NOVEL STOCHASTIC PROGRAMMING FORMULATIONS FOR ASSEMBLE-TO-ORDER SYSTEMS

NOVEL STOCHASTIC PROGRAMMING FORMULATIONS FOR ASSEMBLE-TO-ORDER SYSTEMS

By

HONGFENG LIANG, M.Eng.

A Thesis

Submitted to the Department of Computing and Software

and the School of Graduate Studies

of McMaster University

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

McMaster University © Copyright by Hongfeng Liang, July 2017

Doctor of Philosophy (2017)	McMaster University
(Computing and Software)	Hamilton, Ontario, Canada

TITLE: Novel Stochastic Programming Formulations for Assemble-to-Order Systems

AUTHOR: Hongfeng Liang
M.Eng.
McMaster University, Hamilton, Canada
B.A.S.
JiLin University, Changchun, China

SUPERVISOR: Dr. Antoine Deza

NUMBER OF PAGES: xv, 95

Abstract

We study a periodic review assemble-to-order (ATO) system introduced by Akçay and Xu (2004) which jointly optimizes the base stock levels and the component allocation with an independent base stock policy and a first-come-first-served allocation rule. The formulation is a non-smooth and thus theoretically and computationally challenging. In their computational experiments, Akçay and Xu (2004) modified the right hand side of the inventory availability constraints by substituting linear functions for piece-wise linear ones. This modification may have a significant impact on low budget levels. The optimal solutions obtained via the original formulation, i.e., the formulation without modification, include zero base stock levels for some components and thus indicate a bias against component commonality. We study the impact of component commonality on periodic review ATO systems. We show that lowering component commonality may yield a higher type-II service level. The lower degree of component commonality is achieved via separating inventories of the same component for different products. We substantiate this property via computational and theoretical approaches. We show that for low budget levels the use of separate inventories of the same component for different products can achieve a higher reward than with shared inventories. Finally, considering a simple ATO system with one component shared by two products, we characterize the budget ranges such that either separate or shared inventory component (i.e., component commonality) is beneficial.

Acknowledgements

I would like to give my gratitude to my supervisor, Dr. Antoine Deza, for his invaluable guidance, generous support and continuous encouragement to my research studies and my life.

My special thanks go to the members of the supervisory and defense committees: Dr. Joachim Arts, Dr. Antoine Deza, Dr. Frantisek Franek, Dr. Kai Huang and Dr. Ridha Khedri.

I appreciate the help and moral support from all my colleagues and in particular the members of the Advanced Optimization Laboratory.

Furthermore, I am grateful for the financial aid provided by the Queen Elizabeth II Graduate Scholarship in Science and Technology program.

Finally, I would like to thank my family and friends from the bottom of my heart for always believing in me, and for their constant love, support, and encouragement.

Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations and Symbols

n	number of components
m	number of products
i	index of component $i = 1, \ldots, n$
j	index of product $j = 1, \ldots, m$
S_i	base stock level of component $i = 1, \ldots, n$
c_i	unit base stock level cost of component $i = 1, \ldots, n$
L_i	lead time of component $i = 1, \ldots, n$
L	maximum lead time among all components; that is, $L = \max_{i=1}^{n} L_i$
w_j	time window of product j
w	maximum time window among all products; that is, $w = \max_{j=1}^{m} w_j$
k	index of period k corresponding to the duration $[k, k+1)$; $k = 0$ implies the current period; negative values of k imply previous periods
$x_{j,k}$	number of product j assembled in period k
$r_{j,k}$	reward for satisfying the demand for product j in period k
$a_{i,j}$	number of component i used to assemble one unit of product j; that is, the bill of materials (BOM)
B	budget; $\sum_{i=1}^{n} c_i \times S_i \leq B$
$P_{j,k}$	demand of product j at period k
P_j	demand of product j at the current period; that is, $P_{j,0}$
$D_{i,k}$	demand of component i at period k; that is, $D_{i,k} = \sum_{j=1}^{n} a_{i,j} P_{j,k}$
M	number of independent samples
N	number of realizations in one sample
l	index of sample $l = 1, \ldots, M$
h	index of realization $h = 1, \ldots, N$
x^+	positive part of x; that is, $x^+ = (\lvert x \rvert + x)/2$
$I_{i,k}$	inventory of component i at the end of period k;
$A_{i,k}$	replenishment order of component i arriving at period k;

Table 1: Notations

Chapter 1

Introduction

Due to the pressure of high capital costs and a competitive environment, assemble-to-order (ATO) systems play a key role in industry. ATO system can help increase the degree of product customization and reduce response time compared to make-to-stock (MTS) approaches. The key difference between ATO and MTS systems is that ATO systems eliminate the need for final product inventories. When a customer order arrives, an ATO system satisfies the order by assembling the products from component inventories, while an MTS system needs to have stocks of the final products. The advantages of ATO systems over MTS ones typically assume that product assembly times are negligible compared with the component replenishment lead times. Though ATO systems provide significant benefits, matching the product demands with the components supply efficiently is a challenging task. In particular, if the matching problem is not efficiently handled, those benefits may be offset, see Song and Zipkin [31]. We investigate the theoretical and computational aspects of the formulation of Akçay and Xu [3] which jointly optimizes the base stock levels and the component allocation on a periodic review ATO system with an independent base stock policy and a first-come-first-served (FCFS) allocation rule. In particular, we discuss the impact of substituting linear inventory availability constraints for piece-wise linear ones in the Akçay and Xu formulation and the efficiency of component commonality for ATO systems. Part of the work presents in this thesis is published in Deza et al. [9].

1.1 Preliminaries

1.1.1 ATO Systems

An ATO system requires two levels of articles, products, and components. It can be considered as a hybrid system between make-to-stock system (MTS) and make-to-order (MTO) system. A make-to-stock system only keeps the inventory in product level; that is, looks at its product inventory to see whether it can meet its demand when an order arrives. A make-to-order system does not keep any inventory. It starts its entire manufacture process when an order is received and includes component manufacture/ordering and final product assembling. An ATO system lays between MTS and MTO systems: Keeping the component inventory provides a fast response time compared to an MTO system, and provides more flexibility or customizability than an MTS system by assembling the required product when the order is received. ATO systems are usually classified based on review period policies, demand distributions, objectives of the optimization problem, decision process, replenishment policies, allocation rules etc. The review period policy of an ATO system can be single-period, multi-period, or continuous. The demand distribution can be assumed to be normal, uniform, Poisson and whether correlated across different periods. The objective function can be minimizing the production cost, the inventory cost, or the capital cost, or maximizing profit or service level. In terms of decision process, we can decide the assignment of the components when the order arrives, the base-stock levels, or both together. The allocation rule can be first-in-first-out based on component or prioritized products.

1.1.2 ATO systems terminology

A **product** in ATO system is an article needed to satisfy customers' demand. An article can be a product or a component. A **component** is a part of a product and can be used in different products. The relationship between products and components is given by the **bill-of-material (BOM)**. A BOM is a list of the components and their quantities used to assemble a product. The BOM is decided in the product design stage. In emerging industries the BOM may be created by the customers from a pre-existing BOM with some rules, e.g. changing the numbers of RAM or type of the hard drive for a desktop. An **inventory review system** is used to track the inventory. In a **periodic review** system, inventory checking is executed at regular time intervals, e.g., 3 days, a week, or a month. In a **continuous review** system, we track each item and update the inventory each time an item is consumed from the inventory. A **base stock policy** is an inventory management policy that describes the way inventory is replenished. At the beginning, we set a **base stock level**, BS and base stock policy requires that when we review the inventory, we need to bring back the inventory level to BS. So when we review the inventory, we obtain an inventory level I, the difference, $Q = BS - I$, is the replenishment order we need to place. An **FCFS** service rule meets the customer demands in the order of their arrival, without any preferences. The **lead time** of a component is the time period between the replenishment order placement and arrival. For more details see, for example, Song and Zipkin [31].

1.1.3 ATO systems literature review

As mentioned by Song [29] and by Glasserman and Wang [16], keeping inventory at the component level can increase the fill rate of customer orders which is key to ATO systems. Component commonality is an important feature of ATO systems. Component commonality is widely adopted and often preferred to offset the reduction of the economy of scale of products and move to the economy of scale in component production, and for the benefit of risk pooling of component inventory, see Mirchandani and Mishra [25]. The advantages and disadvantages of component commonality for a given model or system have been widely studied. A continuous model has been considered by van Jaarsveld and Scheller-Wolf [34] and by Song and Zhao [30]. Wang [35] assume a normal demand distribution while Song and Zhao [30] assume a Poisson process. Among periodic review models, the single period models have been extensively studied. Eynan and Rosenblatt [12] present three models to compare and analyze the effects of increasing component commonality and demonstrate that some forms of commonality might not always be beneficial. They also provide conditions for which commonality should be either employed or avoided. Mirchandani and Mishra [25] compare a non-commonality model with two different commonality models – based on whether or not the products are prioritized – for a system with two products and independent uniform demand distributions. They derive theoretical conditions when component commonality is beneficial for this specific system. Eynan and Rosenblatt [12] and Mirchandani and Mishra [25] allow the common component to be more expensive than those it replaces, while Baker et al. [5] and Gerchak et al. [15] assume that the costs of the dedicated component and the common component are identical. Those two options have their own applicable areas, and assuming the common component is cheaper than that which replaces may be reasonable due to the economies of scale. Baker et al. [5] study the effect of component commonality on optimal safety stock levels for an ATO system with two end-products and two components. They consider the problem of minimizing safety stock levels while satisfying a service level constraint under independent uniform demand distributions and show that component commonality induces a reduction in the optimal safety stock levels. Gerchak et al. [15] extend this work by investigating whether the results hold for a system with an arbitrary number of products and a general joint demand distribution. Thonemann and Brandeau [33] discuss an optimal commonality, i.e. a good BOM by considering production, inventory, setup and complexity cost in product design.

Multi-period models have been investigated relatively recently. Hillier [18] observes that component commonality is not always beneficial under a simple multi-period ATO model. Hillier [18] studies a periodic review ATO system with zero lead times and uniformly distributed demands and derives a closed-form solution for a cost minimization model with service level constraints. The results demonstrate that, for a multi-period model, the use of a common component is always beneficial if its price does not exceed the price of the replaced components. If the common component is more expensive than the replaced ones, then, in contrast to the single period case, it is almost never beneficial to use it. Hillier [17] further extends these results to systems with an arbitrary number of final products and components. Song and Zhao [30] consider a continuous review ATO system with one common component, two end products, and Poisson demand processes, and show that, while component commonality is generally beneficial, its added value depends strongly on the component costs, lead times, and allocation rules. Based on the general setting proposed by Huang and de Kok [20], our approach aims at further analyzing complicated ATO systems while taking into account component commonality. Huang and de Kok [20] consider a periodic review ATO system with component base stock policy and correlated demands and present an FCFS formulation for a cost minimization model involving the inventory holding cost, remnant stock holding cost, and backlogging cost.

Minimizing inventory level or inventory cost subject to some service level constraints is commonly used to model ATO systems, see Thonemann and Brandeau [33], Song and Zhao [30], and Mirchandani and Mishra [26]. Maximizing profit or reward is considered in, for example, Nonås [28], Jönsson et al. [22], Gerchak and Henig [14], and Akçay and Xu [3]. The difference in optimization problem objectives is not large, the complexity of non-convexity in the objective function may lead to switching minimizing and maximizing formulations. Joint optimization; that is optimizing the base stock and allocation jointly, is relatively recent. Song and Zhao [30], Muharremoglu and Yang [27], Agrawal and Cohen [1], and Hillier [17] focus on finding the optimal base stock level, and Jönsson et al. [22] aim to find an optimal allocation only. Nonås [28], Akçay and Xu [3] and Swaminathan and Tayur [32] optimize jointly the base stock and the allocation. As the demands are random variables, the computation time for a joint optimization problem is one or two orders of magnitude higher than for an allocation problem. Minimizing inventory level or inventory cost subject to some service level constraints is commonly used to model ATO systems. However, the problem we consider follows another line of research: component commonality for systems with a given budget for all the components. Jönsson and Silver [21] analyze the impact of component commonality for an ATO system with two end products and two components, with one being common to both products. Fong et al. [13] pursue the approach of Baker et al. [5] and provide analytical formulations for a commonality problem minimizing the expected shortage subject to a fixed budget constraint and assuming independent Erlang demand distributions. They observe that the relative reduction in the expected shortage can be substantial when the budget level is high relative to the demand requirements for the end products – even if the component is much more expensive. Note that all these models assume a single period. Additional relevant works include Nonås [28] who formulates a two-stage stochastic program for an ATO system with three products and an arbitrary number of components, and introduces a gradient-based search method to find the optimal inventory levels for a profit maximization problem.

There are also a lot of literature about the replenishment policy. Among them, (R,Q) and base stock policy are frequently used. Thonemann and Brandeau [33] and Wang [35] use (R,Q) policy while Akçay and Xu [3], Zhao [37], Lu and Song [24], Deza et al. [9] and Doğru et al. [11] are using base stock policy. In terms of allocation rules, many researchers have assumed FCFS due to its simplicity, see Akçay and Xu [3], Song and Zhao [30], Boute [7], Zhao [37], Lu and Song [24] and Deza et al. [9]. Doğru et al. [11] investigated a continuous review W system and concluded that the FCFS base stock policy is typically suboptimal by allowing revoke previous period decisions. They also provided a lower bound for the optimal objective value and developed a policy attaining the lower bound under some symmetry condition for the cost parameters and a so-called *balanced capacity* condition for the solution. van Jaarsveld and Scheller-Wolf [34] developed a heuristic algorithm for large-scale continuous review ATO systems which improves as the average newsvendor fractiles increase. They show that, for large-scale ATO systems, the best FCFS rule is nearly optimal, and proposed a no-holdback allocation rule which can outperform the best FCFS rule.

1.2 Akçay and Xu formulation

1.2.1 Assumptions

In chapters 2 and 3, we are focusing on models similar to those proposed by Akçay and Xu [3], Huang and de Kok [20], and Huang [19]. In the model, we make the following assumption:

(1) periodic review system,

(2) assembly takes zero time,

(3) replenishment lead time for each component is constant and greater than zero,

(4) independent base stock policies are used for each component,

(5) product demands are satisfied by an FCFS rule,

(6) product demands are correlated within each period, while the demands over different periods are independent,

(7) product rewards are collected if the assembly is completed within the given time window.

In addition, we assume the following sequence of events for each period:

(i) inventory position reviewed,

(ii) new replenishment order of components placed,

(iii) earlier component replenishment order arrives,

(iv) demand realized,

(v) component allocated and product assembled,

(vi) associated rewards collected.

1.2.2 Remarks

We wish to point out the following key equations:

$$I_{i,k-1} + A_{i,k} - D_{i,k} = I_{i,k}$$
$$A_{i,k} = D_{i,k-L_i-1}$$

where $I_{i,k}$ is the net inventory of component i at the end of period k and $A_{i,k}$ is the arrival replenishment order of component i in period k. The first equation models that the inventory of component i at then end of any given period is equal to the inventory of component i at the end of the previous period plus the arrival of the replenishment order of component i at that given period minus the component i that was used to assemble products at that given period. The second equation models that the arrival of the replenishment order of component i at any given period is set to the component i that we used to assemble products at previous period L_i+1. Note that $I_{i,k}$ considers the inventory at the end of the period. While the first equation is straightforward, the second equation is more involved. As the replenishment follows a base stock policy, a replenishment order is placed after reviewing our inventory to bring up the stock to the base stock level. By definition of lead time, the replenishment order placed at period k arrives at $k + L_i$. If you look at the sequence of events, the replenishment order placed at period k is for the inventory position $k-1$. Thus, in order to replenish the demand of period k, we place an order in period $k + 1$ which arrives in period $k + L_i + 1$.

1.2.3 Model description

A periodic review system only creates some checkpoints to analyze inventory for components and reduces the burden of inventory management. The base stock policy can help us create an easier to understand the model. Note that the required numbers of components base stock are the same. In each period; that is, once decided, those numbers are fixed. The requirement of zero time for product assembly allows ignoring the assembly problem. The replenishment lead time for each component being constant simplifies the model as it enforces that a placed replenishment order arrives on-time. The lead time greater than zero means we need to consider all the components since a zero lead time for a component means we have an infinite inventory of that component which is unrealistic. The FCFS allocation rule allows us to circumvent the demand satisfaction problem; we assume the demand is a random variable to be closer to the real-world problem. The objective is to maximize reward collections within a given time window. The model is based on a multi-matching approach proposed by Huang and de Kok [20] and by Huang [19] where multiple components are matched with multiple products to satisfy demands. Axsäter [4] introduced the concept of matching multiple supplies to a single demand. In each period within the time window, rewards are collected by satisfying product demands. We recall that the time window is the number of periods between the order receiving period and the order fulfillment period. In particular, a time window equal to 0 means that the demand must be fulfilled within the period the order is received; that is, we must have enough components to satisfy the demand within that period in order to collect rewards. The base stocks of the ATO system are constrained by a pre-set overall budget.

In order to illustrate the formulation, we need to give an abstract of the system. The system consists of a supply line and an assembly line. When we mention the supply line, we need to deal with components; while we talk about the assembly line, we deal with components and products. On the supply line, we consider component i: At time t, a component demand $D_{i,t} = \sum_{j=1}^{m}(a_{i,j} \times P_{j,t})$ of component i is needed in order to satisfy the product demand. For simplicity, let's consider the current period t. On the supply side, when the demand $D_{i,t}$ arrives, the available on-hand inventory, that is the inventory of the components in stock, is $(S_i - D_{i,[t-L_i,t-1]})^+$ where $D_{i,[t-L_i,t-1]} = \sum_{s=1}^{L_i} D_{i,t-s}$ and $x^+ = \max(0,x)$. The on-hand inventory formula comes from the FCFS rule, the base stock policy, and the deterministic lead times. As we have deterministic lead time, if the on-hand inventory is positive, the replenishment order placed at $t - L_i - 1$ arrives at time t and any order from time $t - L_i$ to $t - 1$ has been satisfied due to the FCFS rule and base stock policy. As time t is not a fixed time period, we can consider any period $t + k$. Thus, the available on-hand inventory in period $t + k$ is:

$$(S_i - D_{i,[t-L_i+k,t-1]})^+$$

for any $0 \leq k \leq L_i$. When $k = L_i$, the on-hand inventory becomes S_i as the replenishment order arrives. Even if the demand at time t may not be met before $t + L_i$, it will be the case at $t + L_i + 1$ as the replenishment order arrives and brings up the on-hand inventory to satisfy the demand. Placing a replenishment order with more than $\sum_{j=1}^{m}(a_{i,j} \times P_{j,t})$ demand would both violate the base stock policy and be disadvantageous. On the assembly line side, the component i matching the demand arrives at time t and is assembled in $t + 0$, $t + 1$,...,$t + L + 1$. We use L, instead of L_i, as some of the components of a given product may be unavailable even if the replenishment order of component i arrived. Thus, our planning horizon is indexed from 0 to $L+1$. The formulation is based on a two-stage decision model. The first stage determines a base stock level for each component, and the second stage one determines products that need to be assembled in each period with respect to some constraints reflecting the inventory availability. The first stage decisions are made before the second stage decisions following a two-stage stochastic programming framework, see Birge and Louveaux [6]. The objective is to maximize the expected total reward collected from the products assembled within given time windows. Note that while all products are eventually assembled within $L+1$ periods, the rewards are collected only within the pre-set time windows. This means that a product assembled after the time windows does not increase the total reward. Thus, we can restrict our decision horizon to w instead of $L+1$ and postpone any assembly after time window w in $L+1$ period to create a full set of product assembly decisions.

1.2.4 Formulation

The second stage corresponds to the allocation problem $\big(Alloc(S,\xi)\big)$ where $S=(S_i)$ is the vector representing base stock levels, $\xi=\{P_{j,k}|j=1,\ldots,m;k=0,-1,\ldots,-w\}$ is the vector representing random demands, and $O_{i,k}$ is the number of component i available at period k. Note that $O_{i,k}=(S_i-D_i^{L_i-k})^+$ for $0\le k\le w$ where $D_i^{L_i-k}=\sum_{s=0}^{L_i-k}D_{i,-s}$, and $O_{i,k}=D_{i,0}$ for $L_i+1\le k\le w$ are inferred from the base stock policy and an FCFS rule.

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{m}\sum_{k=0}^{w_j}(r_{j,k} \times x_{j,k}) && && \big(Alloc(S,\xi)\big) \\
& \sum_{k=0}^{w} x_{j,k} \le P_j && j = 1,\dots,m \\
\sum_{\mu=0}^{k}\sum_{j=1}^{m}&(a_{i,j} \times x_{j,\mu}) \le O_{i,k} && i = 1,\dots,n, \quad k = 0,\dots,w \\
& x_{j,k} \in \mathbb{Z}_+ && j = 1,\dots,m, \quad k = 0,\dots,w
\end{aligned}
$$

The first set of constraints guarantees that the product we assembe will not exceed customer demands. Base on the FCFS and base stock policy, we can conclude $\sum_{k=0}^{L+1} x_{j,k} = P_j$ as we need to assemble all the product demand within $L+1$ periods. However, as we discuss in previous section, the assembling past w doesn't contribute any reward, we can move the assembly to $L+1$. So in this case, we can use $\sum_{k=0}^{w} x_{j,k} \le P_j$ to represent and assemble the left over demand in period $L+1$. From operation research perspective, all the variables $x_{j,k}$ where $k = w+1, \dots L+1$ can be treated as a slack variable, as they are not presented in the objective function. The second set of constraints, called inventory availability constraints, guarantees that assembly can only happen when there is enough component inventory. Please note that we use the same time frame as the first set of constraints as $O_{i,k}$ is in non-decreasing order in terms of k. While an optimal allocation can be computed for a given base stock level S and demand ξ, we need to determine the optimal base stock levels. Thus, we use the two-stage stochastic integer program $\big(Joint(B)\big)$ where the first stage determines the base stock levels while the second stage maximizes the expected value of the component allocations:

$$\begin{aligned} \max \quad & \mathbf{E}[Alloc(S,\xi)] & & \big(Joint(B)\big) \\ \sum_{i=1}^{n}(c_i \times S_i) &\leq B & & \\ S_i &\in \mathbb{Z}_+ & i = 1,\ldots,n & \end{aligned}$$

After applying the sample average approximation (SAA) method, see Section 1.3, we can convert $\big(Joint(B)\big)$ into a deterministic optimization problem.

