HEC Montréal

Value of Stochastic Information in Material Requirements Planning (MRP) under Demand Uncertainty

by

Shuanghao Shi

Master of Science in Administration
(Global Supply Chain Management)

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science (M.Sc.)

December 2017

Abstract

This study examines the value of stochastic information in Material Requirements Planning (MRP) under demand uncertainty. Two MRP models are developed to identify how stochastic programming can address the demand uncertainty issue in a MRP environment. The first model is a MRP optimization model with expected demand. The second model is a two-stage stochastic model that is extended from the first model to deal with uncertain demand. In this two-stage problem, the production batches are determined in the first stage and remain fixed. The production quantity is based on the number of batches determined in the first stage. In the second stage, the demand is realized and unmet demand is assumed to be lost. In the deterministic model, safety stock is predetermined and enforced to deal with uncertainty in demand, whereas there is no imposed safety stock in the stochastic model and the demand uncertainty is represented by a set of demand scenarios. In the computational experiments, 36 instances are generated and tested. We use Gurobi version 7.5 as the optimization solver, whereas the models and experiments are coded in Python 2.1.3. Computational results are presented to compare the performance of both the deterministic model and the stochastic model. In this thesis, the descriptive statistics about the optimal solutions based on the deterministic model with perfect demand information and its average objective value across all the scenarios, the objective values of stochastic model, and the expected objective values of deterministic model are computed. Based on these solution results, we analyse the sensitivity of deterministic solutions with demand uncertainty, the expected value with perfect information (EVPI), and the value of stochastic solutions (VSS). The results show that the optimal solutions from the deterministic optimization model are sensitive to the change of the demand. In addition, our stochastic model can provide average potential savings of 5% in the MRP environment under demand uncertainty.

Key words:

MRP, demand uncertainty, stochastic programming, expected value with perfect information (EVPI), value of stochastic solutions (VSS)

Résumé

Cette étude examine la valeur de l'information stochastique dans la planification des besoins matières (PBM) sous l'incertitude de la demande. Deux modèles de la PBM sont développés pour identifier comment la programmation stochastique peut répondre au problème d'incertitude de la demande dans un environnement de PBM. Le premier modèle est un modèle d'optimisation de la PBM avec demande espérée. Le deuxième modèle est un modèle stochastique en deux étapes qui s'étend du premier modèle pour répondre à une demande incertaine. Dans ce problème en deux étapes, le nombre de lots de production est déterminé dans la première étape et reste fixe. La quantité de production est basée sur le nombre de lots déterminé dans la première étape. Dans la deuxième étape, la demande est réalisée et la demande non satisfaite est suposée être perdue. Dans le modèle déterministe, le stock de sécurité est prédéterminé et appliqué pour faire face à l'incertitude de la demande, alors qu'il n'y a pas de stock de sécurité imposé dans le modèle stochastique et l'incertitude de la demande est représentée par un ensemble de scénarios de demande. Dans les expériences de calcul, 36 exemples sont générés et testés. Nous utilisons la version 7.5 de Gurobi comme solveur, alors que les modèles et les expériences sont codés en Python 2.1.3. Les résultats computationnels sont présentés pour comparer les performances du modèle déterministe et du modèle stochastique. Dans cette thèse, les statistiques descriptives sur les solutions optimales à partir du modèle déterministe avec les informations de demande parfaites et sa valeur objective moyenne de tous les scénarios, les valeurs objectives du modèle stochastique et les valeurs objectives espérées du modèle déterministe sont calculées. Sur la base de ces résultats de solution, nous analysons la sensibilité des solutions déterministes sous l'incertitude de la demande, la valeur attendue avec les information parfaites (EVPI), et la valeur des solutions stochastiques (VSS). Les résultats montrent que les solutions optimales du modèle d'optimisation déterministe sont sensibles à la variation de la demande. En outre, notre modèle stochastique peut fournir une économie potentielle moyenne de 5% dans l'environnement PBM en cas d'incertitude de la demande.

Mots clés:

PBM, incertitude de la demande, programmation stochastique, valeur attendue avec information parfaite (EVPI), valeur des soutions stochastiques (VSS)

Table of Contents

List of figures

List of tables

Acknowledgement

I would like to express my deepest gratitude to Professor Yossiri Adulyasak for his patient and constant guidance and encouragement. Without his help, I would never finish this thesis in this shape. Thanking him for all his time helping me build the model, test the model, and improve my thesis writing, even after his work time or during the weekend. I appreciate all the ideas, knowledge, help, and inspiration that I got from my supervisor. I enjoyed all our meetings and discussions during this thesis writing period.

Heartfelt thanks to my beloved family (my father, my mother and my sister) for their unconditional support to all my life during last 27 years. Without their love and encouragement, I would never be able to chase my dream without any burdens. They are definitely my most powerful driven force to achieve my dreams before, now and in the future.

I owe my sincere appreciation to all my friends in Canada and China. Thank you for your listening and understanding when I'm down. Thank you for enabling me to share my sorrow and happiness with you. My life would never be so wonderful without my loving friends. Special thanks to all my friends, professors and fellow classmates at HEC Montréal who gave me unforgettable memories and experience during my master study.

Chapter 1- Introduction

Material Requirements Planning (MRP) plays an important role in the production planning and control system in manufacturing processes. Arnold et al. (2012) propose that the main objective of MRP is to determine the requirements. Specifically, it is to create a production plan to ensure that the right materials in the right quantities will be available at the right time to meet the Master Production Schedule (MPS). MRP is a rule-based approach that works in a deterministic framework. According to Murthy and Ma (1991), there are three major inputs of MRP. The first one is product structure records. The product structure is represented by the bill of materials (BOM). The second input is the inventory status records which includes the on-hand balance, open orders, lot sizes, and lead times. The last input is the MPS which indicates the quantity and timing of end products. This deterministic demand of the end product is developed from forecast, which is typically created from historical sales data.

In a stochastic environment, actual demand could be very different from the forecast. In the MRP environment, the plan generated by the traditional MRP approach is known to be very sensitive to even a small change in demand. This main drawback of MRP is known as “nervousness”. According to Bregni et al. (2011), even a small change in the upper level of MRP can cause significant changes in the MRP plans. Consequently, this demand uncertainty can result in significant extra rescheduling cost, penalty cost, lost sale, loss the goodwill, etc.

Conventionally, safety stock, safety lead time and overproduction are typical approaches to create a buffer for negative impacts from the demand uncertainty in MRP. However, these measures usually result in extra inventory, and consequently result in extra inventory holding cost. Some research proposes that a better performance can be achieved by the mathematical optimization. Normally the traditional optimization models in the MRP environment deal with optimal decisions, such as the optimal over planning, optimal safety stock. However, as it is mentioned in the research from Yu & Li (2000), the main challenge the supply chain management is facing is the uncertainty of the future. Many optimization problems in logistics consider the uncertainty in a form of scenarios where the parameters have different probability of occurrence. When facing these uncertainties, these traditional methods are often not adequate to deal with different scenarios.

Stochastic programming is a method for optimization problems under uncertainty. It typically makes use of a set of scenarios and probabilities associated with each scenario. In the context of MRP, relatively few studies focused on stochastic programming in the MRP environment. Within the study on stochastic programming, most of them addressed the topics such as supply chain network design, healthcare and disaster relief. Little attention was given to the stochastic programming in MRP environment. Therefore, this thesis will supplement the existing literature by providing the insights on the value of the stochastic information in MRP under demand uncertainty

In response to the problem mentioned above, in this thesis, we explore a deterministic MRP optimization model and a two-stage stochastic model. By comparing the results from two models, we can get valuable insights on the performance of the decisions. Our work in this thesis is different from the other research in several aspects. The main contributions of this thesis can be summarized as follows:

1. We developed two MRP models: the deterministic model and the stochastic model
2. We performed computational experiments and illustrated the value of the stochastic programming model over the deterministic model in the MRP environment
3. Based on the discussion of sensitivity of optimal solution, we provided the insight about the cost effectiveness of the demand change in the MRP environment
4. Based on the discussion of *Expected Value of Perfect Information (EVPI)* and the *Value of Stochastic Solution (VSS)*, we provided the insight about how the stochastic programming can provide the cost related information in the MRP environment

To identify how the stochastic programming can address the demand uncertainty issue in an MRP environment to create a more robust production plan, we study a deterministic model and a stochastic model of the MRP. In this thesis, we use 1000 demand scenarios and 36 instances. The demand data set was created by a simulation. The demand follows a normal distribution with random probability. As in the traditional MRP, the demand is assumed to be dynamic, i.e., different demands in different periods have different average values and standard deviation. This data set is used in the computational experiments as the input of deterministic and stochastic models.

Afterwards, based on the computational results, a comparison between results from deterministic model and results from stochastic model will be presented. Furthermore, we perform the sensitivity analysis of solutions, as well as evaluate the EVPI and VSS in this thesis.

The rest of this thesis is organized as follows: Chapter 2 is the literature review; Chapter 3 presents the deterministic model and the stochastic model; In chapter 4, the computational results and the analysis are provided; Chapter 5 is the conclusion and the future research directions.

Chapter 2 - Literature Review

In this chapter, first, we generally talk about the stochastic programming. Second, we demonstrate the difference between MRP and ROP. In addition, we give an overview of MRP assumption and calculation. Next, we describe the MRP main components that consist of forecast, planning logic, and BOM. Then the last two subsections are general discussion about how to deal with uncertainty and production optimization models under uncertainty.

2.1 Stochastic programming

To deal with both optimization problems and uncertainties, stochastic programming is a good framework. It is a mathematical programming in which the stochastic elements are incorporated. According to the OR notes by Beasley, the coefficients of deterministic mathematical programming are known. In contrast, the coefficients of stochastic mathematical programming are unknown numbers. Instead, the coefficients of stochastic mathematical programming have probabilistic attributes. In our thesis, stochastic programming is involved. Therefore, we generally talk about how the stochastic programming is studied in the literature.

Batun (2012) defined stochastic programming as " a branch of mathematical programming that provides a framework for modeling and solving optimization problems with random parameters". It is claimed in their research that the two-stage stochastic model is the most widely studied model. A decision is made with imperfect information at the first stage. Based on decisions from the first stage, at the second stage, we make decisions after the realization of random scenarios. Additionally, lots of decisions are made based on a sequential period. When uncertainty is involved in multiple stages, a multi-stage stochastic program can be formulated.

Large quantity of research deals with stochastic programming. Batun (2012) developed stochastic models for solving the scheduling of multiple operating room problems during the surgery. In this research, first, a two-stage stochastic program was formulated for the problem of operating room pooling and parallel surgery processing under uncertainty. Afterwards, a three-stage stochastic programming model was developed for the operating room rescheduling.

Bozorgi-Amiri et al. (2013) developed a stochastic model for the disaster relief logistics. In their search, demand is not the only uncertain parameter. Uncertain supplies, cost of procurement and transportation are also considered. It is a multi-objective model, which not only minimizes the sum of expected value and the variance of the total cost, but also maximizes satisfaction levels of affected areas. The stochastic programming is also used for the supply chain network design problem. Santoso et al. (2005) proposed a stochastic programming approach for the supply chain network design. In their model, a sampling strategy was integrated, which helps to compute a larger scale of network design problems with a large quantity of scenarios. EI-Sayed et al (2010) also did research on the topic of network design. In their research, a multi-stage stochastic programming model was developed to maximize the total expected profit. Chouinard (2008) studied a stochastic programming model for designing supply loops. It integrates the reverse logistics into the current supply chain.

In the research from Chang et al. (2007), stochastic programming models were used for the flood emergency logistics problems. These two models were developed as tools for government to determine the rescue resource distribution system when the urban flood happens. The structure of rescue organizations, the location of rescue resources warehouses, and distributions of rescue resources are used as the decision variables in their models.

2.2 The MRP logic

The study of MRP has received a large amount of attention and the MRP related articles are vast. MRP is a system which is used to manage the production process. A schedule is established by MRP to show the components needed at each stage of assembly. And based on the lead time, the time when these components will be required is calculated (Arnold, Chapman, & Clive, 2012). In brief, the main objective of the MRP is to create a production plan to ensure that the right materials in the right quantities will be available at the right time to satisfy customers' demand. Arnold et al. (2012) classified the inputs of MRP as master production schedule (MPS), bill of materials (BOM), and inventory records. The outputs of MRP can be classified as time-phased manufacturing and purchase orders for raw materials and components.

2.2.1 The difference between MRP and ROP

One major difference between these two systems is that the reorder point (ROP) mainly deals with one product in a decentralized fashion while the MRP deals with the entire production process of an end-product in a centralized fashion. We discuss in this subsection the differences between MRP and ROP.

MRP and ROP are two well-known inventory control methods in production planning and inventory control systems. Bregman (1992) used an analytical framework to compare the difference between the MRP and the ROP. In this paper, the author pointed out that, in MRP, the requirements for each component are calculated from the previous stage – Master Production Schedule (MPS); whereas in the ROP system, the amounts of each component required are calculated from theses components' previous quantity. Both MRP and ROP deal with the forecasting. When we use the MRP, we plan for the future. We need to forecast the quantity of the whole products to be produced or manufactured. From the forecast demand, we then calculate the quantity of each component we require based on the MRP logic. However, the ROP works in a different way from the MRP does. When we use ROP, we look back to the historical data and compute the average demand and the standard deviation. This information is used to calculate the inventory control parameters that control the production plan. Bregman (1992) claims that the information from MRP better captures the actual demand for each component than ROP does as the forecasts of ROP are from the past data and the forecasts of MRP are from the MPS. Therefore, many companies have changed their planning systems from the ROP system to MRP system because the benefits yield from this conversion outweighs the large change-over costs for most of the companies. In this research, the author built an analytical framework to determine in which environment the MRP system can yield the highest benefits. The effects of this framework were measured from two aspects: a temporal comparison of ROP and MRP systems, and the comparison of the cost of components forecasting by using the ROP system and the MRP system. The author concluded that, when the production quantities, the stages between an item and MPS, and the gross requirements increase, both temporal penalty and the variance associated with the components stage requirements increase.

Jonsson & Mattsson (2006) compared the MRP system and the ROP system in a different manner. In their research, authors focused on the evaluation of the performances of different production planning systems to control the material flow in various types of inventory. They studied five different methods: The ROP, MRP, Kanban, the fixed order interval method and the run-out time planning. As it is mentioned by the authors, the performance of production planning is partly decided by the correct and appropiate methods they use. To have a better performance for each method, parameters, such as lead times, safety stocks, batch sizes, should be dertermined analytically. The authors designed and analyzed the study in two ways. In order to achieve high performance, firstly, authors separated three different manufacturing environments based on different environment types: inventory of purchased items, inventory of semi-finished items and inventory in distribution operations. Secondly, the author classified the modes of applying methods into two types. The first type is the planning parameters which require analytical frameworks. Another type is the way of using methods which includes review frequency, planning frequency, order modification and automatic re-planning. The data was from a web-based survey from 153 manufacturing and 53 distribution companies. Through the analysis of the data, it is concluded that the MRP is most frequently used in the manufacturing companies whereas ROP is the second frequent undes method in the manufacturing companies. However, when it comes to the distributions operations, ROP and ROP-related methods are the main methods used. Also, MRP is more dependent on the accuracy of planning information than ROP. In general, MRP has a better performance in controlling the semi-finished goods. In addition, the critical parameters to achieve higher performance are different as well between MRP and ROP. For ROP method, the determination of re-order point and the safety stocks are the key parameters to achieve high performance. For MRP method, the lead times and safety stocks are critical to achieve better performance (Jonsson & Mattsson, 2006).

Segerstedt (2006) made a comparison between ROP and MRP from the perspective of forecast, order trigger, lead-times, reshceduling, time phasing, forecasts for suppliers, workloads and bill of materials. We forecast each item individually in ROP method. In the MRP, the demand quantities are obtained from the MPS. When it comes to the order trigger for the replenishment, the order is triggered in ROP when the sum of on-hand inventory and scheduling receipts are less than the reorder point. In MRP, the starting date for a new order is planned in advance. Lastly, for ROP, there are no automatic rescheduling, no forecasts for suppliers, no workloads and no bill of materials. On the contrary, MRP has all these four charactrristics (Segerstedt, 2006).

We can also find the comparison between ROP and MRP in the research from Jonsson & Mattsson (2006). In their research, the author described the reasons for choosing different material planning methods. They claimed that MRP is the most commonly used material planning method. The position of MRP was even enhanced since 1990s. Most of the ROP users also use MRP. However, few companies use MRP to generate demand for the ROP (Jonsson & Mattsson, 2006).

Suwanruji & Enns (2006) compared the performance of ROP and MRP by using the simulation method. The type of demand and the capacity constraints are included as the additional experimental factors. Suwanruji & Enns (2006) concluded that, with seasonal demand, MRP performed better than ROP. Without the seasonal demand, the performance of these two methods depends on the capacity constraints. With capacity constraints, MRP had a better performance than the ROP. However, when it is without capacity constraints, the result is reversed.

2.2.2 MRP assumption and calculation

The input of the MRP system includes the bill of materials, item master data, demand data and the inventory data. As we mentioned earlier, MRP is a computer-based planning system which is driven by the MPS. According the research from Yeung, Wong, & Ma (1998), the demand management process is established in the phase of MPS. The schedule for the end product is created in this phase. There is one MPS in corresponding to one product. More specifically, MPS states the quantity of the end product to be produced in the future. This quantity in MPS becomes the gross requirement in the MRP system. In another word, the goal of MPS is to create a production master schedule for the end product based on the forecast of the future needs (Dolgui & Prodhon, 2007). Once it is developped in MPS, the MRP system will be executed. In such environment, the demand is always assumed to be deterministic. The figure below illustrates how do the MPS and MRP work.

Period		1	2	3	4	5	6
Forecast Sales		75	50	30	40	70	20
Sales Orders		80	45	40	50	50	5
Projected Available	50	70	25	85	35	65	45
ATP		25		10		45	
MPS		90		90		90	

MPS

MRP

Period		1	2	3	4	5	6
Gross Requirement		90		90		90	
Inventory Plan	100	10	10	20	20	30	30
Net Requirement				80		70	
Planned Reciept				100		100	
Planned Release		100		100			

Figure 1 The demonstration of MPS and MRP

According to Ptak & Smith (2011), prior to determining what actions to taken on the production items, one should have an evaluation of the status of inventory. Status information is used to identify the items we have and items we need. Then we can decide what do we do, which can be achieved by executing an evaluation procedure. The MRP system can evaluate the inventory status automatically. Under the MRP system, there are two types of status data: inventory data and requirement data. These two types of data are associated with timing information. The inventory data, which can be verified by inspection, consist of the quantity on hand and quantity on in transit (or on order). The requirement data, which are computed by the system, includes gross requirement quantities, net requirement quantities, and planned-order quantities (planned-order releases). The scheduled receipts can be taken included for the calculation of net requirements quantities which can be calculated as follows:

$$\text{Net Requirement}_t = \text{Gross Requirement}_t - \text{Scheduled Receipt}_t - \text{Projected On Hand}_t$$

The following figure is an example which explains the MRP explosion process. Firstly, we need the information comes from the Master Planning Schedule (MPS). For different components at level one, the MRP system will calculate and generate the planned releases based on the information from the MPS, quantity on hand, and the lead time. Then we move to the planning for the components at level two. For components at level two, the gross requirements are generated from the planned releases from the previous level. The remaining processing logic is the same with that in the previous level. The number of levels of components depends on the BOM. The product at level zero is the final product. Each product in the levels other than level zero is the component of product in the previous level (Turbide, 1993).

