set	Product Mix			Simulation WIP			Half Width 95%		
	P1	P2	P3	WIP1	WIP2	WIP3	P1	P2	P3
1	0	1	0	0.00	4.17	0.00	0.00	0.06	0.00
	0.17	0.83	0	1.16	3.69	0.00	0.02	0.05	0.00
	0.34	0.66	0	2.51	3.19	0.00	0.01	0.01	0.00
	0.50	0.50	0	4.14	2.60	0.00	0.06	0.04	0.00
	0.67	0.33	0	6.10	1.91	0.00	0.10	0.03	0.00
	0.83	0.17	0	8.83	1.10	0.00	0.17	0.03	0.00
	1	0	0	12.38	0.00	0.00	0.26	0.00	0.00
2	0	0	1	0.00	0.00	9.22	0.00	0.00	0.25
	0.17	0	0.83	1.08	0.00	5.62	0.02	0.00	0.10
	0.34	0	0.66	2.36	0.00	4.10	0.04	0.00	0.06
	0.50	0	0.50	3.83	0.00	2.93	0.05	0.00	0.05
	0.67	0	0.33	5.69	0.00	1.97	0.09	0.00	0.04
	0.83	0	0.17	8.34	0.00	1.03	0.16	0.00	0.02
	1	0	0	12.38	0.00	0.00	0.26	0.00	0.00
3	0	0	1	0.00	0.00	9.22	0.00	0.00	0.25
	0	0.17	0.83	0.00	0.65	5.39	0.00	0.01	0.09
	0	0.34	0.66	0.00	1.35	3.82	0.00	0.02	0.06
	0	0.50	0.50	0.00	1.99	2.54	0.00	0.03	0.04
	0	0.67	0.33	0.00	2.69	1.57	0.00	0.04	0.02
	0	0.83	0.17	0.00	3.37	0.76	0.00	0.05	0.01
	0	1	0	0.00	4.17	0.00	0.00	0.06	0.00

Table 3.9: MVA throughput

Set	Product Mix			MVA WIP			MVA Throughput		
	P1	P2	P3	WIP1	WIP2	WIP3	Th1	Th2	Th3
1	0	1	0	0.00	4.17	0.00	0.00	0.20	0.00
	0.17	0.83	0	1.16	3.69	0.00	0.04	0.16	0.00
	0.34	0.66	0	2.51	3.19	0.00	0.07	0.13	0.00
	0.50	0.50	0	4.14	2.60	0.00	0.11	0.09	0.00
	0.67	0.33	0	6.10	1.91	0.00	0.14	0.06	0.00
	0.83	0.17	0	8.83	1.10	0.00	0.18	0.03	0.00
	1	0	0	12.38	0.00	0.00	0.21	0.00	0.00
2	0	0	1	0.00	0.00	9.22	0.00	0.00	0.20
	0.17	0	0.83	1.08	0.00	5.62	0.04	0.00	0.17
	0.34	0	0.66	2.36	0.00	4.10	0.07	0.00	0.14
	0.50	0	0.50	3.83	0.00	2.93	0.11	0.00	0.10
	0.67	0	0.33	5.69	0.00	1.97	0.14	0.00	0.07
	0.83	0	0.17	8.34	0.00	1.03	0.18	0.00	0.03
	1	0	0	12.38	0.00	0.00	0.21	0.00	0.00
3	0	0	1	0.00	0.00	9.22	0.00	0.00	0.20
	0	0.17	0.83	0.00	0.65	5.39	0.00	0.03	0.17
	0	0.34	0.66	0.00	1.35	3.82	0.00	0.06	0.14
	0	0.50	0.50	0.00	1.99	2.54	0.00	0.10	0.10
	0	0.67	0.33	0.00	2.69	1.57	0.00	0.13	0.07
	0	0.83	0.17	0.00	3.37	0.76	0.00	0.16	0.04
	0	1	0	0.00	4.17	0.00	0.00	0.20	0.00

3.4.3 Comparison of Results

The WIP level using the different approaches (bottleneck $M/M/1$ and simulation or MVA) for each of the product mixes is shown in Table 3.10. The simulation or MVA WIP level is roughly three times higher than in the $M/M/1$ case because the simulation WIP is a total across the 11 stations, while the $M/M/1$ WIP is at the bottleneck machine only.

The throughput of the system using the three approaches (bottleneck $M/M/1$, open network simulation, and MVA) for each of the product mixes is shown in Table 3.11. In the $M/M/1$ estimation, the bottleneck machine used for the calculation depends on the product mix and changes according to Table 3.7. Table 3.11 can also be presented graphically. Since in each set, only two products are active, Figures 3.3, 3.4 and 3.5 show system throughput as a function of product mix. In each of these figures, when the proportion of a product is increased (the x-axis in the top part of each figure), the corresponding proportion of the other product decreases (the x-axis in the bottom part of each figure).

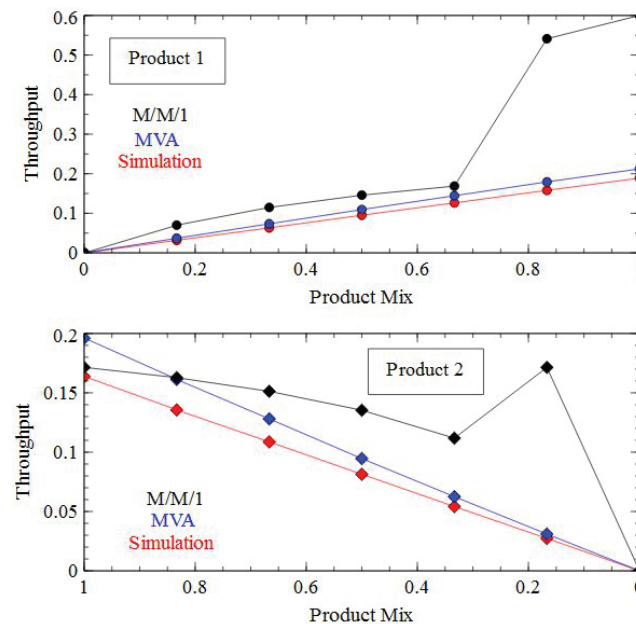


Figure 3.3: Increasing product 1 while decreasing product 2 WIP level

It is expected that as the product proportion increases (or decreases), its throughput also increases (or decreases), though it is expected to taper off asymptotically.

Table 3.10: WIP analysis

Set	M/M/1			Simulation		
	WIP1	WIP2	WIP3	WIP1	WIP2	WIP3
1	0.00	1.69	0.00	0.00	4.17	0.00
	0.35	1.48	0.00	1.16	3.69	0.00
	0.73	1.25	0.00	2.51	3.19	0.00
	1.15	0.99	0.00	4.14	2.60	0.00
	1.62	0.69	0.00	6.10	1.91	0.00
	2.59	0.30	0.00	8.83	1.10	0.00
	4.00	0.00	0.00	12.38	0.00	0.00
2	0.00	0.00	5.60	0.00	0.00	9.22
	0.00	0.00	2.41	1.08	0.00	5.62
	0.79	0.00	1.43	2.36	0.00	4.10
	1.22	0.00	1.11	3.83	0.00	2.93
	1.69	0.00	0.77	5.69	0.00	1.97
	2.64	0.00	0.32	8.34	0.00	1.03
	4.00	0.00	0.00	12.38	0.00	0.00
3	0.00	0.00	5.60	0.00	0.00	9.22
	0.00	0.00	2.41	0.00	0.65	5.39
	0.00	0.61	1.28	0.00	1.35	3.82
	0.00	0.89	0.95	0.00	1.99	2.54
	0.00	1.17	0.62	0.00	2.69	1.57
	0.00	1.43	0.30	0.00	3.37	0.76
	0.00	1.69	0.00	0.00	4.17	0.00

Table 3.11: Throughput analysis

Set	$M/M/1$			Simulation			MVA		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	0.00	0.17	0.00	0.00	0.16	0.00	0.00	0.20	0.00
	0.07	0.16	0.00	0.03	0.14	0.00	0.04	0.16	0.00
	0.11	0.15	0.00	0.06	0.11	0.00	0.07	0.13	0.00
	0.15	0.14	0.00	0.10	0.08	0.00	0.11	0.09	0.00
	0.17	0.11	0.00	0.13	0.05	0.00	0.14	0.06	0.00
	0.54	0.17	0.00	0.16	0.03	0.00	0.18	0.03	0.00
	0.60	0.00	0.00	0.19	0.00	0.00	0.21	0.00	0.00
2	0.00	0.00	0.73	0.00	0.00	0.19	0.00	0.00	0.20
	0.00	0.00	0.61	0.03	0.00	0.14	0.04	0.00	0.17
	0.12	0.00	0.16	0.06	0.00	0.12	0.07	0.00	0.14
	0.15	0.00	0.14	0.10	0.00	0.09	0.11	0.00	0.10
	0.17	0.00	0.12	0.13	0.00	0.06	0.14	0.00	0.07
	0.54	0.00	0.18	0.16	0.00	0.03	0.18	0.00	0.03
	0.60	0.00	0.00	0.19	0.00	0.00	0.21	0.00	0.00
3	0.00	0.00	0.73	0.00	0.00	0.19	0.00	0.00	0.20
	0.00	0.00	0.61	0.00	0.03	0.14	0.00	0.03	0.17
	0.00	0.10	0.15	0.00	0.05	0.12	0.00	0.06	0.14
	0.00	0.13	0.13	0.00	0.08	0.09	0.00	0.10	0.10
	0.00	0.15	0.10	0.00	0.11	0.06	0.00	0.13	0.07
	0.00	0.16	0.06	0.00	0.13	0.03	0.00	0.16	0.04
	0.00	0.17	0.00	0.00	0.16	0.00	0.00	0.20	0.00

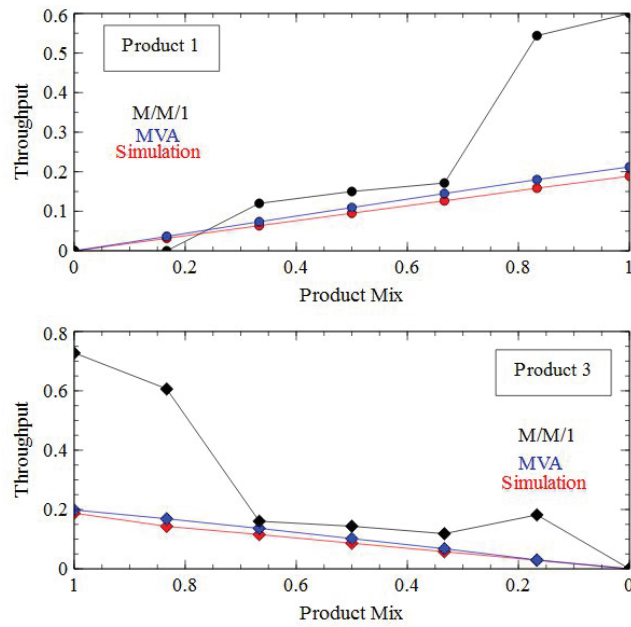


Figure 3.4: Increasing product 1 while decreasing product 3 WIP level

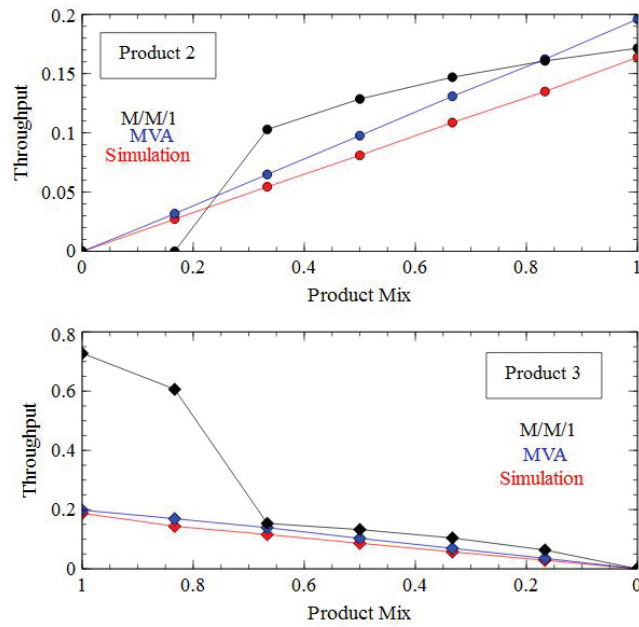


Figure 3.5: Increasing product 2 while decreasing product 3 WIP level

This phenomenon is seen in both the simulation and MVA methods in Figures 3.3, 3.4 and 3.5. However, there are anomalies in the bottleneck $M/M/1$ approach. For example, in Figure 3.3, the throughput of product 2 increases even though its WIP decreases when going from a proportion of 0.33 to 0.17. This is because the bottleneck of the system changes from station 2 to station 1 (refer to Table 3.7) and the throughput of product 2 gets overestimated. The same phenomenon is observed for product 3 in Figure 3.4 as its proportion decreases from 0.33 to 0.17. On the other hand, the throughput of product 2 when product mix is 1 in Figures 3.3 and 3.5, using the $M/M/1$ model lies between the throughput estimations of the MVA and the open network simulation.

Thus, there are significant conceptual and numerical differences between the bottleneck $M/M/1$ CF approach and the MVA or simulation CF approaches. Since the MVA or simulation looks at the system as a whole (all products and all stations are considered simultaneously), it provides a more accurate representation of the production network capacity than the bottleneck approach. This is despite the fact that the bottleneck $M/M/1$ model accounts for the shifting bottleneck as the product mix changes.

3.5 Conclusions and Future Work

This chapter demonstrates how the effect of product mix can be taken into account while developing the CF for a multi-product production network. It also presents the conceptual difference between the bottleneck $M/M/1$ and MVA CF approaches and identifies the weakness of the former. It is apparent that using the MVA CF approach provides consistent results for use in aggregate production planning. From this, a data driven CF MIP is presented in the next chapter as an alternative to the ACF model in Asmundsson, Rardin, Turkseven, and Uzsoy [3]. It would be interesting to see whether the release decisions can be made continuous through an alternative formulation.

Chapter 4

Data Driven Aggregate Production Planning Model based on Queuing Network WIP/Throughput Estimates

PPC is a function within a firm's ERP system required to plan and fulfill customer demand. PPC synchronizes the flow of planning information throughout a firm's hierarchical planning levels. It assigns available shop floor capacity (e.g. resource, material, personnel, etc.) to different products according to anticipated customer demand. The top level in the hierarchy is aggregate planning, where gross customer demand is matched with existing supply. In the second planning level, the MPS specifies the order release size and timing to cater to customer demand. Finally, the MRP translates the former plans into a detailed production plan of the required raw materials based on products BOM.

The essential coordination factor between aggregate planning, MPS, and MRP levels is material lead time. Material lead time or flow time is the total time that the material spend in the shop floor to be processed, including queuing, batching, and processing, to constitute the finished product [25].

The classical aggregate planning model in the literature is based on fixed material lead times, which could affects the feasibility of a production plan. From queuing theory, it is well known that the material lead time increases nonlinearly as the server utilization gets closer to 100%. Therefore, the concept of the CF was developed in the production planning literature to capture this relationship [35]. The CF concept can greatly enhance the feasibility of an aggregate production planning solution since it incorporates the congestion effect.

Most of the analytical approaches in the literature are focused on deriving the CF for the bottleneck resource. However, as seen in the previous chapter, the location of the bottleneck shifts based on the release decision (product mix). Also, the queuing network perspective provides a much better estimate of the relationship between WIP and throughput.

In this chapter, we look at how the CF for a closed-queuing multi-product manufacturing system such as the CONWIP can be incorporated into an aggregate production planning model to optimize the release of parts to the shop floor. In doing so, the queuing network aspect of the CF is explicitly recognized instead of the indirect approach using bottleneck machines.

This is a robust method because the use of CONWIP only makes the MVA approach more accurate; a simulation, or open or closed queuing network analysis, for example, would not need CONWIP, but could still provide data for the proposed approach.

The remainder of this chapter is organized as follows: Section 4.1 proposes an aggregate production planning model. Section 4.2 presents the numerical example. Section 4.3 shows model results and sensitivity analysis scenarios to gain a deeper understanding of the model. Section 4.4 presents supplementary constraint and variables to decrease the computational time. Finally, the summary of the chapter is presented in Section 4.5.

4.1 Fixed Points Release (FPR) Model

The aggregate planning model of Asmundsson, Rardin, and Uzsoy [4] is formulated for each product i , each stage or resource n , and each period t . In fact, Asmundsson, Rardin, Turkseven, and Uzsoy [3] build an aggregate CF at each resource and then allocate it to the different products using a reservation variable Z_{it}^n for each product i on resource n in time t such that $\sum_i Z_{it}^n = 1 \forall n, \forall t$.

In this chapter, we take an alternative approach and present an aggregate production planning model extending the one in Mrishih [54]. The model could be thought of as a data-driven CF estimation approach since it uses estimates of the WIP/Throughput relationship at certain discrete WIP values of each product. The WIP and throughput estimates take a queuing network perspective, and as such are not broken down by resource. In what follows, the WIP and throughput values are estimated using MVA for a closed-queuing network example such as the CONWIP manufacturing system.

The idea here is to use patterns (combinations of product quantities) which result in discrete permissible values of WIP in each period. A binary variable is introduced

to make sure that only one of the proposed WIP patterns is active in a given time period. In what follows, the favors pattern and combination are used interchangeably. The MIP model for the data-driven CF approach is formulated as follows (we call it the fixed points release model from here on):

FPR-MIP model

Indices :

- g Product index,
- p Period index,
- r Release index.

Parameters :

- x_g Raw material release cost of product g ,
- w_g WIP holding cost of product g ,
- h_g Inventory holding cost of product g ,
- b_g Backorder cost of product g ,
- d_{gp} Demand of product g at period p ,
- Q_{gr} WIP releasing pattern (MVA input) of product g in WIP pattern r ,
- O_{gr} Production throughput based on MVA output of product g in WIP pattern r .

Decision Variables :

- Y_{gp} Production quantity of product g in period p ,
- X_{gp} Amount of raw material released of product g in period p ,
- W_{gp} WIP level of product g in period p ,
- I_{gp} Finished goods inventory level of product g in period p ,
- B_{gp} Backorder level of product g in period p ,
- $Z_{rp} = \begin{cases} 1, & \text{if WIP pattern } r \text{ is chosen in period } p \\ 0, & \text{Otherwise} \end{cases}$

Objective function :

$$\min \sum_{g \in G} \sum_{p \in P} (x_g X_{gp} + w_g W_{gp} + h_g I_{gp} + b_g B_{gp}) \quad (4.1)$$

Subject to :

$$W_{gp} = W_{g,p-1} + X_{gp} - Y_{gp} \quad \forall g \in G, p \in P \quad (4.2)$$

$$I_{gp} = I_{g,p-1} + Y_{gp} - B_{g,p-1} + B_{gp} - d_{gp} \quad \forall g \in G, p \in P \quad (4.3)$$

$$W_{gp} = \sum_{r \in R} Q_{gr} Z_{rp} \quad \forall g \in G, p \in P \quad (4.4)$$

$$Y_{gp} = \sum_{r \in R} O_{gr} Z_{rp} \quad \forall g \in G, p \in P \quad (4.5)$$

$$\sum_{r \in R} Z_{rp} = 1 \quad p \in P \quad (4.6)$$

$$Y_{gp}, I_{gp}, B_{gp}, W_{gp}, X_{gp} \geq 0 \quad \forall g \in G, p \in P \quad (4.7)$$

$$Z_{rp} \in \{0, 1\} \quad \forall r \in R, p \in P \quad (4.8)$$

The objective function is to minimize the sum of raw material release, WIP release, inventory, and backorder costs. Constraints (4.2), (4.3) are the flow conservation constraint for WIP and finished goods inventory. The WIP-Throughput relationship (to model congestion) is represented by constraints (4.4), (4.5) and (4.6). Constraint (4.4) states that the system WIP is equal to the WIP represented by discrete WIP patterns for which the throughput of the system is estimated using MVA. Constraint (4.5) states that the throughput of the system is equal to the MVA estimate for a given WIP pattern. Constraint (4.6) ensures that only one of the WIP patterns is chosen. Finally, constraints (4.7), (4.8) are the non-negativity constraint for the model variables, and the binary constraint for the WIP pattern choice variable Z_{rp} , respectively. Therefore, this formulation has $5|G||P|$ continuous decision variables, $|R||P|$ binary variables, and $4|G||P|+|P|$ constraints.

As can be seen in the model, the CF is no longer modeled on an individual resource (workstation) basis; rather, a set of WIP patterns is generated beforehand and for each WIP pattern, the corresponding throughput of each product is estimated using

the MVA (the estimates could also come from an open or closed queuing network simulator).

4.2 Illustrative Example of the Data-Driven CF Aggregate Planning Model

The example presented in Chapter 3 is used to illustrate how the patterns are generated. Table 4.1 shows the different product mixes considered in the model, the WIP values in the patterns (Q), and system throughput computed by the MVA approximation (O) and can be built as follows:

1. The first step is to decide on how interesting WIP patterns can be generated. Since there are three products, using a step size of three for example, the product mix ratios are stratified as 0.0, 0.33, 0.67, or 1.0. Since the product mix proportions of each product should add to up to one (with the exception of 0.0, 0.0, 0.0), there are 11 resulting combinations. The (0.0, 0.0, 0.0) combination is a valid option because the aggregate planning model should be allowed to decide to produce nothing in a particular period.
2. In the second step, for each product mix, the WIP level is estimated using an open network simulation of the production queuing network. Strictly speaking, the WIP patterns can be chosen directly, but using the product mix proportion approach makes the choice systematic and provides good calibration of the inputs. In a practical application, the pattern mixes should cover the arrival proportions of the various products.
3. In the third step, the MVA algorithm is run for each WIP combination to obtain product throughputs which can then be entered into FPR, the data-driven aggregate planning MIP model. Although, an open queuing network simulation model is used to obtain representative WIP levels, the throughput estimates are for a closed queuing network, which is stable by definition.

The demand for the three products in each time period are first randomly generated between 0 and 1. Then, the three numbers are normalized so that the sum of

Table 4.1: Pattern, product mix, WIP, and throughput

Pattern	Product Mix			WIP (units)			MVA Throughput (units\week)		
	1	2	3	Q1	Q2	Q3	O1	O2	O3
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	9.22	0.00	0.00	11.10
3	0.00	0.33	0.67	0.00	1.35	3.82	0.00	3.63	7.79
4	0.00	0.67	0.33	0.00	2.69	1.57	0.00	7.33	3.88
5	0.00	1.00	0.00	0.00	4.17	0.00	0.00	10.98	0.00
6	0.33	0.00	0.67	2.36	0.00	4.10	4.12	0.00	7.64
7	0.33	0.33	0.33	2.40	1.55	1.75	4.13	3.61	3.86
8	0.33	0.67	0.00	2.51	3.19	0.00	4.11	7.17	0.00
9	0.67	0.00	0.33	5.69	0.00	1.97	8.11	0.00	3.80
10	0.67	0.33	0.00	6.10	1.91	0.00	8.09	3.50	0.00
11	1.00	0.00	0.00	12.38	0.00	0.00	11.88	0.00	0.00

the product mix is 1.00. Finally, the normalized numbers are multiplied by the corresponding arrival rate for each product and the weekly working hours, as shown in Table 4.2 to obtain the demand in units/week. The cost coefficients of raw material, WIP, inventory, and backorder used in the example are 3, 7, 15, and 20 respectively.

Table 4.2: Generated demand

Period	Random Number			Sum	Normalized RN			Sum	Demand (units\week)		
	P1	P2	P3		P1	P2	P3		P1	P2	P3
1	0.75	0.04	0.23	1.02	0.73	0.04	0.23	1.00	8.22	0.34	2.35
2	0.97	0.80	0.08	1.85	0.52	0.43	0.04	1.00	5.87	4.17	0.43
3	0.03	0.87	0.57	1.47	0.02	0.59	0.39	1.00	0.26	5.68	3.92
4	0.79	0.28	0.20	1.27	0.62	0.22	0.16	1.00	6.99	2.11	1.59
5	0.94	0.26	0.25	1.45	0.65	0.18	0.17	1.00	7.28	1.72	1.75
6	0.74	0.60	0.94	2.28	0.33	0.26	0.41	1.00	3.66	2.52	4.18
7	0.99	0.11	0.78	1.88	0.52	0.06	0.42	1.00	5.87	0.57	4.24
8	0.99	0.15	0.11	1.25	0.79	0.12	0.09	1.00	8.87	1.17	0.87
9	0.80	0.88	0.58	2.25	0.35	0.39	0.26	1.00	3.95	3.75	2.61
10	0.88	0.18	0.03	1.08	0.81	0.17	0.03	1.00	9.04	1.60	0.27

The data-driven aggregate planning MIP now is run and the optimal plan generated for 10 periods, where the length of each period is 7 days and each day has 8 working hours. Table 4.3 shows the optimal FPR model solution using the allowable

WIP patterns for products 1, 2, and 3. The model selects the appropriate product mix in each period, and uses an initial WIP of 12.38 units for product 1. This WIP level is one of the discrete release patterns. The model also chooses an inventory level of 0.34 and 2.83 for product 2 and 3 respectively. While the initial WIP and inventory are parameters in the model, we chose to let these be variables to decrease computation time and ensure feasibility. However, in a practical computation, the actual value initial WIP should be used.

Table 4.3: FPR model output

Period	X			Y			W			I			B			D		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	2.01	3.19	0.00	11.88	0.00	0.00	2.51	3.19	0.00	3.67	0.00	0.49	0.00	0.00	0.00	8.22	0.34	2.35
2	4.00	5.53	1.75	4.11	7.17	0.00	2.40	1.55	1.75	1.91	3.00	0.06	0.00	0.00	0.00	5.87	4.17	0.43
3	1.73	2.07	2.11	4.13	3.61	3.86	0.00	0.00	0.00	5.77	0.94	0.00	0.00	0.00	0.00	0.26	5.68	3.92
4	5.69	0.00	1.97	0.00	0.00	0.00	5.69	0.00	1.97	0.00	0.00	0.00	1.22	1.17	1.59	6.99	2.11	1.59
5	4.82	1.55	3.57	8.11	0.00	3.80	2.40	1.55	1.75	0.00	0.00	0.46	0.38	2.89	0.00	7.28	1.72	1.75
6	4.13	3.61	3.86	4.13	3.61	3.86	2.40	1.55	1.75	0.09	0.00	0.14	0.00	1.80	0.00	3.66	2.52	4.18
7	14.11	2.07	2.11	4.13	3.61	3.86	12.38	0.00	0.00	0.00	1.24	0.00	1.66	0.00	0.23	5.87	0.57	4.24
8	1.91	1.55	1.75	11.88	0.00	0.00	2.40	1.55	1.75	1.35	0.07	0.00	0.00	0.00	1.10	8.87	1.17	0.87
9	7.83	3.98	2.11	4.13	3.61	3.86	6.10	1.91	0.00	1.53	0.00	0.15	0.00	0.07	0.00	3.95	3.75	2.61
10	1.98	1.59	0.00	8.09	3.50	0.00	0.00	0.00	0.00	0.57	1.83	0.00	0.00	0.00	0.12	9.04	1.60	0.27

4.3 Results and Sensitivity Analysis

To obtain insight into how changing the model parameters and the demand pattern affects the behaviour of the FPR-MIP model, and to make things more general, several input scenarios were examined, highlighting their relationship to the model output.

All model variants in this chapter are solved using Gurobi Optimizer 7.0.1TM from within the MPL 5.0TM for Windows environment. The run times for the optimization are based on a 64-bit Windows 10.0TM computer with 24GB RAM and 3.3Ghz clock speed.

4.3.1 Effect of Introducing the CF on the Model

In the first scenario, we compare model output with and without the nonlinear CF estimate. To do so, the CF constraints (4.4), (4.5), and (4.6) are dropped and a new capacity constraint on total production time is added as follows:

$$\sum_{g \in G} TPT_{gm} Y_{gp} \leq T \quad \forall m \in M \quad p \in P \quad (4.9)$$

The aggregate capacity constraint states that the total production time ($TPT_{gm} Y_{gp}$) in each period p should not exceed the length of each planning period T , where TPT_{gm} is the total processing time of product g on machine m , as shown in constraint (4.9). This is the constant level line apparent in Figure 2.3.

Figures 4.1, 4.2, and 4.3 show the effect of the CF on the aggregate model for product 1, product 2, and product 3 respectively. The model with the aggregate capacity constraint is able to satisfy customer demand without holding inventory or creating backorders. On the other hand, the model with the nonlinear CF has quite a few capacity restrictions and uses inventory and backorders. This again underlines the fact that the nonlinear CF gives a more realistic production throughput estimation, which would be lost with just an aggregate capacity constraint.

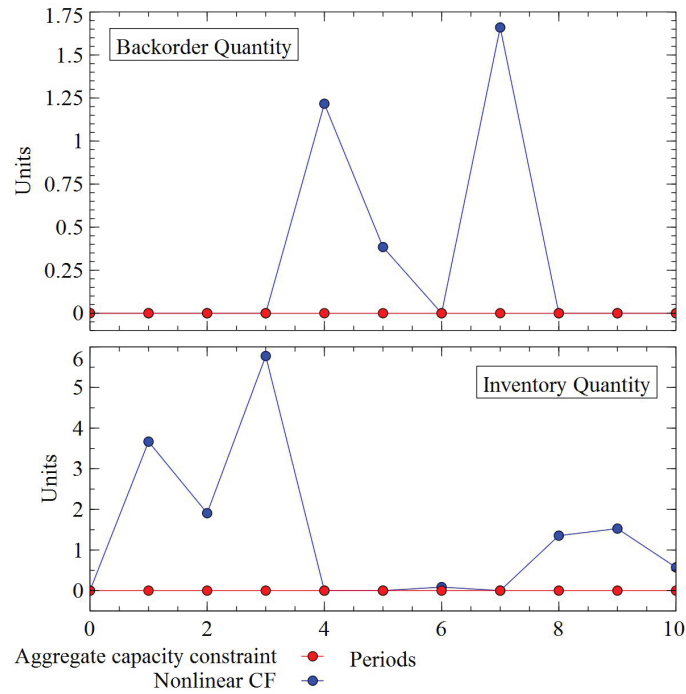


Figure 4.1: CF effect on product 1

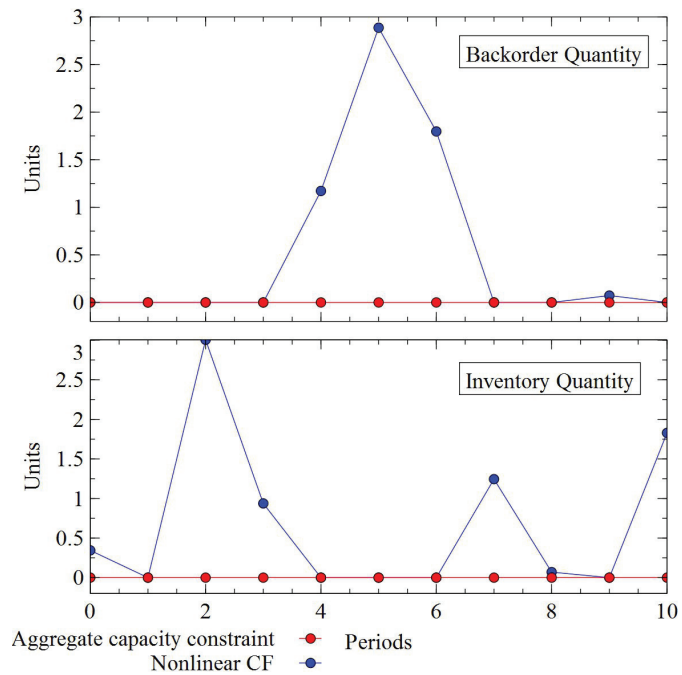


Figure 4.2: CF effect on product 2

4.3.2 Increasing Backorder Costs

This scenario looks at the effect of increasing the backorder cost on inventory levels. These costs are varied from 20 to 160 for the three products. As expected, there

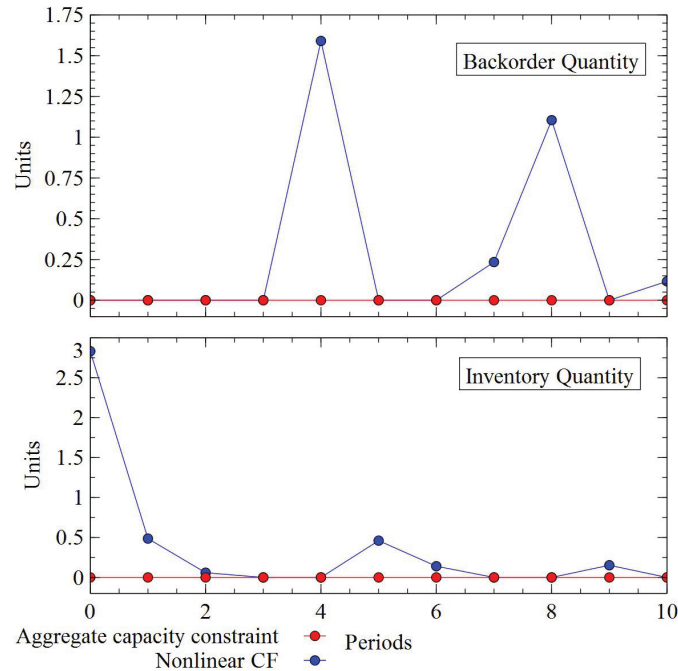


Figure 4.3: CF effect on product 3

is an inverse correlation between backorder cost and backorder level, while there is a positive correlation with inventory level (when the backorder cost goes up, the model chooses to increase inventory to avoid backorders). Figure 4.4 shows these relationships with backorder cost. When the backorder cost increases, the backorder levels decrease and the inventory levels increase with a tapering off effect in both cases.

4.3.3 Effect of Increasing and Decreasing Demand

In this scenario, the demand is increased and decreased by 10%, 20% and 30% for all products in all periods. Table 4.4, shows the behaviour of the FPR-MIP model.

The objective function value generally increases (or decreases) as demand increases (or decreases). This is because it is more expensive to build an efficient production plan when the resource utilization is high, as is the case with increased demand. However, when the demand is 10% higher, the optimal objective function is actually lower than the baseline. This is because the MILP is forced to pick a fixed release pattern and an increase of 10% in demand seems to provide a better match between supply and demand. Figure 4.5 shows the effect of the demand level (as a fraction of

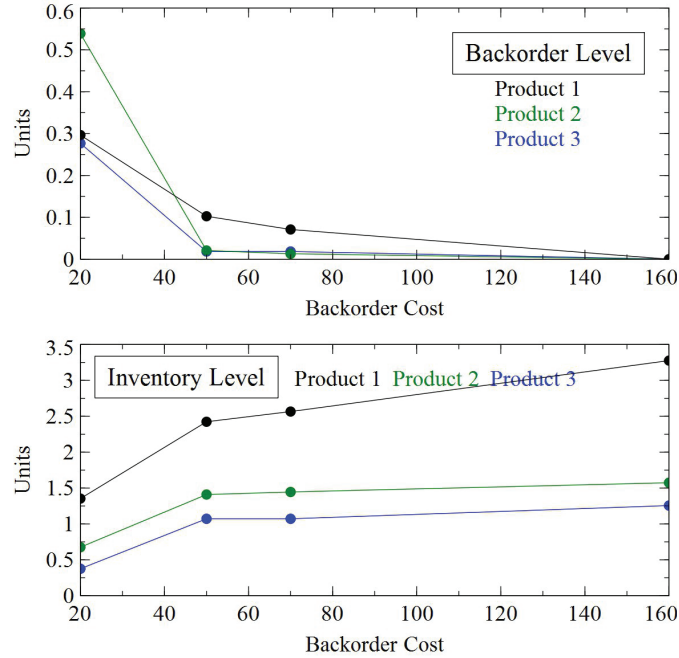


Figure 4.4: Effect of increasing backorder cost

baseline) on backorder and inventory. When this fraction is 1.1 (i.e., when demand is increased by 10%), the MILP is able to find a solution that has the least total cost of backordering and inventory.

Interestingly, the solution time decreases in this example with both increase and decrease in demand. Since the aggregate production planning is an MILP, both lower and higher demands can make the problem easier, as seen in this case. With very low demand in each period, production in the period can match demand, making the problem simple. When demand is very high in each period for all products the simple solution would be to make as much as possible of each, with prioritizing products with higher inventory and backorder costs.