1.3 Sample average approximation method

We briefly describe how the SAA method is incorporated in the formulation, more details about this method can be found in, for example, Kleywegt et al. [23]. The objective function $\big(Joint(B)\big)$ maximizes an expected value with two inputs: the base stock S and the random product demand variables ξ. Note that in $\big(Joint(B)\big)$, the base stock levels S_i are non-negative integer variables and the sum of the cost of all base stock level is less than or equal to a given budget while the objective function is maximizing the expected value of the allocation problem for the base stock subject to the random variable ξ. Thus, applying the SAA to $\big(Joint(B)\big)$ consists of the following steps:

(i) generate M independent samples for $l = 1,\ldots,M$ with N realizations for each sample. The vector $\xi_l^N = (\xi(\omega_l^1), \xi(\omega_l^2), \ldots, \xi(\omega_l^N))$ represents the N realizations of the l-th sample,

(ii) solve the optimization problem $(INLP)$ for each sample, which is the associated deterministic version of $\big(Joint(B)\big)$ where the objective function is set to

$\frac{1}{N}\sum_{h=1}^{N} Alloc(S, \xi(\omega_l^h))$ as described below. Note that (*INLP*) is non-linear due to both the integrality constraints and the right-hand side of the inventory availability constraints. Let $\hat{S}_l$ denote the optimal base stock levels for (*INLP*) and $\hat{G}(\hat{S}_l)$ denote its optimal objective value.

(iii) generate a different sample $\xi^{N'}$ with $N' \gg N$ realizations and compare the performance among all the base stock vectors $\hat{S}_l$ solved in (*ii*) by solving $\big(Alloc(S, \xi^{N'})\big)$ with $S = \hat{S}_l$. Let $\bar{G}(\hat{S}_l)$ be the new optimal objective value.

(iv) select the optimal base stock vector $\hat{S}^*$ achieving the best performance among all the base stock vectors; that is, $\hat{S}^* = \text{argmax}\{\bar{G}(\hat{S}_l) : l = 1, \ldots, M\}$.

$$
\begin{aligned}
\max \quad & \frac{1}{N}\sum_{h=1}^{N}\sum_{j=1}^{m}\sum_{k=0}^{w_j}(r_{j,k} \times x_{j,k}^h) && && && (INLP)\\
& \sum_{k=0}^{w} x_{j,k}^h \le P_j^h && && j = 1, \ldots, m, && h = 1, \ldots, N\\
& \sum_{\mu=0}^{k}\sum_{j=1}^{m}(a_{i,j} \times x_{j,\mu}^h) \le O_{i,k}^h && i = 1, \ldots, n, && k = 0, \ldots, w, && h = 1, \ldots, N\\
& \sum_{i=1}^{n}(c_i \times S_i) \le B && && && \\
& S_i \in \mathbb{Z}_+ && && && i = 1, \ldots, n\\
& x_{j,k}^h \in \mathbb{Z}_+ && j = 1, \ldots, m, && k = 0, \ldots, w, && h = 1, \ldots, N
\end{aligned}
$$

We solve M times (*INLP*) with different sample data to obtain M candidate base stocks and, for each candidate base stock, we solve N' $\big(Alloc(S, \xi)\big)$; that is, a total of MN' $\big(Alloc(S, \xi)\big)$. Note that (*INLP*) is much harder than $\big(Alloc(S, \xi)\big)$ since it has N-times the number of decision variables. (*INLP*) cannot break into N different small subproblems and requires a large memory. On the other hand, the MN'

$\big(Alloc(S,\xi)\big)$ problems are independent of each other and thus can be solved in a parallel computational framework. Let $\hat{G}_M = \frac{1}{M}\sum_{l=1}^{M} \hat{G}(\hat{S}_l)$, $\bar{G}_{N'} = \bar{G}(\hat{S}^*)$, and G^* be the optimal objective value of $\big(Joint(B)\big)$. Since $\bar{G}_{N'} \leq G^* \leq \hat{G}_M$ under certain conditions for N, M, N', see Birge and Louveaux [6], $\bar{G}_{N'}$ and $\hat{G}_M$ are, respectively, a lower and an upper bound for G^*. For more details concerning the statistical testing of optimality for the SAA method, and the selection of N, M, and N', see Kleywegt et al. [23]. Note that since $O_{i,k} = (S_i - D_i^{L_i-k})^+$ is a piece wise function of S_i; we use the standard Big-M method to check whether $(S_i - D_i^{L_i-k})$ is positive.

Chapter 2

Impact of modifying the inventory availability constraints

This chapter focuses on the impact of the modification of the inventory availability constraints in Akçay and Xu [3]. The inventory availability constraints are used to ensure that each component allocation is at most the available inventory. While some models allow allocation larger than the available components, we disregard this case to simplify the model and exposition. Allocating a non-existing component to the assembly line should be avoided, and keeping the available inventory constraints makes our solution executable; that is, the assembly line can run according to our solution without waiting for non-existing components arrival. However, Akçay and Xu [3] modified those constraints arguing computationally gains. We will discuss some difficulties arising in sample generation in the modified system and investigate the impact on the SAA method. Afterward, we show the computational results for the Zhang system and the IBM system to illustrate the impact of the modification.

The inventory availability constraints is represented by a plus sign in the (*INLP*) formulation. In the computational experiments performed by Akçay [2], $(S_i - D_i^{L_i-k})^+$ is substituted by $(S_i - D_i^{L_i-k})$, and they call this substitution a modification of the inventory availability constraints. The obtained new formulation (*ILP*) allows faster computations. Note that the feasible region of (*ILP*) is a subset of the feasible region of (*INLP*). In addition, while relaxing the integrality constraints on the variables would make (*ILP*) become a smooth function, (*INLP*) would remain non-smooth due to the $(S_i - D_i^{L_i-k})^+$ term on the right hand side of the inventory availability constraints. Note that substituting $(S_i - D_i^{L_i-k})$ for $(S_i - D_i^{L_i-k})^+$ may lead to infeasibility. This issue can be addressed by filtering out samples leading to infeasibility and by assuming sufficiently large budget level; that is, by allowing large base stock levels. We argue that substituting $(S_i - D_i^{L_i-k})$ for $(S_i - D_i^{L_i-k})^+$ might yield an intractable sample generation process for the SAA approach for low budget levels.

2.1 Impact of modifying the inventory availability constraints

The substitution shrinks the feasible region of (*ILP*) to a subset of the feasible region of (*INLP*). While for $(S_i - D_i^{L_i-k}) \geq 0$ there is no difference between (*ILP*) and (*INLP*), the modification may lead to infeasibility if $(S_i - D_i^{L_i-k}) < 0$ as the left hand side of the contraints $\sum_{\mu=0}^{k} \sum_{j=1}^{n} a_{ij} x_{j\mu}$ is non-negative. The original model (*INLP*) with a plus sign ensures there is always a feasible solution, i.e. a solution setting all the base stocks and assembled products to zero. Since we use SAA method to find a base stock, the infeasibility is a challenge as we may encounter realizations yielding samples which are infeasible for the given budget. For any given budget, it is always possible to find a sample making the modified model infeasible and, as the samples in our calculation are generated randomly from the distributions, we can not guarantee that the SAA method is applicable. On the other hand, we can determine the probability of a sample to be feasible for a given distribution and a given budget. We filter out the samples leading to infeasible solutions and assume our budget is sufficiently large. For instance, we can use $\sum_{i=1}^{n} a_{i,j} P_j (L_i + 1) c_i$ as a sufficiently large budget where the mean value of product j represents P_j. Note that as we modify the samples, their properties are hard to track which complicates the analysis. The mean and variance of generated samples are impacted, and thus both the distribution of the samples and the SAA method are impacted

The challenge we are facing for low to medium budgets is to generate a sufficient number of feasible samples for the SAA method. Note that infeasible sample can be easily created manually for the modified model (*ILP*), and under the extreme case, when the budget is set to zero, the only feasible sample is the trivial zero sample. Thus, we need to bound the number of in sample generation trials to warrant the process is finite. Consequently, we generate samples for (*ILP*) until either the required number of samples, or a pre-set number of feasibility tests, is reached. This may lead to an undecidable SAA as we may not have enough samples. The key idea behind this is based on comparing with a computed minimum budget of a sample that can lead to a feasible solution. The computed minimum budget is determined from the (*ILP*) minimum base stock levels using the algorithm described below. It is based on the fact that non-negativity of the left hand side of the inventory availability constraints implies $(S_i - D_i^{L_i - k}) \geq 0$ for all components. Once we know the minimum base stock of each component to meet the constraint, i.e., the maximum of D_i^0 among all the realizations, we can multiply the correspond cost of each component and sum them up to get the minimum budget.

The following algorithm computes the minimum feasible budget of a given sample.

The input of the algorithm is a generated sample. Then, based on the number of the realizations and lead time, we can slice a sample into realizations. For each realization, the maximum number of components requirement can only appear at time $t + 0$ as the replenishment order will arrive after and consequently reduce the number of requirement. So for each component i, we compare D_i^0, the number of demand at time $t+0$, with the maximum base stock, $\max S(i)$. If it is greater than $\max S(i)$, we assign it as $\max S(i)$. At the beginning, we set all the $\max S(i)$ to 0, and after the iteration, comparing among all the realizations, the maximum demand of each component is known. Then, we just calculate the cost, sum up all the numbers, and the obtained figure is the minimum feasible budget and $\max S(i)$ contains the minimum base stock of each component i. This holds since in the formulation of (ILP) the left hand side of the inventory availability constraints is non-negative and, to satisfy those constraints, we need to make sure that the right hand side is non-negative and that all the constraints are satisfied. In other words, if we know the maximum number of demand of component i among all the realizations, we can determine the minimum base stock for component i.

Algorithm 2.1 Computing minimum feasible budget

```
Initialize maxS ← zeros(n)
for any realization h do
    for for any component i do
        if D_i^{L_i} > maxS(i) then
            maxS(i) ← D_i^0
        end if
    end for
end for
B = Σ_{i=1}^{n} c_i × maxS(i)
```

The modification may yield another challenge for the SAA method. As mentioned earlier, the properties of the sample distribution are impacted and thus the SAA method is impacted too. Namely, the optimal objective value may fluctuate; note in particular that the algorithm does not depend on the current period demand. Moreover, one of the properties used in SAA method no longer hold. Using notations similar to the ones used in Section 1.3, let $\bar{G}^{\bullet}_{N'}$, $G^{*}_{\bullet}$, and $\hat{G}^{\bullet}_{M}$ denote respectively the (*ILP*) lower bound, optimal value, and upper bound. The discussions in Section 1.3 imply that $\bar{G^{\bullet}_{N'}} \leq G^{*}_{\bullet} \leq \hat{G^{\bullet}_{N}}$. Akçay[3] states that an upper bound of (*ILP*) is a valid upper bound for (*INLP*); that is $G^{*} \leq \hat{G^{\bullet}_{N}}$, by assuming (*ILP*) is a relaxation of (*INLP*). On the other hand, we argue that $\hat{G^{\bullet}_{N}}$ may not be an upper bound for (*INLP*) as (*INLP*) is a relaxation of (*ILP*). As $x \leq M$ implies that $x \leq M^{+}$, any feasible solution of (*ILP*) is feasible for (*INLP*), while the reverse is false. For example, a zero base stock is a feasible solution for (*INLP*) but infeasible for (*ILP*) if the demands are non-zero; that is, $\hat{G^{\bullet}_{N}} \leq \hat{G_{N}}$ may hold.

2.2 Algorithmic implementation

This section focuses on the implementation of sample generation and the filtering out process. We have included a tailored AtoNorInfo.m file in C.3. The AtoNorInfo is a class used to generate an ATO system and receives from AtoInfo the necessary input such as lead time; in other words, AtoNorInfo is used as a class factory. If no ATO specification is given, we use the one in Akçay and Xu [3]. The key t function is genreal(obj,nReal) which generates the set of realizations. We used the built-in function randn which returns a set of pseudorandom values drawn from the standard normal distribution. In particular, line 23 generates values from a normal distribution with mean exDemProdsArr and standard deviation sdDemProdsArr. Note that since the values may contain negative instances, we need to regenerate another set of normal distributions with mean exDemProdsArr and standard deviation sdDemProdsArr to replace the negative instances. At line 24, $dem < 0$ returns a matrix of a binary value corresponding to dem, as true is treated as 1 and false is treated as 0 in Matlab. The maximum number of times negative instances are regenerated is set to 10, and set the result to 0 after 10 attempts. The results are rounded to the closest integer.

The calculation of the minimum budget for (ILP) is done by the getMinBudg function. We first generate the number of realizations in the given input data, then we compute or each realization the D_i^k for each component. We only take time $t+0$ as the max D_i^k can only be on $t+0$ since a base stock of that given component satisfying time $t+0$ will satisfy all future periods as further replenishment order will arrive. Once we get the all the realization data, we compute the maximum component requirements among all realizations. One has to be careful that the max should be applied to the same component in different realizations. By multiplying the max components and cost of components, we can determine the minimum budget satisfying the demand. The format of the data is $[t-3, t-2, t-1...$ and thus in computDik function the matrix is flipped, and we end up with $end - 1$ as the last item is $t+0$. Then, we compute the cumulative sum of the matrix up to L_i since anything greater than L_i can be ignored as further replenishment orders will arrive. Getting enough samples to properly compare the the two models for small budgets can be challenging. In particular, applying the algorithm directly to generate enough samples might be too time consuming. Thus, a recursive approach is used: We first apply the algorithm to sample with one realization, then after filtering, we use the filtered samples to generate samples with two realizations, then using the new filtered samples to generate samples with five realizations, and so forth by increment of 5 and 15 for the Zhang system, and 5 ,10 for the IBM system. In addition, this recursive approach allows to understand how the number of realization may affect the sample generation process. Since the size of table is too large to list all the mean value of the samples, so we only include the mean of the samples after the recursion used for computation in the appendix.

2.3 Computational results

2.3.1 Computational results for the Zhang system

We consider an ATO system proposed in 1997 by Zhang [36] with four products and five components as described in Table 2.1. We present our computational results in the figures where *LB* and *UB* denote, respectively, the lower and upper bounds for the (*ILP*) and (*INLP*) formulations. The detailed numbers are available in Tables A.1, A.2, A.3 and A.4 where *N/A* corresponds to budgets for which not enough sample yielding a feasible formulation can be generated. Note that since we can not generate enough samples for budget level lower than 75, so we skip the *N/A* for (*ILP*) when budget level is less than $7,500$. Also, the numbers in the header mean the budget we use to filter out sample, so 75 means we use the samples that are filtered with a budget of $7,500$. The parameters for the SAA method are set to: $N = 25, N' = 100$ and $M = 5000$. If a million samples are not enough to yield 100 feasible (*INLP*) samples, the process stops and outputs *N/A*. For (*INLP*), we first use the algorithm 2.1 to filter out the samples with different budgets. Then, the obtained filtered samples are used to run the joint formulation without plus sign. After determining the optimal base stock levels for a given set of samples, we obtain the maximum base stock for each component among different optimal base stock levels. The maximum base stock of each component is used to filter out samples for allocation formulation without plus sign. As the allocation formulation is significantly easier to solve, one could use the allocation formulation with plus sign for (*INLP*) for the allocation process. However, we use the formulation without plus version in order ro analyse its behaviour.

The impact of the filtering process is illustrated in Figures 2.1 and 2.2 where, for instance, *Filter at 75* means that the samples are filtered with a budget of $7,500$. The two algorithms are compared in Figures 2.3 and 2.4 using the same filtered sample, and the metric used is the percentage of Type II service levels difference. The combined impact is analyzed in Figure 2.5; we compare (*ILP*) with different filtered samples to (*INLP*) with unfiltered sample.

						Component				
					i	1	2	3	4	5
					c_i	2	3	6	4	1
		Product			L_i	3	1	2	4	4
j	*Mean*	*StdDev*	r_j	w_j		Bill of Materials				
1	100	25	1	0		1	2	1	0	0
2	150	30	1	0		1	1	1	0	0
3	50	15	1	0		0	1	1	1	0
4	30	11	1	0		0	0	0	1	1

Table 2.1: Settings for the Zhang system

Figures 2.1 and 2.2 illustrate the behaviour of the filtered sample if we use the original formulation. When the threshold increases, the degree of alteration of the sample decreases and so does the Type-II service level. The rational explaining why the modified samples have a higher service level is the fact that the mean of the past demands significantly decrease. When the past demands decrease, the required budget for the base stock decreases as well and the leftover budget can be used to satisfy current period demand. The increase of the Type-II service level only creates a bias as it does not provide an accurate estimation of the demand. In Figures 2.3 and 2.4, the same sample is used with different formulations, one with plus sign, the other without plus sign. The gap between the two formulations is relatively small. e.g. 3.5% using samples filtered at $7,500$. Concerning the difference between the two methods, the sample filtering process contributes more to the difference than the formulations even if the gap between the two formulations is still noticeable. The combined impact is analyzed in Figure 2.5, i.e. $(INLP)$ with the unfiltered sample, (ILP) with the samples filtered at $7,500$ and $8,000$. There is an issue with (ILP) samples for allocation since the distribution is quite different for joint and allocation formulation sample. Thus, the allocation's service level is higher than for the joint one. The mean value of the data, included in the appendix, explains this behaviour.

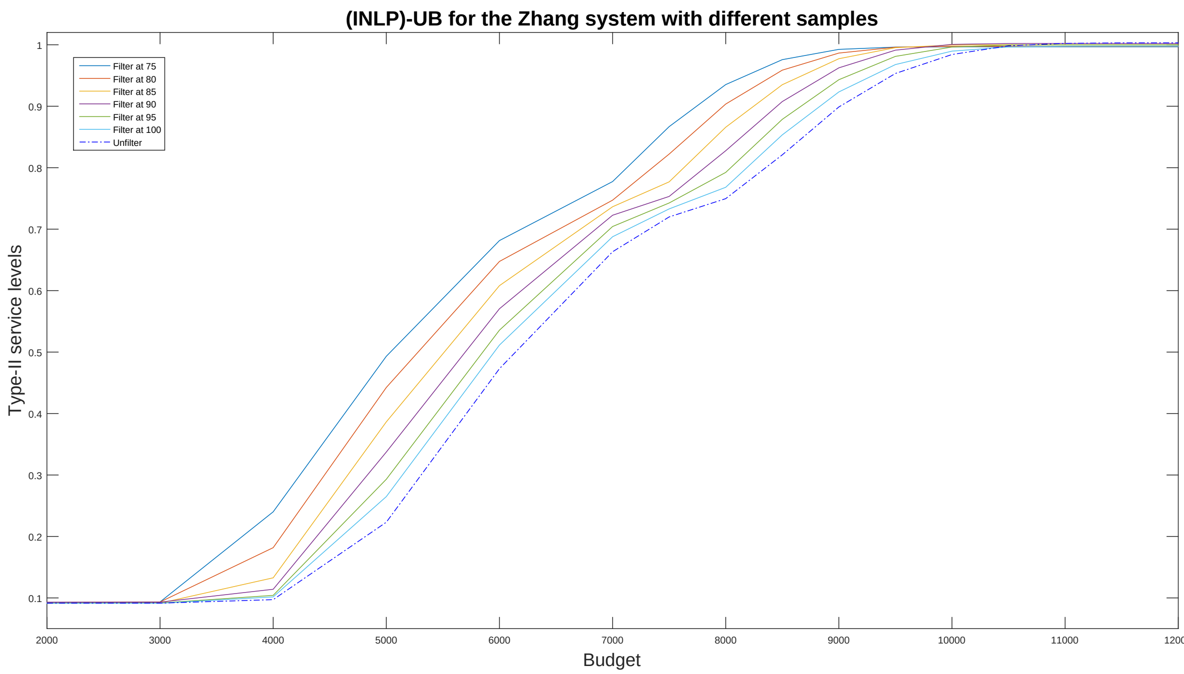

Figure 2.1: Type II service levels for $(INLP)$-UB for the Zhang system using different samples

As the time window is 0 in the Zhang system, any demand that can not be filled immediately will not provide any reward. This may appear as a strict requirement but might be close to real world constraints as one may loose customers if the time to fulfill an order is long. From budget level from two to seven thousand, which is a low to medium budget level, (ILP) can not generate enough samples for computation.

