2013

Optimal consignment stocking policies for a supply chain under different system constraints

Hui Zhi Yi

Louisiana State University and Agricultural and Mechanical College

OPTIMAL CONSIGNMENT STOCKING POLICIES FOR A SUPPLY CHAIN UNDER DIFFERENT SYSTEM CONSTRAINTS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Interdepartmental Program
in Engineering Science

by
Hui Zhi Yi
B.S., Central South University of Technology, 1994
M.S.I.E., Louisiana State University, 2012
December 2013

ACKNOWLEDGEMENTS

In many ways, the best part of my years at LSU has been the amazing people that I have had the opportunity to meet and collaborate with personally. This dissertation would never have happened without their enthusiasm and support.

First and foremost, I would like to express my deepest thanks to my advisor and committee chair, Professor Bhaba R. Sarker. It has been his indispensable advice, experience, technical skill, and mentoring that has made this dissertation ever possible. I am grateful to Dr. Sarker for being a consistent source of support, supervision and encouragement, and for showing me the principle of doing good research which will benefit my future career forever.

My special thanks are to Professor Lawrence Mann, Jr. of MIE Department, Professor Guoli Ding of Mathematics Department, Professor Xuebin Liang of ECE Department, and Professor Kenneth J. Reichelt of Accounting Department (Graduate School Representative) for serving on my committee. They have provided valuable feedback and interesting perspectives which certainly helped improve the quality of this research. My sincere thanks are also to Dr. Junfang Yu of University of Tennessee at Knoxville, TN without him I would not have come here. I especially thank Professor T. Warren Liao—what I learnt from his valuable courses induced an important portion of this research. My special thanks are also to Professor Fereydoun Aghazadeh for his constant encouragement and Professor Craig M. Harvey for his support, to complete this work.

My gratitude is also extended to Professor Cunrong Li of Wuhan University of Technology (currently CSC Visiting Scholar at LSU), Hongye Wang, Bingqing Wu, Qinglin Duan, Lu Jin, Jian He, Xiaoyue Jiang, Jun Zou and all other people who have helped me in various ways complete this work.

Finally, I would like to thank my dearest wife, Weiwen Liao, for her amazing encouragement and support with her unconditional love and sacrifice in achieving my goals. Last, but definitely not the least, I would like to address my thanks to my father Dean Yi, my mother Xihua He, my sisters Xincan He and Xinkui He for their continuous love, sacrifice, support, and encouragement throughout my life. My son, Jiaqi, is still trying to understand what his father is trying to do, not giving him enough time to play!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The research aims are to enable the decision maker of an integrated vendor-buyer system under Consignment Stock (CS) policy to make the optimal/sub-optimal production/replenishment decisions when some general and realistic critical factors are considered. In the system, the vendor produces one product at a finite rate and ships the outputs by a number of equal-sized lots within a production cycle. Under a long-term CS agreement, the vendor maintains a certain inventory level at the buyer's warehouse, and the buyer compensates the vendor only for the consumed products. The holding cost consists of a storage component and a financial component. Moreover, both of the cases that the unit holding costs may be higher at the buyer or at the vendor are considered. Based upon such a system, four sets of inventory models are developed each of which considers one more factor than the former. The first set of models allows a controllable lead-time with an additional investment and jointly determines the shipping size, the number of shipments, and the lead time, that minimize the yearly *joint total expected cost* (*JTEC*) of the system. The second set of models considers a buyer's capacity limitation which causes some shipments to be delayed so that the arrival of these shipments does not cause the buyer's inventory to go beyond its limitation. As a result, the number of delayed shipments is added as the fourth decision variable. A variable demand rate is allowed in the third set of models. Uncertainty caused by the varying demand are controlled by a safety factor, which becomes the fifth decision variable. Finally, the risk of obsolescence of the product is considered in the fourth model. The first model is solved analytically, whereas the rest are not, mainly

because of the complexity of the problem and the number of variables being considered. Three doubly-hybrid meta-heuristic algorithms that combine two different hybrid meta-heuristic algorithms are developed to provide a solution procedure for the rest of models. Numerical experiments illustrate the solution procedures and reveal the effects of the buyer's capacity limitation, the effects of the variable demand rate, and the effects of the risk of obsolescence, on the system. Furthermore, sensitivity analysis shows that some of the system parameters (such as the backorder penalty, the extra space penalty, the ratio of the unit holding cost of the vendor over that of the buyer) are very influential to the joint system total cost and the optimal solutions of the decision variables.

Keywords: Supply chain; inventory; consignment; integrated production; lead time; space limitation; variable demand; obsolescence

CHAPTER I
INTRODUCTION

This year marks the century of the first economic order quantity (EOQ) model (Harris, 1913), which is regarded as the beginning of modeling of an inventory system. During this period, countless developments have been made in the area of industrial engineering and operations research. This research follows the footprints of former researchers by studying the models of supply chain systems, and especially focusing on the following two facets: (1) how some important realistic factors, such as controllable lead time, buyer's space limitation, variable demand rate, and obsolescence, impact a vendor-buyer integrated inventory system under consignment stocking policy, and (2) whether there is any solution methodology that may be used to solve a variety of complicated inventory models with multiple variables.

1.1 BACKGROUND

To avoid confusions, it is necessary to first explain the terminologies of supply chain (SC) and consignment stock (CS). Generally speaking, SC activities transform resources, raw materials and/or components into finished product so as to satisfy the demand of the end user, and supply chain management (SCM) deals with planning, designing, executing, and controlling of people, resources, information, and activities pertaining to the transformation and movement of raw materials to customers. Simply put, the objective of SCM is to get the right resources to the right places at the right time, under different constraints (such as budget, time, space, technological limitations, etc.), so as to satisfy the

customers requirement and optimize the performance measurement (such as cost, throughput rate, etc.). Consignment stocking policy, among numerous other policies, is an innovative approach of inventory management for efficiently operating a supply chain system. Under such a policy, the vendor places and maintains a certain amount of goods at a buyer location but does not receive the payment from the buyer immediately. The buyer only pays and owns the products when it is withdrawn from his/her stock. Shortly put, a consignment stock (CS) is inventory that is not paid for until sold or used. The customer may return the unsold or unused items to the vendor at any point of time (see Sarker 2013 for details).

1.1.1 Research trends in consignment stocking

During the last two decades, the development of supply chain and inventory management has been moving from the study of integrated system (IS) to vendor managed inventory (VMI) systems and to CS policies. Nowadays, many companies have realized the importance of collaboration among vendors and buyers to create a successful supply chain system. An early form of such a collaboration system, which was widely studied in the 1980s to 1990s, is the vendor-buyer integrated inventory system. Under such a system, a central planner uses the demand and inventory information of both the vendor and buyer to make the production and replenishment decisions so that the entire system cost is minimized or the entire system profit is maximized. The system enjoys a better overall decision, but both trading parties take care of their own inventory. Considerable amount of

research has revealed that optimal decisions based on an integrated system work better than decisions based on any individual company.

A recent trend of development in collaboration among businesses is the VMI wherein the vendor places and maintains a reasonable amount of goods at the buyer location. As compensation, the vendor gains full and real-time access to the demand and inventory information of the buyer. Using this information, the vendor manages the buyer's inventory and makes replenishment decisions. Under VMI, even though the vendor maintains the buyer's inventory level, the expenses related to the buyer's inventory are still bared by the buyer.

