Two-Echelon Supply Chain Design for Spare Parts with Time Constraints

by

Muhammad Waqas Riaz

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We consider a single-part, two-echelon supply chain problem for spare parts. The network consists of a single manufacturing plant, a set of service centers (SCs) and a set of customers. Both echelons keep spare parts using the base-stock replenishment policy. The plant behaves as an M/M/1 queueing system and has limited production and storage capacity. Demand faced by each SC follows an independent Poisson process. The problem is to determine optimal location-allocation and optimal base-stock levels at both echelons while satisfying the target service levels and customer preferences of SCs. We develop a mixed integer non-linear programming model and use cutting-plane method to optimize the inventory-location decisions. We present an exact solution procedure for the inventory stocking problem and demonstrate the limitations of using traditional inventory models like METRIC-like and Approximate in case of high utilization rates. We show the effectiveness of our proposed cutting-plane algorithm and provide important managerial insights for spare parts management.

Acknowledgements

First and foremost, I am thankful to ALLAH almighty for giving me the strength, ability and will-power to complete this thesis. Without his blessings, this would not have been possible.

I would like to express my sincere and deepest gratitude to my supervisors Prof. Fatma Gzara & Prof. Elizabeth Jewkes for their extensive support, guidance and encouragement on this research work. I would also like to thank my committee members Prof. James Bookbinder & Prof. Hossein Abouee-Mehrizi for their constructive and valuable feedback. Besides my supervisors and readers, I would further like to extend my appreciation to Prof Samir Elhedhli for his motivation and encouragement earlier in my Master's and all the faculty, friends, fellow graduates and staff in the Department of Management Science & University of Waterloo.

I am grateful to my family (including in-laws) for their continuous support, prayers, patience and un-conditional love. Last but not the least; I am thankful to all my friends for their moral support and making my stay very special in Waterloo.

Dedication

This work is dedicated to my beloved parents, siblings and my loving wife.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we study a two-echelon supply chain design problem for spare parts with customer preferences and response time requirements. The research is motivated by the problem faced by Bombardier Inc. in designing its spare parts logistics system. The system operated by the company consists of a central manufacturing plant having limited production capacity in the first echelon, multiple service centers (SCs) in the second echelon and spatially dispersed customers. The SCs keep stock to fulfill customer demand and are replenished by the plant. Both the plant and SCs have limited storage capacity and operate using continuous review (one-for-one) replenishment policy. Demand faced by each SC follows an independent Poisson process. Customers are assigned to the SCs based on their preferences. Each customer requires a mean target response time (the average time between when the customer places an order and when the order is filled). The company wants to decide on the SC to open, the assignment of customers to SCs, and the base-stock levels at the plant and SCs. These decisions have to be made simultaneously since response time depends on the assigned SCs and the base-stock levels at both echelons.

The supply chain design problem we study is applicable to several other industries that have a significant spare parts business with low demand and high inventory holding cost. Spare parts inventories are different from raw materials, work-in process or finished goods inventories because spare parts are used to maintain the operations of any manufactured equipment. For example in industries like the aerospace, automotive, computer manufacturing, telecommunication networks and military, products are comprised of different interdependent parts. In such high technology products, failure of even a single component may lead to complete system failure. The difference between spare parts and other regular products arises because of maintenance policies, reliability information, failure process of parts, high costs, loss of production and obsolescence [33]. Thus special inventory management solutions are required as compared to generic supply chain policies for spare parts management.

Continuous review base-stock replenishment policy, referred to as (S-1,S) policy, is appropriate for managing inventory for spare parts which are characterized by infrequent and low demand, high inventory holding and shortage costs, and relatively low order setup costs. This policy has been used extensively in the literature for multi-echelon inventory systems [26, 30, 44, 45, 58, 62, 63, 66]. Moinzadeh and Lee [43] study the problem of determining optimal batch order size and stocking policy at all stocking echelons in multi-echelon inventory systems. Their results are in accordance with the practice of using base-stock policy for items with low demand and high inventory holding cost compared to ordering cost.

A service requirement between a manufacturer and customers is a key element in spare parts management. These requirements vary depending on the nature of the industry and products being manufactured. For spare parts, time-based service requirements are considered because customers are highly sensitive to response times. When there is a breakdown of any component that may result in an interruption in production, customers want to recover operations as quickly as possible. Therefore to ensure system reliability, different industries where customers are sensitive to the response times now use multi-echelon inventory control systems and keep supply of spare parts close to the customers so that they can maintain the target service levels.

One way to satisfy customer demand within the target response time is to stock ample supply of spare parts. Flint [23] points that the aviation industry stocks $45 billion worth of spare parts including 4000 spare engines. The industry incurs almost $2 Billion every year in maintenance cost for such a high valued inventory. Thus, maintaining ample supply is not an economical option as most of the spare parts are expensive, have erratic and low demand and require high maintenance. This has become a major challenge for supply chain professionals to trade-off between high inventory holding costs and maintaining customer service levels. Manufacturers can make significant savings using an efficient spare parts supply chain network designs, which incorporate customer service requirements in the design stage. Sherbrooke [64] finds that commercial airlines can achieve a 20% increase in target service levels with a 40% reduction in inventory holding costs using efficient supply chain designs.

Since supply chain networks are becoming more complex, competitive and integrated than ever, the facility location and inventory stocking decisions need to be made simultaneously rather than sequentially. The facility location and inventory stocking decisions are two main problems in supply chain network design [32]. Traditionally, the facility location decisions are made separately from inventory stocking decisions which usually results in sub-optimal supply chain designs [12, 14, 39, 61]. In the United States, inventories account for one-third of all assets of a typical company [17]. The Canadian manufacturing sector incurs 11% higher inventory holding costs as compared to the U.S.A whereas the retail sector suffers from 31% higher costs than in the U.S.A. [57]. Cohen et al. [13] show that maintenance contracts yield 63% of the after-sales service revenue whereas part sales account for 5%, and time and material contracts and internal service add up to 21%. These statistics suggest that inventory investments must be considered while designing any supply chain network. Therefore, there is a need to integrate both the facility location and inventory stocking decisions in order to design efficient supply chain systems.

The rest of the thesis is organized as follows: Chapter 2 presents a review of literature related to the facility location, inventory stocking, and inventory-location problems. Chapter 3 describes the formulation of the inventory-location problem. Chapter 4 presents the formulation of the inventory stocking problem, discusses existing models to calculate the distribution of the number of the outstanding orders at SCs, and gives an exact solution procedure to solve the inventory stocking problem. Chapter 5 proposes an exact cutting-plane algorithm for the inventory-location problem. We test our solution approach and report results in Chapter 6, perform sensitivity analysis in Chapter 7 and conclude the thesis in Chapter 8.

Chapter 2

Literature Review

This section reviews three different streams of research which are related to our work. This review is not meant to be exhaustive but we review the key contributions in these research areas. The first stream of research is on the facility location problem that focuses on locating facilities and allocating customers to open facilities to fulfill demand. The second stream is on the inventory stocking problem which determines base-stock levels and the third stream is on the inventory-location problem which incorporates the facility location and inventory stocking problems.

2.1 The Facility Location Problem

The facility location problem has been studied considerably in the Operations Research literature. The classical facility location problem determines the location of facilities and allocates customers to these locations to fulfill their demand. These models aim to minimize facility location and transportation costs and ignore inventory holding and shortage costs.

The P-median problem and the uncapacitated fixed-charge location problem (UFLP) are two classical facility location problems. In P-median problem, P facilities are to be selected on a network in order to minimize total demand-weighted distances whereas in UFLP, the model locates facilities and assigns customers to these facilities while minimizing facility location and transportation costs. Mirchandani and Francis [42], Daskin [15] and Drezner [18] provide a detailed review on location models and extensions. Snyder [67] reviews the location models under uncertainty, in which facilities may become unavailable due to unforeseen circumstances. Recently, Klose and Drexl [34], ReVelle and Eiselt [55] and ReVelle et al. [56] provide an extensive overview of facility location problems. For facility location problems in the context of supply chain management, the reader is referred to Owen and Daskin [50], and Melo et al. [38].

The location feature of our problem is a special case of a typical facility location problem because we allocate customers to facilities based on their preferences and not on minimum allocation cost.

2.2 The Inventory Stocking Problem

A rich literature exists on multi-echelon inventory stocking problems. These problems focus on finding optimal base-stock levels at all echelons so that inventory holding and backorder costs are minimized. Sherbrooke [62] presents the METRIC model, one of the most studied models in the multi-echelon inventory literature. He develops an approximation technique to minimize expected backorder level at each echelon in a two-echelon inventory system for recoverable items. The METRIC model assumes a Compound Poisson failure process, ample repair capacity and an (S-1,S) replenishment policy. Since then, many extensions of the METRIC model have been studied. Muckstadt [44] develops the MOD-METRIC model considering multi-indenture levels i.e., hierarchical parts structures. Slay [66] suggests to fit the negative binomial distribution to approximate the outstanding orders in the VARI-METRIC model. In his seminal work, Graves [26] presents a model to find the exact steady state distribution of the number of the outstanding orders at each SC assuming Compound Poisson failure processes and deterministic shipment times. He also suggests a two-moment approximation for the distribution of the number of outstanding orders under the assumption of Poisson failure processes, (S,S-1) replenishment policy and ample repair capacity. He shows that the two-moment approximation performs better than the METRIC approximation in terms of accuracy.

A key restricting assumption of these multi-echelon inventory models is "ample repair capacity". This is an unrealistic assumption for most of the modern business frameworks. It means there are no queueing effects and no item has to wait for repair as the repair process is modeled by an $M/G/\infty$ queue. In order to relax this assumption, Gross et al. [27; 28; 29], Albright [2], Albright and Gupta [3], Albright and Soni [4], Avsar and Zijm [5] and recently Wong et al. [71] present different queueing models. Gross et al. [27] study a multi-echelon repairable network with limited repair capacity for the first time. Their work is an extension of Mirasol [41], who study a single-echelon capacitated repairable system. Diaz and Fu [16] study the impact of limited repair capacity on inventory levels for different types of repair processes. They find that the ample repair capacity assumption may yield misleading results and underestimate the spare parts requirements for high utilization rates. They suggest using a double negative binomial approximation and demonstrate improvement over traditional models like the METRIC and Graves [26] in case of high utilization rates.

An important aspect of our inventory stocking problem is the response time requirements imposed by customer agreements. Caglar et al. [11] propose a heuristic to minimize total inventory holding cost at both echelons subject to a mean response time requirement. They use Lagrangian decomposition and show that their approach works well for relatively large scale problems. Their problem is a special case of Hopp et al. [31], who assume a general (r,Q) replenishment policy for the central warehouse and a continuous review (S-1,S) policy for the regional warehouses. Ettl et al. [22] develop a multi-echelon inventory model to optimize inventory investments while satisfying time-based customer service requirements. They use a queueing based approximation to incorporate actual lead times and use a conjugate gradient method [54] to find optimal solutions.

Kutanoglu [35] considers time-based service levels in a two-echelon distribution system. He allows emergency lateral shipments, a possibility of sharing inventory among local stocking locations whenever another local stocking location stocks out. He suggests that in service parts logistics, time-based fill rates are more appropriate than traditional fill rates as customers are sensitive to the time-based target service levels. Recently, Wong et al. [71] and Topan and Bayindir [69] develop greedy heuristic approaches in multi-product two-echelon spare parts inventory systems in order to minimize the system-wide inventory holding costs under aggregate mean response time service level. Caggiano et al. [10] suggest an efficient procedure to compute channel fill rates for multi-product, multi-echelon service parts inventory system. They define channel fill rate as the probability of fulfilling demand for a specific part at a specific location within a target response time. Muckstadt [46] provides an excellent review of multi-echelon inventory management.

The inventory feature of our problem is different from the papers mentioned so far in this section. None of the papers consider both time-based response requirements and limited repair capacity, whereas our problem assumes limited repair capacity and includes target response time constraints.

2.3 The Inventory-Location Problem

The inventory-location problem has attracted attention in the last decade and spawned a lot of research. It integrates the inventory stocking and facility location problems. The inventory-location problem aims to minimize the facility location, transportation and inventory holding costs.

Erlebacher and Meller [21] incorporate inventory holding costs in addition to facility location and transportation costs. They develop a heuristic approach to solve a two-echelon distribution problem. Nozick and Turnquist [48] include inventory holding costs in the fixed-charge facility location model by estimating a linear relationship between inventory holding costs and the number of distribution centers. They assume a base-stock policy and Poisson demand in a single-echelon inventory-distribution system. Nozick and Turnquist [49] extend this analysis to a multi-product two-echelon system where inventory is held at both echelons.

Daskin et al. [14] study three-echelon supply chain design which consists of a single supplier, a set of distribution centers and a set of retailers. They explicitly include cycle stock and safety-stock inventory holding costs in the UFLP. The costs include facility location costs, local delivery costs, cycle stock and safety-stock inventory holding costs. They develop a location model with risk pooling (LMRP) which aims to capture risk pooling effects, grouping the retailers to get significant inventory holding cost savings [20]. LMRP uses a (Q,r) inventory control strategy by assuming an economic order quantity (EOQ) based ordering policy. They present a non-linear integer-program and use a Lagrangian relaxation algorithm to solve the special case of their problem in which ratio of the mean to the variance of the demand distribution is identical for all the retailers. Shen et al. [61] study a model similar to Daskin et al. [14]. They use a set-covering formulation and develop a column generation based solution algorithm. Miranda and Garrido [39] analyze a model identical to LMRP. They develop a Lagrangian relaxation based heuristic approach and their results demonstrate the potential benefits of taking an integrated approach.

Different extensions and variations of LMRP have been studied and analyzed. Miranda and Garrido [40] and Ozsen et al. [51] extend LMRP to the capacitated warehouse location model with risk pooling (CLMRP) by including a stochastic capacity constraint. Balcik [7], Shen [59] and Vidyarthi et al. [70] present a multi-product case of LMRP. Shen and Daskin [60] include customer service consideration to the LRMP model of Shen et al. [61], while Snyder et al. [68] study the stochastic version of LMRP, stochastic location model with risk pooling (SLMRP). Gebennini et al. [24] introduce a dynamic version of LMRP and Ozsen et al. [52] extend CLMRP to the multi-sourcing capacitated inventory-location model with risk pooling (MCLMRP).

Candas and Kutanoglu [12] integrate location and inventory stocking decisions in a multi-product two-echelon setting. They minimize facility location, transportation and inventory holding costs while satisfying a system wide target service level. They demonstrate the potential benefits of integrating facility location, inventory level and corresponding variable fill rates. They linearize the non-linear integer programming model and solve reasonable size problems. Their model assumes infinite plant capacity, deterministic lead time, a base-stock replenishment policy, Poisson distribution for customer demand and first come first serve (FCFS) service discipline. Benjaafar et al. [8] consider a single-echelon joint demand allocation and inventory control problem in which inventory is only kept at SCs. They assume Poisson demand, stochastic production and supply lead time, limited production capacity, base-stock replenishment policy, and a FCFS policy to fill the orders. They develop a mixed integer linear program and present an exact solution procedure. Their goal is to optimize demand allocation and base-stock levels at each location while minimizing expected total costs. Abouee-Mehrizi et al. [1] study a two-echelon joint production-inventory problem. They aim to determine optimal base-stock levels at both echelons, optimal demand allocation to open facilities while minimizing inventory holding, backorder and transportation costs. They present a formulation of their problem using Flow-Unit approach by Axsater [6]. They assume queueing system at plant with limited production capacity, base-stock policy at both echelons, Poisson demand and FCFS basis to satisfy the orders.

All the papers mentioned so far in this section either consider deterministic replenishment time or they do not incorporate service levels or they are single-echelon inventory-location models. Nozick and Turnquist [49] is the only exception who study a model in which inventory is stored at both echelons, consider stochastic replenishment lead time and service considerations, whereas our problem also includes response time constraints.

The work most related to our problem is due to Mak and Shen [37]. They consider a two-echelon integrated inventory-location system for spare part items. They assume M/M/1 queueing system at the plant with limited production and storage capacity, base-stock policy for both echelons, FCFS approach for filling the outstanding orders, Poisson demand and deterministic shipment times between plant and SCs. They develop a nonlinear mixed-integer program and use Lagrangian relaxation to determine optimal location of SCs, optimal allocation of the customers to SCs, and optimal base-stock levels at both echelons while satisfying time-based service levels. Our work differs from that of Mak and Shen [37] in three main aspects. First, the allocation decisions are based on customer preferences and not on minimum allocation cost. Second, we develop an exact algorithm to solve the inventory stocking problem which is able to use any inventory model including the METRIC, and Exact and Approximate models [26]. Third, we propose a novel exact cutting-plane algorithm to solve the inventory-location problem.

The contribution of this work is three-fold. First, we incorporate customer preferences in our optimization model. Second, we present an iterative exact procedure to solve the inventory stocking problem. Third, to the best of our knowledge, this is the first work to propose an exact cutting-plane algorithm for inventory-location problem.

Chapter 3

The Inventory-Location Problem

3.1 Problem Description

The problem that we address is to design a single-part, two-echelon supply chain system for spare parts. It consists of a single manufacturing plant (the upper echelon), a number of SCs (the lower echelon) and a number of customers. The plant manufactures and stocks items to fill SC orders. The SCs, in turn, hold inventory to fulfill customer demand. Both the plant and SCs have limited storage capacity. Parts fail at each customer site according to a Poisson process independent of other sites. Customers have a preference ordering of the SCs for the purpose of parts replacement and are assigned to a SC based on their preferences.

The plant and SCs use a base-stock (S-1,S) replenishment policy: When a part fails at a customer site, the customer places an order from its assigned SC. If there is inventory on hand, the customer order is filled, and the SC orders a replacement part from the plant. The plant, if it has inventory, will immediately send one part to the SC and at the same time will trigger an order to produce another item. There is a fixed shipment lead time between the plant and the SC. If either the SC or the plant is out of stock, the item is backordered until a replacement part becomes available. Backorders are filled on a FCFS basis. The time between when the customer places an order and when the order is filled is referred to as the customer response time. Customers require a mean target response time. Figure 3.1 shows the replenishment process at the plant and SCs.

The manufacturing facility can produce items at a rate μ, and has processing times that are independent and exponentially distributed. Since the arrival rate of orders at the plant is the superposition of independent Poisson processes, the plant can be viewed as an M/M/1 queueing system. We are interested in the long-run behavior of this system, so we consider only its steady state behavior.

The problem is to locate the SCs, assign customers to the SCs, and determine base-stock levels at the SCs and the plant so that customer response time requirements are met and customer preferences are satisfied. The objective is to minimize total costs which include inventory holding costs at the plant and SCs, facility location costs associated with the SCs, and backorder costs at the SCs.

3.2 Problem Formulation

We formulate the problem using the following notation, some of which is also used in Mak and Shen [37].

Parameters:

I = Set of customers.

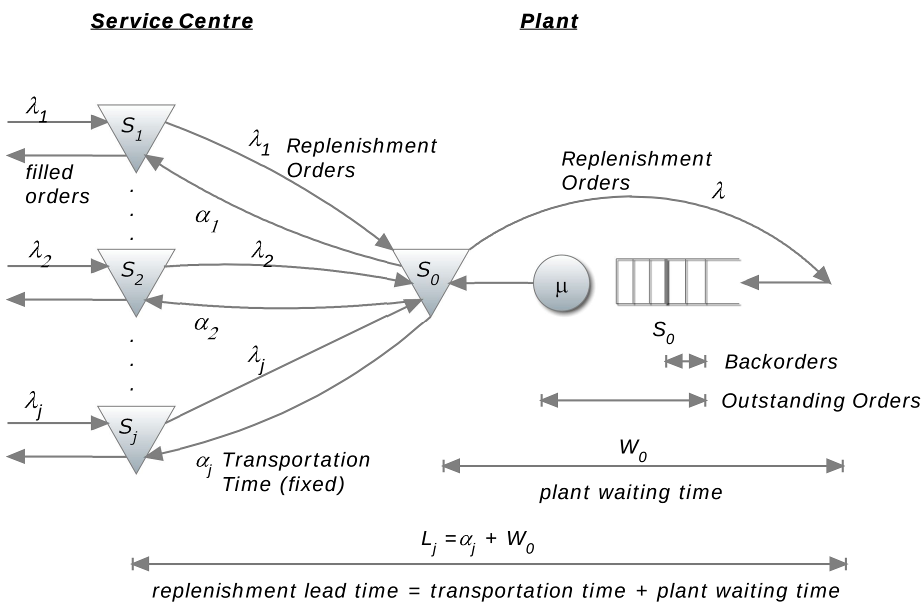

Figure 3.1: Replenishment process at the plant and SCs

J = Set of potential SC locations.

h_j = The unit inventory holding cost per unit time at SC $j, j \in J$.

p = The unit backorder cost per unit time.

f_j = The fixed cost of locating a SC $j, j \in J$.