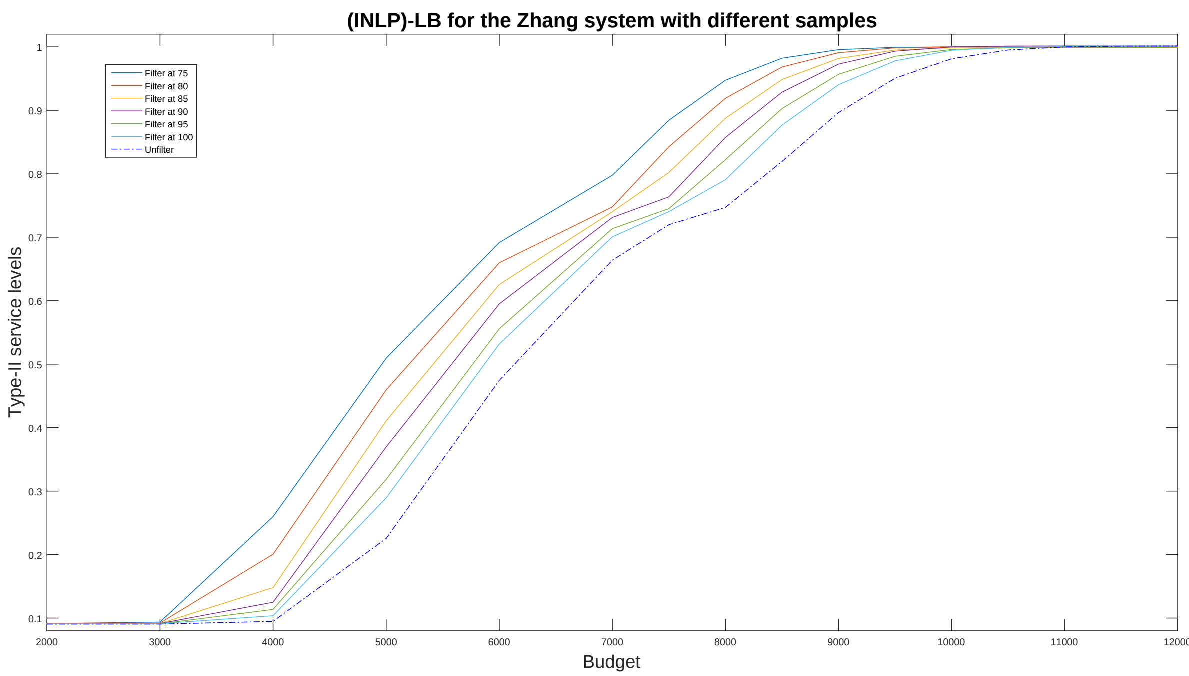

Figure 2.2: Type II service levels for $(INLP)$-LB for the Zhang system using different samples

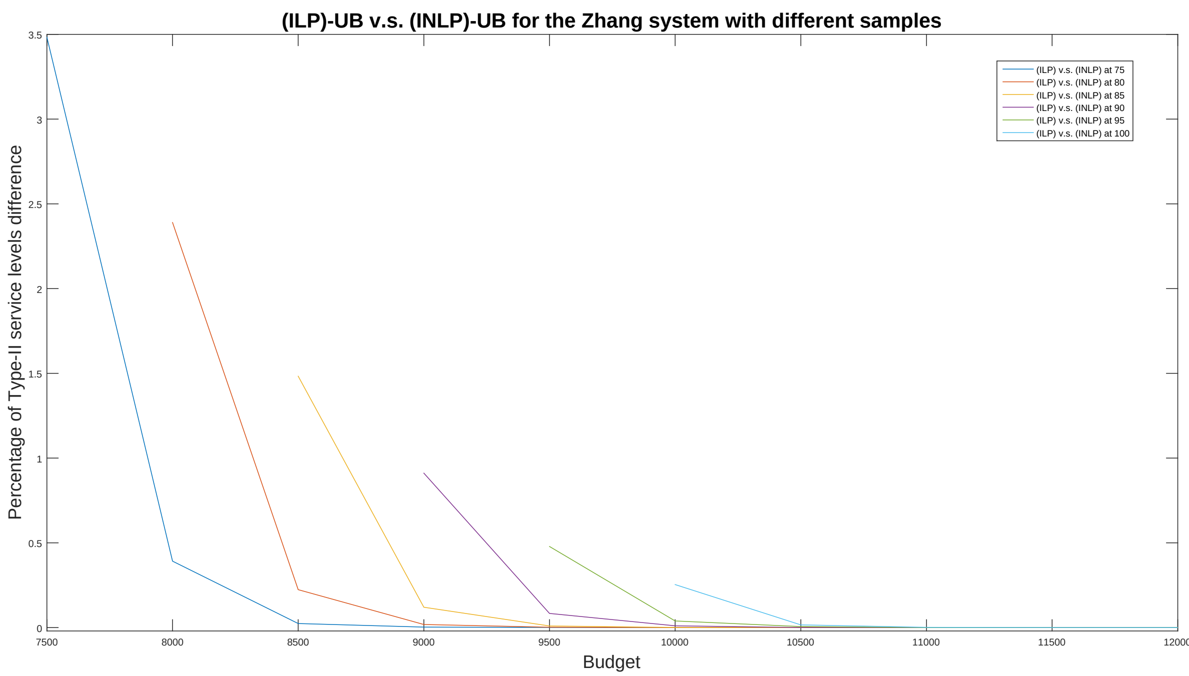

Figure 2.3: Type II service levels difference for (*INLP*)-UB v.s. (*ILP*)-UB for the Zhang system using different samples

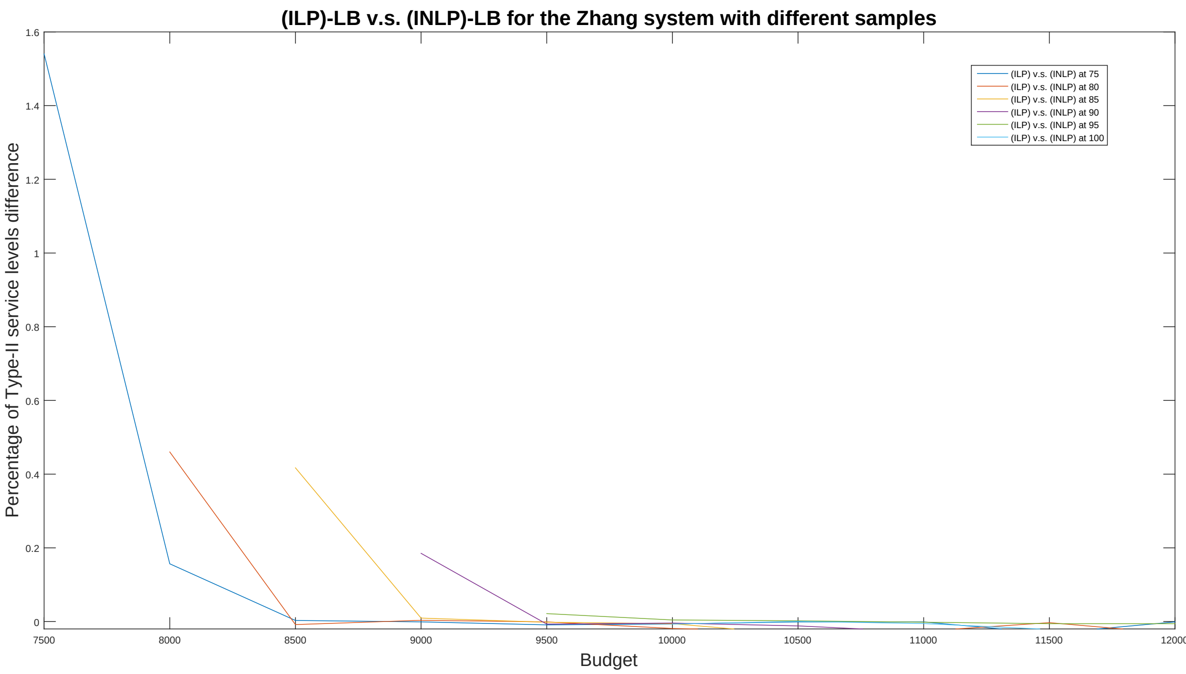

Figure 2.4: Type II service levels difference for $(INLP)$-LB v.s. (ILP)-LB for the Zhang system using different samples

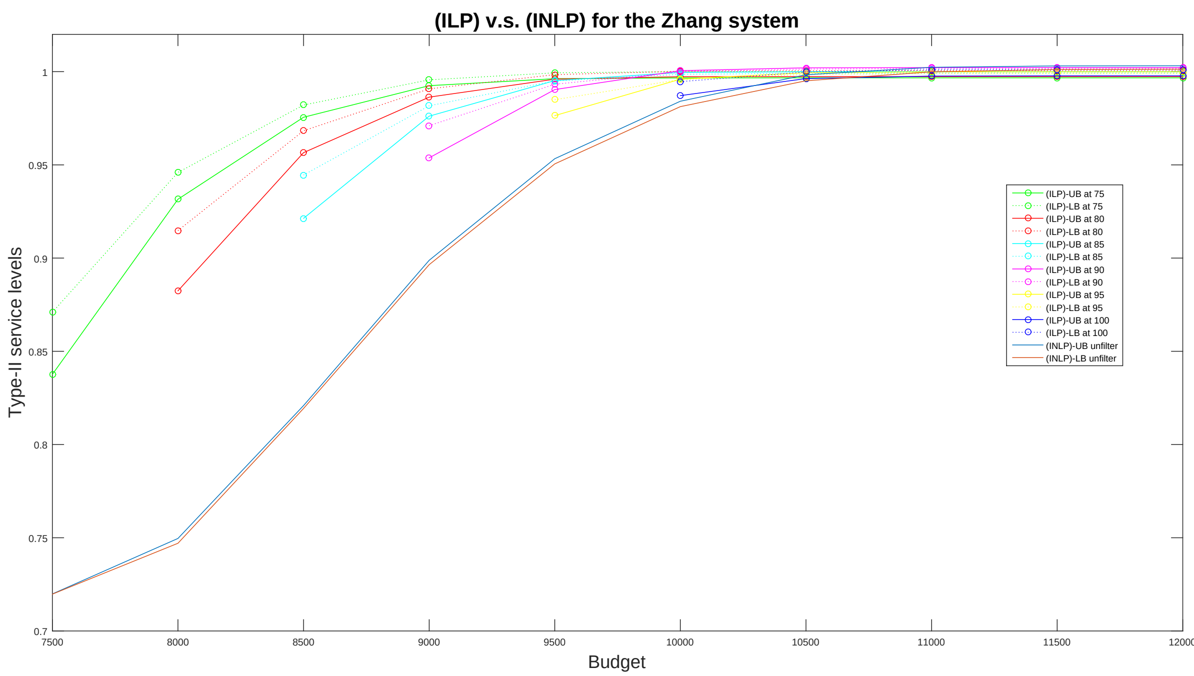

Figure 2.5: Type II service levels for $(INLP)$ v.s. (ILP) for the Zhang system combined effect

The service level of (*INLP*) is quite low but has its real world meaning, especially for emerging company or start-up, and it also applies to some products. One of the major application is IT industry where instances of low service level occur under the so-called hunger marketing or guerrilla marketing strategies. Those strategies are driven not only by targeting more sales but also by restricted capital cost, i.e. limited budget. When we consider the example of a new iPhone released by Apple, the service level is actually low as many people are waiting for a significant time for their orders to be filled given that the time window is quite small. Similarly, a company like Xiaomi grew exponentially over the past few years by reducing significantly the required capital. Though Xiaomi's service level is quite low, resulting in many complaints from their customers, they could cut their price to half the one of their competitors with similar functionality. Reducing the service level can be strategically preferred in some situations and may even be unavoidable.

2.3.2 Computational results for the IBM system

To illustrate our approach, we use a larger ATO system proposed in 2002 by Cheng et al. [8] with six products and seventeen components as described in Table 2.2. The computational results are presented in Figure 2.6 to Figure 2.10 where *LB* and *UB* denote, respectively, the lower and upper bounds for the (*ILP*) and (*INLP*) formulations. The detailed numbers are available in Table B.20, B.21, B.22 and B.23 where *N/A* corresponds to budgets for which not enough sample yielding a feasible formulation can be generated. The parameters for the SAA method are set to: $N = 20, N' = 100$ and $M = 5000$. If a million samples are not enough to yield 100 feasible (*INLP*) samples, the process stops and outputs *N/A*.

The differences are less signifiant in Figures 2.6 and 2.7 as the number data points

			Component						
			j	1	2	3	4	5	6
			c_i	1363	1595	1765	1494	1494	1628
i	c_i	L_i	Bill of Materials						
1	42	5		1	1	1	1	1	1
2	114	5		1	1	1	1	1	1
3	114	5		1	1	1	1	1	1
4	307	5		1	0	0	0	0	0
5	538	5		0	1	0	0	0	0
6	395	5		0	0	1	1	1	0
7	790	5		0	0	0	0	0	1
8	290	5		1	1	1	1	1	1
9	155	5		1	1	0	0	0	1
10	198	5		0	0	1	1	1	0
11	114	5		1	1	1	1	1	1
12	114	5		1	1	1	0	1	0
13	114	5		0	0	0	1	0	0
14	43	5		0	0	1	0	0	0
15	114	5		0	0	1	0	0	0
16	114	5		1	1	1	1	1	0
17	114	5		0	0	1	0	0	0

Table 2.2: Settings for the IBM system

is larger. A larger gap can be seen in the data given in the Appendix. As we increase the filter threshold significantly, say we filter at 115, the gap is close to 0. Figure 2.8 shows that the gap between the two formulations do down rapidly from 350% to 25% when we increase the budget by only half million. In addition, Figure 2.9 provides a negative gap; that is, (ILP) outperforms $(INLP)$. In Figure 2.10, the bias does appear and we can not warrantee whether it is positive or negative; that is we can not easily solve the problem of the modified method.

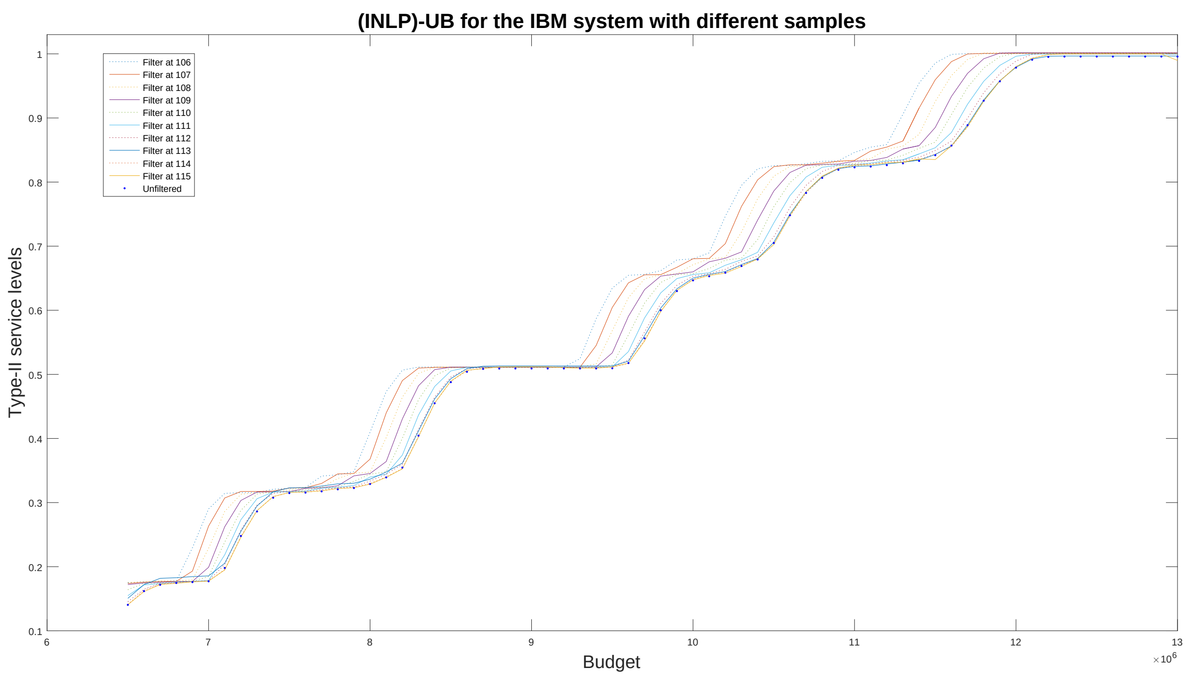

Figure 2.6: Type II service levels for $(INLP)$-UB for the IBM system with different samples

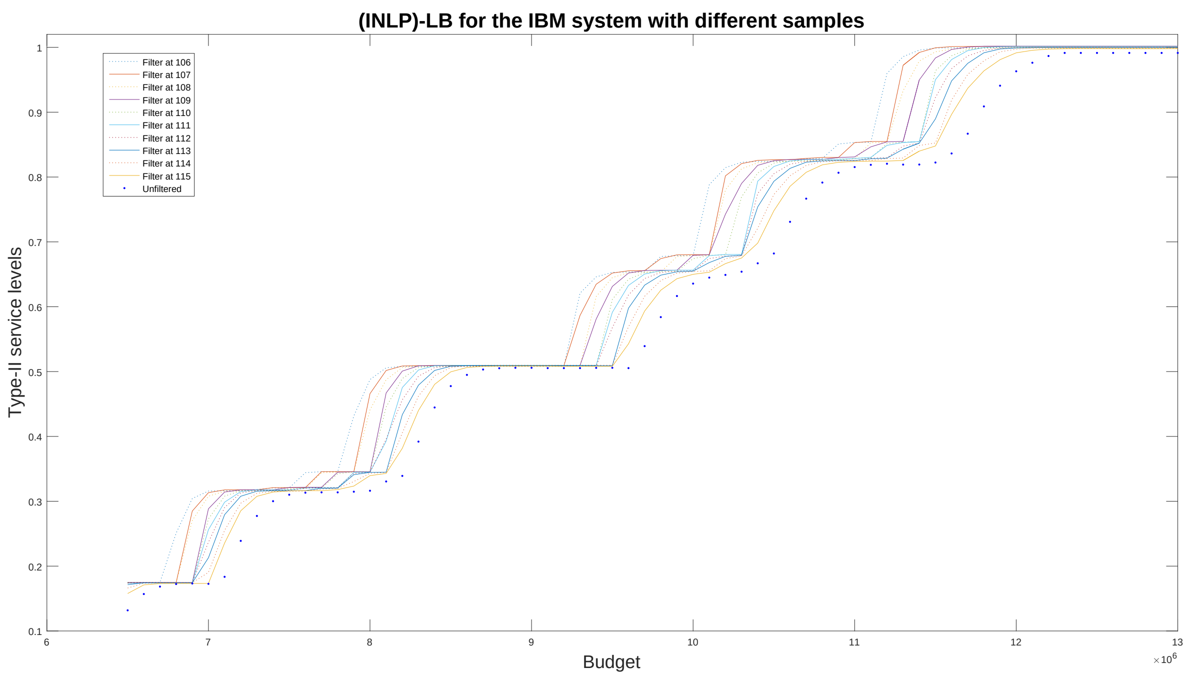

Figure 2.7: Type II service levels for (*INLP*)-LB for the IBM system with different samples

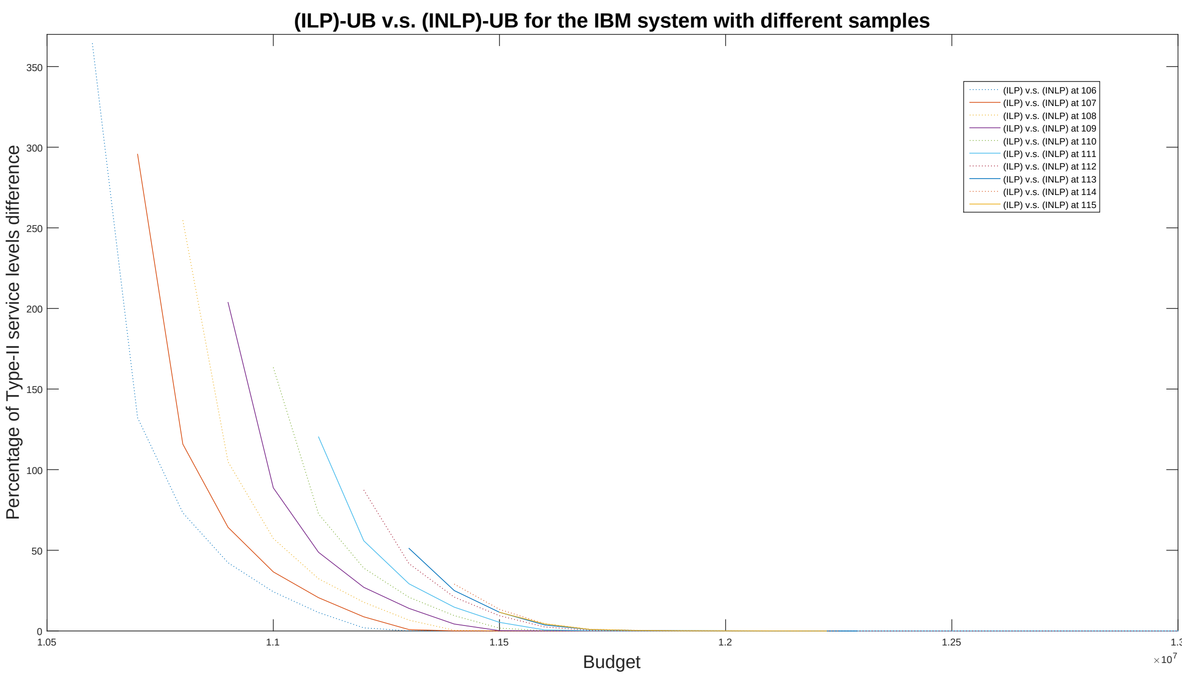

Figure 2.8: Type II service levels difference for (*INLP*)-UB v.s. (*ILP*)-UB for the IBM system with different samples

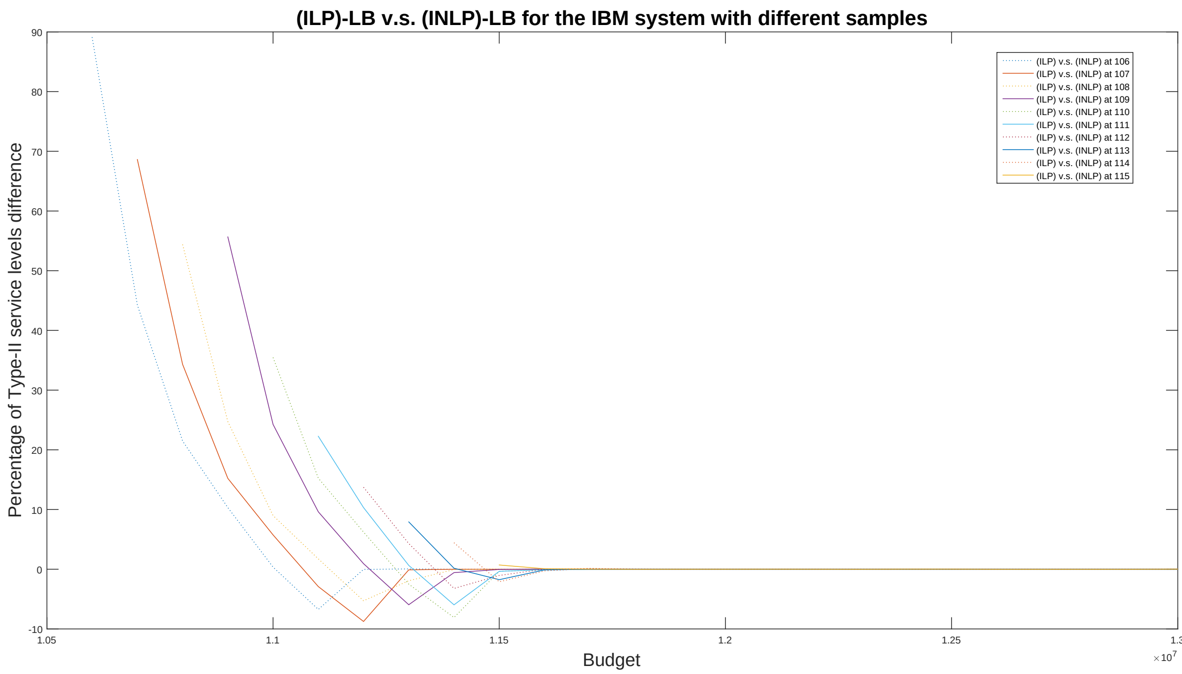

Figure 2.9: Type II service levels difference for (*INLP*)-LB v.s. (*ILP*)-LB for the IBM system with different samples

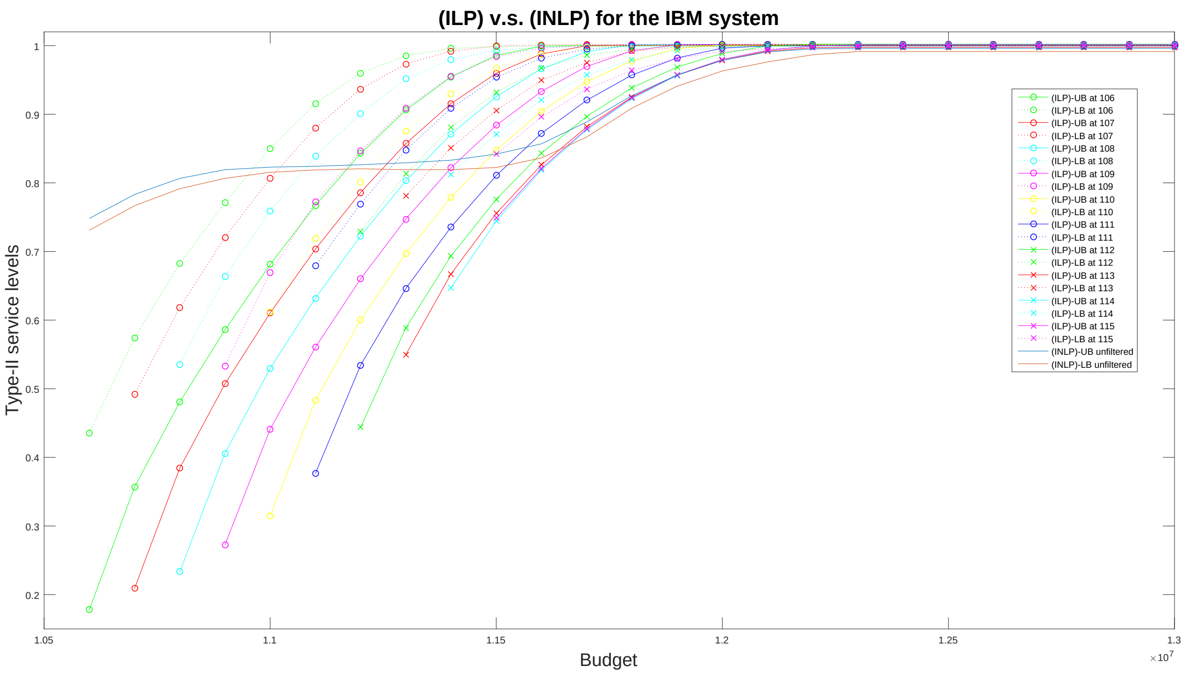

Figure 2.10: Type II service levels for $(INLP)$ v.s. (ILP) for the IBM system combined effect

Chapter 3

Component commonality for specific ATO systems

This chapter presents our observations based on the computational results for the Zhang system and a resulting strategy to improve the optimal value for small budget levels. Namely, the base stock computational results lead to a minor alteration for the Zhang and the IBM systems. We give a proof for Λ-system, i.e. a system with one component two products, to illustrate the key concept supporting the proposed alteration.