The CS policy, also known as Vendor/Supplier Owned Inventory (VOI/SOI), furthers the VMI practice in supply chain management by allowing a delay in the payment transaction from the buyer to the vendor. There usually are two parties, the consignor and the consignee, in a CS agreement. The consignor, usually the manufacturer or the distributor, provides the products to be sold or used, and the consignee, typically the retailer or buyer, receives and sell/use the products. While the consignee maintains the physical possession of the inventory, the consignor retains the ownership of the goods. Upon sold or used, the consignee compensates the consignor a portion of the profits. The CS policy is becoming more and more common in the cloth, furniture, book, antique, jewelry, and second-hand goods industries. Popular sites to see CS products being stored include 7-11, Circle-K, gas-stations and many neighborhood corner-shops.

Compared to VMI, the CS policy is more beneficial to the buyer for an increased cash flow since the payment from the buyer to the vendor is delayed for a certain period. Other common benefits of the CS policy to the buyer include lower inventory cost, reduced ordering cost, reduced procurement lead time, reduced stock-out risks, and increased service level. The benefits of CS policy for the vendor include increased production lot size flexibility and warehouse space, improved demand visibility, and long term relationship.

On the downside of the CS policy, the consignor will carry a larger burden in its cash flow, so the vendor may run into problems when the payback is too slow to cover the next productions. Moreover, the vendor have to stand the risk of a high returning products from the buyer after a long time, which brings a higher uncertainty to the consignor's profitability. As a result, the consignee typically agrees a higher percentage of profit sharing to the consignor than other policies.

1.1.2 Consignment stock controlling factors

An integrated inventory model under CS policy cannot be considered without constraints. In reality, a constraint often influences the performance of such a system. Such factors need to be carefully addressed.

(1) Controllable lead time

First of all, lead time is one of the most important factors that need to be considered. In this study, lead time refers to the time required to ship the products from the vendor to the buyer. In most cases, there are multiple choices of transportations which may take different

lead times and at different shipping costs. On one hand, a longer lead time may itself be more inexpensive. On the other hand, however, it will delay the time that the products can be sold or used, thus reduce the turnover rate of the capital invested to the products. Moreover, because the demand may be varying, a longer lead time forces the buyer to stock more inventory against the risk of lost sale, which increases the holding cost of the system. As a result, the decision maker often has to balance between spending more in holding more inventories at hand and spending more to reduce the lead time.

(2) Buyer's space limitation

Secondly, the buyer always has finite space limitation. Holding more of one product in the warehouse may mean holding less of the others. Therefore, the buyer may want to offer the vendor a space limitation. How the limitation affects the performance is then crucial to the success of the CS policy.

(3) Variable demand rate

Thirdly, demand rate is always an uncertainty. Because of the uncertainty, the buyer may be faced with backorders, whereas, extra space may be needed for unsold products, both of which incur added costs to the supply chain. Hence, the performance of the CS policy is influenced by this uncertainty. As a result, how the CS inventory model works under a varying demand deserves study.

(4) Risk of obsolescence

Lastly, the demand for most of the products may not exist forever, that is, the product may be obsolescent eventually and the residue value of the out dated products is often trivial. Holding too much inventory may incur a higher obsolescent cost, whereas holding too little will decrease the profits of a company by unsatisfied demand. Therefore, how should the company modify the replenishment policy against the risk of obsolescence is of value.

1.1.3 The needs of using meta-heuristic methods

As for the models to be developed in this research, ideally we hope they can be solved mathematically, then the solutions found are guaranteed to be globally optimal. Traditional ways of solving inventory models depend on the convexity and differentiation of the objective function. However, if the objective function is too complicated or contains many decision variables, it becomes difficult to prove the convexity of the objective function. Moreover, the objective function sometimes may be not differentiable. Therefore, the inventory model developed may not always be solvable by traditional mathematical ways. We, thus, need to find a solution methodology that does not require the mathematical information of the objective function. Meta-heuristic algorithms, which have been shown to work in numerous engineering problems, may be a good choice. In this research, we will develop some meta-heuristic algorithms that may be used to solve the complicated, non-differentiable inventory models.

1.2 RESEARCH GOAL AND OBJECTIVES

The goal of this research is to enable the decision maker of an integrated system, under CS policy, to make the optimal/sub-optimal production/replenishment decisions when some more general and more realistic factors are considered. According to the literature and the limitation of current research, such factors could be, but are not limited to, the controllable lead time, the buyer capacity limitation, the uncertain demand, and the obsolescence.

Another goal of this research is to jointly determine the number of shipments, the size of each shipment, the number of delayed shipments, the safety level, and the lead time, that minimize the yearly *joint total expect cost* (*JTEC*) of the production system in which the lead-time is controllable with an extra investment, the buyer has a space limitation to the vendor, the products have a finite life cycle (obsolescence), and the demand is variable. Moreover, in order to provide a solution method that is capable of coping with more complicated models for concurrent and future researchers; this research develops three doubly-hybrid meta-heuristic algorithms. Each of the doubly-hybrid algorithms is a hybrid of two hybrid meta-heuristic algorithms, which, in turn, are a hybrid of a population based meta-heuristic algorithm and a local search algorithm.

In order to reach the goals of this research, four generalized inventory models have to be built to quantify the effects of the four critical factors. Moreover, a robust solution procedure, that can be easily modified to solve these four different models, has to be developed. Specifically, the detailed objectives of this research are:

(1) *One-factor CS model*: To determine the optimal number of shipments, the size of each shipment, and the lead time, that minimize the annual *JTEC* of the integrated production system under CS policy in which the lead-time is controllable with an extra investment.

(2) *Two-factor CS model*: To determine the optimal number of shipments, the size of each shipment, the number of delayed shipments, and the lead time, that minimize the annual *JTEC* of the integrated production system under CS policy in which the lead-time is controllable with an extra investment and the buyer has a space limitation in its warehouse.

(3) *Three-factor CS model*: To determine the optimal number of shipments, the size of each shipment, the number of delayed shipments, the safety factor, and the lead time, that minimize the annual *JTEC* of the integrated production system under CS policy in which the lead-time is controllable with an extra investment, the buyer has a space limitation in its warehouse, and the demand is variable.

(4) *Four-factor CS model*: To determine the optimal number of shipments, the size of each shipment, the number of delayed shipments, the safety factor, and the lead time, that minimize the annual *JTEC* of the integrated production system under CS policy in which the lead-time is controllable with an extra investment, the buyer has a space limitation in its warehouse, the product has a finite life cycle, and the demand is variable.

(5) *Doubly-hybrid meta-heuristic methods*: To improve the performance of basic meta-heuristic methods, such as Particle Swarm, Harmony Search, Differential Evolution, and Hooke and Jeeves local search so that the optimal solutions of the four sets of CS models can be efficiently located. In this effort, multiple heuristics will be developed to evaluate the quality of the heuristic solutions.

Since all these factors have not been taken into consideration in past CS models, this research divides the final model into four sub-models, starting with the basic CS model and adding one impact factor at a time.

1.3 SCOPE

This research deals with the optimal producing/replenish/ delivery policy problem in a vendor buyer integrated system under a CS policy and with some system constraints. The solution will provide the decision-maker with an optimal plan for not only the traditional variables such as the economical production/ordering quantity, and the number of shipments within each production cycle, but also some other critical factors that are commonly faced. Example of those important factors include: investments on “crushing” the lead time, placing a space limitation in the buyer’s warehouse, offering a discount when the demand is stochastic and shortage is encountered, and allowing the demand to disappear at some sudden point. Because of the searching nature of the solution methodology, it is believed that it can also be applied to solve other inventory models that consider other important factors.

Since the doubly-hybrid meta-heuristic algorithms developed here do not rely on the differential information of the objective function, they are believed to be able to solve other constrained or unconstrained engineering problems as long as they have an objective function, examples of which can be seen from Yi, *et al.* (2013) who showed the application of the doubly-hybrid algorithm to 18 different engineering problems. The maximum number of variables solved in the 18 engineering problems is 10, but it is believed that it can cope with problems with more variables.