Table 4.4: Effect of demand changes on objective function

	Increase			Original 0%	Decrease		
	30%	20%	10%		10%	20%	30%
Cost	1911.66	1492.25	1295.11	1401.41	1314.14	1282.70	1145.85
Run Time (sec.)	0.08	0.14	0.35	3.14	2.16	1.80	0.86

To generalize the results in Table 4.4, a Monte Carlo simulation was performed

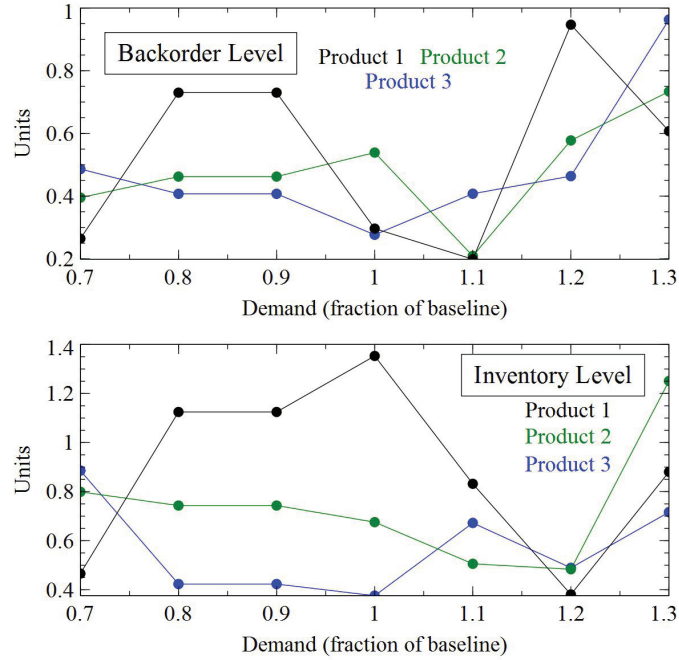


Figure 4.5: Effect of decreasing and increasing the demand on the backorder and inventory level

with randomly generated demand depending on the scenario (+30%, +20%, etc.). The simulation was replicated 100 times to estimate the mean of objective function value and run time. Table 4.5 shows a similar result to Table 4.4. Therefore, the same conclusions can be drawn, i.e., the objective function is lowest for a 10% increase in demand and run time is the lowest for the original demand.

Table 4.5: Generalizing the effect of decreasing and increasing demand on objective function and run time

	Increase			Original 0%	Decrease		
	30%	20%	10%		10%	20%	30%
Obj. Fun. Value	1727.78	1262.41	1122.76	1228.80	1170.08	1104.61	1036.63
Run Time (secs)	0.04	0.45	4.21	20.44	10.95	7.48	5.01

4.3.4 Effect of Step Size and the Planning Horizon Length

This scenario looks at the effect of step size changes on the objective function value and model computation time. The step sizes are varied from 2 to 6 in increments

of 1. Increasing the step size gives more flexibility to the FPR-MIP model since the WIP increments are smaller and the MILP solution can find better tuned solutions.

As seen in Table 4.6, the objective function value decreases with step size for the 10, 20, and 30 period models. In the case of the 10 period model, an optimal solution is found for all step sizes. However, for the 20 and 30 period models, the optimal solution took a long time, exceeding 10 hours sometimes. Therefore, we report the objective function solution for a 15% optimality gap. It may be seen that the objective function value is lower, as step size increases.

It may also be observed for the 10 period model, that the run time increases as step size goes up from 2 to 5. However, it drops significantly when the step size is 6. For a 15% MILP gap, the same phenomenon is observed for the 20 period problem while it increases with step size for all cases in the 30 period problem.

Since MILP solution times depend greatly on problem data (actual demand, in particular), we generalize the relationship between step size and objective function value (or run time) in the next section.

Table 4.6: Step size and planning horizon effect on the objective function

		Step Size				
		2	3	4	5	6
10 Periods	Objective Function	1552.36	1401.41	1320.41	1292.76	1267.37
	Run Time (secs)	0.26	3.14	16.74	71	1.29
	Gap	0%	0%	0%	0%	0%
20 Periods	Objective Function	3092.76	2706.72	2549.96	2520.87	2440.57
	Run Time (secs)	0.07	7.45	21.51	133.2	70.8
	Gap	15%	15%	15%	15%	15%
30 Periods	Objective Function	4634.67	3997.11	3787.82	3734.88	3582.58
	Run Time (secs)	0.64	21.55	41.97	1696.8	1715.4
	Gap	15%	15%	15%	15%	15%

4.3.4.1 Generalizing the Effect of Step Size on the Operational Cost

In order to generalize the relationship between step size and the objective function value (or run time), we conduct a Monte Carlo simulation where the demand is generated based on a random number between 0 and 1 for 10 periods for the three products. The simulation is replicated 100 times to estimate the mean and standard

deviation of the total cost and computational time for the different step sizes.

In the previous section 4.3.4, it was hypothesized that as the step size increases, the objective function value should decrease because increasing the step size gives more flexibility to the FPR-MIP model to match supply and demand and to reduce inventory and backorder costs. In order to test the hypothesis, the confidence intervals limits are developed from the following equations [76]:

$$LCL = \bar{x} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \quad (4.10)$$

$$UCL = \bar{x} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \quad (4.11)$$

where LCL and UCL are the lower and upper confidence limits, and $z_{\alpha/2} = 1.96$ from z -table for a 95% confidence interval.

Table 4.7: Objective function confidence limits

Step Size	\bar{x}	σ	Confidence Interval	
			LCL	UCL
2	1437.83	44.12	1429.18	1446.48
3	1284.62	32.94	1278.17	1291.08
4	1224.27	25.13	1219.34	1229.20
5	1197.86	25.43	1192.87	1202.84
6	1181.10	27.14	1175.79	1186.42

As seen in Table 4.7, the 95% confidence intervals do not overlap. Thus, it may be concluded that increasing step size reduces the objective function value (i.e., better solutions are found as the step size is increased).

The same reasoning is extended for run time (in seconds) in Table 4.8, which is developed in a similar manner for a 95% confidence interval using 100 replications for randomly generated demand. As can be seen, increasing the step size increases run time (the confidence intervals do not overlap). The main reason behind that is increasing the step size increases the number of WIP pattern choices, making the problem more complex.

Hence, the essential trade-off is that by increasing step size, the solution quality improves but takes longer to obtain.

Table 4.8: Run time confidence limits

Step Size	\bar{x}	σ	Confidence Intervals	
			LCL (s)	UCL (s)
2	0.6077	0.1112	0.5859	0.6295
3	4.9810	2.2033	4.5492	5.4129
4	19.583	8.6012	17.897	21.269
5	72.266	45.732	63.302	81.229
6	204.13	156.48	173.46	234.81

4.4 Supplementary Constraints and Variables to Decrease Computation Time

This section presents supplementary constraints and variables which may be added to the FPR-MIP formulation mentioned in section 4.1, to decrease model run time. This set of supplementary constraints limits production in each period to be within a fraction of demand in that period and penalizes production below or above demand.

$$Y_{gp} = d_{gp} + \delta_{gp}^+ - \delta_{gp}^- \quad (4.12)$$

$$\delta_{gp}^+ \leq f \bar{d}_{gp} \quad (4.13)$$

$$\delta_{gp}^- \leq f \bar{d}_{gp} \quad (4.14)$$

Effectively, δ_{gp}^+ and δ_{gp}^- are the backorder and inventory contributions in period p , respectively. f represents the fraction of demand variation allowed in each period, and is applied to the average demand \bar{d}_{gp} over all periods p .

In the objective function δ_{gp}^+ and δ_{gp}^- are penalized, by adding the following term:

$$M (\delta_{gp}^+ + \delta_{gp}^-) \quad (4.15)$$

The choice of the penalty parameter M needs to be chosen carefully. When it is very small, the model run time is high, but the cost of the solution is closer to the optimal of the original problem. This is because, larger deviations of production from the average demand can be tolerated better. When M is large, the run time is low but the values of δ_{gp}^+ and δ_{gp}^- are smaller and the cost of the solution is further away from the optimal of the original problem.

Similarly, when the value of f gets higher, there is greater latitude in the δ_{gp}^+ and δ_{gp}^- variables. Therefore, the solutions get better but take longer to find.

These trends are shown in Figure 4.6 for the case in Table 4.6 with 30 periods and a step size of 6. In this figure, the penalty costs (M) range from 0 to 125 and the value of f ranges from 100% to 250% of average demand. The one exception to the rule is the case of $M = 125$ and $f = 250\%$, where the run time is slightly lower than for an equivalent $f = 200\%$. This could be because of inherent uncertainties with the MILP solution process. It should be noted that the values on the Y-axis (run time) are shown using the logarithmic scale.

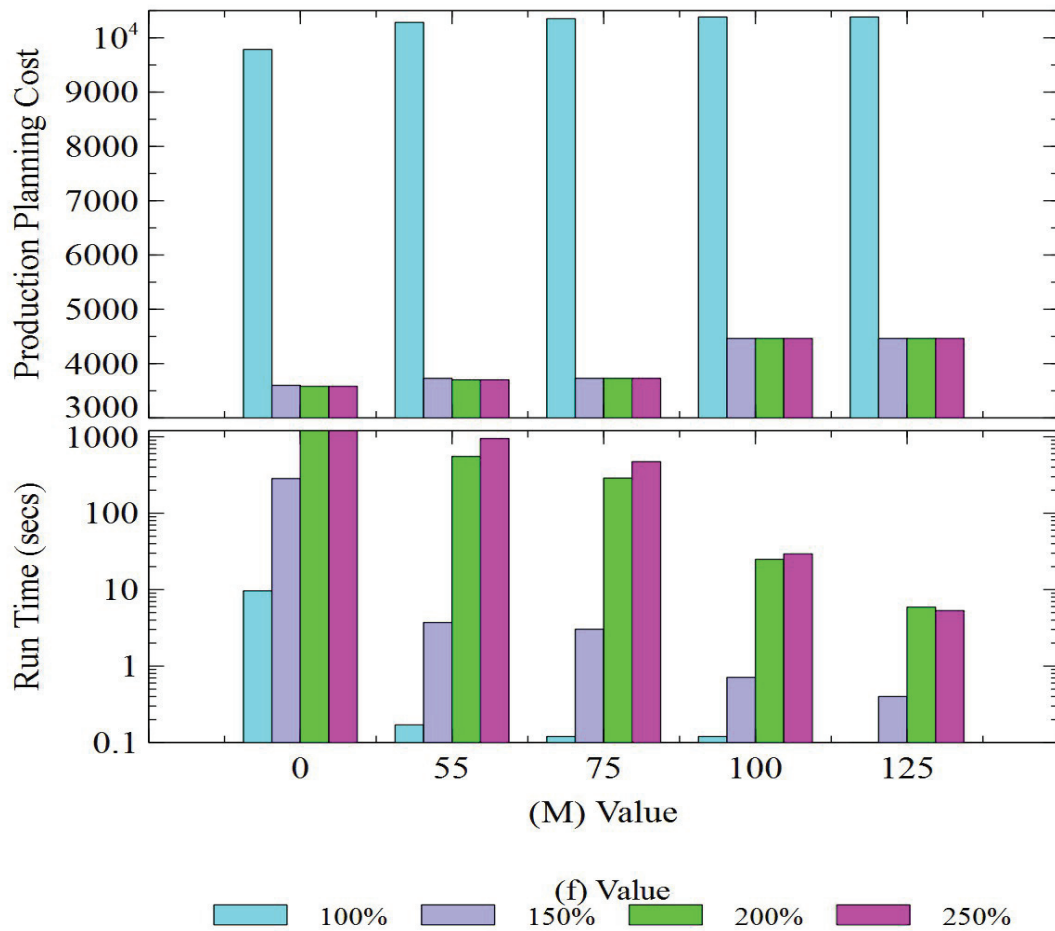


Figure 4.6: Objective function value and the run time

4.5 Conclusion

This chapter presented a data-driven aggregate production planning model to incorporate the congestion effects in multi-product manufacturing networks. While the data-driven approach was illustrated using MVA for closed-queueing manufacturing networks such as the CONWIP, any other estimation method could also have been applied. These include: analytical estimates, numerical methods, and the simulation approach.

The model behavior was examined by changing certain parameters such as backorder cost, demand, etc. Increasing backorder cost decreases the backorder level while increasing inventory to compensate for the decrease. The objective function value generally increases as demand goes up.

It was seen statistically that as the number of steps increases (i.e., step size is small), the solution quality increases (i.e., the objective function value decreases). However, a small step size makes the problem harder to solve (i.e., the computation time increases). Also, run times can be improved at the expense of solution accuracy using the supplementary constraints and variables shown in Section 4.4.

An interesting research question is whether the model presented can be extended to include decision points that lie between the point estimates of the clearing function. This will be explored in the next chapter.

Chapter 5

Cubic Approximation (CA) Approach

The CA method may be used to find a solution to the aggregate planning model in the previous chapter such that the WIP values in each period are continuous. This is done by first identifying the WIP step size. When the WIP step size is small, the approximation is more accurate.

Figure 5.1 shows the approximation graphically when there are three products in the system. For a larger number of products, the method is still valid, but the visualization can no longer be graphical.

For three products, the WIP levels for each product from (0, 0, 0) to (1, 1, 1) is seen in the small cube shown in the left of Figure 5.1. Since the WIP values of different products may have different ranges, the cuboid in the centre of the figure shows product WIP in the range (0, 12.4), (0, 9.2), and (0, 4.2), for products 1, 2, and 3. For greater accuracy, this cube can be divided into eight smaller cuboids using a step size of 2, as seen in the centre cuboid of Figure 5.1. The size of the smaller cuboids is based on the accuracy requirements and these cubes can further subdivided if necessary, or a different step size such as 3 or 5 could be chosen.

The throughput levels are calculated using MVA for the corner points of the cuboids as can be seen in the cuboids on the right. The CA aggregate planning model formulated below allows for WIP combinations in continuous space, as opposed to being limited to discrete points (or patterns) in the previous chapter.

5.1 CA Model

The CA parameters and decision variables are described in the following constraints:

Indices :

- c Cuboid index,
- g Product index,
- p Period index.

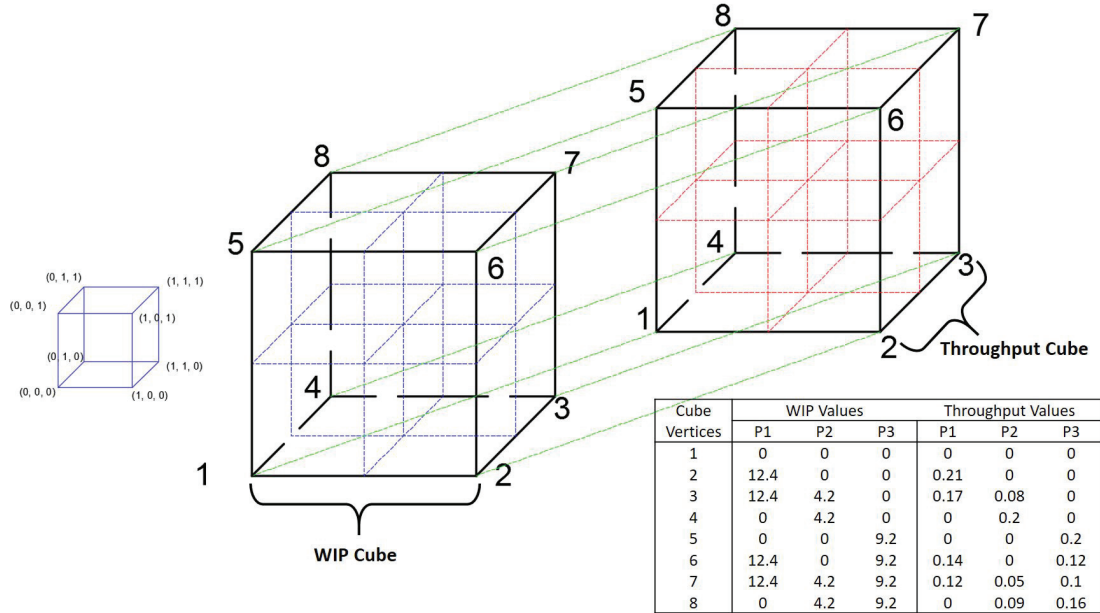


Figure 5.1: Cubic representation of product mix with corresponding throughput

Parameters :

- x_g Raw material release cost of product g ,
- w_g WIP holding cost of product g ,
- h_g Inventory holding cost of product g ,
- b_g Backorder cost of product g ,
- d_{gp} Demand of product g in period p ,
- ∇_g^c Throughput gradient of product g in cuboid c .
- TH_g^c Throughput of product g at the lower left hand corner point of cuboid c ,
- LB_{gp}^c WIP level of product g at the lower left hand corner point of cuboid c ,
- UB_{gp}^c WIP level of product g at the upper right hand corner point of cuboid c ,
- M Big M (a very large number).

Decision variables :

- Y_{gp} Production quantity of product g in period p ,
- X_{gp} Amount of raw material released of product g in period p ,
- W_{gp} WIP level of product g in period p ,
- W_{gp}^c WIP level of product g in period p of the selected cuboid c ,
- I_{gp} Finished goods inventory level of product g in period p ,
- ζ_p^c Binary variable to ensure exactly one cuboid c is selected in each period p .

Objective function :

$$\min \sum_{g \in G} \sum_{p \in P} (x_g X_{gp} + w_g W_{gp} + h_g I_{gp} + b_g B_{gp}) \quad (5.1)$$

Subject to :

$$W_{gp} = W_{g,p-1} + X_{gp} - Y_{gp} \quad \forall g \in G, p \in P \quad (5.2)$$

$$I_{g,p} = I_{g,p-1} + Y_{gp} - B_{g,p-1} + B_{gp} - d_{gp} \quad \forall g \in G, p \in P \quad (5.3)$$

$$Y_{gp} \leq TH_g^c + (W_{gp}^c - LB_g^c) \nabla_g^c + M(1 - \zeta_p^c) \quad \forall g \in G, p \in P \quad (5.4)$$

$$LB_g^c \zeta_p^c \leq W_{gp}^c \leq UB_g^c \zeta_p^c + M(1 - \zeta_p^c) \quad \forall g \in G, p \in P \quad (5.5)$$

$$\sum_c \zeta_p^c = 1 \quad \forall c \in C, p \in P \quad (5.6)$$

$$Y_{gp}, I_{gp}, B_{gp}, W_{gp}, X_{gp} \geq 0 \quad \forall g \in G, p \in P \quad (5.7)$$

$$\zeta_p^c \in \{0, 1\} \quad \forall c \in C, p \in P \quad (5.8)$$

The essential difference between this model and that in the previous chapter is due to the difference in how the nonlinear CF is modelled using constraints (5.4), (5.5), and (5.6). Constraint (5.4) ensures that the production does not exceed the CF throughput estimate. Note that it is simply a generalization of the piecewise linearization method used for single-variables CFs by [47, 3]. This constraint is only active for a cuboid that is active. Constraint (5.5) limits the WIP level to the dimensions of the chosen cuboid. Finally, constraint (5.6), ensures that only one cuboid is active. This model has $5|G||P|$ continuous variables, $|C||P|$ binary variables, and $4(|G|+|C|)|P|$ constraints.

Let ω_g^i and τ_g^i be the WIP level and throughput of product g for a corner point i

in any cuboid respectively. Also, let E_g^c be the set of edges over which the gradient ∇_g^c of product g in cuboid c is estimated.

Figure 5.2 shows the gradient of product 1 is estimated for the edges on the X-axes (i.e., lines coloured with in cyan). Similarly, for products 2 and 3, the gradients are estimated for the edges on the Y-axes (i.e., lines coloured in red) and Z-axes (i.e., lines coloured in green) respectively.

Thus:

$$E_g^c = \{(2, 1), (3, 4), (6, 5), (7, 8)\} \quad g = 1$$

$$E_g^c = \{(4, 1), (3, 2), (8, 5), (7, 6)\} \quad g = 2$$

$$E_g^c = \{(5, 1), (6, 2), (7, 3), (8, 4)\} \quad g = 3$$

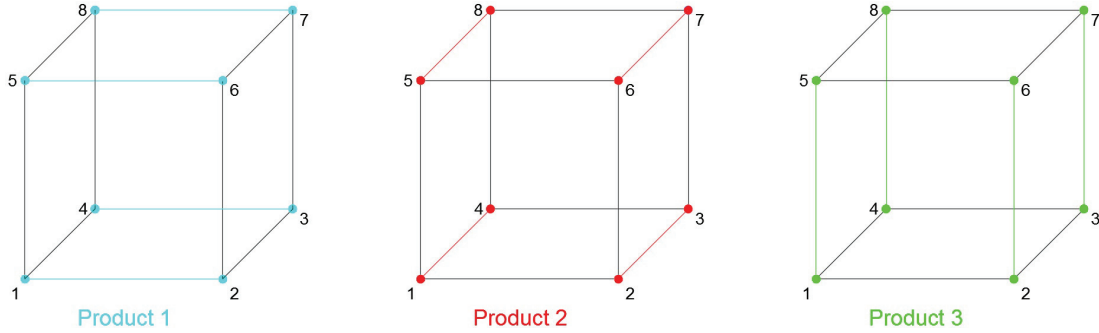


Figure 5.2: Slope of each products

Thus, the gradient ∇_g^c in constraint (5.4) is computed using the following equation:

$$\nabla_g^c = \sum_{(ij) \in E_g^c} \left(\frac{\tau_g^j - \tau_g^i}{\omega_g^j - \omega_g^i} \right) / |E_g^c| \quad (5.9)$$

5.2 Numerical Example

The example presented in Chapter 3 is used to implement the CA aggregate planning model, which can be built as follows:

1. The first step is to determine the size of the large cuboid. For this, the product mix combinations $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, corresponding to the

maximum open network arrival rates for the three products (one at a time) are chosen from Table 4.1. The corresponding WIP for the three products from the table (rows 11, 5, and 2 respectively) are 12.38, 4.17, and 9.22. In fact, the large cuboid in the center of Figure 5.1 is of these dimensions.

2. The second step is to choose a step size. Using an example step size of 3, 27 cuboids with their lower and upper corner points shown in Table 5.1 result. Table 5.1 also shows the CA throughput (TH_g^c) for the closed queuing network using MVA and the product slopes (∇_g^c).

The essence of the CA method is to choose one of the smaller inside cuboids. The selection between the smaller cuboids are based on the MIP model, where the WIP solution point is within the cuboid, without being restricted to only the cuboid corner points.

Table 5.1: CA throughput, slope, lower corner, and upper corner

Set	Cubic Lower Corner			Cubic Upper Corner			Throughput (unit\week)			Slope		
	LB1	LB2	LB3	UB1	UB2	UB3	TH1	TH2	TH3	$\nabla 1$	$\nabla 2$	$\nabla 3$
1	0.00	0.00	0.00	4.13	1.39	3.07	0.00	0.00	0.00	1.63	2.76	2.09
2	4.13	0.00	0.00	8.25	1.39	3.07	8.12	0.00	0.00	0.63	1.75	1.56
3	8.25	0.00	0.00	12.38	1.39	3.07	10.76	0.00	0.00	0.31	1.27	1.22
4	0.00	1.39	0.00	4.13	2.78	3.07	0.00	5.63	0.00	1.41	1.77	1.84
5	4.13	1.39	0.00	8.25	2.78	3.07	6.92	3.28	0.00	0.72	0.73	1.42
6	8.25	1.39	0.00	12.38	2.78	3.07	9.59	2.24	0.00	0.34	1.02	1.11
7	0.00	2.78	0.00	4.13	4.17	3.07	0.00	8.95	0.00	1.25	1.20	1.63
8	4.13	2.78	0.00	8.25	4.17	3.07	6.02	5.58	0.00	0.61	0.99	1.26
9	8.25	2.78	0.00	12.38	4.17	3.07	8.65	3.97	0.00	0.35	0.83	1.02
10	0.00	0.00	3.07	4.13	1.39	6.15	0.00	0.00	7.90	1.31	2.07	0.79
11	4.13	0.00	3.07	8.25	1.39	6.15	6.34	0.00	5.74	0.58	1.36	0.75
12	8.25	0.00	3.07	12.38	1.39	6.15	8.85	0.00	4.37	0.34	1.03	0.67
13	0.00	1.39	3.07	4.13	2.78	6.15	0.00	3.98	6.93	1.15	1.36	0.79
14	4.13	1.39	3.07	8.25	2.78	6.15	5.51	2.45	5.08	0.65	0.59	0.73
15	8.25	1.39	3.07	12.38	2.78	6.15	8.02	1.76	3.94	0.35	0.85	0.63
16	0.00	2.78	3.07	4.13	4.17	6.15	0.00	6.42	6.07	1.03	0.98	0.77
17	4.13	2.78	3.07	8.25	4.17	6.15	4.88	4.24	4.53	0.56	0.83	0.69
18	8.25	2.78	3.07	12.38	4.17	6.15	7.33	3.16	3.58	0.35	0.71	0.61
19	0.00	0.00	6.15	4.13	1.39	9.22	0.00	0.00	10.25	1.10	1.68	0.38
20	4.13	0.00	6.15	8.25	1.39	9.22	5.22	0.00	8.21	0.53	1.12	0.44
21	8.25	0.00	6.15	12.38	1.39	9.22	7.52	0.00	6.60	0.34	0.87	0.44
22	0.00	1.39	6.15	4.13	2.78	9.22	0.00	3.13	9.43	1.03	0.62	0.42
23	4.13	1.39	6.15	8.25	2.78	9.22	4.58	1.94	7.47	0.53	0.88	0.45
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
27	8.25	2.78	6.15	12.38	4.17	9.22	6.35	2.62	5.60	0.35	0.62	0.42

The optimal plan is generated for 10 periods, where the length of each period is 7 days and each day has 8 working hours. The demand for the three products in each time period is based on a random number generated between 0 and 1. The product mix is normalized as described in Chapter 4 and the normalized numbers are multiplied by the corresponding arrival rate for each product and the weekly working hours, to obtain the demand in units per week, as shown in Table 5.2.

The CA model is solved for cost coefficients of raw material, WIP, inventory, and backorder equal to 3, 7, 15, and 20, respectively.

Table 5.2: Generated demand

Periods	Random Number			Sum	Normalized RN			Sum	Demand (units\week)		
	P1	P2	P3		P1	P2	P3		P1	P2	P3
1	0.75	0.04	0.23	1.02	0.73	0.04	0.23	1.00	8.22	0.34	2.35
2	0.97	0.80	0.08	1.85	0.52	0.43	0.04	1.00	5.87	4.17	0.43
3	0.03	0.87	0.57	1.47	0.02	0.59	0.39	1.00	0.26	5.68	3.92
4	0.79	0.28	0.20	1.27	0.62	0.22	0.16	1.00	6.99	2.11	1.59
5	0.94	0.26	0.25	1.45	0.65	0.18	0.17	1.00	7.28	1.72	1.75
6	0.74	0.60	0.94	2.28	0.33	0.26	0.41	1.00	3.66	2.52	4.18
7	0.99	0.11	0.78	1.88	0.52	0.06	0.42	1.00	5.87	0.57	4.24
8	0.99	0.15	0.11	1.25	0.79	0.12	0.09	1.00	8.87	1.17	0.87
9	0.80	0.88	0.58	2.25	0.35	0.39	0.26	1.00	3.95	3.75	2.61
10	0.88	0.18	0.03	1.08	0.81	0.17	0.03	1.00	9.04	1.60	0.27

The optimal aggregate production plan can be seen in Table 5.3, with an almost perfect demand match that has no backorders and a small inventory after period 1 for product 1 of about 0.04 units. **Note:** The optimal solution requires an initial WIP of 4.34, 0.20, and 1.51 units for product 1, product 2, and product 3.

Table 5.3: CA model output

Period	X			Y			W			I			B			D		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	8.04	1.54	1.07	8.25	0.34	2.35	4.13	1.39	0.23	0.04	0.00	0.00	0.00	0.00	0.00	8.22	0.34	2.35
2	1.89	4.20	2.33	5.83	4.17	0.43	0.18	1.42	2.13	0.00	0.00	0.00	0.00	0.00	0.00	5.87	4.17	0.43
3	4.20	5.47	2.81	0.26	5.68	3.92	4.13	1.21	1.02	0.00	0.00	0.00	0.00	0.00	0.00	0.26	5.68	3.92
4	6.99	1.88	1.69	6.99	2.11	1.59	4.13	0.98	1.12	0.00	0.00	0.00	0.00	0.00	0.00	6.99	2.11	1.59
5	5.39	1.65	2.63	7.28	1.72	1.75	2.24	0.91	2.00	0.00	0.00	0.00	0.00	0.00	0.00	7.28	1.72	1.75
6	5.02	1.82	4.21	3.66	2.52	4.18	3.61	0.21	2.03	0.00	0.00	0.00	0.00	0.00	0.00	3.66	2.52	4.18
7	7.60	1.04	2.77	5.87	0.57	4.24	5.33	0.67	0.56	0.00	0.00	0.00	0.00	0.00	0.00	5.87	0.57	4.24
8	5.97	1.87	1.56	8.87	1.17	0.87	2.43	1.36	1.25	0.00	0.00	0.00	0.00	0.00	0.00	8.87	1.17	0.87
9	7.13	3.31	1.53	3.95	3.75	2.61	5.60	0.91	0.17	0.00	0.00	0.00	0.00	0.00	0.00	3.95	3.75	2.61
10	3.44	0.68	0.10	9.04	1.60	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.04	1.60	0.27

5.3 Effect of Step Size and the Planning Horizon Length

To obtain an insight into how changing the step size affect the behaviour of the CA model, the size of the smaller cuboids (which represent WIP release patterns) could be changed based on size.

We consider step sizes in a range from 2 to 6 in increments of 1. It is expected that increasing the step size gives more flexibility to the MIP model to satisfy demand at a lower cost. As the grid size gets smaller (number of steps increase), there are more cube corner points, which increases accuracy.

For sake of exposition, let us assume that the graphical solution for a problem is as illustrated in Figure 5.3, where the WIP points shown on a two-dimensional plane for three different step size are as shown in cyan for step size 3, red for step size 6, and green for step size 9. The grid lines are the dashed lines in gray, while the X and Y axes are shown in black. It can be clearly seen that the points in step size 9 are the closest to a grid point, followed by step size 6 and subsequently by step size 3. Therefore, increasing the step size provides a more accurate approximation.

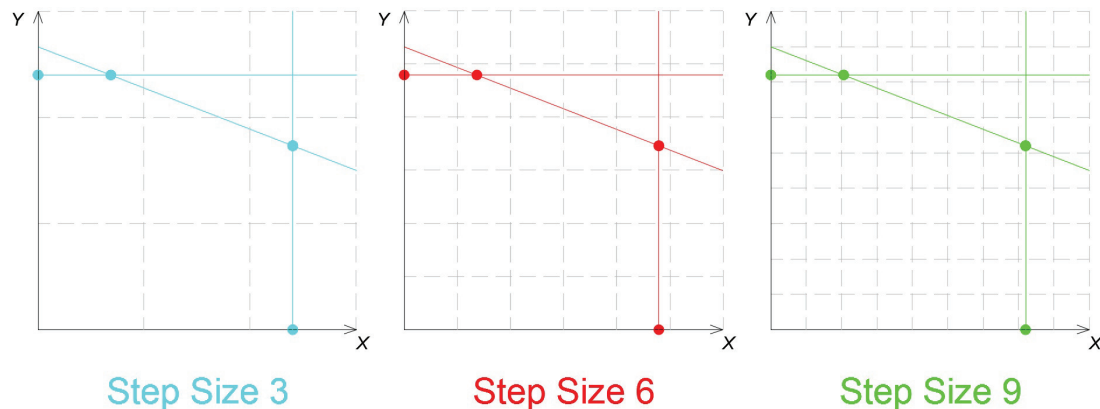


Figure 5.3: Step size accuracy

However, the step size and the computational time are expected to be positively correlated, i.e., when step size is increased, the computational time is also expected to increase.

Table 5.4 shows the effect of step size on the objective function and run time to optimality for different step size and time period combinations. In the case of the 10 period model, an optimal solution is found for all step sizes in reasonable time. However, for the 20 and 30 period models, the optimal solution exceeded 10 hours

for larger step sizes. Therefore, we report the objective function solution for a 15% optimality gap, as in the previous chapter.

It may be seen that the objective function value is lower, as step size increases. Also, for the 10 period model, the run time increases as step size goes up from 2 to 6. For a 15% MILP gap, the same phenomenon is true for the 20 and 30 period problems.

Table 5.4: Step size and planning horizon effect on the objective function

		Step Size				
		2	3	4	5	6
10 Periods	Objective Function	798.24	701.75	661.17	644.58	642.20
	Run Time (secs)	1.02	4.95	47.35	268	5687
	Gap	0%	0%	0%	0%	0%
20 Periods	Objective Function	1558.97	1453.36	1290.99	1249.61	1242.66
	Run Time (secs)	0.94	8.87	125.00	706	31035
	Gap	15%	15%	15%	15%	15%
30 Periods	Objective Function	2394.80	2078.90	1889.68	1835.72	1831.45
	Run Time (secs)	1.62	21.80	183.00	1421	51562
	Gap	15%	15%	15%	15%	15%

5.4 Supplementary Constraints and Variables to Decrease Computation Time

To decrease the computational time of the MIP the same supplementary constraints 4.12 to 4.15 in Section 4.4 are used in this section to the 30 period problem with step size 6 in Table 5.4 .

Figure 5.4 shows the sensitivity of the solution cost and run time to the parameters M and f . As in Section 4.4, as the value of M goes up, the solution worsens, but the run time reduces. The same trend can be seen with respect to f , with the following exceptions: $M = 55$, $f = 150\%$ and $M = 75$, $f = 200\%$. Again, the Y-axis in the bottom graph (for run time) is shown on a logarithmic scale.

5.5 Comparing CA with the FPR Approach

This section looks at the differences between the CA and the FPR aggregate production planning model outputs for a randomly generated 10-period problem. In both

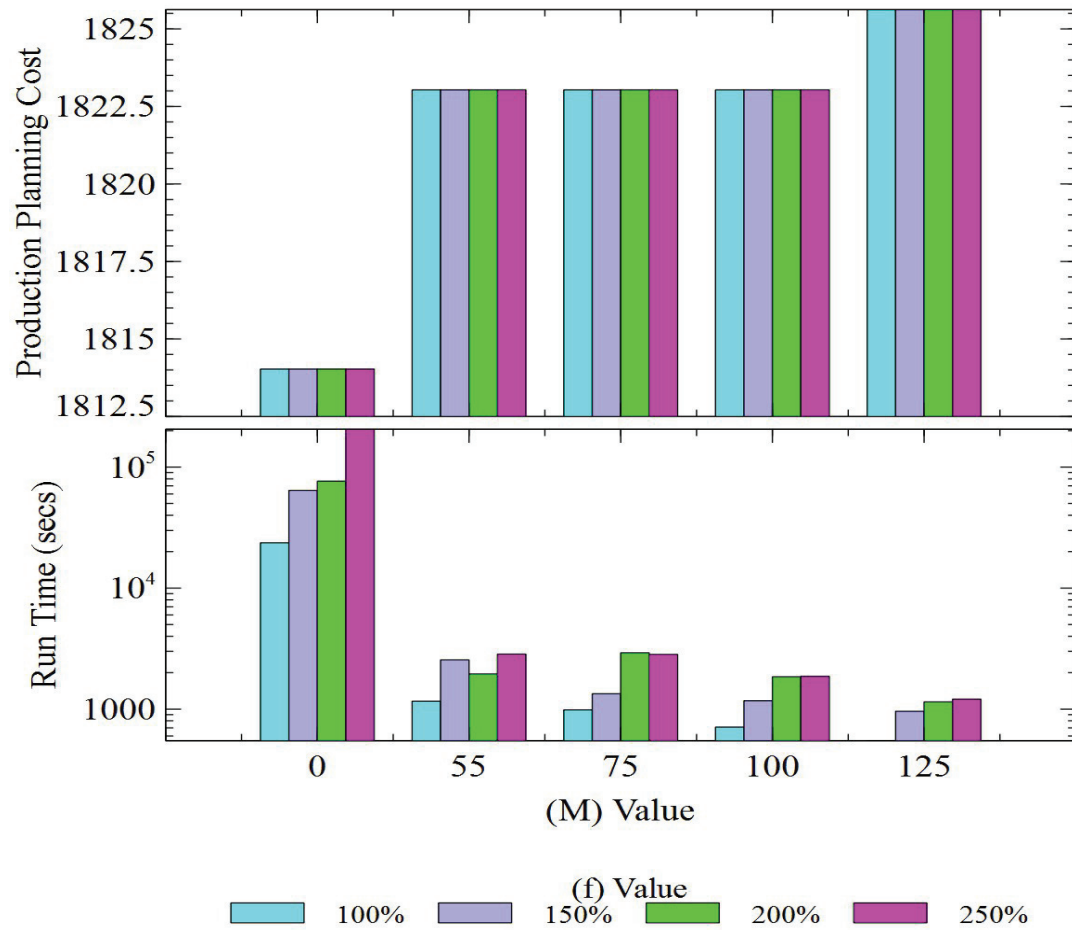


Figure 5.4: Objective function value and the run time

cases, the step size is 3. Figures 5.5, 5.6, and 5.7 show the difference in production, inventory, and backorder level.

It can be seen in Figure 5.5 that there is production in every planning period in the CA solution. However, there are some periods with no production in the FPR approach. This is because FPR reduces the feasible solution space and it is harder to match supply and demand. For the same reason, the CA approach holds no inventory, as shown in Figure 5.6. On the other hand, some inventory is held in the FPR approach in almost every period. The same phenomena can be seen for backorder quantity in Figure 5.7. Since the formulation in both approaches consider inventory and backorder costs, the objective function value in CA is expected to be lower than in the FPR approach.

