HEURISTIC METHODS FOR THE STOCHASTIC LOT-SIZING PROBLEM UNDER THE STATIC-DYNAMIC UNCERTAINTY STRATEGY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ CEM RANDA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE IN OPERATIONAL RESEARCH PROGRAM
IN
INDUSTRIAL ENGINEERING

JULY 2013

Approval of the thesis:

HEURISTIC METHODS FOR THE STOCHASTIC LOT-SIZING PROBLEM UNDER THE STATIC-DYNAMIC UNCERTAINTY STRATEGY

submitted by **ALİ CEM RANDA** in partial fulfillment of the requirements for the degree of **Master of Science in Operational Research Program in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Murat Köksalan
Head of Department, **Industrial Engineering** ________________

Assist. Prof. Dr. Cem İyigün
Supervisor, **Industrial Engineering Dept., METU** ________________

Examining Committee Members:

Assoc. Prof. Dr. Haldun Süral
Industrial Engineering Dept., METU ________________

Assist. Prof. Dr. Cem İyigün
Industrial Engineering Dept., METU ________________

Assist. Prof. Dr. İsmail Serdar Bakal
Industrial Engineering Dept., METU ________________

Assist. Prof. Dr. Ulaş Özen
Faculty of Econ. and Admin. Sciences, Özyeğin University ________________

Assist. Prof. Dr. Seçil Savaşaneril Tüfekci
Industrial Engineering Dept., METU ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ALİ CEM RANDA

Signature :

ABSTRACT

HEURISTIC METHODS FOR THE STOCHASTIC LOT-SIZING PROBLEM UNDER THE STATIC-DYNAMIC UNCERTAINTY STRATEGY

Randa, Ali Cem
M.S. in Operational Research, Department of Industrial Engineering
Supervisor : Assist. Prof. Dr. Cem İyigün

July 2013, 56 pages

We consider a lot-sizing problem in a single-item single-stage production system facing non-stationary stochastic demand in a finite planning horizon. Motivated by practice, the set-up times need to be determined and frozen once and for all at the beginning of the horizon while decision on the exact lot sizes can be deferred until the setup time. This operating scheme is referred to as the static-dynamic uncertainty strategy in the literature. For a capacitated system with minimum lot size restrictions, it has been shown that a modified base stock policy is optimal under the static-dynamic uncertainty strategy. However, the optimal policy parameters require an exhaustive search for which the computational time grows exponentially in the number of periods in the planning horizon. In order to alleviate the computational burden for real-life size problems, we develop and test seven different heuristics for computational efficiency and solution quality. Our extensive numerical experiments show that optimality gaps below 1% can be attained in reasonable running times by using a combination of these heuristics.

Keywords: static-dynamic uncertainty, non-stationary, stochastic demand, heuristic, penalty cost

ÖZ

STATİK-DİNAMİK BELİRSİZLİK STRATEJİSİ ALTINDAKİ RASSAL PARTİ BÜYÜKLÜĞÜ PROBLEMİ İÇİN SEZGİSEL YÖNTEMLER

Randa, Ali Cem
Yüksek Lisans, Endüstri Mühendisliği Bölümü Yöneylem Araştırması Programı
Tez Yöneticisi : Yrd. Doç. Dr. Cem İyigün

Temmuz 2013, 56 sayfa

Çalışmamızda bir parça için bir kademede gerçekleşen, sabit olmayan rassal taleple karşılaşan, sonlu planlama ufkuna sahip bir parti büyüklüğü problemini ele aldık. Uygulama gereği, üretim için kurulum zamanları planlama ufkunun başında belirlenip sabitlenmekte, parti büyüklükleri ise kurulum gerçekleşene kadar kesinleşmemektedir. Bu işletim şekli literatürde statik-dinamik belirsizlik stratejisi olarak bilinmektedir. Kapasite ve minimum parti büyüklüğü kısıtlarına sahip bir ortamda, statik-dinamik belirsizlik stratejisi için optimal politikanın temek stok politikası olduğu ispatlanmıştır. Ancak, optimal politika parametrelerinin hesaplanması planlama ufku uzunluğu ile üssel büyüyen çözüm zamanları üreten zahmetli bir tarama gerektirmektedir. Gerçek hayat problemlerinde ortaya çıkabilecek bu sorunu gidermek adına, yedi farklı sezgisel yöntem üreterek bunları çözüm kalitesi ve hesaplama verimliliği açısından karşılaştırdık. Geniş çaplı sayısal çalışmamız, geliştirdiğimiz sezgisel yöntemler ve kombinasyonlarının optimal sonuçtan %1'in altında sapma gösterdiğini ispatlamıştır.

Anahtar Kelimeler: statik-dinamik belirsizlik, sabit olmayan, rassal talep, ceza maliyeti

to my Grandfathers

ACKNOWLEDGMENTS

I would like to thank to Dr. Mustafa Kemal Doğru and Assist. Prof. Dr. Ulaş Özen for their contribution to this work from beginning through the end. Their invaluable support for introducing the problem to me, providing ideas for particular solutions and other suggestions made this research possible.

I also would like to offer my appreciations to Assoc. Prof. Dr. Haldun Süral, not only for his endless support throughout this work, and but also tolerating me on his classes that I assisted. Above that, I would like to thank him for helping me to be a better person and standing besides me every moment that I needed them through the past three years.

More than anyone, I would like to thank to Assist. Prof. Dr. Cem İyigün. He was the one supported all of my actions in the past three years. If I had not met him, I would have gone to US soon before I should have. I could have lost under a heavy load that I could not lift on my own. His support and advices contributed as the primary factor in my research and PhD. applications. He helped me to gain back the self-confidence and the hope to be the researcher that I always wanted to be.

I also would like to express my gratitude to my parents and my siblings for being supportive throughout my life. They always shared the most stressful instants of my life and made them bearable. They were always there for me, and I know that they will always be.

Lastly, I would like to thank to my girl friend. Her love and patient support kept my heart warm and gave me the strength to finish this work with success. She kindly excused me for the time that I spent working on this thesis instead of being with her. Luckily, I will have more time in which I will make these years up to her and more.

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grant No: 2210. Generosity of TUBITAK by supplying this grant was very helpful by eliminating the stress of having financial issues. Hence I would like to offer my biggest appreciations to members of TUBITAK.

TABLE OF CONTENTS

ABSTRACT v
ÖZ vi
ACKNOWLEDGMENTS viii
TABLE OF CONTENTS ix
LIST OF TABLES xi
LIST OF FIGURES xiii
CHAPTERS

1 INTRODUCTION 1

2 LITERATURE REVIEW AND MODEL 3

2.1 Literature Review 3

2.1.1 Static Uncertainty Strategy 3

2.1.2 Dynamic Uncertainty Strategy 4

2.1.3 Capacity Constrained Models 4

2.1.4 Static Dynamic Uncertainty Strategy 4

2.1.4.1 Service Level Models 4

2.1.4.2 Pure Cost Models 5

2.2 Model 5

3 OPTIMAL SOLUTION FOR PROBLEM P 9

3.1 Characterisation of the Optimal Policy 9

3.2 Computing Optimal Order Levels 11

3.3 Bounds and Partial Characterizations 11

3.3.1 Lower Bound 1 12

3.3.2 Lower Bound 2 12

3.3.3 Lower Bound 3 14

3.4 Computing the Parameters of an Optimal Policy 15

4 DYNAMIC PROGRAMMING BASED HEURISTICS 17

4.1 Approximation Heuristic (AH) 17

4.2 Modified Approximation Heuristic I (AH I) 18

4.3 Modified Approximation Heuristic II (AH II) 20

5 COMBINATORIAL BASED HEURISTICS 21

5.1 Merging Method I (MMI) 22

5.2 Merging Method II (MMII) 23

5.3 Dividing Method I (DMI) 24

5.4 Dividing Method II (DMII) 25

6 COMPUTATIONAL EXPERIMENTS AND DISCUSSION 27

6.1 Problem Set 27

6.2 Lower Bound Effectiveness 28

6.3 Heuristic Accuracy 30

6.3.1 Constant Parameters 30

6.3.2 Dynamic Capacity 33

6.4 Computational Times 35

6.5 Statistical Analysis 37

6.6 Convexity 39

6.7 Discussion 39

7 CONCLUSION AND EXTENSIONS 41

REFERENCES 43

APPENDICES 44

A Heuristic Accuracy Tables 45

B Paired t-test Results 47

B.1 Heuristic Performance Comparisons for Constant Capacity 47

B.2 Heuristic Combinations Performance Comparisons for Constant Capacity . . . 49

B.3 Heuristic Performance Comparisons for Varying Capacity 51

B.4 Heuristic Combination Performance Comparisons for Varying Capacity 53

LIST OF TABLES

TABLES

Table 6.1 Parameters for λ_t . . . 28

Table 6.2 LB_3 Effectiveness with Number of Line Segments . . . 29

Table 6.3 Lower Bound Effectiveness Summary . . . 29

Table 6.4 Summary of Heuristics $N = 12$ . . . 30

Table 6.5 Minimum of Two Heuristics $N = 12$ . . . 32

Table 6.6 **e** vectors used for dynamic capacity scenarios . . . 33

Table 6.7 Summary of Heuristics $N = 12$ for Dynamic Capacity . . . 34

Table 6.8 Minimum of Two Heuristics $N = 12$ for Dynamic Capacity . . . 34

Table A.1 Summary of Heuristics $N = 18$ . . . 45

Table A.2 Summary of Heuristics $N = 24$ . . . 45

Table A.3 Summary of Heuristics $N = 30$ . . . 45

Table A.4 Summary of Heuristics $N = 36$ . . . 45

Table A.5 Minimum of Two Heuristics $N = 18$ . . . 45

Table A.6 Minimum of Two Heuristics $N = 24$ . . . 46

Table A.7 Minimum of Two Heuristics $N = 30$ . . . 46

Table B.1 Paired T-Test and CI: AH; AH II n=1 . . . 47

Table B.2 Paired T-Test and CI: AH; AH II n=2 . . . 47

Table B.3 Paired T-Test and CI: AH II n=1; AH II n=2 . . . 47

Table B.4 Paired T-Test and CI: AH II n=1; AH II n=3 . . . 48

Table B.5 Paired T-Test and CI: AH II n=3; AH II n=4 . . . 48

Table B.6 Paired T-Test and CI: AH II n=4; MMI . . . 48

Table B.7 Paired T-Test and CI: MMI; MMII . . . 48

Table B.8 Paired T-Test and CI: MMII; DMI . . . 48

Table B.9 Paired T-Test and CI: DMI; DMII . . . 49

Table B.10 Paired T-Test and CI: MM I-AH; MMI-AH I . . . 49

Table B.11 Paired T-Test and CI: MMI-AH I; MMI-AH II . . . 49

Table B.12 Paired T-Test and CI: MMII-AH I; MMII-AH . . . 49

Table B.13 Paired T-Test and CI: MMII-AH I; MMII-AH II . . . 49

Table B.14 Paired T-Test and CI: MMI-AH II; MMII-AH I . . . 50

Table B.15 Paired T-Test and CI: DMI-AH; DMI-AH I . . . 50

Table B.16 Paired T-Test and CI: DMI-AH I; DMI-AH II . . . 50

Table B.17 Paired T-Test and CI: MMII-AH II; DMI-AH . . . 50

Table B.18 Paired T-Test and CI: DMII-AH; DMII-AH I . . . 50

Table B.19 Paired T-Test and CI: DMII-AH I; DMII-AH II 51
Table B.20 Paired T-Test and CI: DMI-AH II;DMI-AH II 51
Table B.21 Paired T-Test and CI: AH; AH II n=1 51
Table B.22 Paired T-Test and CI: AH II n=1; AH II n=2 51
Table B.23 Paired T-Test and CI: AH II n=2; AH I 51
Table B.24 Paired T-Test and CI: AH I; AH II n=3 52
Table B.25 Paired T-Test and CI: AH II n=3; AH II n=4 52
Table B.26 Paired T-Test and CI: AH II n=4; MMII 52
Table B.27 Paired T-Test and CI: MMII; MMI 52
Table B.28 Paired T-Test and CI: MMI; DMI 52
Table B.29 Paired T-Test and CI: DMI; DMII 53
Table B.30 Paired T-Test and CI: MMI-AH; MMI-AH I 53
Table B.31 Paired T-Test and CI: MMI-AH I; MMI-AH II 53
Table B.32 Paired T-Test and CI: MMII-AH I; MMII-AH 53
Table B.33 Paired T-Test and CI: MMII-AH I; MMII-AH II 53
Table B.34 Paired T-Test and CI: DMI-AH; DMI-AH I 54
Table B.35 Paired T-Test and CI: DMI-AH I; DMI-AH II 54
Table B.36 Paired T-Test and CI: MMI-AH II; DMI-AH 54
Table B.37 Paired T-Test and CI: MMII-AH II; DMI-AH 54
Table B.38 Paired T-Test and CI: DMII-AH; DMII-AH I 54
Table B.39 Paired T-Test and CI: DMII-AH I; DMII-AH II 55
Table B.40 Paired T-Test and CI: MMI-AH II; DMII-AH 55
Table B.41 Paired T-Test and CI: MMII-AH II; DMII-AH 55
Table B.42 Paired T-Test and CI: DMI-AH I; DMII-AH I 55
Table B.43 Paired T-Test and CI: DMI-AH I; DMII-AH II 55
Table B.44 Paired T-Test and CI: DMI-AH II; DMII-AH I 56
Table B.45 Paired T-Test and CI: DMI-AH II; DMII-AH II 56

LIST OF FIGURES

FIGURES

Figure 6.1 Demand Patterns 28
Figure 6.2 Optimality Gap Distribution for DP Based Heuristics 31
Figure 6.3 Optimality Gap Distribution for Combinatorial Based Heuristics 31
Figure 6.4 Optimality Gap Distribution vs Maximum Lot Size 32
Figure 6.5 Dynamic Capacity Patterns 33
Figure 6.6 Computational Times versus *N* for Optimal Solution 35
Figure 6.7 Computational Times versus *N* for Dynamic Programming Based Heuristics 36
Figure 6.8 Computational Times versus *N* for Local Search Based Heuristics 37

CHAPTER 1

INTRODUCTION

The stochastic lot sizing problem basically deals with finite planning horizons under periodic review. The aim is to find the periods in which production will take place and the corresponding production quantities such that the total expected cost is minimized. One of the basic cost parameters is the setup cost. When a production decision is made, a fixed amount of cost should be incurred regardless of the production amount in order to compensate for the initial effort before production. Besides the setup cost, unit production cost is incurred for every item produced. Since the demand is stochastic, there is no guarantee to achieve zero inventory at the end of the periods. Also, the nature of the production environment may make it necessary to carry some of the items from one period to the next. In this case every unit on hand at the end of a period is charged with an inventory holding cost. In a lot sizing problem, it is likely to face demand realizations more than the inventory position after the production decision. In probabilistic models, unfulfilled demand is sometime inevitable because of the stochastic nature of demand. However, it is up to the decision maker whether or not to associate a cost with this unfulfilled demand. Each unit backlogged can be charged with a backlog cost on a per unit or per unit per unit time basis or backorders can be dealt with additional constraints that assure likeliness of a backlog below a certain probability.

Depending on the type of production, there may be instantaneous procurement or there may be some lead time for lot production. If there exists lead times, it may be a fixed amount, proportional with production amount, or even stochastic. Existence of lead times puts additional complexity to the problem, a need for safety stocks in order to meet lead time demand.

Another variant of the problem comes with possible capacities. There may be a restriction for maximum lot sizes. If the production environment is capacitated and/or the capacity is to be shared among various items to be produced, this restriction is likely to happen. Likewise, there might be a minimum amount that should be produced once a setup is done. This may be justified by high setup costs or supplier preference if present. There may also be upper limits for inventories kept on hand. Considering all these costs and constraints, ”when and how much to produce?" question is answered.

The stochastic lot sizing problem is one of the most popular problems with a large area for application. It is implementable for many industries where a setup with an associated cost is required for production. The problem is directly implementable for practice. The implementation of solution methods bring considerable improvements in terms of costs even for most general cases. Therefore, it is one of topics drawn most of the research attention.

While performing the cost minimization, the strategy to be followed also matters. There are three types of strategies used for deciding when and how much to produce. These strategies differ in terms of which decisions are made at what point of the planning horizon. The strategies are named as ”Static Uncertainty Strategy", ”Dynamic Uncertainty Strategy", and ”Static-Dynamic Uncertainty Strategy".

The simplest strategy can be named as "Static Uncertainty Strategy" where the timing and amount of production is fixed at the beginning of the planning horizon. Following this strategy brings many advantages for production planning activities. Having information about setup times and lot sizes with certainty is the main motivation. However, this type of strategy will produce cost-wise good results only when the demand distributions for periods realize low variance or are deterministic. Otherwise, it is very likely to have shortages or excessive inventories.

Another strategy to approach to the stochastic lot sizing problem is "Dynamic Uncertainty Strategy" where neither the timing nor the amount of production is fixed at the beginning of the planning horizon, but to be determined as time evolves. This strategy allows flexibility for inventory management by adjusting the inventory position at the beginning of each period. However, planning the production activities will be much harder because the lot sizes will not be known in advance. Above the exact lot sizes, even the presence of a setup will not be known for an upcoming period. However, corresponding inventory holding and backlog cost can be much lower compared to static uncertainty strategies.

Although Dynamic Uncertainty Strategies are cost effective, they are impracticable. Real life applications do not favor uncertainty on setups. For stochastic demand models, in balance between these extremes, there is "static-dynamic uncertainty strategy" where the timing of the production is fixed at the beginning of the planning horizon, however the exact quantities are to be determined when the production period comes. It is introduced by Bookbinder and Tan (1988). This type of strategy favors advance planning by deterministically declaring the periods for setup. It also favors inventory control activities by not determining the exact amount of production until the setup.

In this work, we will deal with a finite planning horizon under periodic review. We will have non-stationary and stochastic demand. We will consider a model with setup cost, unit production cost, inventory holding cost, and backlogging cost. And also, we will have upper and lower quantity limits for lot sizes. We will attempt to solve the problem for deciding the setup periods at the beginning of the planning horizon, and exact quantities when the setup period comes which is referred as static dynamic uncertainty strategy. Our work contributes to the literature in the following sense. We first proved the type of the optimal policy and showed that solving the problem for optimality requires an exponentially exhaustive search. For this reason, we adopted two existing lower bounds to our model and proposed a new lower bound to the problem in order to ease to effort for the search for optimal solution. Then we adopted one heuristic procedure from literature and made two extensions on that procedure to adjust for our model. Then we proposed four new heuristic procedures which take their bases from combinatorial analysis. We evaluated all of our heuristic procedures in terms of optimality gap, and saw that they are complementary on maximum lot sizes. Hence their combination can promise less than 1% optimality gap under certain parameters. We also tested the computational complexity of our procedures and observed that they are polynomial.

The rest of the manuscript is organized as follows: In Chapter 2, we will introduce the related literature for the topic in general and we will introduce the model and notation that we used in this work. Chapter 3 will characterize the optimal policy, how it is found, and how the solution time can be shortened by making use of lower bounds. Chapters 4 will present our Dynamic Programming Based heuristics which have their main idea from the other works in the literature. Chapter 5 will present Combinatorial Based heuristics which are our own contribution for solving the problem. Chapter 6 will give the result of our experimental study and some discussion on the findings. Chapter 7 will summarize our major work done and introduce our intended future work.

CHAPTER 2

LITERATURE REVIEW AND MODEL

In this chapter, we will first introduce some of the basic articles in the literature which are relevant to lot sizing problem. We will classify them according to which strategy is employed, what type of demand is used, what type of costs are considered, and the presence of capacities. Then, we will introduce the basics of our model, its notation, and where it falls under the classification criteria mentioned.

2.1 Literature Review

As mentioned before, lot sizing problem is one of the most dealt problems in the literature. In order to find the relevance with our work, we will classify the literature under the type of strategy that to-be-mentioned manuscripts refer to.