1.4 Industrial Applications of the CS Policy

Because the CS policy benefits both the vendor and the buyer, it has been increasingly followed as industrial practice. Examples include second-hand goods industry (Bolen 1988), vending machine services and soft drink industry (Ong, *et al.*, 1996), parts supply in assembly systems of automobile or personal computer industry (Gerchak and Wang 2004, Gumus, *et al.*, 2008), aircraft industries (Micheau 2005), supermarkets and convenience stores (Coughlan, *et al.*, 2001), retailing (Turcsik 2002), on-line commerce and seafood (Chen, *et al.* 2010), and hospital operation (Lee and Wang 2008), etc.

1.5 Research Significance

Consignment stocking policy is a special form of cooperation, and it is becoming more popular and dominant, especially in some industries, such as auto parts, e-commerce (e-Bay, Amazon, etc.), medicine, and fashion. This research has a significant impact on the decision makers of many industries that adopt the CS policy in supply chain management.

Compared to many applications of the CS policy, the study of a supply chain system under the CS policy is rare. The effects of many key factors that may be influential to the system remain unclear under the CS policy. Moreover, literature review reveals that current research about the supply chain system under CS policy contain various flaws.

This research developed four general CS models considering the impacts of controllable lead time, buyer space limitation, variable demand rate, and obsolescence, which are practical problems that most industries now face. The variable lead time problems tags cost parameters based on the transport mode and customer satisfaction. So, an optimal lead time leads to an economic policy. A typical warehouse or storage facility of a buyer (retailer or any party of the supply chain) has usually limited space which is costly. So, managing this space is important. This research also involves stochastic demand, and limited life cycle (obsolescence) of products—these system characteristics define a realistic problem.

Next, the research develops new meta-heuristic search procedures that can be used for solving a class of problems, defined by a complicated adjustable objective function with many variables. The doubly-hybrid meta-heuristic methods presented in this research do not require the traditional derivative approach to solve a convex objective function—it uses the search procedure that is novel to many other existing techniques that faces difficulty in finding the optimal solutions.

The models and the solution procedures developed here are expected to have significant impact on the economic benefits and practical utility for different industries.

CHAPTER II
LITERATURE REVIEW AND PROBLEM JUSTIFICATION

This Chapter summarizes current literature in the field of both the inventory modeling under the CS policy and the state of art in meta-heuristic methods. A few shortcomings of previous studies are then discussed. Finally, the background studies and the development of this research are summarized.

2.1 LITERATURE ON INVENTORY MANAGEMENT

In a vendor-buyer integrated supply chain system, the economic order quantity (EOQ) of the buyer is often not acceptable to the vendor, or the buyer will not accept the economic manufacturing quantity (EMQ) policy which is preferable by the vendor (Lu, 1995). Therefore, a certain compromise by either or both trading partners is necessary for the benefit of the integrated system. As a result, recent research has paid attention to the cooperation and/or coordination between the buyer and the vendor, and has proved that the collaboration between the buyer and the vendor gives a greater benefit than a non-collaborative relationship can do.

2.1.1 Integrated systems

Over the years, general mathematical models have been developed to describe the behavior in such integrated systems and to determine optimal control policies. Among early researchers, Goyal (1976; 1977) developed an infinite production rate, lot-for-lot *Joint Economic Lot Size* (*JELS*) model and addressed the most competitive approach for this system to minimize the joint total cost of both the buyer and the vendor, rather than to treat

the buyer and the vendor independently. Banerjee (1986) extended Goyal's models (1976; 1977) by adopting a finite production rate. The lot-for-lot assumption was further relaxed by Goyal (1988) by splitting the single shipment into multiple shipments and assuming the vendor's lot size to be an integer multiple of the buyer's order size. The joint total cost in Goyal's (1988) model is shown to be lower than that of Banerjee's (1986).

Golhar and Sarker (1992) considered an integrated vendor-manufacturer situation where the demand of the customers was related to the requirement of raw material supplies from the vendor through a finished-goods-to-raw-material conversion factor while the carrying cost of raw material was proportion to the production period only. They proposed a general solution for *imperfect matching* situation *lot-for-1-order* policy in which the production can stop at any time, that is, either at the time of delivery or during the delivery interval time of the finished products. They are also the first to consider the case of *imperfect matching* where the production time may not be an integer multiple of the manufacturer's delivery interval time. Sarker and Parija (1994) improved the study with a more efficient method to find the optimal solution. The *lot-for-1-order* of raw material policy was further generalized by Sarker and Parija (1996) with a manufacturer's *n-order* policy. In this model, the vendor satisfies the manufacturer's demand by a *lot-for-n-order* policy. Sarker and Coates (1997) studied the reduction of setup cost under variable lead time and finite opportunities for investment. Parija and Sarker (1999) further developed a closed-form solution to determine the optimal ordering policy for raw materials and the

optimal production batch size at a manufacturer which supplies the finished products to multiple customers, with different shipment sizes and time-intervals for each of the customers. More recently, Sarker and Diponegoro (2009) extended the research to a system with multiple suppliers and multiple buyers. Sarker, et al. (2009) further studied the scenario where recovery and procurement are under multiple setups.

Lu (1995), extending Goyal's (1988) model, presented an approach to finding the optimal production and shipment policy assuming an equal shipment size. Goyal (1995) addressed a better shipment policy in which the shipment sizes increase by the ratio of the production rate to the demand rate. Goyal's (1995) results showed that his shipping policy might, in some cases, lower the total cost. A globally optimal solution for lot sizing and shipping policy was presented by Hill (1999), who suggested a number of shipments that increase by the ratio of the production rate to the demand rate followed by a number of equal-sized shipments. Hill also showed that both the equal shipment policy and the increasing shipment policy in previous models on this issue were particular cases of his new model.

Recently, variations of general models mentioned above have been widely discussed with reference to imperfect quality (Huang, 2002; Goyal, *et al*., 2003; Chakraborty and Giri, 2012), allowed shortage (Wu and Ouyang, 2003), linearly varying lead time with lot size (Ben-Daya and Hariga, 2004), milk-run supplies of materials to multiple buyers (Chen and Sarker, 2010), emergency orders (Giri and Dohi, 2009), manufacturing setup cost reduction

(Huang, *et al*., 2011), continuous price decrease (Yu and Sarker, 2011), and production uncertainty (Giri, 2011).

2.1.2 Consignment stocking policies

Among all other models, inventory management under CS case has gotten special attention. Corbett (2001) is probably the first to have studied the benefits of the CS policy to bring an integrated single-vendor single-buyer inventory system. He showed that the CS policy can help reduce the cycle stock by providing an additional incentive to the vendor to reduce the batch size and by giving the buyer an incentive to increase the safety stock. Braglia and Zavanella (2003) developed a two-variable analytical CS model for a single-vendor and single-buyer system that considers equal shipment with or without delayed deliveries. They only studied the situation where the unit stockholding costs increase as stock moves down the supply chain. He also provided a comparison between a deterministic integrated system under CS policy and Hill's (1997; 1999) model under the traditional agreement. The numerical results of their developed two-variable CS model showed that CS policy might be more preferable than traditional agreement under some uncertain environments. Valentini and Zavanella (2003) provided performance analysis of the CS policy for a single-vendor single-buyer system. Gerchak and Khmelnitsky (2003) studied an interesting case, in which suppliers cannot verify retailer's sales reports. Piplani and Viswanathan (2003), using another term: Supplier-owned inventory (SOI), studied a similar situation of the CS policy (also see Yap, 1999). They stated that one major benefit of SOI

strategy is that the buyers can adopt the just-in-time (JIT) procurement without incurring high replenishment costs. They also showed, with numerical examples, that although the benefit to the supplier from SOI is dependent on the problem parameters, the SOI arrangement is always beneficial for the entire integrated system. Recently, Sarker (2013) made a critical review and compared different perspectives of about 60 consignment stock policy models for supply chain systems for over 100 recent technical articles. Other researchers extended the study of the CS policy problem to many new dimensions and their results are beneficial to the field of supply chain systems (Srinivas and Rao, 2004; Zanoni, *et al.*, 2005; Sarker, *et al.*, 2011; and Yu, *et al.*, 2012).