3.1 Component commonality for the Zhang system

The computational experiments performed for the Zhang system with $(INLP)$ formulation show that, for some small budget levels, the optimal base stock levels of some components are set to zero, see Table 3.1. This feature indicates a bias against component commonality and suggests that allocating the components to different products independently of each other may yield a higher objective value. For example, for a budget of 2000, the inventory levels for C_1, C_2 and C_3 are set to zero, implying that an optimal solution only considers assembling product 4. Similarly, for a budget between 5000 and 8000, the optimal base stock levels for components C_4 and C_5 are set to zero, and thus products 3 and 4 are ignored. Note that while all products are eventually assembled within $L+1$ periods, the rewards are collected only within the pre-set time windows.

Budget	C_1	C_2	C_3	C_4	C_5	**UB**	**LB**
2000	0	0	0	398	196	9.15	9.05
3000	0	393	0	406	196	9.15	9.05
4000	0	328	252	333	172	9.72	9.48
5000	616	492	382	0	0	22.32	22.55
6000	702	598	467	0	0	47.30	47.44
7000	782	724	544	0	0	66.32	66.38
7500	822	786	583	0	0	71.99	71.97
8000	862	848	622	0	0	74.96	74.70
8500	757	721	561	327	149	82.09	81.94
9000	783	777	596	344	151	89.88	89.65
9500	821	834	630	354	160	95.33	95.05
10000	848	886	662	377	166	98.41	98.12
10500	886	933	700	391	165	99.83	99.51
11000	926	968	739	408	177	100.23	99.99
11500	954	1042	776	408	177	100.31	100.12
12000	977	1088	799	408	177	100.32	100.13

Table 3.1: Optimal base stock levels and Type-II service levels for the Zhang system given different budgets

We propose a model separating component inventories with respect to the different products; that is, each product is served by dedicated components. We consider a modified bill of material (BOM) for the Zhang system as described in Table 3.2. In the first row, the subscript is the component index in the original BOM, and the superscript is the index of the product served by the component. The components with the same subscript have the same cost and lead time. Computational experiments, presented in Table A.5, are performed to compare the Zhang system with maximum component commonality (original BOM), denoted as $(INLP)$, and the Zhang system with no component commonality, denoted as $(INLP_{\Delta})$. Table A.5 indicates that the $(INLP_{\Delta})$ model outperforms the original $(INLP)$ model for a budget no greater than 8500. While the gap decreases with the increase of the budget, it is still significant for a low to medium budget. Intuitively, the situation reflects the idiom "don't put all your eggs in one basket". Putting all eggs in one basket; that is, trying to serve all products having shared components, may increase the risk of losing all eggs; that is, losing the rewards associated with serving those products together. Namely, two or more products using one or more shared component are tied together. Since the model uses a first come first served policy, we need to satisfy all the demands that are ahead of time t in order to gain reward for a product for the current period, i.e. at time t. If the non-shared components of those products have non-zero base stocks, they will claw back each other to satisfy all the non-reward demands that are before time t which increases the risk of insufficient stocks to meet the reward-generated demand, i.e. demand of time t. The rationale behind the proposed alteration is based on possible insufficient inventories to meet the high reward demand generated at time t, while serving one is possible. In other words, our results highlight that tweaking the dominated of the FCFS policy can be beneficial.

	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_3^2	C_2^3	C_3^3	C_4^3	C_4^4	C_5^4
P_1	1	2	1	0	0	0	0	0	0	0	0
P_2	0	0	0	1	1	1	0	0	0	0	0
P_3	0	0	0	0	0	0	1	1	1	0	0
P_4	0	0	0	0	0	0	0	0	0	1	1

Table 3.2: Bill of materials for the Zhang system without component commonality

The computational results are given in Figure 3.1 where LB and UB denote, respectively, the lower and upper bounds for the $(INLP)$ and $(INLP_{\Delta})$ formulations. The detailed numbers are available in Table A.5.

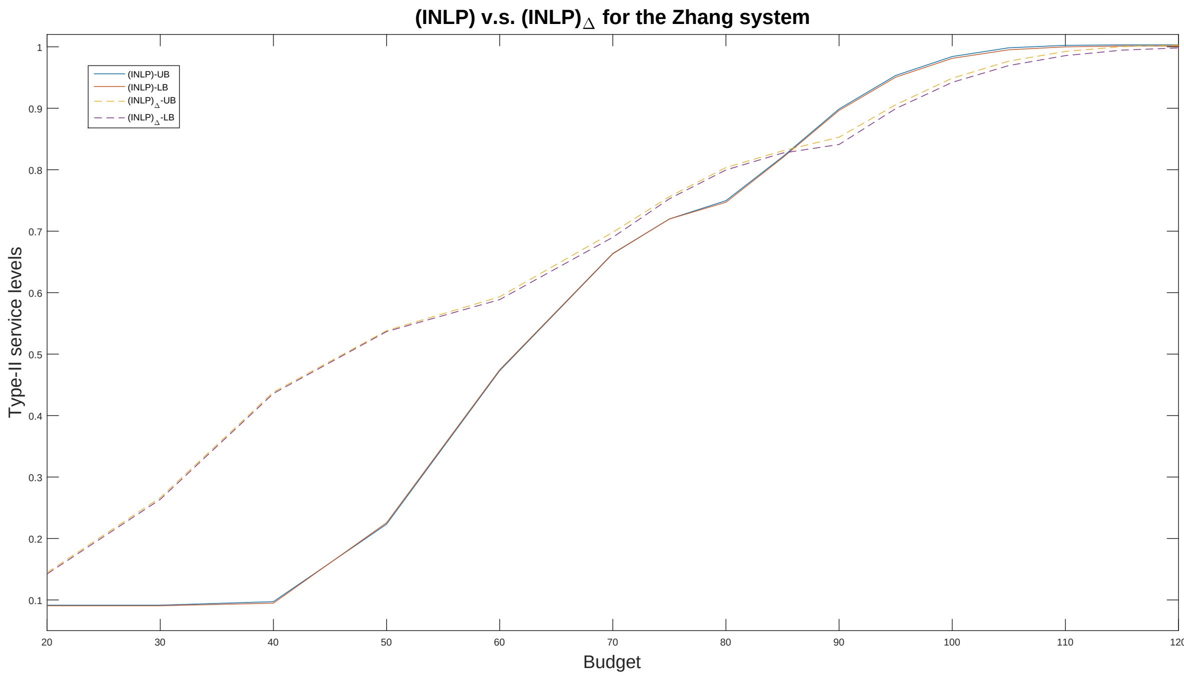

Figure 3.1: Type II service levels for $(INLP)$ and $(INLP_{\Delta})$ for the Zhang system

3.2 Component commonality for the IBM system

As in Section 2.3.2, we compare our formulation with the original one for the IBM system. See Figure 3.2 where LB and UB denote, respectively, the lower and upper bounds for the $(INLP)$ and $(INLP_{\Delta})$ formulations. The detailed numbers are available in Table B.24. Our formulation is at least as good as the original one for a service level around 83%, and a budget up to around 11.4 million. For a budget above 11.4 million, using a common component is beneficial.

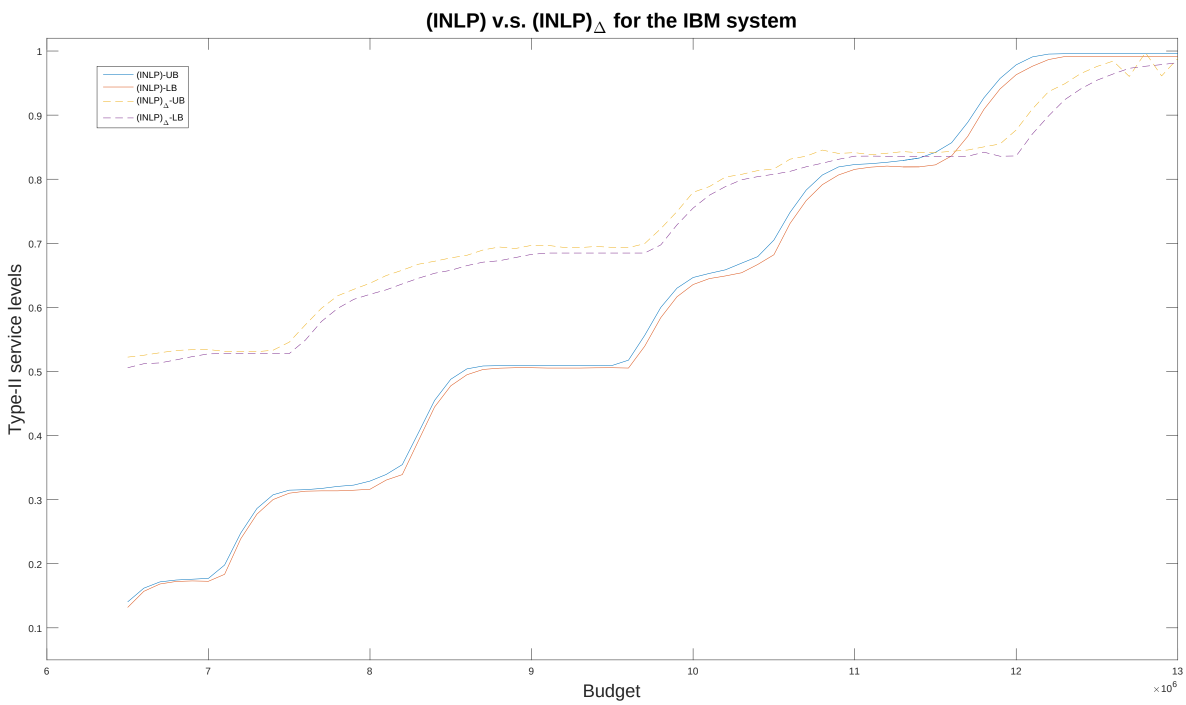

Figure 3.2: Type II service levels for ($INLP$) and ($INLP_\Delta$) for the IBM system

3.3 Component commonality for Λ-system

While the gap between the $(INLP_\Delta)$ and $(INLP)$ models is substantiated computationally in Section 3.1 and Section 3.2, we can provide a theoretical analysis for a simpler system, denoted Λ-system, consisting of one component shared by two products. This system is widely considered in the literature, see Mirchandani and Mishra [25], Eynan and Rosenblatt [12], Song and Zhao [30] and Baker et al. [5]. The original Λ-system and our modified model, denoted Λ_Δ-system and removing component commonality, are presented in Table 3.3.

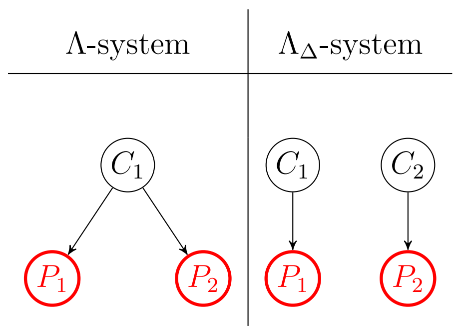

Table 3.3: Bill of materials for Λ- and Λ_Δ-systems

	$Mean$	$StdDev$	$r_{j,0}$
P_1	100	25	1
P_2	150	30	1

Table 3.4: Parameters for Λ- and Λ_Δ-systems

To simplify the analysis, the component costs and product rewards are all set to 1 and product time windows are set to 0. The corresponding SAA formulations $(INLP^N)$

and ($INLP^N_\Delta$) are as follows:

$$\begin{aligned}
\max \quad & \frac{1}{N}\sum_{h=1}^{N}(x_1^h + x_2^h) && && \big(INLP^N\big)\\
x_1^h + x_2^h &\leq (B - D_1^h - D_2^h)^+ && h = 1,\ldots,N\\
x_1^h \leq P_1^h, \quad x_2^h &\leq P_2^h && h = 1,\ldots,N\\
x_1^h, x_2^h &\in \mathbb{Z}_+ && h = 1,\ldots,N
\end{aligned}$$

$$\begin{aligned}
\max \quad & \frac{1}{N}\sum_{h=1}^{N}(x_1^h + x_2^h) && && \big(INLP^N_\Delta\big)\\
x_1^h &\leq (B_1 - D_1^h)^+ && h = 1,\ldots,N\\
x_2^h &\leq (B_2 - D_2^h)^+ && h = 1,\ldots,N\\
x_1^h \leq P_1^h, \quad x_2^h &\leq P_2^h && h = 1,\ldots,N\\
B_1 + B_2 &= B\\
x_1^h, x_2^h &\in \mathbb{Z}_+ && h = 1,\ldots,N\\
B_1, B_2 &\in \mathbb{R}_+.
\end{aligned}$$

Theorem 3.3.1 characterizes the budget ranges such that component commonality is beneficial for the Λ-system over the Λ_Δ-system. In particular, the $<$ sign in Table 3.5 means that common commonality is not beneficial for a budget ranging from B_{min} to B^+_{min} as specified in Theorem 3.3.1. The proof of Theorem 3.3.1 is given in Section 3.4.

Theorem 3.3.1. *Given a budget B, let $f^*(B)$ and $f^*_\Delta(B)$ denote, respectively, the optimal objective values of ($INLP^N$) and ($INLP^N_\Delta$). The sign of $f^*(B)$ -$f^*_\Delta(B)$ is* *given in Table 3.5 where*

$$
\begin{aligned}
B_{min} &= \min_{i=1}^{2}\{\min_{h=1}^{N}\{D_i^h\}\},\\
B_{min}^{+} &= \min_{h=1}^{N}\{D_1^h + D_2^h\},\\
B_{max}^{+} &= \max_{h=1}^{N}\{D_1^h + D_2^h\},\\
B_{max}^{\Sigma} &= \sum_{i=1}^{2} \max_{h=1}^{N}\{D_i^h + P_i^h\}.
\end{aligned}
$$

N \ B	$[0, B_{min}]$	$(B_{min}, B_{min}^{+}]$	$(B_{min}^{+}, B_{max}^{+}]$	$(B_{max}^{+}, B_{max}^{\Sigma}]$	$(B_{max}^{\Sigma}, +\infty)$
1	$=$	$<$	$<$	$\leq$	$=$
2	$=$	$<$	$\leq$	$\leq$ or $>$	$=$
N_0	$=$	$<$	$\leq$ or $>$	$\leq$ or $>$	$=$

Table 3.5: Component commonality benefits and disadvantages

The key idea behind Theorem 3.3.1 arises from the non-smooth of the formulation resulting from the right hand side of the inventory availability constraints, i.e., $O_{i,k} = (S_i - D_i^{L_i - k})^{+}$. If the budget level is sufficiently high, the inventory availability constraints become linear constraints, and, thus the non-smooth function disappeared. Using a management science formulation, if the base stock is large enough to meet the demand, the inventory availability become less important and the number of back orders decreases. If the budget level, which can de derived from the sample, can meet the demand, the base stock is not an issue.

3.4 Proof of Theorem 3.3.1

3.4.1 Case $N = 1$

Let's consider separately the case $N = 1$; that is for one realization in the SAA method. The associated formulations $(INLP^1)$ and $(INLP^1_\Delta)$ correspond to a deterministic demand where P_1^1 and P_2^1 represent the demands in the current period for, respectively, product 1 and 2, and D_1^1 and D_2^1 represent the overall demands from all previous periods. The budget level B is given and since the cost of the component is set to one, the budget level is equivalent to the sum of the base stocks.

$$\begin{aligned} \max \qquad & x_1^1 + x_2^1 \\ x_1^1 + x_2^1 &\leq (B - D_1^1 - D_2^1)^+ \\ x_1^1 \leq P_1^1, &\quad x_2^1 \leq P_2^1 \\ x_1^1, x_2^1 &\in \mathbb{Z}_+ \end{aligned} \tag{$INLP^1$}$$

$$\begin{aligned} \max \qquad & x_1^1 + x_2^1 \\ x_1^1 &\leq (B_1 - D_1^1)^+ \\ x_2^1 &\leq (B_2 - D_2^1)^+ \\ x_1^1 \leq P_1^1, &\quad x_2^1 \leq P_2^1 \\ B_1 + B_2 &= B \\ x_1^1, x_2^1 &\in \mathbb{Z}_+ \\ B_1, B_2 &\in \mathbb{R}_+ \end{aligned} \tag{$INLP^1_\Delta$}$$

Property 1. *Given a budget B, let $f^*(B)$ and $f^*_\Delta(B)$ be the optimal objective values* *of* $\left(INLP^1\right)$ *and* $\left(INLP^1_\Delta\right)$. *Both* $f^*(B)$ *and* $f^*_\Delta(B)$ *are monotonically non-decreasing with* B *and* $f^*(B) \leq f^*_\Delta(B)$.

Proof. Since the feasible region of $\left(INLP^1\right)$ for a given B is a subset of the feasible region of $\left(INLP^1\right)$ for $B' \geq B$, $f^*(B)$ is non-decreasing with B increasing. The same holds for $f^*_\Delta(B)$. We then prove that $f^*(B) \leq f^*_\Delta(B)$ by showing that an optimal solution for $\left(INLP^1\right)$ yields a feasible solution for $\left(INLP^1_\Delta\right)$. Assume first that an optimal solution for $\left(INLP^1\right)$ satisfies $(x^1_1)^* = 0$. Then, the solution $\hat{x}^1_1 = (x^1_1)^* = 0$, $\hat{x}^1_2 = (x^1_2)^*$, $B_1 = 0$, $B_2 = B$ is feasible for $\left(INLP^1_\Delta\right)$ as $\hat{x}^1_2 \leq (B - D^1_2)^+$ holds since $\hat{x}^1_2 = (x^1_2)^* \leq (B - D^1_1 - D^1_2)^+ \leq (B - D^1_2)^+$. Assume then that an optimal solution for $\left(INLP^1\right)$ satisfies $(x^1_1)^* > 0$. Then, the solution $\hat{x}^1_1 = (x^1_1)^*$, $\hat{x}^1_2 = (x^1_2)^*$, $B_1 = (x^1_1)^* + D^1_1$, $B_2 = B - (x^1_1)^* - D^1_1$ is feasible for $\left(INLP^1_\Delta\right)$ as $\hat{x}^1_2 = (x^1_2)^* \leq (B - (x^1_1)^* - D^1_1 - D^1_2)^+$ holds since $(x^1_1)^* > 0$ implies $B > D^1_1 + D^1_2$; that is, $B - D^1_1 - D^1_2 \geq (x^1_1)^* + (x^1_2)^*$ by the first constraint of $\left(INLP^1\right)$. □

Property 2 refines the inequality $f^*(B) \leq f^*_\Delta(B)$ for $N = 1$ by providing budget ranges for which the inequality is strict or holds with equality.

Property 2. *Given a budget* B, *let* $f^*(B)$ *and* $f^*_\Delta(B)$ *be the optimal objective values of* $\left(INLP^1\right)$ *and* $\left(INLP^1_\Delta\right)$. *We have:*

$f^*(B) = f^*_\Delta(B)$ *if* $B \leq B_{min}$ *or* $B \geq D^1_1 + D^1_2 + \max\{P^1_1, P^1_2\}$, *and*

$f^*(B) < f^*_\Delta(B)$ *if* $B_{min} < B < D^1_1 + D^1_2 + \max\{P^1_1, P^1_2\}$.