2.1.1 Static Uncertainty Strategy

Literature on Static Uncertainty Strategy mainly deals with cases with deterministic demand. The important factor is that since the demand is deterministic back orders may or may not be allowed. If back orders are allowed than an associated cost per unit or per unit/time should be charged accordingly. Wagner and Whitin (1958), the first example of this type of strategy where varying holding and setups without backlog are considered for a single item problem with deterministic demand. Note that no unit production cost is considered. It solves the problem by dynamic programming, then turns into one of the most popular solution method to be used. It proves that in the optimal solution there should either be on hand inventory or a setup in a period which shows a complementary slackness between lot size and the opening inventory level for a period. Zangwill (1969) extends the Wagner-Whitin model allowing backlogging. It shows that Wagner-Whitin model can be represented as a single source network with concave costs and presents a dynamic programming algorithm for optimal solution. Vargas (2009) also finds the optimal solution to the stochastic version of Wagner-Whitin model employing static strategy by constructing an acyclic network and solving for shortest path. Sox (1997) solves Wagner-Whitin model with random demand allowing backlogging and constructing a mixed integer nonlinear program with non-stationary costs. Sox and Muckstadt (1997) extends this model to multi-item, capacitated case, and proposes a near optimal procedure through solving for lagrangian duals.

2.1.2 Dynamic Uncertainty Strategy

Dynamic Uncertainty Strategy is suitable for stochastic demand environments. Since the demand cannot be known for certainty, no backlogging cannot be assured. Backlogging are handled either incurring backlog cost or assuring a certain service level in the models. Because neither the setup periods nor the lot sizes are determined at the beginning of the planning horizon in dynamic strategy; through the time, at the beginning of each period a decision for setup and lot size is made. A setup is only meaningful, if the proposed cost reduction for inventory and backlog related costs can justify the cost of setup. Scarf (1959) introduces this concept also defining K-convexity. It proves that (s, S) policy is optimal for such problems. (s, S) policy refers to the operations that at the beginning of a period if the inventory level is above s, no setup is placed, otherwise a setup takes place and current inventory position is set to S by a lot production. Veinott and Wagner (1965) extends the Wagner-Within model with stochastic demand and adding lead times. The optimal policy turn out to be an (s, S) policy under this setting. Then it develops a computational approach for finding optimal (s, S) inventory parameters. Sethi and Cheng (1997) also proves that if there is Markovian demand, (s, S) policy is optimal for such strategy. It also studies the presence of no ordering periods and storage constraints, and shows that (s, S) policy is still optimal. Guan and Miller (2008) models the stochastic lot sizing problem with a scenario tree based demand distribution and suggests a polynomial time algorithm proving optimality conditions. Huang and Kucukyavuz (2008) follows this scenario tree based distribution expanding it to stochastic and dependent costs, demands, and order lead times. Then it proposes a polynomial time algorithm in tree size for finding optimal solution.

2.1.3 Capacity Constrained Models

Lot sizing problem gets more complicated when there is capacity constraints on the lot sizes. Presence of capacity constraints has an essential effect on the problem complexity. Bitran and Yanase (1982) discusses the complexity of capacitated lot sizing problem for different setup cost, holding cost, production cost, and capacity structures. It concludes that most of the cases, when capacity is monotone are NP-Hard. Even constant capacity cases are considered to be hard to solve for stochastic demand problems. Akbalik and Rapine (2012) provides polynomial-time algorithms for constant capacity problem with deterministic demand and stepwise production cost. Okhrin and Richter (2011) embeds minimum order quantities into the deterministic model, and suggests a polynomial-time algorithm for exact solution. Önal et al. (2012) approaches to the capacitated lot sizing problem with inventory bounds with several polynomial time algorithms none of which guarantees the optimal solution.

2.1.4 Static Dynamic Uncertainty Strategy

Static dynamic uncertainty strategy has the benefits of both of the strategies discussed above. Following this strategy, there are two types of models regarding to how they deal with backlogging. One constraints the backlogging probability under a certain service level. The other just quantifies the cost of backlog and takes it as a part of the objective function.

2.1.4.1 Service Level Models

The first type of model ignores backlogging costs imposing service level constraints to provide a certain probability against backorders. Tarim and Kingsman (2004) presents a mixed integer program embracing this model. It includes chance constraints to ensure service level. Tarim et al. (2009) takes the perspective of finding the optimality parameters of (R, S) policy by computationally efficient methods for the method introduced in Tarim and Kingsman (2004). Tarim and Kingsman (2006) finds an approximate solution by relaxing service level constraints. Özen et al. (2012) also deals with static dynamic uncertainty strategy with service level constraints by a dynamic programming based exact and heuristic solution methods.

2.1.4.2 Pure Cost Models

The other type of model dwells upon costs only, including backlogging costs to the objective function. Özen et al. (2012) also deals with this type of model following the solution approaches mentioned above. However, Özen et al. (2012) deals with a very special case of the problem where setup, holding, and backlogging costs are present but unit production cost and lot size restrictions are ignored. Two heuristic procedures are introduced namely approximation heuristic (AH) and relaxation heuristic (RH). Power of these two heuristics are tested under a numerical study.

In this work we will deal with a similar pure cost problem in Özen et al. (2012) under additional presence of unit production costs, upper and lower limits on lot size. We will take the same approach while solving the optimal solution and adopt its heuristic procedures if appropriate in terms of unit production cost, upper and lower limits for lot sizes. We will extend our adopted heuristics, as well as suggesting new heuristic procedures. Then, we will evaluate our heuristics and their combinations in terms of time efficiency and accuracy.

2.2 Model

Consider a single-stage single-item production system in a finite horizon under periodic review. The planning horizon consists of equal length periods of one time unit (days, weeks, etc.). Let $t \in \{1, 2, ..., N\}$ be the index for periods with N being the last period in the planning horizon. Let D_t be a non-negative continuous or discrete random variable which denotes the demand in period t. It is assumed that demands in different periods are independent; however, they are not necessarily identically distributed. If there is unfulfilled demand of one unit in period t, it is backlogged to be satisfied in the following periods and a unit penalty cost of b_t per unit is incurred at the end of period t. Similarly, for inventory carrying, holding cost h_t per unit is charged for period t. We assume that the production environment is capacitated in terms of lot size. Once there is a setup, the lot size to be produced should be greater than a minimum requirement (o_t). Also, the size of a lot should be less than capacity for the period (u_t). Additionally, production lead time is assumed to be deterministic. Hence, zero lead time can be assumed without loss of generality. A set up cost A_t is incurred if a setup is scheduled for period t. Also, unit production cost of c_t is paid for each unit in the lot to be produced in period t.

We assume that all cost parameters and quantity restrictions are nonnegative: $c_t, h_t, b_t, A_t, u_t, o_t \geq 0$. The anticipated cost of inventory at the end of the planning horizon is assumed to be a convex function of the net inventory level at the end of period N which will lead to our analysis in Chapter 3 for finding the optimal policy.

The system evolves as follows. The *production periods* are determined at the beginning of period 1 in accordance with static-dynamic uncertainty strategy, i.e. the periods in which setups will be present are scheduled and fixed. The function δ_t indicates these production periods by taking the value 1 if there is a setup in period t, and 0 otherwise. Note that the setups are done at the beginning of the periods, and the procurement is instantaneous. The consecutive periods which are to be covered by the same lot produced, i.e. the periods between two consecutive setups, will be called as a production cycle. Index $i \in \{1, \ldots m\}$ will denote these cycles where m is the last cycle. Let $(m, \mathbf{r})$ be a *production schedule*, where m denotes the number of lots to be produced in the planning horizon ($m \leq N$) and $\mathbf{r} = (\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_m})$ is an ordered vector of periods in which there is a setup such that $\delta_{r_i} = 1$ for $i = 1, \ldots, m$. We assume that a setup is scheduled at the beginning of the planning horizon, i.e. $\delta_1 = 1$. The following sequence of events takes place in a given period t (ie., $\delta_t = 1$) with a setup:

- Setup cost of A_t is incurred.
- At the beginning of the period, the inventory level is observed. The current cycle's lot size x_i is determined, $u_t \geq x_i \geq o_t$. Then instantaneous production takes place and the production cost $c_t x_i$ is incurred (*at the beginning of the period*);
- Demand occurs within the period, D_t;
- Holding and penalty costs are incurred according to the ending inventory position (*at the end of the period*).

If it is not a setup period (ie. $\delta_t = 0$);

- Inventory level is observed at the beginning. No lot is produced;
- Demand occurs within the period, D_t;
- Holding and penalty costs are incurred according to the ending inventory position (*at the end of the period*).

In order to get into more detail about the overall cost structure of a single cycle, let us consider the costs incurred. We will also be incurring costs due to inventory holding and backlog. Since our demand is random, we shall work with expected costs. For the production cycle i, let $\mathcal{L}_i(I_i, x_i)$ denote the expected holding and penalty costs given the net inventory level at the beginning of the cycle I_i and the lot size x_i:

$$\mathcal{L}_i(I_i, x_i) = \mathsf{E}\left[\sum_{k=r_i}^{r_{i+1}-1}\left(h_k\Big(I_i + x_i - \sum_{j=r_i}^{k} D_j\Big)^+ + b_k\Big(I_i + x_i - \sum_{j=r_i}^{k} D_j\Big)^-\right)\right],$$

where $u^+ = \max\{u, 0\}$ and $u^- = \max\{-u, 0\}$ for any $u \in \mathbb{R}$. Let $y_i = I_i + x_i$ be the inventory level just after the production in cycle i. Note that $\mathcal{L}_i(I_i, x_i)$ is a function of y_i, hence we can rewrite the expression above as

$$\mathcal{L}_i(y_i) = \mathsf{E}\left[\sum_{k=r_i}^{r_{i+1}-1}\left(h_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^+ + b_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^-\right)\right]. \tag{2.1}$$

Equation 2.1 helps us to change our perspective in the sense that taking "up to how much to produce?" decisions, rather than "how much to produce?" decisions.

Thinking in terms of a production cycle alone, we incur a fixed setup cost, variable cost proportional to the size of the lot, and inventory/backlogging cost depending on the opening inventory and demand realizations. Let us define function $\bar{\mathcal{L}}(I_i, x_i)$:

$$\bar{\mathcal{L}}(I_i, x_i) = A_{r_i} + c_{r_i} x_i + \mathcal{L}_i(I_i + x_i) \tag{2.2}$$

Equation (2.2) helps us to understand this overall cost frame for a single production cycle i considering these costs but ignoring the cycles thereafter. However, in reality cycle costs are not incurred separately. They are indeed linked by nested calculations of expected costs.

Let us denote the net inventory level at the beginning of the i-th production cycle (at the beginning of period r_i) as I_i. Note that it is the net inventory level at the end of production cycle $i-1$ before a setup and production. Starting from the beginning of a setup period (just after production), there is no inflow of the inventory on hand. But inventory position will go down when the demand occurs in the periods through the production cycle. The relation of the initial inventory levels of production cycles in the system under a production schedule $(m, \mathbf{r})$ is given by equation below:

$$I_{i+1} \quad = \quad I_i + x_i - \sum_{k=r_i}^{r_{i+1}-1} D_k, \qquad i = 1, \ldots, m, \tag{2.3}$$

where I_1 and I_{m+1} are the net inventory at the beginning and the end of the planning horizon, respectively, and $r_{m+1} = N + 1$. Equation 2.3 will constitute the base of our analysis on how particular decisions in different production cycles affect each other.

At the beginning of the planning horizon, the timing of the setups is fixed, but the exact lot sizes depend on how the demand is realized. Hence, the setup cost will be known with certainty as soon as the production schedule is determined. However, others costs of production will vary according to the demand realizations and end-of-cycle inventory levels which will determine the next amount of production cost and corresponding inventory related costs. This nested scheme will makes its way through the planning horizon and we will end up with an expected cost associated with our decisions. Our objective is to find the production schedule $(m, \mathbf{r})$ and the inventory levels just after production (y_i) in the corresponding production period i for all $i = 1, \ldots, m$ such that the expected total cost over the planning horizon is minimized, or equivalently, to solve the following problem:

$$\min_{(m,\mathbf{r})} \Bigg\{ \min_{y_1 : I_1 + o_{r_1} \geq y_1 \geq I_1 + u_{r_1}} \Bigg\{ A_{r_1} + c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\ldots,D_{r_2-1}} \Bigg[\min_{y_2 : I_2 + o_{r_2} \geq y_2 \geq I_2 + u_{r_2}} \Big\{ A_{r_2} + c_{r_2}(y_2 - I_2)$$
$$+ \mathcal{L}_2(y_2) + \mathsf{E}_{D_{r_2},\ldots,D_{r_3-1}} \Big[\cdots \big[\min_{y_m : I_m + o_{r_m} \geq y_m \geq I_m + u_{r_m}} \{ A_{r_m} + c_{r_m}(y_m - I_m) + \mathcal{L}_m(y_m) +$$
$$\mathsf{E}_{D_{r_m},\ldots,D_{r_{m+1}-1}} [B(I_{m+1})] \} \big] \cdots \Big] \Big\} \Bigg] \Bigg\} \Bigg\}$$

Since there is no time value of money is considered, we can modify this problem and define it as **Problem P**:

$$\min_{(m,\mathbf{r})} \Bigg\{ \sum_{k=1}^{N} A_k \delta_k + \min_{y_1 : I_1 + o_{r_1} \geq y_1 \geq I_1 + u_{r_1}} \Bigg\{ c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\ldots,D_{r_2-1}} \Bigg[\min_{y_2 : I_2 + o_{r_2} \geq y_2 \geq I_2 + u_{r_2}} \Big\{ c_{r_2}(y_2 - I_2)$$
$$+ \mathcal{L}_2(y_2) + \mathsf{E}_{D_{r_2},\ldots,D_{r_3-1}} \Big[\cdots \big[\min_{y_m : I_m + o_{r_m} \geq y_m \geq I_m + u_{r_m}} \{ c_{r_m}(y_m - I_m) + \mathcal{L}_m(y_m) +$$
$$\mathsf{E}_{D_{r_m},\ldots,D_{r_{m+1}-1}} [B(I_{m+1})] \} \big] \cdots \Big] \Big\} \Bigg] \Bigg\} \Bigg\}$$

(The subscript of the expectation operator E shows the random variables over which the expectation is taken.) It is worthwhile to note that decision y_i (the inventory level after production in period r_i) depends on the information revealed up to cycle i including the demand materialized in production cycles $1, \ldots, i-1$ and the previous decisions $y_1, \ldots, y_{i-1}$, and the future production periods $r_{i+1}, \ldots, r_m$.

We assume that the anticipated cost of inventory at the end of the planning horizon is a linear function of end of horizon inventory level i.e. $B(q) = -c_{r_m} q$, $\forall q \in \mathbb{R}$. Such that, at the end of the horizon; if there is on hand inventory, it will be sold from the price of the last items produced; if there is back orders , it will be cleared with the same price.

Finally, we assume $c_{t_1} + \sum_{i=t_1}^{N} h_i > c_{t_2} \quad \forall t_1 < t_2$ in order to prevent speculative orders due to this inventory clearing. Suppose this condition is not satisfied for periods t_1 and t_2. Then consider the replenishment schedule $r_1 = 1$, $r_2 = t_1$, $r_3 = t_2$ where $m = 3$. Given that the condition does not hold, if a unit produced at time t_1 can be kept until the end of horizon, selling the item will bring benefits. Then, producing as many items as possible will be the most cost effective way. Even though the problem is capacitated, the lot size decision would not change in this manner. This will cause speculative setups and is very unlikely in practise. Therefore these aspects of the problem would not be considered in our analysis.

CHAPTER 3

OPTIMAL SOLUTION FOR PROBLEM P

In this chapter, we will introduce our analysis on finding the optimal solution to Problem P. Since our method is built on evaluation of every single production schedule, we start analyzing for a given production schedule. We will than explain how we compute the optimal base stock levels for a given production schedule. We will last discuss our lower bounds constructed on Problem P.

3.1 Characterisation of the Optimal Policy

Consider Problem P under a given production schedule $(m, \mathbf{r})$. Note that the setup periods have already been selected, therefore the aim is to find the optimal inventory levels, (y_i), just after production. The following backward dynamic program with m stages where a stage is a production cycle solves our problem. The recursive function is given by

$$f_{m+1}(I_{m+1}) = -c_{r_m} I_{m+1} \tag{3.1}$$

$$\begin{aligned} f_i(I_i) &= \min_{x_i : o_{r_i} \geq x_i \geq u_{r_i}} \left\{ A_{r_i} + c_{r_i}(x_i) + \mathcal{L}_i(x_i + I_i) + \mathsf{E}\left[f_{i+1}(I_i + x_i - \sum_{j=r_i}^{r_{i+1}-1} D_j) \right] \right\} \\ &= \min_{y_i : I_i + o_{r_i} \geq y_i \geq I_i + u_{r_i}} \left\{ A_{r_i} + c_{r_i}(y_i - I_i) + \mathcal{L}_i(y_i) + \mathsf{E}\left[f_{i+1}(y_i - \sum_{j=r_i}^{r_{i+1}-1} D_j) \right] \right\} \\ &= \min_{y_i : I_i + o_{r_i} \geq y_i \geq I_i + u_{r_i}} G_i(y_i) \end{aligned} \tag{3.2}$$

for $i = 1, \ldots, m$ where $f_i(I_i)$ is the minimum expected cost of the system over production cycles $i, i+1, \ldots, m$ (i.e. over periods $r_i, r_i + 1, \ldots, N$) given the net inventory level at the beginning of production cycle i, I_i, and

$$G_i(y_i) := A_{r_i} + c_{r_i}(y_i - I_i) + \mathcal{L}_i(y_i) + \mathsf{E}\Big[f_{i+1}\Big(y_i - \sum_{j=r_i}^{r_{i+1}-1} D_j \Big) \Big]. \tag{3.3}$$

The solution for $f_i(I_i)$ gives the optimal inventory level just after production, denoted by $y_i^*(I_i)$, for a given production schedule $(m, \mathbf{r})$ and starting net inventory level I_i. Starting from $f_m(I_m)$ and solving for $f_1(I_1)$ by backward recursion gives $\mathbf{y}^* = (y_1^*(I_1), \ldots, y_m^*(I_m))$ for a given production schedule $(m, \mathbf{r})$ and opening net inventory level I_1.

Since we have upper and lower limits on the lot sizes in our problem, a modified base stock policy with base stock levels $\mathbf{S} = (\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_m})$ will yield inventory levels just after ordering in cycle i such that;

$$y_i(I_i) = \begin{cases} I_i + u_i & \text{if} \quad S_i - I_i < u_{r_i} \\ I_i + o_i & \text{if} \quad S_i - I_i > o_{r_i} \\ S_i & \text{otherwise.} \end{cases} \tag{3.4}$$

If S_i is attainable considering the limits, new inventory level will be S_i. If the upper limit for the lot size is not sufficient to reach S_i, a lot of u_{r_i} will be produced to realize as much cost reduction as possible. If producing the minimum lot size already causes an inventory position more than S_i, only a lot of o_{r_i} is produced to stay away from incurring any further costs.

Özen et al. (2012) proves that optimal policy for Problem P under a given production schedule $(m, \mathbf{r})$ is a modified base stock policy by the following theorem.

Theorem 3.1 *Given a replenishment schedule* $(m, \mathbf{r})$*, it is optimal to follow a modified stock policy with* $\mathbf{S}^*$ *where*

$$S_i^* = \arg\min\{G_i(y_i)\}, \quad i = 1, 2, \ldots, m.$$

Proof.