λ_i = The demand rate of customer $i, i \in I$.

λ = The total demand rate at the plant $(= \sum_{i \in I} \lambda_i)$.

μ = The production rate at the plant.

ρ = The utilization rate of the plant$(= \lambda/\mu)$.

τ = The mean target response time.

d_{max} = The upper limit on the distance between a customer and the assigned SC.

J_i = The ordered list of SC locations indicating the preference of customer i (only those locations which are within d_{max} are considered).

α_j = The deterministic shipment lead time between the plant and SC $j, j \in J$.

C_0 = The storage capacity at the plant.

C_j = The storage capacity at SC $j, j \in J$.

Decision Variables:

X_j = 1 if SC j is opened, 0 otherwise, $j \in J$.

Y_{ij} = 1 if customer i is assigned to SC j, 0 otherwise, $i \in I, j \in J$.

S_j = The base-stock level at SC $j, j \in J$.

S_0 = The base-stock level at the plant.

Auxiliary Variables:

$\bar{I}_0$ = The steady state mean inventory level at the plant.

$\bar{B}_0$ = The steady state mean backorder level at the plant.

$\overline{W}_0$ = The steady state mean response time at the plant.

$\bar{I}_j$ = The steady state mean inventory level at SC $j, j \in J$.

$\bar{B}_j$ = The steady state mean backorder level at SC $j, j \in J$.

$\overline{W}_j$ = The steady state mean response time to a customer at SC $j, j \in J$.

$\bar{L}_j$ = The steady state mean replenishment lead time at SC $j, j \in J$.

Let Z be the total inventory holding and backorder cost in the system, $Z = \sum_{j \in J}\{(h_j \bar{I}_j + p\bar{B}_j) + h_o \bar{I}_0\}$. Using the notation described above, the inventory-location model is formulated as:

$$
\begin{aligned}
\text{[P]:} \quad \min \quad & \sum_{j \in J} f_j X_j + Z && && (3.1)\\
\text{s.t.} \quad & \sum_{j \in J} Y_{ij} = 1 && i \in I && (3.2)\\
& Y_{ij} \leqslant X_j && i \in I, j \in J_i && (3.3)\\
& Y_{ij} \geqslant X_j - \sum_{l=1}^{j-1} X_l && i \in I, j \in J_i && (3.4)\\
& S_j \leqslant C_j X_j && j \in \{0\} \cup J && (3.5)\\
& \overline{W}_j \leqslant \tau && j \in J && (3.6)\\
& S_j \geqslant 0, \ \text{integer} && j \in \{0\} \cup J && (3.7)\\
& Z \geqslant 0 && && (3.8)\\
& X_j \in \{0, 1\} && j \in J && (3.9)\\
& Y_{ij} \in \{0, 1\} && i \in I, j \in J && (3.10)
\end{aligned}
$$

The objective function (3.1) minimizes the sum of the inventory-location costs namely facility location costs, inventory holding and backorder costs at the SCs, and inventory holding costs at the plant. Constraints (3.2) ensure that all customers are assigned to SCs. Constraints (3.3) link Y_{ij} and X_j variables; they state that customers are only assigned to open SCs. Constraints (3.4)[1] ensure that the location preference with the smallest index

[1]These are closest assignment constraints, see Gerrard and Church [25].

among the available location preferences is selected (when j=1, the summation term drops and $Y_{ij} = X_j$ by (3.3) and (3.4)). Constraints (3.5) require that the base-stock level at a SC should be less than or equal to the storage capacity. Constraints (3.6) are the response time constraints. They ensure that the mean response time should not exceed the target response time. Constraints (3.8)–(3.10) are non-negativity and integer requirements.

The formulation [P] is different from Mak and Shen [37]. The primary difference lies in the assignment of customers. In [P], customers are assigned on the basis of customer preferences and not on the minimum transportation costs.

It is difficult to solve [P] directly using commercial software due to the complicating response time constraints (3.6). The mean response time $\overline{W}_j$ is a function of mean backorder level $\bar{B}_j$ and the demand faced by the SC j, whereas, $\bar{B}_j$ is a function of the base-stock level at SC j and the plant. In order to calculate the mean response time, we first need to find mean backorder and base-stock levels at the SCs and plant, which makes [P] a complex problem to solve. One approach to address this problem is to split these complicating response time requirements from other constraints using some decomposition technique. Mak and Shen [37] use Lagrangian relaxation to decompose the problem, whereas we propose an exact cutting-plane algorithm to handle the complicating constraints.

Before describing the solution algorithm, in the next chapter we derive the mean inventory and backorder level expressions required to calculate the mean response time.

Chapter 4

The Inventory Stocking Problem

The inventory stocking problem itself is of great significance for the spare parts industry. It has a similar problem setting and assumptions as the inventory-location problem described in Chapter 3. Given location and assignments of customers, the inventory stocking problem aims to find base-stock levels so that the inventory holding and backorder costs are minimized while satisfying storage capacity (3.5) and response time requirements (3.6).

4.1 Problem Formulation

Let $\widehat{J} \subseteq J = \{j : \widehat{X}_j = 1\}$ be the set of open SCs and $\widehat{Y}_{ij} \subseteq Y_{ij} = \{ij : \widehat{Y}_{ij} = 1\}$ be the set of customers allocated to $\widehat{J}$. For a given $\widehat{J}$ and $\widehat{Y}_{ij}$, [ISP] is formulated as:

$$\text{[ISP]:} \quad Z_j \quad = \quad \min \quad \sum_{j \in \bar{J}} (h_j \bar{I}_j + p\bar{B}_j) + h_o \bar{I}_0 \tag{4.1}$$

$$\text{s.t.}\quad S_j \leqslant C_j \widehat{X}_j \qquad j \in \{0\} \cup \widehat{J} \tag{4.2}$$

$$\overline{W}_j \leqslant \tau \qquad j \in \widehat{J} \tag{4.3}$$

$$S_j \geqslant 0,\ \text{integer} \qquad j \in \{0\} \cup \widehat{J} \tag{4.4}$$

The objective function (4.1) minimizes the sum of the inventory holding costs at the plant and inventory holding and backorder costs at the open SCs. Constraints (4.2) enforce the storage capacity, the base-stock level at an open SC should be less than or equal to the storage capacity. Constraints (4.3) are the response time constraints. They ensure that the mean response time at open SC should not exceed the target level. Constraints (4.4) are sign and integrality requirements on S_j.

The complicating constraints (4.3) make [ISP] a difficult problem to solve. However, considering each SC as a queueing system, we use Little's law [36] to find the waiting time expression as:

$$\overline{W}_j = \frac{\bar{B}_j}{\lambda_j}. \tag{4.5}$$

Using (4.5), we replace constraints (4.3) by

$$\bar{B}_j \leqslant \tau \lambda_j. \tag{4.6}$$

In order to solve [ISP], we need to find mean inventory and backorder level expressions.

4.2 Inventory Level at the Plant and SCs

Considering the steady state behavior of the queueing system at the plant, we use standard inventory and backorder level expressions[1]

$$\bar{I}_0 = S_0 - E[N_0] + \bar{B}_0 \tag{4.7}$$

where N_0 denotes the steady state number of outstanding orders in the queueing system at the plant,

$$\bar{B}_0 = E[N_0] - \sum_{s=0}^{S_0-1} [1 - F_0(s)] \tag{4.8}$$

and

$$F_0(s) = \sum_{m=0}^{s} P(N_0 = m).$$

Considering an M/M/1 queueing system at the plant and substituting the steady state probabilities for M/M/1 in equations (4.7) and (4.8), we get the mean inventory and backorder levels at the plant as given by Buzacott and Shanthikumar [9]:

$$\bar{I}_0 = [S_0 - \frac{\rho}{1-\rho}(1 - \rho^{S_0})], \tag{4.9}$$

$$\bar{B}_0 = \frac{\rho^{S_0+1}}{(1-\rho)}. \tag{4.10}$$

[1] See Caglar et al. [11] and Mak and Shen [37].

Hence, the mean waiting time at the plant is calculated as:

$$\overline{W}_0 = \frac{\bar{B}_0}{\lambda} = \frac{p^{S_0+1}}{\lambda(1-p)}. \tag{4.11}$$

Similarly, considering each SC as a queueing system, the mean inventory level, backorder level and waiting time at SCs is:

$$\bar{I}_j = S_j - E[N_j] + \bar{B}_j \tag{4.12}$$

where N_j denotes the number of outstanding orders at SC j (that are either in transit from the plant to SCs or backordered at the plant),

$$\bar{B}_j = E[N_j] - \sum_{s=0}^{S_j-1} [1 - F_j(s)] \tag{4.13}$$

and

$$F_j(s) = \sum_{m=0}^{s} P(N_j = m).$$

In order to solve equations (4.12-4.13), we need to find the probability distribution of the number of outstanding orders $N_j, j \in J$.

4.3 Distribution of the Number of Outstanding Orders at the SCs

In this section, we analyze different algorithms proposed in the multi-echelon inventory management literature to find the distribution of the number of outstanding orders, N_j.

Exact Model

Graves [26] suggests an exact algorithm to obtain the steady state distribution of N_j. In order to find the exact steady state distribution, we first find the distribution of the aggregate outstanding orders and then we disaggregate this distribution into the distributions of the number of outstanding orders at each SC.

The number of aggregate outstanding orders at all SCs is derived from Graves [26] as:

$$N = B(S_0) + D \tag{4.14}$$

where N is the aggregate outstanding orders at all sites, $B(S_0)$ is the back-orders at the plant for base-stock level S_0 and D is the aggregate failures at all SCs. $B(S_0)$ and D are independent random variables due to the fact the failure process is Poisson.

In order to get the distribution of N, we convolve the distribution of $B(S_0)$ and D

$$Pr(N = h) = \sum_{i=0}^{h} Pr(B = i)Pr(D = h - i).$$

Assuming an M/M/1 repair system, the distribution of $B(S_0)$ is given by Buzacott and

Shanthikumar [9]:

$$Pr(B = i) = (1 - \rho)\rho^{i+S_0} \quad \text{where } i = 0, 1, 2, 3, ... \tag{4.15}$$

As the shipment time from the plant to the SCs is deterministic, D has a Poisson distribution. We assume that the shipment time from plant to SC is the same for all the SCs, $\alpha_1 = \alpha_2 = \alpha_3... = \alpha$,

$$Pr(D = h - i) = \frac{e^{-(\lambda\alpha)}(\lambda\alpha)^{h-i}}{(h-i)!} \quad \text{where } i = 0, 1, 2, 3, ... \quad h = 0, 1, 2, 3, ... \tag{4.16}$$

Using equations (4.15) and (4.16),

$$\begin{aligned} Pr(N = h) &= \sum_{i=0}^{h}(1 - \rho)\rho^{i+S_0}\frac{e^{-(\lambda\alpha)}(\lambda\alpha)^{h-i}}{(h-i)!} \\ &= (1 - \rho)\rho^{S_0}e^{-(\lambda\alpha)}\sum_{i=0}^{h}\frac{\rho^{i}(\lambda\alpha)^{h-i}}{(h-i)!} \end{aligned}$$

multiplying and dividing by ρ^h:

$$\begin{aligned} &= (1 - \rho)\rho^{S_0+j}e^{-(\lambda\alpha)}\sum_{i=0}^{h}\frac{(\frac{\lambda\alpha}{\rho})^{i}}{i!} \\ &= (1 - \rho)\rho^{S_0+j}e^{-(\lambda\alpha)}\sum_{i=0}^{h}\frac{(\mu\alpha)^{i}}{i!} \end{aligned}$$

multiplying and dividing by $e^{(\mu\alpha)}$:

$$= (1 - \rho)\rho^{S_0+j}\sum_{i=0}^{h}\frac{e^{(\mu\alpha)}(\mu\alpha)^{i}}{i!}(e^{-\lambda\alpha+\mu\alpha})$$

$$Pr(N = h) = \left\{(1-\rho)\rho^{S_0+j}(e^{\alpha(\mu-\lambda)})\right\} \sum_{i=0}^{h} \frac{e^{(-\mu\alpha)}(\mu\alpha)^i}{i!}. \tag{4.17}$$

According to Graves [26], once we find the distribution of the number of aggregate outstanding orders, we disaggregate this distribution into the distributions of outstanding orders for each SC as:

$$Pr(N_j = m) = \sum_{h=m}^{\infty} [Pr(N = h)] [Pr(N_j = m|N = h)] \quad \text{for each } j \in J. \tag{4.18}$$

Since the plant fills the backorder requests on a FCFS basis, we use the binomial distribution for the conditional distribution $Pr(N_j = m|N = h)$.

Thus, the steady state distribution of the number of the outstanding orders at each SC is

$$Pr(N_j = m) = \sum_{h=m}^{\infty} [Pr(N = h)] \binom{h}{m} \left[\frac{\lambda_j}{\lambda}\right]^m \left[\frac{\lambda - \lambda_j}{\lambda}\right]^{h-m} \quad \text{for each } j \in J. \tag{4.19}$$

METRIC-like Model

The METRIC-like model approximates the distribution of the number of the outstanding orders N_j with a Poisson distribution which requires only the expression of the mean of N_j given by Graves [26]:

$$E[N_j] = \frac{\lambda_j}{\lambda}\bar{B}_0 + \lambda_j \alpha_j. \tag{4.20}$$

Sherbrooke's METRIC model [62] assumes Compound Poisson failure processes and ample repair capacity, and thus the repair process behaves like an M/G/∞ queue. The distribution of N_j is thus asymptotically Poisson [53]. Since the shipment time from the plant to the SCs is deterministic, D has a Poisson distribution. This implies that $B(S_0)$ has a Poisson distribution since N and D are assumed Poisson [26]. Thus, the METRIC-like model approximates the backorder level at the plant (4.10) with a Poisson random variable in case of deterministic shipment time to the SCs [26] and as a result we replace $F_j(s)$ by the Poisson CDF with mean $\lambda_j \bar{L}_j$ [11, 37].

The Poisson distribution is given by:

$$Pr(N_j = m) = \frac{e^{-(\lambda_j \bar{L}_j)}(\lambda_j \bar{L}_j)^m}{m!} \tag{4.21}$$

where

$$\bar{L}_j = \overline{W}_0 + \alpha_j = \frac{p^{S_0+1}}{\lambda(1-p)} + \alpha_j. \tag{4.22}$$

Approximate Model

The Approximate Model suggested by Graves [26] approximates N_j by a negative binomial distribution. It assumes that all failure processes are Poisson and the plant fills order requests on a FCFS basis. It requires both the mean of N_j given in equation (4.20) and the variance of N_j calculated in [26]:

$$Var(N_j) = \left(\frac{\lambda_j}{\lambda}\right)^2 Var\{B(S_0)\} + \left(\frac{\lambda_j}{\lambda}\right)\left(\frac{\lambda - \lambda_j}{\lambda}\right) \bar{B}_0 + \lambda_j \alpha_j. \tag{4.23}$$

The variance of the backorder level at the plant for base-stock level S_0 is calculated as

$$Var\{B(S_0)\} = \frac{\rho^{S_0+1}(2-\rho-\rho^{S_0+1})}{(1-\rho)^2}. \tag{4.24}$$

The negative binomial distribution is given by

$$Pr(N_j = m) = \binom{r+m-1}{m} q^r (1-q)^j \text{ for } m = 0, 1, 2, ... \tag{4.25}$$

where ($0 \leqslant q \leqslant 1$) and ($r \geqslant 0$),

$$E(N_j = m) = r(1-q)/q, \tag{4.26}$$

$$Var(N_j = m) = r(1-q)/q^2. \tag{4.27}$$

Graves [26] compares the results of the METRIC and Approximate models with the Exact model. They find that both approximations are effective in approximating the distribution of the number of the outstanding orders N_j, however, "the negative binomial approximation virtually dominates the METRIC approximation" in terms of accuracy. Mak and Shen [37], use the METRIC-like approximation, claiming that finding the exact distribution is computationally expensive and using the negative binomial distribution requires rounding of the parameter r, which makes the optimization difficult. They use the METRIC-like approximation without verifying whether it is appropriate. We on the other hand experiment with all the three models; the Exact, METRIC-like and Approximate.

We present an exact solution of [ISP] in the next section.

4.4 Exact solution of [ISP]

In this section, we propose an exact procedure to solve inventory stocking problem [ISP] introduced in section 4.1. Given location and allocation variables $\widehat{X}_j$ and $\widehat{Y}_{ij}$, [ISP] calculates the base-stock levels S_0 and S_j that satisfy response time (4.6) and capacity constraints

(4.2) while minimizing inventory holding costs (4.1).

The iterative procedure is able to use any of the inventory models presented in section 4.3, and iteratively finds the minimum inventory holding cost solution for a given set of facility locations and customer allocations.

The algorithm starts with $S_0 = C_0$ and finds all the possible feasible base-stock levels S_j that satisfy constraints (4.2) and (4.6). Once we find all possible feasible solutions we pick the solution which has the lowest inventory holding cost, this is a local minimum cost solution. However, if we are unable to find a feasible base-stock level S_j for given value of S_0 then [ISP] is infeasible.

Once we find a local minimum cost solution for S_0, we decrease S_0 by one and repeat the above procedure. This process is continued till S_0 reaches zero or we are unable to find local minimum cost solution for some given value of S_0. After finding all the local minimum cost solutions, we pick the minimum, this becomes the global minimum inventory holding cost solution $\widehat{Z}_j$ for given $(\widehat{X}, \widehat{Y})$.

The [ISP] Algorithm is summarized as:

Inventory Stocking Problem Algorithm [ISPA]

Initialize: $S_0 = C_0$

while $S_0 \geqslant 0$ **do**

  Initialize $S_j = 0$

  for each open SC **do**

  $\bar{B}_j = E[N_j] - (1 - F_j(s))$

    while $\bar{B}_j \geqslant \tau\lambda_j$ **do**

      if $S_j < C_j$ **then**

      set $S_j \leftarrow S_j + 1$

      set $\bar{B}_j \leftarrow \bar{B}_j - (1 - F_j(s))$

      end if

    end while

    for each feasible S_j **do**

    set $S_j \leftarrow S_j + 1$

    set $\bar{B}_j \leftarrow \bar{B}_j - (1 - F_j(s))$

    end for

  end for

  set $S_0 \leftarrow S_0 - 1$

end while

The [ISPA] calculates the steady state state parameters and may use the METRIC-like, Approximate or Exact model to do so.

We present the solution approach based on the cutting-plane algorithm in the following chapter.

Chapter 5

Exact Solution of the Inventory-Location Problem by Cutting-Planes

The formulation [P] is a mixed integer non-linear program. It is difficult to solve because of complicating constraints (3.6). In this chapter, we propose a cutting-plane method that solves a relaxation of [P] where constraints (3.5) and (3.6) are dropped. We calculate minimum cost inventory solution by solving [ISP] and based on [ISP] solution we propose a family of valid cuts to strengthen the relaxation.

In general, a cutting-plane algorithm would first solve a relaxed master problem where complicating constraints are dropped. The relaxed master problem solution gives a lower bound to the original problem. Then based on the relaxation solution, a subproblem is solved to get an upper bound to the original problem and valid cutting-planes, or cuts, are derived. These cuts are added to the relaxed master problem to tighten the relaxation.

If the relaxation solution is feasible then an optimality cut is added to improve the lower bound, and if the relaxation solution is infeasible then a feasibility cut is added. This process keeps on repeating to refine the relaxation until an optimal solution (when lower bound becomes equal to upper bound) is obtained.