Proof. Consider first the case $B \leq B_{min} = \min(D^1_1, D^1_2)$, then $(x^1_1)^* = (x^1_2)^* = (\hat{x}^1_1)^* = (\hat{x}^1_2)^* = 0$, and thus $f^*(B) = f^*_\Delta(B) = 0$. Consider then the case $B \geq D^1_1 + D^1_2 + \max\{P^1_1, P^1_2\}$. Adding the last two constraints of $\left(INLP^1_\Delta\right)$ yields that $P^1_1 + P^1_2$ is an upper bound; that is, $f^*(B) \leq f^*_\Delta(B) \leq P^1_1 + P^1_2$. Without loss of generality, we assume $P_1^1 > P_2^1$ and consider two sub-cases. Sub-case $B \geq D_1^1 + D_2^1 + P_1^1 + P_2^1$: then the solution $(x_1^1)^* = P_1^1$ and $(x_2^1)^* = P_2^1$ is feasible for $\left(INLP^1\right)$ and, thus, $P_1^1 + P_2^1 \leq f^*(B) \leq f_\Delta^*(B) \leq P_1^1 + P_2^1$ which implies $f^*(B) = f_\Delta^*(B)$. Sub-case $D_1^1 + D_2^1 + \max\{P_1^1, P_2^1\} \leq B < D_1^1 + D_2^1 + P_1^1 + P_2^1$: then an optimal solution for $\left(INLP^1\right)$ satisfies $(x_1^1)^* = P_1^1$ and $(x_2^1)^* = B - D_1^1 - D_2^1 - P_1^1$. Furthermore, for $\left(INLP_\Delta^1\right)$, if $B_1 - D_1^1 < 0$ then $x_1^1 = 0$ and $x_2^1 \leq P_2^1 < P_1^1 < f^*(B)$which is not an optimal solution, therefore we can assume that $B_1 - D_1^1 \geq 0$. In addition, if $B_2 - D_2^1 < 0$ then $x_2^1 = 0$ and $x_1^1 \leq P_1^1$ which can not yield a strictly larger objective value. Thus, we can assume that $B_1 - D_1^1 \geq 0$ and $B_2 - D_2^1 \geq 0$. Adding the first two constraints shows that $f_\Delta^*(B) \leq B - D_1^1 - D_2^1$, and thus a strictly larger objective value can not be achieve; that is $f^*(B) = f_\Delta^*(B)$. Finally, consider the case $\min(D_1^1, D_2^1) < B < D_1^1 + D_2^1 + \max\{P_1^1, P_2^1\}$. We consider 2 sub-cases. Sub-case $\min(D_1^1, D_2^1) < B \leq D_1^1 + D_2^1$: then $f^*(B) = 0$ while $B_1^* = B$ and $B_2^* = 0$ yields a feasible solution for $\left(INLP_\Delta^1\right)$ which a strictly positive objective value and, thus, $f_\Delta^*(B) > f^*(B)$. Sub-case $D_1^1 + D_2^1 < B < D_1^1 + D_2^1 + \max\{P_1^1, P_2^1\}$ and, without loss of generality, $P_1^1 > P_2^1$: then $f^*(B) \leq B - D_1^1 - D_2^1 < P_1^1$ by the first constraint of $\left(INLP^1\right)$. On the other hand, setting $B_1^* = B$, $B_2^* = 0$ and $\hat{x}_1^1 = \min\{B - D_1^1, P_1^1\}$ yields a feasible solution for $\left(INLP_\Delta^1\right)$ with an objective value of at least P_1^1; that is, $f_\Delta^*(B) \geq P_1^1 > f^*(B)$. $\square$

3.4.2 Case $N = 2$

The case $N = 2$ corresponds to the simplest random demand with only two realizations. We assume that both realizations have probability 0.5 and omit this constant term in the objectives for clarity. In the associated formulations $(INLP^2)$ and $(INLP_\Delta^2)$ below, superscripts are use to distinguish different realizations. For example, x_1^2, D_1^2, and P_1^2 refer to the second realization.

$$
\begin{aligned}
\max \qquad & x_1^1 + x_2^1 + x_1^2 + x_2^2 \qquad\qquad (INLP^2)\\
x_1^1 + x_2^1 &\leq (B - D_1^1 - D_2^1)^+\\
x_1^2 + x_2^2 &\leq (B - D_1^2 - D_2^2)^+\\
x_1^1 \leq P_1^1, \quad x_2^1 &\leq P_2^1\\
x_1^2 \leq P_1^2, \quad x_2^2 &\leq P_2^2\\
x_1^1, x_2^1, x_1^2, x_2^2 &\in \mathbb{Z}_+
\end{aligned}
$$

$$
\begin{aligned}
\max \qquad & x_1^1 + x_2^1 + x_1^2 + x_2^2 \qquad\qquad (INLP^2_\Delta)\\
x_1^1 &\leq (B_1 - D_1^1)^+\\
x_2^1 &\leq (B_2 - D_2^1)^+\\
x_1^2 &\leq (B_1 - D_1^2)^+\\
x_2^2 &\leq (B_2 - D_2^2)^+\\
x_1^1 \leq P_1^1, \quad x_2^1 &\leq P_2^1\\
x_1^2 \leq P_1^2, \quad x_2^2 &\leq P_2^2\\
B_1 + B_2 &= B\\
x_1^1, x_2^1, x_1^2, x_2^2 &\in \mathbb{Z}_+\\
B_1, B_2 &\in \mathbb{R}_+
\end{aligned}
$$

As the number of cases to consider in order to provide an analogue of Property 2 essentially increases exponentially with the number of realizations, comparing $(INLP^2)$ and $(INLP^2_\Delta)$ can be tedious. Thus, Property 3 focuses on the 3 scenarios: (i) the demands are large for both realizations, (ii) the demands are large for one realization but not for the other, and (*iii*) the demands are insufficient for both realizations.

Property 3. *Given a budget B, let $f^*(B)$ and $f^*_\Delta(B)$ be the optimal objective values of $\left(INLP^2\right)$ and $\left(INLP^2_\Delta\right)$ We have:*

$f^*(B) < f^*_\Delta(B)$ *if* $B_{min} < B \leq B^+_{min}$,

$f^*(B) \leq f^*_\Delta(B)$ *if* $B^+_{min} < B \leq B^+_{max}$, *and*

$f^*(B) = f^*_\Delta(B)$ *if* $0 \leq B \leq B_{min}$ *or* $B \geq \max\{D^1_1+P^1_1, D^2_1+P^2_1\}+\max\{D^1_2+P^1_2, D^2_2+P^2_2\}$.

Proof. Consider first the case $B \leq B_{min}$, then $(x^1_1, x^2_1, x^1_2, x^2_2)$ must be set to $(0,0,0,0)$ to obtain a feasible solution for $\left(INLP^2\right)$ and $\left(INLP^2_\Delta\right)$. Thus, $f^*(B) = f^*_\Delta(B) = 0$. Consider the case $B \geq \max\{D^1_1+P^1_1, D^2_1+P^2_1\}+\max\{D^1_2+P^1_2, D^2_2+P^2_2\}$. Note first that $P^1_1+P^1_2+P^2_1+P^2_2$ is an upper bound for both $f^*(B)$ and $f^*_\Delta(B)$ as implied by adding the last 4 constraints. Then, as $x^h_i = P^h_i$ is a feasible solution for $\left(INLP^2\right)$, $f^*(B) = P^1_1+P^1_2+P^2_1+P^2_2$. Similarly, $x^h_i = P^h_i, B_1 = \max\{D^1_1+P^1_1, D^2_1+P^2_1\}$ and $B_2 = B - B_1$ a feasible solution for $\left(INLP^2_\Delta\right)$ and the corresponding objective is also $P^1_1+P^1_2+P^2_1+P^2_2$; that is, $f^*_\Delta(B) = f^*(B)$. Consider the case $B \leq B^+_{min} = \min\{D^1_1+D^1_2, D^2_1+D^2_2\}$, then while $f^*(B) = 0$, setting $B^*_1 = B$ and $B^*_2 = 0$ yields a feasible solution for $\left(INLP^2_\Delta\right)$ with a strictly positive objective value; that is, $f^*(B) < f^*_\Delta(B)$. Consider the case $B^+_{min} < B \leq B^+_{max}$, and assume without loss of generality that $D^2_1 + D^2_2 > D^1_1 + D^1_2$. Since $B \leq D^2_1 + D^2_2$, the second constraints of $\left(INLP^2\right)$ is $x^2_1 + x^2_2 \leq 0$; that is $x^2_1 = x^2_2 = 0$. In other words, we can restrict to $(x^1_1, x^1_2, 0, 0)$ feasible solutions and use Property 2 to derive $f^*(B) \leq f^*_\Delta(B)$. ☐

3.4.3 Case $N = N_0$

As in Section 3.4.2, we assume that the N_0 realizations have probability $1/N_0$ and omit this constant term in the objective. In the associated formulations ($INLP^{N_0}$) and ($INLP_{\Delta}^{N_0}$ below superscripts are use to distinguish different realizations. For example, $x_1^h, x_2^h, D_1^h, D_2^h, P_1^h$, and P_1^h refer to the h-th realization.

$$\begin{aligned}
\max \quad & \sum_{h=1}^{N_0}(x_1^h + x_2^h) && \qquad (INLP^{N_0}) \\
x_1^h + x_2^h \leq (B - D_1^h - D_2^h)^+ & && h = 1, \ldots, N_0 \\
x_1^h \leq P_1^h, \quad x_2^h \leq P_2^h & && h = 1, \ldots, N_0 \\
x_1^h, x_2^h \in \mathbb{Z}_+ & && h = 1, \ldots, N_0
\end{aligned}$$

$$\begin{aligned}
\max \quad & \sum_{h=1}^{N_0}(x_1^h + x_2^h) && \qquad (INLP_{\Delta}^{N_0}) \\
x_1^h \leq (B_1 - D_1^h)^+ & && h = 1, \ldots, N_0 \\
x_2^h \leq (B_2 - D_2^h)^+ & && h = 1, \ldots, N_0 \\
x_1^h \leq P_1^h, \quad x_2^h \leq P_2^h & && h = 1, \ldots, N_0 \\
B_1 + B_2 = B & && \\
x_1^h, x_2^h \in \mathbb{Z}_+ & && h = 1, \ldots, N_0 \\
B_1, B_2 \in \mathbb{R}_+. & &&
\end{aligned}$$

As in Section 3.4.2, the number of cases being essentially intractable, Property 4 focuses on the 2 scenarios: (i) the demands are large enough for all N_0 realizations, and (ii) the demands are insufficient for all N_0 realizations.

Property 4. *Given a budget B, let $f^*(B)$ and $f_{\Delta}^*(B)$ be the optimal objective values* *of* $\left(INLP^{N_0}\right)$ *and* $\left(INLP^{N_0}_{\Delta}\right)$ *We have:*

$f^*(B)= f^*_{\Delta}(B)$ *if* $0 \leq B \leq B_{min}$ *or* $B \geq B^{\Sigma}_{max}$*, and*

$f^*(B)< f^*_{\Delta}(B)$ *if* $B_{min} < B \leq B^{+}_{min}$*.*

Proof. As for Property 3, for $B \leq B_{min}$, x^h_1 and x^h_2 must be set to 0 for $h = 1, \ldots, N_0$ to obtain a feasible solution for $\left(INLP^{N_0}\right)$ and $\left(INLP^{N_0}_{\Delta}\right)$. Thus, $f^*(B)$=$f^*_{\Delta}(B)$=0. Consider then $B \geq B^{\Sigma}_{max} = \sum_{i=1}^{2} \max_{h=1}^{N_0}\{D^h_i + P^h_i\}$. Note first that $\sum_{i=1}^{2}\sum_{h=1}^{N_0}\{P^h_i\}$ is an upper bound both $f^*(B)$ and $f^*_{\Delta}(B)$ as implied by adding the last $2N_0$ constraints. Since $x^h_i = P^h_i$ is a feasible solution for $\left(INLP^{N_0}\right)$, $f^*(B) = \sum_{i=1}^{2}\sum_{h=1}^{N_0}\{P^h_i\}$. Similarly, $x^h_i = P^h_i, B_1 = \max_{h=1}^{N_0}\{D^h_1 + P^h_1\}$ and $B_2 = B - B_1$ a feasible solution for $\left(INLP^{N_0}_{\Delta}\right)$ and the corresponding objective is $\sum_{i=1}^{2}\sum_{h=1}^{N_0}\{P^h_i\}$; that is, $f^*_{\Delta}(B) = f^*(B)$.

Consider the case $B \leq B^{+}_{min} = \min_{h=1}^{N_0}\{D^h_1 + D^h_2\}$. Wwhile $f^*(B) = 0$, setting $B^*_1 = B$ and $B^*_2 = 0$ yields a feasible solution for $\left(INLP^{N_0}_{\Delta}\right)$ with a strictly positive objective value; that is, $f^*(B)$ <$f^*_{\Delta}(B)$. □

3.5 Application of the proposed alteration

Sections 3.1, 3.2 and 3.3 suggest that component commonality is not always beneficial while component commonality could be beneficial in the product design, supply chain management etc. The design of the product may require common components and the volume of easy to acquire common components may be an overriding factor. We propose a combined inventory management system applicable to any ATO system with common components: Compute first the optimal objective value $OBJ1$ and the corresponding base stock $BS1$ using the original formulation. Then separate component inventories with respect to the different products; that is, each product is served by dedicated components, and apply the formulation again to compute another optimal objective value $OBJ2$ and the corresponding base stock $BS2$. If $OBJ1 \geq OBJ2$, keep the original BOM in the management system and use $BS1$ as base stock. Otherwise, apply the modified BOM and use $BS2$ as the base stock by marking components corresponding to products. When we place a replenishment order, we convert the product-dedicated components back to the general components. When the replenishment order arrives, we need to mark the general components to product-dedicated components according to the modified BOM and previous period demand. Thus, each time the budget changes or the demand changes significantly, the joint problem can be resolved to decide which BOM to use in the system. Note that switching between the two models is easy and that the implementation is straightforward as it acts essentially as a black box for the supply chain system. The input is the budget and demand, the output is the base stock, replenishment order and products assembled in each period. Converting back the original components to issue the replenishment order does not impact the supply chain. From the inventory management perspective, the way the components are kept is unchanged as some components are simply labeled in our model not physically. We provide a computational result of the combined inventory management system for the IBM system in Figure 3.3 where LB and UB denote, respectively, the lower and upper bounds for the ($INLP$) and ($INLP_{comb}$).

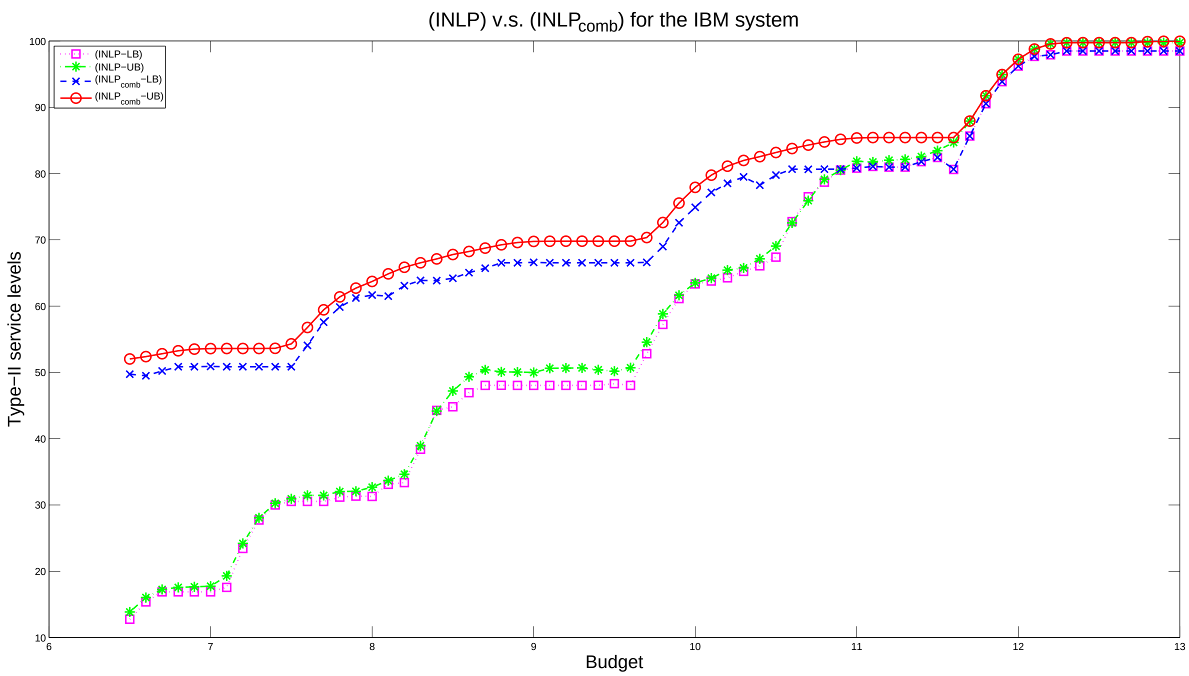

Figure 3.3: Type II service levels for ($INLP$) and ($INLP_{comb}$) for the IBM system

Chapter 4

Conclusions and future work

We highlighted the critical role played by the piece-wise linear inventory availability constraints and the associated feasibility issue and challenges for sample generation. The computational results analyze the impact resulting from substituting linear functions for piece-wise linear ones. While the impact decreases when the budget increases, it remains significant for low to medium level budgets. In addition, the benefits of component commonality are analyzed from both the theoretical and computational aspects and illustrated for specific ATO systems. We introduce a simple inventory control method applicable in practice where a more flexible design of products and components allows us to exploit the different degrees of component commonality according to the budget. Future works include an enhanced analysis of the sample generation process for (*ILP*) and a tighter estimate of the gap between the optimal objective values of (*ILP*) and (*INLP*). Additional future directions include a proposed inventory control method which might be achieved via a different degree of component commonality for the subset of components and products, see [10] for preliminary results.

Appendix A

Tables for the Zhang's system

A.1 (*INLP*)-UB Table

A.2 (*INLP*)-LB Table

A.3 (*ILP*)-UB Table

A.4 (*ILP*)-LB Table

A.5 (*INLP*) and ($INLP_\Delta$) Table

A.6 Sample statistics Table

Budget	**75**	**80**	**85**	**90**	**95**	**100**	**Unfiltered**
2000	9.13	9.27	9.13	9.30	9.16	9.12	9.15
3000	9.34	9.33	9.15	9.31	9.17	9.13	9.15
4000	24.01	18.17	13.27	11.40	10.41	10.16	9.72
5000	49.31	44.24	38.64	33.73	29.30	26.48	22.32
6000	68.14	64.74	60.81	57.05	53.56	51.14	47.30
7000	77.74	74.71	73.64	72.26	70.43	68.76	66.32
7500	86.68	82.22	77.69	75.32	74.26	73.29	71.99
8000	93.52	90.35	86.58	82.76	79.23	76.81	74.96
8500	97.58	95.87	93.50	90.76	87.87	85.34	82.09
9000	99.25	98.65	97.73	96.25	94.31	92.32	89.88
9500	99.62	99.60	99.53	99.12	98.11	96.78	95.33
10000	99.66	99.73	99.96	100.06	99.63	98.96	98.41
10500	99.67	99.73	100.00	100.21	99.95	99.64	99.83
11000	99.67	99.73	100.00	100.22	99.99	99.76	100.23
11500	99.67	99.73	100.00	100.22	99.99	99.78	100.31
12000	99.67	99.73	100.00	100.22	99.99	99.78	100.32

Table A.1: Type II service levels for (*INLP*)-UB for the Zhang system using different samples

Budget	**75**	**80**	**85**	**90**	**95**	**100**	**Unfiltered**
2000	9.13	9.17	9.14	9.14	9.11	9.13	9.05
3000	9.37	9.22	9.14	9.14	9.11	9.13	9.05
4000	25.98	20.06	14.80	12.49	11.36	10.37	9.48
5000	50.94	45.94	41.07	36.95	31.85	28.92	22.55
6000	69.16	65.97	62.55	59.49	55.57	53.17	47.44
7000	79.77	74.78	74.02	73.12	71.35	70.06	66.38
7500	88.44	84.26	80.21	76.37	74.51	74.04	71.97
8000	94.74	91.90	88.75	85.72	82.22	79.06	74.70
8500	98.21	96.82	94.85	92.86	90.27	87.66	81.94
9000	99.56	99.08	98.18	97.28	95.68	94.03	89.65
9500	99.93	99.84	99.50	99.32	98.52	97.80	95.05
10000	99.99	100.00	99.85	99.98	99.59	99.46	98.12
10500	100.00	100.02	99.88	100.13	99.85	99.98	99.51
11000	100.00	100.03	99.91	100.14	99.90	100.10	99.99
11500	99.97	100.05	99.91	100.14	99.90	100.10	100.12
12000	100.00	100.02	99.91	100.14	99.90	100.10	100.13

Table A.2: Type II service levels for (*INLP*)-LB for the Zhang system using different samples

Budget	**75**	**80**	**85**	**90**	**95**	**100**
7500	83.76	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
8000	93.16	88.24	*N/A*	*N/A*	*N/A*	*N/A*
8500	97.55	95.66	92.13	*N/A*	*N/A*	*N/A*
9000	99.25	98.63	97.61	95.38	*N/A*	*N/A*
9500	99.62	99.60	99.52	99.04	97.64	*N/A*
10000	99.66	99.73	99.96	100.05	99.60	98.71
10500	99.67	99.73	100.00	100.21	99.94	99.63
11000	99.67	99.73	100.00	100.22	99.99	99.76
11500	99.67	99.73	100.00	100.22	99.99	99.78
12000	99.67	99.73	100.00	100.22	99.99	99.78

Table A.3: Type II service levels for (*ILP*)-UB for the Zhang system using different samples

Budget	**75**	**80**	**85**	**90**	**95**	**100**
7500	87.09	N/A	N/A	N/A	N/A	N/A
8000	94.59	91.48	N/A	N/A	N/A	N/A
8500	98.21	96.83	94.46	N/A	N/A	N/A
9000	99.56	99.08	98.17	97.10	N/A	N/A
9500	99.94	99.84	99.51	99.33	98.50	N/A
10000	100.00	100.02	99.86	99.99	99.58	99.46
10500	100.00	100.05	99.92	100.14	99.85	99.98
11000	100.00	100.05	99.93	100.17	99.90	100.10
11500	100.00	100.05	99.93	100.17	99.91	100.12
12000	100.00	100.05	99.93	100.17	99.91	100.12

Table A.4: Type II service levels for (*ILP*)-LB for the Zhang system using different samples