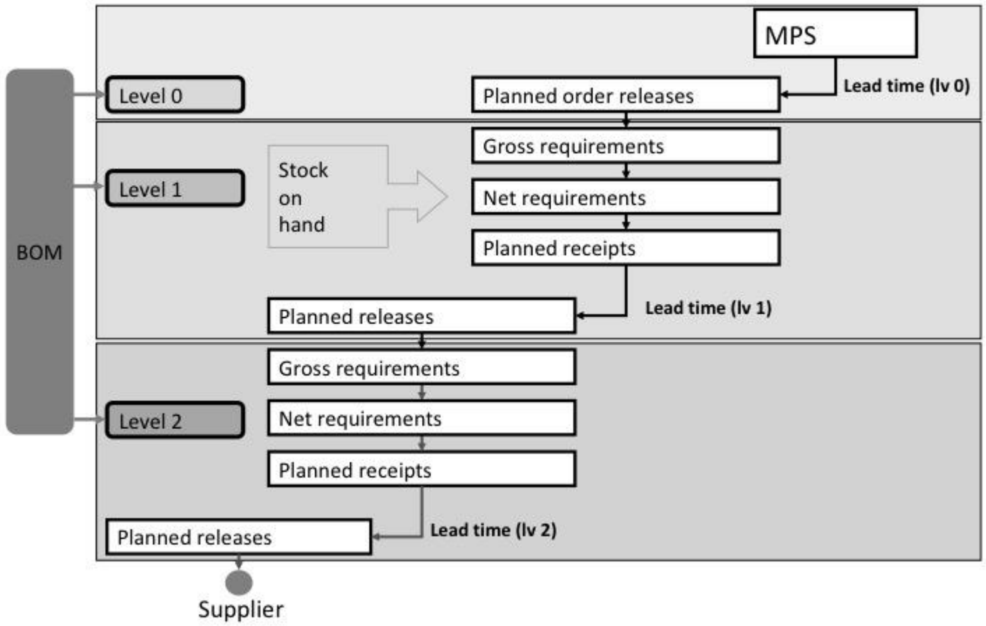

Figure 2 The MRP explosion process (adapted from Turbide, 1993)

However, in the real world, the demand is typically uncertain. The variation also includes the changes in the components (Dolgui & Prodhon, 2007). Koh, Saad, & Jones (2002) describe the sources of demand uncertainty as the inaccurate forecasts and customer order changes. According to the research from Murthy & Ma (1991), demand uncertainty comes from the forecasting errors and customer orders. There are still some other types of uncertainties in MRP system, such as the uncertainty in vendor supply, variations in product quality, variations in product structure, variations in production lead time, equipment breakdown and the dynamic lot sizing (Murthy & Ma, 1991). In this thesis, we focus on the aspect of demand uncertainty in MRP.

2.2.3 MRP Demand

MRP deals with the underlying components of the final product according to the product schedule in MPS. The demand at the first level in MRP system comes from MPS. As it is proposed by Proud (1994), "the objective of master scheduling is to plan the impact of demand on materials and capacity". There are four components in the demand section of a MPS matrix: item forecast (independent demand), option forecast (dependent demand), actual demand, and total demand. Normally, in an MPS matrix, the total demand is calculated as the sum of item forecast, option forecast, and actual demand (Proud, 1994). Demand forecast is the basis of the production planning. According to Chopra et al. (2013), there are a variety of forecast factors that are associated with the demand forecast, such as the past demand, lead time of the product replenishment, states of the economy, planned advertising or marketing efforts, etc. All these factors are not constant. Therefore, forecasts are always inaccurate. Chopra et al. (2013) classified the forecasting methods into four types: qualitative, time series, causal, and simulation. When there is limited historical data, the qualitative forecasting methods are most appropiate as these methods are more subjective and based on human judgements. The time series forecasting methods are more appropiate when the historical demand data does not vary significantly. When the demand forecast is highly correlated with the certain factors, the causal methods is usually used in the demand forecasting. The simulation method is used by imitating the consumer choices. Furthermore, Chopra et al. (2013) indicated that short-term forecasts are more accurate than the long-term forecasts, and disaggregate forecasts are usually less accurate than the aggregate forecasts. However, due to the inaccuracy of the demand forecasting, the forecast error (demand uncertainty) should be taken into account when making production decisions.

2.2.4 Bill of materials (BOM)

The interface between MPS and MRP is required when we deal with the underlying components and the final products. This interface is made via a bill of materials (BOM) (Proud, 1994). The BOM is defined as "a listing of all the subassemblies, intermediates, parts and raw materials that going to making the parent assembly showing the quantities of each required to make an assembly" (Arnold, Chapman, & Clive, 2012). On the bill of materials, all the information related to the item are included, such as all the required parts, unique part number, required quantity, function description, etc. Arnold et al. (2012) discuss a variety of important formats of BOM: product tree, parent-component relationship, and multilevel bill. We take the bicycle and the chair as examples. The examples of BOMs are as follows.

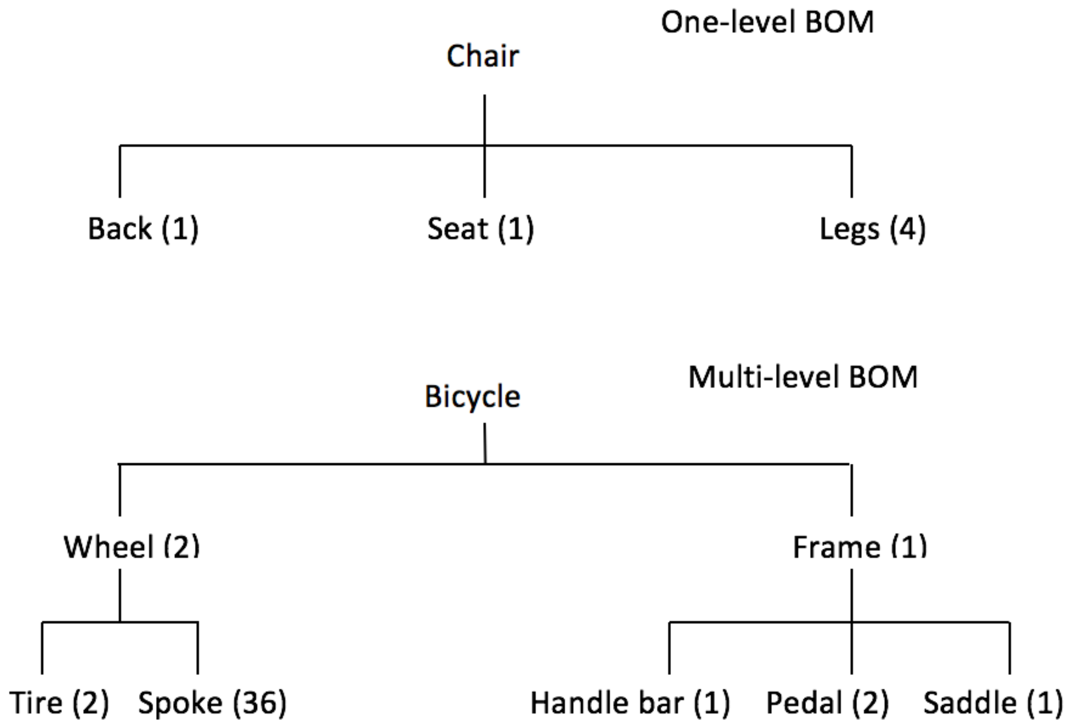

Figure 3 Examples of BOM

2.3 Dealing with uncertainty in production planning

In this section, first, we generally talk about the studies about dealing with uncertainty in the context of production planning. Second, we will demonstrate production optimization models under uncertainty.

2.3.1 Dealing with uncertainty in general

Uncertainties can affect the operation of the company significantly and negatively. According to Murthy & Ma (1991), the demand uncertainty can result in the order modifications which have further impacts on the total quantities of end products. This order instability caused by the demand uncertainty is called nervousness. This can provoke delays or orders not being met. Dolgui & Prodhon (2007) shared the same conclusion in their research. They claimed that the demand variation can provoke the nervousness. Another negative impact of the demand uncertainty is on the inventories. It can result in shortages or surplus. Murthy & Ma (1991) summarized the impacts as the follows: more frequent rescheduling, order not being serviced effectively, unplanned setups for unplanned demands, loss of sales and goodwill, and excess and insufficient inventories. All these impacts can cause great increase in costs.

To deal with the different types of uncertainty in MRP system, several approaches are used. Murthy & Ma (1991) summarized four different methods in their research. They claimed that safety stocks, safety lead tims, hedging and over planning, and yield factors. In the safety stock method, the level of inventory is increased to buffer the uncertainty. To use this method, the size of the safety stock have to be determined. According to their research, there are two methods to determine the size of the safety stock: economic method and service level method. The goal of the economic method is to minimize the total costs which include the setting up cost, stock holding cost, and penalties for shortages. In the service level method, the level of the customer service they wish to provide has to be determined before the determination of the size of the safety stock. In the method of the safety lead times, orders are released earlier than what is stated in the requirement plan. In this way, the end product can be delivered before their due time. The safety lead times method is mostly used to deal with the supplier uncertainty. When the delivery of items purchased from the suppliers is delayed, there will be a slack to buffer this delay.

However, as it is mentioned in the research from Yu & Li (2000), the main challenge the supply chain management is facing is the the uncertainty of the future. Many logistics optimization models are facing uncertainties of different scenarios. Variables or parameters have different probabilities of occurrence. When facing these uncertainties, these optimal solutions (safety stock or safety lead times) are not robustness enough.

Stadtler (2005) also pointed out the limitations of deterministic models in their research. All these models assumed that the demand follows a constant demand rate, which is contrast to stochastic demand observations in variety of areas. Stochastic programming is another method to deal with the uncertainty. In the stochastic programming model, the model size is exponentially expanded as there are lots of scenarios in each period in a multi-period model. Thus the stocahstic programming can provide a more robust solution when facing different scenarios (Stadtler, 2005).

2.3.2 Production optimization models under uncertainty

Many studies consider optimization models with uncertain parameters. The following is a discussion about the models which deal with demand and lead time uncertainties, two common uncertain inputs in practice.

Grubbström & Molinder (1996) developed basic one and two-level models by using the Laplace transform methodology together with the input-output analysis . This basic model formulated the theoretical description of material requirements planning (MRP). The sum of the expected average of set up costs, inventory holding costs, and backlogging costs are used as the objective function of their one and two-level models. The demand of this model follows the Poisson distribution. And the aim of theirs models was to calculate the optimal safety stock values. Followed this research, Grubbström (1998) made an extension of the previous research (Grubbström & Molinder, 1996). In this new paper, the principle to determine the size of the safety stock was provided. A set of safety stock amount was decided before the demand was known. In the previous paper, the average cost approach was used. However, the annuity stream was used as a criterion in this new paper. The annuity is a "variation of net present value" (Grubbström & Molinder, 1996). The author used the one level model assuming the demand follows the Poisson distribution. The difference between the annuity stream and the method of traditional average cost method was examined. In addition, Grubbström & Tang (1999) further developed the safety stocks in an MRP systems. In their new research, it is claimed that the variance in demand has a significant impact on the level of the optimal safety stock. The optimal safety stock in multi-level MRP systems was studied in this paper. The multi-level inventory model was used to maximize the net present value. Different from the previous studies which Poisson distribution was used, the demand in this model was Gamma-distributed. It provided a better understanding of the method of determining the optimal safety stock when demand follows a Gamma distribution. Following all the research mentioned above, Grubbström & Wang (2003) studied a stochastic model of multi-level capacity-constrained system. In this model, the objective function was chosen as the expected net present value, and the solution procedure was the dynamic programming. The author further discussed the transforms. The dynamic programming was examined as a practical way when it comes to the multi-stage optimization. In addition, the basic model was illustrated by the numerical examples.

Louly, Dolgui, & Hnaien (2008) conducted a research in optimal planning in MRP with uncertain lead times. This systems was designed for the assemblies. The objective function of this model was to minimize the sum of different costs which include the average component holding cost and average backlogging cost. In this research, there are a variety of types of components, and the lead time of each component is random. The actual lead times are chosen as the random variables to find out the optimal planned lead times for all the components. The assumption of this model is that the demand of the end product is deterministic and the production capacity is infinite. Different kinds of components have different kinds of probability distributions. In the research conducted by the Dolgui & Ould-Louly (2002), a similar model was used. In their model, the objective function was also to minimize the sum of expected backlogging cost and holding cost. The demand was constant as well. This paper is also in the sake of finding out the optimal value of of the lead time.

Bai et al (2002) used a simulation model to to examine how the system parameters and operating factors can affect the performance of MRP system. System parameters include the schedule frozen interval, schedule replanning interval, lot-sizing and safety stock; operating factors consist of demand forecast accuracy and product lead times. The MRP system performance was defined as the schedule instability, service level and total cost. The author claimed that all those four parameters (schedule frozen interval, schedule replanning interval, lot-sizing and safety stock) and operating factors have significant impact on the MRP performance. However, because of the interaction between all the parameters and factors, there is no win-win combination to achive better system performance under all the situations.

Jeunet & Jonard (2000) evaluated the lot-sizing techniques based on the cost-effectiveness criterion and the robustness of set-up in the uncertain demand environment. A variety of measures of robustness was discussed and proposed in the paper. The author examined how the production schedules can be affected by the demand variability for single level assembly. According to the simulation results, the relationship between cost-effectiveness and and the robustness was inverse. The table below summarizes models above.

Authors	Uncertainty	Objective Function	Parameters	Type of Model
Grubbström&Molinder (1996)	Demand	Minimize the sum of setup/holding/backlogging costs	Optimal Safety Stock Value	One-level and two-level MRP
Grubbström (1998)	Demand	Annuity Stream (a variation of net present value)	Safety Stocks	One-level MRP
Grubbström&Tang (1999)	Demand	Net Present Value	Optimal Safety Stocks	Multi level MRP
Grubbström&Wang (2003)	Demand	Expected Net Present Value	Safety Stocks	Multi level stochastic model
Louly, Dolgui, & Hnaien (2008)	Lead Time	Minimize the sum of holding/backlogging costs	Different Components	Optimization
Dolgui & Ould-Louly (2002)	Lead Time	Minimize the sum of holding/backlogging costs	Planned Lead Times	Single-item/Multi-item Model
Bai, Davis, Canet, Cantrell, & Patterson (2002)	Demand	Schedule Instability/ Service Level/ Total Cost	Frozen interval/ Replanning interval/ Lot-sizing/ Safety Stocks	Multi-level
Jeunet & Jonard (2000)	Demand	Cost-effectiveness/ Set-up robustness	Lot-sizing	Single-level

Table 1 Summary of optimization models

Following the topics in the literature review, in the next chapter, we demonstrate the details of the determinisitic model and the stochastic model. The MRP, stochastic programming are involved in our models.

Chapter 3 - Material Requirements Planning (MRP) Under Demand Uncertainty

This chapter is organized as follows. In Section 3.1, the original MRP model is discussed; in Section 3.2, the description and the mathematical formulation of the deterministic MRP model are provided; in Section 3.3, we will demonstrate the description and the mathematical formulation of the stochastic MRP model.

3.1 Traditional MRP model

The MRP model used in this thesis is adapted from a model by Voß & Woodruff (2006). In the original model, the classic MRP plan (as it is shown in the table below) was used.

Day	1	2	3	4	5	6	7	8
Gross requirement	20	30	10	20	30	20	30	40
Inventory plan (90)	70	40	30	10	80	60	30	90
Net requirement					20			10
Planned receipt					100			100
Planned release			100			100		

Table 2 The MRP plan in the original model

In this plan, the main calculation logic is

$$Inventory\ Plan_{\,t} = Inventory\ Plan_{\,t-1} + Planned\ Receipt_{\,t} - Gross\ Requirement_{\,t}$$

The data used for the MRP formulation in the original model is as the table shown below.

Number of SKU
Number of time buckets (i.e., the planning horizon)
Number of *item* i that is needed to make one j
Demand for i in period t
Initial inventory of SKU_i
Minimum lot size for SKU_i
A sufficiently large number
Quantity of SKU_i to start or order in period t

Table 3 The data used for the MRP formulation in the original model

In the optimization model of the MRP, the objective function is to make the production of each item as late as possible (Voß & Woodruff, 2006). Denote by K the set of SKUs and by T the set of time buckets. If the production quantity is defined as q_{it}, the objective function can be written as follows.

$$\sum_{i \in K} \sum_{t \in T} (|T| - t) q_{it}$$

The constraints include the demand and material requirements (which is the calculation of the inventory plan shown above), lot size constraint, production indicator constraint and the constraint of the non-negative production. These constraints are formally described in the deterministic MRP model in the subsequent section.

3.2 Deterministic MRP model

We develop a MRP model with deterministic demand based on the MRP optimization model from Voß & Woodruff (2006). In our model, the inventory holding cost and the stock out cost are considered. Instead of making the production of each item as late as possible, the objective function is to minimize the sum of the inventory holding cost and the stock out cost. We multiply the inventory at the end of each period and the inventory holding cost per unit to get the total inventory holding cost for one period. The sum of all the inventory holding cost and stock out cost for each period is the objective we want to minimize through our model. Note that if the stock out cost is not allowed (this can simply be done by setting the bound on the stock out quantity variable to zero) and the inventory cost in different levels of production is the same, the model is equivalent to that of Voß & Woodruff (2006). The input of this MRP model, which includes the BOM, the number of production periods, the holding cost for each item, the stock out cost for the end item and the demand sets, are used as the parameters of optimization model. The decision variables are the quantity of inventory at the end of each period for each item and the number of batches for each item at each period to produce. In this setting, the production is done in batches, where each batch has a fixed quantity to produce. The children are the components used to produce the upper level item, and the parent is the corresponding upper level item to the children. In our model, we assume that each child item corresponds to one parent item (but one parent can consist of multiple children items). The unmet demand of the end product is assumed to be lost and the cost of lost sales is paid for each unit of lost sales. The detailed definition of variables, parameters and mathematical formulations are provided as follows.

General notations for the formulations are as follows:

Define the following parameters:

Set:

T	Set of time periods {1, …, m}
K	Set of components in BOM {0, …, n} where 0 is the level of end product

Parameters:

C_h^k	Inventory holding cost of item k per period, $\forall k \in K$
d_t^0	Demand for the end item at the period t, $\forall t \in T$
LT^k	The number of time periods between issuing the production order of k until the item is available, $\forall k \in K$
B^k	Batch size of item k, $\forall k \in K$
C_l	Cost of lost sale per item of the end product
$parent\ (k)$	Index of parent item of child item k
$R_{parent\ (k)}^k$	Number of item k that is needed to make one parent item of item k

Decision variables:

I_t^k	Quantity of item k in stock at the end of period t, $\forall t \in T, \forall k \in K$
q_t^k	Quantity of item k available at the beginning of period t, $\forall t \in T, \forall k \in K$
y_t^k	The number of batches of item k received at the beginning of period t, $\forall t \in T, \forall k \in K$
L_t	Amount of lost sale, $\forall t \in T$

The formulation of the deterministic MRP can be written as follows:

$$\text{Min} \quad \sum_{t \in T} \sum_{k \in K} C_h^k I_t^k \; + \; \sum_{t \in T} C_l L_t \tag{1}$$

Subject to:

$$I_{t-1}^0 + q_{t-LT^k}^0 = I_t^0 + d_t^0 - L_t \qquad \forall t \in T \tag{2}$$

$$I_{t-1}^k + q_{t-LT^k}^k = I_t^k + R_{parent\,(k)}^k q_t^{parent(k)} \qquad \forall k \in K \backslash \{0\}, \forall t \in T \tag{3}$$

$$q_t^k = B^k y_t^k \qquad \forall k \in K, \forall t \in T \tag{4}$$

$$I_t^k, q_t^k, L_t \geq 0, y_t^k \in Z^+ \tag{5}$$

This formulation is a representation of the modified MRP deterministic model. The objective function (1) is to minimize the sum of inventory holding costs and the cost of lost sale. Constraint (2) is the material flow equation for the end product at level zero. Constraint (3) is the material flow equation for the components of the end product which belong to the levels higher than zero. Constraint (4) is the setup and production batch constraint for the quantity available at the beginning of each period. This constraint imposes that the number of units in period $t +$

$LT^k_{parent(k)}$ depends essentially on the number of production batches in period t and the size of batch B^k. Constraint (5) defines the variables.