Figure 5.8 illustrates that there is an inverse relationship between step size and best found objective function value (for both the CA and FPR approaches). The best found objective value is the optimal for the 10 period problems and a solution with 15% gap for the 20 and 30 period problems. As the step size increases, the objective function value of the solution decreases, though the marginal improvement reduces. The figure also shows that the best found objective function value is superior in the CA model.

However, there is a significant increase in run time to optimality (again, for both the CA and FPR approaches), as can be seen in Figure 5.9, 5.10, and 5.11 for the 10, 20, and 30 time period cases; Here, the step size is divided across two graphs; the one on the top shows a step size from 2 to 4 and the bottom is for step sizes 5 and 6. In general, the CA takes longer to run except in the 30 period case in Figure 5.11; here it is almost the same as FPR in case of step size 3, and it slightly lower in case of step size 5. This could be because only one instance of the problem is being solved.

Therefore, it appears that the CA method yields better solutions, but takes longer to run to solve.

5.6 Conclusion

This chapter presented an aggregate production planning model to incorporate the congestion effects in multi-product manufacturing networks. The MVA approach was applied to estimate the CF using CA approach. The product mix concept used to

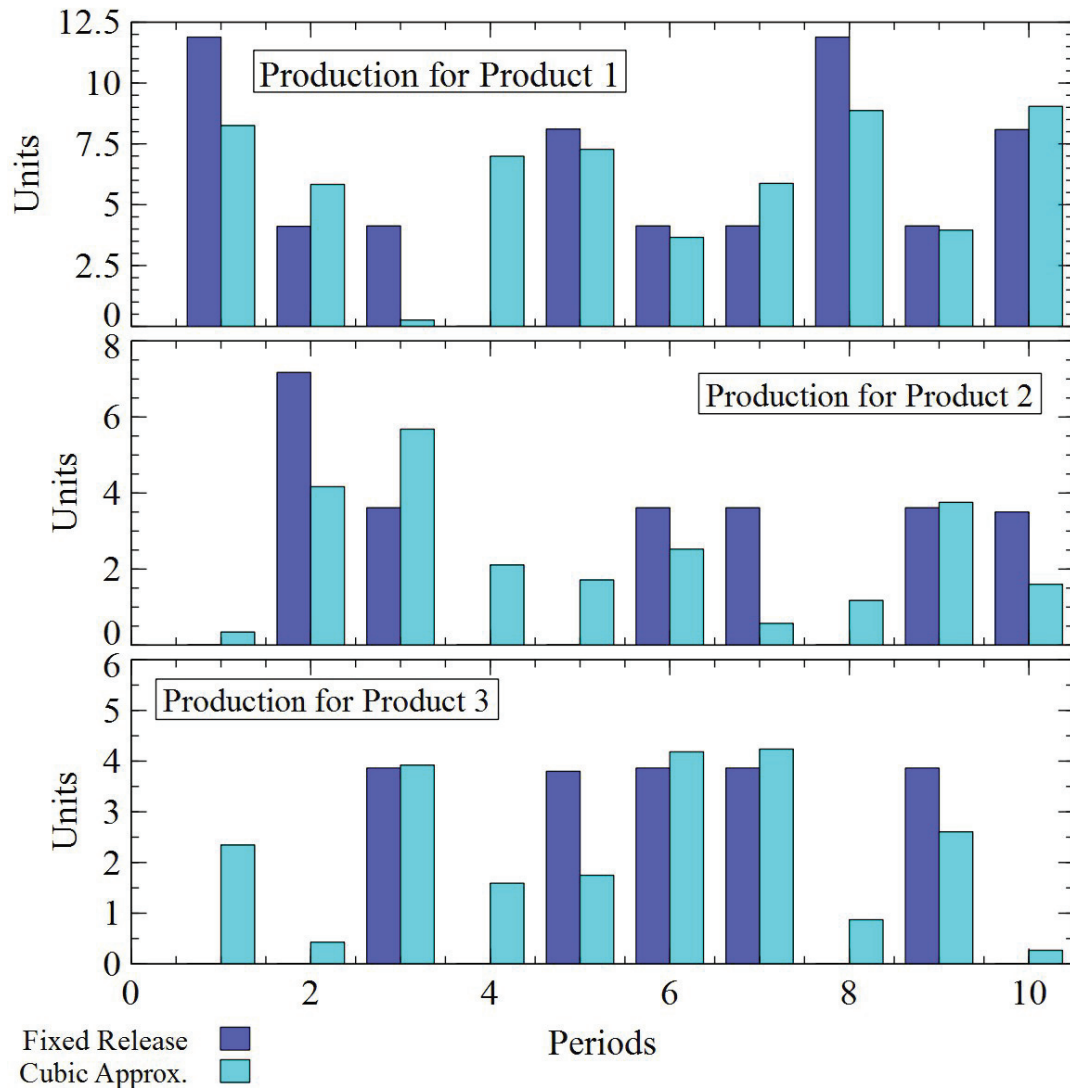


Figure 5.5: Aggregate production planning comparison for production

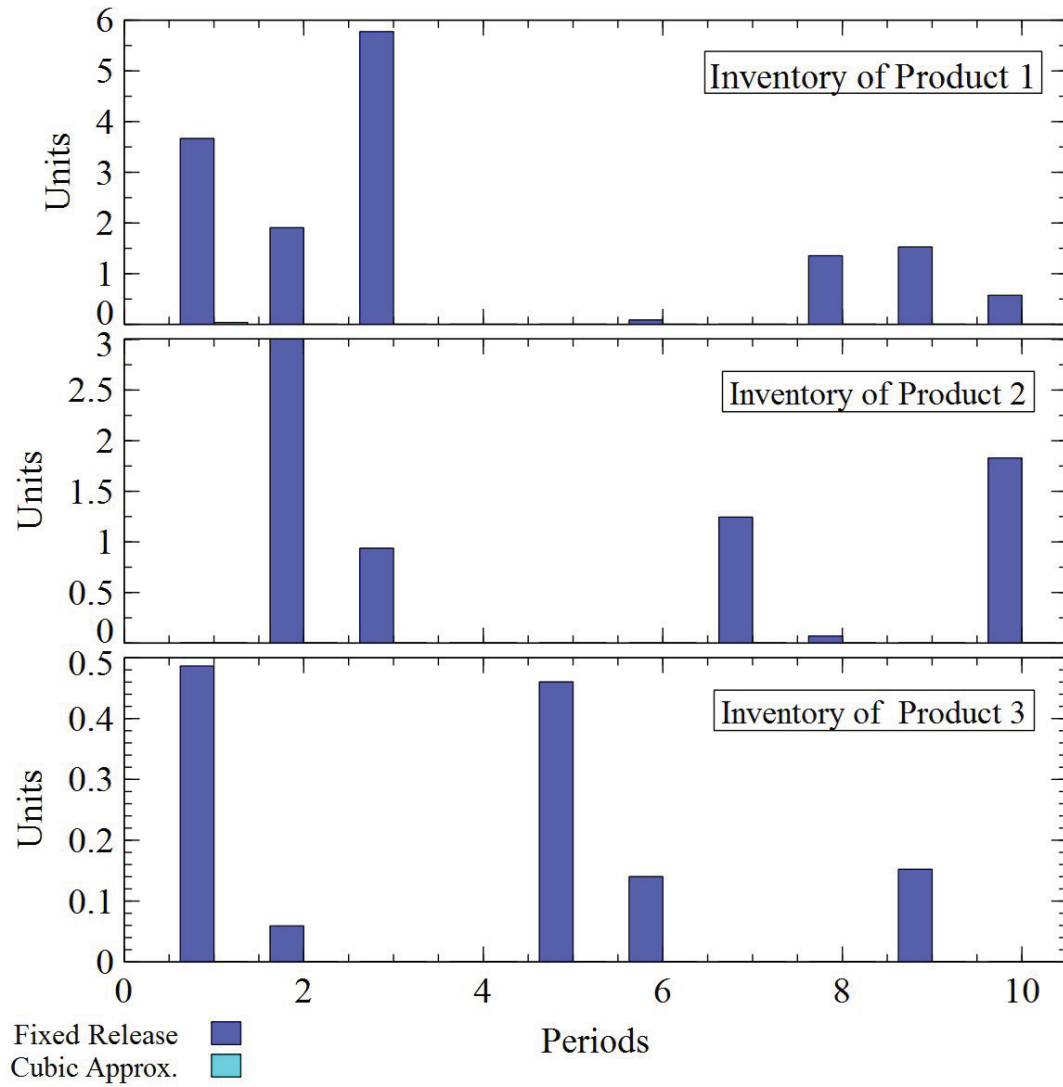


Figure 5.6: Aggregate production planning comparison for inventory

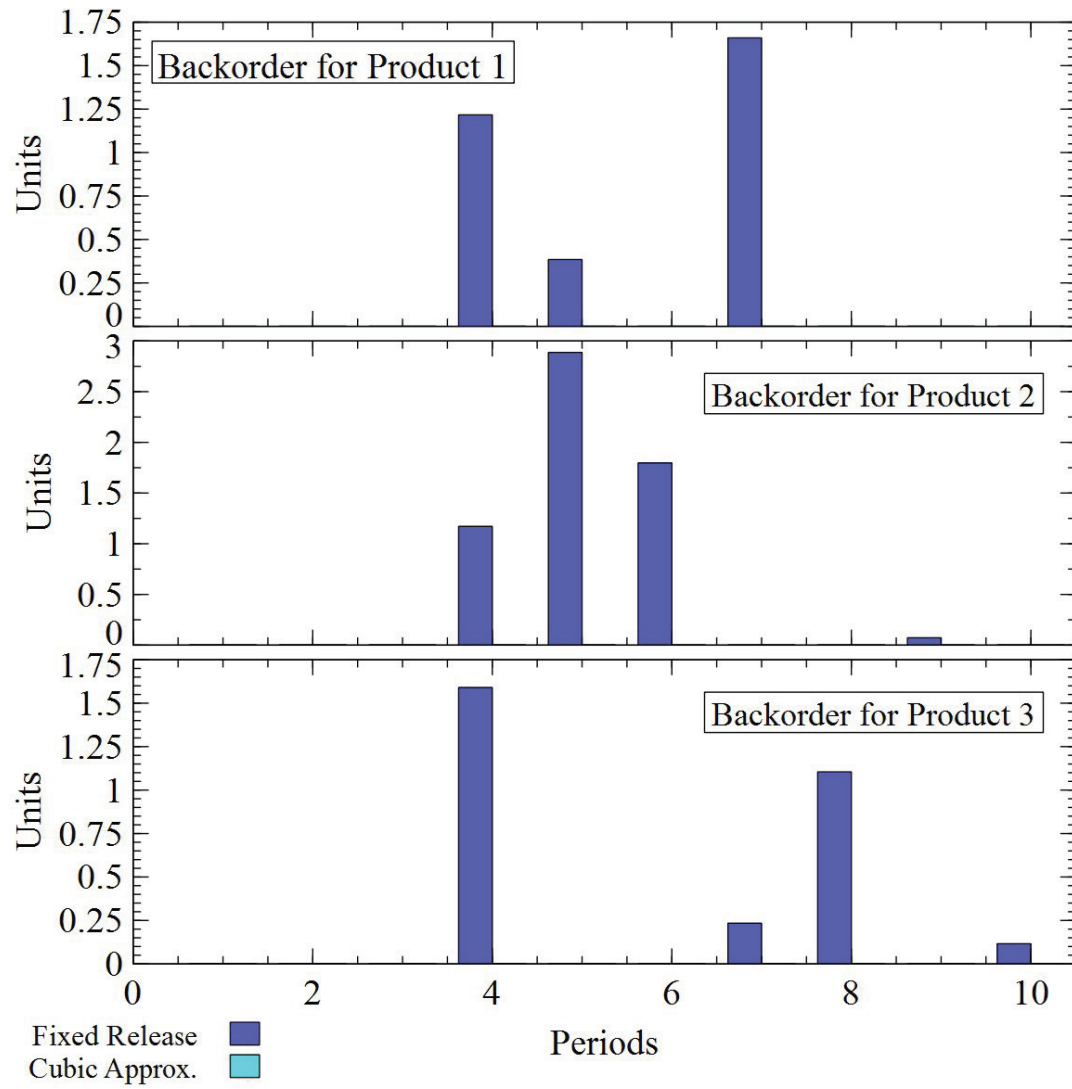


Figure 5.7: Aggregate production planning comparison for backorder

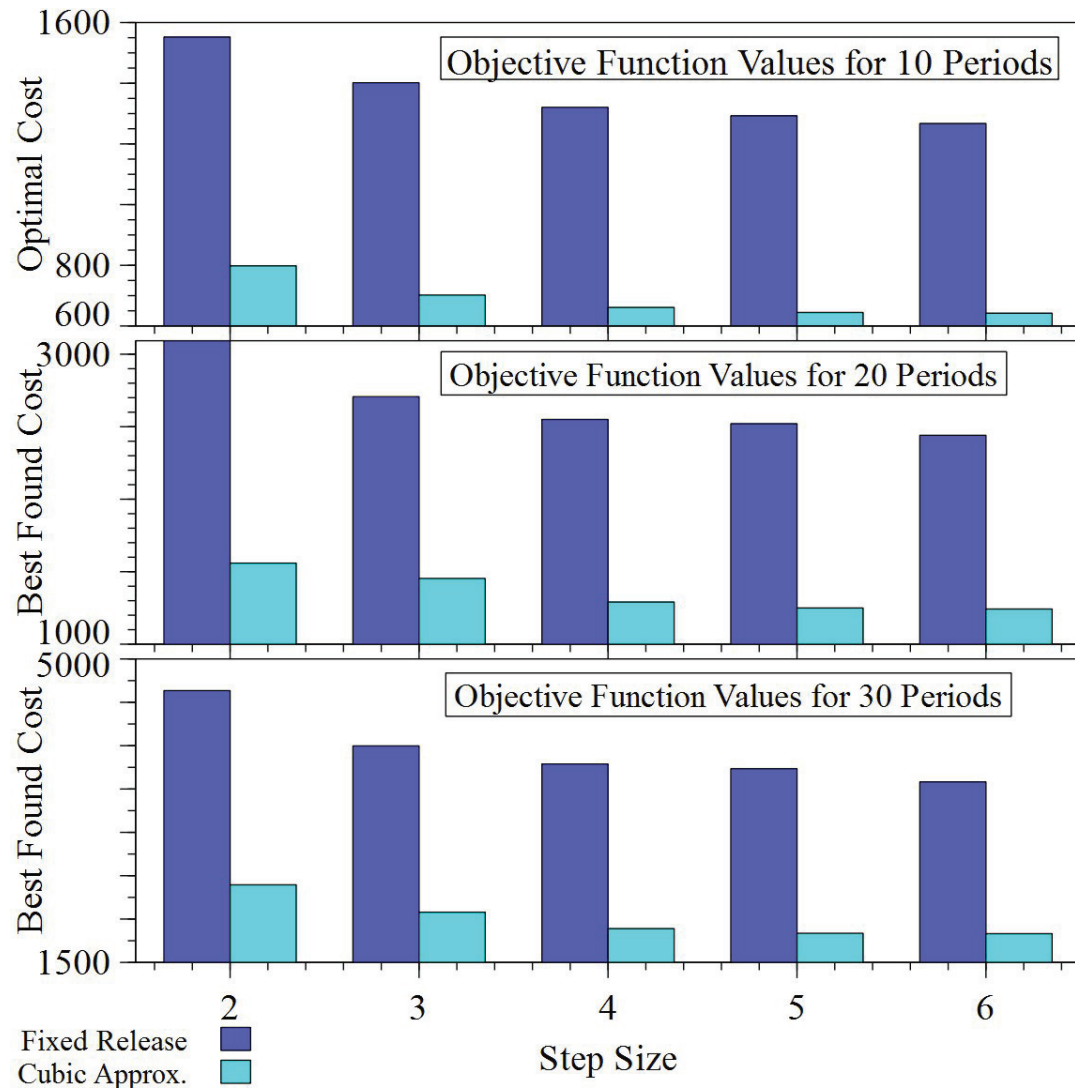


Figure 5.8: Total operational cost comparison

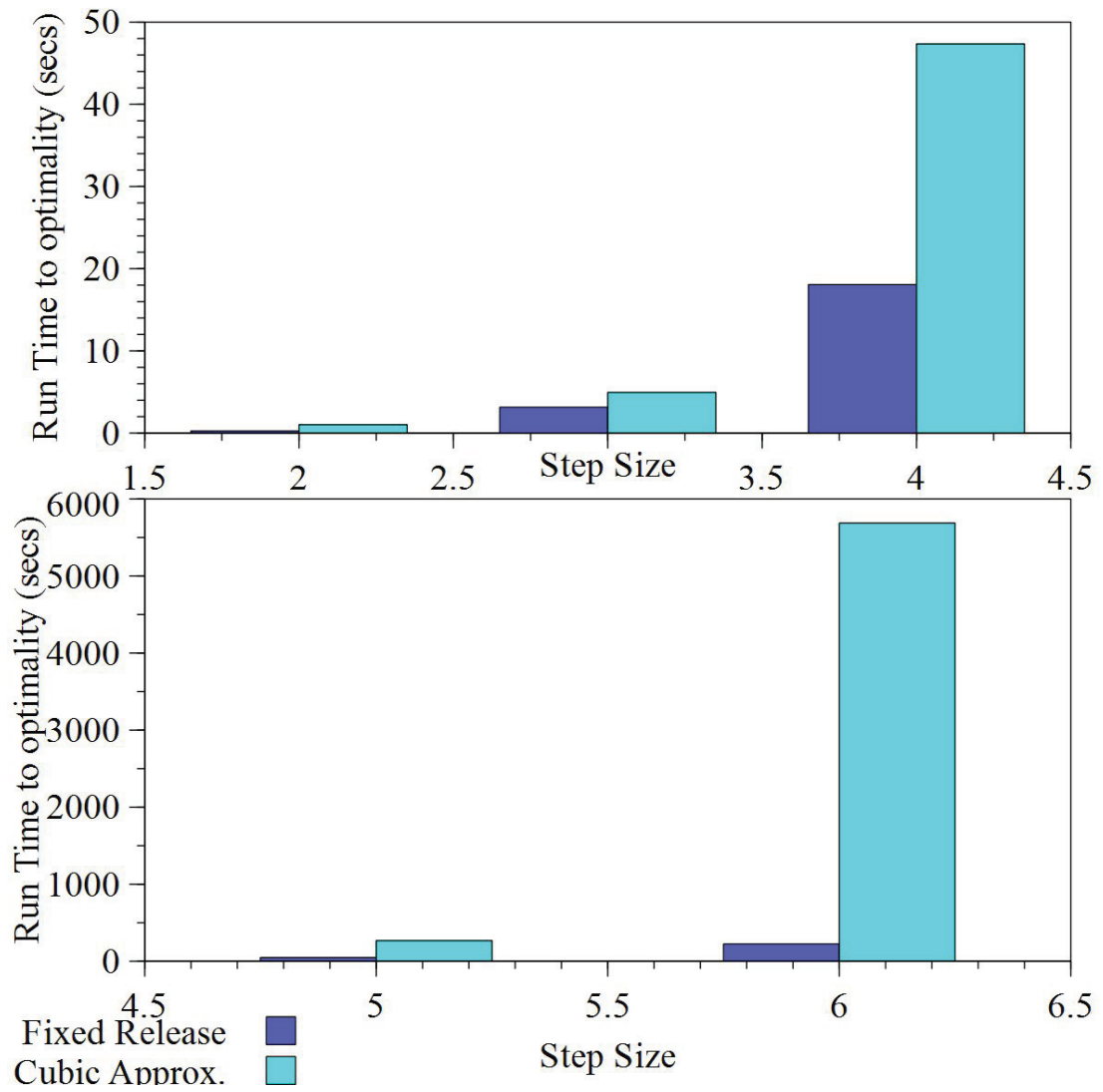


Figure 5.9: Computational time comparison for 10 periods

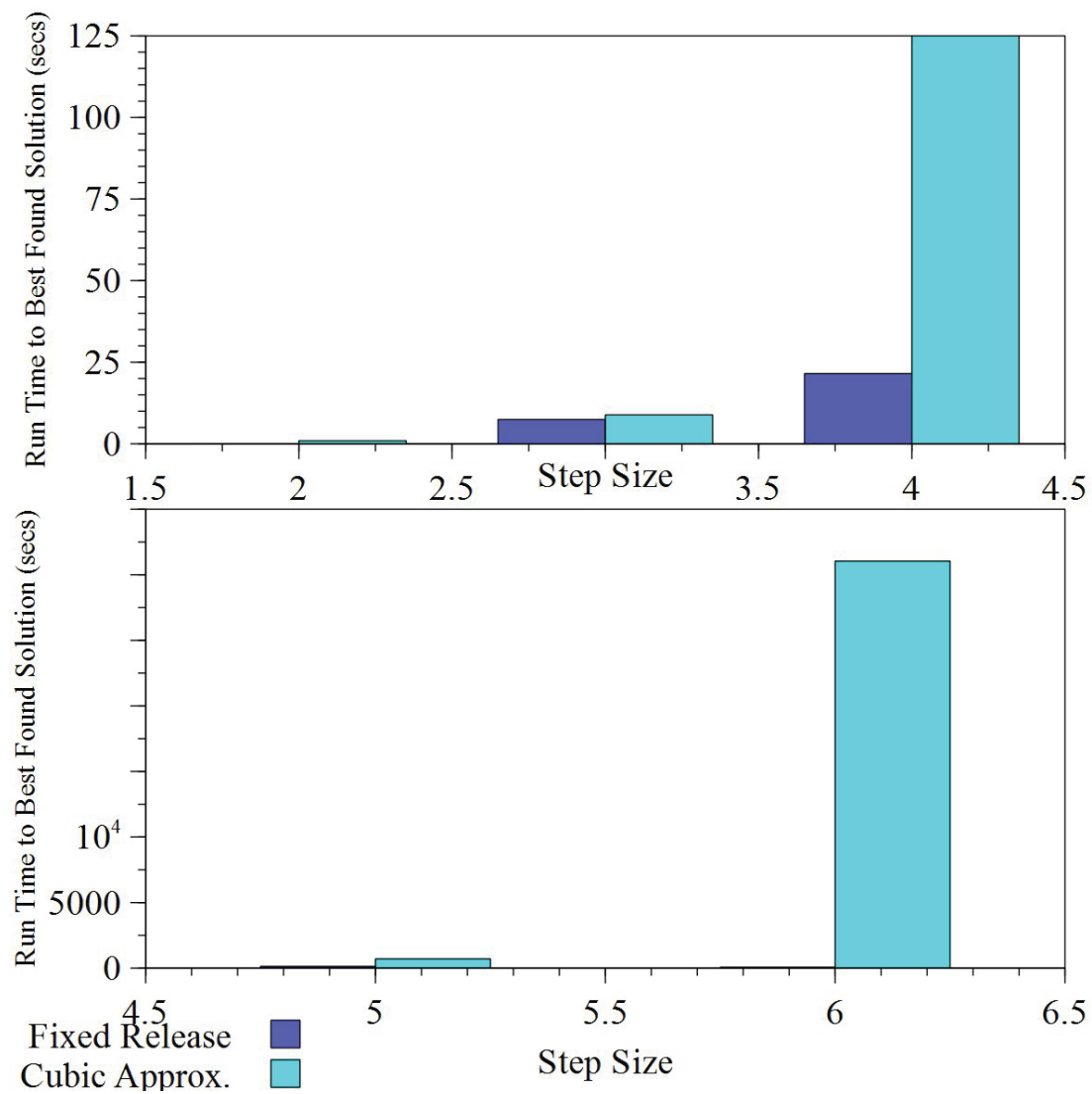


Figure 5.10: Computational time comparison for 20 periods

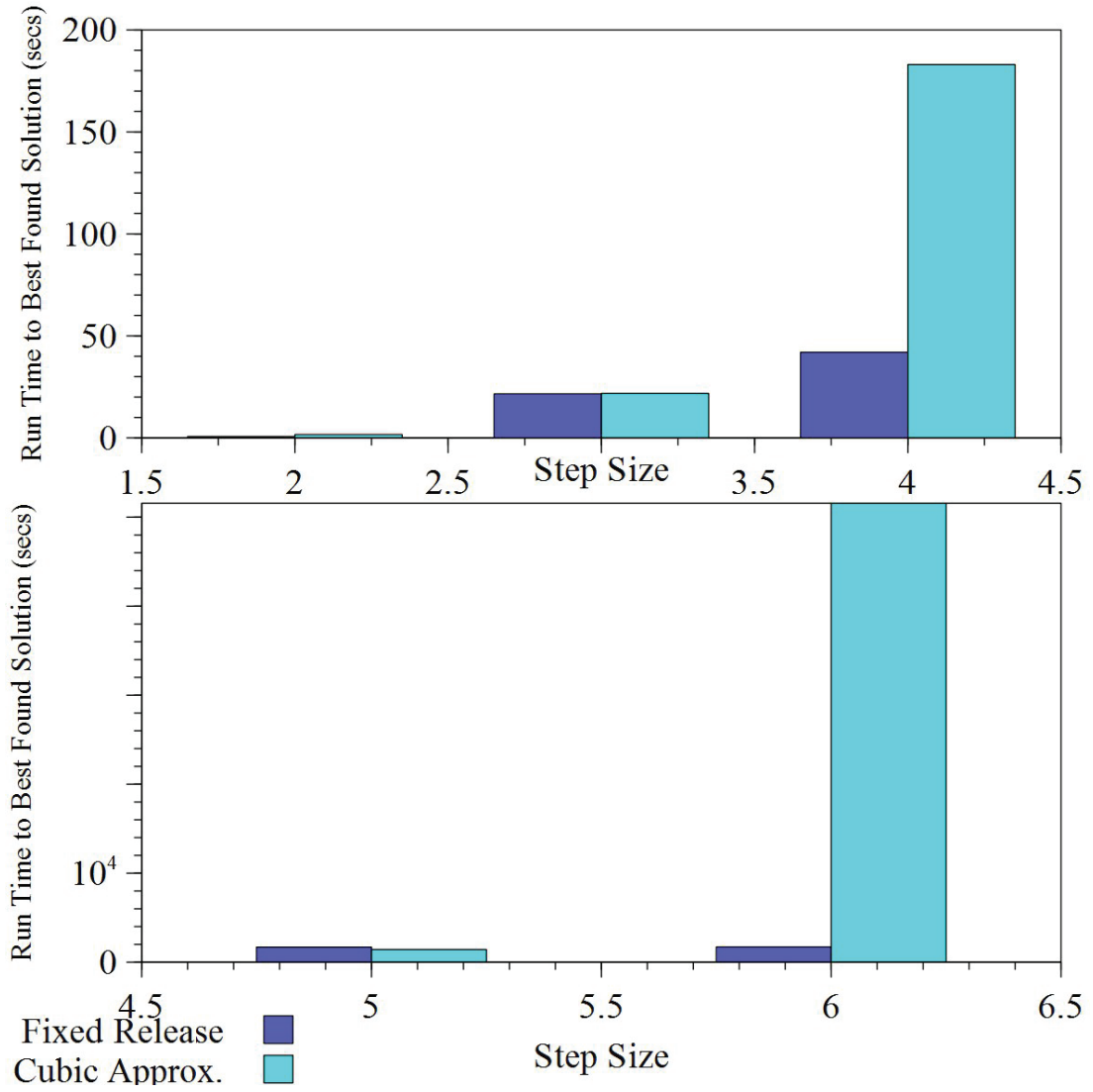


Figure 5.11: Computational time comparison for 30 periods

represent throughput as a function of WIP. The solution space is expanded in CA to include the entire region inside the smaller cuboids whose dimensions depend on step size. There is a trade-off between the quality of solution obtained and run time as the step size is increased. CA appears to yield better solutions but takes longer.

Chapter 6

Comparative Analysis of the ACF, the FPR and the CA Approaches

In this chapter, we look at how the FPR and CA approaches developed in the thesis compare with the ACF approach of [3]. The main difference between ACF and FPR/CA is that in the ACF approach, the clearing function is estimated at each resource (workstation) statistically, based on a simulation of the production facility. In FPR and CA, the aggregated production facility is considered as a whole and estimates of throughput are made for discrete values of product WIP. In what follows, the comparison to the ACF approach paper will be made based on the numerical results obtained by Kacar, Irtem, and Uzsoy [29].

The comparison is conducted on the case mentioned in Chapter 3. This case is also presented in Kacar, Irtem, and Uzsoy [29] to illustrate the performance of the ACF approach. In Chapters 3, 4 and 5 there were no batching or reliability considerations, both of which are used in Kacar, Irtem, and Uzsoy [29]. To be consistent, the case is adapted as follows:

- Since the facility in [29] is assumed to operate 24 hours per day, 7 days per week, this is used in all comparisons.
- The batching workstations in [29] are given as 1 and 2, with the batch size being two or four products of any of the three products type at the same time as one batch. The FPR and CA model throughput estimations use MVA, which does not allow for batching. Thus, the processing time on workstations 1 and 2 are reduced by a factor of four to approximate the batching effect. The ACF throughput estimates are obtained for an open-queuing network simulation implemented in Simio 8.13TM. While SimioTM allows batching, this was not implemented in the ACF. Instead, to keep the comparison consistent, the processing time on these workstations is also reduced by a factor of four.

- In Kacar, Irdem, and Uzsoy [29], workstation 4, the bottleneck workstation, has two servers. Instead of using this, we assume that this workstation has only one station with a processing time of 20 minutes, instead of 40 minutes in Kacar, Irdem, and Uzsoy [29]. This change is also implemented consistently in the SimioTM simulation model for ACF and MVA.
- While processing times in Kacar, Irdem, and Uzsoy [29] are not exponentially distributed, we assume exponential processing times. This is because the MVA algorithm is based on this assumption.
- In Kacar, Irdem, and Uzsoy [29], there are two unreliable workstations (3 and 7) which could starve the bottleneck workstation. The mean time between failure (MTBF) and mean time to repair (MTTR) follow the gamma distribution with specific parameters shown in Table 6.1. Using the approach in Kacar [28], the machine availability (a) is calculated from the following equation:

$$a = \frac{MTBF}{MTBF + MTTR} = \frac{7200}{7200 + 1800} = 0.8 \quad (6.1)$$

Therefore, in the MVA, workstations 3 and 7 are assumed to have 0.8 machines. However, in the simulation model of the open queuing network to gather throughput data for the ACF approach, the MTBF and MTTR values with the gamma distribution parameters are directly entered into SimioTM.

Table 6.1: Machine reliability parameters

Machine	Time Between Failure			Time to Repair		
	α	β	Mean	α	β	Mean
3	7200	1	7200	1200	1.5	1800
7	7200	1	7200	1200	1.5	1800

As a result of these changes, the average processing times in Table 6.2 are used for all comparisons.

Table 6.2: Processing time

Workstation	Mean Processing Time
1	20
2	55
3	45
4	20
5	25
6	22
7	20
8	100
9	50
10	50
11	70

6.1 Numerical Implementation of ACF

The first step in ACF is to find the relationship between resource load (WIP+release) with aggregate throughput at the individual workstation level. As mentioned earlier, this is implemented as a queuing network in SimioTM.

1. The base arrival rate in Kacar, Irdem, and Uzsoy [29] is 3:1:1 for the three products. In order to obtain the clearing functions, the arrival rates for the three products are adjusted (to preserve the 3:1:1 ratio) to get an average utilization (ρ) for the bottleneck machine (machine 4) equal to 49%, 60%, 70%, 77%, 87%, 94%, and 99%. These match the average utilizations in Kacar, Irdem, and Uzsoy [29]. This is done by first computing the effective arrival rate using the following equation:

$$\lambda_{effective} = \frac{\rho_j \mu_j}{\sum_i (PM_i N_{ij})} \quad (6.2)$$

Where, ρ_j (in this case, 49%, 60%, etc.) is the utilization of machine j (which is the bottleneck machine), PM_i is the product mix (in this case, 3:1:1) for each product i , N_{ij} is the number of visits of product i to machine j , $\lambda_{effective}$ is the arrival rate to the machine j , and μ_j is the process time of machine j . To obtain the arrival rate for each product i the following equation is used:

$$\lambda_i = \lambda_{effective} PM_i \quad (6.3)$$

where, λ_i constitutes the arrival rate for product i .

2. An open network simulation model of the network is developed in SimioTM to generate arrivals for the WIP-Throughput relationship calibration experiment. It should be noted that interarrival time is assumed to be exponential (i.e., the number of arrivals in a given period is Poisson). Arrivals are generated for each of the 7 bottleneck machine utilization scenarios. Each run is 100 weeks long, with a warm-up of 10 weeks (i.e., no data is collected during the warm-up period). There are 5 replications for each scenario, resulting in 35 data points for each workstation.
3. The regression function in Minitab 17.3.1TM is used to develop piecewise linear estimates of the clearing function at each workstation. In order to do this, the points are grouped into 7 clouds, each cloud representing the replications for a utilization scenario. In the first regression, the point (0, 0), which is equivalent to zero throughput for zero WIP and the first cloud of points are used to get a regression line. Successive clouds are added and the regression line is revised so long as the R^2 value of the regression is greater than or equal to 95%. If the R^2 value is less than 95%, the cloud is not added. In this case, the previous line is kept and the next two clouds are used to continue with the process. This way, the change in the CF curvature slope is captured.

The results of the regression are shown in Figures 6.1 to 6.11. In each figure, the graph on top shows the data points for the resource output (throughput) and load (WIP+release) from the simulation, while the graphs on the bottom show the regression results. Also, the line in green on the top is the constant level capacity of Figure 2.3. This line has a slope of zero, which means increasing WIP further will have no effect on throughput.

4. The ACF model of Kacar, Irdem, and Uzsoy [29] can now be run for the different demand scenarios to compare its results versus both the FPR and CA approach.