Consider the relaxed location-allocation master problem [RLAMP] defined by (3.1)-(3.4), (3.8)-(3.10). Let $(\widehat{X}, \widehat{Y}, \widehat{Z})$ be a minimum cost solution to [RLAMP], $\widehat{J} = \{j : \widehat{X}_j = 1\}$ be the set of open SCs, and $\widehat{\lambda}_j = \sum_{i \in I} \lambda_i \widehat{Y}_{ij}$ be the demand rate at open SC j. For solution $(\widehat{X}, \widehat{Y}, \widehat{Z})$ to be feasible to [P], it has to satisfy the relaxed constraints (3.5) and (3.6). To verify this, we solve the inventory stocking problem [ISP]. The [ISP] solves for the minimum cost solution $(\widehat{Z})$ given $(\widehat{X}, \widehat{Y})$ by finding the base-stock levels that satisfy the relaxed constraints (3.5) and (3.6). If [ISPA] finds a feasible solution an optimality cut is added to [RLAMP] to improve the lower bound, and if [ISPA] is unable to find a feasible solution then a feasibility cut is added to [RLAMP] to remove the current relaxed solution.

5.1 Valid Cuts and Cutting-Plane Algorithm

In case we get a feasible [ISP] solution, it gives us the minimum inventory holding cost, $\widehat{Z}_j$. Then an optimality cut for [RLAMP] is formally stated as:

$$Z \geqslant \widehat{Z}_j - \widehat{Z}_j \sum_{j \in \widehat{J}} (1 - X_j) \tag{5.1}$$

where $\widehat{J}$ is the set of open facilities and $\widehat{Z}_j$ is the minimum inventory holding cost incurred at location j in order to satisfy customer demand. If the same SCs are open again by [RLAMP], then the summation term in cut (5.1) is dropped and the optimality cut is reduced to the form

$$Z \geqslant \widehat{Z}_j \tag{5.2}$$

which forces Z to be greater than or equal to the minimum inventory holding cost.

If the [ISPA] is unable to find the feasible solution then we add feasibility cut to [RLAMP] as:

$$\sum_{j \notin \widehat{J}} X_j + \sum_{j \in \widehat{J}} (1 - X_j) \geqslant 1 \tag{5.3}$$

If the same customers are assigned to the same SC by [RLAMP], then $\sum_{j \in \widehat{J}} (1 - X_j) = 0$ and cut (5.3) becomes

$$\sum_{j \notin \widehat{J}} X_j \geqslant 1 \tag{5.4}$$

which forces [RLAMP] to remove the current infeasible solution and to find a new solution.

Constraints (5.1) and (5.3) are valid cuts since they do not cut any feasible solution and they do remove the current infeasible solution from [RLAMP] and lead to optimality by narrowing down the solution space and closing the gap between bounds.

We decompose [P] into relaxed location-allocation master problem [RLAMP] and inventory stocking problem [ISP]. The cutting-plane algorithm iteratively solves the [RLAMP] by locating facilities, and allocating customers to the open facilities. Since, [RLAMP] is a relaxation of the original minimization problem [P], it gives a lower bound to [P]. Then given $(\widehat{X}, \widehat{Y})$, we solve [ISP] to generate a feasible solution and an upper bound to [P]. If the [ISPA] finds a feasible solution, we generate valid optimality cut to tighten the relaxation and to improve the lower bound. However, if the [ISP] solution is infeasible, we add a feasibility cut to [RLAMP]. This procedure is repeated until an optimal solution is obtained.

The formal description of the cutting-plane algorithm is given as:

Cutting-Plane Algorithm [CPA]

Initialize: $UB = \inf$, $LB = 0$.

While LB $\neq$ UB

Step 1. Solve [RLAMP], obtain solution $(\widehat{X}, \widehat{Y}, \widehat{Z})$, update LB.

Step 2. Test: if solution $(\widehat{X}, \widehat{Y}, \widehat{Z})$ is feasible with respect to (3.5) and (3.6), UB = LB, Stop.

Step 3. Solve [ISP], to construct a feasible solution

- If [ISP] is feasible:
    - Construct feasible solution $(\widehat{X}, \widehat{Y}, \widehat{Z}_j)$, update UB.
    - Add optimality cut (5.1) to [RLAMP].
- If [ISP] is not feasible:
    - Add feasibility cut (5.3) to [RLAMP].

Chapter 6

Numerical Testing

In this chapter, we analyze the effectiveness of cutting-plane algorithm [CPA]. The [CPA] is implemented in Matlab 7.14 on VAIO computer with Intel (R) Core i5-2540M CPU @ 2.60 GHz, 8.00 GB RAM, and Windows 7. The relaxed mixed-integer problem [RLAMP] is solved using GUROBI 4.6, a mixed-integer programming solver.

6.1 Performance of CPA

In this section, we run a set of experiments to test the performance of [CPA] using industrial data obtained from Bombardier Inc. and Daskin datasets [15].

6.1.1 Daskin Instances

We have used three data sets from Daskin [15] ; a 49-node, 88-node and a 150-node data set. The 49-node dataset contains capitals of the 48 states of the United States along with

Washington, D.C. in the 1990 U.S. census. The 88-node data set is defined on the 50 most populous cities in the 1990 U.S. census along with the 48 capitals of the continental U.S. The 150-node data set represents the 150 largest cities in the 1990 U.S. census. Every node is considered a potential SC location and a customer. These datasets are modified from Daskin [1995] in the following way: we have used the same facility location costs f_j. The demand rates λ_i are obtained by dividing the 1990 population figure (First Demand) by 10^6. The per unit inventory holding costs (h_o, h_j) and backorder costs p are 50 and 150, respectively.

We have used two different versions of each data set. In version 1 (v1) of the Daskin instances, shipment lead time α_j is obtained by dividing the distance between the corresponding demand node and and Springfield (node 6), IL, in the 49-node dataset and Chicago (node 3) in the 88-node and 150-node datasets by 100. The Plant and SC capacities (C_o, C_j), utilization rate ρ, response time requirement τ, and the distance requirement d_{max} are set to 10, 0.9, 5.5 and 2000, respectively. For version 2 (v2), the shipment lead time α_j is obtained by dividing the distance between the corresponding demand node and and Springfield (node 6), IL, in the 49-node dataset and Chicago (node 3) in the 88-node and 150-node datasets by 1000. The Plant and SC capacities (C_o, C_j), utilization rate ρ, response time requirement τ, and the distance requirement d_{max} are set to 5, 0.5, 1.5 and 500, respectively.

Version 2 of all the datasets contains relatively tougher instances because of smaller target time, smaller value of d_{max} and low storage capacity. Therefore, we expect to see more than one facility required to open to fulfill customer demand for version 2 datasets and less number of facilities for version 1. However, high utilization rate for version 1 will make a congested system and as a result we expect high inventory holding costs for version 1 datasets as compared to version 2 datasets.

In the Tables 6.1-6.2: Iter denotes the number of iterations; #SC stands for number of open SCs; UB denotes upper bound; Z refers to optimal inventory holding costs as percentage of UB; Gap is an optimality gap and calculated as $\frac{UB-LB}{LB} \times 100$; Time is the total clock time used in seconds; ISPS is the [ISP] solution time as percentage of total time; Solver is the solver time as percentage of total time; Fgap stands for Gap in the first solution; - stands for infeasible solution; M stands for METRIC-like model; A stands for Approximate model and E stands for Exact model

Table 6.1: Performance of CPA for Daskin instances

Instance	Iter	Model	#SC	Z	UB	Gap	Solver	ISPS	Time	Fgap
49_v1	30	M	1	48.7	119919	0	98	1.9	168.27	1.47
49_v1	30	A	1	48.7	119919	0	99.8	0.1	179.28	1.47
88_v1	12	M	1	8.5	58457	0	97.9	2.1	90.52	62.1
88_v1	12	A	1	8.5	58457	0	99.8	0.1	93.65	62.1
150_v1	44	M	1	5.5	105880	0	99.1	0.8	1889.84	-
150_v1	44	A	1	5.5	105880	0	99.7	0.2	1768.38	-
49_v2	8	M	5	7.52	313161	0	80.2	17.7	2.77	-
49_v2	8	A	5	7.52	313161	0	94.7	5.1	2.63	-
88_v2	1	M	7	0.61	419600	0	96.9	2.9	2.74	-
88_v2	1	A	7	0.61	419601	0	97.6	2.2	2.61	-
150_v2	1073	M	6	0.62	603751	0	94.4	5.5	5203.06	0.77
150_v2	1073	A	6	0.62	603751	0	99.2	0.7	3782.29	0.77

Table 6.1 shows the performance of [CPA] for Daskin instances in terms of speed and optimality gap. For all the instances tested, optimality gap is 0% and they are solved from 3 seconds up to 85 mins. For 49-node and 88-node dataset, version 1 optimal solution is

found within 3 seconds, whereas, version 2 takes 3 mins. The 150-node dataset is solved from 32 mins up to 85 mins.

6.1.2 Bombardier Instances

We have also used industrial dataset, which is obtained from Bombardier Inc. This dataset consists of 20 potential SC locations and 121 customers. We have used a spare part which has the highest demand.

The distance from customer i to facility j is given in hours and the demand is given monthly. Shipment lead time α_j is 0.23 months for every j. Two different version of Bombardier instances (BBD) are used to test the algorithm. In version 1, the Plant and SC capacities (C_o, C_j), utilization rate ρ, response time requirement τ, and the distance requirement d_{max} are set to 10 units, 0.9, 0.025 months and 40 hours, respectively whereas in version 2, these values are set to 5 units, 0.5, 0.01 months and 25 hours, respectively.

Similar to the Daskin instances, Version 2 of (BBD) instances are considered to be difficult relative to version 1.

Table 6.2: Performance of CPA for Bombardier instances

Instance	Iter	Model	#SC	Z	UB	Gap	Solver	ISPS	Time	Fgap
20_v1	15	M	1	57.23	116,911	0	96.9	3.0	5.68	133.8
20_v1	15	A	1	59.88	124,626	0	87.9	11.9	6.04	149.2
20_v2	43	M	2	35.42	190,998	0	99.9	0.04	863.06	-
20_v2	43	A	2	24.23	190,259	0	99.9	0.05	562.07	-

Table 6.2 shows the performance of [CPA] for Bombardier instances. For the METRIC-like model, the solution times range from 6 seconds to 14 min and for Approximate Model,

the solution times range from 6 seconds to 9.5 mins. Optimality gap is 0% for all the tested instances.

Results in Tables 6.1 and 6.2 show that [CPA] solves the inventory-location problem effectively for industry size instances. All instances are solved to optimality within reasonable computational time. We observe that as expected, version 2 datasets of the Bombardier and Daskin instances are difficult to solve and require more than one SC to fulfill customer demand and incur low inventory holding costs as compared to corresponding version 1 datasets. We also observe that solver takes most of the time for all the instances tested and [ISPS] is almost negligible.

Moreover, we find that 50% of time first solution is infeasible for both Bombardier and Daskin instances. For the first solutions which are feasible we see that the gap is improved from 149.2 % to 0% for Bombardier instances, whereas maximum improvement for Daskin instances is from 62.1% to 0%. The Fgap shows the benefits of using an integrated approach.

Furthermore, the results for Daskin instances show that both the METRIC-like and Approximate models give the same solutions all the time, whereas the Bombardier instances show that using the METRIC-like versus the Approximate model does not lead to the same solution all the time. Moreover, for Bombardier instances version 1, solution is different due to different base-stock levels, whereas for version 2, solution differs in base-stock levels as well as in SC locations.

It is important to understand when these models differ and if possible find out which is a better approximation. We carry out further testing in the following section to investigate these issues.

6.2 Comparison of the METRIC-like and Approximate Models

In this section, we run a set of experiments to verify the results of the METRIC-like and Approximate models with the Exact model described in chapter 4. For these tests we solve inventory-location problem [P] using the METRIC-like model [M], and based on location decision $\widehat{X}_j$, we solve [ISP] with the Approximate model [A] and Exact model [E].

In computing the exact distribution, we find that the aggregation and disaggregation procedures are very time consuming especially for high demand and large values of S_o even for small scale problems, which makes the [ISPA] computationally expensive. Thus, we perform the following set of experiments only on version 1 profiles of all datasets, with low value of demand and $S_o = 0$, because for such setting we are able to compute the distribution in reasonable time.

In the rest of the chapter, we use the following notation in addition to that introduced in Sections 3.2 and 6.1.1. $\widehat{X}_j$ denotes the open SC location obtained by solving [P] using [M], $\widehat{Z}_j$ refers to optimal inventory holding costs at open SC as percentage of total cost, $\widehat{S}_j$ refers to the base-stock level at open SC, $\widehat{B}_j$ refers to the mean backorder level at open SC, $\widehat{I}_j$ refers to the mean inventory level at open SC, $\widehat{W}_j$ refers to the mean response time to a customer at open SC, and %dev refers to percentage deviation calculated as (exact cost-approximation cost)/exact cost)$\times$100.

Table 6.3: Comparison of approximations with the Exact model for BBD 20_v1
$J = 20, I = 121, \lambda = 17.44,\ d_{max} = 40,\ \tau = 0.025, \alpha = 0.2308$

Model	$\widehat{X}_j$	$\widehat{Z}_j$	S_o	$\bar{I}_o$	$\bar{B}_o$	$\overline{W}_o$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	UB	%dev
						$\rho = 0.1$						
E	12	36.10	0	0	0.1111	0.0064	6	0.2271	2.0913	0.013	78250	-
M	12	36.07	0	0	0.1111	0.0064	6	0.2261	2.0903	0.013	78209	-0.14
A	12	36.32	0	0	0.1111	0.0064	6	0.2337	2.098	0.0134	78522	+0.96
						$\rho = 0.3$						
E	12	36.72	0	0	0.4286	0.0246	6	0.3251	1.8719	0.0186	79011	-
M	12	36.20	0	0	0.4286	0.0246	6	0.3094	1.8562	0.0177	78372	-2.20
A	12	35.77	0	0	0.4286	0.0246	6	0.2966	1.8434	0.017	77850	-4
						$\rho = 0.5$						
E	12	40.33	0	0	1	0.0573	7	0.3352	2.3105	0.0192	83788	-
M	12	38.10	0	0	1	0.0573	7	0.2614	2.2368	0.015	80781	-8.89
A	12	36.94	0	0	1	0.0573	7	0.2247	2.2001	0.0129	79288	-13.31
						$\rho = 0.7$						
E	12	51.05	0	0	2.3333	0.1338	10	0.3689	4.0109	0.0212	102146	-
M	12	40.21	0	0	2.3333	0.1338	8	0.4146	2.0567	0.0238	83632	-35.5
A	12	40.17	0	0	2.3333	0.1338	8	0.413	2.0551	0.0237	83566	-35.6

Table 6.4: Comparison of approximations with the Exact model for Daskin 49_v1
$J = 49, I = 49, \lambda = 40, d_{max} = 2000, C_j = 10, \tau = 5.5, \alpha = 1.5986$

Model	$\widehat{X}_j$	$\widehat{Z}_j$	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	UB	%dev
$\rho = 0.1$												
E	15	11.65	0	0	0.1111	0.0028	10	54.0562	0	1.3514	69608	-
M	15	11.65	0	0	0.1111	0.0028	10	54.0562	0	1.3514	69608	0
A	15	11.65	0	0	0.1111	0.0028	10	54.0562	0	1.3514	69608	0
$\rho = 0.3$												
E	15	11.71	0	0	0.4286	0.0107	10	54.3736	0	1.3593	69656	-
M	15	11.71	0	0	0.4286	0.0107	10	54.3736	0	1.3593	69656	0
A	15	11.71	0	0	0.4286	0.0107	10	54.3736	0	1.3593	69656	0
$\rho = 0.5$												
E	15	11.82	0	0	1	0.025	10	54.945	0	1.3736	69742	-
M	15	11.82	0	0	1	0.025	10	54.945	0	1.3736	69742	0
A	15	11.82	0	0	1	0.025	10	54.945	0	1.3736	69742	0
$\rho = 0.7$												
E	15	12.07	0	0	2.3333	0.0583	10	56.2784	0	1.407	69942	-
M	15	12.07	0	0	2.3333	0.0583	10	56.2784	0	1.407	69942	0
A	15	12.07	0	0	2.3333	0.0583	10	56.2784	0	1.407	69942	0
$C_j = 70, \rho = 0.9$												
E	15	1.75	0	0	9	0.225	70	6.2173	3.2722	0.1554	62596	-
M	15	1.38	0	0	9	0.225	70	5.0548	2.1098	0.1264	62364	-21.21
A	15	1.43	0	0	9	0.225	70	5.2094	2.2643	0.1302	62395	-18.39

Table 6.5: Comparison of approximations with the Exact model for Daskin 88_v1
$J = 88, I = 88, \lambda = 44.8400, h_o = 50, h_j = 50$ for every j
$d_{max} = 2000, C_j = 10, \tau = 5.5, \alpha = 0.8592$

Model	$\widehat{X}_j$	$\widehat{Z}_j$	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	UB	%dev
$\rho = 0.1$												
E	17	7.43	0	0	0.1111	0.0025	10	28.6375	0	0.6387	57795.6	-
M	17	7.43	0	0	0.1111	0.0025	10	28.6375	0	0.6387	57795.6	0
A	17	7.43	0	0	0.1111	0.0025	10	28.6375	0	0.6387	57795.6	0
$\rho = 0.2$												
E	17	7.47	0	0	0.25	0.0056	10	28.7764	0	0.6418	57816.5	-
M	17	7.47	0	0	0.25	0.0056	10	28.7764	0	0.6418	57816.5	0
A	17	7.47	0	0	0.25	0.0056	10	28.7764	0	0.6418	57816.5	0
$\rho = 0.3$												
E	17	7.51	0	0	0.4286	0.0096	10	28.9549	0	0.6457	57843.2	-
M	17	7.51	0	0	0.4286	0.0096	10	28.9549	0	0.6457	57843.2	0
A	17	7.51	0	0	0.4286	0.0096	10	28.9549	0	0.6457	57843.2	0
$\rho = 0.4$												
E	17	7.57	0	0	0.6667	0.0149	10	29.193	0	0.651	57879	-
M	17	7.57	0	0	0.6667	0.0149	10	29.193	0	0.651	57879	0
A	17	7.57	0	0	0.6667	0.0149	10	29.193	0	0.651	57879	0
$C_j = 70, \rho = 0.9$												
E	17	1.4	0	0	9	0.2007	53	2.4368	7.9105	0.0543	54261	-
M	17	0.82	0	0	9	0.2007	52	1.1036	5.5772	0.0246	53944	-41.61
A	17	0.9	0	0	9	0.2007	52	1.3153	5.789	0.0293	53987	-36.04

Table 6.6: Comparison of approximations with the Exact model for Daskin 150_v1
$J = 150, I = 150, \lambda = 58.1970, d_{max} = 2000, C_j = 10, \tau = 5.5, \alpha = 0.7678$

Model	$\widehat{X}_j$	$\widehat{Z}_j$	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	UB	%dev
						$\rho = 0.1$						
E	126	4.96	0	0	0.1111	0.0019	10	34.7944	0	0.5979	105219	-
M	126	4.96	0	0	0.1111	0.0019	10	34.7944	0	0.5979	105219	0
A	126	4.96	0	0	0.1111	0.0019	10	34.7944	0	0.5979	105219	0
						$\rho = 0.2$						
E	126	4.98	0	0	0.25	0.0043	10	34.9333	0	0.6003	105240	-
M	126	4.98	0	0	0.25	0.0043	10	34.9333	0	0.6003	105240	0
A	126	4.98	0	0	0.25	0.0043	10	34.9333	0	0.6003	105240	0
						$\rho = 0.3$						
E	126	5	0	0	0.4286	0.0074	10	35.1118	0	0.6033	105267	-
M	126	5	0	0	0.4286	0.0074	10	35.1118	0	0.6033	105267	0
A	126	5	0	0	0.4286	0.0074	10	35.1118	0	0.6033	105267	0
						$\rho = 0.4$						
E	126	5.04	0	0	0.6667	0.0115	10	35.3499	0	0.6074	105303	-
M	126	5.04	0	0	0.6667	0.0115	10	35.3499	0	0.6074	105303	0
A	126	5.04	0	0	0.6667	0.0115	10	35.3499	0	0.6074	105303	0
						$C_j = 70, \rho = 0.9$						
E	126	0.77	0	0	9	0.1546	60	2.3074	8.6241	0.0396	100777	-
M	126	0.47	0	0	9	0.1546	59	1.0351	6.3519	0.0178	100473	-39.17
A	126	0.49	0	0	9	0.1546	59	1.1332	6.45	0.0195	100492	-36.64

Tables 6.3-6.6 compare the results of approximations with the Exact model. We observe that the METRIC-like model outperforms the Approximate model for Bombardier

instances in terms of % deviation whereas the Approximate model dominates the METRIC-like model for Daskin instances. We also observe that, for Daskin instances, both approximations predict the same base-stock, backorder and inventory levels as the Exact model for low and medium utilization rates, and differ for high utilization rates and high capacity. For Bombardier instances, the approximations differ for all of the cases tested and the differences are more significant for high utilization rates.