(a) Obsolescence

It is important to study the impact of outdating on the integrated system under the CS policy because products may rapidly become outdated and the demand of the products may disappear in such industries as electronics and fashion. Persona, *et al.* (2005) are the first to extend Braglia and Zavanella's (2003) deterministic model to allow for obsolescence. Battini, *et al.* (2010a) then relaxed the deterministic demand assumption and claimed that they also took space limitation into consideration. Battini, *et al.* (2010b) further extended Battini, *et al.* (2010a) model to the single-vendor multi-buyer cases.

(b) Revenue sharing

CS policy, with revenue sharing, is a specific form of cooperation between buyers and vendors where the vendor takes more control and decides the retail prices. It has been

widely applied in many industries, especially in online marketplaces, such as Amazon.com, Ebay.com, and Alibaba.com, etc. Wang, *et al.* (2004), for the first time, studied the channel performance of a CS system with revenue sharing and concluded that the channel performance and the performance of each individual firm depends on the demand price elasticity and on the retailer's share of channel cost. The authors also indicated that a decentralized supply chain cannot be perfectly coordinated. Gerchak and Wang (2004) conducted a similar study to the case of Vendor Managed Inventory (VMI) with revenue sharing. Li and Hua (2008) improved Wang, *et al.*'s (2004) research by providing a cooperative game model that enables perfect coordination between vendor and buyer in a decentralized system. Li, *et al.* (2009) presented a more comprehensive study on how the parameters of a CS model, with revenue sharing, may affect the decision making. Recently, Zhao and Wu (2011) allowed stochastic demand and stochastic output in CS modeling.

(c) Multiple vendors/buyers

Zavanella and Zanoni (2009) extend Braglia and Zavanella's (2003) single-vendor single-buyer model to the case of single-vendor and multi-buyer. Srinivas and Rao (2010) tried to use the Genetic Algorithm (GA) to solve and optimize the solution of a single-vendor multi-buyer CS model. Battini, *et al.* (2010a) considered several important factors including single-vendor multi-buyer, stochastic demand, allowing obsolescence, and buyer space limitation.

(d) Splitting unit holding cost

Also, based on Braglia and Zavanella's (2003) model, Hill and Omar (2006) considered both the cases that the buyer's unit holding cost might be greater or less than the vendor's. Huang and Chen (2009) provided a study by further dividing the unit holding cost into a financial component and a storage component. The first stands for the opportunity costs of immobilizing capital in the products, while the second portion is the cost that has to do with storage, movement, management, insurance, etc. When the goods are in the buyer's warehouse, both two portions of holding cost will be at the expense of the buyer with a traditional agreement since the buyer purchased the goods before placing them into inventory. On the contrary, under the CS policy, the buyer only bear the stock component of the holding cost and the supplier still owns the products and thus must sustain the financial component. Table 2.1 shows the difference between CS model and traditional inventory model.

Table 2.1 Relevant inventory costs under CS policy and under traditional agreements

Relevant Costs \ Position of Goods	Vender		Buyer	
	Traditional agreement	CS policy	Traditional agreement	CS policy
Vender	F[a], S[b]	F, S	0	F
Buyer	0	0	F, S	S

a: F is the pure financial portion of the holding costs.
b: S is the pure storage portion of the holding costs.

In a different context, Sharma (2008) examined the situation where the production rate may be flexible according to the increase of decrease of the demand rate. Sharma (2009)

also addressed some other important issues such as the procurement of many input materials and allowable fractional backorders.

(e) Uncertain demand

There is less literature related to consignment inventory under uncertainty of demand. However, there are some models that deal with variability in demand for such situations. Corbett (2001) proposed a stochastic inventory model for a supply chain system with asymmetric information such as cycle stocks, safety stocks and consignment stock. Wang (2006) developed a joint pricing production decision model for supply chain system of complementary products with uncertain demand. Zhao and Wu (2011) developed a model for Agri-food supply chain coordination with revenue-sharing under stochastic output and stochastic demand. Yu, *et al.* (2012) developed another model for consignment inventory with generalized demand distribution where any distribution can be used to estimate the profit margin and the order quantity for better economic planning.

2.2 META-HEURISTIC METHODS

Since the term “meta-heuristic” was first introduced by Glover (1986), meta-heuristic has become a major branch of optimization methodologies and a critical component in soft computing studies and applications. Over more than two decades, two major trends in meta-heuristic have been: (i) the development of new popular solo meta-heuristic algorithms such as genetic algorithm (GA) (Holland, 1992, Aytug and Vergara, 2003, Chaudhury and Luo, 2005, Daniel and rajendran, 2005), particle swarm optimization (PSO) (Clerc and Kennedy,

2002), differential evolution (DE) (Storn and Price, 1997, Brest, *et al*., 2006, Qin, *et al*., 2009, Zhang and Sanderson, 2009, Das and Suganthan, 2011), ant colony optimization (ACO) (Dorigo, *et al*., 1999), Harmony Search (HS) (Geem, *et al*., 2001), artificial bee colony (ABC) (Karaboga and Basturk, 2008) based on new and innovative ideas, and (ii) the development of hybrid meta-heuristic algorithms in an attempt to take advantage of the good features from more than one meta-heuristic, even from traditional optimization methods other than meta-heuristics.

Intensification and diversification are two major issues when designing a meta-heuristic. Diversification refers to the ability to visit many and different regions of the search space, whereas intensification refers to the ability to dig deep into each local optima to obtain high quality solutions. Although most solo meta-heuristic algorithms attempt to achieve this objective according to its paradigms and philosophies, it turns out that some of them show certain specialization in intensification and others, in diversification. This explains the increasing need for hybrid meta-heuristics (Glover and Kochenberger, 2003). Hybrid meta-heuristics are algorithms that do not purely follow the concept of one single traditional meta-heuristic, rather they combine various algorithmic ideas, sometimes from outside of the meta-heuristic field. The hybridizations of different algorithmic concepts is usually motivated by the desire to obtain better performing systems that exploit and unite advantages of the individual pure strategies, i.e. such hybrids are believed to benefit from synergy.

Numerous hybrid meta-heuristics have been developed and applied to both combinatorial and continuous optimization problems. Talbi (2002) developed taxonomy of hybrid meta-heuristics in an attempt to provide a common terminology and classification mechanisms. Later, Raidl (2006) groups hybrids of meta-heuristics according to several criteria, which include algorithms used, level of hybridization, order of execution, and control strategy. Memetic algorithms are hybrid meta-heuristics with control strategy to be of the integrative type. Note that in integrative approaches, one algorithm is considered a subordinate, embedded component of another algorithm. In the case of memetic algorithms, the subordinate is a local search method and the main algorithm is genetic algorithm. On the other hand, the control strategy of cooperative meta-heuristics is collaborative. Parallel meta-heuristics are hybrid meta-heuristics with concurrent order of execution. This research will only review integrative and cooperative types because of their relevancy to the subject study.