Let us take an inductive approach. First, note that $\mathcal{L}_i(y_i)$ is the summation of $r_{i+1} - r_i$ newsvendor cost functions, which are known to be convex, for $i = 1, 2, \ldots, m$. Since summation preserves convexity, we can conclude that $\mathcal{L}_i(y_i)$ is convex for all $i = 1, 2, \ldots, m$.

$$\begin{aligned} B(I_{m+1}) &= -c_m I_{m+1} \\ \mathsf{E}[B(I_{m+1})] &= \mathsf{E}[-c_m I_{m+1}] \\ &= \mathsf{E}[-c_m(y_m - D_{r_m} - D_{r_m+1} - \ldots - D_{r_{m+1}-1})] \\ &= -c_m(y_m - \mathsf{E}[D_{r_m} + D_{r_m+1} + \ldots + D_{r_{m+1}-1}]) \\ G_m(y_m) &= A_{r_m} + c_{r_m}(y_m - I_m) + \mathcal{L}_m(y_m) - c_m(y_m - \mathsf{E}[D_{r_m} + D_{r_m+1} + \ldots + D_{r_{m+1}-1}]). \end{aligned}$$

which is a convex function in $y_m \in \mathbb{R}$. Let S_m^* be the global minimizer of this function. Then it is optimal to follow a modified base stock policy such that:

$$y_m(I_m) = \begin{cases} I_m + u_{r_m} & \text{if} \quad S_m^* - I_m < u_{r_m} \\ I_m + o_{r_m} & \text{if} \quad S_m^* - I_m > o_{r_m} \\ S_m^* & \text{otherwise.} \end{cases}$$

This proves our argument in the base case.
Second, assume that $G_i(.)$ is convex and let $g_i(I_i)$ be the expected cost function from cycle i and onwards. Given that the replenishment policy for cycle i is a modified base stock policy with level S_i^*, where S_i^* minimizes $G_i(y_i)$. Then,

$$g_i(I_i) = \begin{cases} G_i(I_i + u_{r_i}) & \text{if} \quad I_i + u_{r_i} > S_i^* \\ G_i(I_i + o_{r_i}) & \text{if} \quad I_i + o_{r_i} < S_i^* \\ G_i(S_i^*) & \text{otherwise.} \end{cases}$$

Assuming that $G_i(\cdot)$ is a convex function, it is easy to check that $g_i(\cdot)$ is convex as well.

$$G_{i-1}(y_{i-1}) = A_{r_{i-1}} + c_{r_{i-1}}(y_{i-1} - I_{i-1}) + \mathcal{L}_{i-1}(y_{i-1}) + \mathsf{E}\Big[f_i\Big(y_{i-1} - \sum_{j=r_{i-1}}^{r_i-1} D_j\Big)\Big]$$

$$G_{i-1}(y_{i-1}) = A_{r_{i-1}} + c_{r_{i-1}}(y_{i-1} - I_{i-1}) + \mathcal{L}_{i-1}(y_{i-1}) + \mathsf{E}\Big[g_i\Big(y_{i-1} - \sum_{j=r_{i-1}}^{r_i-1} D_j\Big)\Big]$$

$g_i(I_i)$ is convex in $I_i \in \mathbb{R}$ where $I_i = y_{i-1} - \sum_{j=r_{i-1}}^{r_i-1} D_j$. Then g_i is convex in y_{i-1} and $\mathsf{E}\Big[g_i(I_i)\Big]$ is convex combination of function g_i's. Therefore, $\mathsf{E}\Big[g_i(I_i)\Big]$ is also convex in y_{i-1}. Then, G_{i-1} is convex in y_{i-1}, and modified base stock policy is optimal. And this concludes the inductive argument. ■

Since the optimal ordering policy for a given production schedule is in the form of a modified base stock policy, the following corollary is immediate.

Corollary 3.2 *There exists an optimal "production policy"* $(m^*, \mathbf{r}^*)$ *and* $\mathbf{S}^*$ *for Problem P, where* $(m^*, \mathbf{r}^*)$ *is the optimal "production schedule" and the optimal "production policy" is a modified base stock policy with* $\mathbf{S}^*$.

3.2 Computing Optimal Order Levels

After identifying type of the policy, finding the optimal policy parameters becomes straightforward. In order to solve the Problem P for a given production schedule $(m, \mathbf{r})$, we have to find optimal basestock levels, S_i^* such that;

$$S_i^* = \arg\min_{y_i}\{G_i(y_i)\}, \quad i = 1, 2, \ldots, m.$$

Since $G_i(.)$ is a convex function in y_i, a simple binary search on the interval $[0, \infty)$ is necessary to find S_i^* value for all $i = 1, \ldots, m$. In order to speed up our calculations, however, we did not scan the entire interval from 0 to ∞. We first solve the following problem that cycle i along creates to find $\bar{S}_i^*$.

$$\bar{S}_i^* = \arg\min_{y_i}\{(y_i - I_i)^+ c_{r_i} + \mathcal{L}(y_i)\} \quad i = \{1, \ldots, m-1\}$$

Since the end-of-cycle action in cycle m is different form the other cycles, we calculated S_m^* directly by

$$\bar{S}_m^* = S_m^* = \arg\min_{y_m}\{G_m(y_m)\}$$

Since we are solving a backward DP, being at stage i means that we have already solved for S_{i+1}^*. Then we start with interval $(\min\{\bar{S}_i^*, S_{i+1}^*\}, \max\{\bar{S}_i^*, S_{i+1}^*\})$. Then we check derivatives of the function $G_i(.)$ at these extreme points. If the lower limit is a decreasing point and the upper limit is an increasing point, we search through this interval. Otherwise, we modify the interval bounds until this condition is satisfied and perform the search afterwards. The minimizer is where the derivative is 0 for continuous cases, and where the derivative turn nonnegative from negative for discrete cases. We can solve the entire problem by starting from cycle m to cycle 1 by the procedure explained above.

3.3 Bounds and Partial Characterizations

Halman et al. (2009) shows that the problem of calculating the optimal base stock levels in a single-stage production system facing non-stationary stochastic demand with linear costs using standard DP approach is NP-hard. However, if the demand distributions are regenerative as in the case where demand is Normal or Poison, the convolutions can be computed efficiently. Then, solving the standard DP is a polynomial algorithm.

Even though finding the optimal policy parameters for a given production schedule takes polynomial-time, every possible production schedule should be evaluated individually in order to find the optimal solution. This results in exponentially growing solution time with respect to the length of the planning horizon. However, if we can find strong lower bounds for a given production schedule, we can eliminate the effort to solve certain production schedules for optimality, knowing that they will not be optimal before the attempt to solve for optimal parameters. Next, we proceed with defining lower bounds to our problem P.

3.3.1 Lower Bound 1

Özen et al. (2012) proves that sum of all fixed costs to be incurred in a certain production schedule provides a lower bound (LB_1) to the Problem P. The following propositions also give insights about this lower bound.

Proposition 3.3 *For the unconstraint problem $u = 0$ and $o \to \infty$, given a replenishment schedule $(m, \mathbf{r})$, it is optimal to follow base stock policy with $\mathbf{S}^*$ where S_i^*'s are finite if the following condition holds:*

$$c_{r_i} + \sum_{j=r_i}^{N} h_j \geq c_{r_m} \quad for\ 1 \leq i < m \tag{3.5}$$

Proof.

Consider the newsvendor function for a given cycle.

$$\mathcal{L}_i(y_i) = \mathsf{E}\left[\sum_{k=r_i}^{r_{i+1}-1}\left(h_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^+ + b_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^-\right)\right] \geq 0$$

If y_i is sufficiently large, then $y_i - \sum_{j=r_i}^{k} D_j$ is always positive. Then, penalty cost is never incurred. In this case marginal cost of ordering one more unit is $c_{r_i} + \sum_{j=r_i}^{N} h_j$ since this unit will be kept in stock throughout the remainder of the horizon. When the planning horizon ends, it will be sold with price c_{r_m}. If Equation 3.5 does not hold, then buying this unit results in reduction in cost i.e. $y_i^* = \infty$. Else y_i^* is finite. ■

Proposition 3.4 (LB_1) *If Equation (3.5) holds for given a replenishment schedule $(m, \mathbf{r})$, $\sum_{i=1}^{N} A_i\delta_i$ is a lower bound to Problem P.*

Proof.

If Equation (3.5) holds, then there is no revenue of buying one more unit in a given replenishment, which means all cycle costs are positive. Then, $\sum_{i=1}^{N} A_i\delta_i$ constitutes a lower bound for a schedule where $u_t = 0$ and $o_t = \infty$. Then it is also a lower bound for the case $u_t > 0$ and $o_t < \infty$. ■

3.3.2 Lower Bound 2

If the constraints on the lot sizes are ignored, and it is assumed that every unit on hand (backlogged) can be sold (cleared) at the price unit production cost of the next cycle; problem for the given production schedule $(m, \mathbf{r})$ can be decomposed into m separate minimization problems. Özen et al. (2012) proves that sum of the optimal costs of these m problems constitute a lower bound (LB_2) for Problem P.

Proposition 3.5 *Given a replenishment schedule* $(m, \mathbf{r})$*, the following problem has a bounded optimal solution.*

$$\begin{aligned}\sum_{k=1}^{N} A_k\delta_k \quad & + \min_{y_1:y_1\geq I_1}\Big\{c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\dots,D_{r_2-1}}\big[-c_{r_2}I_2\big]\Big\}\\ & + \min_{y_2:y_2\geq 0}\Big\{c_{r_2}(y_2) + \mathcal{L}_2(y_2) + \mathsf{E}_{D_{r_2},\dots,D_{r_3-1}}\big[-c_{r_3}I_3\big]\Big\}\\ & + \cdots + \min_{y_m:y_m\geq 0}\Big\{c_{r_m}(y_m) + \mathcal{L}_m(y_m) + \mathsf{E}_{D_{r_m},\dots,D_{r_{m+1}-1}}\big[B(I_{m+1})\big]\Big\}\end{aligned} \tag{3.6}$$

if the following condition holds:

$$c_{r_i} + \sum_{j=r_i}^{r_{i+1}-1} h_j \geq c_{r_{i+1}} \quad for\ 1 \leq i < m \tag{3.7}$$

Proof.

If Equation (3.7) does not hold for a cycle i, consider the subminimization problem;

$$\min_{y_i:y_i\geq 0}\Big\{c_{r_i}(y_i) + \mathcal{L}_i(y_i) + \mathsf{E}_{D_{r_i},\dots,D_{r_{i+1}-1}}\big[-c_{i+1}I_{i+1}\big]\Big\}$$

For sufficiently large y_i, $\mathcal{L}_i(y_i)$ converges to $\mathsf{E}\Big[\sum_{k=r_i}^{r_{i+1}-1}\Big(h_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^+\Big)\Big]$, the marginal cost of buying one more unit is $c_{r_i} + \sum_{k=r_i}^{N} h_k$. Since Equation 3.7 does not hold, buying one more unit decreases the cost in this subproblem. Since all problems are seperate, $y_i^* = \infty$. ■

Proposition 3.6 (LB_2) *Given a replenishment schedule* $(m, \mathbf{r})$*, the optimal solution to problem (3.6) is a lower bound to Problem P.*

Proof.

$$\begin{aligned}\sum_{k=1}^{N} A_k\delta_k \quad & + \min_{y_1:y_1\geq I_1}\Big\{c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\dots,D_{r_2-1}}\big[-c_{r_2}I_2\big]\Big\}\\ & + \min_{y_2:y_2\geq 0}\Big\{c_{r_2}(y_2) + \mathcal{L}_2(y_2) + \mathsf{E}_{D_{r_2},\dots,D_{r_3-1}}\big[-c_3I_3\big]\Big\}\\ & + \cdots + \min_{y_m:y_m\geq 0}\Big\{c_{r_m}(y_m) + \mathcal{L}_m(y_m) + \mathsf{E}_{D_{r_m},\dots,D_{r_{m+1}-1}}\big[B(I_{m+1})\big]\Big\}\end{aligned}$$

since y_i's are independent, it is equivalent to solve the problem;

$$\begin{aligned} = \quad & \sum_{k=1}^{N} A_k\delta_k + \min_{y_1:y_1\geq I_1}\Big\{c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\dots,D_{r_2-1}}\big[-c_{r_2}I_2 + \min_{y_2:y_2\geq 0}\Big\{c_{r_2}(y_2) + \mathcal{L}_2(y_2) +\\ & \mathsf{E}_{D_{r_2},\dots,D_{r_3-1}}\big[-c_3I_3 + \cdots + \min_{y_m:y_m\geq 0}\Big\{c_{r_m}(y_m) + \mathcal{L}_m(y_m) + \mathsf{E}_{D_{r_m},\dots,D_{r_{m+1}-1}}\big[B(I_{m+1})\big]\Big\}\cdots\big]\Big\}\big]\Big\}\\ = \quad & \sum_{k=1}^{N} A_k\delta_k + \min_{y_1:y_1\geq I_1}\Bigg\{c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\dots,D_{r_2-1}}\Big[\min_{y_2:y_2\geq 0}\Big\{c_{r_2}(y_2 - I_2) + \mathcal{L}_2(y_2) +\\ & \mathsf{E}_{D_{r_2},\dots,D_{r_3-1}}\Big[\cdots\big[\min_{y_m:y_m\geq 0}\{c_{r_m}(y_m - I_m) + \mathcal{L}_m(y_m) + \mathsf{E}_{D_{r_m},\dots,D_{r_{m+1}-1}}\big[B(I_{m+1})\big]\}\big]\cdots\Big]\Big\}\Big]\Bigg\}\end{aligned}$$

$$\leq \sum_{k=1}^{N} A_k\delta_k + \min_{y_1:y_1\geq I_1}\Bigg\{c_{r_1}(y_1-I_1)+\mathcal{L}_1(y_1)+\mathsf{E}_{D_{r_1},\ldots,D_{r_2-1}}\Big[\min_{y_2:y_2\geq I_2}\{c_{r_2}(y_2-I_2)+\mathcal{L}_2(y_2)+$$

$$\mathsf{E}_{D_{r_2},\ldots,D_{r_3-1}}\Big[\cdots[\min_{y_m:y_m\geq I_m}\{c_{r_m}(y_m-I_m)+\mathcal{L}_m(y_m)+\mathsf{E}_{D_{r_m},\ldots,D_{r_{m+1}-1}}[B(I_{m+1})]\}]\cdots\Big]\Big\}\Big]\Bigg\}$$

$$\leq \sum_{k=1}^{N} A_k\delta_k + \min_{y_1:I_1+o_{r_1}\geq y_1\geq I_1+u_{r_1}}\Bigg\{c_{r_1}(y_1-I_1)+\mathcal{L}_1(y_1)+\mathsf{E}_{D_{r_1},\ldots,D_{r_2-1}}\Big[\min_{y_2:I_2+o_{r_2}\geq y_2\geq I_2+u_{r_2}}\{c_{r_2}(y_2-I_2)$$

$$+\mathcal{L}_2(y_2)+\mathsf{E}_{D_{r_2},\ldots,D_{r_3-1}}\Big[\cdots[\min_{y_m:I_m+o_{r_m}\geq y_m\geq I_m+u_{r_m}}\{c_{r_m}(y_m-I_m)+\mathcal{L}_m(y_m)+$$

$$\mathsf{E}_{D_{r_m},\ldots,D_{r_{m+1}-1}}[B(I_{m+1})]\}]\cdots\Big]\Big\}\Big]\Bigg\}$$

■

3.3.3 Lower Bound 3

LB_1 and LB_2 work well for the cases where $c_t = 0$, $o_t = 0$, and $u_t = \infty$ which are not valid in the Problem P that we are trying to solve. Lot sizes (x_i) depend on how the stochastic demand is realized, therefore cannot be known for certainty. Incurring variable production cost ($c_t > 0$) weakens the strength of LB_1. Especially, capacity and lower limit on lot sizes ($o_t > 0$ and $u_t < \infty$) weakens the quality of these lower bounds both which do not take the relations of end of cycle inventories into account. This reason makes it necessary to develop a new lower bound.

Proposition 3.7 *Let $g_i(I_i)$ be a piecewise linear, continuous, and convex function whose line segments are tangent to $\hat{f}_i(I_i)$ which is a convex function. Then, $g_i(I_i) \leq \hat{f}_i(I_i)$ $\forall I_i$.*

Proof.

Since $\hat{f}_i(I_i)$ is convex, it is immediately correct. ■

While solving the DP for a given production schedule, optimal base stock level for current cycle is found using the cost-to-go values as a function of end of cycle inventory. Different demand realizations result in different end of cycle inventories, making it necessary to calculate cost-to-go function for multiple points which is main source of computational load. While solving the DP, instead of using true cost-to-go values, we can make use of a piecewise linear lower estimation, $\hat{f}_i(I_i)$, which ease the computational burden. This leads us to the following proposition.

Proposition 3.8 *Given a replenishment schedule $(m, \mathbf{r})$, solving the following recursive equations for $\hat{f}_1(I_1)$ provides a lower bound (LB_3) to Problem P.*

$$\hat{f}_{m+1}(I_{m+1}) = g_{m+1}(I_{m+1}) = -c_{r_m}I_{m+1}$$

$$\hat{f}_i(I_i) = \min_{y_i:I_i+o_{r_i}\geq y_i\geq I_i+u_{r_i}}\left\{A_{r_i}+c_{r_i}(y_i-I_i)+\mathcal{L}_i(y_i)+E\left[g_{i+1}(y_i-\sum_{j=r_i}^{r_{i+1}-1}D_j)\right]\right\}$$

Proof.

Let us take an inductive approach. It is given that;

$$g_{m+1}(I_{m+1}) = \hat{f}_{m+1}(I_{m+1}) = f_{m+1}(I_{m+1})$$

which proves the base case. Assume

$$\hat{f}_i(I_i) = \min_{y_i: I_i+o_{r_i}\geq y_i \geq I_i+u_{r_i}} \left\{ A_{r_i} + c_{r_i}(y_i - I_i) + \mathcal{L}_i(y_i) + \mathsf{E}\left[g_{i+1}(y_i - \sum_{j=r_i}^{r_{i+1}-1} D_j) \right] \right\}$$

is convex and;

$$\hat{f}_i(I_i) \leq f_i(I_i) = \min_{y_i: I_i+o_{r_i}\geq y_i \geq I_i+u_{r_i}} \left\{ A_{r_i} + c_{r_i}(y_i - I_i) + \mathcal{L}_i(y_i) + \mathsf{E}\left[f_{i+1}(y_i - \sum_{j=r_i}^{r_{i+1}-1} D_j) \right] \right\}$$

$$\hat{f}_{i-1}(I_{i-1}) = \min_{\substack{y_{i-1}: I_{i-1}+o_{r_{i-1}}\geq y_{i-1} \\ y_{i-1}\geq I_{i-1}+u_{r_{i-1}}}} \left\{ A_{r_{i-1}} + c_{r_{i-1}}(y_i - I_{i-1}) + \mathcal{L}_{i-1}(y_{i-1}) + \mathsf{E}\left[g_i(y_{i-1} - \sum_{j=r_{i-1}}^{r_i-1} D_j) \right] \right\}$$

is convex since $g_i(.)$ is a convex function. Then, also;

$$\begin{aligned} \hat{f}_{i-1}(I_{i-1}) &\leq f_{i-1}(I_{i-1}) \\ &= \min_{\substack{y_{i-1}: I_{i-1}+o_{r_{i-1}}\geq y_{i-1} \\ y_{i-1}\geq I_{i-1}+u_{r_{i-1}}}} \left\{ A_{r_{i-1}} + c_{r_{i-1}}(y_{i-1} - I_{i-1}) + \mathcal{L}_{i-1}(y_{i-1}) + \mathsf{E}\left[f_i(y_{i-1} - \sum_{j=r_{i-1}}^{r_i-1} D_j) \right] \right\} \end{aligned}$$

since $g_i(I_i) \leq \hat{f}_i(I_i) \leq f_i(I_i)$ which concludes the induction argument. ■

LB_3 is more complicated compared to the other lower bounds in terms of implication. In order to implement LB_3, we have to decide on with how many line segments we will approximate the cost functions. It is trivial that having more line segments will increase the accuracy; however, it will increase the computational effort at the same time. What is basically being done while computing LB_3 for the given production schedule $(m, \mathbf{r})$ is the following:

- Find the optimal base stock level for cycle-m.
- Derive the optimal cost function in terms of starting inventory level which is a convex function.
- Approximate this function with a number of line segments.
- Replace the recursive equations with this function and solve the DP.

Note that finding LB_3 requires solving an additional DP which is computationally expensive. Although it can be very strong for the problems with tight capacity values, its accuracy depends on the number of line segments to be fit for approximating the cost functions. Intersection points of tangent lines to the cost functions also affect the quality of the LB_3. Nevertheless, as it will be discussed in Chapter 6, it reduces the overall solution time for finding the optimal solution.

3.4 Computing the Parameters of an Optimal Policy

We will use our analysis so far for finding the optimal solution for Problem P. We assume that there is a setup at the beginning of the planning horizon, ie., $\delta_1 = 1$. Therefore, there are 2^{N-1} different production schedules to be evaluated, and the optimal solution will be one of these. By Corollary 3.2, the optimal ordering policy is a modified base stock policy which are calculated by backward recursion and binary search for production schedule $(m, \mathbf{r})$. Optimizing the base stock levels for all possible production schedules and comparing the corresponding expected costs yields the optimal solution for Problem P. In this approach, lower bounds LB_1, LB_2 and LB_3 can be utilized to elevate the computational burden of an enumerative search.