Budget	**(*INLP*)-UB**	**(*INLP*)-LB**	**($INLP_\Delta$)-UB**	**($INLP_\Delta$)-LB**
2000	9.15	9.05	14.46	14.23
3000	9.15	9.05	26.65	26.36
4000	9.72	9.48	43.82	43.59
5000	22.32	22.55	53.81	53.66
6000	47.30	47.44	59.32	58.87
7000	66.32	66.38	69.81	69.00
7500	71.99	71.97	75.61	75.27
8000	74.96	74.70	80.36	79.96
8500	82.09	81.94	83.07	82.70
9000	89.88	89.65	85.29	84.10
9500	95.33	95.05	90.56	89.93
10000	98.41	98.12	94.89	94.21
10500	99.83	99.51	97.66	96.94
11000	100.23	99.99	99.24	98.56
11500	100.31	100.12	100.03	99.45
12000	100.32	100.13	100.27	99.81

Table A.5: Type II service levels for (*INLP*) and ($INLP_\Delta$) for the Zhang system

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.99	150.05	50.20	30.24
t-1	69.10	114.96	39.67	27.56
t-2	69.71	115.84	39.53	27.52
t-3	95.47	142.92	46.33	27.66
t-4	100.02	150.36	46.40	27.87

Table A.6: Mean Value of Sample used in Joint for the Zhang system at 75

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	100.07	149.28	49.72	30.26
t-1	75.63	122.38	41.90	28.30
t-2	76.15	123.27	41.89	28.38
t-3	96.12	145.34	47.10	28.14
t-4	100.38	149.89	46.65	28.13

Table A.7: Mean Value of Sample used in Joint for the Zhang system at 80

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	100.06	150.06	50.20	30.03
t-1	81.74	130.24	43.64	28.56
t-2	82.03	129.92	44.01	28.52
t-3	97.04	145.98	47.87	28.90
t-4	99.94	150.61	48.04	28.70

Table A.8: Mean Value of Sample used in Joint for the Zhang system at 85

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.97	150.27	49.56	30.40
t-1	87.62	136.15	45.95	29.20
t-2	87.32	136.02	45.73	29.17
t-3	98.45	148.17	48.58	28.97
t-4	100.52	149.82	48.80	29.31

Table A.9: Mean Value of Sample used in Joint for the Zhang system at 90

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	100.33	150.12	50.12	30.11
t-1	92.58	142.00	47.48	29.45
t-2	92.87	141.17	47.51	29.37
t-3	98.37	148.72	49.31	29.32
t-4	99.92	149.83	48.91	29.73

Table A.10: Mean Value of Sample used in Joint for the Zhang system at 95

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	100.07	148.89	50.04	30.05
t-1	95.90	144.96	48.88	29.66
t-2	96.73	145.45	48.96	29.90
t-3	99.54	150.24	49.64	29.74
t-4	100.71	149.97	49.51	29.90

Table A.11: Mean Value of Sample used in Joint for the Zhang system at 100

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.75	150.76	50.43	29.94
t-1	100.33	149.92	50.04	30.16
t-2	99.63	150.25	50.23	29.91
t-3	100.01	150.22	50.10	30.22
t-4	100.32	149.56	50.15	30.26

Table A.12: Mean Value of Sample used in Joint for the Zhang system - Unfiltered

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	100.28	149.85	49.89	30.10
t-1	66.58	113.32	38.70	25.67
t-2	66.40	112.56	38.69	25.98
t-3	93.11	139.99	45.31	25.89
t-4	100.17	149.73	45.28	25.72

Table A.13: Mean Value of Sample used in Alloc for the Zhang system at 75

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.96	150.00	50.09	30.11
t-1	73.34	119.55	41.04	26.42
t-2	73.02	119.53	40.94	26.35
t-3	92.64	139.33	45.87	26.63
t-4	100.19	150.21	45.78	26.51

Table A.14: Mean Value of Sample used in Alloc for the Zhang system at 80

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.91	149.95	49.65	30.12
t-1	79.13	126.86	42.58	26.91
t-2	78.73	126.64	42.45	26.88
t-3	93.97	141.95	45.91	26.71
t-4	100.19	150.11	46.26	26.82

Table A.15: Mean Value of Sample used in Alloc for the Zhang system at 85

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.90	149.77	50.06	30.20
t-1	83.07	133.38	44.52	27.17
t-2	83.35	133.08	44.35	27.37
t-3	96.44	145.24	47.38	27.19
t-4	100.20	150.29	47.21	27.19

Table A.16: Mean Value of Sample used in Alloc for the Zhang system at 90

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.75	150.38	49.90	30.08
t-1	90.08	138.51	45.29	26.83
t-2	89.72	138.54	44.95	27.01
t-3	96.21	144.88	46.77	26.83
t-4	100.31	149.73	46.61	26.76

Table A.17: Mean Value of Sample used in Alloc for the Zhang system at 95

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	100.42	149.85	50.14	30.04
t-1	92.50	142.51	47.40	28.22
t-2	92.53	141.96	47.45	28.27
t-3	97.51	146.72	48.48	28.38
t-4	100.09	150.05	48.36	28.45

Table A.18: Mean Value of Sample used in Alloc for the Zhang system at 100

Period	Prod 1	Prod 2	Prod 3	Prod 4
t-0	99.67	150.22	50.39	30.07
t-1	99.96	149.32	49.94	30.18
t-2	99.58	149.97	49.87	30.15
t-3	100.23	150.35	50.07	30.11
t-4	100.38	149.83	50.00	30.12

Table A.19: Mean Value of Sample used in Alloc for the Zhang system - Unfiltered

Appendix B

Tables for the IBM system

B.1 (*INLP*)-UB Table

B.2 (*INLP*)-LB Table

B.3 (*ILP*)-UB Table

B.4 (*ILP*)-LB Table

B.5 (*INLP*) and ($INLP_{\Delta}$) Table

B.6 Sample statistics Table

Budget	106	107	108	109	110	111	112	113	114	115	Unfiltered
6.5×10^6	17.47	17.38	17.56	17.21	16.41	15.48	14.53	15.06	14.13	14.06	14.05
6.6×10^6	17.66	17.59	17.59	17.46	17.41	17.16	16.47	17.20	16.33	16.12	16.19
6.7×10^6	17.66	17.71	17.77	17.56	17.46	17.36	17.39	18.18	17.29	17.22	17.17
6.8×10^6	17.78	17.71	17.78	17.71	17.70	17.57	17.52	18.28	17.52	17.41	17.46
6.9×10^6	22.95	19.31	17.93	17.71	17.71	17.71	17.73	18.45	17.68	17.63	17.59
7×10^6	29.01	26.33	22.86	19.91	18.46	17.84	17.80	18.56	17.76	17.75	17.72
7.1×10^6	31.44	30.72	28.62	26.23	23.84	21.86	20.36	20.54	19.58	19.49	19.80
7.2×10^6	31.55	31.73	31.29	30.33	28.78	27.33	25.72	25.52	24.72	24.60	24.79
7.3×10^6	31.57	31.74	31.57	31.64	31.33	30.60	29.61	29.47	28.76	28.75	28.62
7.4×10^6	32.08	31.80	31.57	31.68	31.70	31.65	31.43	31.67	30.92	30.93	30.76
7.5×10^6	32.22	32.27	32.01	31.71	31.70	31.68	31.78	32.30	31.56	31.58	31.48
7.6×10^6	32.36	32.31	32.33	32.22	31.98	31.77	31.80	32.33	31.64	31.65	31.56
7.7×10^6	34.13	32.99	32.37	32.32	32.37	32.24	32.02	32.58	31.82	31.84	31.75
7.8×10^6	34.33	34.48	33.76	32.62	32.42	32.37	32.34	32.92	32.21	32.19	32.07
7.9×10^6	34.82	34.54	34.59	34.17	33.33	32.67	32.46	33.02	32.38	32.32	32.27
8×10^6	40.99	36.78	34.74	34.54	34.43	33.98	33.36	33.65	32.98	32.88	32.90
8.1×10^6	47.32	43.99	40.15	36.40	34.80	34.43	34.43	34.78	34.01	33.94	33.93
8.2×10^6	50.66	49.03	46.44	43.06	40.04	37.41	35.99	36.16	35.31	35.26	35.47
8.3×10^6	51.14	50.98	50.15	48.22	46.05	43.66	41.43	41.18	40.35	40.27	40.45
8.4×10^6	51.14	51.10	51.13	50.74	49.77	48.12	46.46	46.18	45.50	45.53	45.49
8.5×10^6	51.14	51.10	51.18	51.13	51.02	50.53	49.58	49.33	48.90	48.95	48.79
8.6×10^6	51.14	51.10	51.18	51.13	51.12	51.10	50.89	50.90	50.55	50.66	50.41
8.7×10^6	51.14	51.10	51.18	51.13	51.12	51.13	51.09	51.28	51.03	50.88	50.86
8.8×10^6	51.14	51.10	51.18	51.13	51.12	51.13	51.10	51.32	51.11	51.18	50.91
8.9×10^6	51.14	51.10	51.18	51.13	51.12	51.13	51.10	51.32	51.12	51.19	50.92
9×10^6	51.14	51.10	51.18	51.13	51.12	51.13	51.10	51.32	51.12	51.19	50.92
9.1×10^6	51.14	51.10	51.18	51.13	51.12	51.13	51.10	51.32	51.12	51.19	50.92
9.2×10^6	51.14	51.10	51.18	51.13	51.12	51.13	51.10	51.32	51.12	51.03	50.92
9.3×10^6	52.45	51.11	51.18	51.13	51.12	51.13	51.10	51.32	51.12	50.96	50.92
9.4×10^6	58.68	54.51	51.62	51.17	51.12	51.13	51.10	51.32	51.12	50.96	50.92
9.5×10^6	63.47	60.44	56.82	53.35	51.48	51.16	51.14	51.32	51.12	51.17	50.95
9.6×10^6	65.45	64.28	61.97	59.06	56.16	53.56	52.23	52.06	51.70	51.76	51.77
9.7×10^6	65.59	65.53	64.91	63.23	61.13	58.81	56.61	56.02	55.41	55.19	55.62
9.8×10^6	66.17	65.56	65.56	65.33	64.34	62.71	61.04	60.35	59.89	59.82	59.99
9.9×10^6	67.85	66.71	65.69	65.66	65.58	64.94	63.91	63.35	63.09	63.11	63.00
1×10^7	68.02	68.04	67.11	66.00	65.71	65.58	65.33	65.02	64.83	64.75	64.66
1.01×10^7	68.96	68.09	68.02	67.56	66.52	65.85	65.71	65.60	65.55	65.42	65.30
1.02×10^7	74.68	70.40	68.24	68.11	67.83	67.02	66.39	66.04	65.96	65.79	65.86
1.03×10^7	79.48	76.21	72.20	69.10	68.13	67.88	67.63	67.13	67.04	66.84	66.90
1.04×10^7	82.06	80.34	77.41	74.07	71.04	69.04	68.48	68.07	67.96	68.01	67.92
1.05×10^7	82.57	82.40	80.91	78.60	76.15	73.62	71.45	70.60	70.27	70.26	70.50
1.06×10^7	82.58	82.69	82.44	81.49	79.92	77.88	76.04	75.00	74.65	74.77	74.82
1.07×10^7	82.87	82.70	82.59	82.64	82.08	80.79	79.49	78.49	78.30	78.51	78.30
1.08×10^7	83.23	82.94	82.69	82.73	82.68	82.31	81.65	80.82	80.77	80.94	80.65
1.09×10^7	83.31	83.26	83.16	82.81	82.71	82.63	82.64	82.10	82.10	82.19	81.92
1.1×10^7	84.63	83.38	83.35	83.25	82.91	82.68	82.78	82.45	82.55	82.67	82.29
1.11×10^7	85.45	84.83	83.65	83.37	83.33	82.96	82.87	82.52	82.69	82.72	82.42
1.12×10^7	85.85	85.46	85.12	83.84	83.43	83.29	83.15	82.83	82.96	83.01	82.64
1.13×10^7	90.60	86.41	85.62	85.16	84.17	83.47	83.43	83.10	83.26	83.09	82.93
1.14×10^7	95.44	91.56	87.45	85.69	85.26	84.39	83.87	83.43	83.52	83.61	83.30
1.15×10^7	98.57	95.97	92.58	88.52	86.23	85.36	84.92	84.30	84.37	83.51	84.19
1.16×10^7	99.92	98.79	96.61	93.36	90.53	87.73	86.39	85.63	85.52	85.65	85.69
1.17×10^7	100.04	99.98	99.14	96.96	94.78	92.15	89.96	88.90	88.58	88.67	88.87
1.18×10^7	100.04	100.07	100.13	99.23	97.78	95.72	93.93	92.80	92.61	92.64	92.70
1.19×10^7	100.04	100.07	100.17	100.13	99.56	98.21	96.90	95.81	95.79	95.76	95.72
1.2×10^7	100.04	100.07	100.17	100.18	100.12	99.63	98.88	97.92	98.02	98.00	97.85
1.21×10^7	100.04	100.07	100.17	100.18	100.14	99.96	99.92	99.20	99.36	99.32	99.08
1.22×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.63	99.90	99.89	99.52
1.23×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.01	99.98	99.58
1.24×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	99.98	99.58
1.25×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	99.98	99.58
1.26×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	99.98	99.58
1.27×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	99.98	99.58
1.28×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	99.98	99.58
1.29×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	99.98	99.58
1.3×10^7	100.04	100.07	100.17	100.18	100.14	99.98	100.16	99.66	100.02	98.97	99.58

Table B.20: Type II service levels for (*INLP*)-UB for the IBM system using different samples

Budget	106	107	108	109	110	111	112	113	114	115	Unfiltered
6.5×10^6	17.50	17.46	17.43	17.47	17.45	17.37	17.39	17.14	16.62	15.78	13.18
6.6×10^6	17.50	17.46	17.43	17.47	17.46	17.40	17.44	17.43	17.37	17.10	15.70
6.7×10^6	17.50	17.46	17.43	17.47	17.46	17.40	17.44	17.43	17.44	17.32	16.84
6.8×10^6	25.08	17.46	17.43	17.47	17.46	17.40	17.44	17.43	17.44	17.33	17.24
6.9×10^6	30.42	28.49	27.04	17.47	17.46	17.40	17.44	17.43	17.44	17.33	17.32
7×10^6	31.58	31.33	30.79	28.84	27.28	25.62	23.69	21.29	19.03	17.33	17.27
7.1×10^6	31.62	31.76	31.54	31.42	30.79	29.86	29.04	27.93	25.52	23.56	18.35
7.2×10^6	31.62	31.77	31.58	31.76	31.63	31.50	31.29	30.77	29.70	28.53	23.89
7.3×10^6	31.82	31.77	31.58	31.77	31.68	31.75	31.66	31.55	31.23	30.76	27.74
7.4×10^6	32.05	32.10	31.58	31.77	31.69	31.76	31.69	31.63	31.54	31.42	30.01
7.5×10^6	32.05	32.11	31.93	32.11	31.84	31.76	31.69	31.64	31.57	31.59	31.01
7.6×10^6	34.41	32.11	31.93	32.16	32.05	32.02	32.01	31.64	31.57	31.63	31.32
7.7×10^6	34.57	34.56	34.40	32.16	32.05	32.07	32.11	32.00	31.88	31.63	31.37
7.8×10^6	34.57	34.57	34.50	34.48	34.24	32.07	32.12	32.02	31.99	31.80	31.37
7.9×10^6	43.17	34.57	34.50	34.55	34.53	34.45	34.30	34.10	33.05	32.34	31.48
8×10^6	48.77	46.59	44.09	34.55	34.54	34.49	34.45	34.45	34.32	33.95	31.62
8.1×10^6	50.59	50.17	48.74	46.70	44.53	39.28	39.59	34.48	34.44	34.29	33.05
8.2×10^6	50.80	50.86	50.63	50.07	48.92	47.54	45.65	43.35	40.55	38.17	33.91
8.3×10^6	50.80	50.91	50.84	50.88	50.62	50.24	49.29	47.90	46.22	44.00	39.20
8.4×10^6	50.80	50.91	50.85	50.94	50.87	50.90	50.62	50.17	49.38	48.03	44.46
8.5×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.84	50.82	50.62	49.96	47.76
8.6×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.90	50.92	50.63	49.49
8.7×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.80	50.32
8.8×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.85	50.51
8.9×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.85	50.58
9×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.85	50.58
9.1×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.85	50.53
9.2×10^6	50.80	50.91	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.81	50.53
9.3×10^6	62.08	58.65	50.85	50.94	50.89	50.97	50.85	50.91	50.98	50.81	50.53
9.4×10^6	64.60	63.47	61.57	58.10	50.89	50.97	50.85	50.91	50.98	50.81	50.57
9.5×10^6	65.30	65.19	64.48	63.13	61.13	59.13	56.77	50.91	50.98	50.81	50.58
9.6×10^6	65.36	65.54	65.26	65.19	64.24	63.29	61.82	59.78	56.91	54.28	50.53
9.7×10^6	65.36	65.56	65.35	65.59	65.26	65.04	64.36	63.33	61.61	59.33	53.92
9.8×10^6	67.75	67.42	65.35	65.63	65.47	65.56	65.28	64.87	64.03	62.55	58.40
9.9×10^6	67.87	68.00	67.74	65.63	65.48	65.64	65.50	65.38	65.18	64.32	61.65
1×10^7	67.87	68.02	67.91	67.93	67.40	65.64	65.52	65.49	65.44	65.00	63.56
1.01×10^7	78.81	68.02	67.92	68.02	67.94	67.88	67.43	66.81	65.53	65.32	64.49
1.02×10^7	81.45	80.18	78.00	74.26	67.97	68.05	67.80	67.77	67.42	66.65	64.91
1.03×10^7	82.28	82.07	81.27	79.01	76.94	68.06	67.86	67.94	67.83	67.51	65.40
1.04×10^7	82.42	82.57	82.19	81.81	80.62	79.36	77.47	75.39	72.10	69.81	66.70
1.05×10^7	82.43	82.67	82.40	82.52	82.14	81.62	80.51	79.34	77.30	74.80	68.21
1.06×10^7	82.43	82.68	82.42	82.70	82.48	82.48	81.98	81.33	80.20	78.57	73.09
1.07×10^7	82.84	82.88	82.42	82.71	82.57	82.70	82.43	82.29	81.79	80.73	76.66
1.08×10^7	82.86	83.02	82.69	82.71	82.57	82.73	82.52	82.49	82.36	81.88	79.15
1.09×10^7	85.09	83.02	82.78	83.01	82.57	82.73	82.53	82.54	82.51	82.27	80.66
1.1×10^7	85.34	85.31	82.78	83.10	82.92	82.86	82.53	82.55	82.55	82.40	81.54
1.11×10^7	85.37	85.48	85.28	84.65	82.94	83.04	82.92	82.82	82.55	82.46	81.88
1.12×10^7	95.97	85.48	85.34	85.42	85.15	84.88	82.96	82.91	82.85	82.46	82.05
1.13×10^7	98.57	97.23	93.31	85.49	85.39	85.33	84.77	84.29	82.93	82.55	81.92
1.14×10^7	99.57	99.21	97.88	94.97	85.43	85.46	85.25	85.23	84.87	83.99	81.93
1.15×10^7	99.89	99.93	99.31	98.36	96.39	95.02	92.25	88.96	85.25	84.79	82.24
1.16×10^7	99.92	100.11	99.74	99.67	98.70	98.12	96.71	94.83	91.87	89.66	83.63
1.17×10^7	99.92	100.14	99.83	100.04	99.69	99.49	98.68	97.53	95.83	93.69	86.68
1.18×10^7	99.92	100.14	99.85	100.18	99.96	99.97	99.51	99.16	97.92	96.37	90.89
1.19×10^7	99.92	100.14	99.85	100.18	100.02	100.11	99.89	99.75	99.31	98.10	94.09
1.2×10^7	99.92	100.14	99.85	100.19	100.03	100.13	99.96	99.91	99.72	99.15	96.31
1.21×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.97	99.91	99.54	97.62
1.22×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.98	99.73	98.66
1.23×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.78	99.14
1.24×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.80	99.14
1.25×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.80	99.14
1.26×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.78	99.14
1.27×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.78	99.14
1.28×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.78	99.14
1.29×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.78	99.14
1.3×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	100.00	99.80	99.14

Table B.21: Type II service levels for (*INLP*)-LB for the IBM system using different samples

Budget	**106**	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**
1.06×10^7	17.78	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.07×10^7	35.68	20.90	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.08×10^7	48.05	38.43	23.32	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.09×10^7	58.56	50.69	40.59	27.26	N/A	N/A	N/A	N/A	N/A	N/A
1.1×10^7	68.11	61.04	52.98	44.10	31.48	N/A	N/A	N/A	N/A	N/A
1.11×10^7	76.72	70.34	63.20	56.05	48.30	37.66	N/A	N/A	N/A	N/A
1.12×10^7	84.27	78.60	72.29	66.01	60.05	53.43	44.37	N/A	N/A	N/A
1.13×10^7	90.59	85.73	80.28	74.73	69.66	64.60	58.88	54.97	N/A	N/A
1.14×10^7	95.44	91.55	87.10	82.19	77.89	73.57	69.35	66.74	64.73	N/A
1.15×10^7	98.57	95.97	92.56	88.39	84.78	81.07	77.62	75.57	74.43	74.88
1.16×10^7	99.92	98.79	96.61	93.33	90.41	87.21	84.34	82.61	81.88	82.09
1.17×10^7	100.04	99.98	99.14	96.96	94.75	92.04	89.68	88.17	87.73	87.88
1.18×10^7	100.04	100.07	100.13	99.23	97.78	95.71	93.85	92.50	92.28	92.40
1.19×10^7	100.04	100.07	100.17	100.13	99.56	98.21	96.88	95.68	95.66	95.73
1.2×10^7	100.04	100.07	100.17	100.18	100.12	99.63	98.88	97.85	97.97	97.99
1.21×10^7	100.04	100.07	100.17	100.18	100.14	99.97	99.93	99.19	99.35	99.33
1.22×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.69	99.95	99.96
1.23×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.13	100.12
1.24×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13
1.25×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13
1.26×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13
1.27×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13
1.28×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13
1.29×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13
1.3×10^7	100.04	100.07	100.17	100.18	100.14	99.99	100.21	99.75	100.15	100.13

Table B.22: Type II service levels for (*ILP*)-UB for the IBM system using different samples