2.2.1 Integrative hybrid meta-heuristics

In integrative hybrid meta-heuristics, one algorithm is embedded into another one. The followings are some examples of integrative meta-heuristics developed for continuous optimization applications. Most of them are for unconstrained continuous optimization. Al-Sultan and Al-Fawzan (1997) presented a hybrid algorithm, in which search directions are generated by tabu search (TS) and they are then used in the Hooke and Jeeves optimization algorithm. Digalakis and Margaritis (2004) evaluated the performance of three memetic algorithms with different local search techniques. Sun, *et al*. (2005) incorporates the

estimation of distribution algorithm into the DE algorithm in order to create solutions that are more promising. Pan, *et al*. (2006) enhanced the PSO with the optimal computing budget allocation (OCBA) technique and hypothesis testing. Liu, *et al*. (2007) proposed a new algorithm for multi-objective optimization, which combines PSO with a local search heuristic. Wang, *et al*. (2007) incorporated clonal selection principles into ACO, in which the cloning and mutation operations are embedded in the ant colony to enhance its search capability. Dimopoulos (2007) incorporates genetic operators into the PSO algorithm. Yin, *et al*. (2010) proposed cyber swarm algorithms, which improve particle swarm optimization using adaptive memory strategies. The set of the interacting solutions for each particle is augmented to become the reference set of scatter search (SS) and path relinking (PR). Duarte, *et al*. (2011) shows that their adaptive memory framework which coupling SS and TS with a post-processing modified simplex method is competitive in comparison with the state-of-the-art methods in terms of the average gap from the optima. LaTorre, *et al*. (2011) explores the use of a hybrid memetic algorithm based on a multiple offspring framework. In their framework, a DE algorithm and the first one of the local searchers of the multiple trajectory search algorithm (Tseng and Chen, 2008) are combined to produce competitive results.

2.2.2 Cooperative hybrid meta-heuristics

In a cooperative hybrid meta-heuristic method, different algorithms are used to search the solution space separately, but the results are shared to all other algorithms. Just like the integrative hybrid meta-heuristics, the cooperative strategies for hybrid meta-heuristics are

also frequently used to better explore the search space. Several studies (Crainic, *et al*., 2004; Pelta, *et al*., 2006; Cadenas, *et al*., 2009), have shown that multi-thread techniques are expected to produce better solutions than their sequential components, even when the available execution time for each thread is shorten. The combined use of different threads, each using a different searching strategy, increases robustness of the global search with respect to changes in the problem instances. However, designing a good cooperative strategy is important. As shown in Crainic, *et al*. (2004), an unrestricted information exchange may cause some problems such as premature convergence. Therefore, it is of research interest to find a way to control this information exchange intelligently.

The application of cooperative hybrid meta-heuristic algorithms to continuous optimization problems is briefly described in the following. Lin, *et al*. (2001) proposed a co-evolutionary hybrid DE algorithm to solve mixed integer nonlinear programming problems. It consists of an integer-valued variable evolution on the outer loop and a real-valued variable co-evolution on the inner loop. Bergh and Engelbrecht (2004) presented the cooperative particle swarm optimizer (CPS) for unconstrained continuous optimization, employing cooperative behavior to significantly improve the performance of the original algorithm. Huang, *et al*. (2007) proposed a DE approach based on a co-evolution mechanism to solve the constrained continuous optimization problems. He and Wang (2007) proposed a co-evolutionary PSO for constrained optimization, in which one population evolves the solution and the other evolves the penalty factor. Kao and Zahara (2008) proposed a cooperative

GA+PSO hybrid, in which the better half of the population are used to generate offspring by GA and the worse half are evolved by PSO. Liao (2010) proposed a cooperative hybrid based on differential evolution and harmony search and showed that it outperformed pure DE for constrained continuous optimization. Lung and Dumitrescu (2010) presented an algorithm called Evolutionary Swarm Cooperative Algorithm (ESCA), which uses three populations: two EA populations and one PSO population, and evaluated its performance on unconstrained optimization problems. Cadenas, *et al*. (2011) use a system composed by three different meta-heuristics, a GA, a TS and a simulated annealing (SA). They are executed in a parallel way while they cooperate under the supervision of a coordinator. This coordinator is able to control the cooperation using a collection of Support Vector Machine models and a fuzzy decision framework. They have applied their cooperative strategy to both combinatorial and continuous optimization problems.

2.3 SHORTCOMINGS OF PREVIOUS RESEARCH

Since the study of the CS policy within the framework of inventory control is relatively new (the original study of the inventory model with CS policy was Corbett, 2001), the effects of some critical factors are remain unclear. A number of other critical factors are yet to be considered into the system. Some of the shortcomings of previous research are summarized as the following.

2.3.1 Need for considering controllable lead time

Although in Braglia and Zavanella's (2003) work, the authors showed that the CS policy might be a strategic and profitable approach where demand or delivery lead times vary over time, they did not discuss how this policy works with a controllable lead time. They also discussed a stochastic demand case with zero delivery lead time, which is similar to the traditional inventory model in that the lead time is hypothesized as known (Kim and Park, 1985; Ravichandran, 1995) or with a certain probability distribution (Foote, *et al.*, 1988). Pan and Yang (2002) pointed out that in many practical cases, lead time can be reduced with an additional "crushing" cost, meaning that it is controllable. By adding this controllable lead time into decision variable, they extended Goyal's (1988) model and provided a procedure to find the optimal order quantity, lead time and delivering number when the probability distribution of the lead time demand is a normal one. They also showed that, for the given data, their model yields a lower joint total yearly cost than that of Goyal's (1988). However, Pan and Yang (2002) did not discuss the effects of the investment in reducing the lead time on the joint total yearly cost, nor did they consider more general cases such as what would be the role of controllable lead time in a system where the buyer's holding cost is lower than that of the vendor, or in a system of CS case. It is therefore interesting to study a CS system with investment on reducing lead time as an option.

2.3.2 Consideration of buyer's space limitation

Braglia and Zavanella (2003) presented a general deterministic single-vendor single-buyer CS-k model. In the CS-k model, the buyer offers a space limitation to the vendor so the vendor cannot place all his/her inventory at the buyer's warehouse. It follows that the last *k* numbers of shipments of each production cycle have to be delayed for a certain period. They showed that the CS-k model reduces to Hill's (1999) model when the value of *k* takes its maximum possible value and the general CS-k model reduces to a basic CS model when the value of *k* is zero. Braglia and Zavanella (2003) did not analytically solve any of these three models. Instead, they provided a numerical way of solving the two reduced models. Zanoni and Grubbstrom (2004) observed that there is a favorable property lying in Braglia and Zavanella's (2003) two reduced models. Using this property, Zanoni and Grubbstrom (2004) analytically solved the two reduced models over two decision variables: the shipping size, *q*, and the number of shipments within one cycle, *n*. However, the generalized CS-k model was remain unsolved.

In Lee and Wang's (2008) paper, the author provided a constrained deterministic single-vendor single-buyer CS model based on Braglia and Zavanella's (2003) CS-k model. However, Lee and Wang took a point of view of the manufacturer/vendor instead of the integrated system. Moreover, the authors used commercial software EXCEL to determine the solution of the model and did not provide a specific solution procedure or algorithm for the CS-k model.

Battini, *et al.* (2010a; 2010b) claimed to have taken the buyer space constraint into consideration in their models. However, they followed the method that Persona, *et al*. (2005) used to calculate the buyer average inventory and Persona, *et al*. (2005) did not take buyers' space constraint into consideration. With a buyer capacity limitation, both the inventory of the buyer and the vendor would be affected, which, in terms, would change the optimal solution of the number of delayed shipments. As a result, in Battini, *et al.* (2010a; 2010b) studies, the average inventory level of the buyer and the vendor are questionable as are the models developed there. Although the research showed the solution of the models they developed with a numerical example, the authors did not provide any information about how they arrived at the results.