These results suggest that the Approximate and METRIC-like models produce solutions that differ from the Exact model solutions at high utilization rates. Since the inventory and backorder levels depend on the probability distribution of the number of outstanding orders, we further explore and plot the distributions of the number of outstanding orders for the three models for different values of utilization rates to verify the differences.

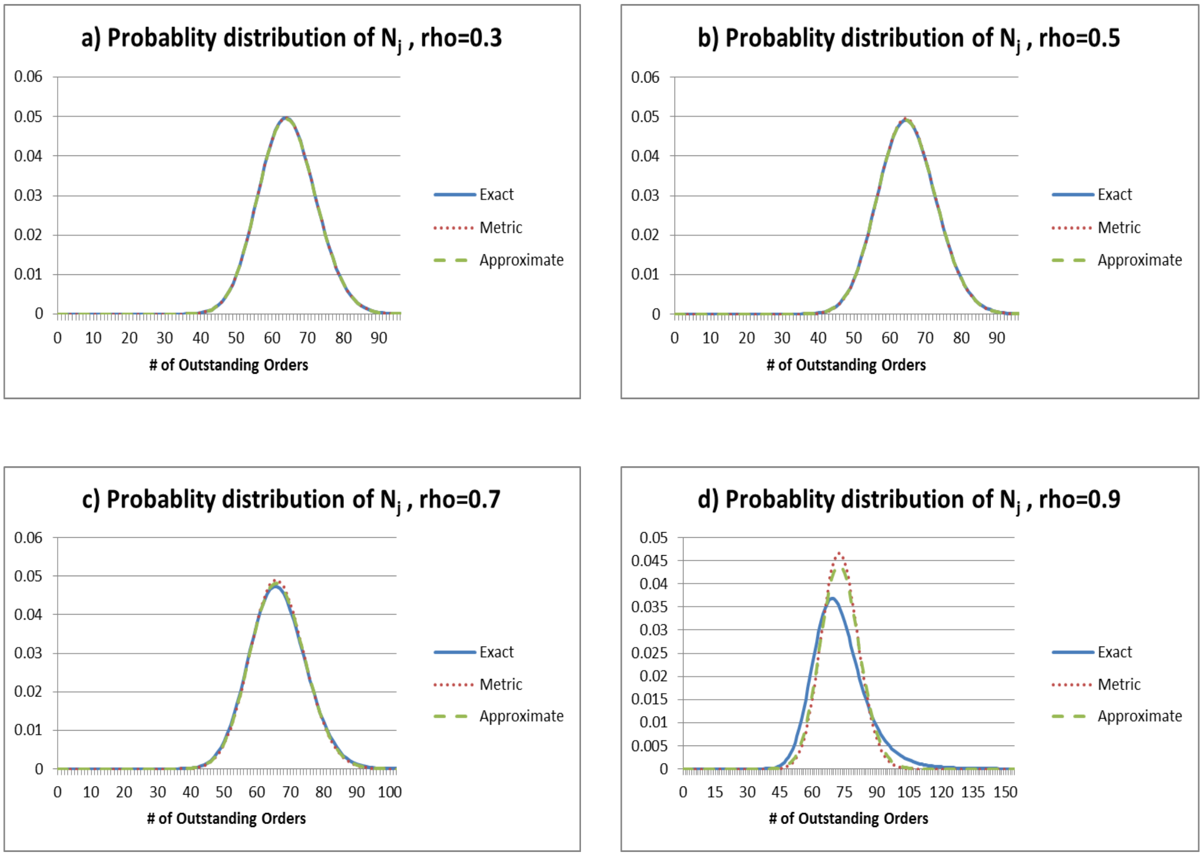

Figure 6.1: Probability distribution of N_j for Daskin 49_v1

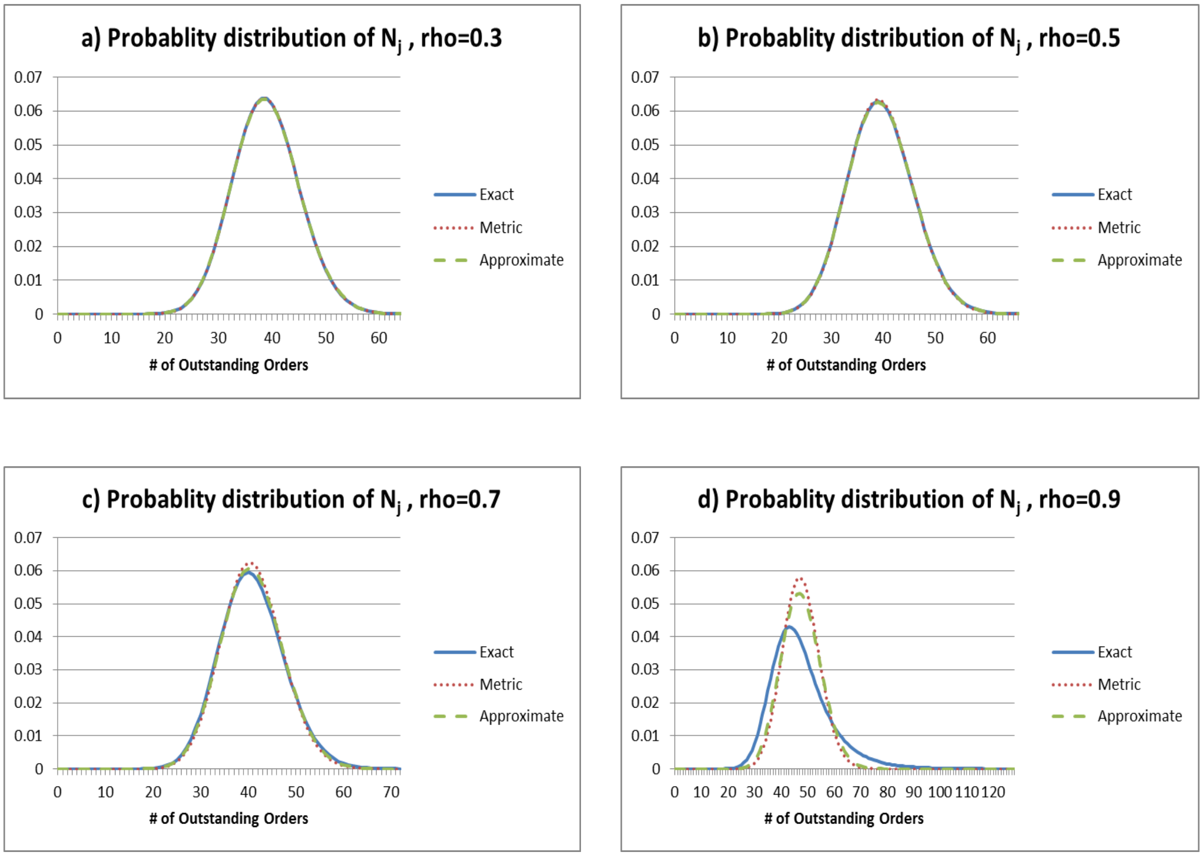

Figure 6.2: Probability distribution of N_j for Daskin 88_v1

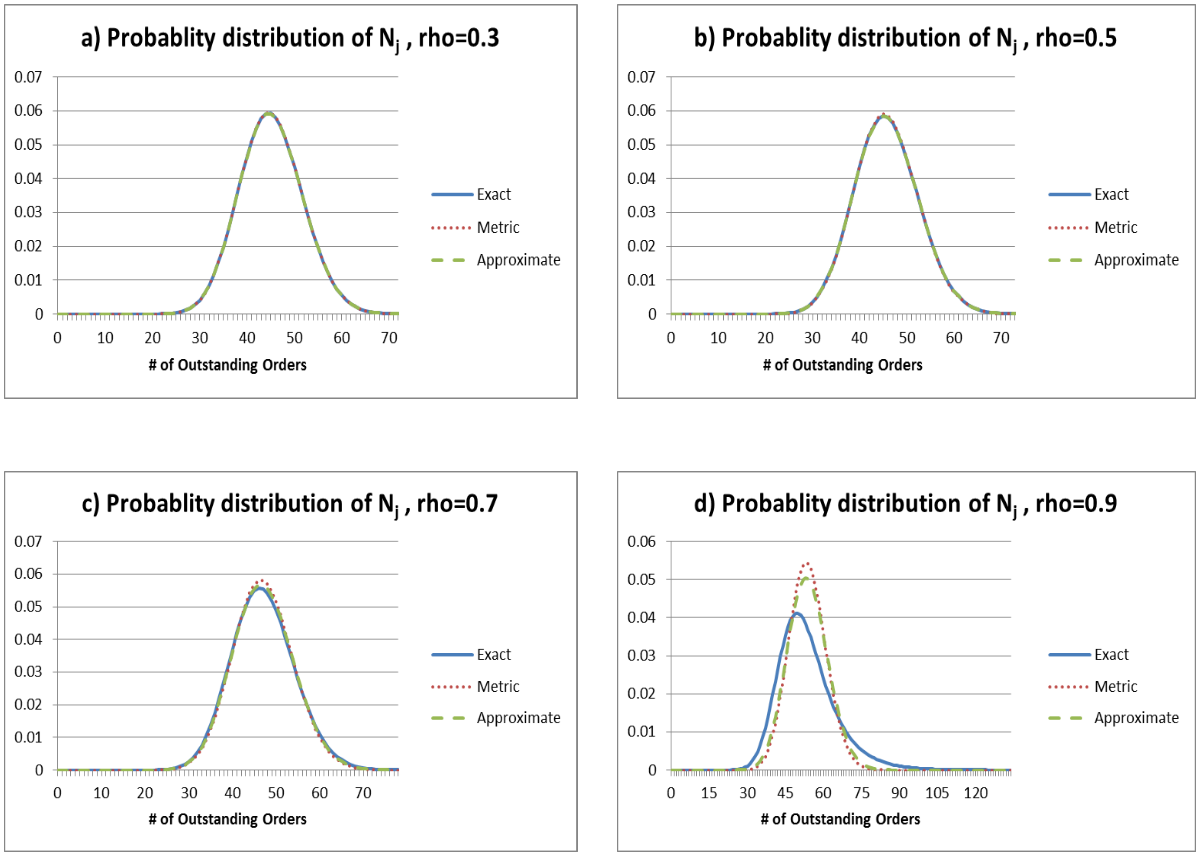

Figure 6.3: Probability distribution of N_j for Daskin 150_v1

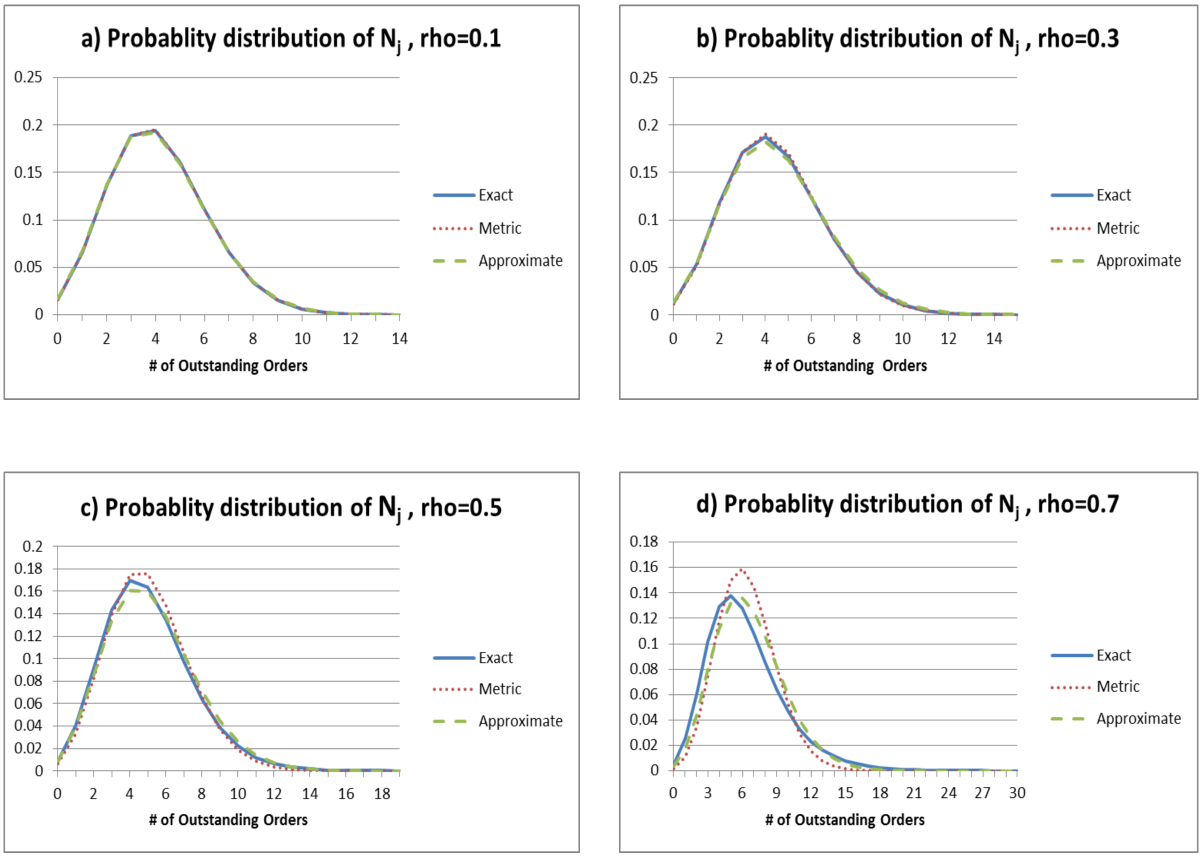

Figure 6.4: Probability distribution of N_j for BBD 20_v1

Figures 6.1-6.4 show the probability distribution plots of the number of the outstanding orders for the METRIC-like, Approximate and Exact models for Bombardier and Daskin instances. It is observed that in case of deviation, the Approximate model overestimates the exact distribution whereas the METRIC-like model underestimates the exact distribution. Graves [26] has made similar observations as well. We observe that the METRIC-like and Approximate models provide accurate approximation for low (ρ=0.1, 0.3) and medium (ρ=0.5) utilization rates, however they deviate from the exact distributions in case of high

(ρ=0.7, 0.9) utilization rates.

Our results are in agreement with those shown by Diaz and Fu [16]. They have demonstrated that the METRIC and Approximate models do not work well in case of the high utilization rates because these models ignore the queueing effects in the repair process.

Chapter 7

Sensitivity Analysis

In this chapter, we perform a set of experiments to investigate the impact of capacity, backorder cost and the utilization rate on the solution in terms of total cost, base-stock level and backorder level. In the following experiments, we solve inventory-location problem [P] using the METRIC-like model [M]. In this chapter, we use the notation introduced in Sections 3.2 and 6.1.1.

7.1 Effects of Capacity

In this section we vary the capacity level for Daskin and Bombardier instances so we can get a better understanding of the effect of capacity on the problem solution.

Table 7.1: Effects of capacity on Daskin 49_v1
$J = 49, I = 49, \lambda = 247.05, \rho = 0.9, \tau = 5.5, \alpha = 1.5986$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	Cost	Time
$C_0, C_j = 10$	1	48.7	10	4.14	3.14	0.01	10	388.08	0	1.57	119919	168.27
$C_0, C_j = 30$	1	47.4	13	6.29	2.29	0.01	30	367.23	0	1.49	116899	191.27
$C_0, C_j = 50$	1	46	13	6.29	2.29	0.01	50	347.23	0	1.41	113899	187.98
$C_0, C_j = 70$	1	44.5	13	6.29	2.29	0.01	70	327.23	0	1.32	110899	248.68
$C_0, C_j = 90$	1	43	13	6.29	2.29	0.01	90	307.23	0	1.24	107899	322.57

Table 7.2: Effects of capacity on Daskin 88_v1
$J = 88, I = 88, \lambda = 44.8400, \rho = 0.9, \tau = 5.5, \alpha = 0.8592$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	Cost	Time
$C_0, C_j = 10$	1	8.5	10	4.14	3.14	0.07	10	31.66	0	0.71	58457	95.98
$C_0, C_j = 30$	1	3.5	12	5.54	2.54	0.06	30	11.15	0.08	0.25	55454	56.33
$C_0, C_j = 50$	1	0.8	1	0.1	8.1	0.18	50	1.39	4.76	0.03	53951	62.04
$C_0, C_j = 70$	1	0.8	0	0	9	0.2	52	1.1	5.58	0.02	53944	89.42
$C_0, C_j = 90$	1	0.8	0	0	9	0.2	52	1.1	5.58	0.02	53944	133.59

Table 7.3: Effects of capacity on Daskin 150_v1
$J = 150, I = 150, \lambda = 58.1970, \rho = 0.9, \tau = 5.5, \alpha = 0.7678$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	Cost	Time
$C_0, C_j = 10$	1	5.5	10	4.14	3.14	0.05	10	37.82	0	0.65	105880	1889.84
$C_0, C_j = 30$	1	2.8	13	6.29	2.29	0.04	30	16.98	0.01	0.29	102862	1941.99
$C_0, C_j = 50$	1	0.6	6	1.78	4.78	0.08	50	2.55	3.08	0.04	100625	2601.48
$C_0, C_j = 70$	1	0.5	0	0	9	0.15	59	1.04	6.35	0.02	100473	2817.76
$C_0, C_j = 90$	1	0.5	0	0	9	0.15	59	1.04	6.35	0.02	100473	3850.55

Tables 7.1-7.3 show the change in problem solution due to an increase in capacity for Daskin instances.

As capacity at SC and the plant increases, base-stock levels at the SC increase which results in a decrease in backorder levels and increase in inventory levels. On the other hand, base-stock levels decrease at the plant which results in an increase in backorder levels and decrease in inventory levels at the plant. The trade-off between backorder and inventory holding costs leads to a decrease in total cost. Increasing capacity means that capacity constraints are less tight so either the previous solution is still optimal or the new solution has a better (lower) cost. It is worth noting that holding more stock at the SCs does not necessarily lead to smaller base-stock levels at the plant as is the case for Daskin 49_v1. One possible explanation is that the high utilization rate at the plant may push stock to be held at the plant rather than at SCs.