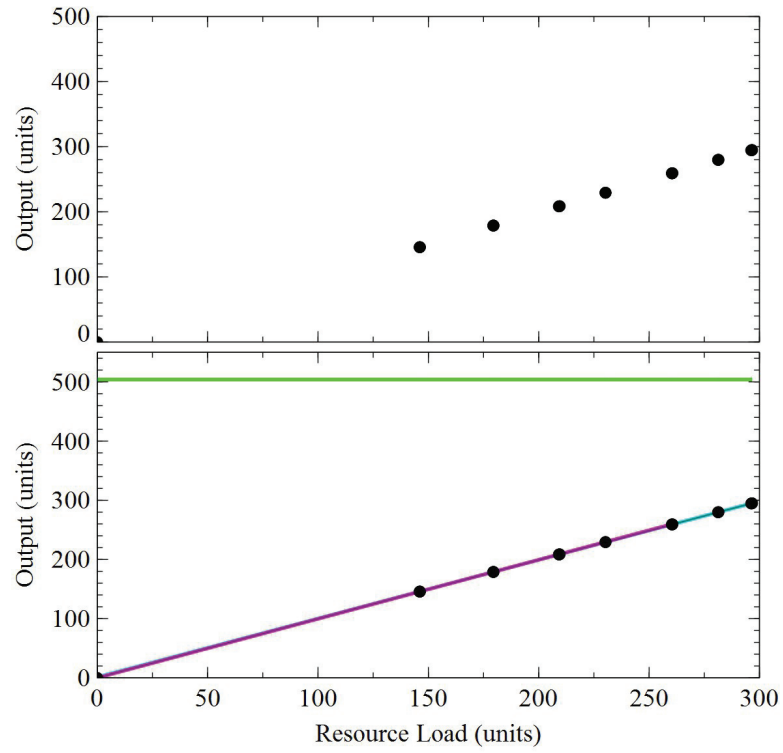


Figure 6.1: ACF for machine 1

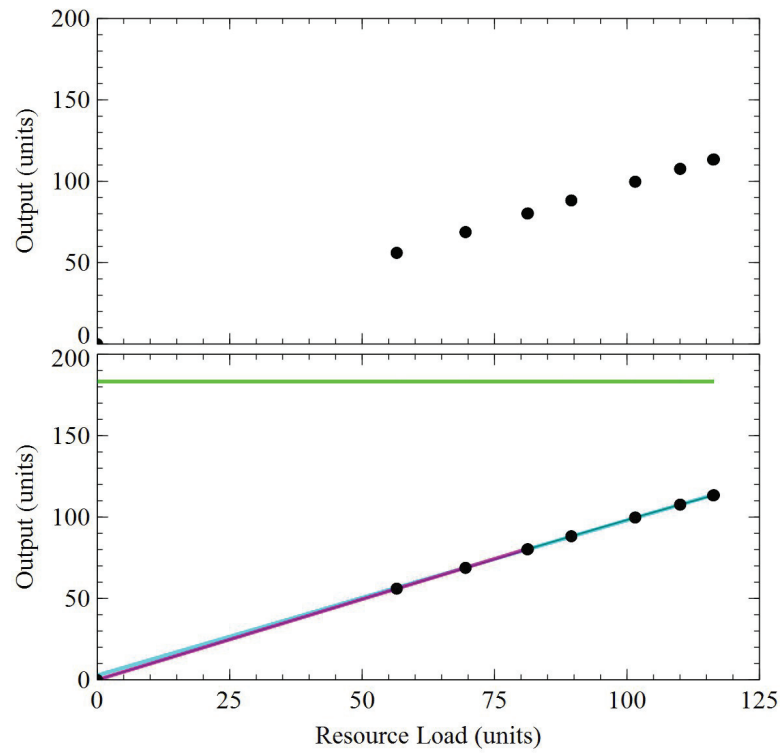


Figure 6.2: ACF for machine 2

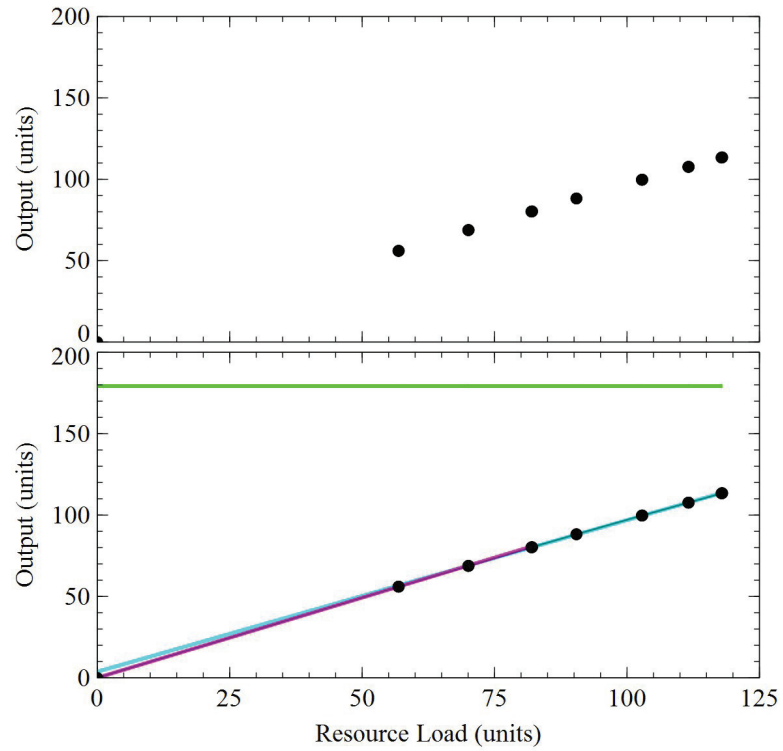


Figure 6.3: ACF for machine 3

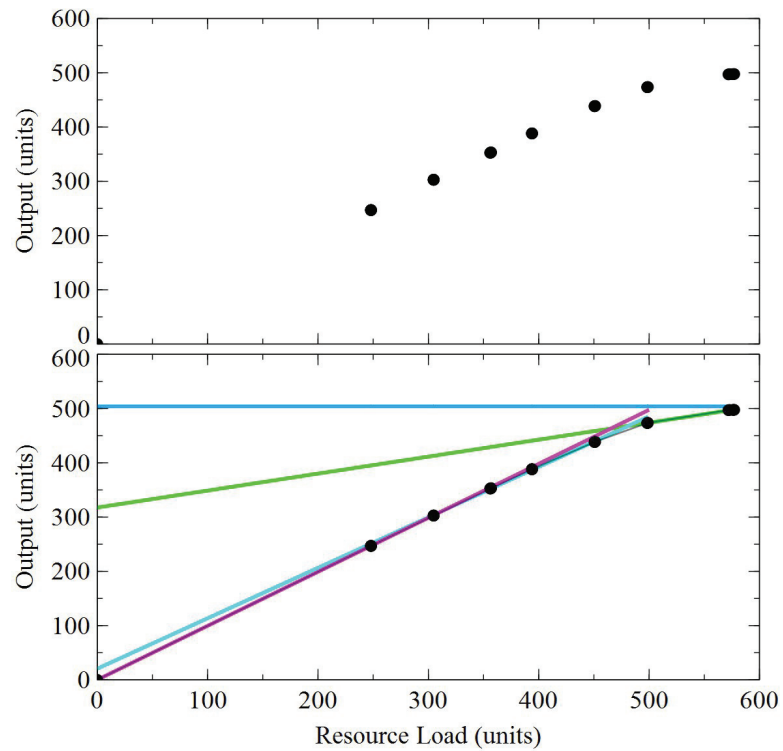


Figure 6.4: ACF for machine 4

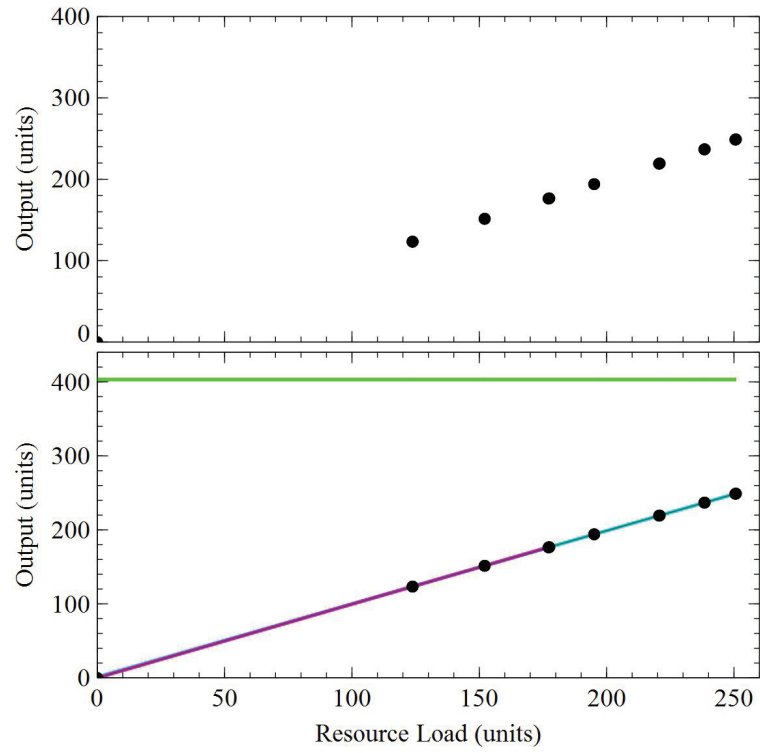


Figure 6.5: ACF for machine 5

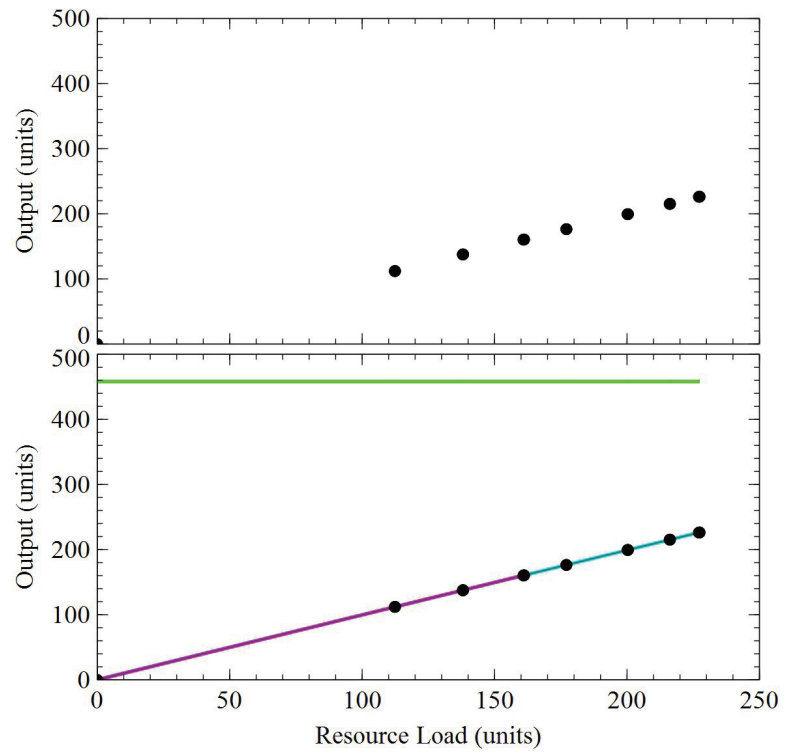


Figure 6.6: ACF for machine 6

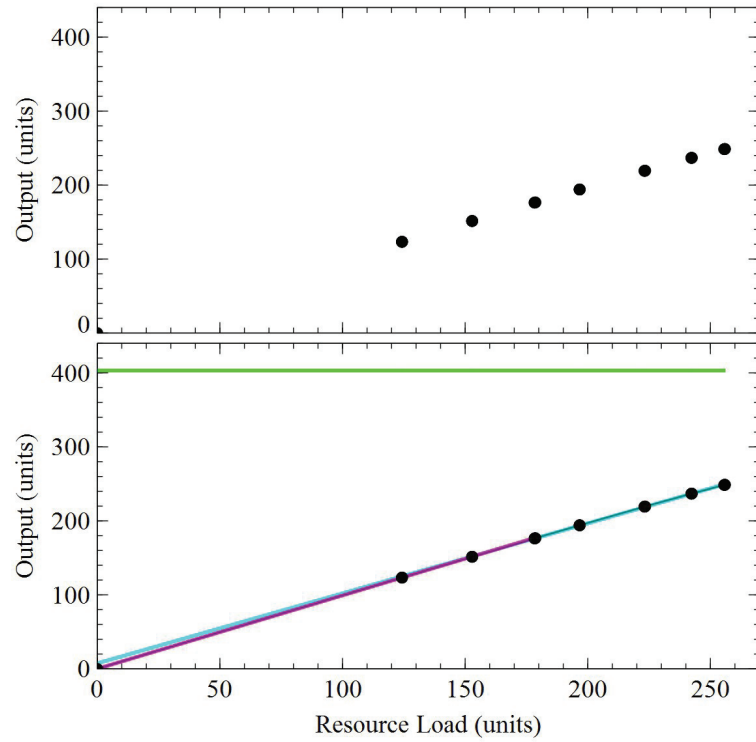


Figure 6.7: ACF for machine 7

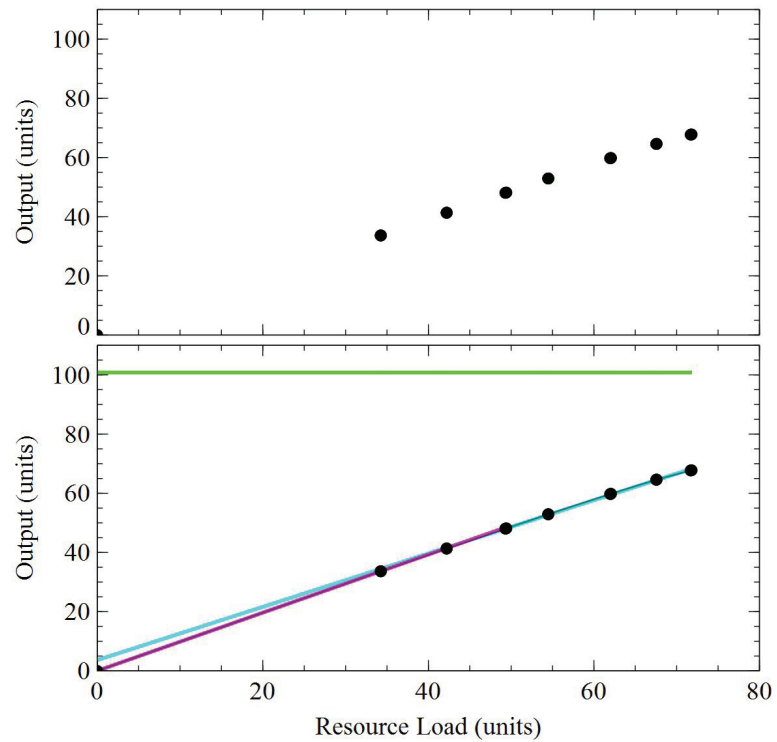


Figure 6.8: ACF for machine 8

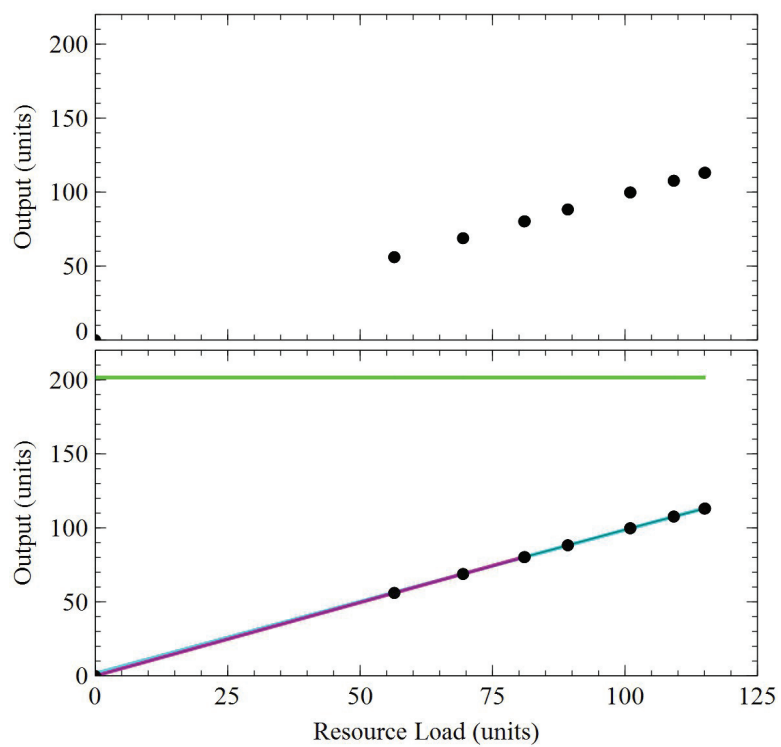


Figure 6.9: ACF for machine 9

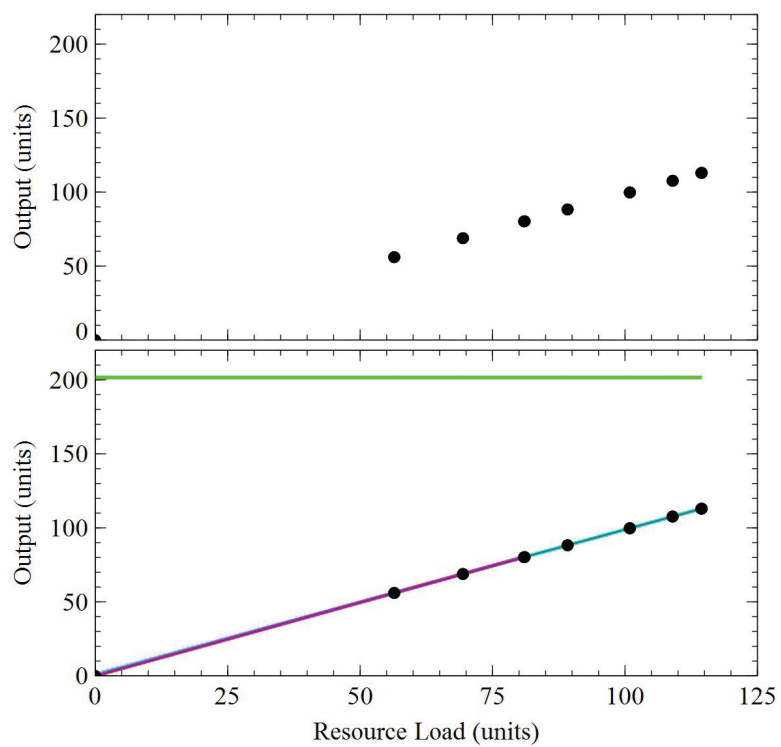


Figure 6.10: ACF for machine 10

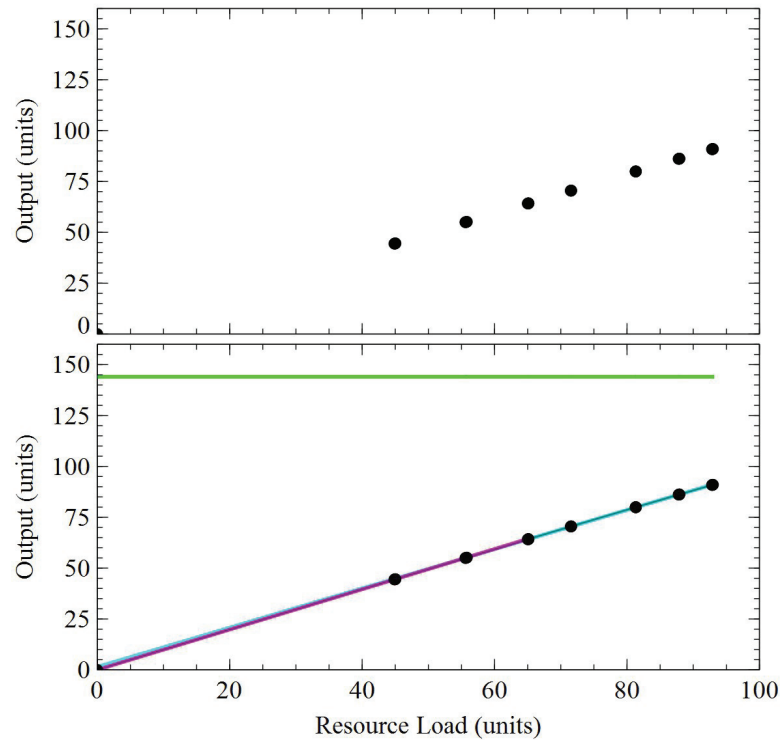


Figure 6.11: ACF for machine 11

6.1.1 Kacar and Uzsoy [37] ACF Model Formulation

After deriving the ACF in the previous subsection, the LP model can be run to compare it with the proposed approaches. The LP formulation is shown in both Kacar and Uzsoy [37] and Kacar, Irdem, and Uzsoy [29]:

Indices :

- t Time index,
- g Product index,
- l Operation index,
- k Workcenter index.

Parameters :

- w_g The cost of holding WIP of product g ,
- h_g The cost of holding finished goods inventory of product g ,
- b_g The cost of backorder for product g ,
- d_{gt} The demand of product g in period t ,
- $C(k)$ The set of line segment used to linearize the CF for workcenter k ,

- μ_k^n The intercept of the n^{th} linear segment of workcenter k for the CF,
 β_k^n The slope the n^{th} line segment of workcenter k for the CF.

Variables :

- Y_{gtl} The total production quantity of product g completes
 l^{th} operation in period t ,
 Y_{gt} The total production quantity of product g in period t ,
 X_{gtl} The released quantity of product g to operation l in period t ,
 W_{gtl} The WIP level of product g of operation l at the end of period t ,
 $W_{gt(k)}$ The WIP level of product g at workcenter k in period t ,
 here $W_{gt(k)} = \sum_{l \in L(k)} W_{gtl}$,
 I_{gt} The inventory level of the final product g in period t ,
 B_{gt} The backorder quantity of product g in period t ,
 Z_{gtl}^k The fraction throughput of k^{th} workcenter allocated to operation l
 of product g in period t .

Objective function :

$$\min z \sum_{g \in G} \left[\sum_{t=1}^T \sum_{k=1}^K w_g W_{gt(k)} + \sum_{t=1}^T h_g I_{gt} + \sum_{t=1}^T b_g B_{gt} \right] \quad (6.4)$$

Subject to:

$$Y_{gtl} + I_{g,t-1} - B_{g,t-1} - I_{g,t} + B_{g,t} = d_{gt} \quad g \in G, t = 1, \dots, T, l \in L \quad (6.5)$$

$$W_{gtl} = W_{g,t-1,l} + X_{gtl} - Y_{gtl} \quad g \in G, t = 1, \dots, T, l \in L \quad (6.6)$$

$$Y_{gtl} \leq Z_{gtl}^k \mu_k^n + \beta_k^n (X_{gtl} + W_{g,t-1,l}) \quad g \in G, t = 1, \dots, T, l \in L, k \in K, n \in C(K) \quad (6.7)$$

$$\sum_{g \in G, l \in L(k)} Z_{gtl}^k = 1 \quad t = 1, \dots, T, k \in K \quad (6.8)$$

$$Y_{gtl}, W_{gtl}, I_{gt}, X_{gtl}, B_{gt}, Z_{gtl}^k \geq 0 \quad g \in G, t = 1, \dots, T, l \in L, k \in K \quad (6.9)$$

The objective function of the model is to minimize the total cost of holding WIP and finished goods inventory, and backorder. This is subject to restrictions on the flow conservation of finished goods inventory (6.5), and WIP (6.6). The CF is used in (6.7) to model the machine capacity at each machine k . Since there are multiple products g , the clearing function (capacity) is allocated using the variable Z_{gtl}^k of machine k for the different operation l in (6.8). Note that $\sum_g \sum_l Z_{gtl}^k = l$ in Kacar, Irtem, and Uzsoy in [29]. However, this seems to be a typographical error, with the sum equal to 1 in Kacar and Uzsoy [37]. The non-negativity constraint is shown in (6.9).

The initial inventory and WIP values have been modified in this comparison to be decision variables.

6.2 Fixed Points Release Model Parameters

In order to replicate the numerical experiments in Kacar, Irtem, and Uzsoy [29], where the product demands vary between 30 and 75 for product 1 and between 10 and 25 for products 2 and 3, a grid with step size 5 is created to cover this range, as can be seen in Table 6.3. The product release combinations in Table 6.3 are simulated in SimioTM to estimate system WIP for the open queuing network case. As can be seen in the table, the system throughput values (column 3) are close to the product release values (column 1). There are 160 release combinations in total (10 discrete values for product 1 and 4 discrete values for products 2 and 3). As mentioned in Chapter 4, the finer the grid, the more accurate is the solution (though it takes longer to compute). Table 6.3 shows 38 combinations only; all 160 combinations are shown in Appendix D.1.

The WIP levels from Table 6.3 are then used to get the estimates of the equivalent closed queuing network using MVA. The results are shown in Table 6.4. Once again, Table 6.4 shows only 38 combinations; all 160 combinations are shown in Appendix D.2.

It can be seen that the MVA throughput estimates are higher than the open network simulation throughput estimates, because releases are controlled in a closed queuing network. The WIP-Throughput estimates can now be used in the data-driven FPR model. This can be done for either the open network case in Table 6.3 or the closed network case in Table 6.4.

Table 6.3: Release combinations, simulation WIP and throughput

Set	Product Mix			Simulation WIP			Simulation Th\Week		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	30	10	10	3.66	0.77	1.01	29.99	10.00	9.96
2	30	10	15	3.79	0.81	1.74	29.96	10.04	15.04
3	30	10	20	3.93	0.84	2.76	30.05	9.97	20.01
4	30	10	25	4.08	0.88	4.42	30.00	10.00	24.98
5	30	15	10	3.86	1.23	1.04	29.98	15.04	9.99
6	30	15	15	3.99	1.29	1.78	29.94	15.01	14.93
7	30	15	20	4.15	1.36	2.86	29.98	15.04	20.04
8	30	15	25	4.31	1.41	4.51	30.08	15.00	25.01
9	30	20	10	4.08	1.76	1.08	29.93	20.02	9.99
10	30	20	15	4.23	1.84	1.84	29.98	19.97	14.96
11	30	20	20	4.42	1.93	2.92	30.07	19.99	20.02
12	30	20	25	4.61	2.03	4.73	30.07	20.01	25.00
13	30	25	10	4.34	2.36	1.12	29.99	25.07	9.97
14	30	25	15	4.53	2.47	1.93	30.03	24.98	15.06
15	30	25	20	4.69	2.60	3.03	29.96	25.02	19.98
16	30	25	25	4.90	2.74	4.79	30.01	25.07	24.95
17	35	10	10	4.73	0.84	1.05	35.05	9.94	9.95
18	35	10	15	4.91	0.88	1.81	35.04	9.94	14.99
19	35	10	20	5.06	0.92	2.86	35.03	10.03	20.03
20	35	10	25	5.26	0.96	4.56	35.08	10.00	25.05
21	35	15	10	5.00	1.36	1.10	34.97	15.01	10.05
22	35	15	15	5.20	1.42	1.86	35.10	15.05	14.97
23	35	15	20	5.34	1.48	2.98	34.89	15.04	20.10
24	35	15	25	5.58	1.55	4.67	35.02	15.01	24.93
25	35	20	10	5.32	1.94	1.13	35.04	19.96	10.01
26	35	20	15	5.51	2.04	1.93	35.00	20.01	14.99
27	35	20	20	5.68	2.12	3.04	34.94	20.02	19.93
28	35	20	25	5.96	2.23	4.86	35.04	19.98	25.00
29	35	25	10	5.66	2.62	1.18	34.90	25.01	10.04
30	35	25	15	5.91	2.75	2.01	35.01	24.97	15.02
31	35	25	20	6.11	2.88	3.14	34.94	25.02	19.95
32	35	25	25	6.36	3.02	4.96	34.96	25.05	24.97
33	40	10	10	5.99	0.93	1.11	40.01	10.03	10.04
34	40	10	15	6.22	0.97	1.90	39.98	10.07	15.04
35	40	10	20	6.40	1.00	2.96	39.97	9.99	19.97
36	40	10	25	6.63	1.05	4.70	39.98	9.97	24.89
37	40	15	10	6.39	1.50	1.15	39.95	15.00	10.04
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
160	75	25	25	648.97	146.53	6.87	63.91	22.41	24.95

Table 6.4: Release combinations, simulation WIP, and MVA throughput

Set	Product Mix			Simulation WIP			MVA Throuput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	30	10	10	3.66	0.77	1.01	37.49	12.91	13.53
2	30	10	15	3.79	0.81	1.74	37.59	13.04	20.27
3	30	10	20	3.93	0.84	2.76	37.83	12.92	26.44
4	30	10	25	4.08	0.88	4.42	38.07	13.03	31.44
5	30	15	10	3.86	1.23	1.04	37.61	19.33	13.53
6	30	15	15	3.99	1.29	1.78	37.64	19.46	20.14
7	30	15	20	4.15	1.36	2.86	37.78	19.59	26.45
8	30	15	25	4.31	1.41	4.51	38.06	19.64	31.34
9	30	20	10	4.08	1.76	1.08	37.55	25.89	13.51
10	30	20	15	4.23	1.84	1.84	37.67	25.88	20.13
11	30	20	20	4.42	1.93	2.92	37.92	25.97	26.28
12	30	20	25	4.61	2.03	4.73	38.16	26.14	31.44
13	30	25	10	4.34	2.36	1.12	37.53	32.31	13.49
14	30	25	15	4.53	2.47	1.93	37.76	32.26	20.23
15	30	25	20	4.69	2.60	3.03	37.71	32.49	26.28
16	30	25	25	4.90	2.74	4.79	37.98	32.65	31.25
17	35	10	10	4.73	0.84	1.05	44.14	12.87	13.52
18	35	10	15	4.91	0.88	1.81	44.29	12.93	20.26
19	35	10	20	5.06	0.92	2.86	44.26	13.03	26.39
20	35	10	25	5.26	0.96	4.56	44.60	13.06	31.38
21	35	15	10	5.00	1.36	1.10	44.03	19.40	13.65
22	35	15	15	5.20	1.42	1.86	44.23	19.47	20.09
23	35	15	20	5.34	1.48	2.98	44.07	19.49	26.45
24	35	15	25	5.58	1.55	4.67	44.48	19.53	31.29
25	35	20	10	5.32	1.94	1.13	43.96	25.73	13.56
26	35	20	15	5.51	2.04	1.93	43.98	25.86	20.15
27	35	20	20	5.68	2.12	3.04	43.95	25.84	26.28
28	35	20	25	5.96	2.23	4.86	44.34	25.91	31.32
29	35	25	10	5.66	2.62	1.18	43.61	32.03	13.65
30	35	25	15	5.91	2.75	2.01	43.82	32.09	20.17
31	35	25	20	6.11	2.88	3.14	43.79	32.18	26.18
32	35	25	25	6.36	3.02	4.96	43.98	32.23	31.19
33	40	10	10	5.99	0.93	1.11	50.37	12.93	13.59
34	40	10	15	6.22	0.97	1.90	50.52	12.99	20.30
35	40	10	20	6.40	1.00	2.96	50.51	12.88	26.31
36	40	10	25	6.63	1.05	4.70	50.72	12.95	31.29
37	40	15	10	6.39	1.50	1.15	50.27	19.29	13.61
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
160	75	25	25	648.97	146.53	6.87	68.50	23.25	32.39

6.3 Cubic Approximation Approach Parameters

The grid for the CA approach is also chosen based on the demand range in Kacar, Irdem, and Uzsoy [29], which is (30, 75), (10, 25) and (10, 25) for the three products. From the open-queuing simulation network results, the corresponding WIP levels for the three products are in the range (5.02, 47.09), (1.03, 11.43) and (1.22, 3.43). These WIP levels are covered using a step size of 8, since using few steps reduces numerical accuracy and using more steps made the optimization run time impractical. Note that throughput levels for the closed network case are once again calculated based on the WIP range and step size using MVA. The throughput values are not shown because there are $8^3 = 512$ data points, which is significantly higher than the 160 points used for FPR. Since the CA approach actually interpolates between the product release combination points, it is very important to have a fine grid to make the throughput estimates more realistic.

6.4 Numerical Experiments

The ACF, FPR, and CA approaches are now compared. The cost factors used in all comparisons are 50 for backorder, 35 for WIP, and 15 for inventory holding cost. There are two types of data scenarios in Kacar, Irdem, and Uzsoy [29]: correlated demands and mixed demands.

All model variants in this chapter are solved using Gurobi Optimizer 7.0.1TM from within the MPL 5.0TM for Windows environment. The run times for the optimization are based on a 64-bit Windows 10.0TM computer with 24GB RAM and 3.3Ghz clock speed.

6.4.1 Correlated Demand

We first begin with a simplified correlated demand scenario where the demand is (60, 20, 20) for product 1, 2 and 3, in all 26 periods to understand how the models work. This demand scenario corresponds to a 87% utilization level of the bottleneck workstation 4.

Four production planning models are considered:

1. ACF

2. FPR using open queuing network (FPR-O), where product release rates are (60, 20, 20) and the WIP estimates come from the open network simulation model. The throughput estimates from the simulation, as seen in Table 6.3 are (60.0647, 19.9956, 19.9047). However, demand values of (60, 20, 20) are used for all 26 periods. The grid values for FPR are as shown in Table 6.3.
3. FPR using closed queuing network (FPR-C), where the WIP estimates from the open network simulation are used to estimate throughput in an equivalent closed-queuing network using MVA. Once again, the grid values of Table 6.4 are used.
4. CA-8, where the cubic approximation method for the closed queuing network is used with a step size of 8, as described in Section 6.3.

Figure 6.12 shows the production in each period for ACF, FPR-O, FPR-C, and CA. In the ACF production graph, there is a build up for product 1 after which production matches demand. This is not observed for products 2 and 3, which always match demand. In FPR-O and FPR-C, some oscillations in production are observed, while in CA, production always matches the demand for the three products. Since the WIP choices are discrete in FPR-O and FPR-C, if a particular release combination is not on the grid, the solution oscillates between higher and lower production. For this case, it should be noted that the values in the load (WIP+throughput) grid yield a production that differs fractionally from (60, 20, 20). This fractional error causes the oscillation.

While the ACF model ran in negligible time, the solution time for the FPR and CA models took significantly longer. Therefore, all FPR and CA cases in this chapter were solved using the supplementary variables and constraints in Section 4.4 with $M = 55$ and $f = 0.5$. The objective function value for each of the models is shown in Figure 6.13.

The ACF solution successfully meets demand with very low inventories. The FPR-O solution has a lower cost (since this solution was the optimal with supplementary variables and constraints, it represents an upper bound on the optimal FPR solution). This solution has some backorder costs though the production planning cost (WIP, inventory, and backorder) are lower.

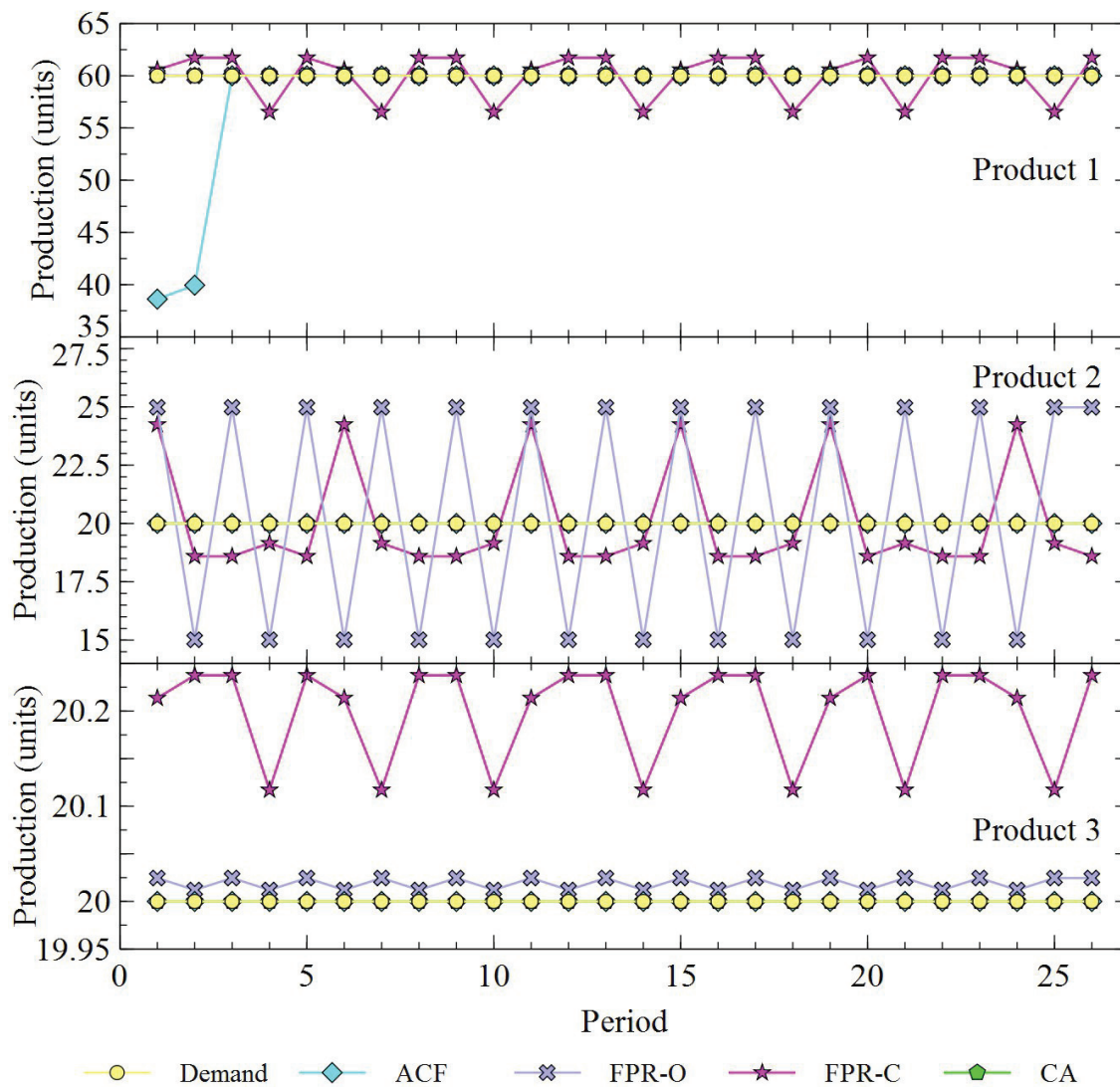


Figure 6.12: Production rate comparison for 60-20-20 case

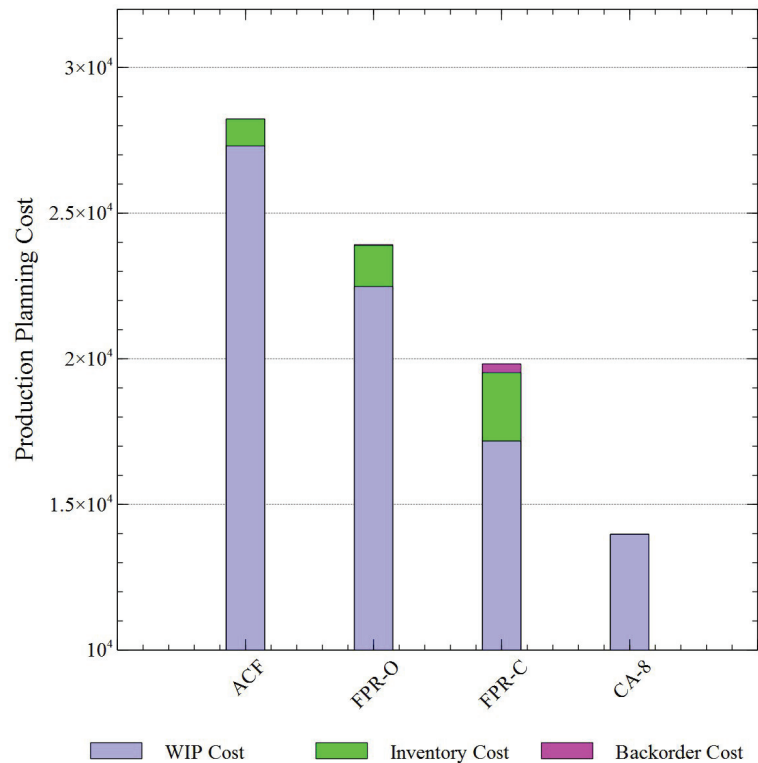


Figure 6.13: Production planning cost results for 60-20-20 demand

The production planning cost of FPR-C is much lower than ACF and FPR-O. The reason for this is that FPR-C is based on the closed queuing network assumption. From Tables D.1 and D.2, it can be seen that the WIP levels (total of 31.24) in row 107 give an open-queuing network throughput of (60, 20, 20), while in row 70, almost the same throughput is realized with a total WIP of approximately 15.03 for the closed-queuing network case. The CA-8 model outperforms FPR-C and is in fact the one with the lowest cost. Since it is an interpolative method, demand can be met without holding any inventory or using backorders. The CA-8 model also has the advantage of the closed queuing network.