Before determining the optimal expected cost of a given schedule $(m, \mathbf{r})$, three lower bounds are considered: LB_1, LB_2, and LB_3 which are given in the ascending order of computational burden. If the best-so-far cost found is less than LB_1, then there is no need to calculate the optimal base stock levels; this production schedule can be eliminated. Otherwise, LB_2 is calculated and compared against the cost best so far. If LB_2 is larger, then this production schedule can be eliminated. However, if LB_2 is lower than the best-so-far cost, then the LB_3 is computed which has the closest effort to solving for optimal base stock levels. If LB_3 is larger than the cost best so far, there is no need to proceed with the current schedule. If the cost calculated with the corresponding optimal base stock levels for the current production schedule is lower than the cost best so far, then the best-so-far cost and policy parameters are updated, and the search moves to the next production schedule. Continuing in this manner all 2^{N-1} different production schedules are considered. The search gives the optimal production schedule, and the corresponding optimal base stock levels and the cost. Note that the optimal solution approach described above is an exhaustive search where the computational effort needed grows exponentially in the size of the planning horizon, N.

CHAPTER 4

DYNAMIC PROGRAMMING BASED HEURISTICS

As we discuss in detail in Chapter 6, the numerical approach outlined in Chapter 3, which utilizes the lower bounds, decreases the computational time to search for an optimal policy drastically. Nevertheless, enumerating all feasible ordering patterns is an exponential search and is not suitable for problems with longer planning horizons (large N). Our numerical experiments demonstrate that even for $N = 18$, the computation times can be in the order of days. Due to this issue, we will develop some heuristics for Problem P. While the first type of heuristics are DP-based approaches that approximates the true cost-to-go function, and the second type of heuristics are based on a greedy search among possible replenishment schedules of Problem P.

4.1 Approximation Heuristic (AH)

The main obstacle in finding an optimal solution for Problem P emanates from the combinatorial nature of the outmost minimization problem. Therefore, Özen et al. (2012) introduces a heuristic based on a backward recursion that selects the replenishment schedule and the corresponding base stock levels simultaneously. Define

- $\hat{y}_t$: inventory level at the beginning of period t just after order
- $\hat{a}_t$: number of periods covered by the lot produced in t

In period t, only the current period's decisions, $\hat{a}_t$ and $\hat{y}_t$, are optimized by solving the following minimization problem:

$$\min_{\substack{\hat{y}_t:\hat{y}_t\geq 0\\ \hat{a}_t:N-t+1\geq\hat{a}_t\geq 1}} \Big\{A_t + \hat{y}_t c_t + \hat{\mathcal{L}}_t(\hat{a}_t,\hat{y}_t) + \mathsf{E}\Big[\varphi_{t+\hat{a}_t}(\hat{y}_t - \sum_{j=t}^{t+\hat{a}_t-1} D_j)\Big]\Big\}, \tag{4.1}$$

where

$$\hat{\mathcal{L}}_t(\hat{a}_t,\hat{y}_t) = \mathsf{E}\Big[\hat{y}_t - \sum_{k=t}^{t+\hat{a}_t-1}\Big(h_k\Big(\sum_{j=t}^{k} D_j\Big)^+ + b_k\Big(\hat{y}_t - \sum_{j=t}^{k} D_j\Big)^-\Big)\Big] \tag{4.2}$$

Even though the decisions for period t are taken in a one step, the cost impact of the decisions is taken into account through the function $\varphi_l(.)$ that is defined for any $N \geq l > t$ as

$$\varphi_l(q) = A_l + c_l(p_l(q,\hat{S}^*_l) - q) + \hat{\mathcal{L}}_l(\hat{a}^*_l, p_l(q,\hat{S}^*_l)) + \mathsf{E}\Big[\varphi_{l+\hat{a}^*_l}(p_l(q,\hat{S}^*_l) - \sum_{j=l}^{l+\hat{a}^*_l-1} D_j)\Big]$$

where q represents the initial inventory for period l while the DP is being solved for a cycle ending at period l. Since there is maximum and minimum lot sizes, $p_l(q,\hat{S}^*_l)$ is the function determines the inventory position after production with the following definition:

$$p_l(q_1, q_2) \quad = \quad \begin{cases} q_1 + u_l & \text{if } q_2 - q_1 \leq u_l \\ q_1 + o_l & \text{if } q_2 - q_1 \geq o_l \\ q_2 & \text{otherwise} \end{cases}$$

$$(\hat{a}_l^*, \hat{S}_l^*) \quad = \quad \arg\min_{\hat{a}_l, \hat{y}_l} \Big\{ A_l + \hat{y}_l c_l + \hat{\mathcal{L}}_l(\hat{a}_l, \hat{y}_l) + \mathsf{E}\Big[\varphi_{l+\hat{a}_l}(\hat{y}_l - \sum_{j=l}^{l+\hat{a}_l-1} D_j)\Big]\Big\}$$

Note that $(\hat{a}_l^*, \hat{S}_l^*)$ for $l > t$ is already known when the optimization problem (4.1) is solved for period t because the proposed heuristic works in a backward fashion solving (4.1) for $t = N, \ldots, 1$. The results of Section **??** guided the development of the heuristic in the sense that what we have to solve for is the number of periods in the cycles and corresponding basestock levels. The optimal ordering policy has been shown as a base stock policy when the production schedule is fixed. AH searches for these deciisons as follows. The minimization problem in (4.1) searches over all possible $\hat{a}_t$ values: $1 < \hat{a}_t < N - t + 1$. If $\hat{a}_t$ is set, the production schedule until the end of the planning horizon is fixed such that there is an setup placed in periods t, $t + \hat{a}_t$, $t + \hat{a}_t + \hat{a}^*_{t+\hat{a}_t}$, $t + \hat{a}_t + \hat{a}^*_{t+\hat{a}_t} + \hat{a}^*_{t+\hat{a}_t+\hat{a}^*_{t+\hat{a}_t}}$, and so on. Let the first setup period be the first period of some cycle i, the second setup period be the beginning of cycle $i + 1, \ldots,$ until the last setup period in the sequence corresponds to the first period of cycle m:

$$r_i = t,\ r_{i+1} = t + \hat{a}_t,\ r_{i+2} = t + \hat{a}_t + \hat{a}^*_{t+\hat{a}_t},$$
$$r_{i+3} = t + \hat{a}_t + \hat{a}^*_{t+\hat{a}_t} + \hat{a}^*_{t+\hat{a}_t+\hat{a}^*_{t+\hat{a}_t}},$$

What makes AH an approximation is that it assumes a zero initial inventory level while solving the problem in (4.1) for period t, i.e., $\hat{y}_t > 0$, and uses this solution while solving the problem for $t - 1$. Hence, AH ignores the impact of the actions on the initial inventory levels, that would be taken in the previous periods. Since, Özen et al. (2012) ignores the impact of unit production cost and lot size restrictions, this fact does not produce in accurate results. Nevertheless, AH calculates the true costs because the inventory balance equations are respected in the recursions.

In AH, since the periods to cover with the lot size should be determined simultaneously with the size of the lot, as much calculations for $\hat{S}_l^*$ as periods left until the planning horizon need to be done. In that sense, when deciding for last period there is 1 calculation for $\hat{S}_N^*$ to be done. For the period before there are 2 for $\hat{S}_{N-1}^*$, and in general for period l, there are $N - l + 1$ basic calculations which yields a complexity of $O(N^2)$ in terms of number of cycles but the effect of the convolution calculations are excluded.

4.2 Modified Approximation Heuristic I (AH I)

Since the variable cost of ordering and the limits on order quantities are present, the assumption that the initial inventory of each period will be zero no longer produces accurate results. However, getting an estimate of the opening inventories requires to solve a problem for the periods prior to the current period. In order to use the information provided by all periods, we can consider to solve nested AH's simultaneously which will still eliminate the exponential growth of the problem in number of periods. Suppose we are solving AH for period t. We are to pick between $\hat{a}_t$ values in order to optimize the decision for period t. Note that evaluating a certain $\hat{a}_t$ value also requires to solve for corresponding basestock level for the cycle formed. While choosing optimal parameters, AH ignores the impact of initial inventory levels. In order to avoid that, we can solve another AH which will add the effect of initial inventories to the model. In order to achieve this, let us define following additional decision variables:

- $\bar{y}_i$: inventory level at the beginning of period i just after production such that period t is the ending period for the subproblem and $\hat{a}_t$ is the number of periods to cover from that period.
- $\bar{a}_i$: number of periods covered by the setup scheduled in i just after the production such that period t is the ending period for the subproblem and $\hat{a}_t$ is the number of periods to cover from that period.

Then, let us solve the following problem.

$$\min_{\substack{\hat{y}_t:\hat{y}_t\geq 0\\ \hat{a}_t:N-t+1\geq\hat{a}_t\geq 1}} \{Q_1^t(\hat{a}_t,\hat{y}_t)\} \quad \forall t \tag{4.3}$$

where

$$Q_i^t(\hat{a}_t,\hat{y}_t) = \min_{\substack{\bar{y}_i:\bar{y}_i\geq 0\\ \bar{a}_i:t-i+1\geq\bar{a}_i\geq 1}} \Big\{A_i + \bar{y}_i c_i + \hat{\mathcal{L}}_i(\bar{a}_i,\bar{y}_i) + \mathsf{E}\Big[\phi_{i+\bar{a}_i}(\bar{y}_i - \sum_{j=i}^{i+\bar{a}_i-1} D_j)\Big]\Big\} \tag{4.4}$$

The definition of $\phi_l(.)$ is given as follows.

$$\phi_l(q) = A_l + (p_l(q,\bar{y}_l^*) - q)c_l + \hat{\mathcal{L}}_l(\bar{a}_l^*, p_l(q,\bar{y}_l^*)) + \mathsf{E}\Big[\phi_{l+\bar{a}_l^*}(p_l(q,\bar{y}_l^*) - \sum_{j=l}^{l+\bar{a}_l^*-1} D_j)\Big] \quad \forall l < t$$

$\phi_l(.)$ represents the cost function of the subproblem where period t is the ending period. At every step of the recursion, we solve for optimal number of periods to cover from period l, $\bar{a}_l^*$, and a corresponding basestock level, ($\bar{y}_l^*$) while period t is the ending period of the subproblem and the action of covering $\hat{a}_t$ many periods is the boundary condition for the subproblem.

$$(\bar{a}_l^*,\bar{y}_l^*) = \arg\min_{\bar{y}_l,\bar{a}_l} \Big\{A_l + \bar{y}_l c_l + \hat{\mathcal{L}}_l(\bar{a}_l,\bar{y}_l) + \mathsf{E}\Big[\phi_{l+\bar{a}_l}(\bar{y}_l - \sum_{j=l}^{l+\bar{a}_l-1} D_j)\Big]\Big\}$$

The variables $\hat{a}_t$ and $\hat{y}_t^*$ do not immediately appear in Equation 4.4. However, where they are included is the end of subproblem solved which is the period t. Below equation points how these variables contribute to the solution for Q_i^t.

$$\phi_t(q) = A_t + (p_t(q,\hat{y}_t^*) - q)c_t + \hat{\mathcal{L}}_t(\hat{a}_t, p) + \mathsf{E}\Big[\varphi_{t+\hat{a}_t}(p_t(q,\hat{y}_t^*) - \sum_{j=t}^{t+\hat{a}_t-1} D_j)\Big]$$

And $\varphi_l(.)$ represents the cost function of the AH to be solved from period t to period N.

$$(\hat{a}_l^*,\hat{S}_l^*) = \arg\min_{\hat{a}_l,\hat{y}_l} \{Q_1(\hat{a}_l,\hat{y}_l)\}$$

Solving (4.3) for optimality has a very similar procedure to solving (4.1). Solution for (4.3) is obtained backward recursion for $t = N,\dots,1$; however, while determining $(\hat{a}_t^*,\hat{y}_t^*)$, cost-to-go function from period t is not considered. Instead, a new optimization problem (4.4) is constructed for this decision such that the action to be taken in period t is the boundary condition. And the horizon of $t-1$ periods is solved with AH. Proceeding this way, we took the impact of initial inventories into consideration which we missed in AH.

Of course, this solution approach increases computational complexity. Suppose we are seeking a solution for a planning horizon of N periods. For any period t while solving AH I, there are $(N-t+1)$ possible options to be considered for determining the number of periods to be cover from period t. Additionally, there is an AH to be solved for each of these possible actions. Note that there should be $(t(t-1)/2)$ subproblems are evaluated for solving one AH. This makes $(N-t+1)(t(t-1)/2)$ subproblems for one period t in AH I. Then the computational complexity of AH I is $O(N^5)$.

4.3 Modified Approximation Heuristic II (AH II)

Since it increases the order of the complexity, instead of solving an additional problem for all periods prior to current period, we can consider to work for fixed number of periods in order to keep our computational time low. If we solve this smaller additional problem for optimality, it can still serve for our intention. Let n be the fixed number of periods before period t to extend our AH. Let us define the following decision variables:

- $\bar{n} = \begin{cases} n & \text{if } n \leq t \\ t & \text{otherwise} \end{cases}$
- $\bar{m}$: number of setups in the subproblem
- $\bar{\mathbf{r}} = \{\bar{r}_1, \ldots, \bar{r}_{\bar{n}}\}$: the schedule of setups for the subproblem where $\bar{r}_1 = t - \bar{n}$
- $\bar{y}_{\bar{r}_i}$: base stock level of production cycle i of subproblem
- $\bar{\delta}_{\bar{r}_i} = 1, \forall i = 1, \ldots, \bar{m}$

Let us consider to solve the following problem.

$$\min_{\substack{\hat{y}_t : \hat{y}_t \geq 0 \\ \hat{a}_t : N-t+1 \geq \hat{a}_t \geq 1}} \{Q^t(\hat{a}_t, \hat{y}_t)\} \quad \forall t \tag{4.5}$$

where

$$\begin{aligned} Q^t(\hat{a}_t, \hat{y}_t) \quad = \quad & \min_{\substack{(\bar{n}, \bar{\mathbf{r}}) \\ \bar{m} : \bar{m} \leq \bar{n}}} \left\{ \sum_{k=t-\bar{n}}^{t-1} A_k \bar{\delta}_k + \min_{\substack{y_{\bar{r}_1} : I_{\bar{r}_1} + o_{\bar{r}_1} \geq \bar{y}_{\bar{r}_1} \\ \bar{y}_{\bar{r}_1} \geq I_{\bar{r}_1} + u_{\bar{r}_1}}} \left\{ c_{\bar{r}_1}(\bar{y}_{\bar{r}_1} - I_{\bar{r}_1}) + \mathcal{L}_{\bar{r}_1}(\bar{y}_{\bar{r}_1}) + \mathsf{E}_{D_{\bar{r}_1}, \ldots, D_{\bar{r}_2 - 1}} \Bigg[\right. \right. \\ & \min_{\substack{\bar{y}_{\bar{r}_2} : I_{\bar{r}_2} + o_{\bar{r}_2} \geq \bar{y}_{\bar{r}_2} \\ \bar{y}_{\bar{r}_2} \geq I_{\bar{r}_2} + u_{\bar{r}_2}}} \left\{ c_{\bar{r}_2}(\bar{y}_{\bar{r}_2} - I_{\bar{r}_2}) + \mathcal{L}_{\bar{r}_2}(\bar{y}_{\bar{r}_2}) + \mathsf{E}_{D_{\bar{r}_2}, \ldots, D_{\bar{r}_3 - 1}} \Bigg[\cdots + \Bigg[\min_{\substack{\bar{y}_{\bar{m}} : I_{\bar{m}} + o_{\bar{r}_{\bar{m}}} \geq \bar{y}_{\bar{m}} \\ \bar{y}_{\bar{m}} \geq I_{\bar{m}} + u_{\bar{r}_{\bar{m}}}}} \Bigg\{ \right. \\ & \left. \left. \left. c_{\bar{r}_{\bar{m}}}(y_{\bar{r}_{\bar{m}}} - I_{\bar{r}_{\bar{m}}}) + \mathcal{L}_{\bar{r}_{\bar{m}}}(y_{\bar{r}_{\bar{m}}}) + \mathsf{E}_{D_{\bar{r}_{\bar{m}}}, \ldots, D_{\bar{r}_{\bar{m}+1} - 1}} \Big[\varphi_t(I_t) \Big] \Bigg\} \Bigg] \ldots \Bigg] \right\} \Bigg] \right\} \right\} \end{aligned} \tag{4.6}$$

And $\varphi_l(.)$ represents the cost function of the AH for the periods from t forward.

$$(\hat{a}_t^*, \hat{S}_t^*) \quad = \quad \arg\min_{\hat{a}_t, \hat{y}_t} \{Q^t(\hat{a}_t, \hat{y}_t)\}$$

Optimal periods to cover from period to, $\hat{a}_t^*$, and corresponding basestock levels,$\hat{S}_t^*$, are determined almost the same way as AH I. By solving optimization problem (4.5), we introduce a new decision criterion for determining the $(\hat{a}_t^*, \hat{y}_t^*)$ pair. We make this decision based on the cost-to-go function obtained by the optimal cost of the n-period problem prior to current period t. Note that the boundary condition for this problem a possible action to be taken in period t. Although the partial history is considered in AH II, it increases the computational burden by constant factor 2^n. It still does not increase the overall complexity compared to AH. However, the value of n itself can be large that it can result in long computational times.

CHAPTER 5

COMBINATORIAL BASED HEURISTICS

This chapter introduces the our new heuristic procedures for solving Problem P. The basic idea of the procedures to be mentioned is determining the number of setups as the it promises low costs at first, and then adjusting the setup periods in order catch further cost improvements. The idea is motivated by the fact that if we already know that a production schedule is optimal, then its associated cost is at least as good as all the other production schedules where the basestock levels are determined optimally. Note that finding the optimal production schedule as explained in Chapter 3 requires to compare a given production schedule with every other possible production schedules and this result id exponentially growing solution times. In the new heuristic procedures, we developed ways to determine a near optimal solution by making comparison with a limited number of production schedules. In order to form the production schedules for comparison we define three different procedures to apply on the current production schedule.

1. Merge: While evaluating a production schedule in terms of optimality, we can combine two consecutive production cycles and form a production schedule to compare with. Note that this procedure create a schedule with one less setup by simply removing one setup from the schedule.

Merging Procedure

Input: a production schedule $(m, \mathbf{r})$
Input: $l \in \mathbb{Z}, l < m$
Output: a production schedule with one cycle less
Output: Optimal cost C
1: Let $(l+1)^{th}$ 1 in vector δ be 0.
2: $m \leftarrow m - 1$
3: Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
4: **return** $(m, \mathbf{r})$ and C

---

2. Divide: As an opposite move to merging, we can divide a production cycle into two and obtain a new schedule. This procedure simply adds a new setup to the current schedule.

Dividing Procedure

Input: a production schedule $(m, \mathbf{r})$
Input: $l \in \mathbb{Z}, l < N$
Output: a production schedule
Output: Optimal cost C
1: **if** $\delta_l = 0$ **then**
2: $\quad \delta_l \leftarrow 1$
3: $\quad m \leftarrow m + 1$
4: **end if**
5: Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
6: **return** $(m, \mathbf{r})$ and C

---

3. Switch: As stated above, even though the number of setups are fixed, there may still be need for

determining the exact location of setups. For this purpose, we can consider to move a location of a setup one period backward or forward.

Switching Procedure

Input: a production schedule $(m, \mathbf{r})$
Input: $l \in \mathbb{Z}, l \leq m$
Input: $s \in \{-1, 1\}$
Output: a production schedule
Output: Optimal cost C
if $r_l + s = 0$ or $r_l + s = N + 1$ **then**
  Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
  return $(m, \mathbf{r})$ and C
end if
if $\delta_{r_l+s} = 1$ **then**
  $m \leftarrow m - 1$
end if
$\delta_{r_l+s} \leftarrow 1$
$\delta_{r_l} \leftarrow 0$
Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
return $(m, \mathbf{r})$ and C

The following heuristic procedures make use of these procedures specified. The idea is to obtain cost improvements on the current production schedule by using these procedures. First two algorithms contains merging and switching procedures. The difference occurs on the order that these procedures are applied and how they are repeated. The other two algorithms combine dividing and switching procedures, again making the some difference in implementation.