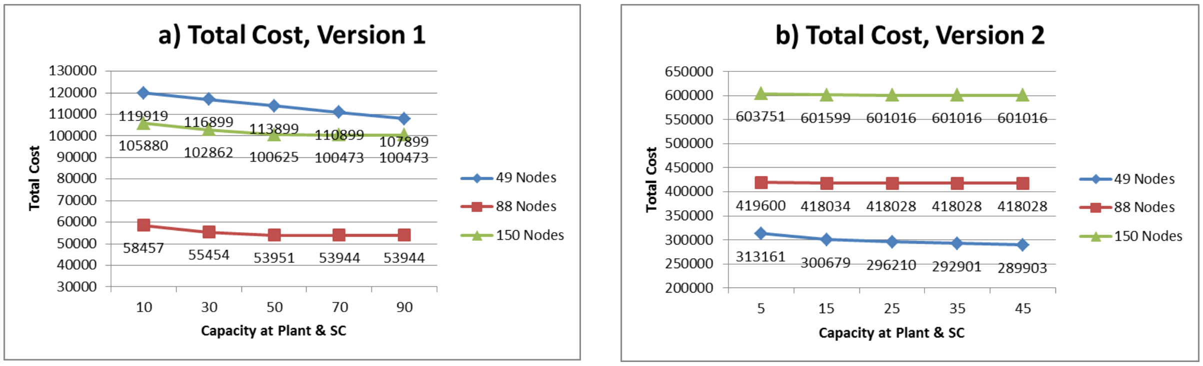

Figure 7.1: Effect on total cost by increasing capacity at the plant and SCs for Daskin instances

Figure 7.1 shows the impact of capacity on total cost for Daskin instances. For version 1, when capacity increases, total cost decreases for 49-node dataset. The total cost decreases because the decrease in backorder costs outweighs the increase in inventory holding costs as the capacity increases. However, the inventory level remains zero at the SC and increases at the plant. The reason for such behavior is high utilization rate ρ. Since ρ is high, it's better to keep inventory at the plant as compared to SCs. For 88-node dataset, the total cost decreases as the capacity is increased but it remains the same for C_j greater than 50. The reason for the initial decrease in the total cost is the same as for the 49-node dataset, however, for C_j greater than 50, the optimal base-base-stock policystock level and backorder level at SCs remain the same. This is because at optimality, $S_0 = 0$ and there is no room for improvement. The 150-node dataset exhibits similar trend as the 88-node dataset. The total cost decreases as the capacity is increased but it remains the same for C_j greater than 70.

Similarly, for version 2, as capacity increases the total cost decreases for the 49-node dataset. However for 88 and 150-node dataset, the total cost initially decreases but it remains the same for C_j greater than 25.

Table 7.4: Effects of capacity on BBD 20_v1
$J = 20, I = 121, \lambda = 17.44,\ \rho = 0.9,\ \tau = 0.025, \alpha = 0.2308$

	#SC	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	Cost	Time
$C_0, C_j = 8$	2[1]	43.2	5	1.31	5.31	0.30	7	0.18	2.5	0.02	175,938	2353
$C_0, C_j = 9$	2[2]	42.7	4	0.9	5.9	0.34	7.5	0.17	2.7	0.02	174,647	2286
$C_0, C_j = 10$	1	57.2	8	2.87	3.87	0.22	10	0.4	2.5	0.02	116,911	5.68
$C_0, C_j = 11$	1	53.6	6	1.78	4.78	0.27	11	0.42	2.62	0.02	107,821	5.57
$C_0, C_j = 12$	1	52.4	5	1.31	5.31	0.3	12	0.35	3.02	0.02	104,964	5.93
$C_0, C_j = 13$	1	51.5	4	0.9	5.9	0.34	13	0.31	3.38	0.02	103,061	6.31

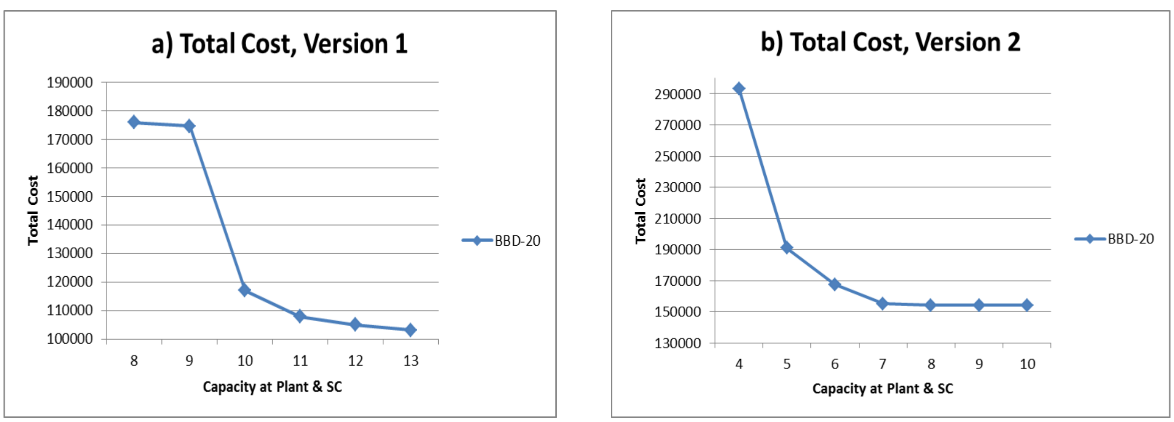

Figure 7.2: Effect on total cost by increasing capacity at the plant and SCs for Bombardier instances

Figure 7.2 shows the impact of capacity on total cost for Bombardier instances. For

[1]Since there are two open SCs, $S_j, \bar{B}_j, \bar{I}_j$ and $\bar{W}_j$ refer to average levels of the two open SCs
[2]See footnote 1

version 1, as capacity increases, total cost decreases as expected. With the increase in the capacity at the SCs and plant, base-stock levels at the SCs increase which results in a decrease in backorder levels and increase in inventory levels at the SCs. On the other hand, base-stock levels decrease at the plant which results in an increase in backorder levels and decrease in inventory levels at the plant. However, we see a drastic change in this trend both at the plant and SC when capacity reaches 10. This is because of the change in the number of open SCs and it leads to an increase in the base-stock level at the only open SC. Since the open SC reaches capacity, it also pushes the plant to hold more stock which results in a decrease in backorder levels and increase in inventory levels. The increase in inventory levels is offset by the decrease in backorder levels and as a result total cost decreases. It is worth noting that total cost changes significantly because of decrease in facility location cost as now it requires only one SC to open.

Version 2 exhibits similar behavior as the version 1 dataset. The total cost decreases drastically when capacity increases from 4 to 5 because of the decrease in the facility location cost as well as inventory holding cost.

7.2 Effects of Backorder Cost

In this section we study the effect of backorder cost for Daskin instances on the problem solution.

Table 7.5: Effects of backorder cost on Daskin 49_v1
$J = 49, I = 49, \lambda = 247.05, \rho = 0.9, \tau = 5.5, \alpha = 1.5986$

	#SC	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$p = 50$	1	37.5	6	1.78	4.78	0.02	10	593.32	0	2.4	79255	27.24
$p = 100$	1	38.8	10	4.14	3.14	0.01	10	388.08	0	1.57	100515	86.12
$p = 150$	1	48.7	10	4.14	3.14	0.01	10	388.08	0	1.57	119919	168.27

Table 7.6: Effects of backorder cost on Daskin 88_v1
$J = 88, I = 88, \lambda = 44.8400, \rho = 0.9, \tau = 5.5, \alpha = 0.8592$

	#SC	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$p = 50$	1	3.1	6	1.78	4.78	0.11	10	33.31	0	0.74	55255	56.05
$p = 100$	1	5.9	10	4.14	3.14	0.07	10	31.66	0	0	56873	103.03
$p = 150$	1	8.5	10	4.14	3.14	0.07	10	31.66	0	0.71	58457	95.98

Table 7.7: Effects of backorder cost on Daskin 150_v1
$J = 150, I = 150, \lambda = 58.1970, \rho = 0.9, \tau = 5.5, \alpha = 0.7678$

	#SC	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$p = 50$	1	2	6	1.78	4.78	0.08	10	39.47	0	0.68	102062	2063.17
$p = 100$	1	3.8	10	4.14	3.14	0.05	10	37.82	0	0.65	103989	1852.04
$p = 150$	1	5.5	10	4.14	3.14	0.05	10	37.82	0	0.65	105880	1889.84

Tables 7.5-7.7 show the change in problem solution due to an increase in backorder cost for Daskin Instances. As backorder cost p increases, backorder levels decrease both at the plant and SCs. This results in an increase in base-stock levels and as a result inventory levels increase at the plant, whereas inventory levels remain the same at SCs as base-stock

levels at SCs are at capacity. The inventory holding costs outweigh the backorder costs and as a result total cost increases. There is no change in backorder levels from $p = 100$ to $p = 150$ because base-stock levels are at capacity both at the plant and at the SCs. However, the increase in backorder cost drives the backorder costs up and hence the total cost increases.

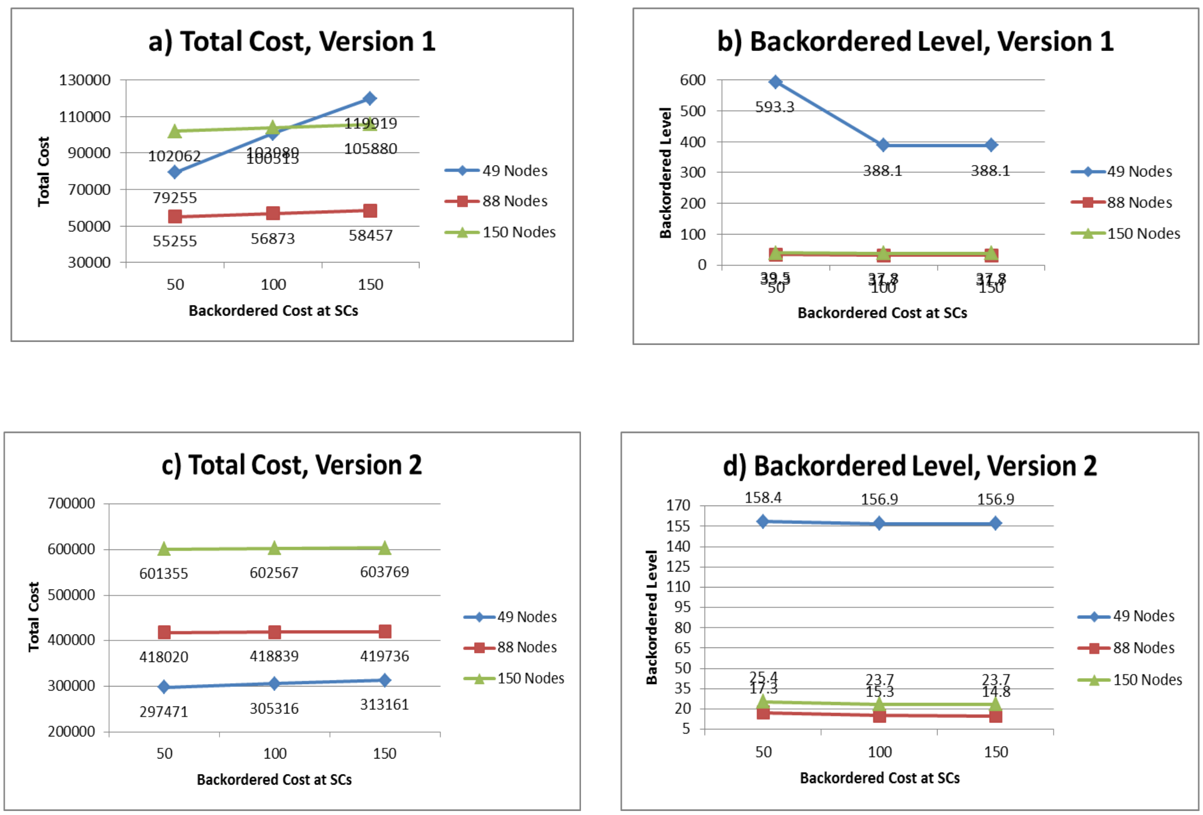

Figure 7.3: Effect on total cost by increasing backorder cost at the SCs for Daskin instances

From Figure 7.3, we see that when backorder cost increases, total cost increases for all three Daskin datasets for both versions as expected.

7.3 Effects of Utilization Rate

In this section we vary the utilization rate for Daskin and Bombardier instances so we can analyze the impact of utilization rate on the problem solution.

Table 7.8: Effects of utilization rate on Daskin 49_v1
$J = 49, I = 49, \lambda = 247.05, \rho = 0.9, \tau = 5.5, \alpha = 1.5986$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$\rho = 0.1$	1	48.4	0	0	0.11	0.0005	10	385.05	0	1.56	119258	161.87
$\rho = 0.5$	1	48.5	2	1.25	0.25	0.001	10	385.19	0	1.56	119341	163.49
$\rho = 0.9$	1	48.7	10	4.14	3.14	0.01	10	388.08	0	1.57	119919	168.27

Table 7.9: Effects of utilization rate on Daskin 88_v1
$J = 88, I = 88, \lambda = 44.84, \rho = 0.9, \tau = 5.5, \alpha = 0.8592$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$\rho = 0.1$	1	7.4	0	0	0.11	0.002	10	28.64	0	0.64	57796	81.26
$\rho = 0.5$	1	7.6	1	0.5	0.5	0.01	10	29.03	0	0.65	57879	83.14
$\rho = 0.9$	1	8.5	10	4.14	3.14	0.07	10	31.66	0	0.71	58457	95.98

Table 7.10: Effects of utilization rate on Daskin 150_v1
$J = 150, I = 150, \lambda = 58.1970, \rho = 0.9, \tau = 5.5, \alpha = 0.7678$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$\rho = 0.1$	1	4.9	0	0	0.11	0.002	10	34.79	0	0.6	105219	1916.25
$\rho = 0.5$	1	5	1	0.5	0.5	0.01	10	35.18	0	0.6	105302	1912.31
$\rho = 0.9$	1	5.5	10	4.14	3.14	0.05	10	37.82	0	0.65	105880	1889.84

Tables 7.8-7.10 show the change in problem solution due to the increase in utilization rate for Daskin instances. As utilization rate ρ increases at the plant, base-stock, backorder and inventory levels increase at the plant. On the other hand, inventory levels remain the same at SCs as base-stock levels at SCs are at capacity, whereas backorder levels increase as utilization rate increases. The increase in the base-stock, backorder and inventory levels in the system leads to an increase in the total cost.

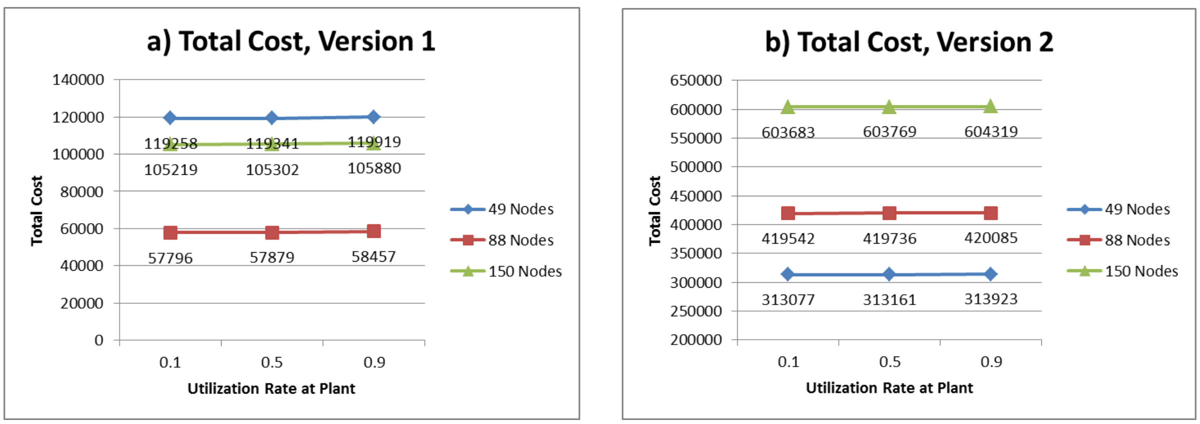

Figure 7.4: Effect on total cost by increasing utilization rate at the plant for Daskin instances

Figure 7.4 shows the impact of utilization rate on total cost for Daskin instances. We observe that as utilization rate increases, total cost increases for all the instances. For $\rho = 0.1$, optimal base-stock level at the plant is zero because there is no need to hold stock at the plant when it's almost idle, and base-stock level at SCs are at capacity. However, for $\rho = 0.5$, the plant holds stock since it becomes slightly busy but base-stock levels at SCs remain at capacity, whereas backorder levels increase both at the plant and at the SCs. For $\rho = 0.9$, the optimal base-stock level at the plant reaches to capacity as the plant is now really busy so there is a need to stock more at the plant to satisfy the response times, whereas base-stock levels at SCs remain at capacity, and backorder levels increase both at the plant and the SCs.

Table 7.11: Effects of utilization rate on BBD 20_v1
$J = 20, I = 121, \lambda = 17.44,\ \rho = 0.9,\ \tau = 0.025, \alpha = 0.2308$

	$\#SC$	Z	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	$\widehat{S}_j$	$\widehat{B}_j$	$\widehat{I}_j$	$\widehat{W}_j$	Cost	Time
$\rho = 0.1$	1	36.1	0	0	0.11	0.01	6	0.23	2.09	0.01	78,209	2.59
$\rho = 0.3$	1	36.2	0	0	0.43	0.02	6	0.31	1.86	0.02	78,372	2.61
$\rho = 0.5$	1	38.1	0	0	1	0.06	7	0.26	2.24	0.01	80,781	2.62
$\rho = 0.7$	1	40.2	0	0	2.33	0.13	8	0.41	2.06	0.02	83,632	2.75
$\rho = 0.9$	1	57.2	8	2.87	3.87	0.22	10	0.4	2.5	0.02	116,911	5.68

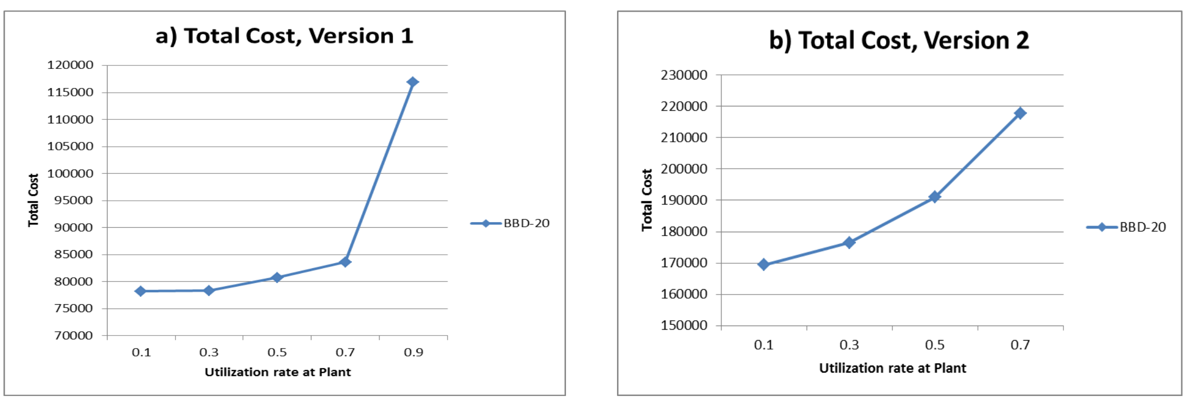

Figure 7.5: Effect on total cost by increasing utilization rate at the plant for Bombardier instances

Figure 7.5 shows the impact of utilization rate on total cost for Bombardier instances. We observe that as ρ increases total cost increases for Bombardier instances. For low values of utilization rate, $\rho = 0.1, 0.3$, the plant is idle and hence optimal base-stock level at the plant is zero whereas SCs hold stock. For relatively high utilization rates, $\rho = 0.5, 0.7$, the optimal base-stock level at the plant is still zero although now the plant is relatively busy as compared to $\rho = 0.1$. The reason could be the increase in the base-stock levels at the SCs. However, for $\rho = 0.9$, the optimal base-stock, backorder and inventory levels increase at the plant. The reason for the sudden increase is the high utilization rate.

The results of the above experiments show us that, in general, we see an increase in total cost with the increase in backorder cost or utilization rate. However, the total cost either decreases or remain the same with the increase in capacity.

Chapter 8

Conclusion

In this thesis, we have considered a supply chain design problem for spare parts that incorporates customer service requirements and customer preferences of SCs. It is a two-echelon inventory-location system consisting of a central manufacturing plant, a set of SCs and multiple customers with stochastic demand. The demand rates at the SCs follow an independent Poisson process. The plant manufactures at a rate μ with independent and exponential production times. Both the plant and SCs hold stock in anticipation of demand and use base-stock replenishment policy. We assume deterministic shipment times between plant and SCs and FCFS service discipline to fill the outstanding orders.