In order to further understand the differences, the WIP-Throughput relationship between the three approaches (ACF, FPR, and CA) are compared in Table 6.5 (also illustrated in Figure 6.14). As can be seen, the total WIP in ACF and the open network simulation estimate are close. The MVA estimates of WIP for a throughput of (60, 20, 20) is found by trial and error to total to be approximately 14.53. The ACF WIP values are found by examining the LP solution for periods where the

throughput is exactly (60, 20, 20). The Z_{gtl}^k values in the solution give us the CF allocations for the operation. Based on these allocations, and the active line in the piecewise regression, we can find the WIP values corresponding to the throughput values (60, 20, 20). While the total WIP between ACF and FPR-O (based on an open queuing network simulation) are close, the individual product WIPs are different. In ACF, these values are approximately (25.05, 3.63, 2.86) while in FPR-O, they are approximately (22.77, 4.68, 3.79).

Table 6.5: Product WIP in ACF, FPR, and CA

Product	ACF	Simulation	MVA
1	25.05	22.77	10.32
2	3.63	4.68	2.11
3	2.86	3.79	2.10
Total	31.54	31.24	14.53

This leads us to believe that all three methods (ACF, FPR, and CA) introduce different types of inaccuracies which can be summarized through the following bullet points:

1. The ACF seems to introduce inaccuracies in individual product WIPs. This is the net result of the piecewise regression procedure to estimate the CF lines and also the allocation of capacity through the Z_{gtl}^k variables.
2. The FPR method can use exact open simulation throughput point estimates, making it in some sense more accurate than ACF. However, since the WIP choices are discrete, the MIP optimal solution is only approximate because if a particular release combination is not on the grid, the solution will need to produce too much or too little in a period.
3. The CA can better match demand, but because of the discrete steps involved, the estimates of throughput for intermediate values of WIP in the grid are approximate.

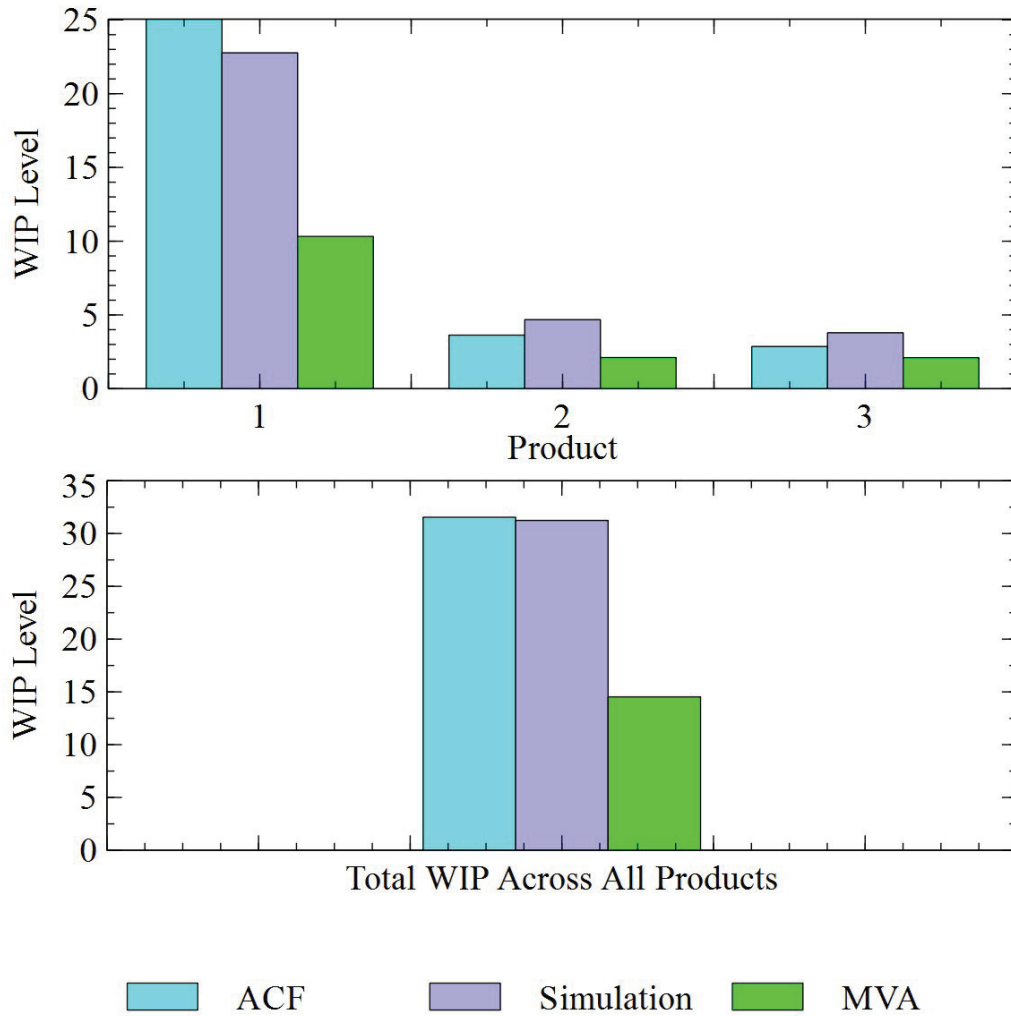


Figure 6.14: Product-WIP level for ACF, FPR, and CA

6.4.1.1 Constant demand based on 70% and 90% utilization

Kacar, Irtem, and Uzsoy [29] discuss two constant demand scenarios based on 70% and 90% utilization level of the bottleneck machine. The demand values for 70% utilization are (49, 16, 16), while for the 90% utilization case, they are (62, 21, 21), for all 26 periods. In Figures 6.15 and 6.16, it can be seen that the ACF model requires a ramp-up for production 1. The production in FPR-O and FPR-C oscillate slightly, while in CA, production matches demand.

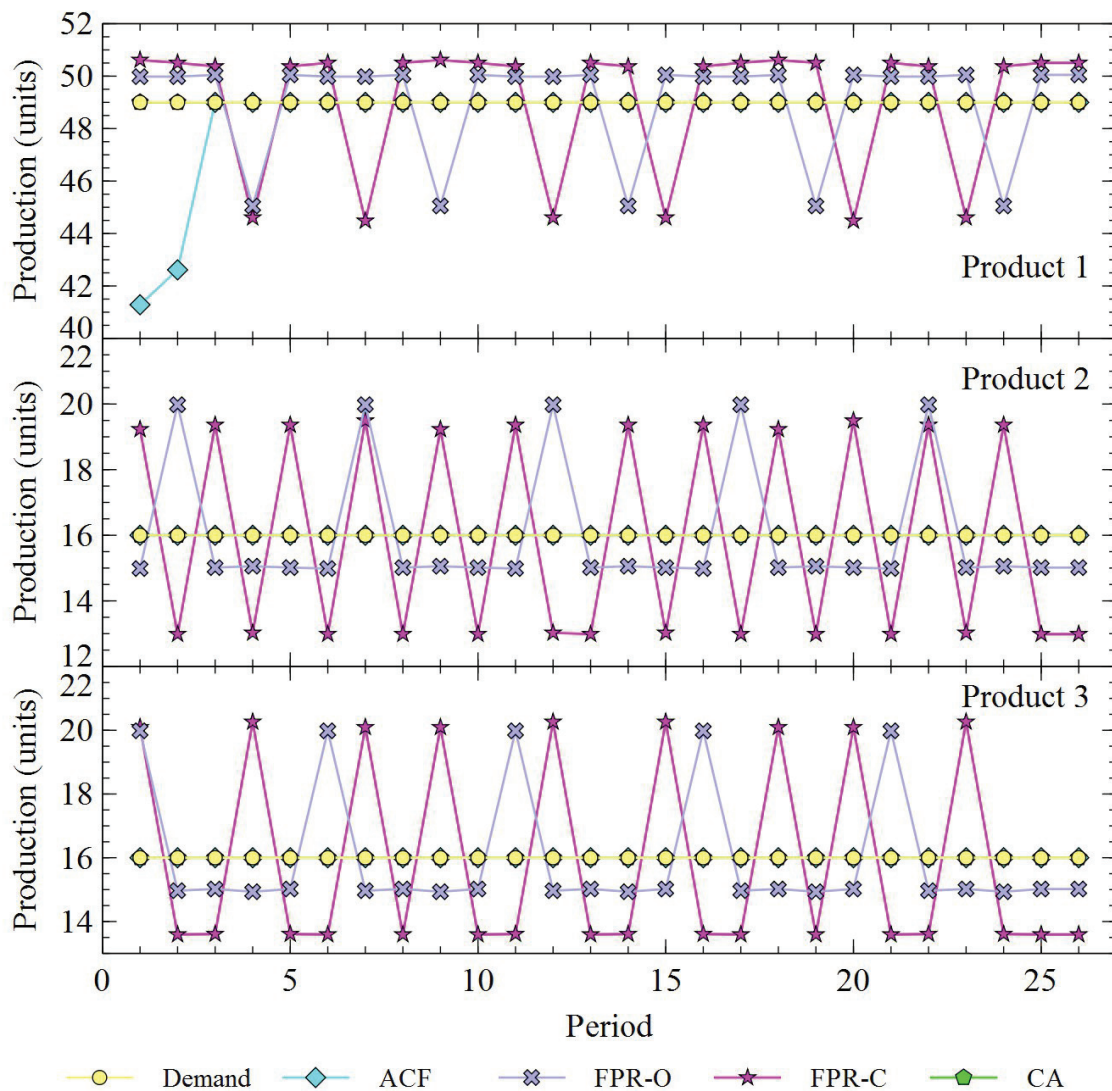


Figure 6.15: Production rate comparison for constant demand based on 70% utilization case

The objective function values for the approaches are shown in Figures 6.17 and 6.18.

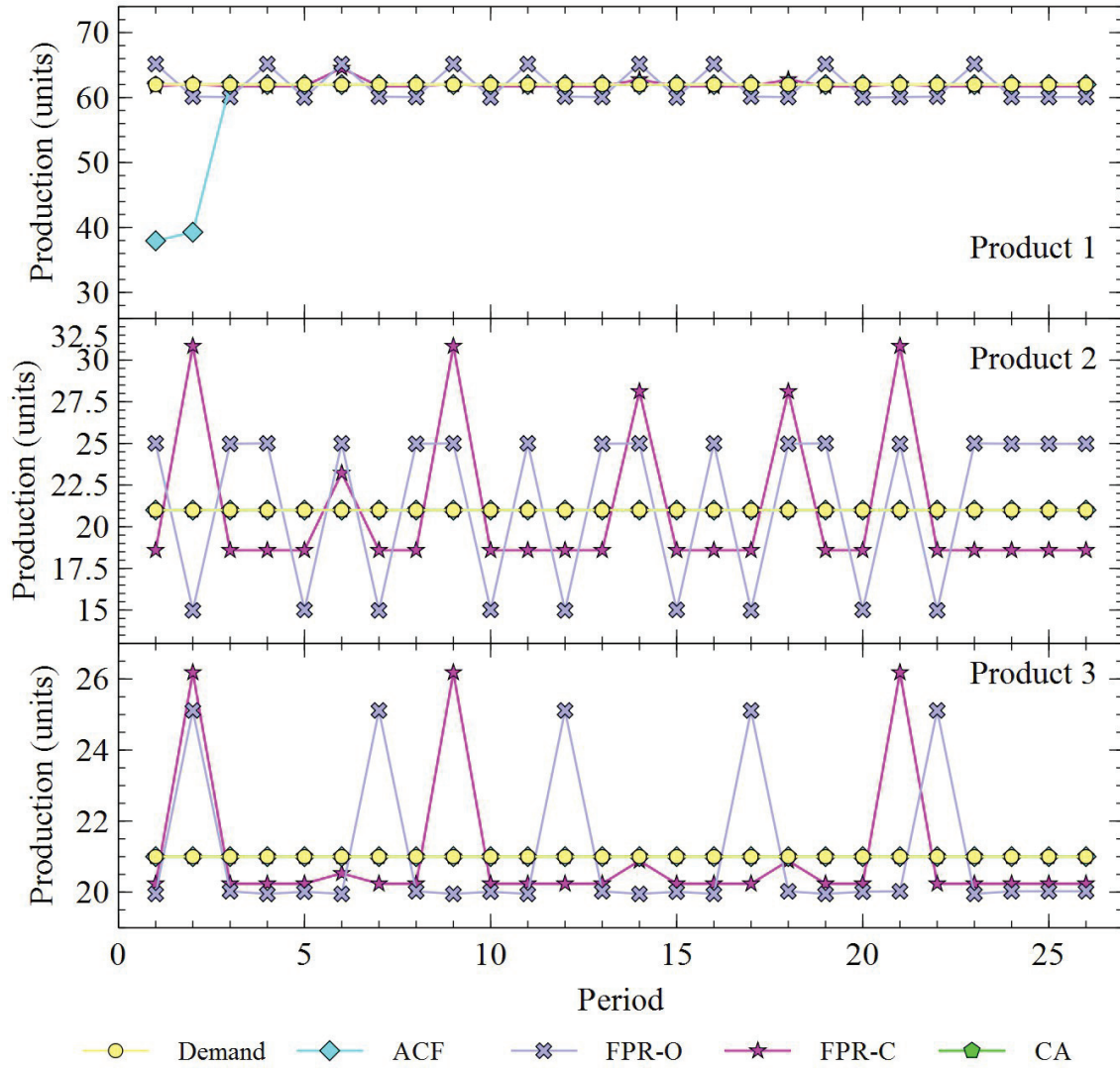


Figure 6.16: Production rate comparison for constant demand based on 90% utilization case

The ACF solution is better than the FPR-O solution for the 70% utilization case, whereas the reverse trend is observed for 90% utilization. For 70% utilization, FPR-C gives a better solution than CA-8, though it uses some inventory and backorders. For 90% utilization, CA-8 gives the best solution.

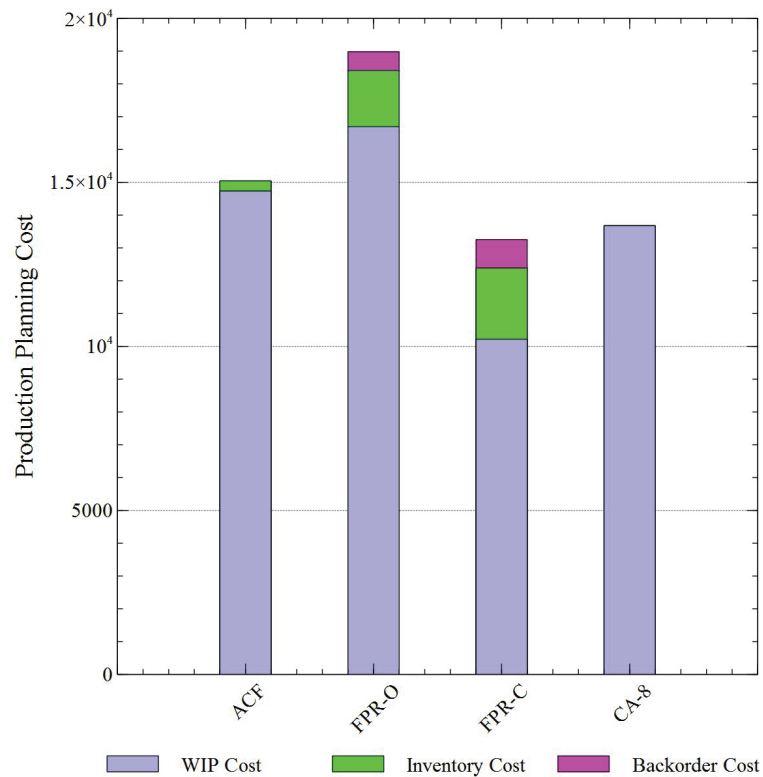


Figure 6.17: Production planning cost results for 70% utilization

6.4.1.2 Varying Demand

In the varying demand scenario in Kacar, Irdem, and Uzsoy [29], the demand is generated based on 70% and 90% utilization levels of the bottleneck machine using the normal distribution. The demand values are shown in Table 6.6 and Table 6.7 for 70% and 90% utilization, respectively. In Figure 6.19 and Figure 6.20, it can be seen that for varying demands, production in ACF, FPR-O, and FPR-C oscillate a little around the demand. In the case of ACF, the oscillation is only for product 1. In CA, production and demand are matched in most instances.

The objective function values for the approaches are shown in Figures 6.21 and 6.22. For 70% utilization with varying demand, FPR-O has a higher cost than ACF, while

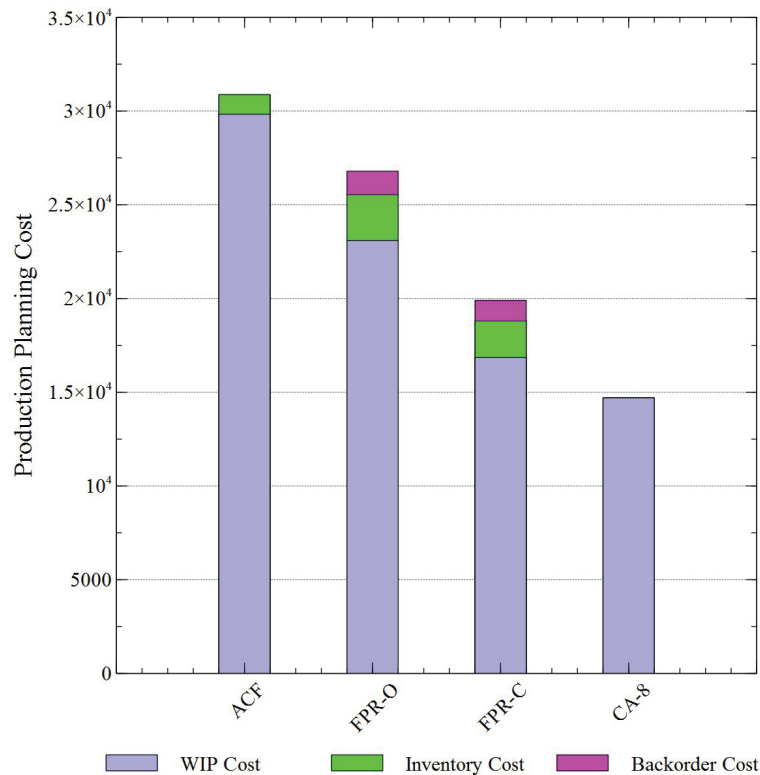


Figure 6.18: Production planning cost results for 90% utilization

it is lower in the two closed queuing network cases (FPR-C and CA-8). In the 90% utilization case with varying demand, the FPR-O solution has a lower cost than ACF.

6.4.2 Mixed Demand

Finally, we generated demands based on the Poisson distribution with a mean of (60, 20, 20). The demand values are shown in Table 6.8.

Figure 6.23 shows the demand and production for each of the cases. Once again, the same behaviour as for the 70% and 90% varying demand scenarios is observed.

The objective function values are shown in Figure 6.24. Again, the same trend as for 87% fixed demand (60-20-20) and 90% utilization with fixed and varying demand is seen.

6.4.3 Production Rate Comparison

Table 6.9 shows a summary of the average absolute value of the difference between demand and production for each of the cases over the 26 periods. As can be seen in

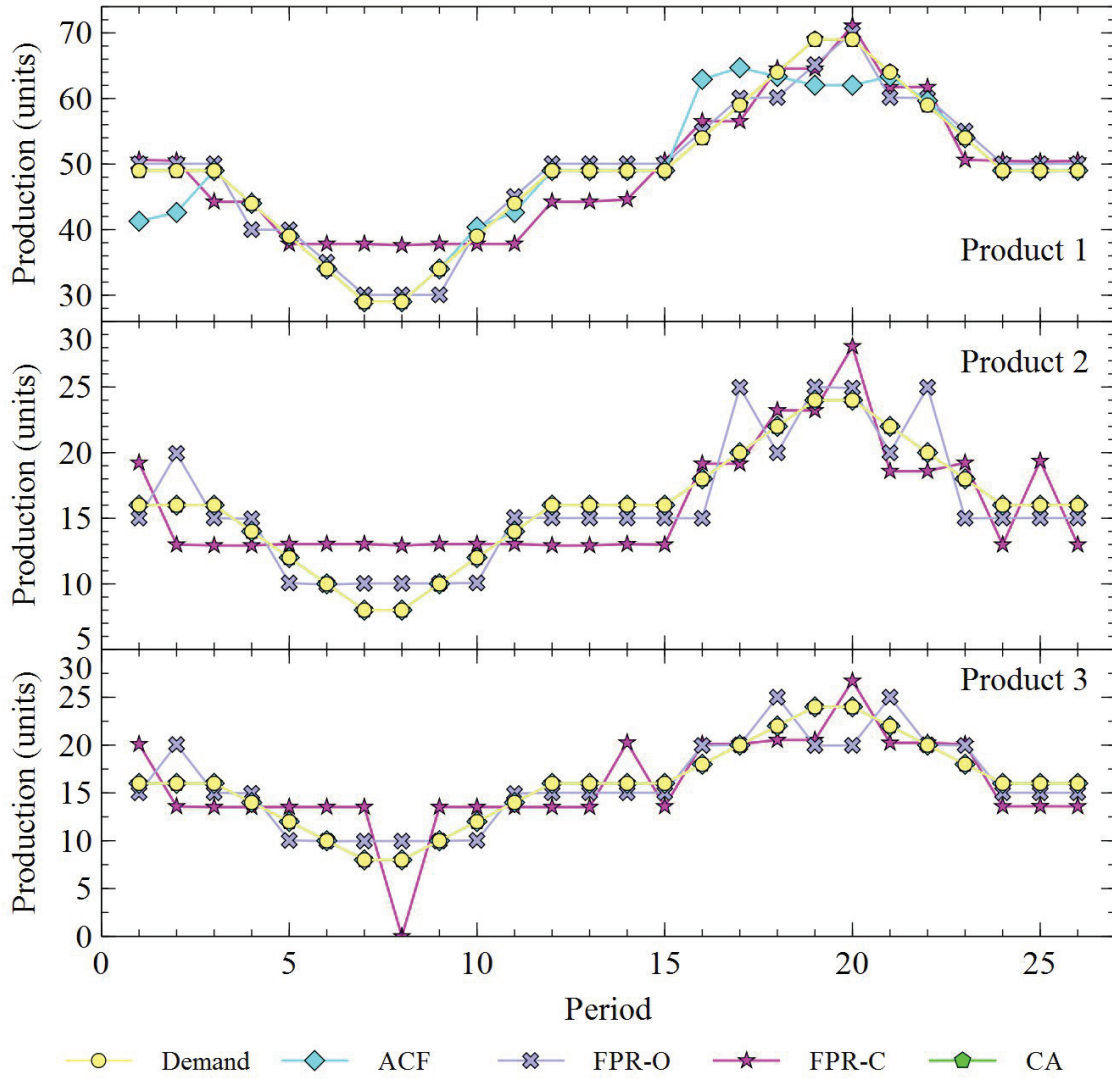


Figure 6.19: Production rate comparison for varying demand based on 70% utilization case

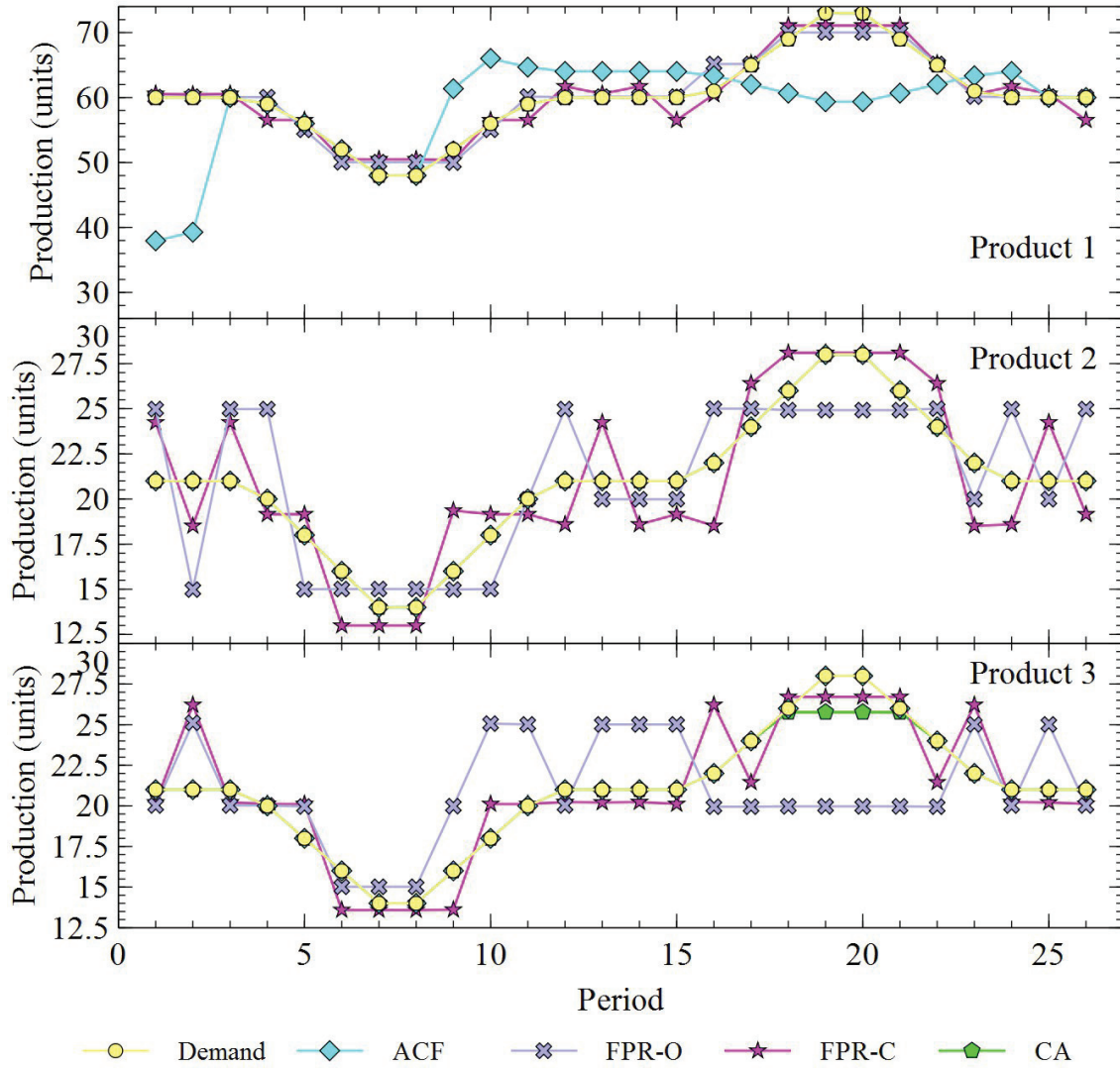


Figure 6.20: Production rate comparison for varying demand based on 90% utilization case

Table 6.6: Demand for 70% utilization

Period	Demand		
	Product 1	Product 2	Product 3
1	49	16	16
2	49	16	16
3	49	16	16
4	44	14	14
5	39	12	12
6	34	10	10
7	29	8	8
8	29	8	8
9	34	10	10
10	39	12	12
11	44	14	14
12	49	16	16
13	49	16	16
14	49	16	16
15	49	16	16
16	54	18	18
17	59	20	20
18	64	22	22
19	69	24	24
20	69	24	24
21	64	22	22
22	59	20	20
23	54	18	18
24	49	16	16
25	49	16	16
26	49	16	16

Table 6.7: Demand for 90% utilization

Period	Demand		
	Product 1	Product 2	Product 3
1	60	21	21
2	60	21	21
3	60	21	21
4	59	20	20
5	56	18	18
6	52	16	16
7	48	14	14
8	48	14	14
9	52	16	16
10	56	18	18
11	59	20	20
12	60	21	21
13	60	21	21
14	60	21	21
15	60	21	21
16	61	22	22
17	65	24	24
18	69	26	26
19	73	28	28
20	73	28	28
21	69	26	26
22	65	24	24
23	61	22	22
24	60	21	21
25	60	21	21
26	60	21	21

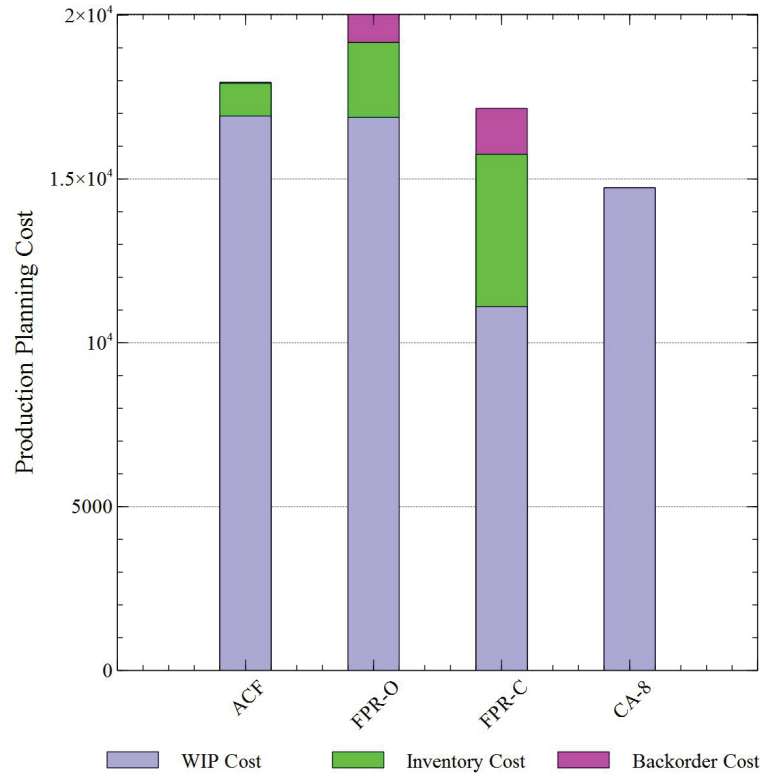


Figure 6.21: Production planning cost for 70% utilization for varying demand

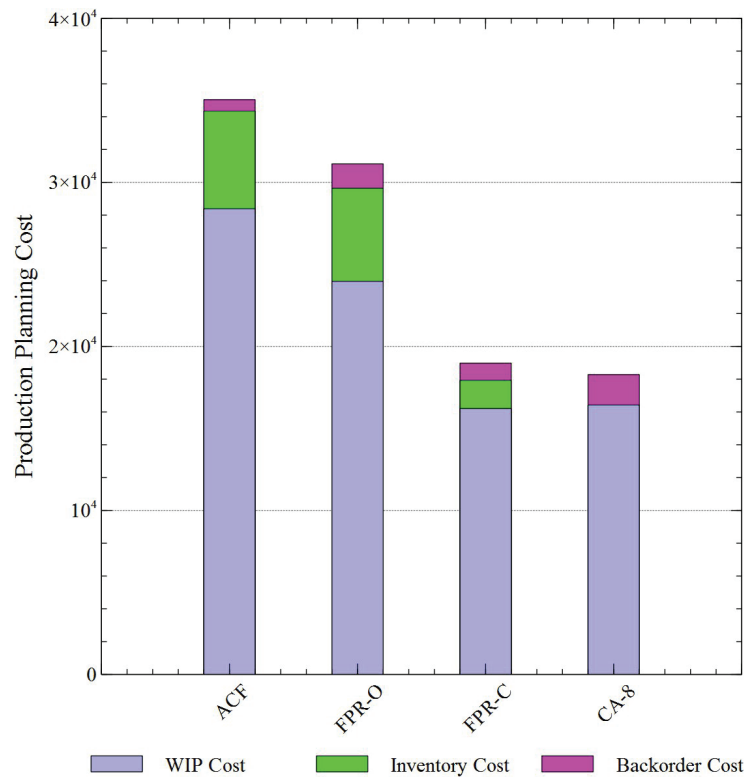


Figure 6.22: Production planning cost for 90% utilization for varying demand

Table 6.8: Demand for Poisson distribution

Period	Demand		
	Product 1	Product 2	Product 3
1	40	18	21
2	69	24	22
3	54	12	10
4	59	31	25
5	71	19	24
6	79	13	21
7	71	24	21
8	59	24	13
9	54	22	15
10	50	20	21
11	66	20	22
12	75	24	21
13	56	26	18
14	62	20	14
15	55	20	15
16	62	21	20
17	54	23	13
18	67	17	22
19	68	21	22
20	65	18	22
21	52	20	23
22	62	19	18
23	62	17	25
24	66	21	18
25	66	17	11
26	44	18	15

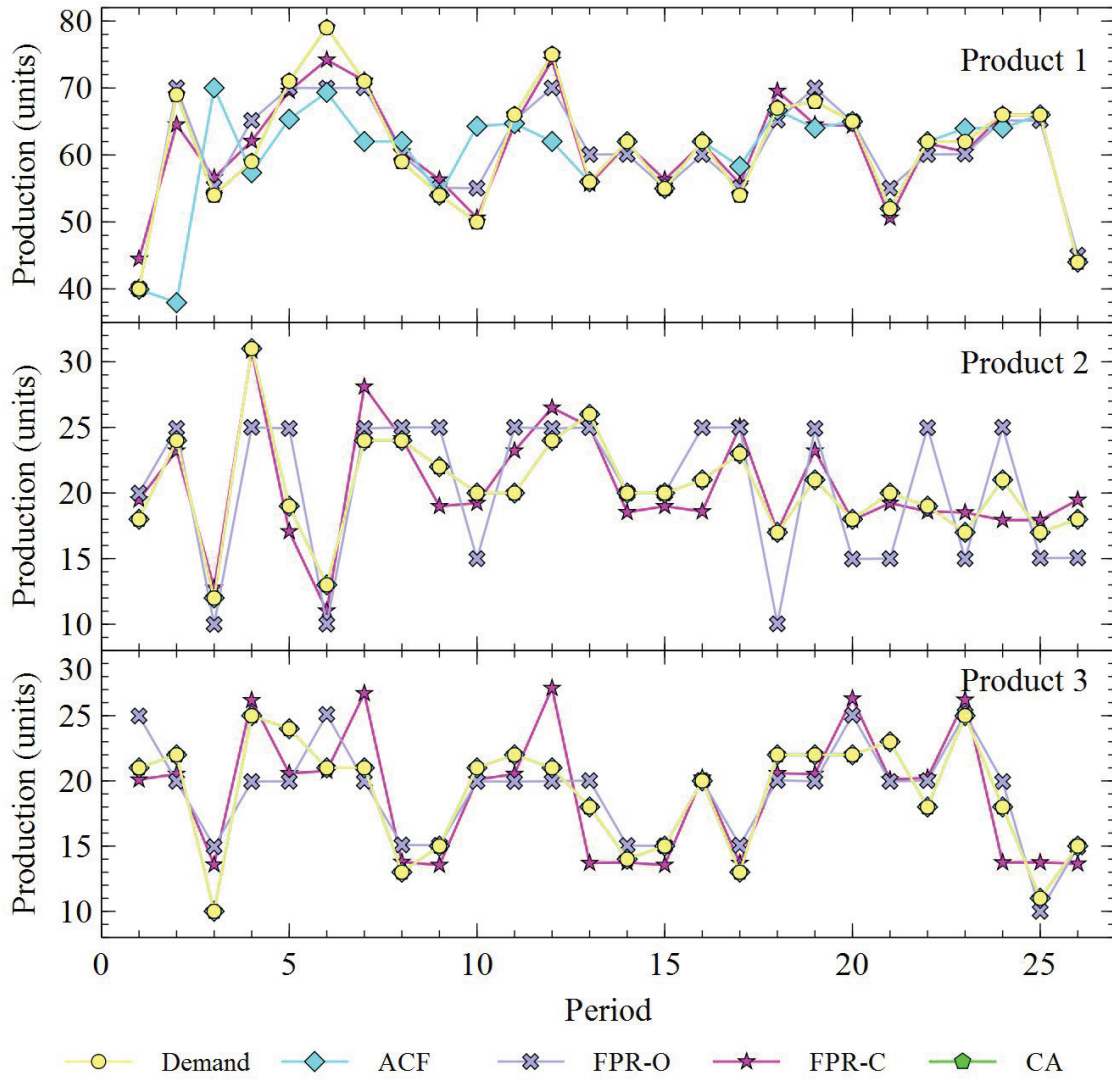


Figure 6.23: Production rate comparison for Poisson demand case

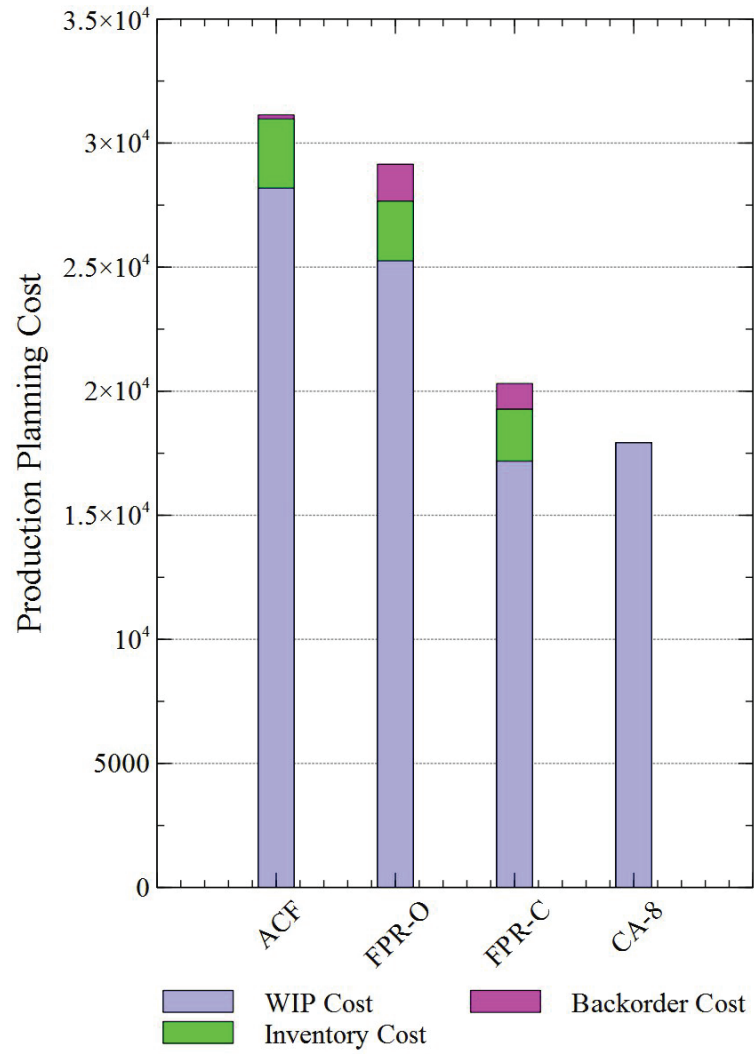


Figure 6.24: Production planning cost results for Poisson distribution

the table, there is a good match between demand and production in ACF and CA, while the FPR-O and FPR-C solutions show larger differences.