Huang and Chen (2009) developed a joint EOQ/EPQ inventory model that considers the number of shipments, n, the size of each shipment, q, and the number of delayed shipments, k, as decision variables to minimize the annual average total cost $C(q, n, k)$. They also provided an algorithm to determine the optimal solution to their model. However, there are flaws in the methodology behind their algorithm in that they failed to prove that the objective function of their model is strictly convex over two of decision variables q, and n. This convexity property of the objective function is required to ensure that their algorithm always results in an optimum solution (Yi and Sarker 2013a; 2013b).

Other researchers, such as Hill and Omar (2006), and Huang and Chen (2009), tried to solve the reduced models. So far no other attempt has been made to find a solution

procedure to solve the generalized CS-k model. However, solving the general CS-k model, which reflects the effects of the buyer's space limitation, is more important than solving the reduced models since it is not often in practice that the vendor can build up as much inventory as s/he want in the buyer's warehouse.

2.3.3 Effects of variable demand

Contrast to the numerous studies on deterministic CS models, the research on an integrated CS inventory with variable demand is rare. On one hand, an unusual high demand at the early and the last stage of a production cycle may incur backorders to the buyer when its inventory level is relatively low. On the other hand, an unusual low demand in the middle of a cycle may leads to an extra space requirement, since the buyer's inventory level is kept at its highest level during that period. Therefore, a variable demand is crucial to the performance of such systems, especially when there is also buyer's limitation constraint. The issue of allowing uncertain demand rate needs to be carefully addressed.

2.3.4 Effects of obsolescence of product

Obsolescence means a finite lifetime of the products, implying the functions fulfilled by the products are no longer required. This is often due to some contextual situations such as the use of new technologies, marketing changes, competitions, etc. (Personna, *et al*., 2005). The risk of obsolescence is an issue that most supply chains must often face. Personna, *et al*. (2005) is the first to have studied this issue in the context of CS policy. The authors

integrated the risk of obsolescence into the basic CS(k=0) model and developed a deterministic inventory model that has one decision variable, the ordering quantity. Battini, *et al*. (2010a; 2010b) extended this study by allowing a variable demand rate. Battini, *et al*.'s (2010a; 2010b) model, follows Personna, *et al*.'s methods in calculating vendor and buyer's inventory. Therefore, their basic model is also the CS(k=0) model, which means the buyer's space limitation is not considered. Therefore, so far, no research has been done to a general CS-k model under the constraints of: controllable lead time, buyer's space limitation, variable demand and obsolescence.

2.3.5 Lack of a general solution procedure.

Starting from Corbett (2001), researchers have built up several production/replenishment CS models. Some solved their models with solution procedures or algorithms (Braglia and Zavanella, 2003; Piplani and Viswanathan, 2003; Hill and Omar, 2006; Huang and Chen, 2009; Srinivas and Rao, 2010; Yi and Sarker, 2013a) while others did not (Valentini and Zavanella, 2003; Lee and Wang, 2008; Battini, *et al*., 2010a; 2010b).

For most of the models that have been solved (Braglia and Zavanella, 2003; Piplani and Viswanathan, 2003; Hill and Omar, 2006; Huang and Chen, 2009; Srinivas and Rao, 2010; Yi and Sarker, 2013a), one common shortcoming of the developed solution procedures or algorithms is that they rely on the convexity of the objective function of the specific model. Or, in other words, they are based on the differential information of the objective function. Therefore, those procedures or algorithms can hardly be used to solve other models when

the objective functions of the other models are considerably different.

Another shortage of previous solution procedures is that they cannot solve relatively complicated problems with many variables. This is because, when a model considers more variables, it becomes more complicated, often associated with this complexity, the convex property of the objective function is lost. Even if the objective function is still convex, it is mathematically difficult to prove this property when the number of decision variable is high.

In short, there isn't a general solution procedure that can handle relatively complicated models with many variables or that can solve different models without considerable modification to the solution procedure. However, in practice, a typical supply chain often involves thousands of products, tens of echelons. Each of these echelons may have hundreds of different suppliers and distributors/retailers. As a result, the inventory models often contain hundreds, even thousands of variables. Moreover, different CS supply chains need to cope with different critical factors. Therefore, it is necessary to develop a solution method that can solve complicated problems and can be applied to different models.

2.4 THE DEVELOPMENT PHASES OF THIS DISSERTATION

This research develops a total of four sets of CS inventory models, each of which considers one more important impact factor than the previous one. Other than that, this study will also develop several hybrid meta-heuristic algorithms to solve the four sets of models. The relationship chart of the relevant research and the works of this dissertation is shown in Figure 2.1.