5.1 Merging Method I (MMI)

Let us have a planning horizon of N periods. Suppose, we start with a production schedule $(m, \mathbf{r})$ where $m = N$. This implies that $\delta_i = 1 \ \ \forall i$ which means there are scheduled setups at the beginning every period. At this point, we can start our local search by merging these small cycles one by one and we solving each production schedule for optimality. Among all possible merging operations to be performed on the current production schedule, we pick the one providing the most improvement. Then we can perform merging operations on the new current production schedule. We continue these merging operations until no improvement is observed. When improvement stops, we look for one more opportunity to improve our cost by switching the current replenishment cycles. We again pick the switching operation with the best improvement. And we continue with this operation until no improve is observed. Then we can merge our current production schedule as explained before, and then continue with switching until no cost improvement occurs with any of these moves. We apply this procedure according to following algorithm and call this:

Algorithm 1 Merging Method I

```
Input:  production schedule (m, r)
Output:  Best cost schedule
 1: Find the optimal basestock levels for schedule (m, r). Let C the optimal cost of this schedule. Let BestSoFar ← C. Let
    min1 ← ∞, min2 ← BestSoFar
 2: for min1 > min2 do
 3:     r̂ ← r, r̄ ← r, l ← 1, n ← m, imp ← 0
 4:     for l < n do
 5:         Merge((m, r̂),l)
 6:         if C < BestSoFar then
 7:             Let BestSoFar ← C, r̄ ← r̂, imp ← 1.
 8:         end if
 9:         r̂ ← r, l ← l + 1.
10:     end for
11:     r ← r̄, r̂ ← r
12:     if imp = 1 then
13:         Go to 3
14:     end if
15:     l ← 1, n ← m, imp ← 0, s ← −1
16:     for s ≤ 1 do
17:         for l ≤ n do
18:             Switch((m, r̂),l,s)
19:             if C < BestSoFar then
20:                 Let BestSoFar ← C, r̄ ← r̂, imp ← 1.
21:             end if
22:             r̂ ← r, l ← l + 1.
23:         end for
24:         s ← s + 2, l ← 1
25:     end for
26:     r ← r̄, r̂ ← r
27:     if imp = 1 then
28:         Go to 15
29:     end if
30:     min1 ← min2, min2 ← BestSoFar
31: end for
32: return  (m, r) and C
```

MMI quickly starts to merge its small cycles to bigger cycles once the algorithm starts. As long as there is still promising cost improvement, merging new cycles continues. The aim to take these kind of action is to come up with a schedule which has close number of setups with the optimal production cycles. Once going in merging direction is no longer promising, we search for other cost improvement by changing the setup locations by one period. This action will enlarge one cycle and shrink an adjacent cycle by period which can also improve our production cycle in terms of costs. And we continue with switching cycles as long as we obtain production schedules with lower costs.

5.2 Merging Method II (MMII)

Merging Method is a depth first method to apply. If an improvement is obtained by any move, it continues to perform that move until no improvement is obtained. We can make a small modification in the previous algorithm in the following way. Again, we start with replenishment schedule $(m, \mathbf{r})$ where $m = N$. We start with merging the cycles, and we take a greedy step with this move. Then, we immediately switch the new cycles and look for best improvement. Then we make another move with merging cycles. And then with switching. We continue on altering these moves one by one until no improvement is observed. Following algorithm summarizes this procedure:

Algorithm 2 Merging Method II

Input: production schedule $(m, \mathbf{r})$
Output: Best cost schedule

Find the optimal basestock levels for schedule $(m, \mathbf{r})$. Let C the optimal cost of this schedule. Let $BestSoFar \leftarrow C$. Let $min1 \leftarrow \infty, min2 \leftarrow BestSoFar$
for $min1 > min2$ **do**
  $\hat{\mathbf{r}} \leftarrow \mathbf{r}, \bar{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow 1, n \leftarrow m$
  for $l < n$ **do**
    Merge($(m, \hat{\mathbf{r}})$,l)
    if $C < BestSoFar$ **then**
      Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
    end if
    $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}, \hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $l \leftarrow 1, n \leftarrow m, s \leftarrow -1$
  for $s \leq 1$ **do**
    for $l \leq n$ **do**
      Switch($(m, \hat{\mathbf{r}})$,l,s)
      if $C < BestSoFar$ **then**
        Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
      end if
      $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
    end for
    $s \leftarrow s + 2, l \leftarrow 1$
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}, \hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $min1 \leftarrow min2, min2 \leftarrow BestSoFar$
end for
return $(m, \mathbf{r})$ and C

In MMI, if a procedure improves the cost, we continue with applying the procedure until cost is no longer improved. However, this may cause us to skip some of the certain schedules which could be promising in terms of cost. In MMII, an action of merging is always followed by switching. And the cost improvements are obtained not going in one direction, merge or switch, only; but making use of both procedure simultaneously.

5.3 Dividing Method I (DMI)

Following the idea in MMI and MMII, we can consider a similar method proceeding backwards. Let us start with a production schedule $(m, \mathbf{r})$ where $m = 1$. Which implies that $\delta_1 = 1$ and $\delta_i = 0 \quad \forall i \neq 1$. This implies that there is only one setup and a cycle containing all of the periods. Starting from this one-cycle production schedule, we may apply our greedy approach to decide on which period to place a new setup which will form two production cycles. Then with the new production schedule, we can look for placing other setups until no cost improvement is observed. Then, we perform switching for every setup present and pick the most promising setup to switch. With this new production schedule, we continue with switching until cost improvement stops. Then we continue with dividing first and switching after until no cost improvement is possible with any of these moves. The following algorithm summarizes this procedure:

Algorithm 3 Dividing Method I

```
Input: production schedule (m, r)
Output: Best cost schedule
Find the optimal basestock levels for schedule (m, r). Let C the optimal cost of this schedule. Let BestSoFar ← C. Let
min1 ← ∞, min2 ← BestSoFar
for min1 > min2 do
    r̂ ← r, r̄ ← r, l ← 1, n ← m, imp ← 0
    for l ≤ N do
        Divide((m, r̂),l)
        if C < BestSoFar then
            Let BestSoFar ← C, r̄ ← r̂, imp ← 1.
        end if
        r̂ ← r, l ← l + 1.
    end for
    r ← r̄, r̂ ← r
    if imp = 1 then
        Go to 3
    end if
    l ← 1, n ← m, imp ← 0, s ← −1
    for s ≤ 1 do
        for l ≤ n do
            Switch((m, r̂),l,s)
            if C < BestSoFar then
                Let BestSoFar ← C, r̄ ← r̂, imp ← 1.
            end if
            r̂ ← r, l ← l + 1.
        end for
        s ← s + 2, l ← 1
    end for
    r ← r̄, r̂ ← r
    if imp = 1 then
        Go to 15
    end if
    min1 ← min2, min2 ← BestSoFar
end for
return (m, r) and C
```

DM I also aims for finding the optimal number of setups first. As oppose to MMI, it starts with $m = 1$ instead of $m = N$. But the logic is exactly the same thinking in terms of how we use dividing and switching procedures.

5.4 Dividing Method II (DMII)

Dividing Method is very similar to Merging Method in terms of keeping on performing a move until it stops improvement. We can modify dividing method to derive another procedure as we did in merging method. Again, we start with production schedule $(m, \mathbf{r})$ where $m = 1$. We start with dividing the cycles, and we take a greedy step with this move. Then, we immediately switch the new cycles and look for best improvement. Then we make another move with dividing cycles. And then with switching. We continue on altering these moves one by one until no improvement is observed. Following algorithm summarizes this procedure:

Algorithm 4 Dividing Method II (DMII)

Input: Replenishment schedule $(m, \mathbf{r})$
Output: Best cost schedule

Find the optimal basestock levels for schedule $(m, \mathbf{r})$. Let C the optimal cost of this schedule. Let $BestSoFar \leftarrow C$. Let $min1 \leftarrow \infty, min2 \leftarrow BestSoFar$
for $min1 > min2$ **do**
  $\hat{\mathbf{r}} \leftarrow \mathbf{r}$, $\bar{\mathbf{r}} \leftarrow \mathbf{r}$, $l \leftarrow 1$, $n \leftarrow m$
  for $l < n$ **do**
    Divide($(m, \hat{\mathbf{r}})$,l)
    if $C < BestSoFar$ **then**
      Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
    end if
    $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $l \leftarrow 1$, $n \leftarrow m$, $s \leftarrow -1$
  for $s \leq 1$ **do**
    for $l \leq n$ **do**
      Switch($(m, \hat{\mathbf{r}})$,l,s)
      if $C < BestSoFar$ **then**
        Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
      end if
      $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
    end for
    $s \leftarrow s + 2, l \leftarrow 1$
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $min1 \leftarrow min2$, $min2 \leftarrow BestSoFar$
end for
return $(m, \mathbf{r})$ and C

CHAPTER 6

COMPUTATIONAL EXPERIMENTS AND DISCUSSION

After analysing Problem P thoroughly and proposing new heuristic procedures, we have conducted computational experiments. In our experiments, we first tried to show time effectiveness of our lower bounds. Then we tested the quality of our heuristic procedures in terms of optimality gaps and computational times. We evaluated the computational times in minutes. And observed how they increase with N. We first started with stationary parameters in our experiments. We identified the parameters which have an impact of heuristic performances, and then we worked with dynamic capacity cases. Finally, we conducted paired-t test on our results to point out statistical significance between our methods. This chapter summarizes the results of these work done.

6.1 Problem Set

We started our analysis with cost parameters and lot size restrictions do not change throughout the planning horizon. For each of the parameters, we worked with at least three levels namely low, medium, and high. In that respect, we used set up costs of $A \in \{2, 20, 50, 200\}$, unit production costs of $c \in \{1, 5\}$, and penalty costs of $b \in \{2, 8, 32\}$. Holding costs are assumed to be a linear function of unit production cost such that $h = 0.1 \times c$. Among these parameters, we excluded the combinations where $c = 5$ and $b = 2$ since it is not realistic. Lot size restrictions are taken as $u \in \{0, 5, 10\}$ and $o \in \{10, 20, 40\}$. And we excluded cases with $u = 5$ & $o = 10$, $u = 10$ &$o = 10$, and $u = 10$ & $o = 20$. We also considered six different poisson distributed demand patterns static (P1), increasing (P2), decreasing (P3), hectic (P4), product life cycle (P5), and seasonal (P6) which are visually illustrated in Figure 6.1. Mean demands for periods where $N = 12$ are summarized in the Table 6.1.

Table 6.1: Parameters for λ_t

Period	λ_t (Poisson Distribution)					
	P1	P2	P3	P4	P5	P6
1	5	1.62	8.38	2	7.5	3.52
2	5	2.23	7.77	1	9.33	7.04
3	5	2.85	7.15	23.5	10	7.04
4	5	3.46	6.54	1	9.33	7.04
5	5	4.08	5.92	2	7.5	7.04
6	5	4.69	5.31	1	5	7.04
7	5	5.31	4.69	2	2.5	6.04
8	5	5.92	4.08	21	0.67	5.04
9	5	6.54	3.46	2	0	4.04
10	5	7.15	2.85	1	0.67	3.04
11	5	7.77	2.23	2	2.5	2.04
12	5	8.38	1.62	1.5	5	1.08
$\sum$	60	60	60	60	60	60

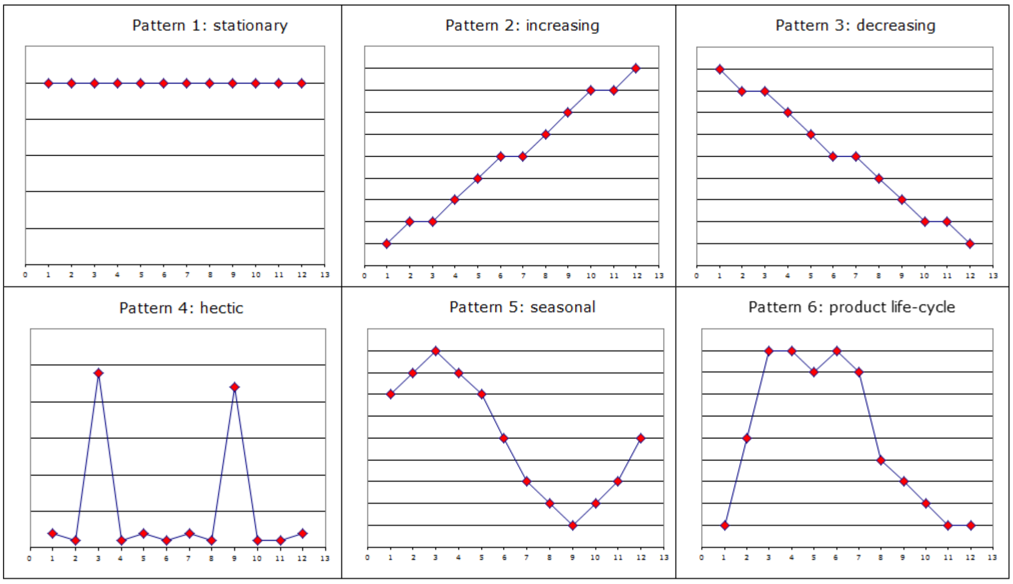

Figure 6.1: Demand Patterns

With the parameters mentioned, we created 720 problem instances. We solved each of them for optimality, and we also applied our heuristics. Following subsections provide the details about the results.

6.2 Lower Bound Effectiveness

As we discussed in Section 3.3.3, the implementation of LB_3 requires to determine a number of line segments for approximation, k. In order to decide on this parameter, we considered approximations with $k \in \{2, 3, 5\}$. For $k = 2$, we considered two points located close to oblique asymptotes of the convex const function on the opposite ends. For $k = 3$, we also considered a point close to the minimizer of the cost function. And for $k = 5$, we included two additional points located between the minimizer and asymptotes. In order to find out about best k value, we attempted to solve our data set utilizing LB_3 only, and recorded our results for $N = 12$. In Table 6.2, the columns represent the number of line segments used for approximation. The first three rows shows the average number of production schedules eliminated by the lower bound for maximum lot sizes of 10,20, and 40 respectively. The last row points the average production schedules eliminated over all the problems. Note that for $N = 12$, there are a total of 2024 production schedules to be considered.

Table 6.2: LB_3 Effectiveness with Number of Line Segments

	k=2	k=3	k=5
o=10	1339.78	1817.43	1487.91
o=20	1303.29	1771.56	943.67
o=40	1289.99	1859.04	1147.43
Total	1302.73	1822.94	1136.25

It can clearly be observed that the best results are obtained by $k = 3$. On the average it eliminates 1822.94 (90.01%) of the problems without solving for optimality. LB_3 with $k = 2$ and $k = 3$ performs very poor in that respect, eliminating no more than 1302.73 (64.32%) problems on the average. Since, we developed this lower bound in order to handle the inaccuracy of the former lower bounds, we also investigated its performance versus maximum lot sizes. We observed that the performance weakens while the maximum lot size increases for $k = 2$ and $k = 5$. On the other hand, the results obtained for $k = 3$ remains consistent while maximum lot size changes. Therefore, we decided to apply our LB_3 by using 3 line segments from now on.

In order to see the effectiveness of the lower bounds, we first solved the 720 problems we introduced above for optimality with only with LB_1, LB_2, and LB_3. Then we solved the problems again using LB_1 & LB_2 and LB_1 & LB_2 & LB_3. The columns of Table 6.3 correspond to these combinations of lower bound for solutions. In the rows 1 through 4, we recorded the average number of production schedules eliminated by these lower bound combinations, for different maximum lot size values and overall. In the last row, we present the CPU time to obtain all the solutions on IntelCore2 4GB Windows.

Table 6.3: Lower Bound Effectiveness Summary

	LB_1	LB_2	LB_3	LB_1&LB_2	LB_1&LB_2&LB_3
o=10	250.03	692.43	1817.43	692.43	1839.58
o=20	746.88	1907.08	1771.56	1907.08	1957.46
o=40	946.31	1978.27	1859.04	1978.27	2020.69
Total	763.79	1740.23	1822.94	1740.23	1969.42
Time (min)	137.71	90.00	100.6	81.55	63.56

From Table 6.3, we can easily see that LB_1 itself do not promise the best results by only eliminating 763.71 production schedules on the average. The time to find the optimal solution appear to be 137.71 minutes, if we only rely on LB_1. LB_2 and LB_3 seems closer when $o = 20$ and $o = 40$, although LB_2 is somewhat better. However, LB_3 dominates LB_2 when $o = 10$ by eliminating near three times of schedules that LB_2 eliminated. Times wise, there is not much difference between these two lower bounds. For the combinations of the lower bounds, we observe the most significant improvements. Although the number of cases eliminated for LB_1&LB_2 and LB_2 are exactly the same because of the fact that LB_2 dominates LB_1, the solution time is reduced to 81.55 with the help of the schedule that LB_1 can eliminate itself. We see $LB_1\&LB_2\&LB_3$ as the best performing column by eliminating 1969.42 (96%) of the production schedules on the average and completing to solve 720 problems to optimality in 63.56 minutes.

6.3 Heuristic Accuracy

First of all, we tested our heuristic procedures the proximity to the optimal solution. Therefore we tried to identify their optimality gaps. Let production schedule obtained by solving Problem P with a heuristic method be $(\bar{m}, \bar{\mathbf{r}})$, and $(m^*, \mathbf{r}^*)$ be the optimal production schedule for Problem P. Then we define the optimality gap as

$$optimality\ gap\ (\%) = \frac{(cost\ of\ schedule\ (\bar{m}, \bar{\mathbf{r}})) - (cost\ of\ schedule\ (m^*, \mathbf{r}^*))}{(cost\ of schedule\ (m^*, \mathbf{r}^*))}$$

6.3.1 Constant Parameters

After solving 720 problems for optimality, we have applied our heuristic procedures. In the following tables, we present our results as follows. The first two columns are for the results for AH and AH I respectively. From column 4 to 7, we see the results for AH II. Since AH II has another parameter, n, 4 different columns are constructed for different n values. The columns after correspond to MMI, MMII, DMI, and DMII respectively. On the other dimension of the tables, the first row represents the number of cases solved optimally among the full problem set by using the heuristic procedure. Rows 2, 3, and 4 show the number of cases solved with less than 1,2, and 5% optimality gap respectively. We also recorded the maximum and average gaps in the last two rows. The results for the 720 problems with constants parameters are summarized in Table 6.4.

Table 6.4: Summary of Heuristics $N = 12$

			AH II				Merge		Divide	
	AH	AH I	n=1	n=2	n=3	n=4	Method I	Method II	Method I	Method II
optimal	440	565	508	524	537	562	649	637	635	640
< 1%	541	662	627	650	647	662	716	713	714	718
< 2%	611	690	675	682	677	692	718	716	719	720
< 5%	655	710	702	701	706	712	720	720	720	720
ave (%)	1.36	0.35	0.50	0.51	0.41	0.27	0.05	0.06	0.06	0.05
max (%)	31.80	25.66	14.17	24.91	14.42	10.20	3.27	3.28	2.19	1.42

We observed that dynamic programming based heuristics fail to solve the problem optimally. The best performer AH I solves 565 problems optimally and AH II with $n = 4$ follows with 562 problems. AH performs the least by solving only 440 problems to optimality. The number of problems solved with reasonable optimality gap is also unsatisfactory. AH can only solve 655 problems with less than 5% optimality gap whereas this number is 710 for AH and 712 for AH II with n=4 which can be considered moderate. When we look at the average gaps, we see that the performances of AH I and AH II are quite satisfactory by providing values around 0.40%. However, AH performs worse than the others with 1.36%. The failure of DP based heuristics can better be understood by investigation of maximum gaps. There are cases that AH solves with 31.80%, AH I solves with 25.6% optimality gaps. The top performing procedure in this class of heuristic, AH II with n=4, can only provide 10.20% optimality gap. Combinatorial based heuristics perform much better by providing less than 5% optimality gap for all of the cases. It can also be seen that no more than 6 cases yield below 1% optimality gap in these type of heuristic. The number of cases solved for optimality appears to be no less than 635 combinatorial based heuristic. In terms of average gap, combinatorial based heuristic give either 0.06% or 0.05% optimality gap. They also dominate the DP based heuristics on maximum gaps. Specifically, DMII performs the best by having a maximum gap of only 1.42%.