3.3 Stochastic MRP model

Based on the deterministic model, a two-stage stochastic model is proposed to deal with uncertain demand. In this stochastic model, the objective function, the variables and parameters are aligned with the ones presented in the deterministic model. However, due to demand uncertainty, different scenarios of demand are incorporated. Each demand set will be associated with a probability. By solving this stochastic model and the stochastic model by using the expected value solution, we can get the solution which can hedge against different scenarios. In this two-stage model, the number of production batches for each component in each period is determined in the first stage. Then, this production decision is "frozen" and the quantity cannot be adjusted (which is in-lined with the frozen period in the MRP setting). Since the production quantities must remain fixed, inventory and lost sale quantities can be calculated for each demand scenario for the entire planning horizon. This is treated as in the second stage in the model.

We define additional parameters and variables for the two-stage stochastic model as follows:

Ω	Set of scenarios {1,2,…r}
P_w	The probability of scenario w, $\forall w \in \Omega$
$d^0_{t,w}$	Demand for end item at the period t in the scenario w, $\forall t \in T, \forall w \in \Omega$

Decision variables:

$I^k_{t,w}$	Quantity of item k in stock at the end of period t in scenario w, $\forall t \in T, \forall k \in K, \forall w \in \Omega$
$q^k_{t,w}$	Quantity of item k available at the beginning of period t in the scenario w, $\forall t \in T, \forall k \in K, \forall w \in \Omega$
$L_{t,w}$	The amount of lost sale in scenario w, $\forall t \in T, \forall w \in \Omega$

The stochastic MRP formulation can be written as follows.

$$\text{Min} \qquad \sum_{w \in \Omega} \left[P_w \left(\sum_{t \in T} \sum_{k \in K} C_h^k I_{t,w}^k + \sum_{t \in T} C_l \, L_{t,w} \right) \right] \tag{6}$$

Subject to:

$$I_{t-1,w}^0 + q_{t-LT^k,w}^0 = I_{t,w}^0 + d_{t,w}^0 - L_{t,w} \qquad \forall t \in T, \forall w \in \Omega \tag{7}$$

$$I_{t-1,w}^k + q_{t-LT^k,w}^k = I_{t,w}^k + R_{parent\,(k)}^k q_t^{parent(k)} \qquad \forall k \in K \backslash \{0\}, \forall t \in T, \forall w \in \Omega \tag{8}$$

$$q_t^k = B^k y_t^k \qquad \forall k \in K, \forall t \in T \tag{9}$$

$$I_{t,w}^k, q_{t,w}^k, L_{t,w} \geq 0, y_t^k \in Z^+ \tag{10}$$

This formulation is a representation of the stochastic MRP model. The objective function (6) is to minimize the sum of inventory holding costs and the cost of lost sale in different scenarios. Constraint (7) is the material flow equation for the end product at level zero in different scenarios. Constraint (8) is the material flow equation for the components of the end product which belong to levels higher than zero in different scenarios. Constraint (9) is the setup and production batch constraint for the quantity available at the beginning of each period in different scenarios. This constraint imposes that the number of units in period $t + LT_{parent\,(k)}^k$ depends essentially on the number of production batches in period t and the size of batch B^k. Constraint (10) defines the variables.

This chapter demonstrates the details of models. We are going to use these models in our experiments and to analyze the solutions obtained by using these models. In the next chapter, we explain the experimental design and provide computational results and analyses.

Chapter 4 - Computational Experiment and Analysis

In this chapter, first, we demonstrate how the experiments are designed and how the instances are generated. Second, following the experimental design, the computational solutions are presented. Next, the sensitivity of the optimal solutions, expected value of perfect information (EVPI) and value of the stochastic solution (VSS) are presented. We use Gurobi version7.5 as the solver, and the model and experiments are coded by Python version 2.1.3. The maximum CPU time for the deterministic model is 3600 seconds, whereas the maximum CPU time for the stochastic model is 9600 seconds. The general time used for solving instances with 100 demand scenarios is around 10 seconds, and the time used for solving instances with 1000 demand scenarios is around 1200 seconds.

4.1 Data instance generation

The instances we generate consist of 8 items whereas one of the items is the end product and the other seven items are components of the end product. By assembling them in three different ways, we generate three BOMs (shown in Figure 4). We will examine these three BOMs one by one in the computational experiments and analyses.

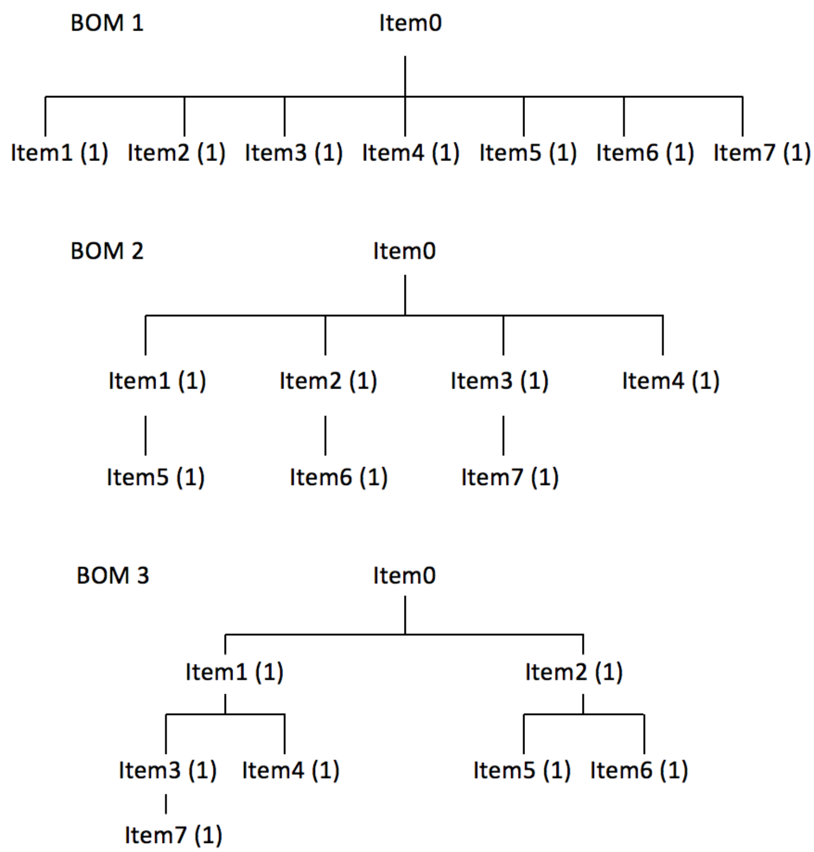

Figure 4 Three BOMs in the MRP model

For the planning horizon, we have two different set, i.e., 8 and 16 production periods. We set the lead time of the end product and components to one period. The data of the initial inventory, batch size, holding cost per unit, and the stock out cost per unit are summarized in the table below.

SKUs	Initial Inventory (I_0^k)	Holding Cost (C_h^k)	Lot size (B^k)
SKU1	150	10.0	50
SKU2	200	3.0	30
SKU3	150	3.0	30
SKU4	150	3.0	30
SKU5	200	3.0	30
SKU6	170	3.0	30
SKU7	160	3.0	30
SKU8	150	3.0	30

Table 4 Summary of the parameters used in the dataset

The safety stock for SKU1 is applied in the deterministic model. It is calculated based on the service level at 95% as follows. In the calculation, the standard deviation is determined based on the scenarios. The lead time is the total lead time from the bottom of the BOM to the top.

$$Safety\ Stock = z_{95\%}\ \times\ Standard\ Deviation\ (Demand_{SKU1}) \times \sqrt{Lead\ Time_{SKU1}}$$

By using this deterministic model, we analyze two different demand sets. The first one contains 100 demand scenarios. The second one contains 1000 demand scenarios.

In the stochastic model, we assume that each demand scenario has a same probability of occurrence. In our experimental test, we analyse two different size of demand sets which is exactly the same we use in the deterministic model. For the demand data with 100 scenarios, each demand set is associated with a probability of 0.01. For the demand data with 1000 scenarios, each demand scenario is associated with a probability of 0.001. In the stochastic environment, for the instances with 8-period planning horizon with 100 scenarios, we can have 100×8 demand values; for the instances with 16-period planning horizon with 1000 demand scenarios, we can have 1000×16 demand values. These large sets of demand scenarios are used to capture demand uncertainty in the MRP environment.

Based on the deterministic model and the stochastic model we discussed above, we develop our experimental design by changing the parameters and combining them in different ways. We have two planning horizons which are 8 and 16 periods. We also test two different sizes of demand scenario set (100 scenarios and 1000 scenarios). In addition, by changing the structure of the product, we get three different BOMs. The changing of values of the inventory holding costs and the stock out cost are involved to generate instances. The first data set of these costs are the standard one which is the original value as it is shown in the table 6. The second data set is ten times value of the inventory holding cost while the stock out cost remains unchanged. The third data set is that the stock out cost is ten times the original stock out cost while the inventory holding cost remain the same. This information is summarized in Table 5 below.

Parameters	Description
BOM	BOM1, BOM2, BOM3
Planning Horizon	8 periods, 16 periods
Demand scenarios	100 scenarios, 1000 scenarios
Cost	Standard, Holding× 10, Stock out×10

Table 5 Parameters Description

By combining all the parameters, we can get 36 combinations. Based on these 36 sets of parameter combinations, we can get 36 sets of solutions for each model. The summary of the parameter combinations is shown in the table below. Each combination is labelled by a number as follows.

N=8	Scenarios	Standard Ch	Ch × 10	Stockout ×10
BOM1	100	1	2	3
BOM2	100	4	5	6
BOM3	100	7	8	9
N=16	Scenarios	Standard Ch	Ch × 10	Stockout ×10
BOM1	100	10	11	12
BOM2	100	13	14	15
BOM3	100	16	17	18
N=8	Scenarios	Standard Ch	Ch × 10	Stockout × 10
BOM1	1000	19	20	21
BOM2	1000	22	23	24
BOM3	1000	25	26	27
N=16	Scenarios	Standard Ch	Ch × 10	Stockout × 10
BOM1	1000	28	29	30
BOM2	1000	31	32	33
BOM3	1000	34	35	36

Table 6 The structure of the parameter combinations

As it is mentioned in the previous sections, for the deterministic MRP, the model's sensitivity of the cost effectiveness will be discussed. The discussion of the expected value of perfect information (EVPI) and the value of stochastic solution (VSS) will also be included for further analysis.

For the EVPI and VSS analysis, to get all the solution data needed, based on the deterministic model and the stochastic model we have, for each combination of parameters, we need four data solutions. To be specific, first, EVPI is the loss of the profit due to the existence of the uncertainty, which is also the value of knowing the future with certainty. It is the maximum amount of money the decision makers would like to pay for the perfect information. If the perfect information were available, we can get the optimal solutions by solving the deterministic model. The expected value of this solution under the perfect information is the average value of optimal solution of each demand instance. In this thesis, we name the objective value of this solution $OBJ_{avg\_perfect}$. Second, we need the solution data from by solving the stochastic model. Here we name the objective value of this solution from the stochastic model OBJ_{stoc}. Then the EVPI can be calculated as the equation below:

$$EVPI = OBJ_{stoc} - OBJ_{avg\_perfect}$$

In addition, the VSS is the possible benefits we can get from the stochastic model. It is the value of knowing and using the distribution on the future outcome. To get VSS, firstly we need to get the solution of the deterministic version of the stochastic model which the random parameters (demand values) are replaced by the expected values, which is defined as SOL_{deter}. This solution of the deterministic model with the input of average demand is then evaluated by the entire set of demand scenarios and the resulting objective value of this solution is named as $OBJ_{\exp\_deter}$. The value of the VSS is the difference between the $OBJ_{\exp\_deter}$ and the OBJ_{stoc}. The equation of the VSS is as shown below.

$$VSS = OBJ_{\exp\_deter} - OBJ_{stoc}$$

To make it clear, this processing is illustrated in the flowchart below.

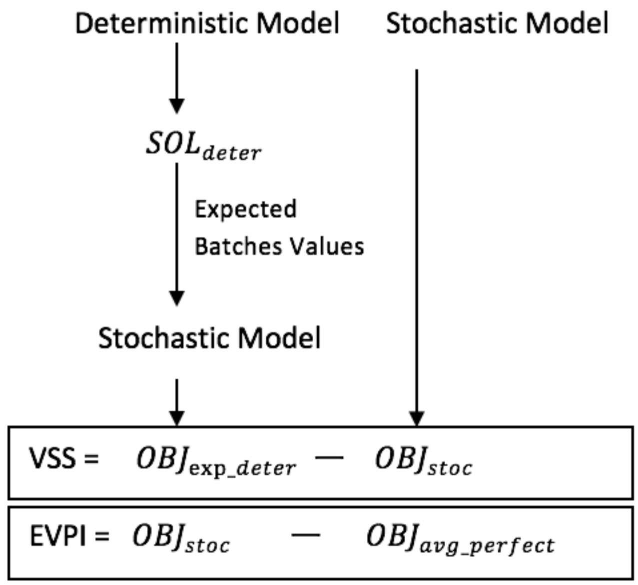

Figure 5 Illustration of the calculation of EVPI and VSS

Therefore, by using the deterministic model and the stochastic model we have, we need to get four solution values for each parameter combination. For example, for parameter combination 1 which has 8-period planning horizon and 100 scenarios with BOM1 and standard cost, we need to get the solution values of SOL_{deter}, $OBJ_{\exp\_deter}$, OBJ_{stoc} and $OBJ_{avg\_perfect}$.

Next, we illustrate the experimental design for the cost sensitivity analysis. In this part, only the deterministic model is involved. In order to test whether the optimal solution is sensitive to the changes in demand, we run the instances with 100 demand scenarios and with 1000 demand scenarios in the deterministic model one by one. Then we can get 100 or 1000 optimal solution values for each instance. Since we have 36 instances. Therefore, we can get 19,800 solutions based on these parameter combinations. Among these solutions, 1800 solutions are from instances with 100 scenarios (100 x 18), the rest are from instances with 1000 scenarios (1000 x 18). By analyzing these data, we will get the insights about the sensitivity of cost effectiveness with uncertain demand in an MRP environment.

4.2 Computational results and analysis

By running the model, we get computational results in this section. The analysis of sensitivity of optimal solutions, EVPI and VSS are provided. Note that the deterministic model is referred to the case when safety stock is imposed. We have also attempted to solve the deterministic model without safety stock, and the results are far inferior to the case with safety stock. The case when safety stock is predetermined and added to the plan is in-lined with the traditional MRP. For the stochastic model, this quantity of safety stock is not necessary since the model takes into account demand uncertainty directly.

4.2.1 Computational results and analysis for the sensitivity of deterministic solutions

We run the deterministic model for 100 times for instances with 100 demand scenarios, and run the model for 1000 times for instances with 1000 demand scenarios in order to obtain the objective values of the solutions when the demand information is assumed to be perfectly known. In this section, we would like to demonstrate that the solutions associated with different demand scenarios vary significantly and thus a solution based on the MRP logic that is produced by a specific scenario could be very different from the optimal solution based on the realized demand once it becomes known. The results we obtain in this section will also be used in EVPI calculations. Each of the 100 demand scenarios or 1000 demand scenarios is used as the input data in this model to obtain solutions. Following the first instance, the same processing is repeated for the instances 2 to 36. Then we get all the optimal solutions for all the instances. In Appendix A, we show detailed results of all optimal solutions.

The values mean, maximum, minimum, standard deviation and coefficient of variation (CV) for all instances are presented in the table below.

Instance	Mean	Maximum	Minimum	SD	CV
1	5843	9360	3910	1146	20%
2	58432	93600	39100	11461	20%
3	5946	9490	4330	1121	19%
4	9930	15240	5670	2332	23%
5	99300	152400	56700	23318	23%
6	9963	15240	5830	2302	23%
7	10887	14780	7240	1796	16%
8	108872	147800	72400	17959	16%
9	10910	14780	7240	1790	16%
10	9300	13130	7520	1206	13%
11	93006	131300	75200	12063	13%
12	9569	13400	7720	1171	12%
13	13432	19050	9100	2463	18%
14	134325	190500	91000	24627	18%
15	13629	19130	9160	2412	18%
16	14910	20850	10690	2374	16%
17	149105	208500	106900	23739	16%
18	15064	20850	10690	2337	16%
19	5771	12320	3270	1213	21%
20	57714	123200	32700	12126	21%
21	5932	34940	3270	1705	29%
22	9688	19890	3900	2403	25%
23	96883	198900	39000	24029	25%
24	9786	35210	3900	2594	27%
25	10680	17990	5170	1870	18%
26	106802	179900	51700	18702	18%
27	10770	36170	5170	2137	20%
28	9226	15650	6780	1231	13%
29	92220	156500	67800	12307	13%
30	9570	38430	6780	1733	18%
31	13206	25180	7410	2534	19%
32	132015	251800	74100	25334	19%
33	13470	38700	7410	2709	20%
34	14670	25730	8500	2442	17%
35	146705	257300	85000	24422	17%
36	14892	39690	8520	2612	18%

Table 7 Mean values of optimal solutions based on the deterministic model with perfect demand information

Instances 1, 4 and 7 use the same cost parameters but for different BOM structures. Under the standard inventory holding cost, the average value of the optimal objective function for these three instances are 5843, 5946, 10887, respectively. Generally, we found that, with the same other parameters, BOM3 is more expensive to produce than those of BOM1 and BOM2. BOM1 is cheaper to produce than those of BOM2. Therefore, for all the instances that use the same cost parameter, the deeper level of BOM the product has, the higher production cost this product is.

Additionally, as it is shown in Table 7, with the same parameter setting, optimal solutions from the instances with ten times inventory holding costs (2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35) are ten times of the optimal solutions from the data structure designs with the standard inventory holding cost (1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34). This can be expected since the objective function of the deterministic MRP is to minimize the sum of the inventory holding cost and stock out cost. If the stock out quantity is zero and the inventory holding costs in different levels are only adjusted by a constant factor, both models yield the same solution.

As we can see from the table, for the same parameter setting, no matter how the production planning horizon is, the optimal solutions of instances with standard holding cost have the same or very close values with those with ten times the stock out cost. The objective is to minimize the sum of inventory holding cost and the stock out cost. But from the comparison between results from standard cost and the results from ten times the stock out cost, we can know that no stock out happened in these situations. And this is true after we check the solutions.

The following parts examine the variability of optimal objective function value. By analyzing the range, standard deviation and the variance of optimal objective function value, we discuss whether the optimal solution is sensitive to the uncertain demand. Here we take the instances 1 and 2 as examples of the analysis.