Table 6.9: Average value of the absolute difference between demand and production

Case	ACF			FPR-O			FPR-C			CA		
	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3
87% Utilization	1.59	0.00	0.00	0.04	4.98	0.02	1.93	1.91	0.20	0.00	0.00	0.00
Constant 70% Utilization	0.54	0.00	0.00	1.58	1.56	1.58	2.16	3.16	3.01	0.00	0.00	0.00
Constant 90% Utilization	1.80	0.00	0.00	2.35	4.69	1.61	0.38	3.62	1.21	0.00	0.00	0.00
Varying 70% Utilization	1.82	0.00	0.00	1.59	1.72	1.57	3.15	2.50	2.55	0.00	0.00	0.00
Varying 90% Utilization	5.48	0.00	0.00	1.02	2.32	3.36	1.49	2.11	1.54	0.00	0.00	0.19
Poisson	4.51	0.00	0.00	2.11	3.02	1.99	1.60	1.51	2.16	0.00	0.00	0.00

6.4.4 Computational Time

Table 6.10 shows the computation time for all cases considered in this chapter. The LP for ACF takes negligible time to solve while, the CA approach always takes the longest (even with the supplementary variables and constraints). However, the ACF has the additional overhead of piecewise linear-regression estimates for each resource (workstation). The grid size used for the CA approach was finer than what was used for the FPR-O and FPR-C approaches (512 release combinations as opposed to 160).

Table 6.10: Production planning models computational time

	Run Time (secs)			
	ACF	FPR-O	FPR-C	CA
Constant Demand (87%)	0.02	1.73	127	69540
Constant Demand (70%)	0.01	1044	1151	35629
Constant Demand (90%)	0.01	1722	1117	14068
Varying Demand (70%)	0.01	90	1502	39038
Varying Demand (90%)	0.01	8.16	160	35657
Poisson Demand (60, 20, 20)	0.02	27.91	3.4	33275

In the papers by Kacar, Mönch, and Uzsoy [30, 31], a semiconductor fabrication case with 2 products and 210 and 245 operations for products 1 and 2 respectively is considered. This case has over 200 machines grouped into 84 workstations. For this case, developing piecewise linear CF estimates for each of the 84 workstations through simulation experiments would be a significant challenge. In the data-driven FPR and CA approaches, similar simulation runs would be needed, if the WIP-Throughput relationship is built on the open network assumption. However, there would be no need to estimate the CF through regression. For the closed queuing case, the MVA procedure would need to be invoked for each release combination. It is expected that this should be tractable for the problem in Kacar, Mönch, and Uzsoy [30, 31].

In terms of the optimization problem, the ACF LP would be significantly larger, than for the cases considered in this chapter, since it has variables and constraints for both workstations and operations. However, the ACF LP only has linear variables, so solving a problem of this size appears computationally tractable. In the data-driven FPR and CA approaches, the throughput of the entire system is estimated for a given WIP combination. Therefore, the number of workstations and operations

does not change the size of the MILP and the solution times are not expected to take significantly longer for the scaled up wafer fabrication facility. However, the main challenge in scaling up for the FPR and CA would be the number of grid points, which increases exponentially with the number of products. Therefore, an increase in number of products would pose difficulties with both WIP-Throughput estimation as well as with the MILP, since the number of integer variables would increase exponentially.

6.5 Conclusion

This chapter presented a comparison between the ACF, FPR-O, FPR-C, and CA approaches. The solutions in all three methods are comparable. The ACF consistently has a lower production planning cost than the FPR-O in the case of low bottleneck utilization level. However, at high utilization levels, the FPR-O has the lower cost. The FPR-C and CA approaches have successively lower costs than FPR-O because the MVA throughput estimate is consistently higher than the open network simulation estimate. However, the FPR and CA approaches require significantly higher computational time than the ACF.

This chapter also shows that there are differences between the WIP levels in the solutions. While ACF and FPR-O have similar total WIP levels, these are quite different at the product level. However, the closed queuing network approaches such as FPR-C and CA require lower WIPs for the same throughput, since they are based on a closed queuing network representation.

Finally, as mentioned in Section 6.4.1, the nature of the approximations and the errors are different between the three approaches. First, the piecewise regression procedure to estimate the CF lines and also the capacity allocation variables used in the ACF seems to introduce inaccuracies in individual product WIPs. Second, the grid of the FPR representing demand patterns may cause oscillations in the FPR optimal solution. Third, the estimates of throughput for intermediate values of WIP in the grid of the CA are also approximate. This problem can be mitigated by using smaller step sizes, i.e., a larger number of steps.

Chapter 7

Conclusion and Future Research Direction

7.1 Conclusions

This thesis consists of seven chapters. The first chapter introduces the problem of production planning in networks with queuing related congestion. Production planning circularity has been defined in the literature as the problem where lead time is dependent on resource load and resource loading depends on the lead time. Therefore, shop floor congestion should be integrated at the higher managerial level (aggregate planning) to improve the feasibility of the production plans and schedules developed at lower levels. One way to do this is by relating the WIP level to the resource throughput using the CF concept proposed in the literature [19, 35, 47, 3, 1].

In particular, this thesis looks at how the CF can be viably used in production planning for multistage multiproduct systems and as should be clear, presents an alternative approach to the ACF approach in Asmsundsson et al. [3]. The second chapter presents a literature review for the existing state-of-art in the literature covering different production planning tools, their advantages and shortcomings, and internal and external factors affecting production. First, the importance of the production planning tools in reaching a firm's production goal is discussed. The production planning targets set at the higher managerial planning level are communicated as goals to the lower managerial planning level. Unfortunately, when congestion at the shop floor is ignored while generating the production plan at the higher managerial planning level, the plan feasibility at the lower managerial level gets affected.

Moreover, the internal and external factors could also affect production plan feasibility. Internal factors are sometimes under the firm's control but external factors are usually beyond a firm's control. An example of an internal factor not under a firm's control may be an unexpected machine failure. The impact of a failure, however, may be mitigated by the firm, depending on what its recourse measures are. The operational factors that most commonly affect plan feasibility has been surveyed

based on the classification by Yeung, Wong, and Ma [79].

MRP is one of the most popular planning tools at the lower managerial planning level. A MRP plan is constructed based on the fixed lead-time assumption which unlimited production capacity. To overcome this deficiency in the MRP, two approaches have been addressed in the literature. The first approach uses an iterative approach where an optimization model generates a release schedule and a simulation model to observe the flow time of that schedule and provides this as feedback to the optimization model. This iteration continues until some satisfactory criteria are met. The convergence of the iterative approach needs further research [52].

The second approach uses the CF to relate resource utilization to the WIP level which is then incorporated into a production planning optimization model through linearization or piecewise linearization [47, 3]. The CF itself can be derived from the steady-state of the queuing system or the transient state of a production resource, or estimated using a simulation model for the production system [29].

While the approach in Missbauer [47] derives the CF for the bottleneck machine only (based on the $M/G/1$ queue) and for non-bottleneck machines uses just the constant capacity constraint. The deficiency of using a CF for just bottleneck machines is presented in Chapter 3, though we show it for the special case of the $M/G/1$, i.e., the $M/M/1$ queue. In the case of multiple products with a shifting bottleneck machine, using an $M/M/1$ throughput estimate at the bottleneck can significantly overestimate throughput. From this chapter, it becomes clear that the entire multiproduct multi-machine queuing network should be used to estimate the CF.

ACF is the dominant approach in the literature to incorporate the CF into an optimization model [3, 29, 37]. In Kacar, Irdem, and Uzsoy [29], the CF used in the ACF LP model is based on an open queuing network simulation model, where several WIP-Throughput estimates are made. A point to note here is that the CF form has total machine load on the X-axis across multiple visits and products, as is the case in re-entrant manufacturing systems. Similarly, the Y-axis shows throughput across multiple visits and products. The allocation variables Z_{gtl}^k are used to disaggregate throughput for each combination of product, operation, and resource. Since the simulation estimates are based on the entire queuing network, this approach works quite well, though it is not clear that the linear disaggregation is always accurate.

The ACF approach relies on deriving the CF for every resource on the shop floor, which requires too much information about it to simulated. For a large number of resources, this can be time consuming. Also, the implementation in Kacar, Irdem, Uzsoy [29], requires piecewise linearization of the CF from point estimates, which is also time consuming, and is by definition, an approximation.

The focus of this thesis is to develop an alternative approach to the ACF, where the WIP-Throughput relationship point estimates are made for an entire queuing network. These estimates can come from an analytical function, a simulation model, an open queuing network calculator, or using the MVA, which is an approximation method for the closed queuing network. This is a robust method, as it could use any data source that could provide an estimate of system throughput for different WIP levels to a suitable degree of accuracy.

Chapter 4 illustrates this approach through the FPR MILP. Here, point estimates of throughput are made on a resource load grid that is dimensioned by product. The illustrative examples in the chapter use MVA, though any point estimation method could be used. The restriction in the model is that the load choice (WIP + release) in each period should correspond to a grid point. The FPR approach is data-driven, because no attempt is made to develop the piecewise linear function through regression or any other curve fitting method. We show how using this model leads to a different (more realistic) solution from the fixed capacity model and look at the behaviour of the model based on different factors such as backorder costs and demand. The computational run time is also calibrated for random problems. The results illustrate that increasing step size increases the accuracy of the method, which manifests itself in a lower optimal objective function value. Naturally, there is a trade-off between accuracy requirements and computational run time. A supplementary variable and constraint approach is presented to reduce run time. Here, production quantities deviating from demand are penalized in the objective function. Also, limits are placed on the inventory and backorder contribution in each period. Once again, when the penalty in the objective function is low and the limits are tight, the solution is further away from the optimal solution to the original problem, but quicker to obtain.

Chapter 5 presents the CA approach, which can be seen as an interpolative method. In this chapter, point estimates of network throughput are made at the

corner points of the load cuboid, dimensioned by product. Effectively, this is an interpolative approach where the slopes in each cuboid are averaged to get an estimate of throughputs for a load combination of products. The advantage of this method is that the load (WIP + release) is no longer restricted to discrete choices. In the chapter, the CA and FPR methods are compared. The cubic approximation is better able to match the supply with the demand than FPR. Furthermore, the effect of step size and planning horizon length on the quality solution obtained and computational time is examined. There is a trade-off between optimal solution accuracy and run time. When the step size is high, solutions are more accurate but take longer to find.

Chapter 6 presents a comparison between ACF, FPR, and CA. The ACF seems to introduce inaccuracies in individual product WIPs. This is the net result of the piecewise regression procedure to estimate the CF lines and also the allocation of capacity through the Z_{gtl}^k variables. The FPR method uses exact throughput estimates at discrete points, making it in some sense more accurate than ACF. However, since the WIP choices are discrete, the MIP optimal solution is only approximate because if a release combination is not on the grid, the solution will need to produce too much or too little in a period. The CA can better match demand, but because of the discrete steps involved, the estimates of throughput for intermediate values of WIP in the grid are approximate.

Kacar, Mönch, and Uzsoy [31] state that “ACF requires explicit representation of WIP at the level of individual operations, which greatly increasing the number of both decision variables and constraints over”. Adding to that, “The piecewise linearization of the CF and the additional allocation variables required by the ACF model increase the size of the formulation even further”. Some of these problems are circumvented in FPR and CA. However, the load choices are discrete, making these approaches challenging as the number of products is increased.

7.2 Future Research Directions

Some areas for future research are:

- Further study to obtain insights on the differences between WIP levels in the partitioning CF proposed by Asmundsson, Rardin, Turkseven, and Uzsoy [3] and the CF-derived using MVA for a given throughput.

- Using neural networks to estimate the WIP-Throughput relationship.
- Considering the case of open queuing networks. The challenge here would be to define the cuboid grid structure such that the network is stable.
- Finding solutions using the FPR and CA methods for scaled up problems such as the one in Kacar, Mönch, and Uzsoy [30, 31]. This may involve using heuristic or metaheuristic algorithms to obtain good solutions with reasonable run time.
- The use of non-separable convex programming approaches to solve the aggregate planning problem with an embedded queuing network CF. One approach would be the triangulation approach of Vielma and Nemhauser [75] to linearize the CF. The other might be to relax the binary variable Z_{rp} in FPR to be a continuous variable between 0 and 1, and make substitutions for WIP level W_{gp} , inventory I_{gp} and backorder B_{gp} in the objective function. A branch and bound procedure may then be developed based on the cuboid neighbourhood structure. In CA, the analogous method would be to relax the ζ_p^c variables.
- Representing different parts of a manufacturing network as queuing networks and piecing them together. The throughput of each queuing network could be estimated analytically, computationally, or through simulation. This approach could eventually help us understand the dynamics of a supply chain network with several entities, each having congestion issues either in manufacturing or distribution.

Bibliography

- [1] Erinc Albey, Ümit Bilge, and Reha Uzsoy. An exploratory study of disaggregated clearing functions for production systems with multiple products. *International Journal of Production Research*, 52(18):5301–5322, 2014.
- [2] Ronald G Askin and Charles R Standridge. *Modeling and analysis of manufacturing systems*, volume 29. Wiley New York, 1993.
- [3] Jakob Asmundsson, Ronald L Rardin, Can Hulusi Turkseven, and Reha Uzsoy. Production planning with resources subject to congestion. *Naval Research Logistics (NRL)*, 56(2):142–157, 2009.
- [4] Jakob Asmundsson, Ronald L Rardin, and Reha Uzsoy. Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. *Semiconductor Manufacturing, IEEE Transactions on*, 19(1):95–111, 2006.
- [5] Haldun Aytug, Mark A Lawley, Kenneth McKay, Shantha Mohan, and Reha Uzsoy. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86–110, 2005.
- [6] U Narayan Bhat. *An introduction to queueing theory: modeling and analysis in applications*. Birkhäuser, 2015.
- [7] Marc A Bodt and Luk N Wassenhove. Cost increases due to demand uncertainty in mrp lot sizing. *Decision Sciences*, 14(3):345–362, 1983.
- [8] John A Buzacott and J George Shanthikumar. *Stochastic models of manufacturing systems*, volume 4. Prentice Hall Englewood Cliffs, NJ, 1993.
- [9] MD Byrne and MA Bakir. Production planning using a hybrid simulation–analytical approach. *International Journal of Production Economics*, 59(1):305–311, 1999.
- [10] Robert C Carlson, Sara L Beckman, and Dean H Kropp. The effectiveness of extending the horizon in rolling production scheduling. *Decision Sciences*, 13(1):129–146, 1982.
- [11] Stephen L Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent & Fuzzy Systems*, 2(3):267–278, 1994.
- [12] Chen-Hua Chung and Lee J Krajewski. Notes and communications replanning frequencies for master production schedules. *Decision Sciences*, 17(2):263–273, 1986.

- [13] Yves Dallery, Rene David, and Xiao-Lan Xie. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE transactions*, 20(3):280–283, 1988.
- [14] Alexandre Dolgui and Caroline Prodhon. Supply planning under uncertainties in mrp environments: A state of the art. *Annual Reviews in Control*, 31(2):269–279, 2007.
- [15] Robert Fildes and B Kingsman. Incorporating demand uncertainty and forecast error in supply chain planning models&star. *Journal of the Operational Research Society*, 62(3):483–500, 2011.
- [16] Ram Ganeshan, Tonya Boone, and Alan J Stenger. The impact of inventory and flow planning parameters on supply chain performance: An exploratory study. *International Journal of Production Economics*, 71(1):111–118, 2001.
- [17] Björn Geißler, Alexander Martin, Antonio Morsi, and Lars Schewe. Using piecewise linear functions for solving minlps. In *Mixed Integer Nonlinear Programming*, pages 287–314. Springer, 2012.
- [18] Edward T Grasso and Bernard W Taylor III. A simulation-based experimental investigation of supply/timing uncertainty in mrp systems. *The International Journal Of Production Research*, 22(3):485–497, 1984.
- [19] Stephen C Graves. A tactical planning model for a job shop. *Operations Research*, 34(4):522–533, 1986.
- [20] Stephen C Graves. Uncertainty and production planning. In *Planning Production and Inventories in the Extended Enterprise*, pages 83–101. Springer, 2011.
- [21] Steven T Hackman and Robert C Leachman. A general framework for modeling production. *Management Science*, 35(4):478–495, 1989.
- [22] C-J Ho, W-K Law, and R Rampal. Uncertainty-dampening methods for reducing mrp system nervousness. *The International Journal Of Production Research*, 33(2):483–496, 1995.
- [23] Wallace J Hopp and ML Spearman. *Factory Physics: Foundations of Manufacturing Management*. Irwin/McGraw-Hill, Boston, 1996.
- [24] Wallace J Hopp and ML Spearman. *Factory physics: Foundations of manufacturing management*, 2000, 2008.
- [25] Yi-Feng Hung and Meng-Chi Hou. A production planning approach based on iterations of linear programming optimization and flow time prediction. *Journal of the Chinese Institute of Industrial Engineers*, 18(3):55–67, 2001.

- [26] Yi-Feng Hung and Robert C Leachman. A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *Semiconductor Manufacturing, IEEE Transactions on*, 9(2):257–269, 1996.
- [27] D Fatih Irtem, N Baris Kacar, and Reha Uzsoy. An exploratory analysis of two iterative linear programming simulation approaches for production planning. *Semiconductor Manufacturing, IEEE Transactions on*, 23(3):442–455, 2010.
- [28] Necip Baris Kacar. Developing and fitting a clearing function form: An experimental comparison of a clearing function model and iterative simulation-optimization algorithm for production planning of a semiconductor fab. 2009.
- [29] Necip Baris Kacar, Durmus Fatih Irtem, and Reha Uzsoy. An experimental comparison of production planning using clearing functions and iterative linear programming-simulation algorithms. *IEEE Transactions on Semiconductor Manufacturing*, 25(1):104–117, 2012.
- [30] Necip Baris Kacar, Lars Mönch, and Reha Uzsoy. Planning wafer starts using nonlinear clearing functions: A large-scale experiment. *Semiconductor Manufacturing, IEEE Transactions on*, 26(4):602–612, 2013.
- [31] Necip Baris Kacar, Lars Mönch, and Reha Uzsoy. Modeling cycle times in production planning models for wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, 29(2):153–167, 2016.
- [32] Necip Baris Kacar and Reha Uzsoy. A comparison of multiple linear regression approaches for fitting clearing functions to empirical data. *International Journal of Production Research*, 52(11):3164–3184, 2014.
- [33] Necip Baris Kacar and Reha Uzsoy. Estimating clearing functions for production resources using simulation optimization. *IEEE Transactions on Automation Science and Engineering*, 12(2):539–552, 2015.
- [34] Yongha Kang, Erinc Albey, Sangwook Hwang, and Reha Uzsoy. The impact of lot-sizing in multiple product environments with congestion. *Journal of Manufacturing Systems*, 33(3):436–444, 2014.
- [35] Uday S Karmarkar. Capacity loading and release planning with work-in-progress (wip) and leadtimes. *Journal of Manufacturing and Operations Management*, 2(105-123), 1989.
- [36] David Kayton, Tim Teyner, Christopher Schwartz, and Reha Uzsoy. Focusing maintenance improvement efforts in a wafer fabrication facility operating under the theory of constraints. *Production and Inventory Management Journal*, 38(4):51, 1997.

- [37] Ali Kefeli and Reha Uzsoy. Identifying potential bottlenecks in production systems using dual prices from a mathematical programming model. *International Journal of Production Research*, pages 1–19, 2015.
- [38] Ali Kefeli, Reha Uzsoy, Yahya Fathi, and Michael Kay. Using a mathematical programming model to examine the marginal price of capacitated resources. *International Journal of Production Economics*, 131(1):383–391, 2011.
- [39] Bokang Kim and Sooyoung Kim. Extended model for a hybrid production planning approach. *International Journal of Production Economics*, 73(2):165–173, 2001.
- [40] Sukgon Kim and Reha Uzsoy. Modeling and analysis of integrated planning of production and engineering process improvement. *Semiconductor Manufacturing, IEEE Transactions on*, 26(3):414–422, 2013.
- [41] JFC Kingman. The single server queue in heavy traffic. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 57, pages 902–904. Cambridge Univ Press, 1961.
- [42] SC Koh, MH Jones, SM Saad, S Arunachalam, and A Gunasekaran. Measuring uncertainties in mrp environments. *Logistics Information Management*, 13(3):177–183, 2000.
- [43] Stephen R Lawrence and Arnold H Buss. Shifting production bottlenecks: causes, cures, and conundrums. *Production and operations management*, 3(1):21–37, 1994.
- [44] SC Lenny Koh and Mike Simpson. Change and uncertainty in sme manufacturing environments using erp. *Journal of Manufacturing Technology Management*, 16(6):629–653, 2005.
- [45] Neng-Pai Lin, Lee Krajewski, G Keong Leong, and WC Benton. The effects of environmental factors on the design of master production scheduling systems. *Journal of Operations Management*, 11(4):367–384, 1994.
- [46] Alan S Manne. A note on the modigliani-hohn production smoothing model. *Management Science*, 3(4):371–379, 1957.
- [47] Jyotiprasad Medhi. *Stochastic models in queueing theory*. Academic Press, 2002.
- [48] Mary J Meixell. The impact of setup costs, commonality, and capacity on schedule stability: An exploratory study. *International Journal of Production Economics*, 95(1):95–107, 2005.
- [49] Herbert Meyr, Michael Wagner, and Jens Rohde. Structure of advanced planning systems. In *Supply chain management and advanced planning*, pages 99–106. Springer, 2015.

- [50] H Missbauer. Aggregate order release planning for time-varying demand. *International Journal of Production Research*, 40(3):699–718, 2002.
- [51] Hubert Missbauer. Models of the transient behaviour of production units to optimize the aggregate material flow. *International Journal of Production Economics*, 118(2):387–397, 2009.
- [52] Hubert Missbauer and Reha Uzsoy. Optimization models of production planning problems. In *Planning Production and Inventories in the Extended Enterprise*, pages 437–507. Springer, 2011.
- [53] Franco Modigliani and Franz E Hohn. Production planning over time and the nature of the expectation and planning horizon. *Econometrica, Journal of the Econometric Society*, pages 46–66, 1955.
- [54] S. Mrishih. Addressing the production circularity problem in conwip environments. Technical report, Dalhousie University, Industrial Engineering. Halifax, Nova Scotia, 2012.
- [55] MK Omar and JA Bennell. Revising the master production schedule in a hpp framework context. *International Journal of Production Research*, 47(20):5857–5878, 2009.
- [56] Rayan Saleem M Omar. Improving the stability of supply chain operations planning considering the effects of congestion. Master’s thesis, Concordia University, 2012.
- [57] Seza Orcun, Reha Uzsoy, and Karl Kempf. Using system dynamics simulations to compare capacity models for production planning. In *Proceedings of the 38th conference on Winter simulation*, pages 1855–1862. Winter Simulation Conference, 2006.
- [58] Seza Orcun, Reha Uzsoy, and Karl G Kempf. An integrated production planning model with load-dependent lead-times and safety stocks. *Computers & Chemical Engineering*, 33(12):2159–2163, 2009.
- [59] Julia Pahl, Stefan Voß, and David L Woodruff. Production planning with load dependent lead times: an update of research. *Annals of Operations Research*, 153(1):297–345, 2007.
- [60] Youngshin Park, Sooyoung Kim, and Chi-Hyuck Jun. Mean value analysis of reentrant line with batch machines and multi-class jobs. *Computers & Operations Research*, 29(8):1009–1024, 2002.
- [61] I Nyoman Pujawan. Schedule instability in a supply chain: an experimental study. *International Journal of Inventory Research*, 1(1):53–66, 2008.
- [62] Martin Reiser and Stephen S Lavenberg. Mean-value analysis of closed multi-chain queuing networks. *Journal of the ACM (JACM)*, 27(2):313–322, 1980.

- [63] HJ Roberts and PRN Barrar. Mrpii implementation: key factors for success. *Computer Integrated Manufacturing Systems*, 5(1):31–38, 1992.
- [64] Funda Sahin, Arunachalam Narayanan, and E Powell Robinson. Rolling horizon planning in supply chains: review, implications and directions for future research. *International Journal of Production Research*, 51(18):5413–5436, 2013.
- [65] Baris Selçuk, Jan C Fransoo, and AG De Kok. Work-in-process clearing in supply chain operations planning. *IIE Transactions*, 40(3):206–220, 2008.
- [66] David Simchi-Levi, Edith Simchi-Levi, and Philip Kaminsky. *Designing and managing the supply chain: Concepts, strategies, and cases*. McGraw-Hill United-States, 2007.
- [67] Mark L Spearman. An analytic congestion model for closed production systems with ifr processing times. *Management Science*, 37(8):1015–1029, 1991.
- [68] Sri V Sridharan, William L Berry, and V Udayabhanu. Measuring master production schedule stability under rollin. *Decision Sciences*, 19(1):147, 1988.
- [69] V Sridharan, William L Berry, and V Udayabhanu. Freezing the master production schedule under rolling planning horizons. *Management Science*, 33(9):1137–1149, 1987.
- [70] V Sridharan and R Lawrence LaForge. The impact of safety stock on schedule instability, cost and service. *Journal of Operations Management*, 8(4):327–347, 1989.
- [71] Ashok Srinivasan, Malachy Carey, and Thomas E Morton. Resource pricing and aggregate scheduling in manufacturing systems. Technical report, Carnegie Mellon University, Tepper School of Business, 1988.
- [72] Kathryn E Stecke. A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems. *European Journal of Operational Research*, 24(3):369–378, 1986.
- [73] Rajan Suri and Richard R Hildebrant. Modelling flexible manufacturing systems using mean-value analysis. *Journal of Manufacturing Systems*, 3(1):27–38, 1984.
- [74] Reha Uzsoy and Cheng-Shuo Wang. Performance of decomposition procedures for job shop scheduling problems with bottleneck machines. *International Journal of Production Research*, 38(6):1271–1286, 2000.
- [75] Juan Pablo Vielma and George L Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1-2):49–72, 2011.
- [76] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*. Prentice Hall Boston, 2012.

- [77] Urban Wemmerlöv. A time-phased order-point system in environments with and without demand uncertainty: a comparative analysis of non-monetary performance variables. *International journal of production research*, 24(2):343–358, 1986.
- [78] D Clay Whybark and J Gregg Williams. Material requirements planning under uncertainty. *Decision sciences*, 7(4):595–606, 1976.
- [79] JHY Yeung, WCK Wong, and L Ma. Parameters affecting the effectiveness of mrp systems: a review. *International journal of production research*, 36(2):313–332, 1998.
- [80] Xiande Zhao and TS Lee. Freezing the master production schedule for material requirements planning systems under demand uncertainty. *Journal of Operations Management*, 11(2):185–205, 1993.

Appendices

Appendix A

Chapter 3

A.1 Python Code for Computing MVA

```
#Combinations function
#This function picks all possible combinations from a list
def myCombinations(list):
    r=[]
    for x in list:
        t = []
        for y in x:
            for i in r:
                t.append(i+[y])
        r = t
    return r;

# Mean value analysis algorithm

def MVA(epsilon, num_products, num_stations, N, Z, c, mu_inv, v):
    C = [];
    s = 0;
    L = [];
    Looping = True;
    while Looping:
        #Step 111
        for j in range(num_stations):
            L_temp = [];
            for p in range(num_products):
                L_temp.append(float(N[p])/float(Z[p]));
```

```

        L.append(L_temp);

#print "L is", L;
#Step 2
W = [];
for j in range(num_stations):
    W_temp = [];
    for p in range(num_products):
        temp1 = float(mu_inv[j][p])/float(c[j])* \
(1+float(N[p]-1)/float(N[p])*L[j][p])
        temp2 = 0;
        for r in range(num_products):
            if (r != p) & (mu_inv[j][p] !=0.0):
                temp2 = temp2 + float(L[j][r])* \
float(mu_inv[j][r])/ \
float(c[j]);
            #print "j= ",j , "p = ", p, "Temp2 is: ", temp2;
            if c[j] > 1:
                temp3 = float(c[j]-1)/float(c[j])* \
float(mu_inv[j][p]);
            else:
                temp3 = 0;
            W_temp.append(temp1+temp2+temp3);
    W.append(W_temp);
#print "W is", W;

#Step 3
X=[];
for p in range(num_products):
    x_temp_denominator = 0;

```



```

        for j in range(num_stations):
            x_temp_denominator = x_temp_denominator + \
float(v[j][p])*W[j][p];
            x_temp = float(N[p])/float(x_temp_denominator);
            X.append(x_temp);
#print "X is", X;

#Step 4-1
#print "Computing L_new"
L_new = [];
for j in range(num_stations):
    L_temp = [];
    for p in range(num_products):
        L_temp.append(float(X[p])*float(v[j][p])* \
float(W[j][p]));
        #print "X[p] is ", X[p];
        #print "v[j][p] is ", v[j][p];
        #print "W1[j][p] is ", W[j][p];
        #print "L_temp is", L_temp;
    L_new.append(L_temp);
#print "L_new is ", L_new;

#Step 4-2
Looping2 = False;
for j in range(num_stations):
    for p in range(num_products):
        if L_new[j][p] > 0.0:
            if (L_new[j][p] - L[j][p])/L_new[j][p] > epsilon:
                Looping2 = True;
if Looping2:
    L = L_new;
Looping = Looping2;

```

```

        #print "Done, L (final) is ", L;
        #print "Done, X (final) is ", X;

        #print len(X);
        #print X[0];

    C_temp = [];
    for p in range(num_products):
        C_temp.append(X[p]);
    C.append(C_temp);

    return C;

epsilon = 0.001;
num_products = 1;
num_stations = 11;
num_hours = 24;
num_minutes = 60;
num_days = 7;
N = [22];
Z = [22];
c = [1,1,1,1,1,1,1,1,1,1,1,1];
mu_inv = [[80],[220],[45],[40],[25],[22],[20], \
[100],[50],[50],[70]];
v = [[3],[1],[1],[6],[3],[2],[3],[1],[1],[1],[0]];
#print "mu_inv is" ,mu_inv[1];
C = [];

#Call mean value analysis function once to test.

C = MVA(epsilon, num_products, num_stations, [12.3804], \
    Z, c, mu_inv, v);

```

```

for i in range(len(C[0])):
    C[0][i] = C[0][i];
    C[0][i] = C[0][i] * 60*8*7;
print ("C MVA for Product 1 i:\n", C[0]);

#Call mean value analysis function within a loop to find \
throughput estimates for different combinations of input products.

step_size = 10;
num_iterations = 1;

R = [];
for p in range(num_products):
    R_temp = [];
    inventory_value = 10;
    for i in range(num_iterations):
        R_temp.append(inventory_value);
        inventory_value = inventory_value + step_size;
    R.append(R_temp);
#print "R is : ", R;

#Inventory combinations
I = myCombinations(R);

##print "Length of I is: ",len(I);
print ("I is : ", I);

#Throughput for each inventory combination

T = [];
for i in range(len(I)):
    T.append(MVA(epsilon, num_products, num_stations, \

```

```

I[i], Z, c, mu_inv, v));

print ("I and T are:");

for i in range(len(T)):
    print (I[i], T[i]);

###Open a file
##filep = open("data.txt", "w");
##filep.write("param Q := \n");
##for g in range(num_products):
##    for r in range(len(I)):
##        w_string = str(g+1);
##        w_string += " ";
##        w_string += str(r+1);
##        w_string += " ";
##        w_string += str(I[r][g]);
##        filep.write("%s \n" %w_string );
##filep.write(";\n");
##
##filep.write("param O := \n");
##for g in range(num_products):
##    for r in range(len(T)):
##        w_string = str(g+1)print "C [6 3] is:\n", C[0];;
##        w_string += " ";
##        w_string += str(r+1);
##        w_string += " ";
##        w_string += str(T[r][0][g]*num_hours*num_minutes*num_days);
##        filep.write("%s \n" %w_string );
##filep.write(";\n");
##filep.close();

```

Appendix B

Chapter 4

B.1 Python Code for the Optimization Model

B.1.1 Main Code

```
#Replication Difference function of run time
def findR_T(timeSet):
    R_T = [];
# The Second Case
##     max_i, min_i = 0, 0
##
##     for i in range(len(timeSet)):
##         if timeSet[i] > timeSet[max_i]:
##             max_i = i
##         if timeSet[i] < timeSet[min_i]:
##             min_i = i
##     R_T.append(timeSet[max_i]-timeSet[min_i])
    for i in range(len(timeSet)-1):
##         R_T.append(timeSet[i+1]-timeSet[i])
# The First Case
        R_T.append(abs(timeSet[i+1]-timeSet[i]))
    #print "R_T is ", R_T;
    return R_T;

#Replication Difference function
def findR(objValueSet):
    R = [];
```

```

# The Second Case

##     max_i, min_i = 0, 0
##
##     for i in range(len(objValueSet)):
##         if objValueSet[i] > objValueSet[max_i]:
##             max_i = i
##         if objValueSet[i] < objValueSet[min_i]:
##             min_i = i
##     R.append(objValueSet[max_i]-objValueSet[min_i])
    for i in range(len(objValueSet)-1):
##         R.append(objValueSet[i+1]-objValueSet[i])
        # The First Case
        R.append(abs(objValueSet[i+1]-objValueSet[i]))
    #print "R is ", R;
    return R;

#Combinations function
#This function picks all possible combinations from a list
def myCombinations(list):
    r=[]
    for x in list:
        t = []
        for y in x:
            for i in r:
                t.append(i+[y])
        r = t
    return r;

#MVA calling function for fixed inventory combinations

```

```

#This function picks fixed inventory combinations I and calls MVA\
for each combination

def get_QO_fromMVA(num_products, epsilon, \
                  num_stations,num_hours,num_minutes,num_days, \
                  N,Z,c,mu_inv,v,I):
    from Closed_ISERC import MVA

    #Call mean value analysis function within a loop to find\
    throughput estimates for different combinations of input products.

    #print "Length of I is: ",len(I);
    #print ("I is :", I);

    #Throughput for each inventory combination
    T = [];
    for i in range(len(I)):
        T.append(MVA(epsilon, num_products, num_stations,\
I[i], Z, c, mu_inv, v));

    #print ("I and T are:");

    ##for i in range(len(T)):
    ##    print I[i], T[i];

    Q={}
    O={}
    #Open a file
    filep = open("data.txt", "w");
    filep.write("param Q := \n");
    for g in range(num_products):
        for r in range(len(I)):