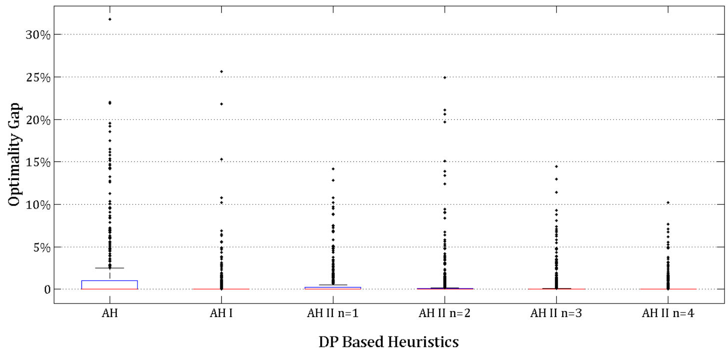

Figure 6.2: Optimality Gap Distribution for DP Based Heuristics

In order to get more insight about the heuristic accuracies, we looked at the gap distributions on side-by-side box plots. In Figure 6.2, we can see the distribution for DP based heuristic procedures. It can easily be seen that, AH is the highest spread among these procedures. It is also the poorest in terms of number of cases solved less than 10% optimality gap. The other procedures solves majority of the problems with less than 10% optimality gap, and similar spread along the y-axis. Only AH I and AH II with n=2 seems to have more spread than the others. However, for these type of heuristics, it can clearly be visualized that the number of problems solved with unacceptable optimality gaps are quite high.

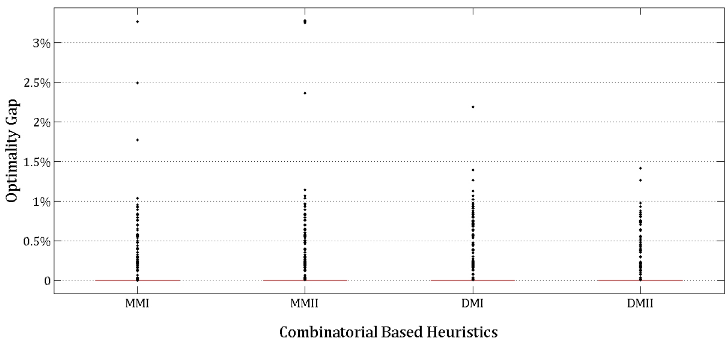

Figure 6.3: Optimality Gap Distribution for Combinatorial Based Heuristics

In Figure 6.3 we can see the similar analysis for combinatorial based heuristics. Clearly, all of the heuristic procedures in these type of heuristics solve majority of the problems with less than 1.5% optimality gap. Only a few number of problems are solved with higher optimality gaps which cause the maximum gaps to be higher when average gaps are very close to 0.

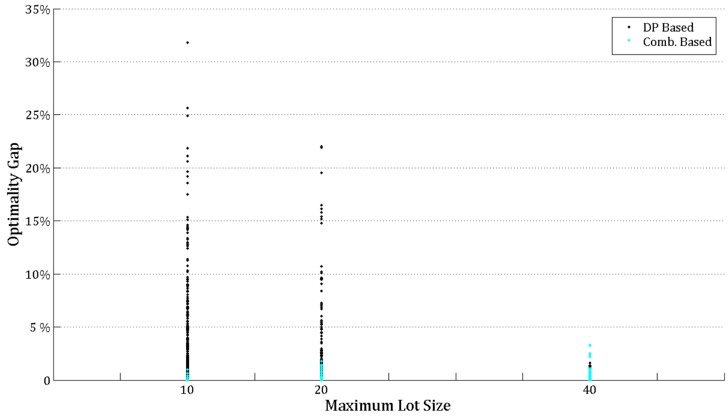

Figure 6.4: Optimality Gap Distribution vs Maximum Lot Size

After capturing an idea about the number of cases with relatively high optimality gaps, we continue our investigation with pointing out the causing parameter for these gap values. When we analyze the results closer in Figure 6.4, we observed that maximum lot size parameters have an important impact on the gap rates. For dynamic programming based heuristics, all of the cases having optimality error more than 2% happen to have $o \in \{10, 20\}$ which can be named as tight capacity. For local search based heuristic procedures, cases with similar optimality gap have $o \in \{40\}$ which is loose capacity. Which gives us the idea that these two heuristic procedure types are complementary. If we pick one heuristic procedure from each type, apply them to the same problem and take the one with the minimum cost as the final solution, there is a good chance for improving the optimality gaps. The results obtained by this approach are summarized in Table 6.5.

Table 6.5: Minimum of Two Heuristics $N = 12$

	MM I	MM I	MM I	MM II	MM II	MM II	DM I	DM I	DM I	DM II	DM II	DM II
	AH	AH I	AH II [1]	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
optimal	687	702	705	710	698	703	697	710	710	698	705	709
< 1%	720	720	720	720	720	720	720	720	720	720	720	720
< 2%	720	720	720	720	720	720	720	720	720	720	720	720
< 5%	720	720	720	720	720	720	720	720	720	720	720	720
ave (%)	0.01	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.01	0.00
max (%)	0.82	0.76	0.76	0.40	0.76	0.76	0.79	0.93	0.76	0.72	0.93	0.46

As can be seen, in line with the previous observations on capacity versus optimality gap, optimality gaps considerable decreases with the procedure applied. All of the cases in the problem set can be solved with less than 1% optimality gap, employing this mixed procedure. Above that, number of cases solved for optimality is close to 700 for all of the procedures. When we look at the average gaps, we see that at most 0.01% of error can be attained. All the pairs result in maximum optimality gaps less than 1% which illustrates the complementariness of the dynamic programming and local search based heuristics where MMII & AH and DMII & AH II paires give 0.40% and 0.46% maximum gaps respectively.

[1] AH II refers to AH II with n=4

6.3.2 Dynamic Capacity

The results in the previous subsection suggest that capacity is the parameter with the most impact on heuristic accuracy. Considering that upper limits on the lot sizes may change over time, we are required to test our heuristic procedure under variable capacity scenarios. In order to produce these scenarios, we used the **e** vectors with entries $\{-1, 0, 1\}$ given as the columns of table 6.6 such that $\sum_t e_t = 0$. A visualization of these patterns are also provided in Figure 6.5.

Table 6.6: **e** vectors used for dynamic capacity scenarios

Period	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
1	-1	-1	1	-1	-1	-1	-1	1	0	1	-1	0
2	1	-1	1	0	1	0	0	-1	1	-1	-1	0
3	0	-1	1	1	-1	1	-1	0	1	-1	1	-1
4	-1	-1	1	0	1	-1	1	-1	-1	-1	-1	0
5	-1	0	0	-1	-1	0	-1	1	-1	1	1	0
6	0	0	0	0	1	1	-1	-1	0	1	-1	1
7	1	0	0	1	-1	-1	1	0	1	0	1	1
8	-1	0	0	0	1	0	1	1	0	1	1	0
9	1	1	-1	-1	-1	1	1	1	-1	1	1	0
10	0	1	-1	0	1	-1	0	0	-1	0	-1	1
11	0	1	-1	1	-1	0	-1	-1	1	-1	-1	-1
12	1	1	-1	0	1	1	1	0	0	-1	1	-1

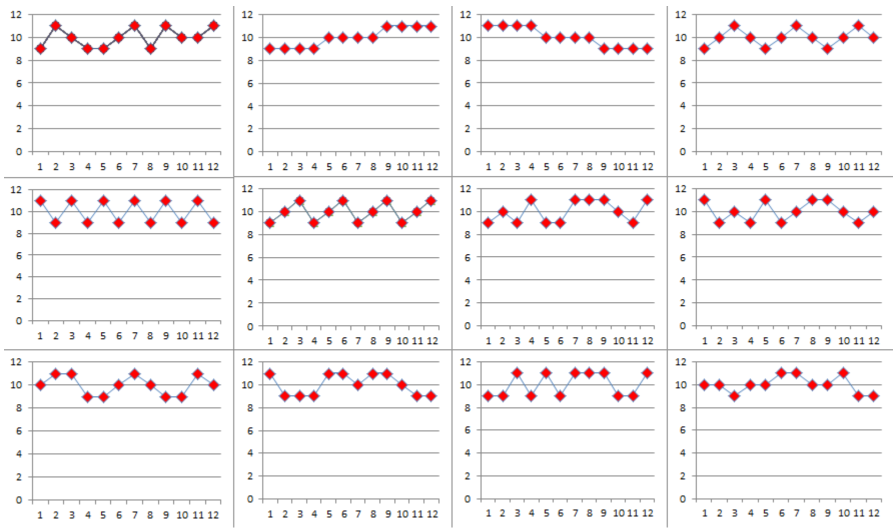

Figure 6.5: Dynamic Capacity Patterns

In combination with these patterns, we also used two more factors to come up with new capacity scenarios. Parameter α is the factor of overall capacity through out the planning horizon. We take $\alpha \in \{0.75, 1, 3\}$ in order to adjust the overall capacity to be low, moderate, high. The other parameter β represents the amplitude deviation from the average capacity in a given period. We take $\beta \in \{1, 3\}$ for implying high and low deviations. Finally, we formed our capacity scenarios by using the formula;

$$o_t = \alpha \times (10 + \beta \times e_t) \quad \forall t \in \{1, 2, \ldots, 12\}$$

For the analysis in this part, we ignored the minimum requirements on lot sizes by simply taking $u = 0$

for all scenarios. We used the combinations of the other parameters as specified in Section 6.3.1 and formed 8640 problems in total. As we did before, we solved all these problems for optimality and applied our heuristics. The results are summarized in Table 6.7

Table 6.7: Summary of Heuristics $N = 12$ for Dynamic Capacity

	AH	AH I	AH II				Merge		Divide	
			n=1	n=2	n=3	n=4	Method I	Method II	Method I	Method II
optimal	3624	5409	4639	4983	5242	5514	7569	7369	7869	7912
< 1%	4794	6664	6054	6245	6475	6706	8197	8056	8530	8544
< 2%	5443	7225	6683	6816	7054	7262	8342	8211	8609	8609
< 5%	6498	8075	7598	7763	8018	8157	8517	8441	8638	8638
ave (%)	3.70	1.18	1.94	1.52	1.12	0.98	0.24	0.37	0.05	0.05
max (%)	50.78	92.74	44.26	34.47	26.00	55.83	23.43	35.77	6.66	6.66

The poor performance of DP based heuristic continue. AH II with n=4 solves 5514 problems for optimality and provides average gap of 0.98%; however, it is very poor with 55.83% maximum gap and failing in terms of worst case solutions. Although the average gaps appear to be around 1.5% for these heuristics, the maximum gap of AH I being 92.74% give hesitation to apply these procedures to have a good solution. Although better than DP based heuristics; unlike the stationary capacity problems, MMI and MMII performs poor for this problem set. Both methods fail to solve more then 1000 problem for optimality, and near 400 with less than 5% gap. Average gaps obtained are still less than 0.40%; however, maximum gaps of 23.43 and 35.77% do not promise much for using these procedure to obtain a final solution. DMI and DMII gives the best results for this experiment. Although the number optimal solutions obtained seem to less in percentage (7869 for DMI, 7912 for DMII), only 2 of the problems are solve with more than 5% optimality gap which is quite successful where all the other heuristics fail. Once again the results obtained by dividing methods ensure a maximum gap of only 6.66% proving that using these procedures are not risky in terms of the worst case scenarios.

Table 6.8: Minimum of Two Heuristics $N = 12$ for Dynamic Capacity

	MM I	MM I	MM I	MM II	MM II	MM II	DM I	DM I	DM I	DM II	DM II	DM II
	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
optimal	7960	8267	8256	8336	8191	8173	8267	8420	8432	8295	8432	8441
< 1%	8366	8519	8508	8543	8480	8469	8577	8613	8612	8578	8610	8612
< 2%	8462	8566	8557	8581	8546	8537	8619	8634	8633	8619	8634	8633
< 5%	8578	8620	8618	8627	8620	8618	8638	8640	8640	8638	8640	8640
ave (%)	0.14	0.05	0.06	0.04	0.06	0.07	0.03	0.01	0.01	0.02	0.01	0.01
max (%)	23.43	8.62	9.13	8.24	8.62	9.13	6.66	3.23	3.23	6.66	3.23	3.23

As we applied for stationary capacities, we again took the minimum of two procedures from each heuristic type. The results are presented in Table 6.8. The results obtained with combinations of MMI and MMII are considerable improved compared to their stand alone performances. However, they are still not as promising as they used to be for stationary capacity cases. It is notable to point out that MMII & AH solves 8336(96%) problems for optimality and provides 8.24% of maximum gap. On the other hand, we see observe much better results for combinations of DMI and DMII. Almost all of the problems are solved with less than 5% optimality gap with these procedures and the number of problems solved for optimality is around 8300. The average gaps are close to 0.01% where DMI & AHI, DMI & AH II, DM II & AH I, and DMII & AH II combinations yield a maximum gap of only 3.23% which is tremendous.

6.4 Computational Times

Just as much as the accuracy of the solutions, computational times are also important while considering the heuristic qualities. As we discussed before, finding the optimal solution has a complexity of 2^N which can be visualized by solving our 720 problems when $N = 12$ and $N = 18$. Computational times occur to be 63.57 and 5782.63 minutes respectively as shown in Figure 6.6. It is worthwhile to note that 5782.63 minutes which is the solution time 720 problems in total is approximately 4 days. When $N \geq 24$ computational times exceeds weeks, or even months in which case we did not attempt to solve the problems optimally.

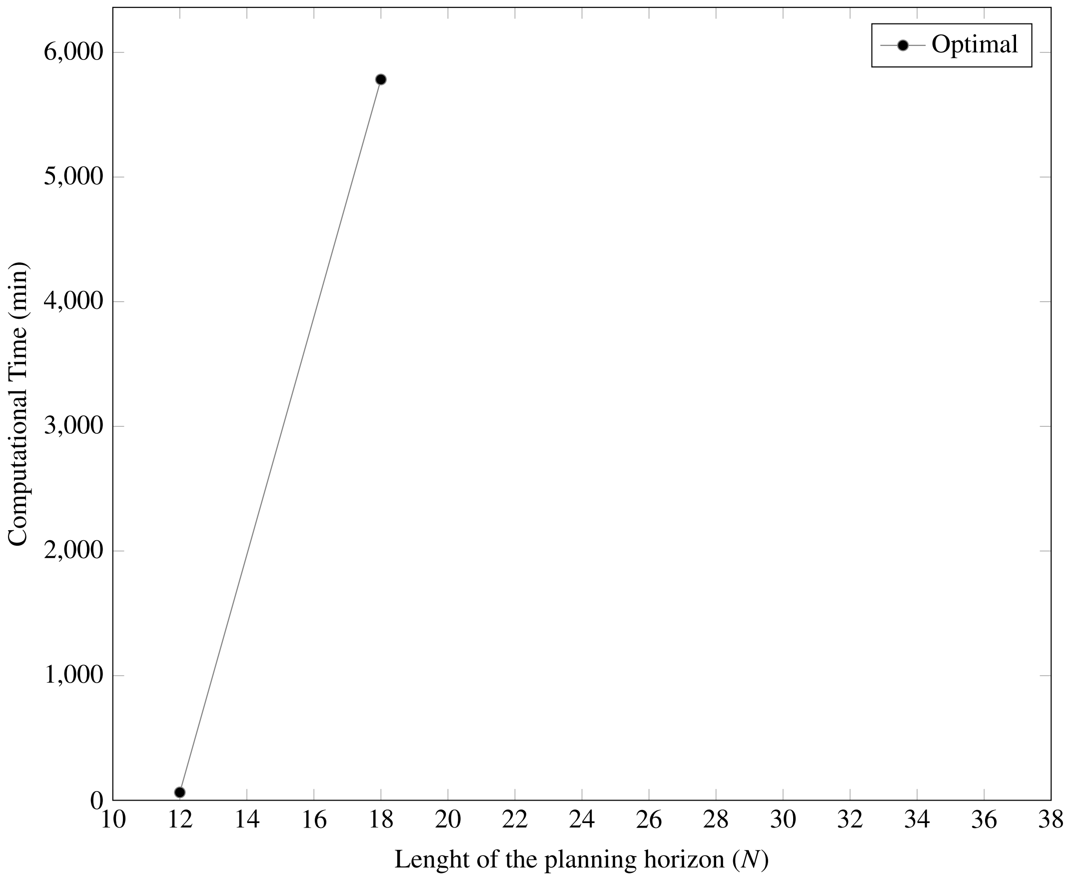

Figure 6.6: Computational Times versus N for Optimal Solution

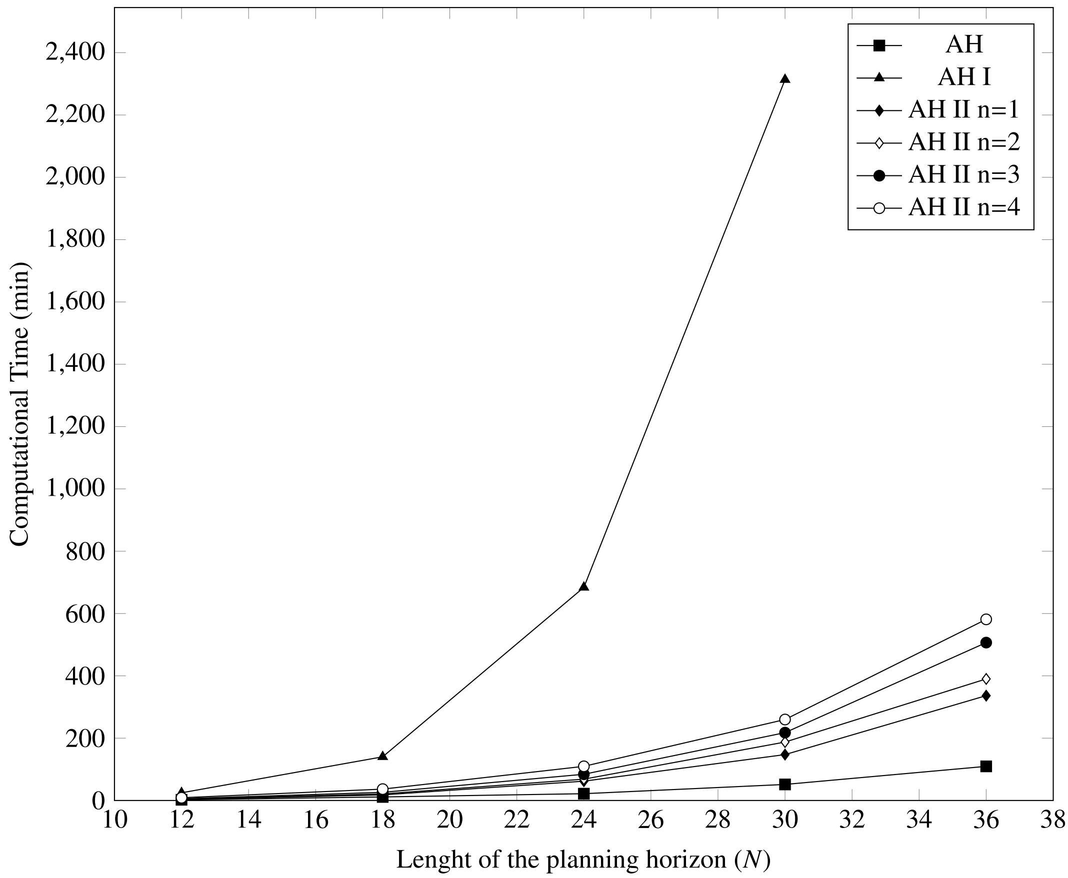

Figure 6.7: Computational Times versus N for Dynamic Programming Based Heuristics

When we investigate the dynamic programming based heuristics, we see that all of them are polynomial in time as shown in Figure 6.7. AH has the lowest complexity and the lowest computational times compared to all other heuristic procedures but note that it has the poorest optimality gaps. AH I has a polynomial complexity in N, however, its order seems higher than all the other heuristic procedures while N grows. Although not exponential in N, solution times become unacceptable for $N = 30$, that we did not attempt to solve for $N = 36$. AH II variants perform as expected. They are not very different from AH in terms of complexity; however, n being 1,2,3, or 4 may result in longer computational times.