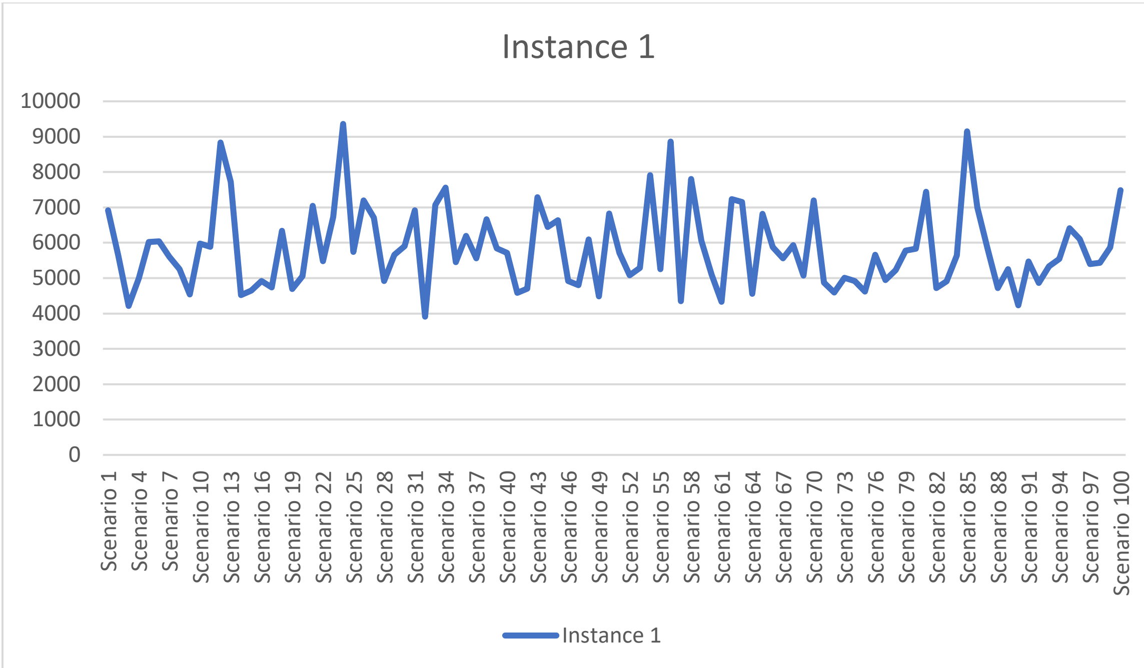

Figure 6 Line chart of optimal solutions for instance 1

The figure above describes how the objective values of the solutions fluctuate with the change of the demand scenarios. In this instance, the mean value is 5843, the maximum value is 9360, and the minimum values is 3910. After the calculation by using the Excel, we obtained the values of range and standard deviation which are 5450 and 1146. As we can see in the line chart above, the optimal solutions fluctuate dramatically in the range of 5450. As the mean value is just 5843, with a big standard deviation which is 1146, we can say that the coefficient of variation of the objective values from the mean value is large (20 % = 1146/5843)). In other words, the optimal solutions are sensitive with the demand change in the instance 1. From the Table 7, we can see that the coefficient of variation varies from 12% to 29%, and the standard deviation varies from 1121 to 25334, which are large to conclude that the optimal solutions are sensitive with the demand change for all the instances.

Figure 7 presents the central tendency of all the optimal solutions for instance 1. It describes more visually how the solution values are distributed around the mean value. It is shown in the figure that the spread of the optimal solutions is great, and the solution points are not clustered tightly. This dispersion figure also supports that the optimal solutions are sensitive to the demand change.

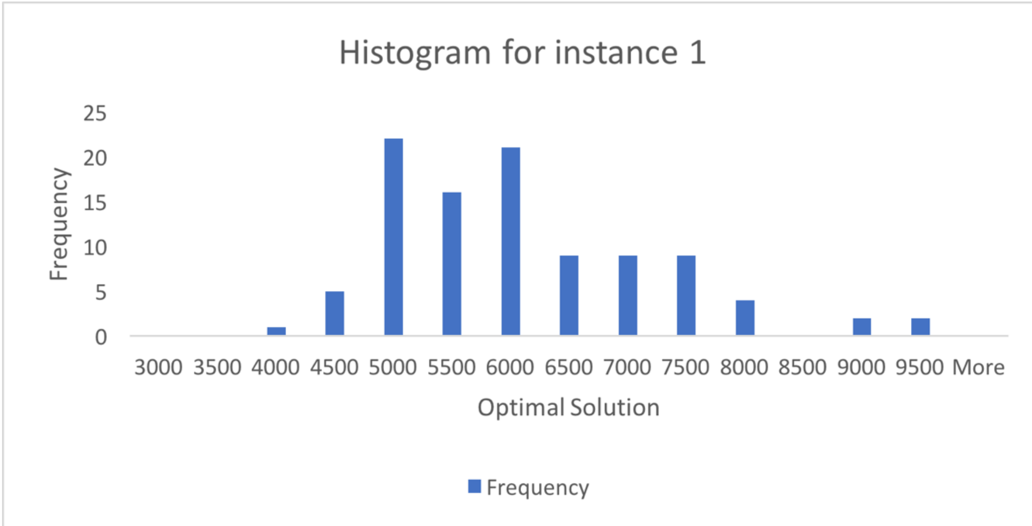

Figure 7 Histogram of optimal solutions for instance 1

Next, we examine how the inventory holding cost can affect the variability of optimal solution values. We compare optimal solutions of instance 1 and instance 2 to examine how the optimal solution values change with ten times the standard inventory holding cost.

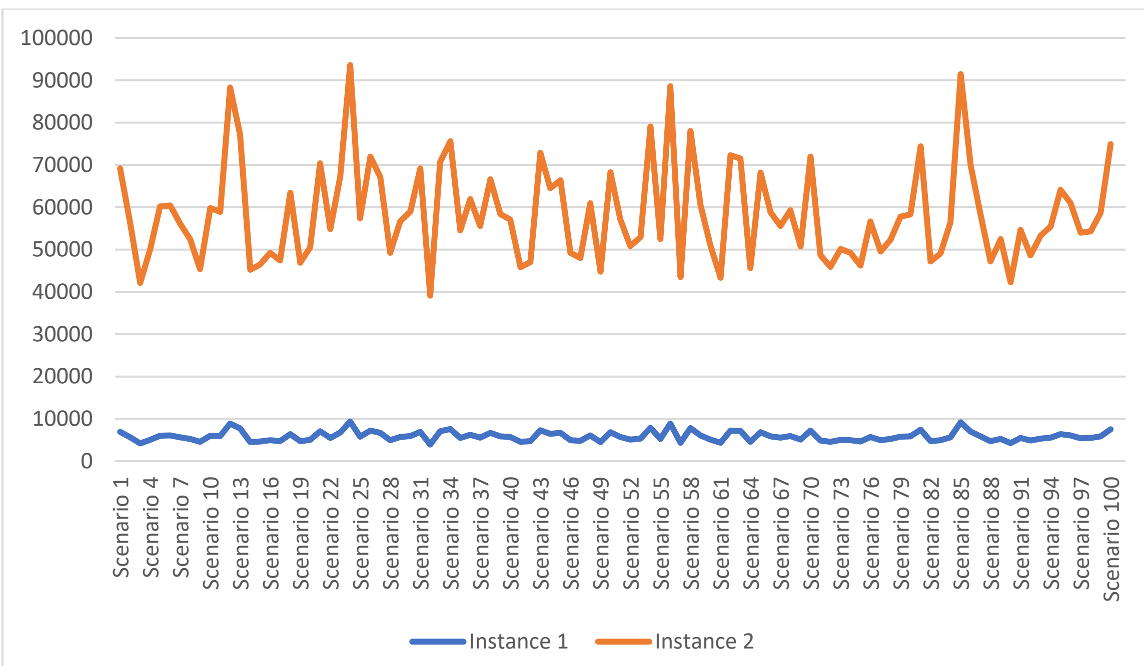

Figure 8 Comparison of optimal solutions between instance 1 and instance 2

We can see from the figure above that the line for instance 2 fluctuates more dramatically than that of instance 1. It is obvious that the optimal solutions of instance 2 change in a larger range.

Therefore, the higher inventory holding cost has increased the variability of the optimal solution values in this case.

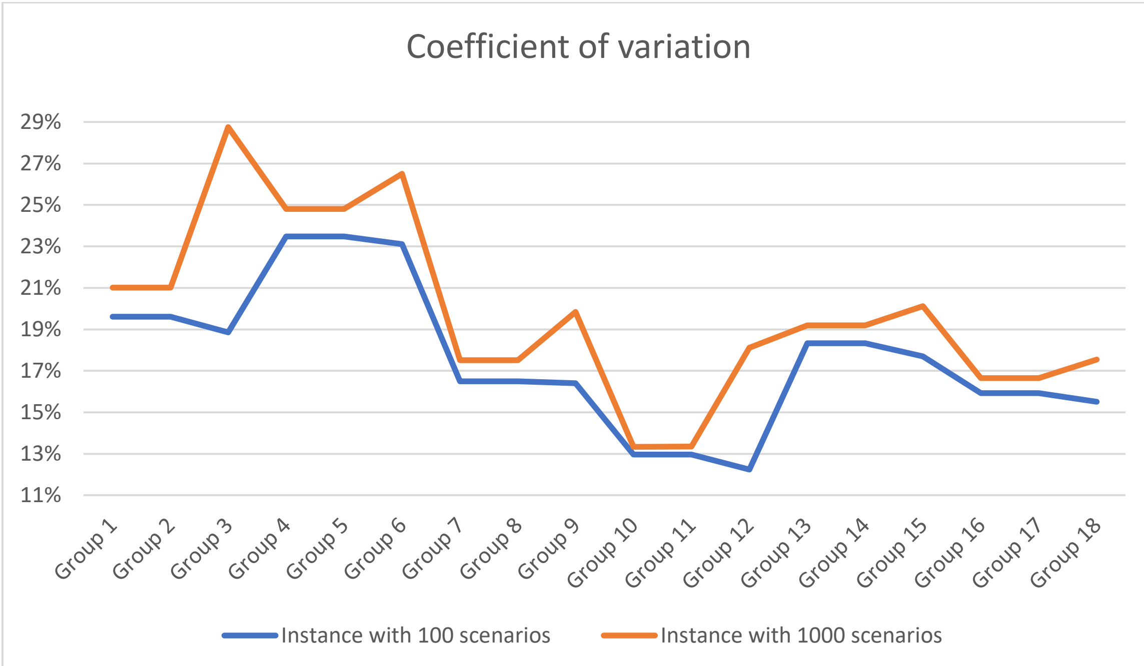

Figure 9 Comparison of coefficient of variation between two scenario sets

The figure above illustrates that the comparison of coefficient of variation (CV) between 18 groups based two scenario sets. We have instances consisting of 100 demand scenarios, and instances consisting of 1000 demand scenarios. Except the difference of the scenarios, instance 1 and instance 9 have the same parameters. Then we can classify instance 1 and instance 9 into group 1. The classification of other instances has the same processing logic. Then we can get 18 groups. Figure 7 is a comparison for these groups. As we can see from the figure, under the same other parameters, the CV of instances with 1000 demand scenarios are slightly larger than those with 100 demand scenarios. Therefore, we can say that the larger the size of the demand scenarios, the greater the variation of the optimal solutions in this case.

In brief, it is shown in the table that the coefficient of variation varies from 12% to 29%. No matter in which way we design our instances, due to the uncertain demand, the solutions fluctuates in large ranges. Therefore, it is concluded that the sensitivity of optimal solutions is great under the demand change. The larger the standard inventory holding cost, the higher the variability of optimal solutions. The deeper the level of the BOM, the higher the production cost.

4.2.2 Computational results and analysis for the expected value of perfect information (EVPI)

In this section, based on the instance (as it is shown in Table 6), we first run the deterministic model and the stochastic model for instances with 100 scenarios (instances 1-18) and instances with 1000 scenarios (instances 19-36). We get the solutions as it is shown in the table below.

Instance	OBJ_{stoc}	$OBJ_{\exp\_deter}$	$OBJ_{avg\_perfect}$	$EVPI$	$EVPI$ (%)	VSS	VSS (%)
1	8909	9480	5843	3066	34%	571	6%
2	89099	94801	58432	30667	34%	5702	6%
3	10442	11171	5946	4496	43%	729	7%
4	13430	13612	9930	3500	26%	182	1%
5	134305	136125	99300	35005	26%	1820	1%
6	14046	14069	9963	4083	29%	23	0%
7	14649	14766	10887	3762	26%	117	1%
8	146496	147663	108872	37624	26%	1167	1%
9	15649	16522	10910	4739	30%	873	5%
10	15503	16763	9300	6203	40%	1260	8%
11	155038	167638	93006	62032	40%	12600	8%
12	21418	27398	9569	11849	55%	5980	22%
13	20341	20958	13432	6909	34%	617	3%
14	203415	209587	134325	69090	34%	6172	3%
15	25010	30778	13629	11381	46%	5768	19%
16	22166	22438	14910	7256	33%	272	1%
17	221669	224386	149105	72564	33%	2717	1%
18	26738	32893	15064	11674	44%	6155	19%
19	8970	9420	5771	3199	36%	450	5%
20	89703	94203	57714	31989	36%	4500	5%
21	11230	11230	5932	5298	47%	0	0%
22	13254	13254	9688	3566	27%	0	0%
23	132547	132547	96883	35664	27%	0	0%
24	14350	14794	9786	4564	32%	444	3%
25	14539	14835	10680	3859	27%	296	2%
26	145391	148357	106802	38589	27%	2966	2%
27	15897	16460	10770	5127	32%	563	3%
28	15346	16363	9226	6120	40%	1017	6%
29	153467	163633	92220	61247	40%	10166	6%
30	22512	24547	9570	12942	57%	2035	8%

31	20084	20984	13206	6878	34%	900	4%
32	200841	209847	132015	68826	34%	9006	4%
33	25917	27922	13470	12447	48%	2005	7%
34	22003	22527	14670	7333	33%	524	2%
35	220033	225277	146705	73328	33%	5244	2%
36	27800	30118	14892	12908	46%	2318	8%
				Average EVPI	35.8%	Average VSS	5.0%

Table 8 Computational results for instances 1-36

Let's take the values from instance 1 as an example. If we have the perfect information about the demand before the start of the production, we have an average total cost of 5843. This is the average of the objective values based on the optimal value of each of the 100 demand scenarios. However, after all the demand scenarios are taken into consideration in the stochastic model, we got a total cost of 8909. The difference between these two solutions is 3066 which is the expected value of perfect information. Let's assume that, before our production, there is a way to buy this perfect demand information. The maximum amount of money you are willing to pay is the EVPI which is 3066 in this case which equals the potential savings with this perfect information.

For all the instances which have the same parameters except BOM, for example instances 1, 4 and 7, the EVPI values increase with the BOM getting deeper. From the table above, we can see that instance 7 has the largest EVPI compared with those of instance 1 and instance 4, and the EVPI of instance 1 is the smallest among these three instances. As we can see in the Figure 10 and Figure 11, this is the same for all the instances with standard holding cost and instances with ten times the standard holding cost. With the same other parameters, the deeper the BOM, the larger the EVPI in this case. Then we can say that the BOM structure has an impact on the EVPI when the stock out cost is not high. It means that, for products with multi-level BOM structures and the lower stock out cost, it is very important to get more accurate demand information as this information can help to save more money than the products with less level BOM structures.

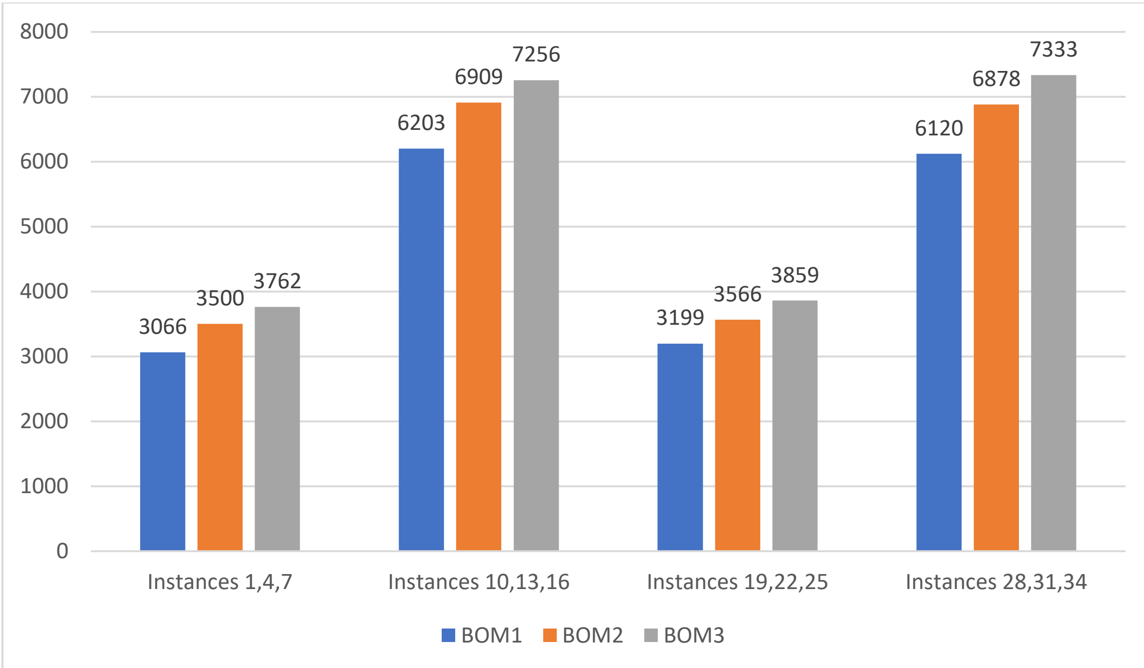

Figure 10 Bar chart of EVPI for instances with standard holding cost

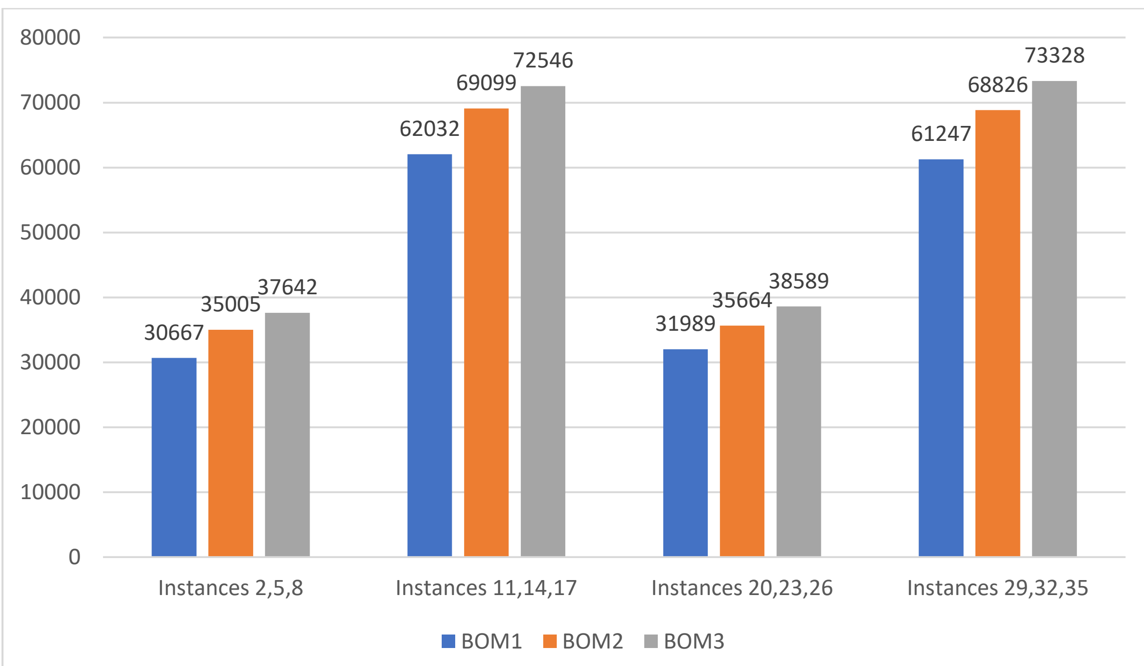

Figure 11 Bar chart of EVPI for instances with holding cost x 10

However, when we increase the stock out cost to the ten times the standard stock out cost, the impact the BOM on the EVIP is different from instances with lower stock out cost. In this case, as we can see in the Figure 12, most of instances with BOM1 have the largest EVPI, and instances with BOM2 have the smallest EVPI. EVPI of instances with BOM3 is in the middle. In this case, we can say that, when the stock out cost is high, the patterns observed in Figure 10 and 11 do not necessarily follow and the impact of the depth of the BOMs in this case are relatively similar among different BOMs.