Budget	**106**	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**
1.06×10^7	43.59	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.07×10^7	57.42	49.15	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.08×10^7	68.21	61.81	53.56	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.09×10^7	77.13	72.06	66.37	53.33	N/A	N/A	N/A	N/A	N/A	N/A
1.1×10^7	85.04	80.68	75.95	66.88	61.21	N/A	N/A	N/A	N/A	N/A
1.11×10^7	91.53	88.03	83.83	77.23	71.96	67.91	N/A	N/A	N/A	N/A
1.12×10^7	96.00	93.68	90.10	84.63	80.16	76.94	72.95	N/A	N/A	N/A
1.13×10^7	98.52	97.33	95.14	90.90	87.58	84.79	81.28	78.10	N/A	N/A
1.14×10^7	99.57	99.21	97.93	95.52	92.95	90.88	88.08	85.09	81.26	N/A
1.15×10^7	99.89	99.93	99.31	98.40	96.73	95.36	93.23	90.55	87.08	84.19
1.16×10^7	99.92	100.11	99.74	99.64	98.77	98.18	96.69	94.94	92.12	89.59
1.17×10^7	99.92	100.14	99.84	100.05	99.69	99.49	98.58	97.53	95.71	93.67
1.18×10^7	99.92	100.14	99.85	100.18	99.96	99.97	99.49	99.16	97.92	96.38
1.19×10^7	99.92	100.14	99.85	100.18	100.02	100.11	99.88	99.75	99.31	98.10
1.2×10^7	99.92	100.14	99.85	100.19	100.03	100.13	99.96	99.91	99.73	99.15
1.21×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.97	99.91	99.54
1.22×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.97	99.73
1.23×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.24×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.25×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.26×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.27×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.28×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.29×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79
1.3×10^7	99.92	100.14	99.85	100.19	100.03	100.14	99.97	99.98	99.99	99.79

Table B.23: Type II service levels for (*ILP*)-LB for the IBM system using different samples

Budget	**(*INLP*)-UB**	**(*INLP*)-LB**	**($INLP_\Delta$)-UB**	**($INLP_\Delta$)-LB**
6.5×10^6	14.05	13.18	52.24	50.58
6.6×10^6	16.19	15.70	52.53	51.21
6.7×10^6	17.17	16.84	52.92	51.34
6.8×10^6	17.46	17.24	53.28	51.81
6.9×10^6	17.59	17.32	53.40	52.32
7×10^6	17.72	17.27	53.43	52.75
7.1×10^6	19.80	18.35	53.13	52.78
7.2×10^6	24.79	23.89	53.11	52.78
7.3×10^6	28.62	27.74	53.09	52.78
7.4×10^6	30.76	30.01	53.32	52.78
7.5×10^6	31.48	31.01	54.57	52.78
7.6×10^6	31.56	31.32	57.28	54.89
7.7×10^6	31.75	31.37	59.87	57.81
7.8×10^6	32.07	31.37	61.81	59.84
7.9×10^6	32.27	31.48	62.82	61.26
8×10^6	32.90	31.62	63.77	62.04
8.1×10^6	33.93	33.05	64.96	62.74
8.2×10^6	35.47	33.91	65.81	63.67
8.3×10^6	40.45	39.20	66.74	64.55
8.4×10^6	45.49	44.46	67.20	65.33
8.5×10^6	48.79	47.76	67.72	65.79
8.6×10^6	50.41	49.49	68.13	66.52
8.7×10^6	50.86	50.32	68.96	67.06
8.8×10^6	50.91	50.51	69.42	67.28
8.9×10^6	50.92	50.58	69.19	67.77
9×10^6	50.92	50.58	69.68	68.27
9.1×10^6	50.92	50.53	69.69	68.48
9.2×10^6	50.92	50.53	69.35	68.48
9.3×10^6	50.92	50.53	69.33	68.48
9.4×10^6	50.92	50.57	69.51	68.48
9.5×10^6	50.95	50.58	69.36	68.48
9.6×10^6	51.77	50.53	69.32	68.48
9.7×10^6	55.62	53.92	69.94	68.48
9.8×10^6	59.99	58.40	72.28	69.75
9.9×10^6	63.00	61.65	74.94	72.87
1×10^7	64.66	63.56	77.96	75.49
1.01×10^7	65.30	64.49	78.84	77.49
1.02×10^7	65.86	64.91	80.33	78.85
1.03×10^7	66.90	65.40	80.76	79.91
1.04×10^7	67.92	66.70	81.35	80.40
1.05×10^7	70.50	68.21	81.60	80.78
1.06×10^7	74.82	73.09	83.13	81.22
1.07×10^7	78.30	76.66	83.60	81.94
1.08×10^7	80.65	79.15	84.55	82.50
1.09×10^7	81.92	80.66	84.02	83.11
1.1×10^7	82.29	81.54	84.15	83.59
1.11×10^7	82.42	81.88	83.82	83.61
1.12×10^7	82.64	82.05	84.05	83.58
1.13×10^7	82.93	81.92	84.31	83.58
1.14×10^7	83.30	81.93	84.13	83.58
1.15×10^7	84.19	82.24	84.14	83.58
1.16×10^7	85.69	83.63	84.35	83.58
1.17×10^7	88.87	86.68	84.55	83.58
1.18×10^7	92.70	90.89	85.04	84.22
1.19×10^7	95.72	94.09	85.50	83.58
1.2×10^7	97.85	96.31	87.68	83.63
1.21×10^7	99.08	97.62	90.95	87.05
1.22×10^7	99.52	98.66	93.66	89.87
1.23×10^7	99.58	99.14	94.89	92.40
1.24×10^7	99.58	99.14	96.49	94.10
1.25×10^7	99.58	99.14	97.59	95.44
1.26×10^7	99.58	99.14	98.43	96.42
1.27×10^7	99.58	99.14	96.03	97.30
1.28×10^7	99.58	99.14	99.76	97.63
1.29×10^7	99.58	99.14	96.11	97.88
1.3×10^7	99.58	99.14	98.84	98.17

Table B.24: Type II service levels for (*INLP*) and ($INLP_\Delta$) for the IBM system

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.71	99.97	100.17	100.33	99.96	100.26
t-1	96.84	96.25	95.51	96.01	96.65	96.38
t-2	96.73	96.20	95.77	96.41	96.36	95.84
t-3	96.77	96.43	95.87	96.32	96.38	95.80
t-4	96.48	95.87	95.48	96.69	95.99	96.03
t-5	97.15	96.75	96.21	96.52	96.94	96.09
t-6	97.01	96.39	95.99	96.42	96.97	96.10
t-7	97.16	96.89	96.51	96.82	96.89	96.42
t-8	97.07	96.57	96.47	96.76	96.81	96.21
t-9	98.32	97.41	96.77	97.64	98.13	96.72
t-10	98.07	97.96	97.21	97.28	97.90	96.75
t-11	98.38	97.16	98.12	98.15	97.82	97.29
t-12	98.48	97.31	97.45	97.81	97.87	96.36
t-13	99.32	98.93	98.37	98.58	98.58	99.09
t-14	98.94	98.85	98.83	98.47	98.76	98.95
t-15	99.47	98.38	98.97	98.92	98.43	99.13
t-16	99.69	99.49	99.26	99.63	99.38	99.63
t-17	99.26	99.37	99.35	99.49	99.85	99.32
t-18	99.66	99.21	99.23	99.77	99.46	99.55

Table B.25: Mean Value of Sample used in Joint for the IBM system at 106

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.77	100.29	99.64	99.95	100.78	99.74
t-1	97.03	97.12	96.50	96.86	97.01	97.00
t-2	97.34	96.97	96.34	97.03	96.74	96.99
t-3	98.05	96.44	96.77	96.83	97.30	96.47
t-4	97.55	97.33	96.27	97.21	97.01	96.82
t-5	97.50	96.90	96.84	97.10	97.52	96.96
t-6	97.46	97.20	97.12	96.85	97.14	96.77
t-7	97.41	97.93	97.24	98.29	97.43	96.30
t-8	98.41	96.73	97.39	97.27	97.84	96.42
t-9	99.15	98.25	97.82	98.26	97.80	97.84
t-10	98.68	97.96	97.44	98.39	98.58	96.60
t-11	98.22	98.00	98.23	98.66	98.47	97.38
t-12	98.45	98.16	98.26	97.57	98.75	97.22
t-13	98.90	99.72	99.31	98.50	99.00	98.95
t-14	98.69	99.42	99.08	98.96	99.19	99.08
t-15	99.00	99.19	99.34	99.29	99.06	98.96
t-16	99.21	99.37	99.59	99.96	99.99	100.17
t-17	99.54	99.42	99.71	99.93	99.26	99.16
t-18	100.26	99.81	99.86	99.47	99.87	99.81

Table B.26: Mean Value of Sample used in Joint for the IBM system at 107

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.81	99.99	100.24	99.88	100.13	100.70
t-1	98.07	97.34	97.26	97.73	97.78	97.11
t-2	97.99	97.59	97.85	98.19	97.85	97.42
t-3	97.77	98.31	97.19	98.27	98.07	97.66
t-4	98.12	97.92	97.06	97.52	97.89	97.61
t-5	98.08	98.03	97.88	97.44	98.64	97.51
t-6	98.37	97.76	97.55	97.95	98.45	97.59
t-7	98.05	98.12	97.77	98.69	98.01	98.05
t-8	99.06	98.43	97.91	98.31	97.90	96.79
t-9	98.92	98.72	97.81	98.00	98.54	97.74
t-10	98.76	98.44	97.81	98.94	98.23	98.61
t-11	99.59	97.72	98.26	98.79	98.70	98.40
t-12	98.78	98.49	98.38	98.56	98.42	97.95
t-13	98.87	99.49	99.07	99.75	99.12	99.56
t-14	99.46	99.46	99.23	99.60	99.59	99.21
t-15	99.32	99.30	98.67	98.99	99.32	99.52
t-16	99.69	100.13	99.64	99.75	99.43	99.54
t-17	99.82	100.18	99.78	99.22	99.11	100.05
t-18	100.04	100.27	100.31	100.05	99.57	100.21

Table B.27: Mean Value of Sample used in Joint for the IBM system at 108

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.79	99.46	99.62	99.99	100.18	100.19
t-1	98.58	98.10	98.11	98.46	98.37	98.74
t-2	98.48	98.31	97.96	98.52	98.47	98.59
t-3	97.71	98.29	97.87	98.13	98.36	98.86
t-4	98.91	98.44	98.37	98.93	98.37	98.34
t-5	98.63	98.70	98.33	98.76	98.11	98.13
t-6	98.34	98.47	98.25	98.74	98.68	98.53
t-7	98.40	98.38	98.50	98.93	99.08	98.71
t-8	99.06	98.21	98.42	98.84	98.76	98.41
t-9	99.20	98.78	98.46	99.49	98.80	98.39
t-10	98.70	98.99	99.16	98.80	98.95	98.64
t-11	99.21	98.66	99.42	98.64	98.69	98.57
t-12	99.45	99.44	99.35	99.32	98.94	98.62
t-13	99.10	99.34	99.73	99.33	99.77	99.95
t-14	99.56	100.23	99.84	99.81	100.11	99.67
t-15	99.82	99.58	99.55	100.43	99.46	99.52
t-16	99.97	100.17	99.66	99.87	99.97	99.49
t-17	99.55	99.34	100.06	99.69	99.81	99.65
t-18	99.61	99.34	99.99	99.82	100.16	99.73

Table B.28: Mean Value of Sample used in Joint for the IBM system at 109

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.13	99.86	100.32	100.40	100.13	99.65
t-1	98.97	99.29	98.50	99.27	98.43	99.02
t-2	99.04	98.62	98.78	98.93	99.28	99.04
t-3	99.78	98.13	99.02	99.21	98.88	98.95
t-4	99.65	99.13	98.99	99.13	98.69	99.22
t-5	98.19	98.92	99.22	99.59	98.95	98.74
t-6	99.23	98.48	98.53	99.65	98.85	99.19
t-7	99.09	99.15	99.39	99.74	99.49	99.00
t-8	99.24	99.08	98.98	99.09	99.00	98.63
t-9	99.41	99.29	99.86	99.52	99.46	99.06
t-10	98.97	99.43	99.63	99.17	99.53	98.97
t-11	99.75	98.74	99.55	99.12	99.46	98.74
t-12	99.39	98.89	99.25	99.26	99.63	98.64
t-13	100.06	99.50	99.55	99.51	99.72	99.52
t-14	99.72	99.87	99.26	99.95	99.47	99.47
t-15	99.30	99.76	99.18	99.39	99.71	99.62
t-16	99.94	99.77	100.02	100.20	99.79	100.04
t-17	99.95	99.64	99.71	99.78	99.87	99.70
t-18	99.19	100.20	99.54	100.31	100.32	100.20

Table B.29: Mean Value of Sample used in Joint for the IBM system at 110

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.75	99.64	99.92	100.07	100.19	99.41
t-1	99.46	99.64	99.38	99.52	99.02	99.20
t-2	99.03	99.73	99.14	99.76	98.99	99.48
t-3	99.26	99.23	99.77	98.96	99.85	99.11
t-4	99.68	99.33	99.78	99.10	99.34	99.32
t-5	99.10	99.53	99.05	99.74	99.59	99.92
t-6	99.40	99.44	99.30	99.32	99.79	98.96
t-7	99.35	99.61	99.80	99.36	100.27	99.57
t-8	99.48	99.33	99.42	99.46	99.50	99.49
t-9	99.66	99.63	99.54	99.52	99.81	99.28
t-10	99.92	99.54	99.19	100.00	99.48	99.61
t-11	99.89	99.66	99.34	99.79	100.30	98.98
t-12	99.61	98.91	99.24	99.83	99.41	99.66
t-13	100.53	99.85	101.01	99.77	100.02	99.24
t-14	99.47	99.75	99.49	100.02	99.72	99.68
t-15	99.36	99.61	100.21	99.72	99.57	99.91
t-16	99.79	99.90	100.24	99.86	99.73	99.68
t-17	99.33	100.00	99.55	99.91	99.52	99.94
t-18	99.36	99.86	99.24	100.21	99.98	100.03

Table B.30: Mean Value of Sample used in Joint for the IBM system at 111

Period	**Prod 1**	**Prod 2**	**Prod 3**	**Prod 4**	**Prod 5**	**Prod 6**
t-0	100.05	100.31	100.14	100.29	99.98	99.87
t-1	100.13	98.90	99.14	99.96	99.42	99.67
t-2	99.96	99.39	99.85	99.83	99.39	99.63
t-3	100.33	99.88	99.42	99.97	99.88	99.62
t-4	99.25	99.51	99.51	99.81	100.57	100.17
t-5	99.57	99.28	100.00	100.19	99.36	99.20
t-6	100.02	99.84	99.84	99.66	99.96	99.84
t-7	99.33	99.94	100.40	99.47	99.92	99.09
t-8	99.68	99.62	99.91	100.14	99.78	99.80
t-9	99.94	99.85	100.10	99.75	99.07	99.60
t-10	100.73	100.04	100.28	100.35	99.68	99.82
t-11	99.92	100.44	100.76	100.30	99.78	99.99
t-12	99.64	100.46	100.03	100.06	99.60	99.95
t-13	99.77	99.93	99.58	99.69	100.20	99.92
t-14	99.90	100.34	100.07	99.92	99.72	99.65
t-15	99.91	100.28	99.90	99.75	99.85	99.89
t-16	100.61	100.09	100.39	100.06	100.61	99.77
t-17	99.89	100.42	100.07	100.09	100.05	99.42
t-18	99.57	99.81	100.32	100.33	99.99	99.80

Table B.31: Mean Value of Sample used in Joint for the IBM system at 112

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.30	100.00	100.39	99.71	99.49	100.03
t-1	99.93	100.09	99.40	100.28	100.35	100.01
t-2	99.64	100.58	99.93	99.72	99.66	100.03
t-3	100.02	101.00	99.59	99.46	99.90	99.47
t-4	99.49	100.15	99.91	100.24	100.28	99.97
t-5	99.72	100.24	100.48	100.21	100.11	99.65
t-6	99.93	100.47	99.92	100.07	99.81	100.30
t-7	100.29	100.00	100.04	100.58	100.75	99.48
t-8	99.91	100.06	99.92	99.48	99.77	99.49
t-9	99.82	100.00	99.07	100.57	99.41	99.95
t-10	100.61	99.68	99.75	99.81	100.05	99.33
t-11	99.74	99.98	100.03	99.48	100.17	99.97
t-12	100.00	99.75	100.19	99.90	100.81	99.45
t-13	100.03	99.95	99.87	100.44	99.74	100.36
t-14	99.78	99.67	100.18	99.11	99.86	100.04
t-15	100.16	100.18	100.22	99.86	99.94	100.19
t-16	99.25	100.03	99.64	100.36	100.16	100.08
t-17	100.00	99.97	100.45	99.87	99.46	99.60
t-18	100.06	100.02	100.67	99.85	100.24	100.15

Table B.32: Mean Value of Sample used in Joint for the IBM system at 113

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.11	100.23	100.48	100.01	100.11	100.64
t-1	99.83	99.86	99.89	100.20	100.08	100.10
t-2	99.95	100.51	99.75	100.16	100.15	99.87
t-3	99.71	99.84	99.94	100.63	100.34	100.06
t-4	100.22	100.09	100.36	99.70	100.10	100.18
t-5	99.89	100.11	99.74	100.01	99.94	100.41
t-6	99.63	100.20	100.29	99.70	100.53	100.31
t-7	100.20	100.05	99.56	99.66	99.92	100.33
t-8	100.39	100.31	100.16	100.09	100.01	99.59
t-9	100.51	100.62	99.64	99.84	99.77	99.41
t-10	100.31	99.84	99.79	100.55	99.82	100.09
t-11	100.37	99.84	100.23	100.47	100.06	99.81
t-12	99.55	99.78	100.09	99.26	100.18	100.44
t-13	100.45	100.20	99.36	100.01	99.95	100.12
t-14	99.82	99.90	99.93	99.91	99.72	99.73
t-15	99.63	99.15	100.17	100.22	100.55	99.48
t-16	99.83	99.36	100.18	99.72	99.68	100.42
t-17	100.30	99.86	100.03	100.07	100.15	100.12
t-18	99.97	100.09	100.28	100.18	99.84	100.29

Table B.33: Mean Value of Sample used in Joint for the IBM system at 114

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.96	100.21	100.54	100.00	100.62	99.87
t-1	99.68	100.29	100.04	100.36	100.26	99.84
t-2	99.83	99.67	99.73	100.06	100.01	100.34
t-3	100.10	100.44	99.87	100.27	100.02	100.11
t-4	99.64	99.92	100.03	99.92	100.46	100.32
t-5	99.77	100.32	100.17	99.45	100.04	100.34
t-6	99.72	100.34	99.83	99.64	100.69	99.63
t-7	99.53	99.68	99.98	100.26	100.06	99.76
t-8	100.39	99.85	99.49	99.53	99.84	100.51
t-9	99.50	100.27	99.47	100.36	100.20	99.85
t-10	100.25	100.14	100.06	99.94	99.51	100.04
t-11	100.25	100.88	99.62	99.72	99.91	100.16
t-12	100.23	99.79	100.42	100.26	99.52	100.24
t-13	100.16	99.97	100.24	99.50	100.04	100.35
t-14	99.58	99.81	100.61	100.41	100.12	100.64
t-15	100.05	100.26	100.35	99.67	99.80	100.18
t-16	100.09	99.82	99.86	100.12	100.06	100.13
t-17	100.24	99.55	99.65	100.69	99.87	99.60
t-18	100.45	99.28	100.16	99.70	100.18	100.02

Table B.34: Mean Value of Sample used in Joint for the IBM system at 115

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.93	99.71	99.96	100.10	99.91	99.76
t-1	100.10	99.52	100.27	99.68	99.90	99.70
t-2	100.26	99.61	100.03	99.91	100.02	100.11
t-3	99.41	100.12	100.60	99.64	100.11	99.62
t-4	99.82	99.53	99.80	99.37	100.12	100.44
t-5	99.33	100.11	100.04	100.11	99.77	100.21
t-6	100.22	100.08	100.28	99.68	100.52	100.17
t-7	100.34	100.04	100.19	99.93	100.44	100.12
t-8	99.94	99.31	99.87	99.58	99.86	99.81
t-9	99.96	100.17	99.95	99.51	99.54	99.78
t-10	99.93	100.35	100.11	100.74	100.15	99.97
t-11	99.96	100.14	99.92	99.09	99.38	99.80
t-12	100.14	99.76	99.81	99.86	99.68	99.99
t-13	100.18	100.18	99.97	100.11	100.04	100.20
t-14	99.69	99.90	100.32	99.98	99.60	99.94
t-15	99.99	100.17	99.98	99.69	99.70	100.19
t-16	100.01	99.92	100.14	100.10	99.80	100.42
t-17	99.52	100.20	99.72	100.24	100.21	99.37
t-18	100.19	99.92	100.01	99.84	100.28	100.15

Table B.35: Mean Value of Sample used in Joint for the IBM system - Unfiltered

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.98	99.97	100.05	99.81	99.90	100.00
t-1	93.04	93.26	92.21	91.38	94.95	93.76
t-2	93.17	93.23	92.11	91.28	95.02	94.04
t-3	93.16	93.14	92.18	91.10	95.37	93.68
t-4	93.21	93.12	91.92	91.03	95.27	93.95
t-5	93.72	93.97	93.30	96.10	96.32	94.09
t-6	94.40	94.17	93.22	96.43	96.45	94.40
t-7	95.17	95.23	94.28	97.37	97.07	94.26
t-8	95.31	95.27	94.46	97.56	97.52	94.14
t-9	95.42	95.63	94.57	97.37	97.55	94.09
t-10	95.63	95.26	94.70	97.86	98.08	94.22
t-11	95.84	95.41	97.62	97.65	97.64	94.59
t-12	95.31	95.25	97.66	97.74	97.89	93.93
t-13	99.30	99.31	99.24	99.06	98.57	98.61
t-14	98.98	99.10	99.18	98.68	98.83	98.82
t-15	98.94	99.04	99.07	98.89	99.11	99.08
t-16	99.33	99.22	99.01	98.99	99.08	99.18
t-17	99.63	99.23	98.59	98.94	98.92	99.35
t-18	99.45	99.02	99.04	98.95	99.32	99.25