```

```

    Q[g+1,r+1]=I[r][g] #Create Q
    w_string = str(g+1);
    w_string += " ";
    w_string += str(r+1);
    w_string += " ";
    w_string += str(I[r][g]);
    filep.write("%s \n" %w_string );
filep.write(";\n");

filep.write("param 0 := \n");
for g in range(num_products):
    for r in range(len(T)):
        #print "g: %s, r: %s"%(g+1,r+1)
        O[g+1,r+1]=float(T[r][0][g]*num_hours*\
num_minutes*num_days)
        w_string = str(g+1);
        w_string += " ";
        w_string += str(r+1);
        w_string += " ";
        w_string += str(T[r][0][g]*num_hours*\
num_minutes*num_days);
        filep.write("%s \n" %w_string );
filep.write(";\n");
filep.close();
#print "0:", 0

#print "num"

return Q, 0, len(I)

#MVA calling function for iterative inventory combinations
#This function picks different inventory combinations and\

```



```

calls MVA for each combination

def get_QO_iterative(num_products,num_iterations, step_size,epsilon, \
                    num_stations,num_hours,num_minutes,num_days, \
                    N,Z,c,mu_inv,v):
    from Closed_ISERC import MVA

    #Call mean value analysis function within a loop to find \
    throughput estimates for different combinations of input products.

    R = [];
    for p in range(num_products):
        R_temp = [];
        inventory_value = 1;
        for i in range(num_iterations):
            R_temp.append(inventory_value);
            inventory_value = inventory_value + step_size;
        R.append(R_temp);
    #print "R is : ", R;
    #print "len R:", len(R)

    #Inventory combinations
    I = myCombinations(R);

    #print "Length of I is: ",len(I);
    #print ("I is :", I);

    #Throughput for each inventory combination
    T = [];
    for i in range(len(I)):
        T.append(MVA(epsilon, num_products, num_stations,\
                    I[i], Z, c, mu_inv, v));

```

```

#print ("I and T are:");

##for i in range(len(T)):
##    print I[i], T[i];

Q={}
O={}

#Open a file
#filep = open("data.txt", "w");
#filep.write("param Q := \n");
for g in range(num_products):
    for r in range(len(I)):
        Q[g+1,r+1]=I[r][g] #Create Q
        '''w_string = str(g+1);
        w_string += " ";
        w_string += str(r+1);
        w_string += " ";
        w_string += str(I[r][g]);'''
        #filep.write("%s \n" %w_string );
#filep.write("; \n");

#filep.write("param O := \n");
for g in range(num_products):
    for r in range(len(T)):
        #print "g: %s, r: %s"%(g+1,r+1)
        O[g+1,r+1]=float(T[r][0][g]*num_hours*\
num_minutes*num_days)
        '''w_string = str(g+1);
        w_string += " ";
        w_string += str(r+1);
        w_string += " ";

```

```

        w_string += str(T[r][0][g]*num_hours*\
num_minutes*num_days);
        filep.write("%s \n" %w_string );'''
#filep.write(";\n");
#filep.close();
#print "0:", 0

#print "num"

return Q, 0, len(I)

def callGurobi(products,Q,0,E,h,b,w,x,T,d,d_factor):
    import pyomo.environ as cpr
    from pyomo.opt import SolverFactory

##### Create Model #####
    m=cpr.ConcreteModel()

#####
##### Sets #####
#####
#Set of Products
    m.G=cpr.RangeSet(1,products)

#Set of time periods
    m.P=cpr.RangeSet(1,T)

#Set of WIP patterns
    m.R=cpr.RangeSet(1,E)

    m.d=cpr.Param(d.keys(),initialize=d)
    m.h=cpr.Param(h.keys(), initialize=h)

```

```

m.b=cpr.Param(b.keys(), initialize=b)
m.w=cpr.Param(w.keys(), initialize=w)
m.x=cpr.Param(x.keys(), initialize=x)
m.d_factor=cpr.Param(d_factor.keys(), initialize=d_factor)

##      #initial inventory
##      m.initial_inventory=cpr.Param(m.G, initialize=0)

#initial backorder
m.initial_backorder=cpr.Param(m.G, initialize=0)

#Q and 0
m.Q=cpr.Param(m.G, m.R, initialize=Q)
m.O=cpr.Param(m.G, m.R, initialize=0)

#####
##### VARS #####
#####

m.X=cpr.Var(m.G, m.P, within=cpr.NonNegativeReals, name="X")
m.Y=cpr.Var(m.G, m.P, within=cpr.NonNegativeReals, name="Y")
m.W=cpr.Var(m.G, m.P, within=cpr.NonNegativeReals, name="W")
m.I=cpr.Var(m.G, m.P, within=cpr.NonNegativeReals, name="I")
m.B=cpr.Var(m.G, m.P, within=cpr.NonNegativeReals, name="B")
m.Z=cpr.Var(m.R, m.P, within=cpr.Binary, name="Z")
m.initial_inventory=cpr.Var(m.G, within=cpr.NonNegativeReals,\
name="initial_inventory")
m.initial_WIP=cpr.Var(m.G, within=cpr.NonNegativeReals,\
name="initial_WIP")

```

```

#####
##### Objective #####
#####

m.obj=cpr.Objective(expr=sum(m.x[g]*m.X[g,p] for g in m.G \
for p in m.P) +\
                    sum(m.w[g]*m.W[g,p] for g in m.G for p in m.P) +\
                    sum(m.h[g]*m.I[g,p] for g in m.G for p in m.P)+\
                    sum(m.h[g]*m.initial_inventory[g] for g in m.G)+\
                    sum(m.w[g]*m.initial_WIP[g] for g in m.G) +\
                    sum(m.b[g]*m.B[g,p] for g in m.G for p in m.P)\
                    , sense=cpr.minimize)

#####
##### Constraints #####
#####

#Constraint 1+2
def WIP_Balance_rule(m, g, p):
    if p==1:
        return m.W[g,p]==m.initial_WIP[g]+m.X[g,p]-m.Y[g,p]
    if p>1:
        return m.W[g,p]== m.W[g,p-1] + m.X[g,p] - m.Y[g,p]

m.WIP_Balance = cpr.Constraint(m.G, m.P, rule=WIP_Balance_rule,\
                               name='WIP_Balance')

#Constraint 3+4
def Inventory_Balance_rule(m, g, p):
    if p==1:
        return m.Y[g,p] + m.initial_inventory[g] - \
m.initial_backorder[g] - m.I[g,p] + m.B[g,p]\

```

```

== (m.d[g,p]*m.d_factor[g])
    if p>1:
        return m.Y[g,p] + m.I[g,p-1] - m.B[g,p-1] - \
m.I[g,p] + m.B[g,p] \
        == (m.d[g,p]*m.d_factor[g])

m.Inventory_Balance=cpr.Constraint(m.G, m.P, \
rule=Inventory_Balance_rule, name="Inventory_Balance")

#Constraint 5
def Capacity_1_rule(m, g, p):
    return sum(m.O[g,r]*m.Z[r,p] for r in m.R) == m.Y[g,p]

m.Capacity_1=cpr.Constraint(m.G, m.P, rule=Capacity_1_rule,\
                            name="Capacity_1")

#Constraints 6+7
def Capacity_2_rule(m, g, p):
    if p==1:
        return m.initial_WIP[g] == \
            sum(m.Q[g,r]*m.Z[r,p] for r in m.R)
    if p>1:
        return m.W[g,p-1] == sum(m.Q[g,r]*m.Z[r,p] for r in m.R)
m.Capacity_2=cpr.Constraint(m.G, m.P, rule=Capacity_2_rule,\
                            name="Capacity_2")

#Constraint 8
def Capacity_3_rule(m, p):
    return sum(m.Z[r,p] for r in m.R) == 1
m.Capacity_3=cpr.Constraint(m.P, rule=Capacity_3_rule,\
                            name="Capacity_3")

```

```

#####
##### Solve #####
#####

import time
start_time = time.time()
#m.write("ACF-1.lp") #Uncomment to write LP File
opt = SolverFactory('gurobi')
#opt.options['Heuristics']=0.9 \
#Use same structure for any parameter changes
results = opt.solve(m, tee=False)
#print("--- %s seconds ---" % (time.time() - start_time))
time = time.time() - start_time
return m.obj(), time

def main():
    import random

    from Closed_ISERC import MVA

    #####
    #####Network Data###
    #####

    #Number of products
    products=3

    #Number of stations
    num_stations = 11;

    #WIP
    N = [1,1,1];

```

```

#Total number of visits of each product
Z = [22,14,14];

#Number of machines in each station
c = [1,1,1,1,1,1,1,1,1,1,1,1];

#Processing times
mu_inv = [[80,80,80], [220,220,220], [45,45,45], [40,40,40], \
[25,25,25], [22,22,22], [20,20,20], [100,100,100], [50,50,50], \
[50,50,50], [70,70,70]];

#Number of visits of each product to each station
v = [[3,2,2], [1,1,1], [1,1,1], [6,4,0], [3,1,1], [2,2,2], [3,1,1], \
[1,0,0], [1,1,1], [1,1,1], [0,0,4]];

#Holding, backorder, WIP, and release costs
#h, b, w, x

h={1:15 , 2:15, 3:15}
b={1:20 , 2:20, 3:20}
w={1:7 , 2:7 , 3:7}
x={1:3 , 2:3 , 3:3}
d_factor={1:1 , 2:1 , 3:1}
T=10 #number of time periods
## T=50 #number of time periods

## #Random Demand 10 Period
d = {
    (1,1):8.21788942643641,
    (1,2):5.86897356542888,
    (1,3):0.260097167302294,

```



```
(1,4):6.9901471523404,  
(1,5):7.27675679301042,  
(1,6):3.6562409455957,  
(1,7):5.87353593952824,  
(1,8):8.87188928884992,  
(1,9):3.95345088433435,  
(1,10):9.04028693859194,  
(2,1):0.344361575605143,  
(2,2):4.16664791424385,  
(2,3):5.67985473388697,  
(2,4):2.1089864854594,  
(2,5):1.71509940753017,  
(2,6):2.52280364837306,  
(2,7):0.571095198164596,  
(2,8):1.17486771608755,  
(2,9):3.75474643382683,  
(2,10):1.59843166656426,  
(3,1):2.34577763820387,  
(3,2):0.427215637533297,  
(3,3):3.92127785742083,  
(3,4):1.5903351042033,  
(3,5):1.74753990745853,  
(3,6):4.18226193754764,  
(3,7):4.23653302661794,  
(3,8):0.870392462770855,  
(3,9):2.60546509957668,  
(3,10):0.268069197348271,  
};
```

```
#####  
####Basic Data####  
#####
```

```

num_hours = 8;
num_minutes = 60;
num_days = 7;

epsilon = 0.001; #For MVA convergence
##   num_iterations=4; # Iterative
##   step_size=20; # Iterative

##   #WIP for Step Size 2
##   I = [[0,0,0], [0,0,9.2246], [0,1.9909,2.5294], [0,4.1741,0], \
[3.8257,0,2.9326], [4.1145,2.5785,0], [12.3804,0,0]];

##   #WIP for Step Size 3
##   I = [[0,0,0], [0,0,9.2246], [0,1.3485,3.8214], [0,2.6941,1.5686], \
[0,4.1741,0], [2.362,0,4.0978], [2.401,1.5456,1.7492], [2.5057,3.185,0], \
[5.6899,0,1.9738], [6.1048,1.908,0], [12.3804,0,0]];

##   #WIP for Step Size 4
##   I = [[0,0,0], [0,0,9.2246], [0,1.0112,4.5352], [0,1.9909,2.5294], \
[0,3.0672,1.1453], [0,4.1741,0], [1.6752,0,4.7463], [1.7091,1.0989,2.7149], \
[1.7467,2.2575,1.2642], [1.8189,3.4348,0], [3.8257,0,2.9326], \
[3.9167,1.2415,1.3299], [4.1145,2.5785,0], [6.8222,0,1.4709], \
[7.3788,1.5279,0], [12.3804,0,0]];

##   #WIP for Step Size 5
##   I = [[0,0,0], [0,0,9.2246], [0,0.8072,5.0351], [0,1.5791,3.2308], \
[0,2.415,1.9457], [0,3.2598,0.9236], [0,4.1741,0], [1.3406,0,5.2353], \
[1.3484,0.8716,3.4455], [1.353,1.7376,2.0728], [1.3522,2.6127,0.9606], \
[1.4271,3.6048,0], [2.891,0,3.6144], [2.9889,0.9525,2.2097], \
[3.0295,1.9249,1.0283], [3.1564,2.9642,0], [4.9295,0,2.3442], \
[5.037,1.0642,1.107], [5.3653,2.232,0], [7.6971,0,1.2139], \

```

```

[8.1342,1.272,0],[12.3804,0,0]];

##      # WIP for Setp Size 6
##      I = [[0,0,0],[6.1048,1.908,0],[2.5057,3.185,0],\
[5.6899,0,1.9738],[2.362,0,4.0978],[0,2.6941,1.5686],\
[0,1.3485,3.8214],[12.3804,0,0],[0,4.1741,0],\
[0,0,9.2246],[1.0831,0,5.6239],[0,0.6544,5.3862],\
[1.076,0.7063,3.8598],[2.401,1.5456,1.7492],[3.8257,0,2.9326],\
[0,1.9856,2.5449],[4.1381,2.6039,0],[1.1571,3.6852,0],[8.34,0,1.0324],\
[0,3.3664,0.7616],[8.8283,1.0974,0],[1.1275,2.9137,0.8023],\
[1.1053,2.1518,1.6582],[1.1014,1.4175,2.6714],[2.4475,2.3424,0.8273],\
[2.38,0.7622,2.7975],[5.9093,0.9309,0.9396],[3.8846,0.817,1.8425]];

#Test
#N = [6.1048,1.908,0];
#C = MVA(epsilon, products, num_stations, N, Z, c, mu_inv, v);
#for i in range(len(C[0])):
    #C[0][i] = C[0][i] * 60;
#print "C is ", C[0];

Q, O, E = get_QO_fromMVA(products,epsilon, \
                        num_stations,num_hours,num_minutes,num_days, \
                        N,Z,c,mu_inv,v,I)

##      obj_Value = callGurobi(products,Q,O,E,h,b,w,x,T,d,d_factor);
##      print obj_Value;

#Repeat for random demands

#print "d[1,1] is ", d[1,1];
T = 10;

```

```

Arrival_rate = [0.2, 0.171428571, 0.181818182];

num_iterations = 100;

objValueSet = [];
timeSet = [];

for k in range(num_iterations):
    for j in range(T):
        total = 0;
        for i in range(products):
            d[i+1,j+1] = random.random();
            #print "d[" , i+1, ",", j+1, "]" is ", d[i+1,j+1];
            total = total + d[i+1,j+1];
        for i in range(products):
            d[i+1,j+1] = float(d[i+1,j+1]/total*Arrival_rate[i]\
*num_hours*num_days);
            #print "d[" , i+1, ",", j+1, "]" is ", d[i+1,j+1];

        obj_Value, time = callGurobi(products,Q,0,E,h,b,w,x,T,\
d,d_factor);
        objValueSet.append(obj_Value);
        timeSet.append(time);

print "Objective values are ", objValueSet;
print "Solution times are ", timeSet;

import numpy;
import math;
x = numpy.mean(objValueSet);
## sigma = numpy.std(objValueSet)/math.sqrt(num_iterations);
## LCL = x-2.58*sigma;

```

```

##    UCL = x+2.58*sigma;
##    percent_limit = 2.58*float(sigma/x)*100;
    d_2 = 1.128;
    R = findR(objValueSet);
##    sigma_E = float(numpy.mean(x)/d_2);
    sigma_E = float(numpy.mean(R)/d_2);
    LCL_E = x-1.96*float(sigma_E/math.sqrt(num_iterations));
    UCL_E = x+1.96*float(sigma_E/math.sqrt(num_iterations));
    percent_limit_E = 1.96*float(float(sigma_E/math.sqrt\
(num_iterations))/x)*100;

    y = numpy.mean(timeSet);
    R_T = findR_T(timeSet);
##    sigma_T = float(numpy.mean(y)/d_2);
    sigma_T = float(numpy.mean(R_T)/d_2);
    LCL_T = y-1.96*float(sigma_T/math.sqrt(num_iterations));
    UCL_T = y+1.96*float(sigma_T/math.sqrt(num_iterations));
    percent_limit_T = 1.96*float(float(sigma_T/math.sqrt\
(num_iterations))/y)*100;

    print "The Mean is ", x;
    print "The Estimated Standard Deviation of the Mean is ", sigma_E ;
    print "The LCL_E is ", LCL_E;
    print "The UCL_E is ", UCL_E;
    print "The Mean of the run time is ", y;
    print "The Estimated Standard Deviation of the Mean is ", sigma_T ;
    print "The LCL_E is ", LCL_T;
    print "The UCL_E is ", UCL_T;

if __name__ == '__main__':
    main()

```

B.1.2 Calling Function for Closed_ISERC

```
# Mean value analysis algorithm

def MVA(epsilon, num_products, num_stations, N, Z, c, mu_inv, v):
    C = [];
    s = 0;
    L = [];
    Looping = True;
    while Looping:
        #Step 1
        for j in range(num_stations):
            L_temp = [];
            for p in range(num_products):
                L_temp.append(float(N[p])/float(Z[p]));
            L.append(L_temp);
        #print "L is", L;

        #Step 2
        W = [];
        for j in range(num_stations):
            W_temp = [];
            for p in range(num_products):
                temp1 = float(mu_inv[j][p])/float(c[j])*\
(1+float(N[p]-1)/float(N[p]+epsilon))*L[j][p])
                temp2 = 0;
                for r in range(num_products):
                    if (r != p) & (mu_inv[j][p] !=0.0):
                        temp2 = temp2 + float(L[j][r])*\
float(mu_inv[j][r])/float(c[j]);
                #print "j= ",j , "p = ", p, "Temp2 is: ", temp2;
```

```

        if c[j] > 1:
            temp3 = float(c[j]-1)/float(c[j])*\  

float(mu_inv[j][p]);
        else:
            temp3 = 0;
            W_temp.append(temp1+temp2+temp3);
        W.append(W_temp);
    #print "W is", W;

#Step 3
X=[];
for p in range(num_products):
    x_temp_denominator = 0;
    for j in range(num_stations):
        x_temp_denominator = x_temp_denominator\  

+ float(v[j][p])*W[j][p];
        x_temp = float(N[p])/float(x_temp_denominator);
        X.append(x_temp);
    #print "X is", X;

#Step 4-1
#print "Computing L_new"
L_new = [];
for j in range(num_stations):
    L_temp = [];
    for p in range(num_products):
        L_temp.append(float(X[p])*float(v[j][p])*\  

float(W[j][p]));
        #print "X[p] is ", X[p];
        #print "v[j][p] is ", v[j][p];
        #print "W[j][p] is ", W[j][p];
        #print "L_temp is", L_temp;

```

```
        L_new.append(L_temp);
    #print "L_new is ", L_new;

    #Step 4-2
    Looping2 = False;
    for j in range(num_stations):
        for p in range(num_products):
            if L_new[j][p] > 0.0:
                if (L_new[j][p] - L[j][p])/L_new[j][p] > epsilon:
                    Looping2 = True;
    if Looping2:
        L = L_new;
    Looping = Looping2;
    #print "Done, L (final) is ", L;
    #print "Done, X (final) is ", X;

    #print len(X);
    #print X[0];

    C_temp = [];
    for p in range(num_products):
        C_temp.append(X[p]);
    C.append(C_temp);

    return C;
```


B.2 MPL Code

```
{ACF-I.mpl}
```

```
TITLE
```

```
ACFIProblem;
```

```
INDEX
```

```
g := 1..3; !Set of products
```

```
p := 1..10; !Set of time 10 periods
```

```
!p := 1..20; !Set of time 20 periods
```

```
!p := 1..30; !Set of time 30 periods
```

```
r := 1..11; !Set of WIP patterns Step Size 3
```

```
!r := 1..15; !Set of WIP patterns Step Size 4
```

```
!r := 1..21; !Set of WIP patterns Step Size 5
```

```
!r := 1..28; !Set of WIP patterns Step Size 6
```

```
!r := 1..55; !Set of WIP patterns Step Size 9
```

```
c := 1..27; !Set of Cubes
```

```
DATA
```

```
f := 2.5;
```

```
e := 100;
```

```
initial_backorder[g] := 0;
```

```
{
```

```
initial_inventory[g] := 0;
```

```
initial_WIP[g] := 0;
```

```
}
```

!Random 1

```
d[g,p] := (
8.217889426 5.868973565 0.260097167 6.990147152 7.276756793 \
3.656240946 5.87353594 8.871889289 3.953450884 9.040286939
0.344361576 4.166647914 5.679854734 2.108986485 1.715099408\
2.522803648 0.571095198 1.174867716 3.754746434 1.598431667
2.345777638 0.427215638 3.921277857 1.590335104 1.747539907\
4.182261938 4.236533027 0.870392463 2.6054651 0.268069197
);
```

```
h[g] := (15 15 15);
```

```
b[g] := (20 20 20);
```

```
w[g] := (7 7 7);
```

```
x[g] := (3 3 3);
```

```
d_factor[g] := (1 1 1);
```

!Step Size 3

```
Q[g,r] := (
0 0 0 0 0 2.362 2.401 2.5057 5.6899 6.1048 12.3804
0 0 1.3485 2.6941 4.1741 0 1.5456 3.185 0 1.908 0
0 9.2246 3.8214 1.5686 0 4.0978 1.7492 0 1.9738 0 0
);
```

```
O[g,r] := (
```

```

0 0 0 0 0 4.11863478 4.127180906 4.10863817 8.109192977 \
8.085416059 11.88458619
0 0 3.632653827 7.326128136 10.97620264 0 3.612623464 \
7.171334603 0 3.499280824 0
0 11.10446571 7.791048149 3.880047713 0 7.641140431 \
3.86219321 0 3.798166813 0 0
);

```

!Step Size 4

```

{
Q[g,r] := (
0 0 0 0 0 1.6752 1.7091 1.75 1.8189 3.8257 3.9167 \
4.1145 6.8222 7.3788 12.3804
0 1.0112 1.9909 3.0672 4.1741 0 1.0989 2.2575 3.4348\
0 1.2415 2.5785 0 1.5279 0
9.2246 4.5352 2.5294 1.1453 0 4.7463 2.7149 1.2642 0\
2.9326 1.3299 0 1.4709 0 0
);
O[g,r] := (
0.00 0.00 0.00 0.00 0.00 3.06 3.10 3.10 3.12 6.13 6.138829775\
6.125359053 9.09 9.081878684 11.88458619
0.00 2.73 5.45 8.27 10.98 0.00 2.70 5.44 8.07 0.00 2.663044471\
5.292651731 0.00 2.608895476 0
11.10 8.65 5.82 2.91 0.00 8.55 5.78 2.95 0.00 5.71 2.849764966\
0 2.80 0 0
);
}

```

!Step Size 5

```

{
Q[g,r] := (
0 0 0 0 0 0 1.3406 1.35 1.353 1.3522 1.4271 2.891 2.9889 \
3.0295 3.1564 4.9295 5.037 5.3653 7.6971 8.1342 12.3804

```

```

0 0.8072 1.5791 2.415 3.2598 4.1741 0 0.8716 1.7376 2.6127\
3.6048 0 0.9525 1.9249 2.9642 0 1.0642 2.232 0 1.272 0
9.2246 5.0351 3.2308 1.9457 0.9236 0 5.2353 3.4455 2.0728\
0.9606 0 3.6144 2.2097 1.0283 0 2.3442 1.107 0 1.2139 0 0
);
O[g,r] := (
0.00 0.00 0.00 0 0 0 2.493731817 2.49095969 2.492893794\
2.46588803 2.503879307 4.892035192 4.969737287 4.951815962\
4.960562253 7.358347594 7.331808583 7.361621856 9.687407662\
9.627292473 11.88458619
0.00 2.18 4.31 6.576133091 8.767073999 10.97620264 0\
2.167597452 4.337122046 6.495258344 8.661713321 0\
2.151009694 4.299682442 6.404593823 0 2.126177147\
4.228770253 0 2.091996766 0
11.10 9.15 7.00 4.678920627 2.37883643 0 9.073367738\
6.972824107 4.670788975 2.339462397 0 6.900416591\
4.620703515 2.311628993 0 4.539541896 2.288530505\
0 2.252448442 0 0
);
}
!Step Size 6
{
Q[g,r] := (
0 0 0 0 0 0 0 1.08 1.076 1.1014 1.1053 1.1275 1.1571 2.362\
2.38 2.401 2.4475 2.5057 3.8257 3.8846 3.9703 4.1381 5.6899\
5.9093 6.1048 8.34 8.8283 12.3804
0 0.6544 1.3485 1.9856 2.6941 3.3664 4.1741 0 0.7063 1.4175\
2.1518 2.9137 3.6852 0 0.7622 1.5456 2.3424 3.185 0 0.817 1.6871\
2.6039 0 0.9309 1.908 0 1.0974 0
9.2246 5.3862 3.8214 2.5449 1.5686 0.7616 0 5.6239 3.8598 2.6714\
1.6582 0.8023 0 4.0978 2.7975 1.7492 0.8273 0 2.9326 1.8425 0.8742\
0 1.9738 0.9396 0 1.0324 0 0

```

```

);
O[g,r] := (
0 0 0 0 0 0 0 2.048212036 2.043602508 2.073488793 2.063473952\
2.064125278 2.071700316 4.11863478 4.136607263 4.127180906\
4.13022617 4.10863817 6.126605385 6.139263069 6.125210632\
6.131847947 8.109192977 8.140041924 8.085416059 10.07953496\
10.04419349 11.88458619
0 1.7785149 3.632653827 5.430542432 7.326128136 9.081037735\
10.97620264 0 1.805420757 3.604938301 5.451391151\
7.265682395 9.034087951 0 1.793343745 3.612623464\
5.400906975 7.171334603 0 1.762207569 3.568915223\
5.317357387 0 1.764455337 3.499280824 0 1.736384852 0
11.10446571 9.47165867 7.791048149 5.849408673 3.880047713\
1.990650222 0 9.44602769 7.655142035 5.823152931 3.888893174\
1.978376099 0 7.641140431 5.765074325 3.86219321 1.928450254\
0 5.706842736 3.831031316 1.912240819 0 3.798166813\
1.891417781 0 1.88348118 0 0
);

```

```

Avg_d[g] := sum(p: d[g,p]);

```

```

DECISION VARIABLES

```

```

initial_inventory[g];

```

```

initial_WIP[g];

```

```

X[g,p];

```

```

Y[g,p];

```

```

W[g,p];

```

I[g,p];

B[g,p];

pp[g,p];

nn[g,p];

BINARY VARIABLES

Z[r,p];

MODEL

MIN o = sum(g,p: x[g]*X[g,p]) +
 sum(g,p: w[g]*W[g,p]) +
 sum(g,p: h[g]*I[g,p]) +
 sum(g,p: b[g]*B[g,p]) +
 sum(g: w[g]*initial_WIP[g]) +
 sum(g: h[g]*initial_inventory[g])
 + e*sum(g,p: pp[g,p] + nn[g,p]);

MACRO

ReleaseCost = sum(g,p: x[g]*X[g,p]);
 WIPCost = sum(g,p: w[g]*W[g,p]);
 InventoryCost = sum(g,p: h[g]*I[g,p]);
 BackorderCost = sum(g,p: b[g]*B[g,p]);
 InitialWIPCost = sum(g: w[g]*initial_WIP[g]);
 InitialInventoryCost = sum(g: h[g]*initial_inventory[g]);
 PenaltyCost = e*sum(g,p: pp[g,p] + nn[g,p]);
 ActualCost = ReleaseCost+WIPCost+InventoryCost+BackorderCost

+InitialWIPCost+InitialInventoryCost;

AverageDemand1 = Avg_d[1]/30;

SUBJECT TO

WIP_Balance_1[g,p=1]: $W[g,p] = \text{initial_WIP}[g] + X[g,p] - Y[g,p];$

WIP_Balance_2[g,p>1]: $W[g,p] = W[g,p-1] + X[g,p] - Y[g,p];$

Inventory_Balance_1[g,p=1]: $Y[g,p] + \text{initial_inventory}[g] - \backslash$
 $\text{initial_backorder}[g] - I[g,p] + B[g,p]$
 $= d[g,p] * d_factor[g];$

Inventory_Balance_2[g,p>1]: $Y[g,p] + I[g,p-1] - B[g,p-1] - \backslash$
 $I[g,p] + B[g,p]$
 $= d[g,p] * d_factor[g];$

!Fixed Releases

Capacity_1[g,p]: $Y[g,p] = \text{sum}(r: O[g,r]*Z[r,p]);$

Capacity_21[g,p=1]: $\text{initial_WIP}[g] = \text{sum}(r: Q[g,r]*Z[r,p]);$

Capacity_22[g,p>1]: $W[g,p-1] = \text{sum}(r: Q[g,r]*Z[r,p]);$

Capacity_3[p]: $\text{sum}(r: Z[r,p]) = 1;$

!Fix solution to f fraction of demand with penalty cost e

P[g,p]: $Y[g,p] = d[g,p] + pp[g,p] - nn[g,p];$

Limit_pp[g,p]: $pp[g,p] \leq f * Avg_d[g] / 30;$

Limit_nn[g,p]: $nn[g,p] \leq f * Avg_d[g] / 30;$

END

Appendix C

Chapter 5

C.1 Python Code for Cubic Approximation

```
#-----  
# Name:          CA-1  
# Purpose:       Cubic Approximation Model for thesis  
#  
# Author:  
#  
# Created:       16/05/2016  
#  
# Copyright:     c)  
#-----  
  
def get_eight_points(num_products, Cube, Cube_Step):  
  
    Cube_Points = [];  
  
    Cube_Points.append(Cube); #Point 1  
    Cube_Points.append([Cube[0]+Cube_Step[0],Cube[1], Cube[2]]); #Point 2  
    Cube_Points.append([Cube[0]+Cube_Step[0],Cube[1]+Cube_Step[1],\  
Cube[2]]); #Point 3  
    Cube_Points.append([Cube[0],Cube[1]+Cube_Step[1],Cube[2]]); #Point 4  
    Cube_Points.append([Cube[0],Cube[1],Cube[2]+Cube_Step[2]]); #Point 5  
    Cube_Points.append([Cube[0]+Cube_Step[0],Cube[1], Cube[2]+\  
Cube_Step[2]]); #Point 6  
    Cube_Points.append([Cube[0]+Cube_Step[0],Cube[1]+Cube_Step[1],\  
Cube[2]+Cube_Step[2]]); #Point 7
```

```

Cube_Points.append([Cube[0],Cube[1]+Cube_Step[1],Cube[2]+\
Cube_Step[2]]); #Point 8

#print "Cube_Points is: ", Cube_Points;
return Cube_Points;

def get_Slope(num_products, T, Cube_Step):
    Temp = [];
    #print "T is ", T;
    #print "T[0] is ", T[0];
    #print "T[0][0] is ", T[0][0];
    #rint "T[0][0][0] is ", T[0][0][0];
    Temp.append(float((T[1][0][0]-T[0][0][0]+
        T[2][0][0]-T[3][0][0]+
        T[5][0][0]-T[4][0][0]+
        T[6][0][0]-T[7][0][0])/4/Cube_Step[0])); \
#Product 1 slope

    Temp.append(float((T[3][0][1]-T[0][0][1]+
        T[2][0][1]-T[1][0][1]+
        T[7][0][1]-T[4][0][1]+
        T[6][0][1]-T[5][0][1])/4/Cube_Step[1])); \
#Product 2 slope

    Temp.append(float((T[4][0][2]-T[0][0][2]+
        T[5][0][2]-T[1][0][2]+
        T[6][0][2]-T[2][0][2]+
        T[7][0][2]-T[3][0][2])/4/Cube_Step[2])); \
#Product 1 slope

```

```

return Temp;

def get_Q0_fromMVA(num_products, epsilon, \
                  num_stations,num_hours,num_minutes,num_days, \
                  N,Z,c,mu_inv,v,I):
    from Closed_ISERC import MVA

    #Call mean value analysis function within a loop to find \
    throughput estimates for different combinations of input products.

    #print "Length of I is: ",len(I);
    #print ("I is :", I);

    #Throughput for each inventory combination
    T = [];
    for i in range(len(I)):
        T.append(MVA(epsilon, num_products, num_stations, \
                    I[i], Z, c, mu_inv, v));
    Q={}
    O={}

    ## #Open a file
    ## filep = open("data.txt", "w");
    ## filep.write("param Q := \n");
    ## for g in range(num_products):
    ##     for r in range(len(I)):
    ##         Q[g+1,r+1]=I[r][g] #Create Q
    ##         w_string = str(g+1);
    ##         w_string += " ";
    ##         w_string += str(r+1);
    ##         w_string += " ";
    ##         w_string += str(I[r][g]);
    ##         filep.write("%s \n" %w_string );

```

```

##     filep.write(";\n");

##     filep.write("param 0 := \n");
##     for g in range(num_products):
##         for r in range(len(T)):
##             #print "g: %s, r: %s"%(g+1,r+1)
##             O[g+1,r+1]=float(T[r][0][g]*num_hours*num_minutes*num_days)
##             w_string = str(g+1);
##             w_string += " ";
##             w_string += str(r+1);
##             w_string += " ";
##             w_string += str(T[r][0][g]*num_hours*num_minutes*num_days);
##             filep.write("%s \n" %w_string );
##     filep.write(";\n");
##     filep.close();

    for g in range(num_products):
        for r in range(len(T)):
            T[r][0][g] = float(T[r][0][g]*num_hours*\
num_minutes*num_days);
            #print "T from MVA is: ", T;

    return Q, O, T, len(I)

def main():
    from Closed_ISERC import MVA
    #####
    #####Network Data###
    #####

    StepSize = 3; #Step size for cubic approximation

```

```
#Number of products
products=3

#Number of stations
num_stations = 11;

#WIP
N = [1,1,1];

Max_WIP = [12.38,4.17,9.22];

#Total number of visits of each product
Z = [22,14,14];

#Number of machines in each station
c = [1,1,1,1,1,1,1,1,1,1,1];

#Processing times
mu_inv = [[80,80,80], [220,220,220], [45,45,45], [40,40,40], \
[25,25,25], [22,22,22], [20,20,20], [100,100,100], [50,50,50], \
[50,50,50], [70,70,70]];

#Number of visits of each product to each station
v = [[3,2,2], [1,1,1], [1,1,1], [6,4,0], [3,1,1], [2,2,2], [3,1,1], \
[1,0,0], [1,1,1], [1,1,1], [0,0,4]];

num_hours = 8;
num_minutes = 60;
num_days = 7;

epsilon = 0.001; #For MVA convergence
```

```

#Obtain step values for each product
CubeStep = [];

for i in Max_WIP:
    CubeStep.append(float(i/(StepSize)));

#print "CubeStep is ", CubeStep;
#print "Number of products is ", Num_Products;

#Define all the points in cubic approximation, \
grouped by product

Point_Array_L = []; #Define lower points array

for i in range(products):
    Temp = [];
    for j in range(StepSize):
        Temp.append(j*CubeStep[i]);
        #print "Temp is ", Temp;
    Point_Array_L.append(Temp);

Point_Array_U = []; #Define upper points array

for i in range(products):
    Temp_U = [];
    for j in range(StepSize):
        Temp_U.append((j+1)*CubeStep[i]);
        #print "Temp is ", Temp;
    Point_Array_U.append(Temp_U);

#print "Point_Array_L is ", Point_Array_L;
#print "Point_Array_U is ", Point_Array_U;

```

```

#Code below works only for three products!

import itertools

Cube_List_L = [];
for combination in itertools.product(Point_Array_L[0],\
Point_Array_L[1], Point_Array_L[2]):
    Cube_List_L.append(combination);

Cube_List_U = [];
for combination in itertools.product(Point_Array_U[0],\
Point_Array_U[1], Point_Array_U[2]):
    Cube_List_U.append(combination);

print "Number of cubes is \n ", len(Cube_List_L);
print "Lower Cube_List is \n", Cube_List_L;
#print "Number of cubes is ", len(Cube_List_U);
#print "Upper Cube_List is ", Cube_List_U;

#Get throughputs

Q, O, T, E = get_QO_fromMVA(products,epsilon, \
                            num_stations,num_hours,num_minutes,num_days, \
                            N,Z,c,mu_inv,v,Cube_List_L);

#print "Q is: ", Q;
#print "O is: ", O;
#print "T is: \n", T;
#print "E is: ", E;

Th = T;