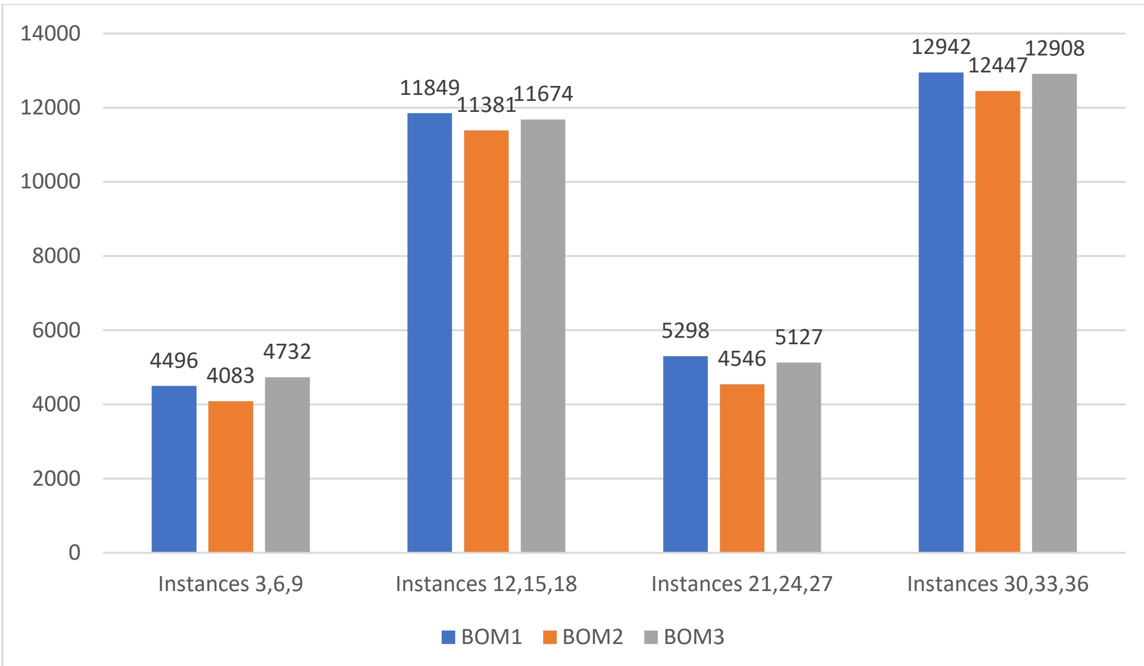

Figure 12 Bar chart of EVPI for instances with stock out cost x 10

In addition, if we look at the percentage values in the table other than EVPI, we can see that instance 1 has 34% percentage of saving, and instances 4 and 7 have the same efficiency which are 26%. We than see from the Figure 13, for other instances which have same parameters and different BOMs, instances with BOM 1 always have the highest percentage of savings. Efficiencies of instances with BOM2 and BOM3 are pretty much the same, which are smaller than the efficiency of BOM1. Thus, under the same situations, products with shallow BOM structures have higher percentage of savings. In other words, with perfect information, products with deeper level BOMs are not as efficient as those with less level BOMs when it comes to the cost saving.

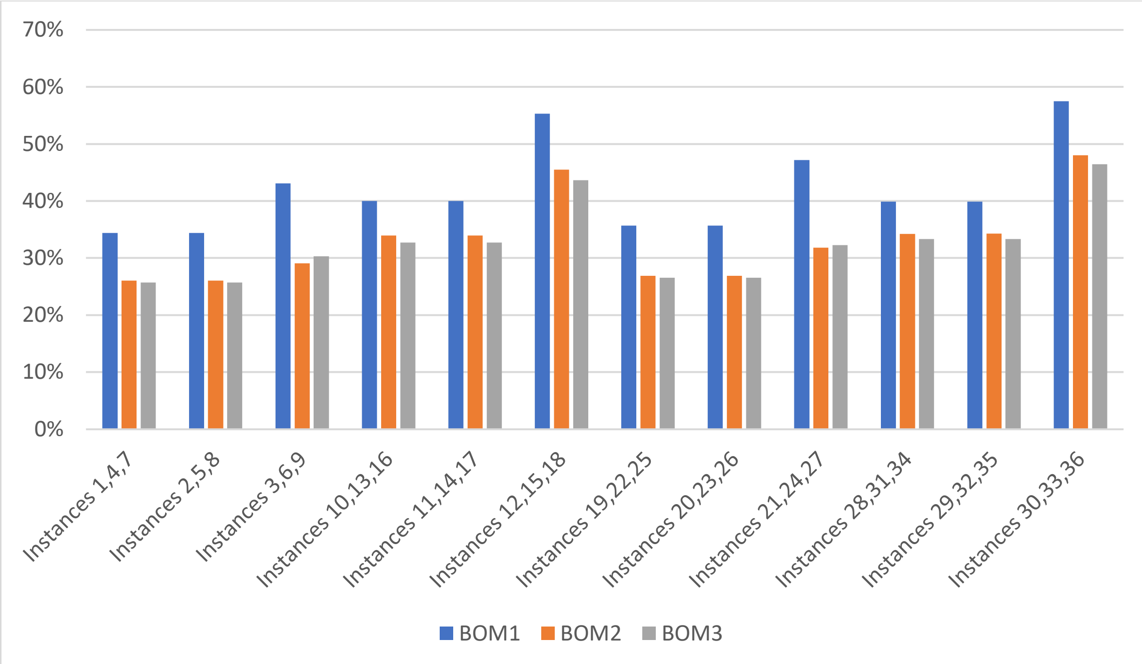

Figure 13 Bar char of EVPI efficiency for instances 1-36

Next, we are going to examine how the number of demand scenarios affects the EVPI and the percentage of savings. The figure below are the line charts of EVPI from two groups. Since the parameters of each of the demand instances s where $s \in [1, \ldots, 18]$ corresponds to the demand instance $s + 18$ except that the size of the demand scenario set increases from 100 (in instances 1-18) to 1000 (in instances 19-36). For example, the parameters in instance 1 correspond to those in instance 19. We name the instance s and $s + 18$ as CDSD 1 in the line chart below. By comparing the EVPI of the corresponding data structure designs, we can get further information.

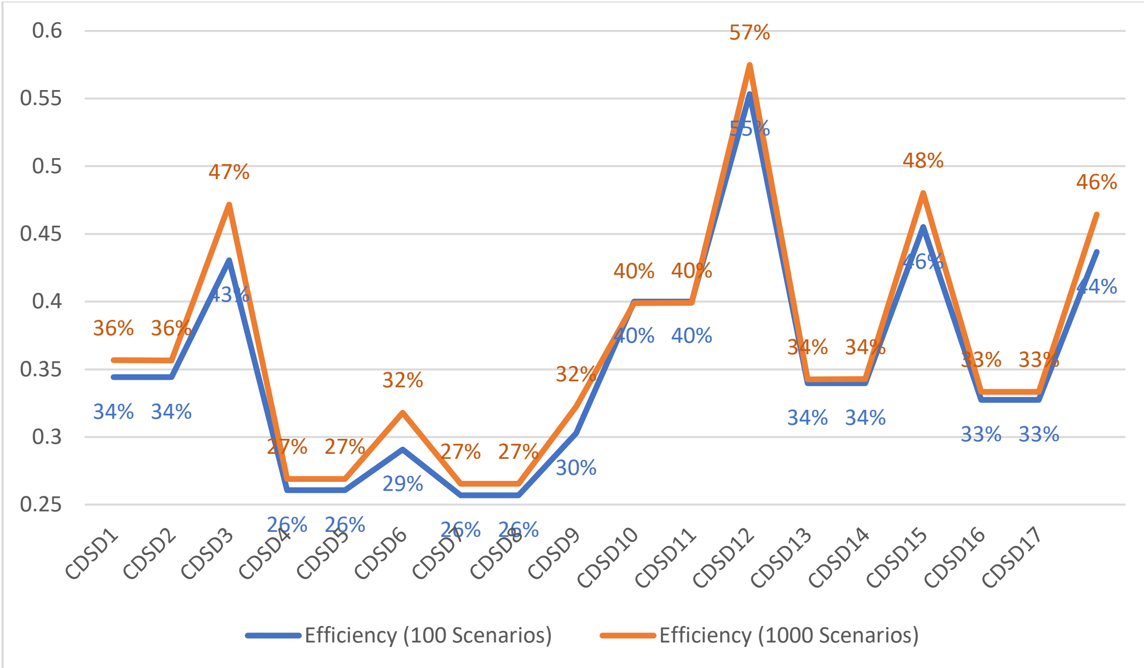

Figure 14 Comparison of EVPI for two different demand instance groups

From the Figure 14, we can see that the efficiency from the group of 1000-demand-scenario instances are generally larger than the values from the group of 100-demand-scenario instances as the orange line is always above the blue line for all the CDSDs but with a very small margin. The average efficiency from the group of 1000-demand-scenario instances is 36.4%, whereas the average efficiency from the group of 100-demand-scenario instances is 35.1%. The average difference is 1.3%.

In reality, more accurate information could possibly be obtained by more intensive data collection and advanced forecasting methods based on rich dataset from multiple sources. The reduction of demand uncertainty can also be achieved by reducing the lead time so that the forecast does not need to be made very early in the process and better forecasted demand can be achieved once more information becomes more available over time.

4.2.3 Computational results and analysis for the value of the stochastic solution (VSS)

In this section, we discuss possible benefits we can obtain by solving the stochastic model. In particular, we examine the value of incorporating stochastic information to determine the stochastic solutions in the MRP environment. Based in the instances generation, we get the objective values, VSS and the percentage of savings. These solution results are in the Table 8.

As it is shown in the Table 8, under most of the instances, stochastic model can provide some cost savings compared to the deterministic model with safety stock. According to our solution results, our stochastic model can provide an average potential saving of 5% in the MRP environment under demand uncertainty. The impact of the BOM on the VSS and efficiency differs with the change of the cost parameter.

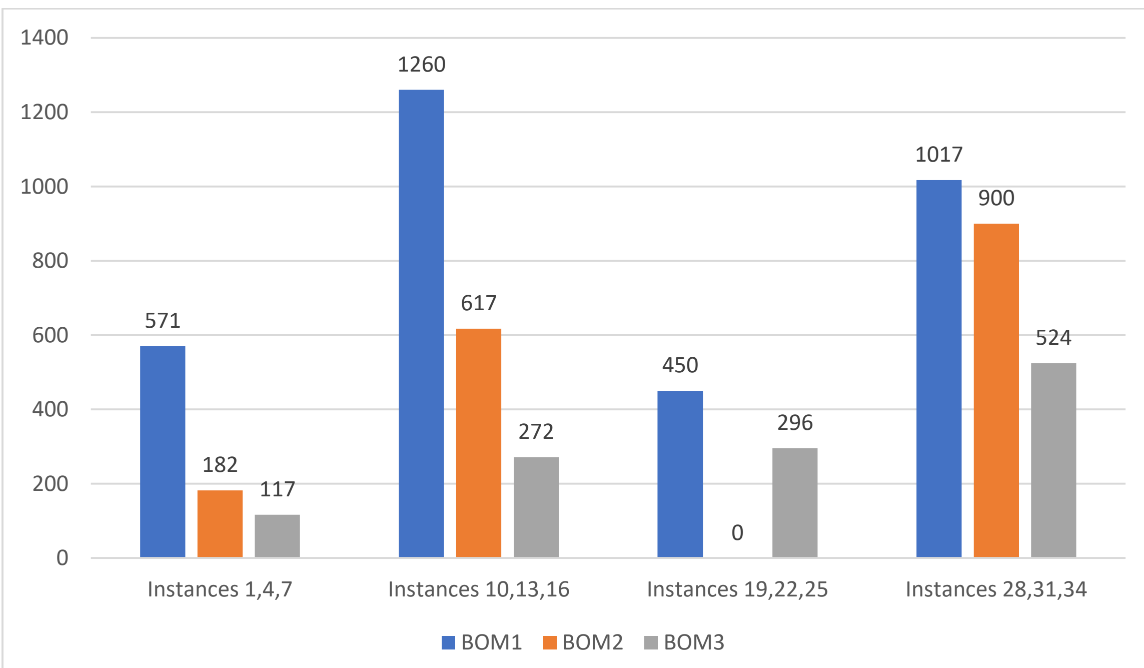

Figure 15 Bar chart of VSS for instances with standard holding cost

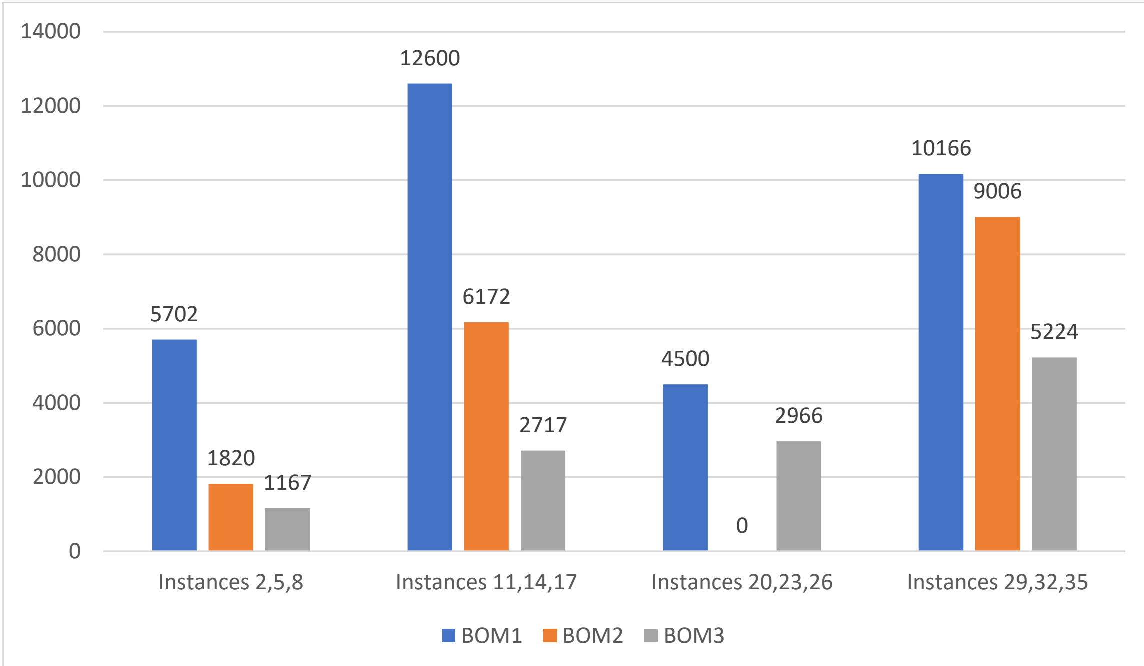

Figure 16 Bar chart of VSS for instances with holding cost x 10

As it is shown in the Figure 15 and Figure 16, generally, for most of instances with standard holding cost and ten times the standard holding cost, under the same parameters, BOM1 always has the highest VSS which is followed by the BOM2 and BOM3. Therefore, it is illustrated that it is more important for products with less levels of BOMs to solve the stochastic model as the stochastic model can provide more cost savings for these products.

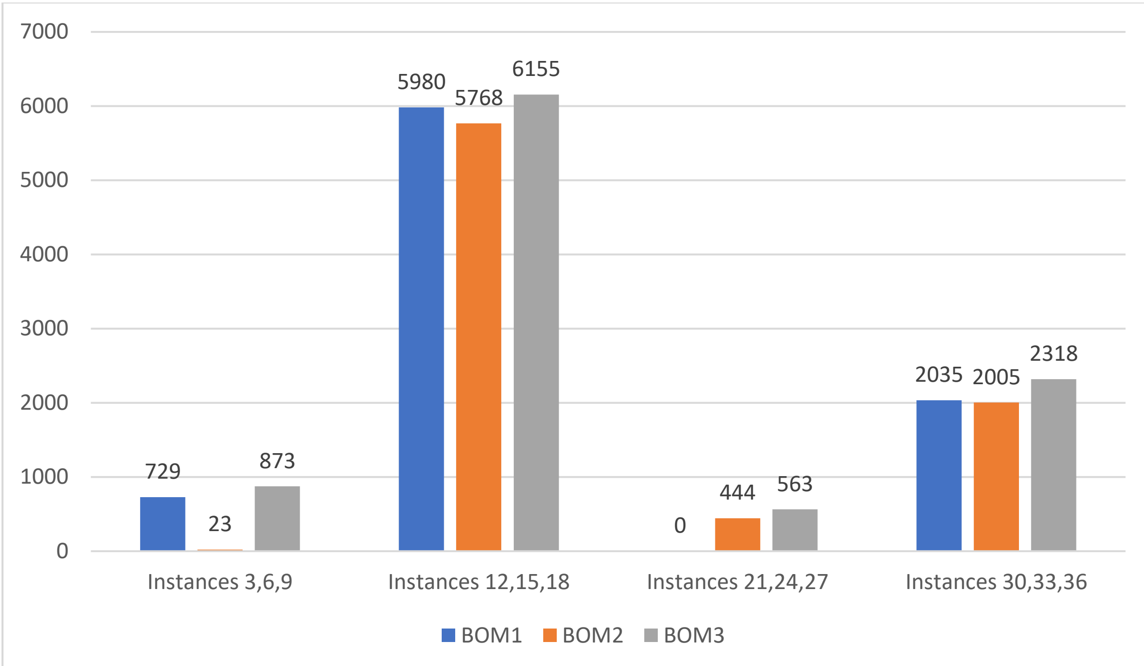

Figure 17 Bar chart of VSS for instances with stock out cost x 10

After we change the stock out cost from the standard stock out cost to ten times the standard stock out cost, as we can see from the Figure 17, the differences between different BOMs become smaller. For most of the instances, increasing the stock out cost reduces the VSS of different BOMs. Thus, under high stock out cost, the BOM structure has little impact on the VSS.

For some instances, there are no VSS values. As we can see from Figure 15 to Figure 17, the value of VSS for instance 21, 22 and 23 are zero. This is because the OBJ_{stoc} and the $OBJ_{\exp\_deter}$ for these three instances are the same.