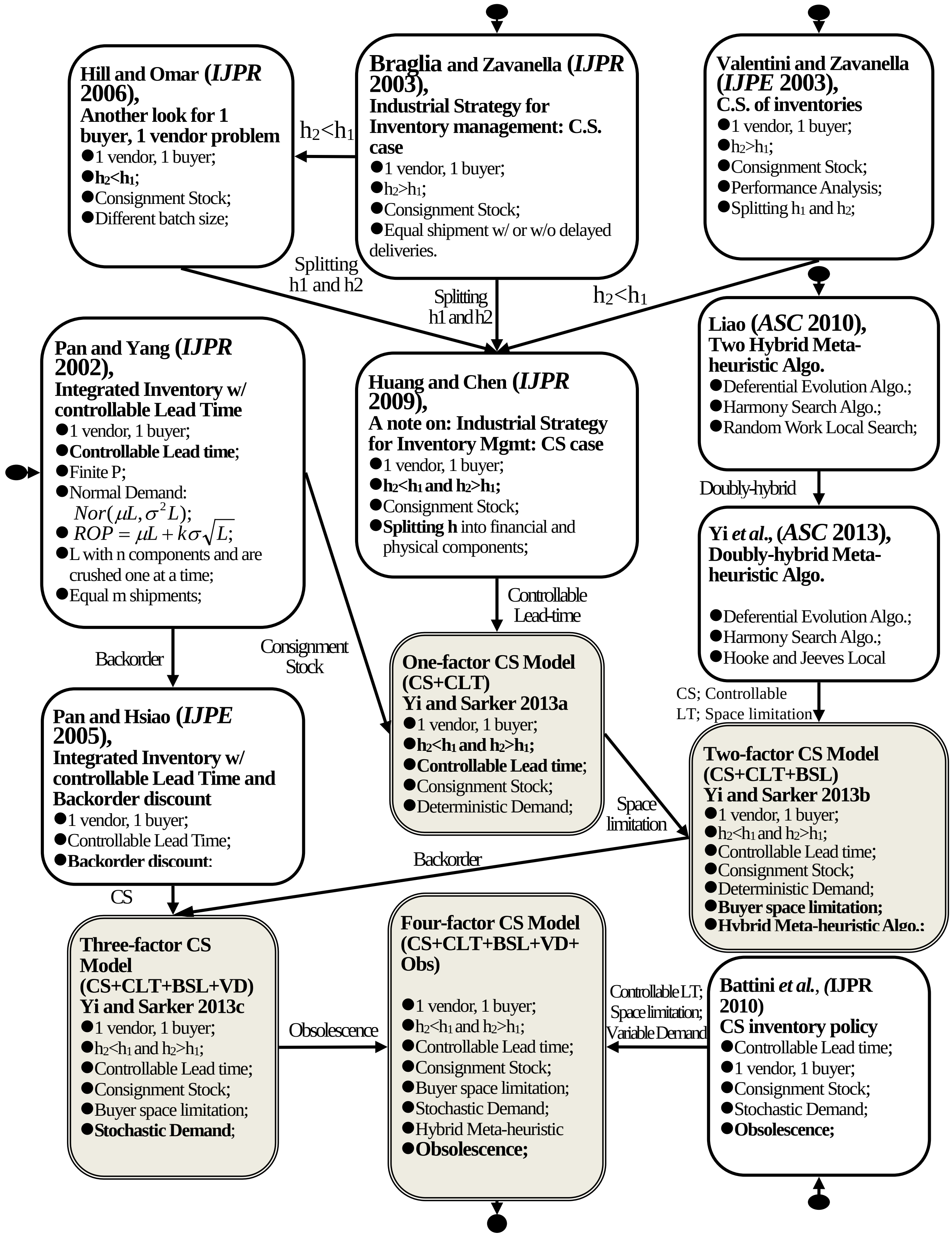


Figure 2.1 Relationship Chart of relevant research and this research

In Figure 2.1, the four sets of CS models of this study are listed inside four darken rounded squares. Specifically, the one-factor CS model is developed on top of the work of Huang and Chen (2009), and the work of Pan and Yang (2002). It is essentially a basic CS-k model plus a controllable lead time and contains three variables. The two-factor model is based on the work one and the work of Yi, *et al*. (2013). It is basically the one-factor model plus a buyer's space limitation and contains four variables. The three-factor CS model is developed on top of the two-factor model and allows the variability in the demand. It has five variables. The four-factor CS model, developed based on the three-factor CS model and the Battini, *et al*.'s (2010a) model, took into account the risk of obsolescence. Lastly, the fifth work of this dissertation, the doubly-hybrid meta-heuristics (Yi, *et al*., 2013), improves the existing heuristics used for consignment stocking policies.

CHAPTER III
SOLUTION METHODS

This Chapter deals with the solution methods used to solve all the optimization inventory models developed in this research. Because the later models are too complicated to be solved analytically, meta-heuristic methods are used. Therefore, the solution of this integrated production system problems is divided into two facets.

3.1 TWO FACETS OF THE SOLUTION METHODS

Most researchers use one of the following two ways to solve an optimization problem: either analytically or heuristically. The analytical way is always superior to the other in that it will be more efficient, and most importantly, it guarantees the solution to be optimal. However, when the objective function is non-differentiable, or when it is difficult to prove the convexity of the objective function, the traditional mathematical method may not be used to solve the problem. In that case, many researchers chose to use the second way: heuristic methods. Typically, a heuristic method is a searching method which does not require any mathematical information of the objective function, but may take more time and not guarantee an optimal solution. In this research, for all the inventory models developed, the mathematic way is used first. When it is not possible to solve the problem, then the heuristic methods will be used.

3.1.1 Analytical methods

The analytical methods require that the objective function and all the constraint functions are convex. The way of showing a function is convex is to show that the Hessian Matrix of the function is positive finite, which requires that the function is differentiable. If all the objective function and constraint functions are convex, then the optimal solution can be found by letting the first derivative of each variable equals zero and solve all those combined equations. Therefore, whether a function is analytically solvable depends on the following three conditions: (1) the function is differentiable, (2) the Hessian Matrix is positive finite, and (3) the combined equations are solvable.

These three conditions are not always easy to be satisfied. First of all, when the objective function is not continuous, then it will be definitely non-differentiable. On the other hand, a continuous function does not mean it is differentiable. Secondly, when the objective function is differentiable but the number of variable is large, it will be difficult to prove the convexity of the objective function. Lastly, even when the objective function is convex, the combined equations of the first derivatives of all the decision variables may still be difficult to solve. As a result, analytically solvable inventory problems often only consider one or two, or at most three decision variables. It is rare that a model considers more than four variables and is still solvable analytically.

The one-factor CS model developed here considers three decision variables; hence, an analytical solution of this model is available for this research. For the rest optimization

models, there is no traditional way of providing closed-form solutions. As a result, meta-heuristic methods are to be used.

3.1.2 Meta-heuristic methods

Meta-heuristic algorithms are approximate algorithms. Generally speaking, meta-heuristic methods orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space. Compared to conventional optimization methods, such as the gradient-based methods mentioned in the last Section, meta-heuristic algorithms are known to be more effective in handling non-differentiable, and non-convex multi-modal functions because they utilize no gradient information.

For the two-factor, three-factor, and four-factor CS models developed here, the objective functions are too complicated to show their convexity. As a result, an efficient doubly-hybrid meta-heuristic method is developed and is discussed below.

The remainder of this Chapter is organized as follows. Section 2 introduces the three new hybrid meta-heuristics. In Section 3, the results of a comprehensive comparative study are presented. Section 4 provides some further discussion on the main results by increasing the maximum number of function evaluations on selected problems and by fixing the step size in the Hooke and Jeeves based hybrids. Finally, Section 5 summarizes the Chapter.

3.2 Hybrid Meta-heuristics: General Solution Methods for CS Models

As mentioned earlier, most CS models developed so far only consider a very limited number of decision variables. This is mainly because of the mathematical difficulty in solving those complicated models that may not be convex over all the decision variables. In order to solve the optimization models to be developed here, a general solution method that does not rely on differentiation is necessary to be developed. In this Section, three doubly-hybrid meta-heuristic algorithms (DHMA), which can be used to solve relatively complicated models with multiple decision variables, but do not require the convexity of the objective function of the models, are to be presented.

As pointed out in Blum (2010), the research on meta-heuristics for optimization problems has shifted from an algorithm-oriented point of view to a problem-oriented point of view. As a result, more researchers are aiming at combining different meta-heuristics together to form hybrid algorithms that are more powerful, more efficient for solving problems. Of interest here are constrained mixed discrete-continuous optimization problems. Constrained mixed discrete-continuous optimization problems are widespread in the mathematical modeling of real world systems for a broad range of applications, including engineering design. It is the study of problems for which one wishes to optimize an objective function (usually of several variables) subject to a collection of restrictions on these variables. It is our belief that more robust hybrid meta-heuristics are promising tools to produce competitive results for these optimization problems. This study is part of our effort in advancing this line of research.

This study presents three new hybrid algorithms and improves upon the two hybrid algorithms described in a recent study by Liao (2010). The two hybrid algorithms presented by him are:

(1) A hybrid of differential evolution and random walk with direction exploitation (MA+MDE');

(2) A hybrid of differential evolution and harmony search (MDE'+IHS).

The three improved algorithms provided by this study are:

(1) A hybrid of differential evolution and Hooke and Jeeves local search method (MDE'+HJ), which is a memetic differential evolution algorithm modified the MA+MDE' by replacing the random walk with direction exploitation with the Hooke and Jeeves method;

(2) A doubly hybrid of differential evolution, harmony search, and Hooke and Jeeves local search method (MDE'+IHS+HJ), which is a memetic cooperative hybrid constructed by adding the Hooke and Jeeves method to the original cooperative hybrid, MDE'+IHS;

(3) Another doubly hybrid of particle swarm, harmony search, and Hooke and Jeeves local search method (PSO+IHS+HJ), which is also a memetic cooperative hybrid. It is a variation of MDE'+IHS+HJ by replacing IHS with PSO.

The effectiveness and efficiency of these three newly proposed hybrid meta-heuristics will be shown based on a total of 18 benchmark test problems.

In the following, the three new hybrid meta-heuristic algorithms proposed here and their major components are described and are presented in order.

3.2.1 Hybrid of modified DE and Hooke and Jeeves (MDE'+HJ)

The first new hybrid is a memetic hybrid algorithm. It essentially enhances the MDE' algorithm by an efficient direct search method, i.e. the Hooke and Jeeves method. The MDE' algorithm is comprised of three components: a slight modified version of the MDE algorithm proposed by Angira and Babu (2006), Deb's constraint handling method (2000), and a generalized discrete variable handling method to deal with mixed integer problems. The MDE' algorithm has been shown effective in solving constrained mixed integer engineering design problems (Liao, 2010).