We present a mixed integer non-linear model that determines the optimal location-allocation and optimal base-stock levels at both echelons by minimizing the facility location costs of SCs, inventory holding costs at the plant and SCs and backorder costs at the SCs subject to a response time requirements and customer preferences.

To the best of our knowledge, our problem is the first to propose an exact cutting-plane algorithm for inventory-location problem. We consider time-based service requirements, customer preferences for SCs and stochastic replenishment process in an integrated inventory-location problem. Customers are allocated to the SCs based on their preferences and not on minimum allocation cost, which makes it a unique problem. The inclusion of time-based service requirement makes it difficult to solve. In order to handle this complexity, we propose a novel cutting-plane algorithm that exploits the structure of inventory-location problem by separating the location decisions from the inventory stocking decisions. We present an exact solution procedure that iteratively solves a relaxation of inventory-location problem in the master problem and the inventory stocking problem to generate valid cuts. We have demonstrated that the traditional inventory models like METRIC-like and Approximate model [26] do not perform well in approximating the distribution of the number of the outstanding orders in case of high utilization rates.

We have tested the cutting-plane algorithm on two different types of datasets; Bombardier datasets and Daskin datasets. Our results show the efficiency of the cutting-plane algorithm in terms of speed and optimality gap. We have achieved optimal solutions with zero optimality gap in reasonable times. We have also performed post optimality experiments to present important managerial insights. These experiments demonstrate a significant decrease in total cost due to increase in capacity and use of medium utilization rate especially in case of Bombardier datasets.

Finally, we suggest potential extensions for this problem setting. One possible extension is to allow for lateral shipments since it is becoming more relevant in modern supply chain networks. Another extension could be to implement the double negative binomial approximation suggested by Diaz and Fu [16] to this framework as most of the manufacturing facilities operate at high utilization rate. One possible future direction is to consider more than one manufacturing facility. Finally, we suggest studying multi-echelon (more than two levels) inventory systems.

References

[1] H. Abouee-Mehrizi, O. Berman, H. Shavandi, and A. G. Zare. An exact analysis of a joint production-inventory problem in two-echelon inventory systems. *Naval Research Logistics*, 58(8):713–730, 2011.

[2] S. C. Albright. An approximation to the stationary distribution of a multiechelon repairable-item inventory system with finite sources and repair channels. *Naval Research Logistics*, 36(2):179–195, 1989.

[3] S. C. Albright and A. Gupta. Steady-state approximation of a multiechelon multi-indentured repairable-item inventory system with a single repair facility. *Naval Research Logistics*, 40(4):479–493, 1993.

[4] S. C. Albright and A. Soni. Markovian multiechelon repairable inventory system. *Naval Research Logistics*, 35(1):49–61, 1988.

[5] Z. M. Avsar and W. H. M. Zijm. Capacitated two-echelon inventory models for repairable item systems. In S. B. Gershwin, Y. Dallery, C. Papadopoulos, and J. M. Smith, editors, *Analysis and Modeling of Manufacturing Systems*, volume 60, pages 1–36. Kluwer Academic Publishers, Boston, 2002.

[6] S. Axsater. Simple solution procedures for a class of two-echelon inventory problems. *Operations Research*, 38(1):64–69, 1990.

[7] B. Balcik. Multi-item integrated location/inventory problem. Master's thesis, Department of Industrial Engineering, Middle East Technical University, 2003.

[8] S. Benjaafar, Y. Li, D. Xu, and S. Elhedhli. Demand allocation in systems with multiple inventory locations and multiple demand sources. *Manufacturing & Service Operations Management*, 10(1):43–60, 2008.

[9] J. A. Buzacott and J. G. Shanthikumar. *Stochastic Models of Manufacturing Systems*. Prentice Hall, New Jersey, 1993.

[10] K. E. Caggiano, P. L. Jackson, J. A. Muckstadt, and J. A. Rappold. Efficient computation of time-based customer service levels in a multi-item, multi-echelon supply chain: A practical approach for inventory optimization. *European Journal of Operational Research*, 199(3):744–749, 2009.

[11] D. Caglar, C. Li, and D. Simchi-Levi. Two-echelon spare parts inventory system subject to a service constraint. *IIE Transactions*, 36(7):655–666, 2004.

[12] M. Candas and E. Kutanoglu. Benefits of considering inventory in service parts logistics network design problems with time-based service constraints. *IIE Transactions*, 39(2):159–176, 2007.

[13] M. Cohen, Y. Zheng, and V. Agrawal. Service parts logistics: A benchmark analysis. *IIE Transanctions*, 29(8):627–639, 1997.

[14] M. Daskin, C. Coullard, and Z.-J. M. Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 10(1–4):83–106, 2002.

[15] M. S. Daskin. *Network and discrete location : models, algorithms, and applications*. Wiley, New York, 1995.

[16] A. Diaz and M. C. Fu. Models for multi-echelon repairable item inventory systems with limited repair capacity. *European Journal of Operational Research*, 97(3):480–492, 1997.

[17] A. Diaz and M. C. Fu. Multi-echelon models for repairable items: A review. Document in Decision, Operations and Information Technologies Research Works Collection, University of Maryland, 2005.

[18] Z. Drezner. *Facility location : a survey of applications and methods*. Springer-Verlag, New York, 1995.

[19] Z. Drezner and H. Hamacher. *Facility location : applications and theory*. Springer, New York, 2002.

[20] G. D. Eppen. Effects of centralization on expected costs in a multi-location newsboy problem. *Management Science*, 25(5):498–501, 1979.

[21] S. J. Erlebacher and R. D. Meller. The interaction of location and inventory in designing distribution systems. *IIE Transactions*, 32(2):155–166, 2000.

[22] M. Ettl, G. Feigin, G. Lin, and D. Yao. A supply network model with base-stock control and service requirements. *Operations Research*, 48(2):216–232, 2000.

[23] P. Flint. Too much of a good thing: Better inventory management could save the industry millions while improving reliability. *Air Transport World*, 32(9):103–106, 1995.

[24] E. Gebennini, R. Gamberini, and R. Manzini. An integrated production-distribution model for the dynamic location and allocation problem with safety stock optimization. *International Journal of Production Economics*, 122(1):286–304, 2009.

[25] R. A. Gerrard and R. L. Church. Closest assignment constraints and location models: Properties and structure. *Location Science*, 4(4):251–270, 1996.

[26] S. C. Graves. A multi-echelon inventory model for a repairable item with one-for-one replenishment. *Management Science*, 31(10):1247–1256, 1985.

[27] D. Gross, D. R. Miller, and R. M. Soland. A closed queueing network model for multi-echelon repairable item provisioning. *IIE Transanctions*, 15(4):344–352, 1983.

[28] D. Gross, L. C. Kioussis, and D. R. Miller. A network decomposition approach for approximating the steady-state behavior of markovian multi-echelon reparable item inventory systems. *Management Science*, 30(11):1453–1468, 1987.

[29] D. Gross, B. Gu, and R. M. Soland. Iterative solution methods for obtaining steady state probability distributions of markovian multi-echelon repairable items inventory systems. *Computer & Operations Research*, 20(8):817–828, 1993.

[30] W. Hausman and G. Scudder. Priority scheduling rules for repairable inventory systems. *Management Science*, 28(11):1215–1232, 1982.

[31] W. J. Hopp, R. Q. Zhang, and M. L. Spearman. An easily implementable hierarchical heuristic for a two-echelon spare parts distribution system. *IIE Transactions*, 31(10): 977–988, 1999.

[32] A. A. Javid and N. Azad. Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E*, 46(5):582–597, 2010.

[33] W. Kennedy, J. W. Patterson, and L. D. Fredendall. An overview of recent literature on spare parts inventories. *International Journal Production Economics*, 76(2):201–215, 2002.

[34] A. Klose and A. Drexl. Facility location models for distribution system design. *European Journal of Operational Research*, 162(1):4–29, 2005.

[35] E. Kutanoglu. Insights into inventory sharing in service parts logistics systems with time-based service levels. *Computers & Industrial Engineering*, 54(3):341–358, 2008.

[36] J. D. C. Little. A proof for the queuing formula: L = w. *Operations Research*, 9(3):383–387, 1961.

[37] H.-Y. Mak and Z.-J. M. Shen. A two-echelon inventory-location problem with service considerations. *Naval Research Logistics*, 56(8):730–744, 2009.

[38] M. Melo, S. Nickel, and F. S. da Gama. Facility location and supply chain management a review. *European Journal of Operational Research*, 196(2):401–412, 2009.

[39] P. A. Miranda and R. A. Garrido. Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. *Transportation Research Part E*, 40(3):183–207, 2004.

[40] P. A. Miranda and R. A. Garrido. A simultaneous inventory control and facility location model with stochastic capacity constraints. *Networks & Spatial Economics*, 6(1):39–53, 2006.

[41] N. M. Mirasol. A queueing approach to logistics systems. *Operations Research*, 12(5):707–724, 1964.

[42] P. B. Mirchandani and R. L. Francis. *Discrete location theory*. Wiley, New York, 1990.

[43] K. Moinzadeh and H. L. Lee. Batch size and stocking levels in multi-echelon repairable systems. *Management Science*, 32(12):1567–1581, 1986.

[44] J. A. Muckstadt. A model for a multi-item, multi-echelon, multi-indenture inventory system. *Management Science*, 20(4):472–481, 1973.

[45] J. A. Muckstadt. A three-echelon, multi-item model for recoverable items. *Naval Research Logistics*, 26(2):199–221, 1979.

[46] J. A. Muckstadt. *Analysis and Algorithms for Service Parts Supply Chains*. Springer, New York, 2005.

[47] J. A. Muckstadt and L. Thomas. Are multi-echelon inventory methods worth implementing in systems with low-demand rate items? *Management Science*, 26(5): 483–494, 1980.

[48] L. K. Nozick and M. A. Turnquist. Integrating inventory impacts into a fixed-charge model for locating distribution centers. *Transportation Research Part E*, 34(3):173–186, 1998.

[49] L. K. Nozick and M. A. Turnquist. A two-echelon inventory allocation and distribution center location analysis. *Transportation Research Part E*, 37(6):425–441, 2001.

[50] S. Owen and M. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.

[51] L. Ozsen, C. R. Coullard, and M. S. Daskin. Capacitated warehouse location model with risk pooling. *Naval Research Logistics*, 55(4):295–312, 2008.

[52] L. Ozsen, C. R. Coullard, and M. S. Daskin. Facility location modeling and inventory management with multisourcing. *Transportation Science*, 43(4):455–472, 2009.

[53] C. Palm. Analysis of the Erlang traffic formula for busy-signal arrangements. *Ericsson Technics*, 5(9):39–58, 1938.

[54] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, second edition, 1992.

[55] C. ReVelle and H. Eiselt. Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19, 2005.

[56] C. ReVelle, H. Eiselt, and M. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184 (3):817–848, 2008.

[57] SCL. Logistics and supply chain management (scm) key performance indicators (kpi) analysis: A canada/united states perspective. Technical report, Industry Canada, 2006.

[58] K. Shanker. Exact analysis of a two-echelon inventory system for recoverable items under batch inspection policy. *Naval Research Logistics*, 28(4):579–601, 1981.

[59] Z.-J. M. Shen. A multi-commodity supply chain design problem. *IIE Transactions*, 37(8):753–762, 2005.

[60] Z.-J. M. Shen and M. Daskin. Trade-offs between customer service and cost in integrated supply chain design. *Manufacturing & Service Operations Management*, 7(3): 188–207, 2005.

[61] Z.-J. M. Shen, C. Coullard, and M. Daskin. A joint location-inventory model. *Transportation Science*, 37(1):40–55, 2003.

[62] C. C. Sherbrooke. Metric: A multi-echelon technique for recoverable item control. *Operations Research*, 16(1):122–141, 1968.

[63] C. C. Sherbrooke. Vari-metric: Improved approximation for multi-indenture, multi-echelon availability models. *Operations Research*, 34(2):311–319, 1986.

[64] C. C. Sherbrooke. *Optimal Inventory Modeling of Systems: Multi-Echelon Techniques*. Kluwer Academic Publishers, Boston, second edition, 2004.

[65] R. M. Simon. Stationary properties of a two-echelon inventory model for low demand items. *Operations Research*, 19(3):761–773, 1971.

[66] F. M. Slay. Vari-metric: An approach to modeling multi-echelon resupply when the demand process is poisson with a gamma prior. Technical Report AF301–3, Logistics Management Institute, Washington, D.C., 1984.

[67] L. V. Snyder. Facility location under uncertainty: a review. *IIE Transactions*, 38(7): 547–564, 2006.

[68] L. V. Snyder, M. S. Daskin, and C.-P. Teo. The stochastic location model with risk pooling. *European Journal of Operational Research*, 179(3):1221–1238, 2007.

[69] E. Topan and Z. P. Bayindir. Multi-item two-echelon spare parts inventory control problem with batch ordering in the central warehouse under compound poisson demand. *Journal of the Operational Research Society*, 63(8):1143–1152, 2012.

[70] N. Vidyarthi, E. Celebi, S. Elhedhli, and E. Jewkes. Integrated production-inventory-distribution system design with risk pooling: Model formulation and heuristic solution. *Transportation Science*, 41(3):392–408, 2007.

[71] H. Wong, B. Kranenburg, G.-J. V. Houtum, and D. Cattrysse. Efficient heuristics for two-echelon spare parts inventory systems with an aggregate mean waiting time constraint per local warehouse. *OR Spectrum*, 29(4):699–722, 2007.

APPENDICES

The results presented in Chapter 6 are summary of the following results.

Appendix A

Daskin Results

Table A.1: Results for Daskin Metric 49_v1
$J = 49, I = 49, \lambda = 247.05, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 2000$, $\rho = 0.9, \tau = 5.5, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	8,	71.69	10	4.14	3.14	0.01	10	815.52	0	3.3	48400	170934	0.88
1	32,	73.45	10	4.14	3.14	0.01	10	835.66	0	3.38	48800	170934	1.54
2	30,	64.25	10	4.14	3.14	0.01	10	591.68	0	2.39	49500	138458	2.2
3	31,	137.61	10	4.14	3.14	0.01	10	1268.88	0	5.14	54600	138458	2.81
4	28,	139.47	10	4.14	3.14	0.01	10	1286.05	0	5.21	54900	138458	3.39
5	44,	-	-	-	-	-	-	-	-	-	59500	138458	3.8
6	14,	77.54	10	4.14	3.14	0.01	10	714.4	0	2.89	60800	138458	4.39
7	15,	48.72	10	4.14	3.14	0.01	10	388.08	0	1.57	61500	119919	5
8	36,	116.08	10	4.14	3.14	0.01	10	926.63	0	3.75	61700	119919	5.64
9	33,	116.72	10	4.14	3.14	0.01	10	931.72	0	3.77	64200	119919	6.62
10	46,	-	-	-	-	-	-	-	-	-	67900	119919	7.11
11	49,	-	-	-	-	-	-	-	-	-	68700	119919	7.62
12	20,	122.36	10	4.14	3.14	0.01	10	976.87	0	3.95	70900	119919	8.9
13	3,	-	-	-	-	-	-	-	-	-	72600	119919	9.89
14	17,	90.16	10	4.14	3.14	0.01	10	719.39	0	2.91	74400	119919	13.22
15	16,	69.87	10	4.14	3.14	0.01	10	557.22	0	2.26	75200	119919	16.19
16	26,	-	-	-	-	-	-	-	-	-	79000	119919	20.74
17	5,44	-	-	-	-	-	-	-	-	-	97900	119919	30.85
18	5,29	-	-	-	-	-	-	-	-	-	98700	119919	41.16
19	37,	-	-	-	-	-	-	-	-	-	99000	119919	53.31
20	5,43	-	-	-	-	-	-	-	-	-	101600	119919	70.49
21	5,35	-	-	-	-	-	-	-	-	-	105600	119919	79.46
22	5,41	-	-	-	-	-	-	-	-	-	106100	119919	86.02
23	5,46	-	-	-	-	-	-	-	-	-	106300	119919	96.47
24	5,49	-	-	-	-	-	-	-	-	-	107100	119919	104.65
25	3,5	-	-	-	-	-	-	-	-	-	111000	119919	114.01
26	5,24	-	-	-	-	-	-	-	-	-	115500	119919	126.28
27	5,18	-	-	-	-	-	-	-	-	-	116200	119919	137
28	5,26	-	-	-	-	-	-	-	-	-	117400	119919	149.04
29	29,44	-	-	-	-	-	-	-	-	-	119800	119919	159.63
30	15	48.72	10	4.14	3.14	0.01	10	388.08	0	1.57	119919	119919	168.27

Table A.2: Results for Daskin Approximate 49_v1
$J = 49, I = 49, \lambda = 247.05, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 2000$, $\rho = 0.9, \tau = 5.5, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	8,	71.69	10	4.14	3.14	0.01	10	815.52	0	3.3	48400	170934	0.66
1	32,	73.45	10	4.14	3.14	0.01	10	835.66	0	3.38	48800	170934	1.31
2	30,	64.25	10	4.14	3.14	0.01	10	591.68	0	2.39	49500	138458	1.96
3	31,	137.61	10	4.14	3.14	0.01	10	1268.88	0	5.14	54600	138458	2.45
4	28,	139.47	10	4.14	3.14	0.01	10	1286.05	0	5.21	54900	138458	2.81
5	44,	-	-	-	-	-	-	-	-	-	59500	138458	3.17
6	14,	77.54	10	4.14	3.14	0.01	10	714.4	0	2.89	60800	138458	3.54
7	15,	48.72	10	4.14	3.14	0.01	10	388.08	0	1.57	61500	119919	3.91
8	36,	116.08	10	4.14	3.14	0.01	10	926.63	0	3.75	61700	119919	4.29
9	33,	116.72	10	4.14	3.14	0.01	10	931.72	0	3.77	64200	119919	4.77
10	46,	-	-	-	-	-	-	-	-	-	67900	119919	5.22
11	49,	-	-	-	-	-	-	-	-	-	68700	119919	5.67
12	20,	122.36	10	4.14	3.14	0.01	10	976.87	0	3.95	70900	119919	6.7
13	3,	-	-	-	-	-	-	-	-	-	72600	119919	7.63
14	17,	90.16	10	4.14	3.14	0.01	10	719.39	0	2.91	74400	119919	10.65
15	16,	69.87	10	4.14	3.14	0.01	10	557.22	0	2.26	75200	119919	14.28
16	26,	-	-	-	-	-	-	-	-	-	79000	119919	18.9
17	5,44	-	-	-	-	-	-	-	-	-	97900	119919	29.22
18	5,29	-	-	-	-	-	-	-	-	-	98700	119919	45.41
19	37,	-	-	-	-	-	-	-	-	-	99000	119919	57.38
20	5,43	-	-	-	-	-	-	-	-	-	101600	119919	73.77
21	5,35	-	-	-	-	-	-	-	-	-	105600	119919	82.28
22	5,41	-	-	-	-	-	-	-	-	-	106100	119919	88.86
23	5,46	-	-	-	-	-	-	-	-	-	106300	119919	100.18
24	5,49	-	-	-	-	-	-	-	-	-	107100	119919	108.74
25	3,5	-	-	-	-	-	-	-	-	-	111000	119919	118.84
26	5,24	-	-	-	-	-	-	-	-	-	115500	119919	132.67
27	5,18	-	-	-	-	-	-	-	-	-	116200	119919	145.4
28	5,26	-	-	-	-	-	-	-	-	-	117400	119919	157.6
29	29,44	-	-	-	-	-	-	-	-	-	119800	119919	169.64
30	15	48.72	10	4.14	3.14	0.01	10	388.08	0	1.57	119919	119919	179.28