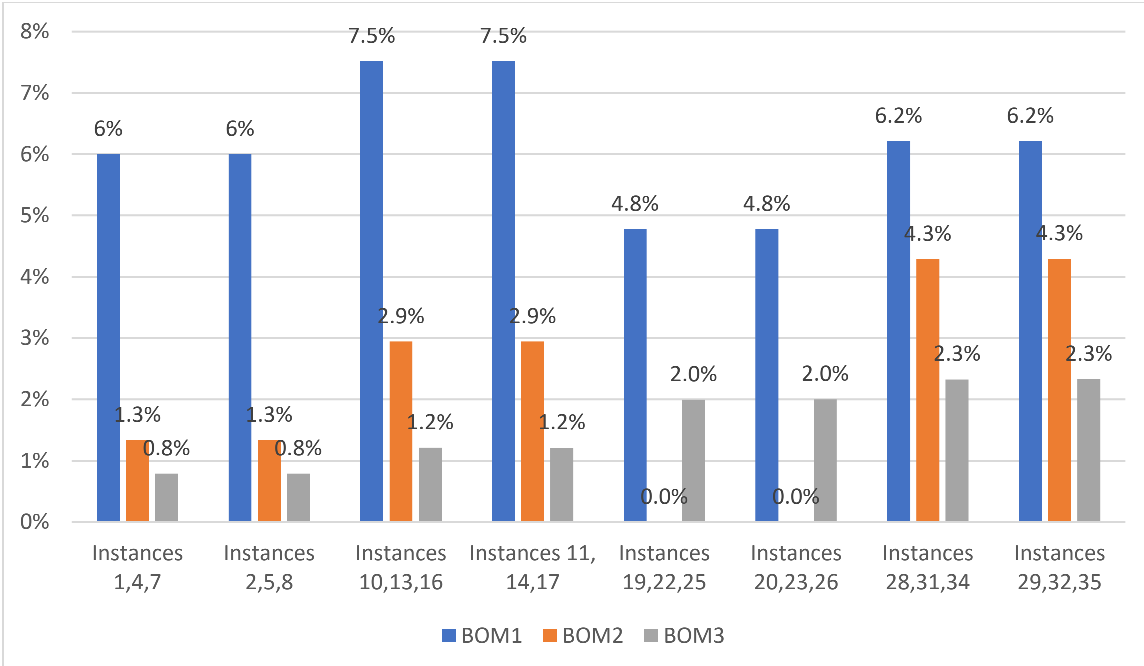

Figure 18 Bar chart of VSS efficiency for instances with standard holding cost and ten times the standard holding cost

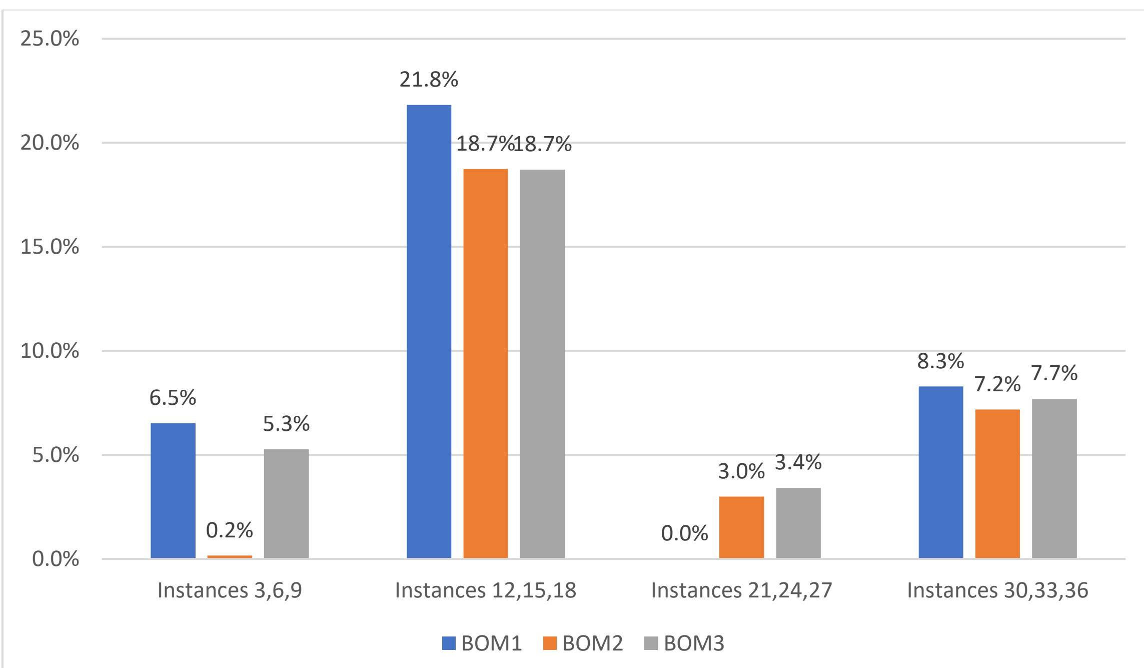

Figure 19 Bar chart of VSS efficiency for instances with stock out x 10

The results from the Figure 18 and Figure 19 demonstrate similar results with Figure 15 and Figure 16. From Figure 18 we can see that, no matter how the holding cost changes, with low stock out cost, BOM structure has a significant impact on the VSS efficiency. Under the same situation, products with less BOM levels tend to have larger VSS efficiency. It means that it is more important for products with less BOM levels to solve the stochastic model as they can get more percentages of savings.

Additionally, from the Table 8 we can see that instance 1 and instance 2 have the same VSS efficiency value. This is the same for all the instances with corresponding standard holding cost and ten times the standard holding cost. Therefore, we can conclude that, generally, other parameters remaining unchanged, the variation of holding cost generally has no impact on the VSS efficiency.

Chapter 5 - Conclusion

In this thesis, a MRP deterministic model and a stochastic model are developed. These MRP optimization models are the generalization of the basic MRP optimization model and the main objective is to minimize the sum of the inventory holding cost and the stock out cost. The main inputs of the models are based on the MRP data which include BOM, lead time, initial inventory, and demand forecast. In the stochastic model, we incorporated demand uncertainty by using a set of different demand scenarios as commonly used in stochastic programming, where each demand scenario is associated with a certain probability of occurrence. The instance set was designed by changing the combinations and the values of different parameters. We performed computational experiments to evaluate the performance of both deterministic and stochastic models. The following conclusions were made from the results obtained. First, the solutions of the deterministic optimization model are very sensitive to the change of demand values. When each of the demand scenarios in the scenario set is used to determine the MRP plan, the total cost fluctuates dramatically over different demand scenarios. The larger the demand uncertainty is, the more the cost is affected. Thus, the demand uncertainty has a significant impact on the total production cost. Second, based on the analysis of the EVPI and VSS from two groups of data (one with 100 demand instances, the other with 1000 demand instances), it is shown that our stochastic model can effectively capture the presence of uncertainty in the MRP environment and provide better solutions compared to the deterministic model with safety stock. In our experiments, we see that significant savings (approximately 5% on average based on the value of stochastic solution) can be achieved if the MRP plan is executed based on the solution from the stochastic model. Through the analysis of the EVPI and VSS, management insights are provided to the decision makers.

There are also some limitations about this thesis. First, in our model, as in the standard MRP logic, we do not impose a capacity limitation in both our deterministic optimization model and the stochastic model and we can produce any quantity of batches. However, this can be easily incorporated by simply adding a capacity constrain into the model. In addition, our stochastic model is a two-stage model. We make decisions about the quantity of batches for all the items and components to produce at the first stage. No further decisions are taken based on the observed demand realizations. This way, the production decisions remain fixed and no recourse decisions could be made. It would be interesting to take the outsourcing and the production recourse decision into the future research.

Bibliography

Arnold, J. T., Chapman, S. N., & Clive, L. M. (2012). *Introduction to Materials Management.* Upper Saddle River: Person Education.

Bai, X., Davis, J. S., Canet, J. J., Cantrell, S., & Patterson, J. W. (2002). Schedule instability, service level and cost in a material requirements planning system. *International Journal of Production Research*, 1725-1758.

Batun, S. (2012). *Scheduling mutiple operating room under uncertainty.* Ann Arbor: UMI Dissertation Publishing .

Beasley, J. E. (n.d.). *OR-Notes*. Retrieved from Brunel University Web Site: http://people.brunel.ac.uk/~mastjjb/jeb/or/sp.html

Bozorgi-Amiri, A., Jabalameli, M. S., & Al-e-Hashem, S. M. (2013). A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty. *OR Sprctrum* , 905-933.

Bregman, R. L. (1992). An analytical framwork for comparing Material Requirements Planning to reorder ponits systems. *European Journal of Operation Research* , 74-88.

Bregni, A., D'Avino, M., & Schiraldi, M. (2011). A new approach to lower MRP nervousness. *Annals & Proceeding of DAAAM International*, 1513-1514.

Chang, M.-S., Tseng, Y.-Y., & Chen, J.-W. (2007). A scenarion planning approach for the flood emergency logistics preparation problem under uncertainty. *Transportation Research* , 737-754.

Chopra, S., & Meindl, P. (2013). *Supply Chain Management - Strategy, Planning, and Operation* (5th Edition ed.). Upper Saddle River, New Jersy, United States of America: Pearson Education.

Chouinard, M., D'Amour, S., & Aït-Kadi, D. (2008). A stochastic programming approach for designing supply loops. *International Journal of Production Economics* , 657-677.

Dolgui, A., & Ould-Louly, M.-A. (2002). A model for supply planning under lead time uncertainty . *International Journal of Produciton Economics*, 145-152.

Dolgui, A., & Prodhon, C. (2007). Supply planning under uncertainties in MRP environments: A state of the art. *Annual Review in Control*, 269-279.

EI-Sayed, M., Afia, N., & EI-Kharbotly, A. (2010). A stochastic model for forward-reverse logistics network design under risk. *Computers & Industrial Engeering*, 423-431.

Grubbström, R. W. (1998). A Net Present Value Approach to Safety Stocks in Planned Production. *International Journal of Production Economics*, 213-229.

Grubbström, R. W., & Molinder, A. (1996). Safety production plans in MRP-systems using transform methodology. *International Journal of Production Economics*, 297-309.

Grubbström, R. W., & Tang, O. (1999). Further developments on safety stocks in an MRP system applying Laplace tranforms and input-output analysis. *Int. J. Production Economics*, 381-387.

Grubbström, R. W., & Wang, Z. (2003). A stochastic model of multi-level/multi-stage capacity-constrained production-inventory system. *International Journal of Production Economics*, 483-494.

Jeunet, J., & Jonard, N. (2000). Measuring the performance of lot-sizing techniques in uncertain environments. *International Journal of Production Economics* , 197-208.

Jonsson, P., & Mattsson, S. A. (2006). Inventory management practices and their implications on perceived planning performance. *International journal of production research*, 1787-1812.

Jonsson, P., & Mattsson, S.-A. (2006). A longtitude study of material planning applications in manufacturing companies. *International Journal of Operations & Production Management*, 971-995.

Koh, S. C., Saad, S. M., & Jones, M. H. (2002). uncertainty under MRP-planned manufacture: review and categorization. *International Journal of Production Research*, 2399-2421.

Louly, M. A., Dolgui, A., & Hnaien, F. (2008). Optimal supply planning in MRP environments for assembly systems with random component procurement times. *International Journal of Production Research* , 5441-5467.

Murthy, D. N., & Ma, L. (1991). MRP with uncertainty: a review and some extensions. *International Journal of Produciton Economics*, 51-64.

Proud, J. N. (1994). *Master Scheduling: A Practical Guide to Competitive Manufacturing* (1st Edition ed.). Essex Junction, Vermont, United States of America: Oliver Wight Publications.

Ptak, C. A., & Smith, C. (2011). *Orlicky's Material Requirements Planning.* McGraw-Hill.

Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal for Operation Research*, 96-115.

Segerstedt, A. (2006, October). Master Production Scheduling and a comparison of Material Requirements Planning and cover time planning. *International Journal of Production Research*, 3585-3606.

Stadtler, H. (2005). Supply chain management and advanced planning - basics, overview and challenges . *European Journal of Operational Research* , 575-588.

Suwanruji, P., & Enns, S. T. (2006). Evaluating the effects of capacity constraints and demand patterns on supply chain replenishment strategies. *International Journal of Production Research* , 4607-4629.

Turbide, D. A. (1993). *MRP+: The adaption, enchancement, and applocation of MRP II.* New York, New York, United States of America: Industrial Press.

Voß, S., & Woodruff, D. L. (2006). *Introduction to computational optimization models for production planning in a supply chain.* Davis: Springer.

Yeung, J. H., Wong, W. C., & Ma, L. (1998). Parameters affecting the effectiveness of MRP systems: a review. *International Journal of Production Research*, 313-331.

Yu, C.-S., & Li, H.-B. (2000). A robust optimization model for stochastic logistic problem . *International Journal of Production Economics* , 385-397.

Appendix

Appendix A: Python source codes of the deterministic model (in iPython Notebook)

In [ ]:

```
# Gurobi Version 7.5 + Python 2.7.13 / Anaconda 4.4.0 (x86_64)
from gurobi import*
from collections import OrderedDict
import numpy as np
import math

N_SKU=8
N_SC =1000
N_PERIOD=16

time_horizon=["t"+str(i) for i in range(1,N_PERIOD+1)]

import pandas as pd
demand_data = pd.read_csv("demand data.csv",index_col=0)
#print demand_data

demand_master = []
demand_sum = []

max_scenarios = N_SC

for i in range(max_scenarios):
    demand_master.append(demand_data.ix[i+1].tolist())
    if(i==0): demand_sum = demand_master[i]
    else: demand_sum = map(sum, zip(demand_sum, demand_master[i]))
    #print(demand_sum)

demand_avg =  [x/max_scenarios for x in demand_sum]

print(demand_avg)


demand_SKU1=demand_avg[:N_PERIOD]
print(demand_SKU1)[0]

child_master1=OrderedDict([
        ("SKU1","SKU1"),
        ("SKU2","SKU1"),
        ("SKU3","SKU1"),
        ("SKU4","SKU1"),
        ("SKU5","SKU1"),
        ("SKU6","SKU1"),
        ("SKU7","SKU1"),
        ("SKU8","SKU1")])
child_master2=OrderedDict([
        ("SKU1","SKU1"),
        ("SKU2","SKU1"),
        ("SKU3","SKU1"),
        ("SKU4","SKU1"),
        ("SKU5","SKU1"),
        ("SKU6","SKU2"),
        ("SKU7","SKU3"),
        ("SKU8","SKU4")])
child_master3=OrderedDict([
```

```
        ("SKU1","SKU1"),
        ("SKU2","SKU1"),
        ("SKU3","SKU1"),
        ("SKU4","SKU2"),
        ("SKU5","SKU2"),
        ("SKU6","SKU3"),
        ("SKU7","SKU3"),
        ("SKU8","SKU4")])
child=child_master3

holding=dict({
        "SKU1":10.0,
        "SKU2":3.0,
        "SKU3":3.0,
        "SKU4":3.0,
        "SKU5":3.0,
        "SKU6":3.0,
        "SKU7":3.0,
        "SKU8":3.0})
stockout = holding["SKU1"]*10


arcs=[("SKU"+str(i+1),"t"+str(t+1)) for i in range(N_SKU) for t in range(N
_PERIOD)]

SKUs, initial_inventory=multidict({
        "SKU1":150,
        "SKU2":200,
        "SKU3":150,
        "SKU4":150,
        "SKU5":200,
        "SKU6":170,
        "SKU7":160,
        "SKU8":150})

SKUs, lotsize=multidict({
        "SKU1":50,
        "SKU2":30,
        "SKU3":30,
        "SKU4":30,
        "SKU5":30,
        "SKU6":30,
        "SKU7":30,
        "SKU8":30})

SKUs=["SKU"+str(i+1) for i in range(N_SKU)]
SKUs=child.keys()

SKU1_leadtime=1
dem_stdev = np.std(demand_SKU1)
safety_stock = round(1.65*dem_stdev*math.sqrt(SKU1_leadtime)+.5)
print safety_stock
demand_SKU1[SKU1_leadtime] = demand_SKU1[SKU1_leadtime] + safety_stock

print demand_SKU1
print stockout
```

In [ ]:
```
m=Model("Toy")

inv_obj=[holding[i[0]] for i in arcs]

inventory=m.addVars(arcs,vtype=GRB.INTEGER, lb=0, obj=inv_obj ,name="Inven
tory")
set_up=m.addVars(arcs, vtype=GRB.INTEGER, name="Set_Up")
quantity=m.addVars(arcs, lb=0, name="Quantity")

stockout_obj = [stockout for i in time_horizon]
stockout_qty = m.addVars(time_horizon, vtype=GRB.INTEGER, lb=0, obj=stocko
ut_obj ,name="stockout_quantity")

m.update()
```

In [ ]:
```
#Create constraints for Leve 0 - SKU1  ps: leftover_inventory=the inventor
y left from the previous period
s="SKU1"
for t, val in enumerate(time_horizon):
    #print(t)
    #print(val)
    #Define the leadtime(SKU1)
    leadtime = SKU1_leadtime

    #Define the leftover inventory for t1 and later periods repectively(SK
U1)
    if(t==0):
        leftover_inventory=initial_inventory[s]
    else:
        leftover_inventory=inventory[(s, time_horizon[t-1])]

    #The constraint for SKU1 when t=0 (t1)
    if(t < leadtime):
        m.addConstr(inventory[(s, val)] - leftover_inventory \
                    + demand_SKU1[t]  - stockout_qty[val] == 0.0, name="c_
"+str(s)+"_"+str(t))


    #The constraint for SKU1 when it is > t1
    else:
        m.addConstr(inventory[(s, val)] - leftover_inventory - quantity[(s
, val)] \
                    + demand_SKU1[t] - stockout_qty[val] == 0.0, name="c_"
+str(s)+"_"+str(t))

    #Define quantity received for period t+leadtime = lotsize*setup[t] or
Qt=Lotsize*Yt-1
    if(t + leadtime < len(time_horizon) and t <= N_PERIOD):
        m.addConstr(quantity[(s, time_horizon[t+leadtime])] - lotsize[s]*s
et_up[(s, val)] == 0.0, \
                    name="set_up"+str(s)+"_"+str(t))
```

In [ ]:
```
#Create constraints for Level 1 > SKU1
for s in SKUs[1:]:
    #Define the leadtime for SKUs in level 1  PS:leadtime = lead_time[s]
    leadtime = 1
```

```
    for t, val in enumerate(time_horizon):

        #Define the leftover inventoy for t1 and later periods repectively
 (SKUs in level 1)
        if(t==0):
            leftover_inventory = initial_inventory[s]
        else:
            leftover_inventory = inventory[(s,time_horizon[t-1])]

        #The constraints for SKUs in level 1 when t=0(t1)
        if(t < leadtime):
            m.addConstr(inventory[(s, val)] - leftover_inventory \
                        + quantity[(child[s], time_horizon[t+leadtime])] =
= 0.0, name="c_"+str(s)+"_"+str(t))

        #The comstraints for SKUs in level 1 when >t2 and <Last period
        elif(t + leadtime < len(time_horizon)):
            m.addConstr(inventory[(s, val)] - leftover_inventory - quantit
y[(s, val)] \
                    + quantity[(child[s], time_horizon[t+leadtime])] == 0.
0, name="c_"+str(s)+"_"+str(t))

        #The constraint for SKUs in level 1 when it comes to the last peri
od
        else:
            m.addConstr(inventory[(s, val)] - leftover_inventory - quantit
y[(s, val)] \
                        == 0.0, name="c_"+str(s)+"_"+str(t))

        #Define quantity received for period t+leadtime = lotsize*setup[t]
 or Qt=Lotsize*Yt-1
        if(t + leadtime < len(time_horizon) and t <= N_PERIOD):
            m.addConstr(quantity[(s, time_horizon[t+leadtime])] - lotsize[
s]*set_up[(s, val)] == 0.0, \
                    name="set_up"+str(s)+"_"+str(t))
```

In [ ]:
```
m.Params.timelimit = 3600.0
m.Params.OptimalityTol = 0.01
m.optimize()
```

In [ ]:
```
print([i for i in time_horizon])

import pandas as pd

stockout_values = [stockout_qty[i].x for i in time_horizon]
setup_values = [set_up[j].x for j in arcs]
inventory_values = [inventory[i].x for i in arcs]
index = arcs
print(arcs[:len(time_horizon)])
stockout_df = pd.DataFrame(stockout_values,columns=["stockout_val"], index
 = arcs[:len(time_horizon)])
another_val = pd.DataFrame(setup_values, columns=["setup_val"], index = ar
cs)
third_val = pd.DataFrame(inventory_values, columns=["inventory_val"], inde
x = arcs)
```

```
combined_df = pd.concat([stockout_df,another_val, third_val], axis=1)

print(combined_df)
ind = 2
combined_df.to_csv("DET_solution_"+str(ind)+".csv")

print([stockout_qty[i].x for i in time_horizon])
```

Appendix B Python source codes of the stochastic model (based on iPython Notebook)