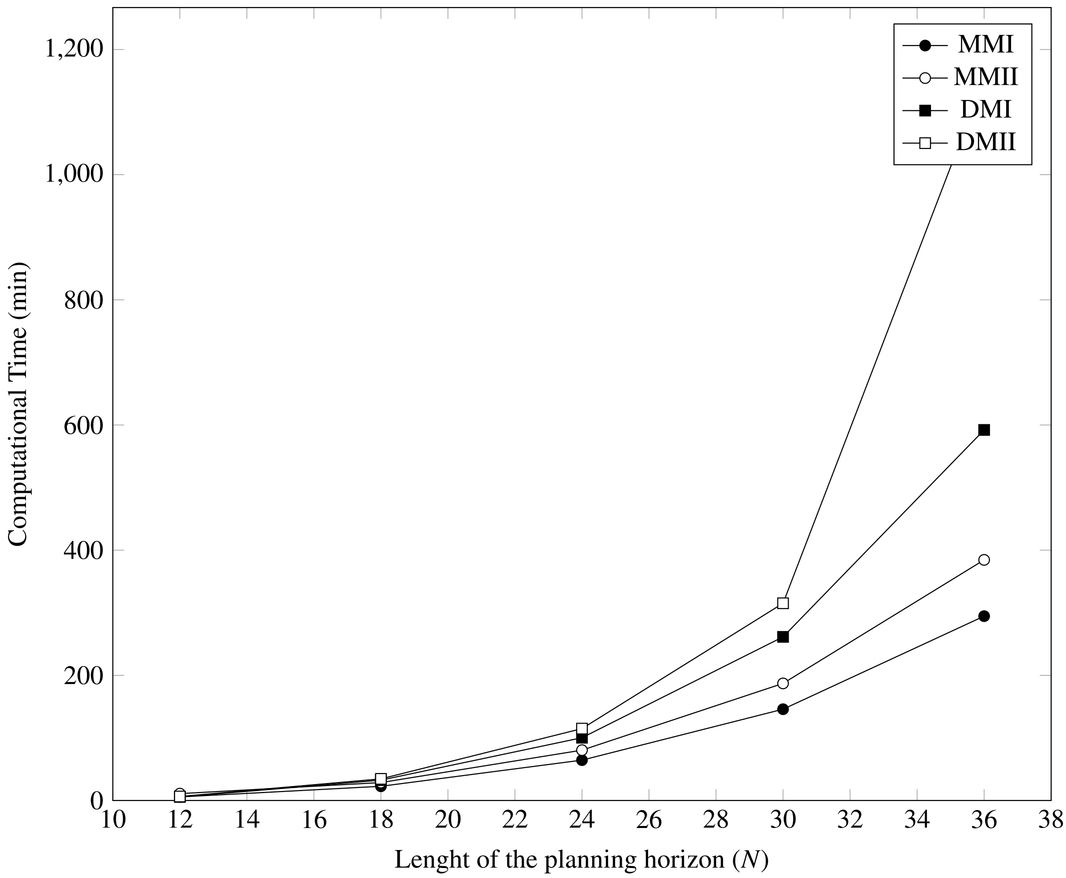

Figure 6.8: Computational Times versus N for Local Search Based Heuristics

Combinatorial Based Heuristics more or less perform the same with each other and with DP based heuristics. In fact dividing methods last a bit longer. Still, all of them exhibit polynomial behavior with small orders. Computational times versus N graph can be seen in Figure 6.8.

Overall, we see that it is not possible to solve problems with $N \geq 24$ because of the excessive solution times. AH appears to be quite successful in terms of solution times; however, it is dominated by every other procedure in terms of optimality gaps. Although it works for smaller N values, AH I also becomes less practicable for longer planning horizons. AH II variants and Combinatorial Based Heuristics seem satisfactory in terms of solution times.

6.5 Statistical Analysis

After running many experiments for constant and variable capacity cases, we continue with a statistical analysis to decide dominance relations between heuristics if there are any. We make our comparison in terms of heuristic accuracy. In order to do that, we made paired t-tests with 95% confidence between the heuristics. Out of the 720 cases we solved for $N = 12$, we take the optimality error of each heuristic in each case as an observation and compare if the mean errors differ from each other. Since we apply our heuristic to the exactly same problems, we can pair the observations in making analysis. For constant capacity cases, the performance of the heuristics can be given as;

$$AH < AH\ II\ n = 1 \equiv AH\ II\ n = 2 < AH\ I \equiv AH\ II\ n = 3$$

$$< AH\ II\ n = 4 < MMI \equiv MMII \equiv DMI < DMII$$

where "$a < b$" means than heuristic a provides significantly less accuracy compared to heuristic b, "$a \equiv b$" means there is no significant difference. Like it can be inferred from Table 6.4, Dividing Method II gives us the best results in terms of heuristic performance.

We already know that none of the heuristic alone preserves satisfactory results. Knowing that we will need a combination out of these heuristics, we are to determine which combination is the best. If we apply the same tests for the combinations of the heuristics, we obtain the following results.

$$MMI\&AH < MMI\&AH\ I \equiv MMI\&AH\ II \equiv MMII\&AH\ I < MMII\&AH \equiv MMII\&AH\ II$$
$$\equiv DMI\ AH < DMI\&AH\ I \equiv DMI\&AH\ II \equiv DMII\&AH \equiv DMII\&AH\ I \equiv DMII\&AH\ II$$

where "$a\&b$" represents the results for the combination of heuristics a and b. By this relation, we can say that for constant capacity cases it is best to work with minimum of Dividing Method II and AH Extention II with $n = 4$. Note that these procures both are shown to have acceptable solution time.

For the variable capacity analysis, we have conducted the same test this time for a sample of 8640 cases for each heuristic. The results are give as;

$$AH < AH\ II\ n = 1 < AH\ II\ n = 2 < AH\ I \equiv AH\ II\ n = 3$$
$$< AH\ II\ n = 4 < MMI < MMII < DMI < DMII$$

There are slight changes in the relations but Dividing Method II still stays as the best among individual heuristic procedures in accordance with the results in Table 6.7.

When we come to analyze the combinations, the relations get a little bit more complicated. We can classify the relations in terms of their involvement with local search based heuristics as follows;

$$MMI\&AH < MMI\&AH\ I \equiv MMI\&AH\ II$$
$$MMII\&AH\ I < MMII\&AH \equiv MMII\&AH\ II$$
$$DMI\&AH < DMI\&AH\ I \equiv DMI\&AH\ II$$
$$DMII\&AH < DMII\&AH\ I \equiv DMII\&AH\ II$$

We can additionally obtain the results that combinations of Merging Method with AH extensions are both dominated by dividing method heuristics by the t-tests. Note that this could also be easily seen from Table 6.8. However, there is no clear dominance between Dividing Method heuristics and their combinations. Therefore, we make another extensive comparison between DMI&AH I, DMI&AH II, DMII&AH I, and DMII&AH II. The results show that;

$$DMI\&AH\ I \equiv DMI\&AH\ II \equiv DMII\&AH\ I \equiv DMII\&AH\ II$$

failing to capture any dominance between these mentioned procedures. Although we cannot distinguish among these procedures, the solution times can make the difference. We already observe that AH I tends to create excessive solution times for longer planning horizons. Knowing that the accuracy will not change significantly, we can choose not to use a combination with AH I just for the sake of shorter solution times. The decision on choosing between DM I& AH II with n=4 and DM II&AH II with n=4 is just up to the decision maker. The statistical output that analyzed for the test used in this section can be found in Appendix B.

6.6 Convexity

Our analysis shows that our total cost function is compositions of other convex functions. All of our one-cycle costs are convex functions of lot sizes. Given production schedules, all cost-to-go functions are convex in lot sizes which provides modified base stock policies to be optimal. In such environment, we defined the function P such that:

$$P(m) = \min_{(m,\mathbf{r})} \Bigg\{ \sum_{k=1}^{N} A_k\delta_k + \min_{y_1:I_1+o_{r_1}\geq y_1\geq I_1+u_{r_1}} \Big\{ c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\ldots,D_{r_2-1}}\Big[\min_{y_2:I_2+o_{r_2}\geq y_2\geq I_2+u_{r_2}} \Big\{ c_{r_2}(y_2 - I_2) + \mathcal{L}_2(y_2) + \mathsf{E}_{D_{r_2},\ldots,D_{r_3-1}}\Big[\cdots \Big[\min_{y_m:I_m+o_{r_m}\geq y_m\geq I_m+u_{r_m}} \{ c_{r_m}(y_m - I_m) + \mathcal{L}_m(y_m) + \mathsf{E}_{D_{r_m},\ldots,D_{r_{m+1}-1}}[B(I_{m+1})]\}\Big] \cdots \Big]\Big\}\Big]\Big\}\Bigg\}$$

Note that $P(m)$ is the optimal cost to be incurred when there should be scheduled exactly m setups.

Conjecture 6.1 *$P(m)$ is a convex function of m.*

Although there is no analytical proof yet, we have experimentally seen that Conjecture 6.1 holds.

If Conjecture 6.1 is correct, than we can find the optimal solution for Problem P by binary search which may partially eliminate exponential complexity of the problem.

6.7 Discussion

Our lower bounds clearly reduce the computational times. In LB_1, it is clear to understand that it eliminates the production schedules with excessive number of setups. Since the com1putational effort to compute LB_1 is only the sum of the setup costs of the corresponding periods, it saves the time to solve such production schedules for optimality. The LB_2 for a given production schedule will certainly be greater than LB_1 for the same schedule. Note that in LB_2 cost structure, the cost elements of LB_1 is already included. Computing LB_2 is more complicated than LB_2 since there are individual cycles costs and minimizers to be evaluated. Still, computational burden of LB_2 is much less tan, solving the same production schedule for optimality and eases to solve the Problem P. Our new lower bound, LB_3 is dependent on many factors in terms of performance. LB_3 is a stronger lower bound over LB_1 for certainty; however, we can not definitely compare it to LB_2 all the time. Especially, the lot size restrictions play an important role on its success. On the other hand, there is no doubt that it has the most computational burden, since there needs another DP to be solved. Nevertheless, LB_3 uses linear approximations while taking the convolution of the cycles, which creates the major computational load. By eliminating that, LB_3 is still time saving compared to solving a production schedule for optimality.

Özen et al. (2012) defines AH in our extended setting and postulates that it would produce accurate results. In our study, we see that it is time efficient, however, yields the worst results in terms of optimality gap. For AH and the other Dynamic Programming Based heuristics, performance essentially depends on capacities. In Özen et al. (2012) where capacity restrictions are not present, AH is a heuristic yielding no more than 2.36% of optimality error. In our study, the cases where capacity is not tight, similar results are obtained. However, for cases with tight capacity, we observe maximum errors more than 30%. This is because the initial inventory levels that we use to solve the particular problems, and the actual values to occur, when the optimal production schedule is applied, do not match. AH simply assumes zero inventories at the beginning of the periods. For tight capacity cases, this makes it to short to attain the actual optimal after production inventory levels. Although more accurate, AH I and AH II also exhibit the same inability by resulting in base stock levels lower than they should be. And this raises the overall costs by causing backlog.

Performance of Combinatorial Based Heuristics seems significantly better than DP based heuristic procedures. Whichever the method is used among these type of heuristics, our search starts with either having 1 or N production cycles. The search than moves to a production schedule with one less or more cycle according to the procedure. Either way, the combinatorial search follows only one way: up or down in terms of number of production cycles. The problem is that, this procedure gets stuck at a certain production schedule which is offered as the final solution. What we see in our experiments is that the production schedule where the procedure gets stuck results far from optimality only when the capacity is high. And these cases are likely to have few setups at optimality. In that sense, MMI and MMII start their search further away from the optimal production schedule in terms of number of setups. Therefore, it is more likely for them to get stuck in a production schedule with unsatisfactory expected cost. For DMI and DMII, since the start off production schedule has only one setup, the cost of the schedule where the algorithm stops does not occur so different from the optimal schedule. In the other cases, optimality gap occurrence is only due to the greedy nature and narrow sight of the algorithm in terms of the production schedules spanned.

CHAPTER 7

CONCLUSION AND EXTENSIONS

In our study, we worked with a pure cost model of the stochastic lot sizing problem under static dynamic uncertainty strategy with constraints on the lot sizes. We have followed an exhaustive search for finding optimal policy parameters, therefore we adopted two existing lower bounds and proposed one different lower bound to reduce the span of the search. In an experimental setup, we see that our lower bound is stronger than the other lower bounds when the capacity is low. Additionally, it reduces the overall effect to solve Problem P for optimality, even though its computational burden is higher.

We also proposed various heuristic procedures can be classified under two families: Dynamic Programming Based and Combinatorial Based. We testes the accuracy of these procedure with another experimental setup. Although there is no theoretic proof; we observed that in terms of upper limits on the lot sizes, and these two families are complementary. Therefore, combining one heuristic form each family together produced satisfactory results. In order to find the best combination, we applied statistical tests to the results that heuristics yield. At the end, we found combinations yielding less than 1% error for stationary capacity cases, and around 3% error for dynamic capacity cases and still polynomial in time. Then, we applied paired-t test to our results, in order to point out which heuristic procedure alone produces the least optimality gaps. We also made use of the same technique to identify the best combination of DP and combinatorial heuristics to use.

Although our model extends Özen et al. (2012) in terms of cost and capacities, there are still other possible extensions motivated by practice. We assume we have capacity restrictions for lot sizes and we agree that they can be varying in time. However; in such environment, there is no guarantee that this varying capacity will be deterministic. Deviations in the capacity from period to period may be due to many factors that cannot be controlled by the decision maker of the lot sizing problem, that it can be known only when the production period starts. Therefore, Problem P can be re-modelled for stochastic capacities and be analyzed over.

Our model ignores the lead time for lot production, taking it as zero. This is also not in line with real practise. In reality, the lead time should be proportional to the lot sizes. Adding lead times would change the setting of the entire model. Time dimension, used to be only considered as periods, will have to be analyzed in more detail. The convexity issues about cycle costs should be checked for appropriateness. Even the backlogging scheme would be different.

We have considered a pure cost model in our study by assigning a per unit backlog cost. Following the relevant literature, service level constrained model may also be studied. In Özen et al. (2012), service level constrained version of Problem P is also studied and AH is applied. The results are found to be satisfactory; however, they can still be extended to the version that we have studied in the manuscript.

REFERENCES

Akbalik, A., C. Rapine. 2012. Polynomial time algorithms for the constant capacitated single-item lot sizing problem with stepwise production cost. *Operation Research Letters* **40** 390–397.

Bitran, G.R., H.H. Yanase. 1982. Computational complexity of the capacitated lot size problem. *Management Science* **28**(10) 1174–1186.

Bookbinder, J. H., J. Y. Tan. 1988. Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science* **34** 1096–1108.

Guan, Y., A.J. Miller. 2008. Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research* **56**(5) 1172–1183.

Halman, N., D. Klabjan, M. Mostagir, J. Orlin, D. Simchi-Levi. 2009. A fully polynomial-time approximation scheme for single-item stochastic inventory control with discrete demand. *Mathematics of Operations Research* **34** 674–685.

Huang, K., S. Kucukyavuz. 2008. On stochastic lot-sizing problems with random lead times. *Operations Research Letters* **36** 303–308.

Okhrin, I., K. Richter. 2011. An $o(t^3)$ algorithm for the capacitated lot sizing problem with minimum order quantities. *European Journal of Operation Research* **211** 507–514.

Önal, M., W. van den Heuvel, T. Liu. 2012. A note on "the economic lot sizing problem with inventory bounds". *European Journal of Operation Research* **223** 290–294.

Özen, U., M.K. Dogru, S.A. Tarim. 2012. Static dynamic uncertainty strategy for single-item stochastic inventory control problem. *Omega* **30** 348–357.

Scarf, H. 1959. Optimality of (s,s) in the dynamic inventory problem. Tech. rep., Applied Mathematics and Statistics Laboratory Stanford University.

Sethi, S. P., F. Cheng. 1997. Optimality of (s, s) policies in inventory models with markovian demand. *Operations Research* **45** 931–939.

Sox, C. R. 1997. Dynamic lot sizing with random demand and non-stationary costs. *Operations Research Letters* **20** 155–164.

Sox, C.R., J.A. Muckstadt. 1997. Optimization-based planning for the stochastic lot-scheduling problem. *IIE Transactions* **29** 349–357.

Tarim, S. A., M. K. Doğru, U. Özen, R. Rossi. 2009. An efficient computational method for non-stationary (r, s) inventory policy with service level constraints. *Working Paper*, Alcatel-Lucent Bell Labs.

Tarim, S. A., B. G. Kingsman. 2004. The stochastic dynamic production/inventory lot-sizing problem with service-level constraints. *International Journal of Production Economics* **88** 105–119.

Tarim, S. A., B. G. Kingsman. 2006. Modelling and computing (r^n, s^n) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research* **174** 581–599.

Vargas, V. 2009. An optimal solution for the stochastic version of the wagner whitin dynamic lot-size model. *European Journal of Operational Research* **198** 447–451.

Veinott, A., H. Wagner. 1965. Computing optimal (s, s) inventory policies. *Management Science* **11** 525–552.

Wagner, H.M., T.M. Whitin. 1958. Dynamic version of the economic lot size model. *Management Science* **5**(1) 88–96.

Zangwill, W.I. 1969. A backlogging model and a multi-echelon model of a dynamic economic lot size production system-a network approach. *Management Science* **15**(9) 506–527.

Appendix A

Heuristic Accuracy Tables

Table A.1: Summary of Heuristics $N = 18$

	AH	AH I	AH II				Merge		Divide	
			n=1	n=2	n=3	n=4	Method I	Method II	Method I	Method II
# of < 5% gap	645	715	699	703	709	714	720	720	720	720
# of < 2% gap	573	667	608	653	669	668	715	714	720	720
# of < 1% gap	514	628	518	587	611	629	701	697	714	717
# of optimal	308	491	312	353	370	439	598	583	580	579
max gap (%)	34.72	15.83	14.60	23.18	19.75	24.82	2.34	4.16	1.45	1.45
ave gap (%)	1.61	0.41	0.90	0.78	0.57	0.45	0.08	0.11	0.06	0.06

Table A.2: Summary of Heuristics $N = 24$

	AH	AH I	AH II				Merge		Divide	
			n=1	n=2	n=3	n=4	Method I	Method II	Method I	Method II
# of < 5% gap	648	707	694	691	703	712	720	720	720	720
# of < 2% gap	545	654	569	592	623	648	720	715	720	720
# of < 1% gap	479	593	443	506	554	580	716	708	714	716
# of optimal	293	401	226	254	280	326	536	511	539	536
max gap (%)	35.64	27.12	13.26	24.80	19.13	19.22	1.52	3.90	1.52	1.52
ave gap (%)	1.76	0.57	1.20	1.10	0.80	0.60	0.07	0.11	0.08	0.07

Table A.3: Summary of Heuristics $N = 30$

	AH	AH I	AH II				Merge		Divide	
			n=1	n=2	n=3	n=4	Method I	Method II	Method I	Method II
# of < 5% gap	632	706	672	673	702	713	720	720	720	720
# of < 2% gap	521	644	544	590	602	627	708	705	720	720
# of < 1% gap	450	585	411	475	522	529	694	685	710	714
# of optimal	213	317	182	218	228	263	513	486	514	519
max gap (%)	35.91	28.36	13.35	23.50	22.61	17.24	3.85	4.93	1.25	1.08
ave gap (%)	1.97	0.65	1.38	1.25	0.96	0.75	0.13	0.17	0.10	0.09

Table A.4: Summary of Heuristics $N = 36$

	AH	AH II				Merge		Divide	
		n=1	n=2	n=3	n=4	Method I	Method II	Method I	Method II
# of < 5% gap	624	678	686	693	708	720	720	720	720
# of < 2% gap	503	551	572	593	610	709	708	718	719
# of < 1% gap	439	432	462	512	531	694	685	714	716
# of optimal	219	178	210	246	234	469	462	489	502
max gap (%)	36.38	12.78	24.71	20.79	14.50	4.53	4.53	2.32	2.03
ave gap (%)	2.09	1.35	1.30	1.01	0.83	0.15	0.17	0.10	0.10

Table A.5: Minimum of Two Heuristics $N = 18$

	MM I	MM I	MM I	MM II	MM II	MM II	DM I	DM I	DM I	DM II	DM II	DM II
	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
# of < 5% gap	720	720	720	720	720	720	720	720	720	720	720	720
# of < 2% gap	720	720	720	720	720	720	720	720	720	720	720	720
# of < 1% gap	718	720	720	720	720	720	720	720	720	720	720	720
# of optimal	652	676	684	692	674	687	657	687	689	661	685	686
max gap (%)	1.17	0.94	0.62	0.94	0.94	0.62	1.00	0.66	1.00	1.00	0.44	1.00
ave gap (%)	0.03	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.02	0.01	0.01