Table A.3: Results for Daskin Metric 88_v1
$J = 88, I = 88, \lambda = 44.8400, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 2000,\ \rho = 0.9, \tau = 5.5, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	67	38.33	10	4.14	3.14	0.07	10	200.8	0	4.48	48800	79126	3.22
1	56	28.56	10	4.14	3.14	0.07	10	130.55	0	2.91	49500	69289	4.46
2	10	-	-	-	-	-	-	-	-	-	49700	69289	5.65
3	34	24.67	10	4.14	3.14	0.07	10	109.3	0	2.44	50700	67303	8.96
4	17	8.48	10	4.14	3.14	0.07	10	31.66	0	0.71	53500	58457	12.39
5	55	-	-	-	-	-	-	-	-	-	54600	58457	15.82
6	47	48.32	10	4.14	3.14	0.07	10	186.94	0	4.17	54600	58457	21.08
7	29	-	-	-	-	-	-	-	-	-	54900	58457	32.3
8	18	54.01	10	4.14	3.14	0.07	10	209.12	0	4.66	55700	58457	43.77
9	31	45.39	10	4.14	3.14	0.07	10	175.5	0	3.91	56100	58457	55.78
10	50	-	-	-	-	-	-	-	-	-	56700	58457	67.15
11	4	-	-	-	-	-	-	-	-	-	58000	58457	79
12	17	8.48	10	4.14	3.14	0.07	10	31.66	0	0.71	58457	58457	90.52

Table A.4: Results for Daskin Approximate 88_v1
$J = 88, I = 88, \lambda = 44.8400, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 2000,\ \rho = 0.9, \tau = 5.5, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	67	38.33	10	4.14	3.14	0.07	10	200.8	0	4.48	48800	79126	3.07
1	56	28.56	10	4.14	3.14	0.07	10	130.55	0	2.91	49500	69289	4.2
2	10	-	-	-	-	-	-	-	-	-	49700	69289	5.41
3	34	24.67	10	4.14	3.14	0.07	10	109.3	0	2.44	50700	67303	8.81
4	17	8.48	10	4.14	3.14	0.07	10	31.66	0	0.71	53500	58457	12.12
5	55	-	-	-	-	-	-	-	-	-	54600	58457	15.66
6	47	48.32	10	4.14	3.14	0.07	10	186.94	0	4.17	54600	58457	20.93
7	29	-	-	-	-	-	-	-	-	-	54900	58457	32.67
8	18	54.01	10	4.14	3.14	0.07	10	209.12	0	4.66	55700	58457	45.68
9	31	45.39	10	4.14	3.14	0.07	10	175.5	0	3.91	56100	58457	57.84
10	50	-	-	-	-	-	-	-	-	-	56700	58457	69.47
11	4	-	-	-	-	-	-	-	-	-	58000	58457	80.99
12	17	8.48	10	4.14	3.14	0.07	10	31.66	0	0.71	58457	58457	93.65

Table A.5: Results for Daskin Metric 150_v1
$J = 150, I = 150, \lambda = 58.1970, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 2000$, $\rho = 0.9, \tau = 5.5, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	150	-	-	-	-	-	-	-	-	-	100000	1000000	607.59
1	88	-	-	-	-	-	-	-	-	-	100000	1000000	614.16
2	126	5.55	10	4.14	3.14	0.05	10	37.82	0	0.65	100000	105880	623.18
3	95	10.93	10	4.14	3.14	0.05	10	75.77	0	1.3	100000	105880	648.81
4	112	33.43	10	4.14	3.14	0.05	10	234.57	0	4.03	100000	105880	659.29
5	84	-	-	-	-	-	-	-	-	-	100000	105880	669.21
6	55	27.8	10	4.14	3.14	0.05	10	194.84	0	3.35	100000	105880	679.92
7	149	37.41	10	4.14	3.14	0.05	10	262.66	0	4.51	100000	105880	690.75
8	147	-	-	-	-	-	-	-	-	-	100000	105880	701.38
9	62	-	-	-	-	-	-	-	-	-	100000	105880	711.85
10	77	24.49	10	4.14	3.14	0.05	10	171.5	0	2.95	100000	105880	738.98
11	143	21.18	10	4.14	3.14	0.05	10	148.13	0	2.55	100000	105880	750.43
12	75	-	-	-	-	-	-	-	-	-	100000	105880	778.56
13	56	21.48	10	4.14	3.14	0.05	10	150.22	0	2.58	100000	105880	804.63
14	80	9.89	10	4.14	3.14	0.05	10	68.46	0	1.18	100000	105880	830.75
15	74	-	-	-	-	-	-	-	-	-	100000	105880	858.28
16	141	-	-	-	-	-	-	-	-	-	100000	105880	870.76
17	4	-	-	-	-	-	-	-	-	-	100000	105880	885.57
18	8	-	-	-	-	-	-	-	-	-	100000	105880	899.7
19	10	-	-	-	-	-	-	-	-	-	100000	105880	914
20	92	44.8	10	4.14	3.14	0.05	10	314.82	0	5.41	100000	105880	927.42
21	13	12.71	10	4.14	3.14	0.05	10	88.31	0	1.52	100000	105880	942.35
22	79	9.29	10	4.14	3.14	0.05	10	64.19	0	1.1	100000	105880	981.16
23	78	38.16	10	4.14	3.14	0.05	10	267.95	0	4.6	100000	105880	1020.06
24	59	-	-	-	-	-	-	-	-	-	100000	105880	1050.98
25	49	-	-	-	-	-	-	-	-	-	100000	105880	1092.7
26	52	-	-	-	-	-	-	-	-	-	100000	105880	1125.24
27	42	-	-	-	-	-	-	-	-	-	100000	105880	1159.04
28	41	28.38	10	4.14	3.14	0.05	10	198.93	0	3.42	100000	105880	1192.68
29	33	20.58	10	4.14	3.14	0.05	10	143.91	0	2.47	100000	105880	1226.92
30	46	34.86	10	4.14	3.14	0.05	10	244.67	0	4.2	100000	105880	1261.23
31	109	-	-	-	-	-	-	-	-	-	100000	105880	1293.43
32	69	-	-	-	-	-	-	-	-	-	100000	105880	1327.53

Continued on Next Page…

Table A.5 – Continued

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
33	17	6.31	10	4.14	3.14	0.05	10	43.16	0	0.74	100000	105880	1361.68
34	25	31.74	10	4.14	3.14	0.05	10	222.67	0	3.83	100000	105880	1399.15
35	28	-	-	-	-	-	-	-	-	-	100000	105880	1431.89
36	27	-	-	-	-	-	-	-	-	-	100000	105880	1472.27
37	107	-	-	-	-	-	-	-	-	-	100000	105880	1511.64
38	138	-	-	-	-	-	-	-	-	-	100000	105880	1555.22
39	37	-	-	-	-	-	-	-	-	-	100000	105880	1595.51
40	30	32.75	10	4.14	3.14	0.05	10	229.81	0	3.95	100000	105880	1630.85
41	124	35.33	10	4.14	3.14	0.05	10	247.99	0	4.26	100000	105880	1664.78
42	18	38.94	10	4.14	3.14	0.05	10	273.48	0	4.7	100000	105880	1701.21
43	26	-	-	-	-	-	-	-	-	-	100000	105880	1741.36
44	126	5.55	10	4.14	3.14	0.05	10	37.82	0	0.65	105880	105880	1889.84

Table A.6: Results for Daskin Approximate 150_v1
$J = 150, I = 150, \lambda = 58.1970, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 2000,\ \rho = 0.9, \tau = 5.5, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	150	-	-	-	-	-	-	-	-	-	100000	1000000	611.83
1	88	-	-	-	-	-	-	-	-	-	100000	1000000	617.9
2	126	5.55	10	4.14	3.14	0.05	10	37.82	0	0.65	100000	105880	625.94
3	95	10.93	10	4.14	3.14	0.05	10	75.77	0	1.3	100000	105880	650.17
4	112	33.43	10	4.14	3.14	0.05	10	234.57	0	4.03	100000	105880	659.73
5	84	-	-	-	-	-	-	-	-	-	100000	105880	669.14
6	55	27.8	10	4.14	3.14	0.05	10	194.84	0	3.35	100000	105880	678.74
7	149	37.41	10	4.14	3.14	0.05	10	262.66	0	4.51	100000	105880	688.39
8	147	-	-	-	-	-	-	-	-	-	100000	105880	697.78
9	62	-	-	-	-	-	-	-	-	-	100000	105880	707.67
10	77	24.49	10	4.14	3.14	0.05	10	171.5	0	2.95	100000	105880	733.11
11	143	21.18	10	4.14	3.14	0.05	10	148.13	0	2.55	100000	105880	743.97
12	75	-	-	-	-	-	-	-	-	-	100000	105880	773.64
13	56	21.48	10	4.14	3.14	0.05	10	150.22	0	2.58	100000	105880	798.47
14	80	9.89	10	4.14	3.14	0.05	10	68.46	0	1.18	100000	105880	822.9
15	74	-	-	-	-	-	-	-	-	-	100000	105880	849.97
16	141	-	-	-	-	-	-	-	-	-	100000	105880	862.36
17	4	-	-	-	-	-	-	-	-	-	100000	105880	876.93

Continued on Next Page...

Table A.6 – Continued

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
18	8	-	-	-	-	-	-	-	-	-	100000	105880	890.26
19	10	-	-	-	-	-	-	-	-	-	100000	105880	903.84
20	92	44.8	10	4.14	3.14	0.05	10	314.82	0	5.41	100000	105880	915.98
21	13	12.71	10	4.14	3.14	0.05	10	88.31	0	1.52	100000	105880	928.94
22	79	9.29	10	4.14	3.14	0.05	10	64.19	0	1.1	100000	105880	965.48
23	78	38.16	10	4.14	3.14	0.05	10	267.95	0	4.6	100000	105880	999.44
24	59	-	-	-	-	-	-	-	-	-	100000	105880	1027.74
25	49	-	-	-	-	-	-	-	-	-	100000	105880	1066.83
26	52	-	-	-	-	-	-	-	-	-	100000	105880	1097.22
27	42	-	-	-	-	-	-	-	-	-	100000	105880	1128.39
28	41	28.38	10	4.14	3.14	0.05	10	198.93	0	3.42	100000	105880	1158.96
29	33	20.58	10	4.14	3.14	0.05	10	143.91	0	2.47	100000	105880	1190.26
30	46	34.86	10	4.14	3.14	0.05	10	244.67	0	4.2	100000	105880	1220.85
31	109	-	-	-	-	-	-	-	-	-	100000	105880	1250.79
32	69	-	-	-	-	-	-	-	-	-	100000	105880	1282.68
33	17	6.31	10	4.14	3.14	0.05	10	43.16	0	0.74	100000	105880	1313.34
34	25	31.74	10	4.14	3.14	0.05	10	222.67	0	3.83	100000	105880	1346.82
35	28	-	-	-	-	-	-	-	-	-	100000	105880	1376.75
36	27	-	-	-	-	-	-	-	-	-	100000	105880	1416.47
37	107	-	-	-	-	-	-	-	-	-	100000	105880	1452.59
38	138	-	-	-	-	-	-	-	-	-	100000	105880	1491.13
39	37	-	-	-	-	-	-	-	-	-	100000	105880	1527.6
40	30	32.75	10	4.14	3.14	0.05	10	229.81	0	3.95	100000	105880	1558.44
41	124	35.33	10	4.14	3.14	0.05	10	247.99	0	4.26	100000	105880	1587.4
42	18	38.94	10	4.14	3.14	0.05	10	273.48	0	4.7	100000	105880	1618.93
43	26	-	-	-	-	-	-	-	-	-	100000	105880	1654.2
44	126	5.55	10	4.14	3.14	0.05	10	37.82	0	0.65	105880	105880	1768.38

Table A.7: Results for Daskin Metric 49_v2
$J = 49, I = 49, \lambda = 247.05, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 500,\ \rho = 0.5, \tau = 1.5, C_j = 5$ for every j

Iter	X_j	Z_j	LB	UB	Time
0	5,29,31,35,36	-	282200	10000000	0.4
1	5,31,35,36,41	7.52	289600	313161	0.63
2	5,20,29,31,35	-	291400	313161	0.81
3	5,21,29,35,36	-	295500	313161	1
4	5,20,31,35,41	7.58	298800	313161	1.28
5	5,24,31,36,41	7.45	299500	313161	1.59
6	5,21,35,36,41	7.87	302900	313161	2.04
7	5,21,24,36,41	7.8	312800	313161	2.51
8	5,31,35,36,41	7.52	313161	313161	2.77

Table A.8: Results for Daskin Approximate 49_v2
$J = 49, I = 49, \lambda = 247.05, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 500,\ \rho = 0.5, \tau = 1.5, C_j = 5$ for every j

Iter	X_j	Z_j	LB	UB	Time
0	5,29,31,35,36	-	282200	10000000	0.4
1	5,31,35,36,41	7.52	289600	313161	0.69
2	5,20,29,31,35	-	291400	313161	0.98
3	5,21,29,35,36	-	295500	313161	1.15
4	5,20,31,35,41	7.58	298800	313161	1.35
5	5,24,31,36,41	7.45	299500	313161	1.55
6	5,21,35,36,41	7.87	302900	313161	1.88
7	5,21,24,36,41	7.8	312800	313161	2.38
8	5,31,35,36,41	7.52	313161	313161	2.63

Table A.9: Results for Daskin Metric 88_v2
$J = 88, I = 88, \lambda = 44.8400, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 500,\ \rho = 0.5, \tau = 1.5, C_j = 5$ for every j

Iter	X_j	Z_j	LB	UB	Time
0	15,22,46,47,55,65,75	-	417000	419600	2.3
1	15,22,46,47,55,65,75	0.61	419600	419600	2.74

Table A.10: Results for Daskin Approximate 88_v2
$J = 88, I = 88, \lambda = 44.8400, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 500,\ \rho = 0.5, \tau = 1.5, C_j = 5$ for every j

Iter	X_j	Z_j	LB	UB	Time
0	15,22,46,47,55,65,75	-	417000	419601	2.28
1	15,22,46,47,55,65,75	0.61	419601	419601	2.61

Table A.11: Results for Daskin Metric 150_v2
$J = 150, I = 150, \lambda = 58.1970, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 500,\ \rho = 0.5, \tau = 1.5, C_j = 5$ for every j

Iter	X_j	Z_j	LB	UB	Time
0	33,84,102,121,144,145	0.77	600000	604652	608.95
1	33,84,96,102,144,145	0.75	600000	604506	610.2
2	77,84,96,102,144,145	0.77	600000	604506	611.46
3	33,73,84,102,144,145	0.75	600000	604506	613.21
4	39,77,84,102,144,145	0.7	600000	604203	615.21
5	33,73,84,139,144,145	0.76	600000	604203	617.06
6	33,39,84,102,144,145	0.68	600000	604086	618.87
7	77,84,96,144,145,146	0.72	600000	604086	620.93
8	29,33,39,61,84,145	0.63	600000	603821	624.28
9	77,84,90,96,145,146	0.73	600000	603821	625.64
10	21,33,39,61,84,145	0.63	600000	603820	628.97
11	33,39,84,133,144,145	0.7	600000	603820	632.23
12	48,84,129,144,145,149	0.73	600000	603820	635.88
13	10,29,39,61,78,145	0.67	600000	603820	639.06

Continued on Next Page…

Table A.11 – Continued

Iter	X_j	Z_j	LB	UB	Time
14	10,29,39,78,145,146	0.67	600000	603820	642.02
15	33,84,102,125,144,145	0.78	600000	603820	643.31
16	29,61,77,84,96,145	0.73	600000	603820	644.93
17	33,64,84,102,144,145	0.71	600000	603820	647.26
18	33,84,102,104,144,145	0.79	600000	603820	649.25
19	33,84,97,102,144,145	0.75	600000	603820	651.45
20	33,84,102,103,144,145	0.73	600000	603820	653.5
21	5,33,84,102,144,145	0.76	600000	603820	655.63
22	33,84,102,108,144,145	0.78	600000	603820	657.9
23	33,54,84,102,144,145	0.75	600000	603820	660.17
24	33,84,102,111,144,145	0.76	600000	603820	662.16
25	61,77,84,90,96,145	0.73	600000	603820	663.91
26	33,84,98,102,144,145	0.78	600000	603820	665.98
27	33,65,84,102,144,145	0.76	600000	603820	668.02
28	1,33,84,102,144,145	0.76	600000	603820	670.07
29	33,48,84,102,144,145	0.68	600000	603820	672.04
30	33,60,84,102,144,145	0.76	600000	603820	673.73
31	61,77,84,91,96,145	0.73	600000	603820	675.39
32	33,84,96,139,144,145	0.76	600000	603820	677.15
33	33,84,102,131,144,145	0.76	600000	603820	678.5
34	33,84,102,136,144,145	0.78	600000	603820	680.47
35	19,33,84,102,144,145	0.75	600000	603820	682.65
36	33,84,102,122,144,145	0.75	600000	603820	684.65
37	12,33,84,102,144,145	0.75	600000	603820	687.28
38	48,84,91,145,146,149	0.69	600000	603820	691.2
39	20,33,84,102,144,145	0.79	600000	603820	692.66
40	33,81,84,102,144,145	0.76	600000	603820	694.22
41	6,33,39,84,144,145	0.67	600000	603820	698.58
42	33,39,61,84,91,145	0.63	600000	603820	702.81
43	33,39,84,129,144,145	0.67	600000	603820	706.92
44	33,36,84,102,144,145	0.77	600000	603820	709.52
45	5,21,33,84,145,146	0.71	600000	603820	713.56
46	21,33,84,104,145,146	0.74	600000	603820	717.61
47	33,39,61,84,144,145	0.62	600000	603753	721.6
48	33,39,50,84,144,145	0.69	600000	603753	725.64
49	1,33,61,84,90,145	0.71	600000	603753	729.27
50	6,30,48,84,144,145	0.7	600000	603753	733.39

Continued on Next Page...

Table A.11 – Continued

Iter	X_j	Z_j	LB	UB	Time
501	19,29,61,77,84,145	0.73	600000	603751	2668.26
502	33,84,98,135,144,145	0.79	600000	603751	2674.43
503	29,30,48,84,145,146	0.66	600000	603751	2680.74
504	6,33,84,136,144,145	0.77	600000	603751	2687.24
505	39,46,61,84,144,145	0.66	600000	603751	2693.48
506	33,54,84,129,144,145	0.75	600000	603751	2699.76
507	39,46,84,144,145,146	0.66	600000	603751	2707.6
508	48,61,77,84,144,145	0.65	600000	603751	2712.98
509	39,61,84,91,145,149	0.66	600000	603751	2719.06
510	39,46,84,135,144,145	0.73	600000	603751	2725.22
511	39,84,102,112,144,145	0.71	600000	603751	2726.9
512	2,39,84,112,144,145	0.73	600000	603751	2733.25
513	29,48,61,84,112,145	0.66	600000	603751	2739.32
514	30,39,66,84,144,145	0.69	600000	603751	2745.9
515	33,61,84,104,144,145	0.73	600000	603751	2749.4
516	19,33,84,132,144,145	0.75	600000	603751	2755.77
517	33,81,84,90,145,146	0.71	600000	603751	2758.75
518	33,50,84,96,144,145	0.76	600000	603751	2762.53
519	21,30,48,84,145,146	0.66	600000	603751	2768.93
520	6,39,77,84,144,145	0.69	600000	603751	2775.73
521	1,33,84,91,145,146	0.72	600000	603751	2780.27
522	33,36,84,91,145,146	0.72	600000	603751	2784.99
523	33,84,90,136,145,146	0.73	600000	603751	2790.55
524	6,33,84,104,144,145	0.78	600000	603751	2796.49
525	2,33,84,104,144,145	0.8	600000	603751	2802.05
526	5,33,84,91,145,146	0.71	600000	603751	2806.81
527	33,73,84,91,145,146	0.71	600000	603751	2811.65
528	33,84,108,129,144,145	0.77	600000	603751	2816.03
529	33,84,91,103,145,146	0.68	600000	603751	2820.07
530	33,66,84,136,144,145	0.78	600000	603751	2825.25
531	12,33,84,90,145,146	0.71	600000	603751	2828.97
532	29,33,73,84,145,146	0.71	600000	603751	2833.16
533	6,33,73,84,144,145	0.75	600000	603751	2837.35
534	21,33,73,84,145,146	0.71	600000	603751	2843.01
535	1,6,33,84,144,145	0.76	600000	603751	2847.27
536	19,33,61,84,144,145	0.69	600000	603751	2852.19
537	1,33,84,144,145,146	0.7	600000	603751	2856.61

Continued on Next Page. . .