In [ ]:
```
# Gurobi Version 7.5 + Python 2.7.13 / Anaconda 4.4.0 (x86_64)
from gurobi import*
from collections import OrderedDict

N_SKU=8
N_SC=1000
N_PERIOD=16
time_horizon=["t"+str(i) for i in range(1,N_PERIOD+1)]
setup_mode = 0 #0 - stochastic, 1- deterministic approximation


import pandas as pd
demand_data = pd.read_csv("demand data.csv",index_col=0)
y_data=pd.read_csv("DET_solution_2.csv", index_col=[0,1])['setup_val']
#print demand_data
#print y_data

#print max(y_data.loc[[('SKU1','t8')]][0],0.0)

demand_master = []
demand_sum = []

max_scenarios = N_SC
for i in range(max_scenarios):
    demand_master.append(demand_data.ix[i+1].tolist())
    if(i==0): demand_sum = demand_master[i]
    else: demand_sum = map(sum, zip(demand_sum, demand_master[i]))
    #print(demand_sum)
#print demand_master
demand_avg =  [x/max_scenarios for x in demand_sum]

#print(demand_avg)


child_master1=OrderedDict([
        ("SKU1","SKU1"),
        ("SKU2","SKU1"),
        ("SKU3","SKU1"),
        ("SKU4","SKU1"),
        ("SKU5","SKU1"),
        ("SKU6","SKU1"),
        ("SKU7","SKU1"),
        ("SKU8","SKU1")])
child_master2=OrderedDict([
        ("SKU1","SKU1"),
        ("SKU2","SKU1"),
        ("SKU3","SKU1"),
        ("SKU4","SKU1"),
        ("SKU5","SKU1"),
        ("SKU6","SKU2"),
        ("SKU7","SKU3"),
        ("SKU8","SKU4")])
child_master3=OrderedDict([
        ("SKU1","SKU1"),
```

```
        ("SKU2","SKU1"),
        ("SKU3","SKU1"),
        ("SKU4","SKU2"),
        ("SKU5","SKU2"),
        ("SKU6","SKU3"),
        ("SKU7","SKU3"),
        ("SKU8","SKU4")])
child=child_master1

holding=dict({
        "SKU1":10.0,
        "SKU2":3.0,
        "SKU3":3.0,
        "SKU4":3.0,
        "SKU5":3.0,
        "SKU6":3.0,
        "SKU7":3.0,
        "SKU8":3.0

        })
stockout = holding["SKU1"]*10



# for setup variables (first stage decisin)
deterministic_arcs=[("SKU"+str(i+1),"t"+str(t+1),"SC1") for i in range(N_S
KU) for t in range(N_PERIOD)]
# for resource decisions
arcs=[("SKU"+str(i+1),"t"+str(t+1),"SC"+str(s+1)) for i in range(N_SKU) fo
r t in range(N_PERIOD) for s in range (N_SC)]
# for stockout
SKU1_arcs=[("SKU1","t"+str(t+1),"SC"+str(s+1)) for t in range(N_PERIOD) fo
r s in range (N_SC)]

#print deterministic_arcs[1][0]

y_lb = []
y_ub = []
for i in range (len(deterministic_arcs)):
    y_lb.append(min(y_data.loc[[(deterministic_arcs[i][0],deterministic_ar
cs[i][1])]][0]-0.01,0.0))
    y_ub.append(max(y_data.loc[[(deterministic_arcs[i][0],deterministic_ar
cs[i][1])]][0]+0.01,0.0))

#print y_lb
#print y_ub

SKUs, initial_inventory=multidict({
        "SKU1":150,
        "SKU2":200,
        "SKU3":150,
        "SKU4":150,
        "SKU5":200,
        "SKU6":170,
        "SKU7":160,
        "SKU8":150})
```

```
SKUs, lotsize=multidict({
        "SKU1":50,
        "SKU2":30,
        "SKU3":30,
        "SKU4":30,
        "SKU5":30,
        "SKU6":30,
        "SKU7":30,
        "SKU8":30})

SKUs=["SKU"+str(i+1) for i in range(N_SKU)]
prob = 1.0/N_SC
#print(prob)

SCs = ["SC"+str(i+1) for i in range(N_SC)]
probability = [prob for i in range(N_SC)]

#print (deterministic_arcs)

#print (len(deterministic_arcs))
```

In [ ]:

```
m=Model("Toy")

#inv_obj=[holding[i[0]]*probability[w] for i in arcs ]
inv_obj=[holding[i[0]]/N_SC for i in arcs]
#print(inv_obj)
inventory=m.addVars(arcs,lb=0.0, obj=inv_obj ,name="Inventory")



stockout_obj = [stockout*probability[w] for i in time_horizon for w in ran
ge(N_SC)]

#print(stockout_obj)

stockout_qty = m.addVars(SKU1_arcs, lb=0.0, obj=stockout_obj ,name="stocko
ut_quantity")

print setup_mode
if(setup_mode == 0): set_up=m.addVars(deterministic_arcs, lb=0.0, vtype=GR
B.INTEGER, name="Set_Up")
elif(setup_mode == 1): set_up=m.addVars(deterministic_arcs, lb=y_lb, ub=y_
ub, vtype=GRB.INTEGER, name="Set_Up")

quantity=m.addVars(deterministic_arcs, lb=0.0, name="Quantity")


m.update()
```

In [ ]:

```
#Create constraints for Leve 0 - SKU1  ps: leftover_inventory=the inventor
y left from the previous period
s="SKU1"
for num, sc in enumerate(SCs):
    #print(num)
    #print(sc)
    #print(demand_master[num][3])
```

```
    for t, val in enumerate(time_horizon):

        leadtime=1

        #Define the leftover inventory for t1 and later periods repectivel
y(SKU1)
        if(t==0):
            leftover_inventory=initial_inventory[s]
        else:
            leftover_inventory=inventory[(s, time_horizon[t-1],sc)]


        #print(stockout_qty)
        #The constraint **************** (SKU1, t1)
        if(t < leadtime):
            m.addConstr(inventory[(s, val, sc)] - leftover_inventory \
                        + demand_master[num][t]  - stockout_qty[("SKU1", v
al,sc)] == 0.0, \
                        name="c_"+str(s)+"_"+str(val)+"_"+str(sc))

        #The constraint ************* (SKU1, t2t3t4.....)
        else:
            m.addConstr(inventory[(s, val,sc)] - leftover_inventory - quan
tity[(s, val,"SC1")] \
                        + demand_master[num][t] - stockout_qty[("SKU1", va
l,sc)] == 0.0, \
                        name="c_"+str(s)+"_"+str(val)+"_"+str(sc))



        ###
        #Define quantity received for period t+leadtime = lotsize*setup[t]
 or Qt=Lotsize*Yt-1    *******(SKU1, quan received)
        if(t + leadtime < len(time_horizon) and t <= N_PERIOD):
            #print(quantity[(s, time_horizon[t+leadtime],"SC1")])
            m.addConstr(quantity[(s, time_horizon[t+leadtime],"SC1")] - lo
tsize[s]*set_up[(s, val, "SC1")] == 0.0, \
                        name="set_up"+str(s)+"_"+str(val)+"_"+str(sc))
```

In [ ]:

```
print SKUs
#Create constraints for Level 1 > SKU1
for s in SKUs[1:]:
    #Define the leadtime for SKUs in level 1  PS:leadtime = lead_time[s]

    leadtime = 1
    for num, sc in enumerate(SCs):

        for t, val in enumerate(time_horizon):



            #Define the leftover inventoy for t1 and later periods repecti
vely (SKUs in level 1)
            if(t==0):
                leftover_inventory = initial_inventory[s]
            else:
```

```
                leftover_inventory = inventory[(s,time_horizon[t-1],sc)]


            #print initial_inventory[s]


            #The constraints *******SKU2SKU3SKU4....t1
            if(t < leadtime):
                m.addConstr(inventory[(s, val,sc)] - leftover_inventory \
                      + quantity[(child[s], time_horizon[t+leadtime]
,"SC1")] == 0.0, name="c_"+str(s)+"_"+str(val)+"_"+str(sc))

            #The comstraints *******SKU2SKU3SKU4.... >t2 and <Last period
            elif(t + leadtime < len(time_horizon)):
                m.addConstr(inventory[(s, val,sc)] - leftover_inventory -
quantity[(s, val,"SC1")] \
                    + quantity[(child[s], time_horizon[t+leadtime],"SC
1")] == 0.0, name="c_"+str(s)+"_"+str(val)+"_"+str(sc))

            #The constraint ********SKU2SKU3SKU4.... last period
            else:
                m.addConstr(inventory[(s, val,sc)] - leftover_inventory -
quantity[(s, val,"SC1")] \
                      == 0.0, name="c_"+str(s)+"_"+str(val)+"_"+str(
sc))


            #Define quantity received for period t+leadtime = lotsize*setu
p[t] or Qt=Lotsize*Yt-1
            if(t + leadtime < len(time_horizon) and t <= N_PERIOD):
                #print(quantity[(s, time_horizon[t+leadtime],"SC1")])
                m.addConstr(quantity[(s, time_horizon[t+leadtime],"SC1")]
- lotsize[s]*set_up[(s, val, "SC1")] == 0.0, \
                    name="setup_"+str(s)+"_"+str(val)+"_"+str(sc))
```

In [ ]:
```
m.ModelSense=GRB.MINIMIZE
#m.setParam('TimeLimit',10.0)
m.Params.timelimit = 9600.0
m.Params.OptimalityTol = 0.01
m.optimize()
```

In [ ]:
```
print([i for i in time_horizon])

import pandas as pd

#print([j for j in SKU1_arcs])
print(stockout_qty[('SKU1', 't1', 'SC2')].x)

stockout_values = [stockout_qty[j].x for j in SKU1_arcs]
setup_values = [set_up[j].x for j in deterministic_arcs]
quantity_values = [quantity[j].x for j in deterministic_arcs]
inventory_values = [inventory[i].x for i in arcs]
index = arcs
print(arcs[:len(time_horizon)])
stockout_df = pd.DataFrame(stockout_values,columns=["stockout_val"], index
 = SKU1_arcs)
```

```
setup_df = pd.DataFrame(setup_values, columns=["setup_val"], index = deter
ministic_arcs)
inventory_df = pd.DataFrame(inventory_values, columns=["inventory_val"], i
ndex = arcs)
quantity_df = pd.DataFrame(quantity_values, columns=["qty_val"], index = d
eterministic_arcs)

combined_df = pd.concat([stockout_df,setup_df, inventory_df, quantity_df],
 axis=1)

print(combined_df)
ind = 2
if(setup_mode==0): combined_df.to_csv("STOC_solution_"+str(ind)+".csv")

if(setup_mode==1): combined_df.to_csv("STOCDET_solution_"+str(ind)+".csv")
```

Appendix C Objective values based on the deterministic model for instances 1-36. For instances from 19-36, we present only the results for the first 100 demand scenarios.

Instance	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12	SC13	SC14	SC15	SC16	SC17	SC18	SC19	SC20	SC21	SC22	SC23	SC24	SC25
Instance 1	6920	5620	4210	5000	6020	6040	5610	5250	4540	5980	5890	8830	7720	4520	4650	4920	4740	6340	4690	5050	7040	5480	6730	9360	5740
Instance 2	69200	56200	42100	50000	60200	60400	56100	52500	45400	59800	58900	88300	77200	45200	46500	49200	47400	63400	46900	50500	70400	54800	67300	93600	57400
Instance 3	6920	5620	4840	5000	6020	6040	5690	5250	4540	5980	5890	8830	7820	5180	4650	4920	5010	6340	4690	5050	7040	5480	6730	9490	5850
Instance 4	14210	8680	6190	6980	11030	10240	9630	8940	7960	9660	9570	14200	12290	7970	7710	8600	8430	11350	8890	8110	12050	10490	12740	15240	9170
Instance 5	142100	86800	61900	69800	110300	102400	96300	89400	79600	96600	95700	142000	122900	79700	77100	86000	84300	113500	88900	81100	120500	104900	127400	152400	91700
Instance 6	14210	8680	6190	6980	11030	10240	9630	8940	7960	9660	9570	14200	12290	7970	7710	8600	8520	11350	8890	8110	12050	10490	12740	15240	9170
Instance 7	14110	10350	7590	8370	11970	11190	10960	10570	9710	11210	10920	13820	12550	9810	9580	10150	10090	11890	10040	9260	12590	11230	12900	14590	9870
Instance 8	141100	103500	75900	83700	119700	111900	109600	105700	97100	112100	109200	138200	125500	98100	95800	101500	100900	118900	100400	92600	125900	112300	129000	145900	98700
Instance 9	14110	10350	7630	8370	11970	11190	10980	10600	9710	11210	10920	13820	12550	9870	9580	10150	10240	11890	10040	9260	12590	11230	12960	14660	9920
Instance 10	10640	8810	7730	8230	9650	9780	9140	8660	7970	9680	9000	12870	11100	8560	7900	8350	8400	9300	8180	8310	10440	8750	9840	13130	8850
Instance 11	106400	88100	77300	82300	96500	97800	91400	86600	79700	96800	90000	128700	111000	85600	79000	83500	84000	93000	81800	83100	104400	87500	98400	131300	88500
Instance 12	10640	9470	8420	8230	9650	9780	9560	9420	8420	9680	9000	13040	11100	8990	8220	8820	9130	9820	8620	8310	10760	9130	9840	13400	8880
Instance 13	18250	11870	9620	10210	14660	13980	13170	12350	11390	13360	12680	18250	15570	11690	10960	12030	12000	14310	12380	11370	15450	13670	16280	19050	12200
Instance 14	182500	118700	96200	102100	146600	139800	131700	123500	113900	133600	126800	182500	155700	116900	109600	120300	120000	143100	123800	113700	154500	136700	162800	190500	122000
Instance 15	18250	12530	9770	10210	14660	13980	13500	13020	11840	13360	12680	18410	15570	11690	11280	12500	12640	14830	12820	11370	15770	14050	16280	19130	12200
Instance 16	20850	13870	11100	11750	16050	15020	14760	14210	13830	15240	14300	18870	16460	14520	12920	13670	14080	16080	13590	12640	16380	15100	17850	19120	12920
Instance 17	208500	138700	111000	117500	160500	150200	147600	142100	138300	152400	143000	188700	164600	145200	129200	136700	140800	160800	135900	126400	163800	151000	178500	191200	129200
Instance 18	20850	14500	11270	11750	16050	15020	15120	15010	14040	15240	14300	18870	16460	14520	13180	14080	14750	16480	13790	12640	16730	15420	17850	19210	12920
Instance 19	6920	5620	4210	5000	6020	6040	5610	5250	4540	5980	5890	8830	7720	4520	4650	4920	4740	6340	4690	5050	7040	5480	6730	9360	5740
Instance 20	69200	56200	42100	50000	60200	60400	56100	52500	45400	59800	58900	88300	77200	45200	46500	49200	47400	63400	46900	50500	70400	54800	67300	93600	57400
Instance 21	6920	5620	4840	5000	6020	6040	5690	5250	4540	5980	5890	8830	7820	5180	4650	4920	5010	6340	4690	5050	7040	5480	6730	9490	5850
Instance 22	14210	8680	6190	6980	11030	10240	9630	8940	7960	9660	9570	14200	12290	7970	7710	8600	8430	11350	8890	8110	12050	10490	12740	15240	9170
Instance 23	142100	86800	61900	69800	110300	102400	96300	89400	79600	96600	95700	142000	122900	79700	77100	86000	84300	113500	88900	81100	120500	104900	127400	152400	91700
Instance 24	14210	8680	6190	6980	11030	10240	9630	8940	7960	9660	9570	14200	12290	7970	7710	8600	8520	11350	8890	8110	12050	10490	12740	15240	9170
Instance 25	14110	10350	7590	8370	11970	11190	10960	10570	9710	11210	10920	13820	12550	9810	9580	10150	10090	11890	10040	9260	12590	11230	12900	14590	9870
Instance 26	141100	103500	75900	83700	119700	111900	109600	105700	97100	112100	109200	138200	125500	98100	95800	101500	100900	118900	100400	92600	125900	112300	129000	145900	98700
Instance 27	14110	10350	7630	8370	11970	11190	10980	10600	9710	11210	10920	13820	12550	9870	9580	10150	10240	11890	10040	9260	12590	11230	12960	14660	9920
Instance 28	10640	8810	7730	8230	9650	9780	9140	8660	7970	9680	9000	12870	11100	8560	7900	8350	8400	9300	8180	8310	10440	8750	9840	13130	8850
Instance 29	106400	88100	77300	82300	96500	97800	91400	86600	79700	96800	90000	128700	111000	85600	79000	83500	84000	93000	81800	83100	104400	87500	98400	131300	88500
Instance 30	10640	9470	8420	8230	9650	9780	9560	9420	8420	9680	9000	13040	11100	8990	8220	8820	9130	9820	8620	8310	10760	9130	9840	13400	8880
Instance 31	18250	11870	9620	10210	14660	13980	13170	12350	11390	13360	12680	18250	15570	11690	10960	12030	12000	14310	12380	11370	15450	13670	16280	19050	12200
Instance 32	182500	118700	96200	102100	146600	139800	131700	123500	113900	133600	126800	182500	155700	116900	109600	120300	120000	143100	123800	113700	154500	136700	162800	190500	122000
Instance 33	18250	12530	9770	10210	14660	13980	13500	13020	11840	13360	12680	18410	15570	11690	11280	12500	12640	14830	12820	11370	15770	14050	16280	19130	12200
Instance 34	20850	13870	11100	11750	16050	15020	14760	14210	13830	15240	14300	18870	16460	14520	12920	13670	14080	16080	13590	12640	16380	15100	17850	19120	12920
Instance 35	208500	138700	111000	117500	160500	150200	147600	142100	138300	152400	143000	188700	164600	145200	129200	136700	140800	160800	135900	126400	163800	151000	178500	191200	129200
Instance 36	20850	14500	11270	11750	16050	15020	15120	15010	14040	15240	14300	18870	16460	14520	13180	14080	14750	16480	13790	12640	16730	15420	17850	19210	12920