Table B.36: Mean Value of Sample used in Alloc for the IBM system at 106

Period	**Prod 1**	**Prod 2**	**Prod 3**	**Prod 4**	**Prod 5**	**Prod 6**
t-0	100.01	99.78	99.88	99.99	100.19	100.23
t-1	95.06	93.88	92.71	93.00	95.60	93.91
t-2	94.71	93.84	92.80	93.67	95.61	93.55
t-3	94.74	93.57	92.69	93.35	95.80	93.64
t-4	95.02	93.93	92.74	93.10	95.96	93.72
t-5	96.27	94.55	93.63	96.72	97.17	93.79
t-6	96.15	95.05	93.90	97.23	97.22	93.97
t-7	96.52	95.22	94.56	97.56	97.60	93.79
t-8	96.54	95.63	94.31	96.93	97.57	94.11
t-9	96.89	95.62	94.72	97.42	97.48	93.70
t-10	96.93	95.84	94.72	97.41	97.68	94.04
t-11	96.69	95.51	97.80	97.68	97.41	94.31
t-12	96.71	95.90	97.36	97.85	97.38	93.67
t-13	99.40	99.56	98.55	98.40	98.25	99.22
t-14	99.44	99.70	98.46	98.59	98.73	99.83
t-15	99.30	99.42	98.43	98.40	98.44	99.01
t-16	99.41	99.63	98.38	98.92	98.44	99.43
t-17	99.63	99.65	98.35	98.76	98.62	98.83
t-18	99.76	99.58	98.89	98.32	98.51	99.52

Table B.37: Mean Value of Sample used in Alloc for the IBM system at 107

Period	**Prod 1**	**Prod 2**	**Prod 3**	**Prod 4**	**Prod 5**	**Prod 6**
t-0	99.86	99.80	99.86	100.14	100.05	99.90
t-1	94.21	94.43	94.39	93.76	96.19	94.58
t-2	94.51	94.62	94.43	93.84	96.37	94.69
t-3	94.52	94.44	94.04	93.52	96.06	94.70
t-4	94.39	94.63	94.32	93.93	96.24	94.71
t-5	95.25	95.18	94.82	97.33	96.86	94.63
t-6	95.48	95.53	95.50	97.64	97.21	95.00
t-7	95.82	96.01	95.22	97.65	97.54	94.92
t-8	95.75	95.90	95.15	97.80	97.44	95.02
t-9	95.80	96.08	95.64	97.99	97.58	95.08
t-10	95.75	95.82	95.47	98.13	97.53	94.96
t-11	95.85	96.10	97.97	97.88	97.97	95.28
t-12	95.90	96.12	98.13	97.73	97.59	94.68
t-13	99.39	99.44	98.56	98.51	98.69	99.48
t-14	99.55	99.83	98.48	98.87	98.57	99.61
t-15	99.44	99.34	98.60	98.33	98.54	99.63
t-16	100.04	100.15	98.72	98.76	98.41	99.58
t-17	99.60	99.41	98.55	98.33	98.72	99.96
t-18	99.76	99.48	98.30	98.36	98.97	99.60

Table B.38: Mean Value of Sample used in Alloc for the IBM system at 108

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.16	100.02	100.22	100.01	100.06	100.00
t-1	95.66	95.82	95.02	94.88	97.07	96.11
t-2	95.88	95.70	95.00	94.93	97.17	96.33
t-3	95.92	95.88	94.65	95.26	97.02	95.72
t-4	95.90	95.61	94.39	95.14	97.09	96.45
t-5	96.48	96.63	95.26	97.85	97.76	95.84
t-6	96.59	96.18	95.27	97.70	97.65	95.81
t-7	96.44	96.58	95.73	97.96	97.86	96.13
t-8	96.48	96.64	95.74	97.89	97.82	96.14
t-9	96.92	97.23	96.09	98.31	98.15	96.04
t-10	97.00	97.17	95.84	97.99	98.04	96.26
t-11	97.17	97.22	98.10	98.02	98.16	96.35
t-12	96.87	97.35	98.16	97.74	98.16	96.19
t-13	99.16	99.36	99.11	98.95	99.18	99.19
t-14	99.01	99.15	99.18	98.86	98.93	99.24
t-15	99.09	99.12	99.14	99.07	98.99	99.10
t-16	98.96	99.32	98.99	99.26	98.94	98.94
t-17	99.23	99.04	99.03	99.36	98.81	98.83
t-18	99.01	99.00	99.17	99.34	99.15	99.00

Table B.39: Mean Value of Sample used in Alloc for the IBM system at 109

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.09	100.37	99.86	100.13	100.05	100.27
t-1	96.15	96.23	95.22	93.98	97.29	96.81
t-2	95.84	96.27	95.38	94.08	97.48	97.06
t-3	96.26	95.78	95.23	93.87	97.77	97.23
t-4	95.90	95.98	95.45	93.78	97.35	97.06
t-5	96.67	96.68	96.10	98.13	98.16	96.88
t-6	96.81	96.89	96.22	98.19	98.30	97.13
t-7	97.17	97.16	96.31	98.37	98.41	97.22
t-8	96.98	97.36	96.72	98.43	98.44	97.26
t-9	97.06	97.24	96.57	98.66	98.39	97.11
t-10	97.26	97.10	96.37	98.74	98.64	97.00
t-11	96.99	97.22	98.84	98.44	98.27	97.16
t-12	97.11	97.17	98.68	98.67	98.39	97.17
t-13	99.42	99.65	99.38	99.25	99.30	99.10
t-14	99.56	99.89	99.51	99.09	99.20	99.56
t-15	99.30	99.70	99.37	99.02	99.26	99.34
t-16	99.46	99.49	99.34	99.38	99.43	99.61
t-17	99.69	99.17	99.32	99.33	99.14	99.71
t-18	99.35	99.50	99.40	99.47	99.44	99.47

Table B.40: Mean Value of Sample used in Alloc for the IBM system at 110

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.15	100.08	100.09	99.95	99.87	99.75
t-1	96.94	96.81	96.21	93.16	98.20	96.49
t-2	97.16	96.52	96.45	93.01	97.98	96.62
t-3	97.03	96.47	96.20	93.01	97.88	96.84
t-4	96.82	96.62	96.13	93.15	98.44	96.66
t-5	97.47	96.97	96.88	98.51	98.79	96.75
t-6	97.65	97.03	97.19	98.63	98.72	96.58
t-7	97.50	97.31	97.15	98.99	98.51	96.76
t-8	97.64	96.87	96.94	98.73	98.58	96.38
t-9	97.38	96.99	97.03	99.18	98.75	96.79
t-10	97.39	97.15	97.04	98.66	98.68	96.55
t-11	97.90	97.56	99.03	99.08	98.93	96.56
t-12	97.72	96.97	98.77	98.91	99.06	96.54
t-13	99.50	99.53	99.76	99.27	99.61	99.60
t-14	99.25	99.56	99.33	99.53	99.39	99.65
t-15	99.50	99.57	99.32	99.54	99.50	99.17
t-16	99.70	99.51	99.60	99.41	99.66	100.05
t-17	99.37	99.60	99.48	99.63	99.65	99.43
t-18	99.44	99.25	99.49	99.64	99.59	99.81

Table B.41: Mean Value of Sample used in Alloc for the IBM system at 111

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	100.07	100.07	100.01	100.07	99.70	99.98
t-1	96.57	97.11	96.74	95.15	98.69	97.00
t-2	97.37	96.82	96.69	94.99	98.54	97.07
t-3	96.62	96.88	96.96	95.05	98.77	97.08
t-4	97.02	97.02	96.75	94.91	98.90	96.93
t-5	97.31	97.53	97.20	99.18	98.82	97.15
t-6	97.25	97.66	97.32	98.94	98.94	97.23
t-7	97.36	97.61	97.43	99.27	99.55	97.26
t-8	97.38	97.83	97.58	99.40	99.12	96.95
t-9	97.43	97.63	97.73	98.89	99.19	97.08
t-10	97.57	97.57	97.34	99.61	99.44	97.19
t-11	97.54	97.87	99.22	99.29	99.13	97.29
t-12	97.65	97.52	99.34	98.99	99.26	97.40
t-13	99.28	99.24	99.41	100.00	99.75	99.31
t-14	99.13	99.40	100.00	99.34	99.66	99.48
t-15	99.49	99.65	99.47	100.23	99.64	99.33
t-16	99.39	99.72	99.82	99.61	99.67	99.52
t-17	99.62	99.54	99.88	99.65	99.79	99.44
t-18	99.53	99.46	99.94	99.52	99.66	99.75

Table B.42: Mean Value of Sample used in Alloc for the IBM system at 112

Period	**Prod 1**	**Prod 2**	**Prod 3**	**Prod 4**	**Prod 5**	**Prod 6**
t-0	99.92	100.11	99.99	99.99	99.65	100.08
t-1	97.10	97.22	97.74	96.87	99.06	97.93
t-2	97.16	96.78	97.55	97.24	98.89	98.10
t-3	97.32	97.25	97.69	97.59	98.55	97.71
t-4	97.48	97.26	97.70	96.97	98.91	97.75
t-5	97.17	97.30	98.26	99.11	98.78	97.77
t-6	97.43	97.34	98.00	99.21	99.28	97.93
t-7	97.22	97.80	97.60	99.37	99.48	97.91
t-8	97.46	97.45	97.85	99.23	99.42	98.21
t-9	97.58	97.37	98.06	99.27	99.21	97.71
t-10	97.07	97.50	97.53	99.52	99.31	98.16
t-11	97.40	97.31	99.43	99.41	99.56	97.54
t-12	97.60	97.66	99.21	99.17	99.13	97.85
t-13	99.84	100.09	99.44	99.93	99.77	99.97
t-14	99.87	100.11	99.79	99.80	99.53	99.60
t-15	99.58	100.00	99.48	99.48	99.79	100.07
t-16	99.96	99.76	99.90	99.73	99.65	99.90
t-17	99.52	99.80	99.39	99.57	99.41	99.54
t-18	100.12	99.79	99.64	99.53	99.78	99.98

Table B.43: Mean Value of Sample used in Alloc for the IBM system at 113

Period	**Prod 1**	**Prod 2**	**Prod 3**	**Prod 4**	**Prod 5**	**Prod 6**
t-0	99.73	99.72	100.17	99.99	99.79	100.00
t-1	97.59	98.12	97.97	98.66	99.47	97.69
t-2	97.50	97.78	97.84	98.76	99.37	98.16
t-3	97.93	97.52	97.87	98.82	99.20	97.80
t-4	97.87	97.88	97.69	98.61	99.32	98.04
t-5	97.62	98.13	98.03	99.56	99.41	98.34
t-6	97.97	98.03	97.88	99.76	99.39	98.08
t-7	98.29	98.32	98.10	99.40	99.57	98.02
t-8	97.69	98.03	97.89	99.85	99.40	98.23
t-9	97.82	98.04	97.91	99.28	99.79	98.28
t-10	97.71	98.20	98.48	99.51	99.67	98.08
t-11	97.53	98.10	99.44	99.89	99.82	98.01
t-12	97.71	97.97	99.70	99.56	99.57	97.99
t-13	99.57	99.88	99.65	99.73	99.60	99.93
t-14	99.86	100.22	99.80	100.12	99.80	99.88
t-15	99.78	99.96	99.69	99.95	99.85	99.88
t-16	99.86	99.99	99.86	99.67	99.78	99.90
t-17	100.15	99.59	99.54	100.04	99.93	100.16
t-18	99.77	99.81	99.85	100.02	99.29	99.86

Table B.44: Mean Value of Sample used in Alloc for the IBM system at 114

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.58	99.97	99.91	99.97	99.86	100.04
t-1	98.75	97.99	98.49	99.42	99.85	98.40
t-2	98.55	98.12	99.11	99.39	99.52	98.60
t-3	98.84	97.82	98.49	99.58	99.75	98.68
t-4	98.68	98.04	98.87	99.48	99.56	98.74
t-5	98.94	97.86	98.59	99.70	99.68	98.60
t-6	98.35	98.05	98.48	99.63	99.63	98.55
t-7	98.59	98.03	98.60	99.36	99.78	98.63
t-8	98.55	98.30	98.53	99.79	99.55	98.48
t-9	98.46	97.84	98.65	99.86	99.42	98.93
t-10	98.58	97.89	98.67	99.47	100.03	98.67
t-11	98.67	98.08	99.55	99.89	99.70	98.67
t-12	98.95	97.88	99.37	99.90	99.78	98.75
t-13	99.66	100.16	99.99	99.78	99.92	99.94
t-14	100.00	100.08	99.79	100.38	99.48	99.75
t-15	100.15	100.03	99.72	99.43	99.88	100.07
t-16	100.05	100.06	99.92	99.93	100.09	99.97
t-17	100.20	99.80	99.73	99.84	100.10	100.20
t-18	99.76	100.08	99.71	99.82	99.51	100.05

Table B.45: Mean Value of Sample used in Alloc for the IBM system at 115

Period	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6
t-0	99.99	100.17	100.16	100.05	100.01	100.10
t-1	100.26	100.04	99.71	99.69	100.26	100.06
t-2	99.85	100.13	99.87	99.75	100.16	99.73
t-3	99.78	99.86	99.47	100.16	100.17	99.58
t-4	100.23	100.14	99.76	99.92	100.00	99.82
t-5	99.86	99.94	99.86	100.10	100.34	99.83
t-6	99.72	100.00	100.04	100.20	100.12	99.90
t-7	100.11	100.21	100.30	99.85	99.90	99.66
t-8	100.13	100.13	100.05	99.91	100.00	100.19
t-9	100.24	100.03	99.93	99.98	99.98	99.81
t-10	100.39	99.96	100.01	99.80	100.03	100.15
t-11	99.90	100.16	100.16	99.91	99.85	100.44
t-12	100.09	99.64	100.20	100.13	100.25	99.82
t-13	99.76	99.74	100.17	100.19	100.03	100.13
t-14	100.49	100.15	99.90	100.16	100.05	99.74
t-15	99.76	100.03	99.94	99.88	99.72	100.02
t-16	99.76	100.07	99.96	99.98	100.27	100.15
t-17	99.79	100.26	99.91	99.60	99.95	99.74
t-18	100.07	100.12	99.90	100.17	100.03	99.85

Table B.46: Mean Value of Sample used in Alloc for the IBM system - Unfiltered

Appendix C

Matlab code sample

C.1 Realization Generation

C.2 Minimum Budget Calculation

C.3 D_i^k Calculation

Listing C.1: Realization Generation

```
classdef AtoNorInfo < AtoInfo
    properties(Constant,Hidden)
        %BOM Related Values
        %Stopping criteria
        MAXPASS = 10;
        %Component related values
    end

    methods
        function inst = AtoNorInfo(cons)
            if nargin ==0
                cons = [];
            end
            inst = inst@AtoInfo(cons);
        end
        % generate one or more realization demand for the product lead. The output is
        % [realSize*(maxLeadTime+2)][numProducts]. The demand at time t is located at
        % the begin of the matrix.
        function rsl = genreal(obj,nReal)
            if nargin<2
```

```
                nReal=25;
            end
            dem = randn((obj.maxLt+1)*nReal,obj.nProds).*repmat(obj.sdDemProdsArr',(obj.maxLt+1)*
                 nReal,1)+repmat(obj.exDemProdsArr',(obj.maxLt+1)*nReal,1);
            temp = dem<0;
            numPass = 0;
            while numPass<MAXPASS && sum(sum(temp))>0
                products1 = randn((obj.maxLt+1)*nReal,obj.nProds).*repmat(obj.sdDemProdsArr',(obj.
                     maxLt+1)*nReal,1)+repmat(obj.exDemProdsArr',(obj.maxLt+1)*nReal,1);
                dem = dem.*~temp+products1.*temp;
                temp = dem<0;
                numPass = numPass + 1;
            end
            if numPass == MAXPASS
                dem = dem.*~temp;
            end
            rsl = round(dem);
        end
    end
end
```

Listing C.2: Minimum Budget Calculation

```
% it assume the data is just one sample
function B = getMinBudg(ato,data)
nReal = ato.getNumReal(data);
tB = zeros(nReal,ato.nComps);
for i=1:nReal
    tprodDem=ato.getReal(data,1,i);
    dik=AtoSimulation.computDik(ato,tprodDem*ato.bomMat);
    tB(i,:) = dik(1,:);
end
B = max(tB,[],1)*ato.costCompsArr;
end
```

Listing C.3: D_i^k Calculation

```
% caculate D_ik
function Dik = computDik(ato,compDem)
% The demand matrix is in [t-3,t-2,t-1... format
Dik=flipud(compDem(1:end-1,:));
% we don't consider any demand exceed L_i
Dik=cumsum(Dik,1);
for i=1:ato.nComps
    Dik(1:ato.ltCompsArr(i),i)=wrev(Dik(1:ato.ltCompsArr(i),i));
    Dik(ato.ltCompsArr(i)+1:end,i)=0;
end
end
```

Bibliography

[1] Narendra Agrawal and Morris A Cohen. Optimal material control in an assembly system with component commonality. *Naval Research Logistics*, 48(5):409–429, 2001.

[2] Yalçin Akçay. *Three essays on resource allocation problems: Inventory management in assemble-to-order systems and online assignment of flexible resources.* PhD thesis, Pennsylvania State University, 2002.

[3] Yalçin Akçay and Susan H. Xu. Joint inventory replenishment and component allocation optimization in an assemble-to-order system. *Management Science*, 50(1):99–116, 2004.

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

[5] Kenneth R. Baker, Michael J. Magazine, and Henry L. W. Nuttle. The effect of commonality on safety stock in a simple inventory model. *Management Science*, 32(8):982–988, 1986.

[6] John R. Birge and François Louveaux. *Introduction to Stochastic Programming.* Springer, 2011.

[7] Robert N Boute. Impact of replenishment rules with endogenous lead times on supply chain performance. *4OR*, 5(3):261–264, 2007.

[8] Feng Cheng, Markus Ettl, Grace Lin, and David D Yao. Inventory-service optimization in configure-to-order systems. *Manufacturing & Service Operations Management*, 4(2):114–132, 2002.

[9] Antoine Deza, Kai Huang, Hongfeng Liang, and Xiao Jiao Wang. On component commonality for periodic review assemble-to-order systems. *Annals of Operations Research*, 253(1), 2017.

[10] Antoine Deza, Kai Huang, Hongfeng Liang, and Xiao Jiao Wang. On inventory allocation for periodic review assemble-to-order systems. *AdvOL-Report 2017/4, McMaster University*, 2017.

[11] Mustafa K. Doğru, Martin I. Reiman, and Qiong Wang. A stochastic programming based inventory policy for assemble-to-order systems with application to the W model. *Operations Research*, 58(4-part-1):849–864, 2010.

[12] Amit Eynan and Meir J. Rosenblatt. Component commonality effects on inventory costs. *IIE Transactions*, 28(2):93–104, 1996.

[13] Duncan K.H. Fong, Haizhen Fu, and Zhaolin Li. Efficiency in shortage reduction when using a more expensive common component. *Computers & Operations Research*, 31(1):123–138, 2004.

[14] Yigal Gerchak and Mordechai Henig. An inventory model with component commonality. *Operations research letters*, 5(3):157–160, 1986.

[15] Yigal Gerchak, Michael J. Magazine, and Gamble A. Bruce. Component commonality with service level reqirements. *Management Science*, 34(6):753–760, 1988.

[16] Paul Glasserman and Yashan Wang. Leadtime-inventory trade-offs in assemble-to-order systems. *Operations Research*, 46(6): 858–871, 1998.

[17] Mark Hillier. Component commonality in multiple-period, assemble-to-order systems. *IIE Transactions*, 32(8):755–766, 2000.

[18] Mark S. Hillier. Component commonality in a multiple-period inventory model with service level constraints. *International Journal of Production Research*, 37(12):2665–2683, 1999.

[19] Kai Huang. Benchmarking non-first-come-first-served component allocation in an assemble-to-order system. *Annals of Operations Research*, 223(1):217–237, 2014.

[20] Kai Huang and Ton de Kok. Optimal FCFS allocation rules for periodic-review assemble-to-order systems. *Naval Research Logistics*, 62(2):158–169, 2015.

[21] Henrik Jönsson and Edward A. Silver. Optimal and heuristic solutions for a simple common component inventory problem. *Engineering Costs and Production Economics*, 16(4):257–267, 1989.

[22] Henrik Jönsson, Kurt Jörnsten, and Edward A Silver. Application of the scenario aggregation approach to a two-stage, stochastic, common component, inventory problem with a budget constraint. *European Journal of Operational Research*, 68(2):196–211, 1993.

[23] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[24] Yingdong Lu and Jing-Sheng Song. Order-based cost optimization in assemble-to-order systems. *Operations Research*, 53(1): 151–169, 2005.

[25] Prakash Mirchandani and Ajay K. Mishra. Component commonality: Models with product-specific service constraints. *Production and Operations Management*, 11(2):199–215, 2002.

[26] Prakash Mirchandani and Ajay K. Mishra. Component commonality: Models with product-specific service constraints*. *Production and Operations Management*, 11(2):199–215, 2002.

[27] Alp Muharremoglu and Nan Yang. Inventory management with an exogenous supply process. *Operations research*, 58(1):111–129, 2010.

[28] Sigrid Lise Nonås. Finding and identifying optimal inventory levels for systems with common components. *European Journal of Operational Research*, 193(1):98–119, 2009.

[29] Jing-Sheng Song. On the order fill rate in a multi-item, base-stock inventory system. *Operations Research*, 46(6):831–845, 1998.

[30] Jing-Sheng Song and Yao Zhao. The value of component commonality in a dynamic inventory system with lead times. *Manufacturing and Service Operations Management*, 11(3):493–508, 2009.

[31] Jing-Sheng Song and Paul Zipkin. Supply chain operations: Assemble-to-order systems. *Handbooks in Operations Research and Management Science*, 11:561–596, 2003.

[32] Jayashankar M Swaminathan and Sridhar R Tayur. Managing broader product lines through delayed differentiation using vanilla boxes. *Management Science*, 44(12-Part-2):S161–S172, 1998.

[33] Ulrich W Thonemann and Margaret L Brandeau. Optimal commonality in component design. *Operations Research*, 48(1):1–19, 2000.

[34] Willem van Jaarsveld and Alan Scheller-Wolf. Optimization of industrial-scale assemble-to-order systems. *INFORMS Journal on Computing*, 27(3):544–560, 2015.

[35] Xiayang Wang. Optimization of a 2-stage series inventory system with stochastic demand and endogenous lead time uncertainty. In *Service Operations, Logistics and Informatics, 2009. SOLI'09. IEEE/INFORMS International Conference on*, pages 508–512. IEEE, 2009.

[36] Alex X. Zhang. Demand fulfillment rates in an assemble-to-order system with multiple products and dependent demands. *Production and Operations Management*, 6(3):309–324, 1997.

[37] Yao Zhao. Evaluation and optimization of installation base-stock policies in supply chains with compound poisson demand. *Operations research*, 56(2):437–452, 2008.