Table A.11 – Continued

Iter	X_j	Z_j	LB	UB	Time
1041	21,33,36,61,84,145	0.72	600000	603751	5046.53
1042	33,84,87,122,144,145	0.77	600000	603751	5051.01
1043	33,84,131,134,144,145	0.76	600000	603751	5055.2
1044	33,84,89,97,144,145	0.77	600000	603751	5060.24
1045	33,84,108,133,144,145	0.8	600000	603751	5065.73
1046	1,29,33,61,84,145	0.72	600000	603751	5071.4
1047	31,33,84,122,144,145	0.77	600000	603751	5076.52
1048	33,84,122,133,144,145	0.77	600000	603751	5080.98
1049	33,60,66,84,144,145	0.76	600000	603751	5086.33
1050	33,50,84,131,144,145	0.77	600000	603751	5091.67
1051	33,54,84,142,144,145	0.77	600000	603751	5096.02
1052	33,65,84,134,144,145	0.77	600000	603751	5100
1053	1,33,84,118,144,145	0.77	600000	603751	5104.81
1054	33,84,131,135,144,145	0.77	600000	603751	5109.83
1055	33,84,89,131,144,145	0.77	600000	603751	5114.12
1056	33,60,84,133,144,145	0.78	600000	603751	5118.95
1057	33,54,84,132,144,145	0.76	600000	603751	5122.96
1058	33,54,57,84,144,145	0.77	600000	603751	5127.2
1059	33,84,131,133,144,145	0.78	600000	603751	5132.71
1060	33,50,84,122,144,145	0.77	600000	603751	5137.18
1061	33,57,84,121,144,145	0.78	600000	603751	5142.03
1062	33,84,111,134,144,145	0.77	600000	603751	5147.09
1063	31,33,60,84,144,145	0.78	600000	603751	5151.22
1064	33,60,84,89,144,145	0.77	600000	603751	5155.73
1065	33,57,84,131,144,145	0.77	600000	603751	5160.79
1066	33,84,122,132,144,145	0.76	600000	603751	5166.75
1067	33,57,84,122,144,145	0.77	600000	603751	5171.7
1068	33,60,84,132,144,145	0.76	600000	603751	5176.68
1069	33,84,111,133,144,145	0.78	600000	603751	5182.09
1070	33,50,65,84,144,145	0.77	600000	603751	5187.54
1071	33,65,84,135,144,145	0.77	600000	603751	5192.71
1072	33,65,84,89,144,145	0.77	600000	603751	5198.66
1073	33,39,84,144,145,146	0.62	603751	603751	5203.06

Table A.12: Results for Daskin Approximate 150_v2
$J = 150, I = 150, \lambda = 58.1970, h_o = 50, h_j = 50$ for every $j, p = 150$
$d_{max} = 500$, $\rho = 0.5, \tau = 1.5, C_j = 5$ for every j

Iter	X_j	Z_j	LB	UB	Time
0	33,84,102,121,144,145	0.77	600000	604652	597.12
1	33,84,96,102,144,145	0.75	600000	604506	598.17
2	77,84,96,102,144,145	0.77	600000	604506	601.04
3	33,73,84,102,144,145	0.75	600000	604506	604.99
4	29,39,61,77,84,145	0.65	600000	603938	607.52
5	33,39,84,102,144,145	0.68	600000	603938	609.09
6	39,77,84,102,144,145	0.7	600000	603938	610.57
7	77,84,96,144,145,146	0.72	600000	603938	611.89
8	33,39,84,133,144,145	0.7	600000	603938	614.93
9	77,84,96,132,144,145	0.78	600000	603938	616.04
10	30,39,84,129,144,145	0.68	600000	603938	618.79
11	10,29,39,61,78,145	0.67	600000	603938	621.32
12	33,61,73,84,144,145	0.7	600000	603938	623.32
13	30,39,61,84,144,145	0.63	600000	603802	625.58
14	29,48,61,84,145,149	0.69	600000	603802	627.99
15	33,48,84,102,144,145	0.68	600000	603802	629.52
16	33,84,102,104,144,145	0.79	600000	603802	631.04
17	33,36,84,102,144,145	0.77	600000	603802	632.95
18	33,84,102,125,144,145	0.78	600000	603802	633.97
19	29,61,77,84,96,145	0.73	600000	603802	635.34
20	61,77,84,96,144,145	0.72	600000	603802	636.59
21	33,84,102,111,144,145	0.76	600000	603802	638.11
22	12,77,84,102,144,145	0.78	600000	603802	639.45
23	33,84,102,108,144,145	0.78	600000	603802	640.97
24	61,77,84,91,96,145	0.73	600000	603802	642.53
25	33,84,102,131,144,145	0.76	600000	603802	644.06
26	33,54,84,102,144,145	0.75	600000	603802	645.58
27	77,84,90,96,145,146	0.73	600000	603802	647
28	21,77,84,96,145,146	0.73	600000	603802	648.39
29	29,77,84,96,145,146	0.73	600000	603802	649.89
30	1,33,84,102,144,145	0.76	600000	603802	651.53
31	77,84,91,96,145,146	0.73	600000	603802	653.09
32	33,39,61,84,90,145	0.63	600000	603802	655.78

Continued on Next Page...

Table A.12 – Continued

Iter	X_j	Z_j	LB	UB	Time
501	33,61,84,103,144,145	0.67	600000	603751	1964.02
502	21,33,61,81,84,145	0.72	600000	603751	1965.78
503	39,57,78,84,144,145	0.72	600000	603751	1968.65
504	12,33,61,84,144,145	0.7	600000	603751	1972.4
505	29,33,61,81,84,145	0.72	600000	603751	1973.83
506	33,84,91,131,145,146	0.71	600000	603751	1975.6
507	30,48,84,144,145,146	0.65	600000	603751	1978.48
508	21,30,48,84,145,146	0.66	600000	603751	1981.2
509	33,84,90,103,145,146	0.68	600000	603751	1983.71
510	33,84,90,98,145,146	0.73	600000	603751	1986.23
511	33,84,87,104,144,145	0.8	600000	603751	1988.69
512	33,61,81,84,90,145	0.71	600000	603751	1989.9
513	2,33,84,136,144,145	0.79	600000	603751	1992.42
514	2,33,84,121,144,145	0.79	600000	603751	1995.32
515	2,33,84,111,144,145	0.78	600000	603751	1997.76
516	33,84,91,104,145,146	0.74	600000	603751	2000.24
517	33,84,87,98,144,145	0.79	600000	603751	2002.65
518	1,33,84,144,145,146	0.7	600000	603751	2005.3
519	33,84,90,125,145,146	0.73	600000	603751	2006.95
520	33,84,91,121,145,146	0.73	600000	603751	2009.87
521	33,50,84,136,144,145	0.79	600000	603751	2012.8
522	33,84,89,96,144,145	0.76	600000	603751	2015.27
523	33,73,84,91,145,146	0.71	600000	603751	2018.52
524	33,84,103,133,144,145	0.75	600000	603751	2020.79
525	33,84,103,144,145,146	0.67	600000	603751	2023.94
526	33,84,103,135,144,145	0.74	600000	603751	2026.23
527	6,33,73,84,144,145	0.75	600000	603751	2028.69
528	33,84,103,129,144,145	0.73	600000	603751	2031.52
529	33,73,84,90,145,146	0.71	600000	603751	2034.18
530	39,84,112,133,144,145	0.73	600000	603751	2037.08
531	33,84,98,129,144,145	0.77	600000	603751	2039.93
532	33,65,84,144,145,146	0.7	600000	603751	2043.24
533	33,84,97,129,144,145	0.75	600000	603751	2046.11
534	6,33,84,131,144,145	0.75	600000	603751	2048.89
531	33,84,98,129,144,145	0.77	600000	603751	2039.93
532	33,65,84,144,145,146	0.7	600000	603751	2043.24
533	33,84,97,129,144,145	0.75	600000	603751	2046.11

Continued on Next Page...

Table A.12 – Continued

Iter	X_j	Z_j	LB	UB	Time
1041	33,65,84,134,144,145	0.77	600000	603751	3672.98
1042	5,29,33,61,84,145	0.71	600000	603751	3676.33
1043	33,84,125,134,144,145	0.78	600000	603751	3679.47
1044	33,84,121,135,144,145	0.78	600000	603751	3682.55
1045	33,84,122,134,144,145	0.76	600000	603751	3686.23
1046	33,50,84,122,144,145	0.77	600000	603751	3689.37
1047	33,54,61,84,91,145	0.71	600000	603751	3693.03
1048	33,36,84,142,144,145	0.78	600000	603751	3696.98
1049	33,65,84,133,144,145	0.78	600000	603751	3700.65
1050	33,84,122,142,144,145	0.77	600000	603751	3704.71
1051	33,66,84,111,144,145	0.77	600000	603751	3708.41
1052	33,36,84,118,144,145	0.78	600000	603751	3711.38
1053	31,33,65,84,144,145	0.78	600000	603751	3715.05
1054	33,84,118,131,144,145	0.77	600000	603751	3718.01
1055	33,50,84,97,144,145	0.77	600000	603751	3721.13
1056	33,84,97,135,144,145	0.77	600000	603751	3724.86
1057	33,60,84,142,144,145	0.77	600000	603751	3728.24
1058	33,84,122,132,144,145	0.76	600000	603751	3731.85
1059	31,33,84,97,144,145	0.77	600000	603751	3735.22
1060	33,84,131,134,144,145	0.76	600000	603751	3738.23
1061	33,84,111,118,144,145	0.78	600000	603751	3741.26
1062	31,33,54,84,144,145	0.77	600000	603751	3744.74
1063	33,54,84,142,144,145	0.77	600000	603751	3747.59
1064	33,66,84,125,144,145	0.78	600000	603751	3750.85
1065	1,33,61,84,91,145	0.72	600000	603751	3754.36
1066	1,29,33,61,84,145	0.72	600000	603751	3757.83
1067	33,84,111,134,144,145	0.77	600000	603751	3761.09
1068	33,84,118,121,144,145	0.79	600000	603751	3764.36
1069	29,33,36,61,84,145	0.72	600000	603751	3767.83
1070	31,33,60,84,144,145	0.78	600000	603751	3770.95
1071	33,66,84,97,144,145	0.75	600000	603751	3774.78
1072	33,84,121,132,144,145	0.78	600000	603751	3778.28
1073	33,39,84,144,145,146	0.62	603751	603751	3782.29

Appendix B

Bombardier Results

Table B.1: Results for BBD Metric 20_v1
$J = 20, I = 121, \lambda = 17.44$
$d_{max} = 40,\ \rho = 0.9, \tau = 0.025, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	12	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	50000	116911	0.64
1	10	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	50000	116911	0.98
2	3	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	50000	116911	1.35
3	7	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	50000	116911	1.67
4	11	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	73346	116911	2
5	4	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	85012	116911	2.35
6	14	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	2.7
7	16	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	3.06
8	15	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	3.42
9	13	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	3.78
10	18	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	4.13
11	17	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	4.47
12	19	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	94168	116911	4.78
13	20	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	100000	116911	5.07
14	9	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	113206	116911	5.38
15	12	57.23	8	2.87	3.87	0.22	10	0.4	2.5	0.02	116911	116911	5.68

Table B.2: Results for BBD Approximate 20_v1
$J = 20, I = 121, \lambda = 17.44$
$d_{max} = 40$, $\rho = 0.9, \tau = 0.025, C_j = 10$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	12	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	50000	124626	0.68
1	10	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	50000	124626	1.05
2	3	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	50000	124626	1.46
3	7	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	50000	124626	1.81
4	11	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	73346	124626	2.14
5	4	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	85012	124626	2.49
6	14	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	2.84
7	16	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	3.19
8	15	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	3.55
9	13	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	3.96
10	18	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	4.33
11	17	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	4.72
12	19	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	94168	124626	5.08
13	20	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	100000	124626	5.4
14	9	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	113206	124626	5.74
15	12	59.88	10	4.14	3.14	0.18	10	0.09	2.92	0.005	124626	124626	6.04

Table B.3: Results for BBD Metric 20_v2
$J = 20, I = 121, \lambda = 17.44$
$d_{max} = 25$, $\rho = 0.5, \tau = 0.01, C_j = 5$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	10,12,	-	-	-	-	-	-	-	-	-	100000	900000	0.65
1	7,12,	-	-	-	-	-	-	-	-	-	100000	900000	1.12
2	3,12,	-	-	-	-	-	-	-	-	-	100000	900000	1.54
3	10,11,	35.42	3	2.13	0.13	0.01	8	0.17	4.02	0.02	123346	190998	3.19
4	4,12,	-	-	-	-	-	-	-	-	-	135012	190998	4.71
5	12,19,	29.52	1	0.5	0.5	0.03	9	0.14	4.61	0.02	144168	190998	5.41
6	12,14,	-	-	-	-	-	-	-	-	-	144168	190998	6.75
7	12,17,	-	-	-	-	-	-	-	-	-	144168	190998	8.09
8	12,18	-	-	-	-	-	-	-	-	-	144168	190998	9.26
9	10,16,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	144168	190998	14.41

Continued on Next Page…

Table B.3 – Continued

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
10	3,13,	-	-	-	-	-	-	-	-	-	144168	190998	18.72
11	10,13,	-	-	-	-	-	-	-	-	-	144168	190998	22.79
12	7,13,	-	-	-	-	-	-	-	-	-	144168	190998	27.94
13	3,16,	-	-	-	-	-	-	-	-	-	144168	190998	37.5
14	10,15,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	144168	190998	38.07
15	7,10,12	-	-	-	-	-	-	-	-	-	150000	190998	46.09
16	3,10,12	-	-	-	-	-	-	-	-	-	150000	190998	57.99
17	3,7,12	-	-	-	-	-	-	-	-	-	150000	190998	67.65
18	4,11,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	158358	190998	81.11
19	7,9,	-	-	-	-	-	-	-	-	-	163206	190998	116.05
20	3,9,	-	-	-	-	-	-	-	-	-	163206	190998	176.3
21	9,10,	-	-	-	-	-	-	-	-	-	163206	190998	221.64
22	11,14,	34.54	2	1.25	0.25	0.01	9	0.12	4.85	0.01	167514	190998	233.41
23	2,3,	-	-	-	-	-	-	-	-	-	168663	190998	274.42
24	2,10,	-	-	-	-	-	-	-	-	-	168663	190998	282.34
25	2,7,	-	-	-	-	-	-	-	-	-	168663	190998	310.97
26	3,11,12	35.35	0	0	1	0.06	11	0.16	6.14	0.02	173346	190998	341.43
27	7,11,12	35.24	0	0	1	0.06	11	0.16	6.13	0.03	173346	190998	369.68
28	3,7,11	45.56	3	2.13	0.13	0.01	10	0.14	5.99	0.02	173346	190998	415.23
29	1,10,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	175653	190998	460.14
30	6,12,	-	-	-	-	-	-	-	-	-	177611	190998	490.73
31	4,16,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	179180	190998	521.02
32	4,13,	-	-	-	-	-	-	-	-	-	179180	190998	561.56
33	4,15,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	179180	190998	569.62
34	4,7,12	-	-	-	-	-	-	-	-	-	185012	190998	578.44
35	3,4,12	-	-	-	-	-	-	-	-	-	185012	190998	615.39
36	4,10,12	-	-	-	-	-	-	-	-	-	185012	190998	659.26
37	14,16,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	188336	190998	676.49
38	13,19,	29.52	1	0.5	0.5	0.03	9	0.14	4.61	0.02	188336	190998	687.57
39	13,18,	-	-	-	-	-	-	-	-	-	188336	190998	698.56
40	13,17,	-	-	-	-	-	-	-	-	-	188336	190998	738.21
41	14,15,	35.44	3	2.13	0.13	0.01	8	0.17	4.02	0.02	188336	190998	802.98
42	13,14,	-	-	-	-	-	-	-	-	-	188336	190998	831.47
43	10,11	35.42	3	2.13	0.13	0.01	8	0.17	4.02	0.02	190998	190998	863.06

Table B.4: Results for BBD Approximate 20_v2
$J = 20, I = 121, \lambda = 17.44$
$d_{max} = 25$, $\rho = 0.5, \tau = 0.01, C_j = 5$ for every j

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
0	10,12,	0	0	0	0	0	0	0	0	0	100000	900000	0.87
1	7,12,	0	0	0	0	0	0	0	0	0	100000	900000	1.34
2	3,12,	0	0	0	0	0	0	0	0	0	100000	900000	1.73
3	10,11,	35.23	3	2.13	0.13	0.01	8	0.15	4	0.02	123346	190447	3.29
4	4,12,	0	0	0	0	0	0	0	0	0	135012	190447	4.68
5	12,19,	24.23	0	0	1	0.06	9	0.14	4.11	0.02	144168	190259	5.54
6	12,14,	0	0	0	0	0	0	0	0	0	144168	190259	6.81
7	10,16,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	144168	190259	10.08
8	12,18,	0	0	0	0	0	0	0	0	0	144168	190259	11.33
9	12,17,	0	0	0	0	0	0	0	0	0	144168	190259	12.43
10	10,13,	0	0	0	0	0	0	0	0	0	144168	190259	17.44
11	7,13,	0	0	0	0	0	0	0	0	0	144168	190259	19.72
12	3,16,	0	0	0	0	0	0	0	0	0	144168	190259	26.38
13	10,15,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	144168	190259	27.69
14	3,13,	0	0	0	0	0	0	0	0	0	144168	190259	31.25
15	7,10,12	0	0	0	0	0	0	0	0	0	150000	190259	40.37
16	3,10,12	0	0	0	0	0	0	0	0	0	150000	190259	46.69
17	3,7,12	0	0	0	0	0	0	0	0	0	150000	190259	49.49
18	4,11,	35.29	3	2.13	0.13	0.01	8	0.15	4	0.02	158358	190259	60.99
19	3,9,	0	0	0	0	0	0	0	0	0	163206	190259	70.25
20	7,9,	0	0	0	0	0	0	0	0	0	163206	190259	83.97
21	9,10,	0	0	0	0	0	0	0	0	0	163206	190259	109
22	11,14,	34.61	2	1.25	0.25	0.01	9	0.12	4.85	0.01	167514	190259	117.03
23	2,10,	0	0	0	0	0	0	0	0	0	168663	190259	139.09
24	2,3,	0	0	0	0	0	0	0	0	0	168663	190259	155.39
25	2,7,	0	0	0	0	0	0	0	0	0	168663	190259	169.11
26	7,11,12	33.37	0	0	1	0.06	11	0.06	6.04	0.01	173346	190259	193.61
27	3,11,12	44.31	2	1.25	0.25	0.01	11	0.07	6.8	0.01	173346	190259	227.89
28	3,7,11	45.42	3	2.13	0.13	0.01	10	0.13	5.98	0.01	173346	190259	260.38
29	1,10,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	175653	190259	278.1
30	6,12,	0	0	0	0	0	0	0	0	0	177611	190259	291.32
31	4,13,	0	0	0	0	0	0	0	0	0	179180	190259	315.9
32	4,15,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	179180	190259	327.7

Continued on Next Page...

Table B.4 – Continued

Iter	X_j	Z_j	S_o	$\bar{I}_0$	$\bar{B}_0$	$\overline{W}_0$	S_j	$\bar{B}_j$	$\bar{I}_j$	$\overline{W}_j$	LB	UB	Time
33	4,16,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	179180	190259	336.12
34	3,4,12	0	0	0	0	0	0	0	0	0	185012	190259	349.41
35	4,7,12	0	0	0	0	0	0	0	0	0	185012	190259	372.96
36	4,10,12	0	0	0	0	0	0	0	0	0	185012	190259	392.33
37	13,19,	24.23	0	0	1	0.06	9	0.14	4.11	0.02	188336	190259	416.32
38	13,14,	0	0	0	0	0	0	0	0	0	188336	190259	461.38
39	14,16,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	188336	190259	469.5
40	13,18,	0	0	0	0	0	0	0	0	0	188336	190259	498.5
41	13,17,	0	0	0	0	0	0	0	0	0	188336	190259	543.39
42	14,15,	35.3	3	2.13	0.13	0.01	8	0.15	4	0.02	188336	190259	551.29
43	12,19	24.23	0	0	1	0.06	9	0.14	4.11	0.02	190259	190259	562.77