Instance	SC26	SC27	SC28	SC29	SC30	SC31	SC32	SC33	SC34	SC35	SC36	SC37	SC38	SC39	SC40	SC41	SC42	SC43	SC44	SC45	SC46	SC47	SC48	SC49	SC50
Instance 1	7200	6710	4920	5660	5900	6920	3910	7070	7560	5450	6190	5560	6660	5840	5710	4580	4700	7290	6450	6640	4920	4800	6090	4480	6830
Instance 2	72000	67100	49200	56600	59000	69200	39100	70700	75600	54500	61900	55600	66600	58400	57100	45800	47000	72900	64500	66400	49200	48000	60900	44800	68300
Instance 3	7700	6910	4920	5660	6170	6920	4710	7070	8300	5450	6190	5560	7040	5840	5710	4580	4700	7290	6530	6640	4920	5120	6090	4880	6830
Instance 4	13450	11920	6900	9240	11540	12290	7360	12080	14590	9940	11200	9500	13050	10850	10720	8000	7760	13040	11000	11650	8340	7100	10290	6020	11840
Instance 5	134500	119200	69000	92400	115400	122900	73600	120800	145900	99400	112000	95000	130500	108500	107200	80000	77600	130400	110000	116500	83400	71000	102900	60200	118400
Instance 6	13450	11920	6900	9240	11540	12290	7770	12080	14590	9940	11200	9500	13050	10850	10720	8000	7760	13040	11000	11650	8340	7100	10290	6320	11840
Instance 7	13390	12660	8290	10510	12160	12710	9140	12420	14360	11080	11540	10850	13150	11590	11460	9750	8910	13260	11860	12390	9370	9210	11240	8070	12380
Instance 8	133900	126600	82900	105100	121600	127100	91400	124200	143600	110800	115400	108500	131500	115900	114600	97500	89100	132600	118600	123900	93700	92100	112400	80700	123800
Instance 9	13470	12660	8290	10510	12160	12710	9160	12420	14490	11080	11540	10850	13270	11590	11460	9750	8910	13260	11860	12390	9370	9210	11240	8070	12380
Instance 10	10960	10120	8260	8900	9860	10630	7520	10530	11130	8810	10140	8940	10320	9000	9430	8470	8060	10290	10340	10080	8230	8170	9410	8450	10270
Instance 11	109600	101200	82600	89000	98600	106300	75200	105300	111300	88100	101400	89400	103200	90000	94300	84700	80600	102900	103400	100800	82300	81700	94100	84500	102700
Instance 12	11820	10380	8260	9160	10420	10750	8590	10530	12290	9080	10250	8940	10540	9050	9430	8470	8560	10290	10380	10080	8230	9270	9410	8450	10270
Instance 13	17680	15390	10240	12480	15590	16040	11070	15540	18610	13210	15150	12880	16980	14010	14440	11890	11120	16470	14850	15090	11650	10610	13610	9890	15280
Instance 14	176800	153900	102400	124800	155900	160400	110700	155400	186100	132100	151500	128800	169800	140100	144400	118900	111200	164700	148500	150900	116500	106100	136100	98900	152800
Instance 15	18000	15390	10240	12740	15790	16120	11650	15540	19010	13480	15170	12880	16980	14060	14440	11890	11620	16470	14850	15090	11650	11250	13610	9890	15280
Instance 16	19260	17210	11780	13870	17200	17440	12930	16390	19960	14620	15600	14500	19120	15970	15540	13520	12300	18220	16940	17000	12830	12860	14740	11730	16720
Instance 17	192600	172100	117800	138700	172000	174400	129300	163900	199600	146200	156000	145000	191200	159700	155400	135200	123000	182200	169400	170000	128300	128600	147400	117300	167200
Instance 18	19550	17210	11780	13890	17460	17440	13130	16390	20470	14740	15600	14500	19120	15970	15540	13520	12740	18220	16940	17000	12830	13330	14740	11730	16720
Instance 19	7200	6710	4920	5660	5900	6920	3910	7070	7560	5450	6190	5560	6660	5840	5710	4580	4700	7290	6450	6640	4920	4800	6090	4480	6830
Instance 20	72000	67100	49200	56600	59000	69200	39100	70700	75600	54500	61900	55600	66600	58400	57100	45800	47000	72900	64500	66400	49200	48000	60900	44800	68300
Instance 21	7700	6910	4920	5660	6170	6920	4710	7070	8300	5450	6190	5560	7040	5840	5710	4580	4700	7290	6530	6640	4920	5120	6090	4880	6830
Instance 22	13450	11920	6900	9240	11540	12290	7360	12080	14590	9940	11200	9500	13050	10850	10720	8000	7760	13040	11000	11650	8340	7100	10290	6020	11840
Instance 23	134500	119200	69000	92400	115400	122900	73600	120800	145900	99400	112000	95000	130500	108500	107200	80000	77600	130400	110000	116500	83400	71000	102900	60200	118400
Instance 24	13450	11920	6900	9240	11540	12290	7770	12080	14590	9940	11200	9500	13050	10850	10720	8000	7760	13040	11000	11650	8340	7100	10290	6320	11840
Instance 25	13390	12660	8290	10510	12160	12710	9140	12420	14360	11080	11540	10850	13150	11590	11460	9750	8910	13260	11860	12390	9370	9210	11240	8070	12380
Instance 26	133900	126600	82900	105100	121600	127100	91400	124200	143600	110800	115400	108500	131500	115900	114600	97500	89100	132600	118600	123900	93700	92100	112400	80700	123800
Instance 27	13470	12660	8290	10510	12160	12710	9160	12420	14490	11080	11540	10850	13270	11590	11460	9750	8910	13260	11860	12390	9370	9210	11240	8070	12380
Instance 28	10960	10120	8260	8900	9860	10630	7520	10530	11130	8810	10140	8940	10320	9000	9430	8470	8060	10290	10340	10080	8230	8170	9410	8450	10270
Instance 29	109600	101200	82600	89000	98600	106300	75200	105300	111300	88100	101400	89400	103200	90000	94300	84700	80600	102900	103400	100800	82300	81700	94100	84500	102700
Instance 30	11820	10380	8260	9160	10420	10750	8590	10530	12290	9080	10250	8940	10540	9050	9430	8470	8560	10290	10380	10080	8230	9270	9410	8450	10270
Instance 31	17680	15390	10240	12480	15590	16040	11070	15540	18610	13210	15150	12880	16980	14010	14440	11890	11120	16470	14850	15090	11650	10610	13610	9890	15280
Instance 32	176800	153900	102400	124800	155900	160400	110700	155400	186100	132100	151500	128800	169800	140100	144400	118900	111200	164700	148500	150900	116500	106100	136100	98900	152800
Instance 33	18000	15390	10240	12740	15790	16120	11650	15540	19010	13480	15170	12880	16980	14060	14440	11890	11620	16470	14850	15090	11650	11250	13610	9890	15280
Instance 34	19260	17210	11780	13870	17200	17440	12930	16390	19960	14620	15600	14500	19120	15970	15540	13520	12300	18220	16940	17000	12830	12860	14740	11730	16720
Instance 35	192600	172100	117800	138700	172000	174400	129300	163900	199600	146200	156000	145000	191200	159700	155400	135200	123000	182200	169400	170000	128300	128600	147400	117300	167200
Instance 36	19550	17210	11780	13890	17460	17440	13130	16390	20470	14740	15600	14500	19120	15970	15540	13520	12740	18220	16940	17000	12830	13330	14740	11730	16720

Instance	SC51	SC52	SC53	SC54	SC55	SC56	SC57	SC58	SC59	SC60	SC61	SC62	SC63	SC64	SC65	SC66	SC67	SC68	SC69	SC70	SC71	SC72	SC73	SC74	SC75
Instance 1	5710	5080	5290	7910	5250	8860	4350	7800	6060	5100	4330	7230	7150	4560	6820	5880	5560	5930	5070	7200	4870	4590	5010	4920	4620
Instance 2	57100	50800	52900	79100	52500	88600	43500	78000	60600	51000	43300	72300	71500	45600	68200	58800	55600	59300	50700	72000	48700	45900	50100	49200	46200
Instance 3	5710	5680	5430	7910	5250	8860	4550	7800	6330	5100	4330	7230	7150	4560	6820	5880	5560	5930	5070	7200	5390	4750	5010	5420	5280
Instance 4	10720	8240	6780	12920	7140	14870	6870	12810	10530	6540	7390	12240	12520	6000	11830	10710	7450	10130	8750	12950	8190	6480	8070	8340	7800
Instance 5	107200	82400	67800	129200	71400	148700	68700	128100	105300	65400	73900	122400	125200	60000	118300	107100	74500	101300	87500	129500	81900	64800	80700	83400	78000
Instance 6	10720	8740	6780	12920	7140	14870	7070	12810	10620	6540	7390	12240	12520	6000	11830	10710	7450	10130	8750	12950	8710	6640	8070	8840	7800
Instance 7	11660	9630	8520	13060	8560	14290	7890	12950	11190	8050	8440	12580	12940	7710	12370	11510	8420	11080	10100	13170	9860	7650	9220	9610	9730
Instance 8	116600	96300	85200	130600	85600	142900	78900	129500	111900	80500	84400	125800	129400	77100	123700	115100	84200	110800	101000	131700	98600	76500	92200	96100	97300
Instance 9	11660	9890	8520	13060	8560	14290	8090	12950	11340	8050	8440	12580	12940	7710	12370	11510	8420	11080	10100	13170	9900	7690	9220	9870	9790
Instance 10	8540	8500	8310	11920	8380	12490	7550	11110	9760	7790	7830	10740	10390	8080	10380	9890	9360	9300	8520	10350	7940	8270	8150	8360	8120
Instance 11	85400	85000	83100	119200	83800	124900	75500	111100	97600	77900	78300	107400	103900	80800	103800	98900	93600	93000	85200	103500	79400	82700	81500	83600	81200
Instance 12	8540	9630	8520	11940	9510	12510	8330	11520	10010	7790	7830	10740	10390	8080	10380	9890	9620	9570	8520	10650	9360	8330	8980	9080	8540
Instance 13	13640	11560	9870	16930	10620	18930	10070	16120	14140	9230	10890	15750	15760	9520	15390	14720	11250	13500	12200	16440	11260	10160	11210	11780	11060
Instance 14	136400	115600	98700	169300	106200	189300	100700	161200	141400	92300	108900	157500	157600	95200	153900	147200	112500	135000	122000	164400	112600	101600	112100	117800	110600
Instance 15	13640	12690	9870	16950	11400	18950	10850	16530	14300	9230	10890	15750	15760	9520	15390	14720	11510	13770	12200	16740	12680	10220	12040	12500	11060
Instance 16	16020	13040	11820	17420	12180	19630	11210	17280	15160	10920	12060	17560	17200	11140	16230	15850	12190	14780	13790	17440	12810	11300	12420	13170	13170
Instance 17	160200	130400	118200	174200	121800	196300	112100	172800	151600	109200	120600	175600	172000	111400	162300	158500	121900	147800	137900	174400	128100	113000	124200	131700	131700
Instance 18	16020	14020	11820	17420	12700	19630	11900	17570	15350	10920	12060	17560	17200	11140	16230	15850	12420	15110	13790	17620	13900	11300	13160	13590	13170
Instance 19	5710	5080	5290	7910	5250	8860	4350	7800	6060	5100	4330	7230	7150	4560	6820	5880	5560	5930	5070	7200	4870	4590	5010	4920	4620
Instance 20	57100	50800	52900	79100	52500	88600	43500	78000	60600	51000	43300	72300	71500	45600	68200	58800	55600	59300	50700	72000	48700	45900	50100	49200	46200
Instance 21	5710	5680	5430	7910	5250	8860	4550	7800	6330	5100	4330	7230	7150	4560	6820	5880	5560	5930	5070	7200	5390	4750	5010	5420	5280
Instance 22	10720	8240	6780	12920	7140	14870	6870	12810	10530	6540	7390	12240	12520	6000	11830	10710	7450	10130	8750	12950	8190	6480	8070	8340	7800
Instance 23	107200	82400	67800	129200	71400	148700	68700	128100	105300	65400	73900	122400	125200	60000	118300	107100	74500	101300	87500	129500	81900	64800	80700	83400	78000
Instance 24	10720	8740	6780	12920	7140	14870	7070	12810	10620	6540	7390	12240	12520	6000	11830	10710	7450	10130	8750	12950	8710	6640	8070	8840	7800
Instance 25	11660	9630	8520	13060	8560	14290	7890	12950	11190	8050	8440	12580	12940	7710	12370	11510	8420	11080	10100	13170	9860	7650	9220	9610	9730
Instance 26	116600	96300	85200	130600	85600	142900	78900	129500	111900	80500	84400	125800	129400	77100	123700	115100	84200	110800	101000	131700	98600	76500	92200	96100	97300
Instance 27	11660	9890	8520	13060	8560	14290	8090	12950	11340	8050	8440	12580	12940	7710	12370	11510	8420	11080	10100	13170	9900	7690	9220	9870	9790
Instance 28	8540	8500	8310	11920	8380	12490	7550	11110	9760	7790	7830	10740	10390	8080	10380	9890	9360	9300	8520	10350	7940	8270	8150	8360	8120
Instance 29	85400	85000	83100	119200	83800	124900	75500	111100	97600	77900	78300	107400	103900	80800	103800	98900	93600	93000	85200	103500	79400	82700	81500	83600	81200
Instance 30	8540	9630	8520	11940	9510	12510	8330	11520	10010	7790	7830	10740	10390	8080	10380	9890	9620	9570	8520	10650	9360	8330	8980	9080	8540
Instance 31	13640	11560	9870	16930	10620	18930	10070	16120	14140	9230	10890	15750	15760	9520	15390	14720	11250	13500	12200	16440	11260	10160	11210	11780	11060
Instance 32	136400	115600	98700	169300	106200	189300	100700	161200	141400	92300	108900	157500	157600	95200	153900	147200	112500	135000	122000	164400	112600	101600	112100	117800	110600
Instance 33	13640	12690	9870	16950	11400	18950	10850	16530	14300	9230	10890	15750	15760	9520	15390	14720	11510	13770	12200	16740	12680	10220	12040	12500	11060
Instance 34	16020	13040	11820	17420	12180	19630	11210	17280	15160	10920	12060	17560	17200	11140	16230	15850	12190	14780	13790	17440	12810	11300	12420	13170	13170
Instance 35	160200	130400	118200	174200	121800	196300	112100	172800	151600	109200	120600	175600	172000	111400	162300	158500	121900	147800	137900	174400	128100	113000	124200	131700	131700
Instance 36	16020	14020	11820	17420	12700	19630	11900	17570	15350	10920	12060	17560	17200	11140	16230	15850	12420	15110	13790	17620	13900	11300	13160	13590	13170

Instance	SC76	SC77	SC78	SC79	SC80	SC81	SC82	SC83	SC84	SC85	SC86	SC87	SC88	SC89	SC90	SC91	SC92	SC93	SC94	SC95	SC96	SC97	SC98	SC99	SC100
Instance 1	5660	4950	5230	5780	5830	7440	4720	4910	5640	9150	7000	5830	4720	5250	4230	5470	4860	5330	5540	6410	6100	5400	5430	5870	7490
Instance 2	56600	49500	52300	57800	58300	74400	47200	49100	56400	91500	70000	58300	47200	52500	42300	54700	48600	53300	55400	64100	61000	54000	54300	58700	74900
Instance 3	5660	4950	5230	6160	5830	7440	4720	4970	5640	9150	7000	5830	4800	5560	4390	5470	4860	5330	5810	6770	6100	5400	5430	5870	7490
Instance 4	10670	8900	8650	11910	10570	13190	7600	7400	9220	15160	12750	9510	7780	6690	5670	9150	8280	9010	10550	11240	10300	8980	10440	10610	12500
Instance 5	106700	89000	86500	119100	105700	131900	76000	74000	92200	151600	127500	95100	77800	66900	56700	91500	82800	90100	105500	112400	103000	89800	104400	106100	125000
Instance 6	10670	8900	8650	11910	10570	13190	7600	7400	9220	15160	12750	9510	7860	6910	5830	9150	8280	9010	10640	11240	10300	8980	10440	10610	12500
Instance 7	11410	10360	10400	12530	11400	13410	9470	8250	10490	14780	12970	11010	8760	8200	7240	10500	10030	10360	11090	11900	11190	10450	11180	11440	12640
Instance 8	114100	103600	104000	125300	114000	134100	94700	82500	104900	147800	129700	110100	87600	82000	72400	105000	100300	103600	110900	119000	111900	104500	111800	114400	126400
Instance 9	11410	10360	10400	12530	11400	13410	9470	8250	10490	14780	12970	11060	8760	8450	7240	10500	10030	10360	11240	11900	11250	10450	11180	11440	12640
Instance 10	9390	8830	8540	8930	9190	11100	8210	8190	8880	12300	10540	9330	8370	8370	7660	9430	8730	8690	9070	9960	9540	8430	9080	8860	10820
Instance 11	93900	88300	85400	89300	91900	111000	82100	81900	88800	123000	105400	93300	83700	83700	76600	94300	87300	86900	90700	99600	95400	84300	90800	88600	108200
Instance 12	9390	8830	8540	9870	9370	11110	8210	8790	9320	12300	10860	9650	8610	8920	7720	9590	8750	8690	9500	10160	9540	8430	9220	8860	11260
Instance 13	14400	12690	11960	15730	13930	17200	11090	11030	12460	18740	16630	13010	11430	9810	9100	13110	12150	12370	13990	14630	13740	12010	14090	13600	15830
Instance 14	144000	126900	119600	157300	139300	172000	110900	110300	124600	187400	166300	130100	114300	98100	91000	131100	121500	123700	139900	146300	137400	120100	140900	136000	158300
Instance 15	14400	12690	11960	16050	14110	17200	11090	11220	12900	18740	16950	13330	11670	10270	9160	13270	12170	12370	14330	14630	13740	12010	14230	13600	16270
Instance 16	15770	14420	13980	18520	15000	18080	13050	11940	13850	19710	18290	14880	12570	11350	10690	14520	14020	13990	14860	16070	15050	13810	15870	14670	16870
Instance 17	157700	144200	139800	185200	150000	180800	130500	119400	138500	197100	182900	148800	125700	113500	106900	145200	140200	139900	148600	160700	150500	138100	158700	146700	168700
Instance 18	15770	14420	13980	18830	15210	18080	13050	12010	14050	19710	18400	15300	12570	11870	10690	14650	14070	13990	15260	16070	15050	13810	15870	14670	17010
Instance 19	5660	4950	5230	5780	5830	7440	4720	4910	5640	9150	7000	5830	4720	5250	4230	5470	4860	5330	5540	6410	6100	5400	5430	5870	7490
Instance 20	56600	49500	52300	57800	58300	74400	47200	49100	56400	91500	70000	58300	47200	52500	42300	54700	48600	53300	55400	64100	61000	54000	54300	58700	74900
Instance 21	5660	4950	5230	6160	5830	7440	4720	4970	5640	9150	7000	5830	4800	5560	4390	5470	4860	5330	5810	6770	6100	5400	5430	5870	7490
Instance 22	10670	8900	8650	11910	10570	13190	7600	7400	9220	15160	12750	9510	7780	6690	5670	9150	8280	9010	10550	11240	10300	8980	10440	10610	12500
Instance 23	106700	89000	86500	119100	105700	131900	76000	74000	92200	151600	127500	95100	77800	66900	56700	91500	82800	90100	105500	112400	103000	89800	104400	106100	125000
Instance 24	10670	8900	8650	11910	10570	13190	7600	7400	9220	15160	12750	9510	7860	6910	5830	9150	8280	9010	10640	11240	10300	8980	10440	10610	12500
Instance 25	11410	10360	10400	12530	11400	13410	9470	8250	10490	14780	12970	11010	8760	8200	7240	10500	10030	10360	11090	11900	11190	10450	11180	11440	12640
Instance 26	114100	103600	104000	125300	114000	134100	94700	82500	104900	147800	129700	110100	87600	82000	72400	105000	100300	103600	110900	119000	111900	104500	111800	114400	126400
Instance 27	11410	10360	10400	12530	11400	13410	9470	8250	10490	14780	12970	11060	8760	8450	7240	10500	10030	10360	11240	11900	11250	10450	11180	11440	12640
Instance 28	9390	8830	8540	8930	9190	11100	8210	8190	8880	12300	10540	9330	8370	8370	7660	9430	8730	8690	9070	9960	9540	8430	9080	8860	10820
Instance 29	93900	88300	85400	89300	91900	111000	82100	81900	88800	123000	105400	93300	83700	83700	76600	94300	87300	86900	90700	99600	95400	84300	90800	88600	108200
Instance 30	9390	8830	8540	9870	9370	11110	8210	8790	9320	12300	10860	9650	8610	8920	7720	9590	8750	8690	9500	10160	9540	8430	9220	8860	11260
Instance 31	14400	12690	11960	15730	13930	17200	11090	11030	12460	18740	16630	13010	11430	9810	9100	13110	12150	12370	13990	14630	13740	12010	14090	13600	15830
Instance 32	144000	126900	119600	157300	139300	172000	110900	110300	124600	187400	166300	130100	114300	98100	91000	131100	121500	123700	139900	146300	137400	120100	140900	136000	158300
Instance 33	14400	12690	11960	16050	14110	17200	11090	11220	12900	18740	16950	13330	11670	10270	9160	13270	12170	12370	14330	14630	13740	12010	14230	13600	16270
Instance 34	15770	14420	13980	18520	15000	18080	13050	11940	13850	19710	18290	14880	12570	11350	10690	14520	14020	13990	14860	16070	15050	13810	15870	14670	16870
Instance 35	157700	144200	139800	185200	150000	180800	130500	119400	138500	197100	182900	148800	125700	113500	106900	145200	140200	139900	148600	160700	150500	138100	158700	146700	168700
Instance 36	15770	14420	13980	18830	15210	18080	13050	12010	14050	19710	18400	15300	12570	11870	10690	14650	14070	13990	15260	16070	15050	13810	15870	14670	17010