```

```

#Get slopes
Slope = [];
for r in range(len(Cube_List_L)):
    Temp1 = get_eight_points(products, Cube_List_L[r], CubeStep);
    Q, O, T, E = get_QO_fromMVA(products,epsilon, \
        num_stations,num_hours,num_minutes,num_days, \
        N,Z,c,mu_inv,v,Temp1);
    #print "T is: ", T;
    Temp2 = get_Slope(products, T, CubeStep);
    Slope.append(Temp2);
print "Slope is ", Slope;

##      Q, O, T, E = get_QO_fromMVA(products,epsilon, \
##          num_stations,num_hours,num_minutes,num_days, \
##          N,Z,c,mu_inv,v,[(6.19, 2.085, 4.61)]);
##      print "T is: ", T;

##      print "Cube_Point_List is: \n ", Cube_Point_List;
##      print
##

#MPL Output

print "LB[g,c] = ("
for g in range(products):
    for r in range(len(Cube_List_L)):
        print Cube_List_L[r][g],;
    print;
print ");"

print "UB[g,c] = ("
for g in range(products):

```



```
        for r in range(len(Cube_List_U)):
            print Cube_List_U[r][g],;
        print;
print ");"

print "TH[g,c] = ("
for g in range(products):
    for r in range(len(Th)):
        print Th[r][0][g],;
    print;
print ");"

print "S[g,c] = ("
for g in range(products):
    for r in range(len(Slope)):
        print Slope[r][g],;
    print;
print ");"
if __name__ == '__main__':
    main()
```

C.2 MPL Code

```
{ACF-I.mpl}
```

```
TITLE
```

```
ACFIProblem;
```

```
INDEX
```

```
g := 1..3; !Set of products
```

```
p := 1..10; !Set of time 10 periods
```

```
!p := 1..20; !Set of time 20 periods
```

```
!p := 1..30; !Set of time 30 periods
```

```
r := 1..11; !Set of WIP patterns Step Size 3
```

```
!r := 1..15; !Set of WIP patterns Step Size 4
```

```
!r := 1..21; !Set of WIP patterns Step Size 5
```

```
!r := 1..28; !Set of WIP patterns Step Size 6
```

```
!r := 1..55; !Set of WIP patterns Step Size 9
```

```
c := 1..27; !Set of Cubes
```

```
DATA
```

```
f := 1.5;
```

```
e := 125;
```

```
initial_backorder[g] := 0;
```

```
{
```

```
initial_inventory[g] := 0;
```

```
initial_WIP[g] := 0;
```

```
}
```

!Random 1

```
d[g,p] := (
8.217889426 5.868973565 0.260097167 6.990147152 7.276756793 \
3.656240946 5.87353594 8.871889289 3.953450884 9.040286939
0.344361576 4.166647914 5.679854734 2.108986485 1.715099408\
2.522803648 0.571095198 1.174867716 3.754746434 1.598431667
2.345777638 0.427215638 3.921277857 1.590335104 1.747539907\
4.182261938 4.236533027 0.870392463 2.6054651 0.268069197
);
```

```
h[g] := (15 15 15);
```

```
b[g] := (20 20 20);
```

```
w[g] := (7 7 7);
```

```
x[g] := (3 3 3);
```

```
d_factor[g] := (1 1 1);
```

```
LB[g,c] = (
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 4.12666666667 4.12666666667\
4.12666666667 4.12666666667 4.12666666667 4.12666666667\
4.12666666667 4.12666666667 4.12666666667 8.25333333333\
8.25333333333 8.25333333333 8.25333333333 8.25333333333\
8.25333333333 8.25333333333 8.25333333333 8.25333333333
0.0 0.0 0.0 1.39 1.39 1.39 2.78 2.78 2.78 0.0 0.0 0.0 1.39 1.39 1.39\
2.78 2.78 2.78 0.0 0.0 0.0 1.39 1.39 1.39 2.78 2.78 2.78
```

```

0.0 3.07333333333 6.14666666667 0.0 3.07333333333\
 6.14666666667 0.0 3.07333333333 6.14666666667 0.0\
 3.07333333333 6.14666666667 0.0 3.07333333333 6.14666666667\
 0.0 3.07333333333 6.14666666667 0.0 3.07333333333 6.14666666667\
 0.0 3.07333333333 6.14666666667 0.0 3.07333333333 6.14666666667
);
UB[g,c] = (
4.12666666667 4.12666666667 4.12666666667 4.12666666667\
 4.12666666667 4.12666666667 4.12666666667 4.12666666667\
 4.12666666667 8.25333333333 8.25333333333 8.25333333333\
 8.25333333333 8.25333333333 8.25333333333 8.25333333333\
 8.25333333333 8.25333333333 12.38 12.38 12.38 12.38 12.38\
 12.38 12.38 12.38 12.38
1.39 1.39 1.39 2.78 2.78 2.78 4.17 4.17 4.17 1.39 1.39 1.39 2.78\
 2.78 2.78 4.17 4.17 4.17 1.39 1.39 1.39 2.78 2.78 2.78 4.17 4.17\
 4.17
3.07333333333 6.14666666667 9.22 3.07333333333 6.14666666667\
 9.22 3.07333333333 6.14666666667 9.22 3.07333333333 6.14666666667\
 9.22 3.07333333333 6.14666666667 9.22 3.07333333333 6.14666666667\
 9.22 3.07333333333 6.14666666667 9.22 3.07333333333 6.14666666667\
 9.22 3.07333333333 6.14666666667 9.22
);
TH[g,c] = (
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 8.11876313895 6.33676900126\
 5.219368117 6.92397533432 5.50939067247 4.57818569287 6.01953427594\
 4.87940958285 4.0989970098 10.7556693553 8.85334760244 7.51706727343\
 9.58943259694 8.01954964484 6.88067971906 8.65065718533\
 7.33658423272 6.35523835726
0.0 0.0 0.0 5.62620360739 3.97833681269 3.12515090913 8.94516267044\
 6.41667545162 5.01638227643 0.0 0.0 0.0 3.28112594488 2.44557387312\
 1.94089934114 5.57310965718 4.23667252573 3.40702516506 0.0 0.0 0.0\
 2.23997768958 1.75997131283 1.44302848667 3.96989404318 3.16107796509\

```

```

2.62233006311
0.0 7.90148203149 10.2453866858 0.0 6.93384252079 9.42866197923 0.0\
6.07232350893 8.58572169317 0.0 5.74247150955 8.20892758362 0.0\
5.07966905838 7.46927908254 0.0 4.52733489449 6.82232688669 0.0\
4.36914246153 6.59614502624 0.0 3.94169784817 6.06400153227 0.0\
3.5788754824 5.60138990363
);
S[g,c] = (
1.62897202021 1.31121042914 1.10487569734 1.41350827134 1.15504743284\
0.979758743813 1.24793399403 1.03498870102 0.885423907895\
0.625753294781 0.583214706553 0.530327260399 0.621804147471\
0.577104341431 0.527936623876 0.608002066531 0.56249796122\
0.515942164853 0.313621516371 0.336448185152 0.336229922833\
0.338660043689 0.34933522169 0.344128697432 0.352340205471\
0.354796594985 0.345977669269
2.75741730901 2.06653973671 1.68030607039 1.76985253002\
1.36453138179 1.10307976856 1.1983007948 0.98072590143\
0.821469570774 1.74939726986 1.36501313197 1.11992983633\
1.29750096597 1.04993393979 0.882611365272 0.992600472019\
0.834918259066 0.722461097639 1.27036204743 1.03223870711\
0.868408252762 1.01786269255 0.847936108527 0.728641167343\
0.829728838189 0.708733281386 0.622591815095
2.08710399568 0.788621763369 0.377962047707 1.83946610487\
0.788461469011 0.419681191305 1.62774593343 0.77000774716\
0.444309647346 1.55637046185 0.748810114997 0.441422420715\
1.39324110223 0.718228317925 0.447521442552 1.25604318668\
0.686741779723 0.44729136738 1.22290028298 0.666403104931\
0.435632223105 1.11390400441 0.634844982837 0.429882317347\
1.02006860906 0.605567532695 0.422536953285
);

Avg_d[g] := sum(p: d[g,p]);

```

DECISION VARIABLES

initial_inventory[g];

initial_WIP[g];

X[g,p];

Y[g,p];

W[g,p];

I[g,p];

B[g,p];

pp[g,p];

nn[g,p];

BINARY VARIABLES

Z[c,p];

MODEL

MIN o = sum(g,p: x[g]*X[g,p]) +
sum(g,p: w[g]*W[g,p]) +
sum(g,p: h[g]*I[g,p]) +
sum(g,p: b[g]*B[g,p]) +
sum(g: w[g]*initial_WIP[g]) +

```
sum(g: h[g]*initial_inventory[g])
+ e*sum(g,p: pp[g,p] + nn[g,p]);
```

MACRO

```
ReleaseCost = sum(g,p: x[g]*X[g,p]);
WIPCost = sum(g,p: w[g]*W[g,p]);
InventoryCost = sum(g,p: h[g]*I[g,p]);
BackorderCost = sum(g,p: b[g]*B[g,p]);
InitialWIPCost = sum(g: w[g]*initial_WIP[g]);
InitialInventoryCost = sum(g: h[g]*initial_inventory[g]);
PenaltyCost = e*sum(g,p: pp[g,p] + nn[g,p]);
ActualCost = ReleaseCost+WIPCost+InventoryCost+BackorderCost
+InitialWIPCost+InitialInventoryCost;
AverageDemand1 = Avg_d[1]/30;
```

SUBJECT TO

```
WIP_Balance_1[g,p=1]: W[g,p] = initial_WIP[g] + X[g,p] - Y[g,p];
```

```
WIP_Balance_2[g,p>1]: W[g,p] = W[g,p-1] + X[g,p] - Y[g,p];
```

```
Inventory_Balance_1[g,p=1]: Y[g,p] + initial_inventory[g] -\
initial_backorder[g] - I[g,p] + B[g,p]
= d[g,p] * d_factor[g];
```

```
Inventory_Balance_2[g,p>1]: Y[g,p] + I[g,p-1] - B[g,p-1] -\
I[g,p] + B[g,p]
= d[g,p] * d_factor[g];
```

!Cubic Approximation

Cubic_Approximation_11[g,c,p=1]: $Y[g,p] \leq TH[g,c] + initial_WIP[g]*S[g,c] - LB[g,c]*S[g,c] + 1000 - Z[c,p]*1000;$

Cubic_Approximation_12[g,c,p>1]: $Y[g,p] \leq TH[g,c] + W[g,p-1]*S[g,c] - LB[g,c]*S[g,c] + 1000 - Z[c,p]*1000;$

Cubic_Approximation_211[g,c,p=1]: $LB[g,c]*Z[c,p=1] \leq initial_WIP[g];$

Cubic_Approximation_212[g,c,p=1]: $initial_WIP[g] \leq UB[g,c]*Z[c,p=1] + 1000 - Z[c,p]*1000;$

Cubic_Approximation_221[g,c,p>1]: $LB[g,c]*Z[c,p] \leq W[g,p-1];$

Cubic_Approximation_222[g,c,p>1]: $W[g,p-1] \leq UB[g,c]*Z[c,p] + 1000 - Z[c,p]*1000;$

Cubic_Approximation_3[p]: $sum(c: Z[c,p]) = 1;$

!Fix solution to f fraction of demand with penalty cost e

P[g,p]: $Y[g,p] = d[g,p] + pp[g,p] - nn[g,p];$

Limit_pp[g,p]: $pp[g,p] \leq f*Avg_d[g]/30;$

Limit_nn[g,p]: $nn[g,p] \leq f*Avg_d[g]/30;$

END

Appendix D

Chapter6

D.1 FPR Grid for the Open Queuing Network

Table D.1: All release combinations, simulation WIP and throughput

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	30	10	10	3.66	0.77	1.01	29.99	10.00	9.96
2	30	10	15	3.79	0.81	1.74	29.96	10.04	15.04
3	30	10	20	3.93	0.84	2.76	30.05	9.97	20.01
4	30	10	25	4.08	0.88	4.42	30.00	10.00	24.98
5	30	15	10	3.86	1.23	1.04	29.98	15.04	9.99
6	30	15	15	3.99	1.29	1.78	29.94	15.01	14.93
7	30	15	20	4.15	1.36	2.86	29.98	15.04	20.04
8	30	15	25	4.31	1.41	4.51	30.08	15.00	25.01
9	30	20	10	4.08	1.76	1.08	29.93	20.02	9.99
10	30	20	15	4.23	1.84	1.84	29.98	19.97	14.96
11	30	20	20	4.42	1.93	2.92	30.07	19.99	20.02
12	30	20	25	4.61	2.03	4.73	30.07	20.01	25.00
13	30	25	10	4.34	2.36	1.12	29.99	25.07	9.97
14	30	25	15	4.53	2.47	1.93	30.03	24.98	15.06
15	30	25	20	4.69	2.60	3.03	29.96	25.02	19.98
16	30	25	25	4.90	2.74	4.79	30.01	25.07	24.95
17	35	10	10	4.73	0.84	1.05	35.05	9.94	9.95
18	35	10	15	4.91	0.88	1.81	35.04	9.94	14.99
19	35	10	20	5.06	0.92	2.86	35.03	10.03	20.03

Continued on Next Page...

Table D.1 – Continued

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
20	35	10	25	5.26	0.96	4.56	35.08	10.00	25.05
21	35	15	10	5.00	1.36	1.10	34.97	15.01	10.05
22	35	15	15	5.20	1.42	1.86	35.10	15.05	14.97
23	35	15	20	5.34	1.48	2.98	34.89	15.04	20.10
24	35	15	25	5.58	1.55	4.67	35.02	15.01	24.93
25	35	20	10	5.32	1.94	1.13	35.04	19.96	10.01
26	35	20	15	5.51	2.04	1.93	35.00	20.01	14.99
27	35	20	20	5.68	2.12	3.04	34.94	20.02	19.93
28	35	20	25	5.96	2.23	4.86	35.04	19.98	25.00
29	35	25	10	5.66	2.62	1.18	34.90	25.01	10.04
30	35	25	15	5.91	2.75	2.01	35.01	24.97	15.02
31	35	25	20	6.11	2.88	3.14	34.94	25.02	19.95
32	35	25	25	6.36	3.02	4.96	34.96	25.05	24.97
33	40	10	10	5.99	0.93	1.11	40.01	10.03	10.04
34	40	10	15	6.22	0.97	1.90	39.98	10.07	15.04
35	40	10	20	6.40	1.00	2.96	39.97	9.99	19.97
36	40	10	25	6.63	1.05	4.70	39.98	9.97	24.89
37	40	15	10	6.39	1.50	1.15	39.95	15.00	10.04
38	40	15	15	6.61	1.57	1.94	39.97	15.03	14.96
39	40	15	20	6.84	1.63	3.05	40.04	14.96	19.96
40	40	15	25	7.11	1.70	4.83	40.00	14.93	24.98
41	40	20	10	6.81	2.16	1.19	39.94	19.99	10.06
42	40	20	15	7.09	2.26	2.02	40.00	19.99	15.00
43	40	20	20	7.32	2.35	3.21	40.00	19.97	20.05
44	40	20	25	7.62	2.47	5.05	40.00	19.98	25.07

Continued on Next Page...

Table D.1 – Continued

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
45	40	25	10	7.42	2.98	1.23	40.02	25.09	9.96
46	40	25	15	7.67	3.08	2.09	39.98	24.95	14.94
47	40	25	20	7.98	3.24	3.29	40.05	25.08	19.99
48	40	25	25	8.27	3.39	5.19	39.96	24.94	25.01
49	45	10	10	7.51	1.03	1.15	44.95	10.00	10.01
50	45	10	15	7.88	1.07	1.97	45.07	9.97	15.02
51	45	10	20	8.11	1.12	3.09	44.99	9.98	20.02
52	45	10	25	8.40	1.18	4.84	44.92	10.09	24.89
53	45	15	10	8.05	1.67	1.19	44.92	15.01	9.99
54	45	15	15	8.36	1.74	2.03	44.88	14.97	14.93
55	45	15	20	8.78	1.85	3.21	44.99	15.06	19.99
56	45	15	25	9.12	1.93	5.04	45.05	15.08	24.94
57	45	20	10	8.85	2.44	1.25	45.03	19.94	10.07
58	45	20	15	9.17	2.57	2.11	45.10	20.01	14.99
59	45	20	20	9.48	2.67	3.33	45.06	19.93	20.05
60	45	20	25	9.91	2.81	5.20	45.17	19.98	24.91
61	45	25	10	9.74	3.44	1.29	45.10	25.09	9.91
62	45	25	15	10.05	3.56	2.19	44.98	25.03	14.99
63	45	25	20	10.32	3.68	3.42	45.04	24.92	19.99
64	45	25	25	11.03	3.96	5.45	45.25	25.08	25.03
65	50	10	10	9.57	1.15	1.20	50.01	9.99	9.96
66	50	10	15	9.94	1.22	2.07	49.98	10.04	15.08
67	50	10	20	10.23	1.26	3.23	49.85	10.03	20.02
68	50	10	25	10.74	1.31	5.13	50.16	9.94	25.08
69	50	15	10	10.52	1.94	1.27	50.03	15.08	10.06

Continued on Next Page...

Table D.1 – Continued

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
70	50	15	15	10.88	2.00	2.14	50.08	15.00	15.02
71	50	15	20	11.25	2.10	3.34	50.06	15.01	19.98
72	50	15	25	11.66	2.18	5.33	50.05	14.99	25.08
73	50	20	10	11.49	2.84	1.31	49.98	19.96	10.05
74	50	20	15	11.90	2.95	2.21	50.07	19.93	14.97
75	50	20	20	12.30	3.09	3.47	49.99	19.98	20.06
76	50	20	25	12.77	3.24	5.47	49.98	19.94	25.12
77	50	25	10	12.84	4.01	1.36	50.04	24.95	10.01
78	50	25	15	13.28	4.20	2.30	49.97	25.09	15.01
79	50	25	20	13.92	4.43	3.59	50.12	25.00	19.96
80	50	25	25	14.39	4.61	5.64	50.08	25.06	25.00
81	55	10	10	12.30	1.33	1.25	55.09	10.02	9.90
82	55	10	15	12.83	1.39	2.14	55.15	9.96	14.95
83	55	10	20	13.31	1.45	3.38	55.19	9.99	20.01
84	55	10	25	13.87	1.54	5.33	55.26	10.00	25.06
85	55	15	10	13.72	2.26	1.32	55.03	15.01	10.01
86	55	15	15	14.14	2.35	2.22	55.04	15.05	14.98
87	55	15	20	14.63	2.45	3.49	55.03	14.96	19.96
88	55	15	25	15.15	2.56	5.47	55.04	15.00	25.03
89	55	20	10	15.29	3.42	1.36	55.04	20.02	9.98
90	55	20	15	16.13	3.61	2.34	55.22	20.01	15.02
91	55	20	20	16.47	3.73	3.65	55.12	20.00	20.04
92	55	20	25	17.12	3.91	5.68	55.11	20.08	25.00
93	55	25	10	17.70	4.99	1.44	55.04	24.91	10.06
94	55	25	15	18.71	5.35	2.46	55.15	25.16	15.06

Continued on Next Page...

Table D.1 – Continued

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
95	55	25	20	18.72	5.37	3.79	54.94	25.00	20.02
96	55	25	25	19.68	5.67	5.90	55.10	24.98	24.99
97	60	10	10	16.10	1.58	1.32	60.07	9.94	9.99
98	60	10	15	16.67	1.67	2.25	60.02	10.06	15.01
99	60	10	20	17.30	1.72	3.55	60.08	9.98	20.08
100	60	10	25	17.91	1.80	5.57	60.00	9.98	25.11
101	60	15	10	18.26	2.75	1.38	59.98	14.98	10.00
102	60	15	15	19.04	2.89	2.35	60.14	15.05	14.98
103	60	15	20	19.58	2.99	3.65	59.93	15.00	20.01
104	60	15	25	20.51	3.13	5.72	60.14	15.01	25.02
105	60	20	10	21.36	4.35	1.45	60.00	19.99	10.00
106	60	20	15	22.48	4.56	2.47	60.15	19.89	15.04
107	60	20	20	22.77	4.68	3.79	60.06	20.00	19.90
108	60	20	25	23.65	4.88	5.97	60.10	19.96	25.05
109	60	25	10	26.15	6.72	1.52	60.00	24.96	10.03
110	60	25	15	27.26	7.08	2.61	60.06	25.03	15.09
111	60	25	20	27.87	7.25	3.98	60.05	25.00	19.95
112	60	25	25	28.70	7.54	6.20	60.08	25.00	24.98
113	65	10	10	22.24	2.00	1.41	65.16	9.98	10.02
114	65	10	15	22.89	2.06	2.38	65.17	9.96	15.02
115	65	10	20	23.65	2.15	3.72	65.22	9.98	20.05
116	65	10	25	24.59	2.26	5.77	65.17	10.03	25.04
117	65	15	10	26.65	3.67	1.47	65.22	14.97	10.00
118	65	15	15	26.83	3.74	2.50	65.00	15.04	15.03
119	65	15	20	28.15	3.94	3.89	65.12	15.05	20.04

Continued on Next Page...

Table D.1 – Continued

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
120	65	15	25	28.99	4.08	6.02	65.24	15.11	25.00
121	65	20	10	33.83	6.30	1.56	65.19	19.90	10.03
122	65	20	15	35.35	6.64	2.64	65.35	20.06	15.05
123	65	20	20	35.50	6.65	4.06	65.21	19.83	20.06
124	65	20	25	36.63	6.94	6.26	65.28	20.03	25.03
125	65	25	10	53.38	12.80	1.63	64.99	24.97	10.00
126	65	25	15	52.34	12.54	2.77	65.02	24.91	15.03
127	65	25	20	56.36	13.70	4.24	64.90	25.05	19.97
128	65	25	25	53.88	13.05	6.60	65.06	24.92	25.10
129	70	10	10	32.65	2.73	1.49	70.02	10.01	10.01
130	70	10	15	33.13	2.77	2.50	69.99	9.95	14.99
131	70	10	20	34.23	2.89	3.93	70.06	10.02	20.07
132	70	10	25	35.57	3.02	6.01	70.09	9.99	24.95
133	70	15	10	44.11	5.68	1.57	69.98	15.03	10.00
134	70	15	15	45.25	5.83	2.64	70.06	14.99	15.00
135	70	15	20	45.11	5.84	4.13	69.99	14.96	20.07
136	70	15	25	47.98	6.28	6.32	70.03	15.07	25.00
137	70	20	10	88.57	15.93	1.66	69.67	19.97	10.00
138	70	20	15	87.06	15.69	2.82	69.49	19.92	15.07
139	70	20	20	98.30	17.87	4.31	69.61	19.96	20.00
140	70	20	25	95.48	17.32	6.67	69.42	19.92	25.09
141	70	25	10	273.47	64.81	1.71	66.00	24.07	10.03
142	70	25	15	267.24	63.28	2.90	66.11	24.04	15.07
143	70	25	20	266.78	63.04	4.51	66.09	24.07	20.11
144	70	25	25	258.52	61.49	6.85	65.98	24.17	24.99

Continued on Next Page...

Table D.1 – Continued

Set	Product Mix			Simulation WIP			Simulation Th (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
145	75	10	10	61.98	4.91	1.59	74.87	9.96	9.97
146	75	10	15	61.88	4.93	2.69	74.77	9.98	15.00
147	75	10	20	62.39	4.97	4.13	74.78	9.95	19.92
148	75	10	25	66.96	5.39	6.39	74.97	10.02	24.97
149	75	15	10	171.06	22.02	1.66	72.99	14.77	9.92
150	75	15	15	163.56	20.90	2.83	73.20	14.71	15.05
151	75	15	20	161.84	20.81	4.35	73.03	14.78	19.98
152	75	15	25	168.58	21.78	6.72	73.14	14.76	25.11
153	75	20	10	384.50	68.40	1.70	68.51	18.84	10.02
154	75	20	15	416.49	74.16	2.85	68.42	18.79	14.94
155	75	20	20	380.49	67.87	4.44	68.61	18.89	19.99
156	75	20	25	400.52	71.24	6.85	68.44	18.78	24.99
157	75	25	10	620.76	140.70	1.73	63.99	22.48	9.95
158	75	25	15	644.15	145.75	2.93	64.05	22.45	15.01
159	75	25	20	634.32	144.02	4.53	63.98	22.52	19.98
160	75	25	25	648.97	146.53	6.87	63.91	22.41	24.95

D.2 FPR Grid for the Closed Queuing Network

Table D.2: All release combinations, simulation WIP, and MVA throughput

Set	Product Mix			Simulation WIP			MVA Throughput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	30	10	10	3.66	0.77	1.01	37.49	12.91	13.53
2	30	10	15	3.79	0.81	1.74	37.59	13.04	20.27
3	30	10	20	3.93	0.84	2.76	37.83	12.92	26.44
4	30	10	25	4.08	0.88	4.42	38.07	13.03	31.44
5	30	15	10	3.86	1.23	1.04	37.61	19.33	13.53
6	30	15	15	3.99	1.29	1.78	37.64	19.46	20.14
7	30	15	20	4.15	1.36	2.86	37.78	19.59	26.45
8	30	15	25	4.31	1.41	4.51	38.06	19.64	31.34
9	30	20	10	4.08	1.76	1.08	37.55	25.89	13.51
10	30	20	15	4.23	1.84	1.84	37.67	25.88	20.13
11	30	20	20	4.42	1.93	2.92	37.92	25.97	26.28
12	30	20	25	4.61	2.03	4.73	38.16	26.14	31.44
13	30	25	10	4.34	2.36	1.12	37.53	32.31	13.49
14	30	25	15	4.53	2.47	1.93	37.76	32.26	20.23
15	30	25	20	4.69	2.60	3.03	37.71	32.49	26.28
16	30	25	25	4.90	2.74	4.79	37.98	32.65	31.25
17	35	10	10	4.73	0.84	1.05	44.14	12.87	13.52
18	35	10	15	4.91	0.88	1.81	44.29	12.93	20.26
19	35	10	20	5.06	0.92	2.86	44.26	13.03	26.39
20	35	10	25	5.26	0.96	4.56	44.60	13.06	31.38
21	35	15	10	5.00	1.36	1.10	44.03	19.40	13.65
22	35	15	15	5.20	1.42	1.86	44.23	19.47	20.09
23	35	15	20	5.34	1.48	2.98	44.07	19.49	26.45
24	35	15	25	5.58	1.55	4.67	44.48	19.53	31.29
25	35	20	10	5.32	1.94	1.13	43.96	25.73	13.56

Continued on Next Page...

Table D.2 – Continued

Set	Product Mix			Simulation WIP			MVA Throughput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
26	35	20	15	5.51	2.04	1.93	43.98	25.86	20.15
27	35	20	20	5.68	2.12	3.04	43.95	25.84	26.28
28	35	20	25	5.96	2.23	4.86	44.34	25.91	31.32
29	35	25	10	5.66	2.62	1.18	43.61	32.03	13.65
30	35	25	15	5.91	2.75	2.01	43.82	32.09	20.17
31	35	25	20	6.11	2.88	3.14	43.79	32.18	26.18
32	35	25	25	6.36	3.02	4.96	43.98	32.23	31.19
33	40	10	10	5.99	0.93	1.11	50.37	12.93	13.59
34	40	10	15	6.22	0.97	1.90	50.52	12.99	20.30
35	40	10	20	6.40	1.00	2.96	50.51	12.88	26.31
36	40	10	25	6.63	1.05	4.70	50.72	12.95	31.29
37	40	15	10	6.39	1.50	1.15	50.27	19.29	13.61
38	40	15	15	6.61	1.57	1.94	50.29	19.36	20.09
39	40	15	20	6.84	1.63	3.05	50.37	19.23	26.21
40	40	15	25	7.11	1.70	4.83	50.62	19.29	31.23
41	40	20	10	6.81	2.16	1.19	49.84	25.52	13.62
42	40	20	15	7.09	2.26	2.02	50.03	25.52	20.16
43	40	20	20	7.32	2.35	3.21	49.99	25.44	26.40
44	40	20	25	7.62	2.47	5.05	50.21	25.52	31.30
45	40	25	10	7.42	2.98	1.23	49.59	31.70	13.58
46	40	25	15	7.67	3.08	2.09	49.62	31.52	20.10
47	40	25	20	7.98	3.24	3.29	49.70	31.54	26.22
48	40	25	25	8.27	3.39	5.19	49.75	31.63	31.21
49	45	10	10	7.51	1.03	1.15	56.29	12.82	13.58
50	45	10	15	7.88	1.07	1.97	56.69	12.77	20.18
51	45	10	20	8.11	1.12	3.09	56.61	12.79	26.25
52	45	10	25	8.40	1.18	4.84	56.74	12.96	31.17

Continued on Next Page...

Table D.2 – Continued

Set	Product Mix			Simulation WIP			MVA Throughput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
53	45	15	10	8.05	1.67	1.19	55.88	19.04	13.55
54	45	15	15	8.36	1.74	2.03	56.01	18.99	20.12
55	45	15	20	8.78	1.85	3.21	56.32	19.15	26.26
56	45	15	25	9.12	1.93	5.04	56.52	19.11	31.20
57	45	20	10	8.85	2.44	1.25	55.67	24.92	13.70
58	45	20	15	9.17	2.57	2.11	55.69	25.06	20.15
59	45	20	20	9.48	2.67	3.33	55.70	24.93	26.30
60	45	20	25	9.91	2.81	5.20	55.94	25.04	31.17
61	45	25	10	9.74	3.44	1.29	54.88	30.90	13.61
62	45	25	15	10.05	3.56	2.19	54.88	30.75	20.16
63	45	25	20	10.32	3.68	3.42	54.80	30.67	26.18
64	45	25	25	11.03	3.96	5.45	55.27	30.84	31.25
65	50	10	10	9.57	1.15	1.20	62.08	12.55	13.49
66	50	10	15	9.94	1.22	2.07	62.15	12.66	20.29
67	50	10	20	10.23	1.26	3.23	62.13	12.57	26.28
68	50	10	25	10.74	1.31	5.13	62.57	12.49	31.31
69	50	15	10	10.52	1.94	1.27	61.50	18.72	13.74
70	50	15	15	10.88	2.00	2.14	61.61	18.53	20.24
71	50	15	20	11.25	2.10	3.34	61.60	18.59	26.22
72	50	15	25	11.66	2.18	5.33	61.72	18.51	31.35
73	50	20	10	11.49	2.84	1.31	60.47	24.23	13.78
74	50	20	15	11.90	2.95	2.21	60.53	24.16	20.21
75	50	20	20	12.30	3.09	3.47	60.50	24.24	26.32
76	50	20	25	12.77	3.24	5.47	60.57	24.28	31.29
77	50	25	10	12.84	4.01	1.36	59.24	29.53	13.88
78	50	25	15	13.28	4.20	2.30	59.20	29.63	20.37
79	50	25	20	13.92	4.43	3.59	59.35	29.62	26.31

Continued on Next Page...

Table D.2 – Continued

Set	Product Mix			Simulation WIP			MVA Throughput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
80	50	25	25	14.39	4.61	5.64	59.39	29.59	31.29
81	55	10	10	12.30	1.33	1.25	67.10	12.21	13.51
82	55	10	15	12.83	1.39	2.14	67.27	12.19	20.12
83	55	10	20	13.31	1.45	3.38	67.36	12.17	26.31
84	55	10	25	13.87	1.54	5.33	67.53	12.25	31.27
85	55	15	10	13.72	2.26	1.32	65.98	17.95	13.74
86	55	15	15	14.14	2.35	2.22	65.97	17.93	20.22
87	55	15	20	14.63	2.45	3.49	66.01	17.89	26.31
88	55	15	25	15.15	2.56	5.47	66.07	17.92	31.26
89	55	20	10	15.29	3.42	1.36	64.37	23.25	13.84
90	55	20	15	16.13	3.61	2.34	64.60	23.18	20.53
91	55	20	20	16.47	3.73	3.65	64.47	23.22	26.52
92	55	20	25	17.12	3.91	5.68	64.54	23.27	31.34
93	55	25	10	17.70	4.99	1.44	62.72	28.04	14.21
94	55	25	15	18.71	5.35	2.46	62.75	28.23	20.88
95	55	25	20	18.72	5.37	3.79	62.62	28.12	26.71
96	55	25	25	19.68	5.67	5.90	62.77	28.10	31.46
97	60	10	10	16.10	1.58	1.32	71.09	11.67	13.77
98	60	10	15	16.67	1.67	2.25	71.06	11.82	20.34
99	60	10	20	17.30	1.72	3.55	71.21	11.66	26.50
100	60	10	25	17.91	1.80	5.57	71.27	11.72	31.38
101	60	15	10	18.26	2.75	1.38	69.31	17.07	13.97
102	60	15	15	19.04	2.89	2.35	69.37	17.12	20.58
103	60	15	20	19.58	2.99	3.65	69.38	17.07	26.51
104	60	15	25	20.51	3.13	5.72	69.56	16.99	31.39
105	60	20	10	21.36	4.35	1.45	67.38	21.95	14.26
106	60	20	15	22.48	4.56	2.47	67.57	21.78	20.99

Continued on Next Page...

Table D.2 – Continued

Set	Product Mix			Simulation WIP			MVA Throughput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
107	60	20	20	22.77	4.68	3.79	67.43	21.89	26.72
108	60	20	25	23.65	4.88	5.97	67.47	21.88	31.58
109	60	25	10	26.15	6.72	1.52	65.34	26.34	14.61
110	60	25	15	27.26	7.08	2.61	65.28	26.47	21.45
111	60	25	20	27.87	7.25	3.98	65.30	26.40	27.11
112	60	25	25	28.70	7.54	6.20	65.26	26.50	31.75
113	65	10	10	22.24	2.00	1.41	74.14	11.02	14.13
114	65	10	15	22.89	2.06	2.38	74.15	10.98	20.77
115	65	10	20	23.65	2.15	3.72	74.16	11.03	26.75
116	65	10	25	24.59	2.26	5.77	74.21	11.06	31.46
117	65	15	10	26.65	3.67	1.47	71.96	15.95	14.39
118	65	15	15	26.83	3.74	2.50	71.81	16.04	21.13
119	65	15	20	28.15	3.94	3.89	71.90	16.02	27.07
120	65	15	25	28.99	4.08	6.02	71.91	16.03	31.66
121	65	20	10	33.83	6.30	1.56	69.70	20.42	14.88
122	65	20	15	35.35	6.64	2.64	69.65	20.50	21.63
123	65	20	20	35.50	6.65	4.06	69.69	20.37	27.40
124	65	20	25	36.63	6.94	6.26	69.62	20.51	31.85
125	65	25	10	53.38	12.80	1.63	67.24	24.82	15.23
126	65	25	15	52.34	12.54	2.77	67.25	24.76	22.09
127	65	25	20	56.36	13.70	4.24	67.13	24.96	27.73
128	65	25	25	53.88	13.05	6.60	67.15	24.87	32.14
129	70	10	10	32.65	2.73	1.49	76.08	10.32	14.56
130	70	10	15	33.13	2.77	2.50	76.09	10.26	21.16
131	70	10	20	34.23	2.89	3.93	76.07	10.31	27.20
132	70	10	25	35.57	3.02	6.01	76.10	10.32	31.66
133	70	15	10	44.11	5.68	1.57	73.61	14.95	15.00

Continued on Next Page...

Table D.2 – Continued

Set	Product Mix			Simulation WIP			MVA Throughput (units\week)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
134	70	15	15	45.25	5.83	2.64	73.63	14.91	21.71
135	70	15	20	45.11	5.84	4.13	73.57	14.92	27.65
136	70	15	25	47.98	6.28	6.32	73.55	15.02	31.95
137	70	20	10	88.57	15.93	1.66	70.92	19.53	15.51
138	70	20	15	87.06	15.69	2.82	70.90	19.54	22.40
139	70	20	20	98.30	17.87	4.31	70.85	19.63	27.97
140	70	20	25	95.48	17.32	6.67	70.86	19.59	32.25
141	70	25	10	273.47	64.81	1.71	67.84	24.25	15.76
142	70	25	15	267.24	63.28	2.90	67.85	24.22	22.66
143	70	25	20	266.78	63.04	4.51	67.88	24.17	28.39
144	70	25	25	258.52	61.49	6.85	67.80	24.29	32.37
145	75	10	10	61.98	4.91	1.59	77.37	9.64	15.17
146	75	10	15	61.88	4.93	2.69	77.33	9.66	21.98
147	75	10	20	62.39	4.97	4.13	77.33	9.65	27.67
148	75	10	25	66.96	5.39	6.39	77.32	9.70	32.05
149	75	15	10	171.06	22.02	1.66	74.31	14.56	15.54
150	75	15	15	163.56	20.90	2.83	74.36	14.46	22.50
151	75	15	20	161.84	20.81	4.35	74.31	14.53	28.14
152	75	15	25	168.58	21.78	6.72	74.27	14.57	32.34
153	75	20	10	384.50	68.40	1.70	71.27	19.12	15.77
154	75	20	15	416.49	74.16	2.85	71.26	19.12	22.50
155	75	20	20	380.49	67.87	4.44	71.24	19.15	28.29
156	75	20	25	400.52	71.24	6.85	71.28	19.09	32.42
157	75	25	10	620.76	140.70	1.73	68.45	23.33	15.90
158	75	25	15	644.15	145.75	2.93	68.47	23.29	22.78
159	75	25	20	634.32	144.02	4.53	68.43	23.35	28.44
160	75	25	25	648.97	146.53	6.87	68.50	23.25	32.39