Table A.6: Minimum of Two Heuristics $N = 24$

	MM I	MM I	MM I	MM II	MM II	MM II	DM I	DM I	DM I	DM II	DM II	DM II
	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
# of < 5% gap	720	720	720	720	720	720	720	720	720	720	720	720
# of < 2% gap	720	720	720	720	720	720	720	720	720	720	720	720
# of < 1% gap	720	720	720	720	720	718	720	720	720	720	720	720
# of optimal	650	660	667	682	655	650	653	665	671	650	660	667
max gap (%)	0.96	0.44	0.40	0.29	0.82	1.06	0.96	0.64	0.59	0.96	0.44	0.40
ave gap (%)	0.02	0.01	0.01	0.00	0.02	0.02	0.02	0.01	0.01	0.02	0.01	0.01

Table A.7: Minimum of Two Heuristics $N = 30$

	MM I	MM I	MM I	MM II	MM II	MM II	DM I	DM I	DM I	DM II	DM II	DM II
	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
# of < 5% gap	720	720	720	720	720	720	720	720	720	720	720	720
# of < 2% gap	715	720	719	720	720	718	720	720	720	720	720	720
# of < 1% gap	712	720	714	720	718	710	720	720	720	720	720	720
# of optimal	599	620	638	642	600	620	614	616	655	619	626	660
max gap	3.13	0.90	2.10	0.79	1.41	2.10	0.77	0.64	0.67	0.93	0.89	0.67
ave gap	0.05	0.02	0.03	0.01	0.03	0.04	0.02	0.02	0.01	0.02	0.02	0.01

Appendix B

Paired t-test Results

B.1 Heuristic Performance Comparisons for Constant Capacity

Table B.1: Paired T-Test and CI: AH; AH II n=1

	N	Mean	StDev	SE Mean
AH	720	0.01360	0.03434	0.00128
AH II n=1	720	0.00498	0.01484	0.00055
Difference	720	0.00862	0.03135	0.00117
95% lower bound for mean difference: 0.00669				
T-Test of mean difference = 0 (vs > 0): T-Value = 7.38 P-Value = 0.000				

Table B.2: Paired T-Test and CI: AH; AH II n=2

	N	Mean	StDev	SE Mean
AH	720	0.01360	0.03434	0.00128
AH II n=2	720	0.00511	0.02147	0.00080
Difference	720	0.00849	0.03763	0.00140
95% lower bound for mean difference: 0.00618				
T-Test of mean difference = 0 (vs > 0): T-Value = 6.05 P-Value = 0.000				

Table B.3: Paired T-Test and CI: AH II n=1; AH II n=2

	N	Mean	StDev	SE Mean
AH II n=1	720	0.004978	0.014843	0.000553
AH II n=2	720	0.005105	0.021474	0.000800
Difference	720	-0.000127	0.013860	0.000517
95% CI for mean difference: (-0.001141; 0.000887)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -0.25 P-Value = 0.806				

Table B.4: Paired T-Test and CI: AH II n=1; AH II n=3

	N	Mean	StDev	SE Mean
AH II n=1	720	0.004978	0.014843	0.000553
AH II n=3	720	0.004126	0.013882	0.000517
Difference	720	0.000852	0.009433	0.000352
95% lower bound for mean difference: 0.000273				
T-Test of mean difference = 0 (vs > 0): T-Value = 2.42 P-Value = 0.008				

Table B.5: Paired T-Test and CI: AH II n=3; AH II n=4

	N	Mean	StDev	SE Mean
AH II n=3	720	0.004126	0.013882	0.000517
AH II n=4	720	0.002737	0.009448	0.000352
Difference	720	0.001389	0.010010	0.000373
95% lower bound for mean difference: 0.000775				
T-Test of mean difference = 0 (vs > 0): T-Value = 3.72 P-Value = 0.000				

Table B.6: Paired T-Test and CI: AH II n=4; MMI

	N	Mean	StDev	SE Mean
AH II n=4	720	0.002737	0.009448	0.000352
MMI	720	0.000488	0.002214	0.000083
Difference	720	0.002248	0.009810	0.000366
95% CI for mean difference: (0.001531; 0.002966)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 6.15 P-Value = 0.000				

Table B.7: Paired T-Test and CI: MMI; MMII

	N	Mean	StDev	SE Mean
MMI	720	0.000488	0.002214	0.000083
MMII	720	0.000639	0.002801	0.000104
Difference	720	-0.000151	0.002551	0.000095
95% CI for mean difference: (-0.000338; 0.000036)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -1.59 P-Value = 0.112				

Table B.8: Paired T-Test and CI: MMII; DMI

	N	Mean	StDev	SE Mean
MMII	720	0.000639	0.002801	0.000104
DMI	720	0.000597	0.002087	0.000078
Difference	720	0.000043	0.003049	0.000114
95% CI for mean difference: (-0.000180; 0.000266)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.38 P-Value = 0.707				

Table B.9: Paired T-Test and CI: DMI; DMII

	N	Mean	StDev	SE Mean
DMI	720	0.000597	0.002087	0.000078
DMII	720	0.000457	0.001666	0.000062
Difference	720	0.000139	0.001574	0.000059
95% lower bound for mean difference: 0.000043				
T-Test of mean difference = 0 (vs > 0): T-Value = 2.38 P-Value = 0.009				

B.2 Heuristic Combinations Performance Comparisons for Constant Capacity

Table B.10: Paired T-Test and CI: MM I-AH; MMI-AH I

	N	Mean	StDev	SE Mean
MM I-AH	720	0.000129	0.000779	0.000029
MMI-AH I	720	0.000050	0.000433	0.000016
Difference	720	0.000079	0.000812	0.000030
95% lower bound for mean difference: 0.000030				
T-Test of mean difference = 0 (vs > 0): T-Value = 2.62 P-Value = 0.004				

Table B.11: Paired T-Test and CI: MMI-AH I; MMI-AH II

	N	Mean	StDev	SE Mean
MMI-AH I	720	0.000050	0.000433	0.000016
MMI-AH II	720	0.000044	0.000414	0.000015
Difference	720	0.000006	0.000133	0.000005
95% CI for mean difference: (-0.000004; 0.000015)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 1.16 P-Value = 0.245				

Table B.12: Paired T-Test and CI: MMII-AH I; MMII-AH

	N	Mean	StDev	SE Mean
MMII-AH I	720	0.000070	0.000535	0.000020
MMII-AH	720	0.000017	0.000205	0.000008
Difference	720	0.000053	0.000494	0.000018
95% lower bound for mean difference: 0.000022				
T-Test of mean difference = 0 (vs > 0): T-Value = 2.85 P-Value = 0.002				

Table B.13: Paired T-Test and CI: MMII-AH I; MMII-AH II

	N	Mean	StDev	SE Mean
MMII-AH I	720	0.000070	0.000535	0.000020
MMII-AH II	720	0.000049	0.000433	0.000016
Difference	720	0.000021	0.000317	0.000012
95% CI for mean difference: (-0.000002; 0.000044)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 1.78 P-Value = 0.075				

Table B.14: Paired T-Test and CI: MMI-AH II; MMII-AH I

	N	Mean	StDev	SE Mean
MMI-AH II	720	0.000044	0.000414	0.000015
MMII-AH I	720	0.000070	0.000535	0.000020
Difference	720	-0.000026	0.000413	0.000015
95% CI for mean difference: (-0.000056; 0.000005)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -1.66 P-Value = 0.097				

Table B.15: Paired T-Test and CI: DMI-AH; DMI-AH I

	N	Mean	StDev	SE Mean
DMI-AH	720	0.000093	0.000646	0.000024
DMI-AH I	720	0.000042	0.000506	0.000019
Difference	720	0.000051	0.000791	0.000029
95% lower bound for mean difference: 0.000003				
T-Test of mean difference = 0 (vs > 0): T-Value = 1.74 P-Value = 0.041				

Table B.16: Paired T-Test and CI: DMI-AH I; DMI-AH II

	N	Mean	StDev	SE Mean
DMI-AH I	720	0.000042	0.000506	0.000019
DMI-AH II	720	0.000036	0.000394	0.000015
Difference	720	0.000007	0.000334	0.000012
95% CI for mean difference: (-0.000018; 0.000031)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.54 P-Value = 0.587				

Table B.17: Paired T-Test and CI: MMII-AH II; DMI-AH

	N	Mean	StDev	SE Mean
MMII-AH II	720	0.000049	0.000433	0.000016
DMI-AH	720	0.000093	0.000646	0.000024
Difference	720	-0.000045	0.000765	0.000029
95% CI for mean difference: (-0.000101; 0.000011)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -1.57 P-Value = 0.118				

Table B.18: Paired T-Test and CI: DMII-AH; DMII-AH I

	N	Mean	StDev	SE Mean
DMII-AH	720	0.000075	0.000579	0.000022
DMII-AH I	720	0.000058	0.000539	0.000020
Difference	720	0.000018	0.000694	0.000026
95% CI for mean difference: (-0.000033; 0.000068)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.68 P-Value = 0.495				

Table B.19: Paired T-Test and CI: DMII-AH I; DMII-AH II

	N	Mean	StDev	SE Mean
DMII-AH I	720	0.000058	0.000539	0.000020
DMII-AH II	720	0.000032	0.000312	0.000012
Difference	720	0.000026	0.000432	0.000016
95% CI for mean difference: (-0.000006; 0.000057)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 1.60 P-Value = 0.110				

Table B.20: Paired T-Test and CI: DMI-AH II;DMI-AH II

	N	Mean	StDev	SE Mean
DMI-AH II	720	0.000036	0.000394	0.000015
DMI-AH II	720	0.000075	0.000579	0.000022
Difference	720	-0.000040	0.000655	0.000024
95% CI for mean difference: (-0.000088; 0.000008)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -1.63 P-Value = 0.104				

B.3 Heuristic Performance Comparisons for Varying Capacity

Table B.21: Paired T-Test and CI: AH; AH II n=1

	N	Mean	StDev	SE Mean
AH	8640	0.037019	0.063425	0.000682
AH II n=1	8640	0.019449	0.044355	0.000477
Difference	8640	0.017570	0.048465	0.000521
95% lower bound for mean difference: 0.016712				
T-Test of mean difference = 0 (vs > 0): T-Value = 33.70 P-Value = 0.000				

Table B.22: Paired T-Test and CI: AH II n=1; AH II n=2

	N	Mean	StDev	SE Mean
AH II n=1	8640	0.019449	0.044355	0.000477
AH II n=2	8640	0.015242	0.033813	0.000364
Difference	8640	0.004207	0.027248	0.000293
95% lower bound for mean difference: 0.003725				
T-Test of mean difference = 0 (vs > 0): T-Value = 14.35 P-Value = 0.000				

Table B.23: Paired T-Test and CI: AH II n=2; AH I

	N	Mean	StDev	SE Mean
AH II n=2	8640	0.015242	0.033813	0.000364
AH I	8640	0.011787	0.034283	0.000369
Difference	8640	0.003455	0.039744	0.000428
95% lower bound for mean difference: 0.002751				
T-Test of mean difference = 0 (vs > 0): T-Value = 8.08 P-Value = 0.000				

Table B.24: Paired T-Test and CI: AH I; AH II n=3

	N	Mean	StDev	SE Mean
AH I	8640	0.011787	0.034283	0.000369
AH II n=3	8640	0.011201	0.024566	0.000264
Difference	8640	0.000587	0.034222	0.000368
95% CI for mean difference: (-0.000135; 0.001308)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 1.59 P-Value = 0.111				

Table B.25: Paired T-Test and CI: AH II n=3; AH II n=4

	N	Mean	StDev	SE Mean
AH II n=3	8640	0.011201	0.024566	0.000264
AH II n=4	8640	0.009760	0.024809	0.000267
Difference	8640	0.001440	0.023464	0.000252
95% lower bound for mean difference: 0.001025				
T-Test of mean difference = 0 (vs > 0): T-Value = 5.71 P-Value = 0.000				

Table B.26: Paired T-Test and CI: AH II n=4; MMII

	N	Mean	StDev	SE Mean
AH II n=4	8640	0.009760	0.024809	0.000267
MMII	8640	0.003710	0.017545	0.000189
Difference	8640	0.006050	0.030432	0.000327
95% lower bound for mean difference: 0.005512				
T-Test of mean difference = 0 (vs > 0): T-Value = 18.48 P-Value = 0.000				

Table B.27: Paired T-Test and CI: MMII; MMI

	N	Mean	StDev	SE Mean
MMII	8640	0.003710	0.017545	0.000189
MMI	8640	0.002409	0.012506	0.000135
Difference	8640	0.001302	0.014848	0.000160
95% lower bound for mean difference: 0.001039				
T-Test of mean difference = 0 (vs > 0): T-Value = 8.15 P-Value = 0.000				

Table B.28: Paired T-Test and CI: MMI; DMI

	N	Mean	StDev	SE Mean
MMI	8640	0.002409	0.012506	0.000135
DMI	8640	0.000521	0.002609	0.000028
Difference	8640	0.001887	0.012728	0.000137
95% lower bound for mean difference: 0.001662				
T-Test of mean difference = 0 (vs > 0): T-Value = 13.78 P-Value = 0.000				

Table B.29: Paired T-Test and CI: DMI; DMII

	N	Mean	StDev	SE Mean
DMI	8640	0.000521	0.002609	0.000028
DMII	8640	0.000484	0.002527	0.000027
Difference	8640	0.000037	0.001076	0.000012
95% lower bound for mean difference: 0.000018				
T-Test of mean difference = 0 (vs > 0): T-Value = 3.21 P-Value = 0.001				

B.4 Heuristic Combination Performance Comparisons for Varying Capacity

Table B.30: Paired T-Test and CI: MMI-AH; MMI-AH I

	N	Mean	StDev	SE Mean
MMI-AH	8640	0.001365	0.008844	0.000095
MMI-AH I	8640	0.000516	0.004177	0.000045
Difference	8640	0.000849	0.007749	0.000083
95% lower bound for mean difference: 0.000711				
T-Test of mean difference = 0 (vs > 0): T-Value = 10.18 P-Value = 0.000				

Table B.31: Paired T-Test and CI: MMI-AH I; MMI-AH II

	N	Mean	StDev	SE Mean
MMI-AH I	8640	0.000516	0.004177	0.000045
MMI-AH II	8640	0.000558	0.004414	0.000047
Difference	8640	-0.000042	0.002315	0.000025
95% CI for mean difference: (-0.000091; 0.000007)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -1.69 P-Value = 0.092				

Table B.32: Paired T-Test and CI: MMII-AH I; MMII-AH

	N	Mean	StDev	SE Mean
MMII-AH I	8640	0.000639	0.004479	0.000048
MMII-AH	8640	0.000410	0.003654	0.000039
Difference	8640	0.000228	0.002488	0.000027
95% lower bound for mean difference: 0.000184				
T-Test of mean difference = 0 (vs > 0): T-Value = 8.54 P-Value = 0.000				

Table B.33: Paired T-Test and CI: MMII-AH I; MMII-AH II

	N	Mean	StDev	SE Mean
MMII-AH I	8640	0.000639	0.004479	0.000048
MMII-AH II	8640	0.000676	0.004667	0.000050
Difference	8640	-0.000038	0.002373	0.000026
95% CI for mean difference: (-0.000088; 0.000012)				
T-Test of mean difference = 0 (vs not = 0): T-Value = -1.47 P-Value = 0.141				

Table B.34: Paired T-Test and CI: DMI-AH; DMI-AH I

	N	Mean	StDev	SE Mean
DMI-AH	8640	0.000262	0.002064	0.000022
DMI-AH I	8640	0.000119	0.001094	0.000012
Difference	8640	0.000143	0.001798	0.000019
95% lower bound for mean difference: 0.000111				
T-Test of mean difference = 0 (vs > 0): T-Value = 7.40 P-Value = 0.000				

Table B.35: Paired T-Test and CI: DMI-AH I; DMI-AH II

	N	Mean	StDev	SE Mean
DMI-AH I	8640	0.000119	0.001094	0.000012
DMI-AH II	8640	0.000117	0.001114	0.000012
Difference	8640	0.000002	0.000516	0.000006
95% CI for mean difference: (-0.000009; 0.000013)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.34 P-Value = 0.735				

Table B.36: Paired T-Test and CI: MMI-AH II; DMI-AH

	N	Mean	StDev	SE Mean
MMI-AH II	8640	0.000558	0.004414	0.000047
DMI-AH	8640	0.000262	0.002064	0.000022
Difference	8640	0.000296	0.004789	0.000052
95% lower bound for mean difference: 0.000212				
T-Test of mean difference = 0 (vs > 0): T-Value = 5.75 P-Value = 0.000				

Table B.37: Paired T-Test and CI: MMII-AH II; DMI-AH

	N	Mean	StDev	SE Mean
MMII-AH II	8640	0.000676	0.004667	0.000050
DMI-AH	8640	0.000262	0.002064	0.000022
Difference	8640	0.000414	0.005022	0.000054
95% lower bound for mean difference: 0.000326				
T-Test of mean difference = 0 (vs > 0): T-Value = 7.67 P-Value = 0.000				

Table B.38: Paired T-Test and CI: DMII-AH; DMII-AH I

	N	Mean	StDev	SE Mean
DMII-AH	8640	0.000243	0.002031	0.000022
DMII-AH I	8640	0.000117	0.001097	0.000012
Difference	8640	0.000126	0.001766	0.000019
95% lower bound for mean difference: 0.000095				
T-Test of mean difference = 0 (vs > 0): T-Value = 6.65 P-Value = 0.000				

Table B.39: Paired T-Test and CI: DMII-AH I; DMII-AH II

	N	Mean	StDev	SE Mean
DMII-AH I	8640	0.000117	0.001097	0.000012
DMII-AH II	8640	0.000115	0.001107	0.000012
Difference	8640	0.000002	0.000522	0.000006
95% CI for mean difference: (-0.000009; 0.000013)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.36 P-Value = 0.719				

Table B.40: Paired T-Test and CI: MMI-AH II; DMII-AH

	N	Mean	StDev	SE Mean
MMI-AH II	8640	0.000558	0.004414	0.000047
DMII-AH	8640	0.000243	0.002031	0.000022
Difference	8640	0.000315	0.004775	0.000051
95% lower bound for mean difference: 0.00023				
T-Test of mean difference = 0 (vs > 0): T-Value = 6.13 P-Value = 0.000				

Table B.41: Paired T-Test and CI: MMII-AH II; DMII-AH

	N	Mean	StDev	SE Mean
MMII-AH II	8640	0.000676	0.004667	0.000050
DMII-AH	8640	0.000243	0.002031	0.000022
Difference	8640	0.000433	0.005008	0.000054
95% lower bound for mean difference: 0.000345				
T-Test of mean difference = 0 (vs > 0): T-Value = 8.04 P-Value = 0.000				

Table B.42: Paired T-Test and CI: DMI-AH I; DMII-AH I

	N	Mean	StDev	SE Mean
DMI-AH I	8640	0.000119	0.001094	0.000012
DMII-AH I	8640	0.000117	0.001097	0.000012
Difference	8640	0.000002	0.000390	0.000004
95% CI for mean difference: (-0.000006; 0.000010)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.50 P-Value = 0.618				

Table B.43: Paired T-Test and CI: DMI-AH I; DMII-AH II

	N	Mean	StDev	SE Mean
DMI-AH I	8640	0.000119	0.001094	0.000012
DMII-AH II	8640	0.000115	0.001107	0.000012
Difference	8640	0.000004	0.000578	0.000006
95% CI for mean difference: (-0.000008; 0.000016)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.66 P-Value = 0.508				

Table B.44: Paired T-Test and CI: DMI-AH II; DMII-AH I

	N	Mean	StDev	SE Mean
DMI-AH II	8640	0.000117	0.001114	0.000012
DMII-AH I	8640	0.000117	0.001097	0.000012
Difference	8640	0.000000	0.000596	0.000006
95% CI for mean difference: (-0.000012; 0.000013)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.03 P-Value = 0.973				

Table B.45: Paired T-Test and CI: DMI-AH II; DMII-AH II

	N	Mean	StDev	SE Mean
DMI-AH II	8640	0.000117	0.001114	0.000012
DMII-AH II	8640	0.000115	0.001107	0.000012
Difference	8640	0.000002	0.000273	0.000003
95% CI for mean difference: (-0.000004; 0.000008)				
T-Test of mean difference = 0 (vs not = 0): T-Value = 0.76 P-Value = 0.447				