The local search method is intended to further exploit the candidate solution generated by MDE' if it is selected randomly based on a pre-specified percentage, p = 0.1. The reason for applying local search probabilistically is to reduce the iterations necessary for applying local search, which is important when the computational cost of the local-search algorithm is high. The Hooke and Jeeves (*HJ*) direct search method was first developed by Hooke and Jeeves (1961). It is a classical and powerful local descent algorithm, making no use of the objective function derivatives. In this research, it is probabilistically applied as a subordinate, embedded component to exploit promising areas suggested by the corresponding main algorithm (single or cooperative hybrid). The probability to perform this local search is consistently set at 0.1 in this research. The flow charts of the improved differential evolution local search hybrid,

MDE’+HJ, implemented for this research is given by Figure 3.1. Interested readers may refer to Appendix B-1 for its pseudo codes. The flow chart of HJ is not presented here since it is so popular and well-known. Interested reader can refer to Appendix B-2 for its pseudo code.

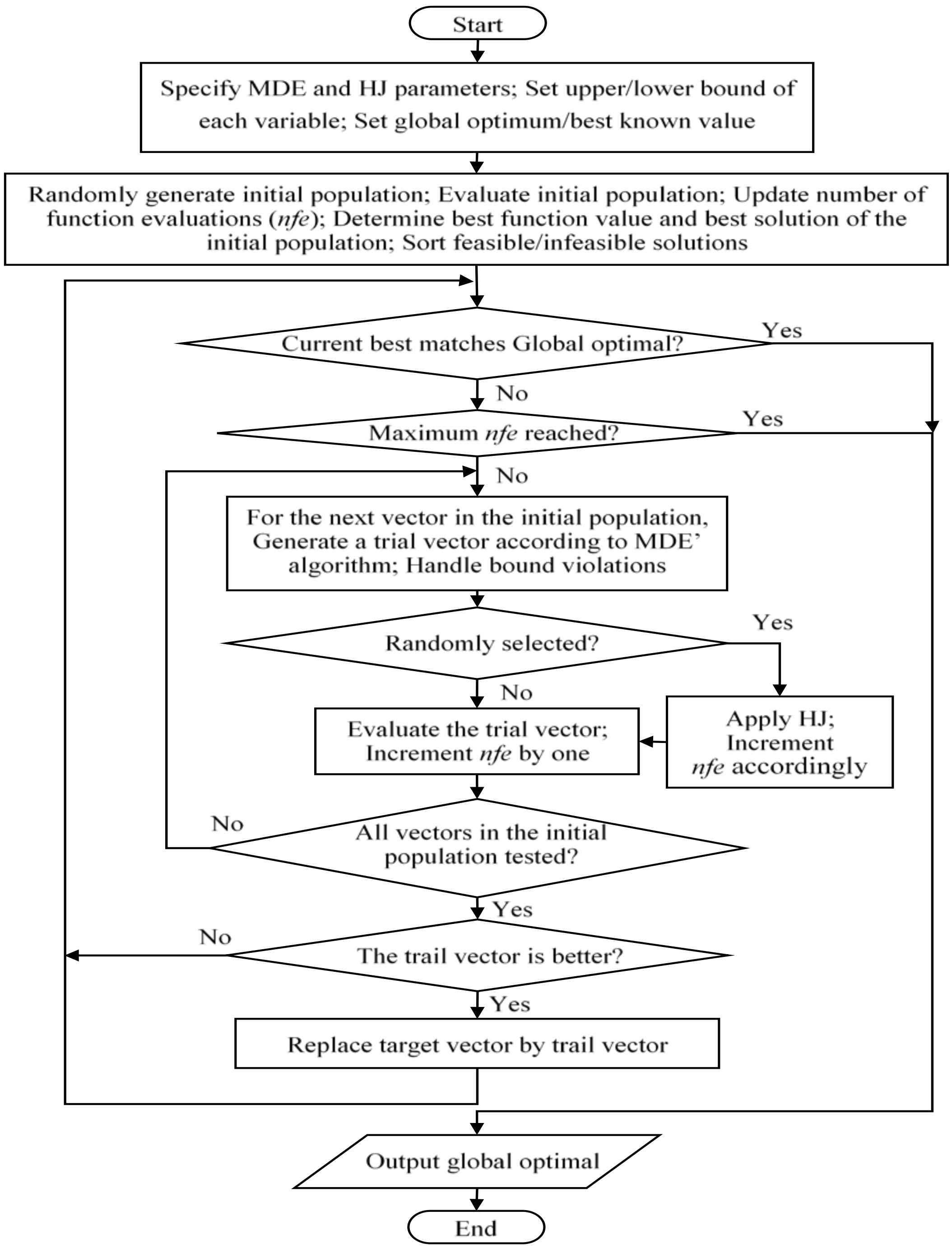


Figure 3.1 The flow chart for MDE’+HJ Hybrid Algorithm

3.2.2 Hybrid of modified DE, improved HS and Hooke and Jeeves (MDE'+IHS+HJ)

The second hybrid, named MDE'+IHS+HJ, is constructed by adding the Hooke and Jeeves to the cooperative hybrid, i.e. MDE'+IHS. Harmony search is a relatively new meta-heurisitc, proposed by Geem, *et al*. (2001). The harmony search algorithm is derived from an artificial phenomenon found in musical performance, mimicking the process of search for better harmony. The main idea is to treat the optimization algorithm seeking a global optimum determined by objective function as the musical performance seeking a fantastic harmony determined by aesthetic estimation. The musician's improvisations are analogous to explore and exploit search operators in optimization plans. The harmony memory with size *HMS* is initially populated with randomly generated solutions sorted by their objective function values. Next, a new trial harmony is improvised from the harmony memory using a stochastic random search mechanism based on the harmony considering rate (*HMCR*) and the pitch adjusting rate (*PAR*).

In this study, we employ the improved harmony search (IHS) proposed by Mahdavi, *et al*. (2007) by adapting two parameters: pitch adjusting rate, *PAR* and band-width, *bw*. For each improvisation step t, *PAR* and *bw* are determined as follows:

$$PAR(t) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{NI} \cdot t\,, \tag{3.1}$$

$$bw(t) = bw_{\max} \cdot \exp(t \cdot c)\,, \tag{3.2}$$

$$c = \frac{Ln(bw_{\min} / bw_{\max})}{NI}\,. \tag{3.3}$$

In the above Equations, NI, $PAR_{\min}$, $PAR_{\max}$, $bw_{\min}$ and $bw_{\max}$ denote number of improvisations, minimum/maximum pitch adjusting rate/bandwidth, respectively. In this research, the current number of evaluations in IHS hybrids is used as improvisation step t. IHS collaborates with MDE' correspondingly in which each algorithm perform search independently while information are only exchanged on improved trail solutions. The HJ local search is applied to exploit the candidate solutions generated by MDE' and IHS if they are randomly selected based on a pre-specified percentage, p=0.1. The flow chart of the pseudo code of the MDE'+IHS+HJ hybrid is summarized in Figure 3.2 and the pseudo code can be found in Appendix B-3.

3.2.3 Hybrid of PSO, Modified DE and Hooke and Jeeves (PSO+MDE'+HJ)

The third new hybrid, named PSO+MDE'+HJ, is a variation of the second new hybrid MDE'+IHS+HJ by replacing IHS with PSO. Particle swarm optimization (PSO), first introduced in 1995 by Kennedy and Eberhart (1995), is one popular meta-heuristic that uses a metaphor of a natural flock or swarm of birds. Each individual bird is called a particle. These particles would adjust their flying direction based on their own previous best performance and the best previous performance of their neighbors. PSO is chosen as a candidate for possible improvement because it has gained widespread appeal among researchers, thanks to its fast computing ability and good performance in a variety of application domains.

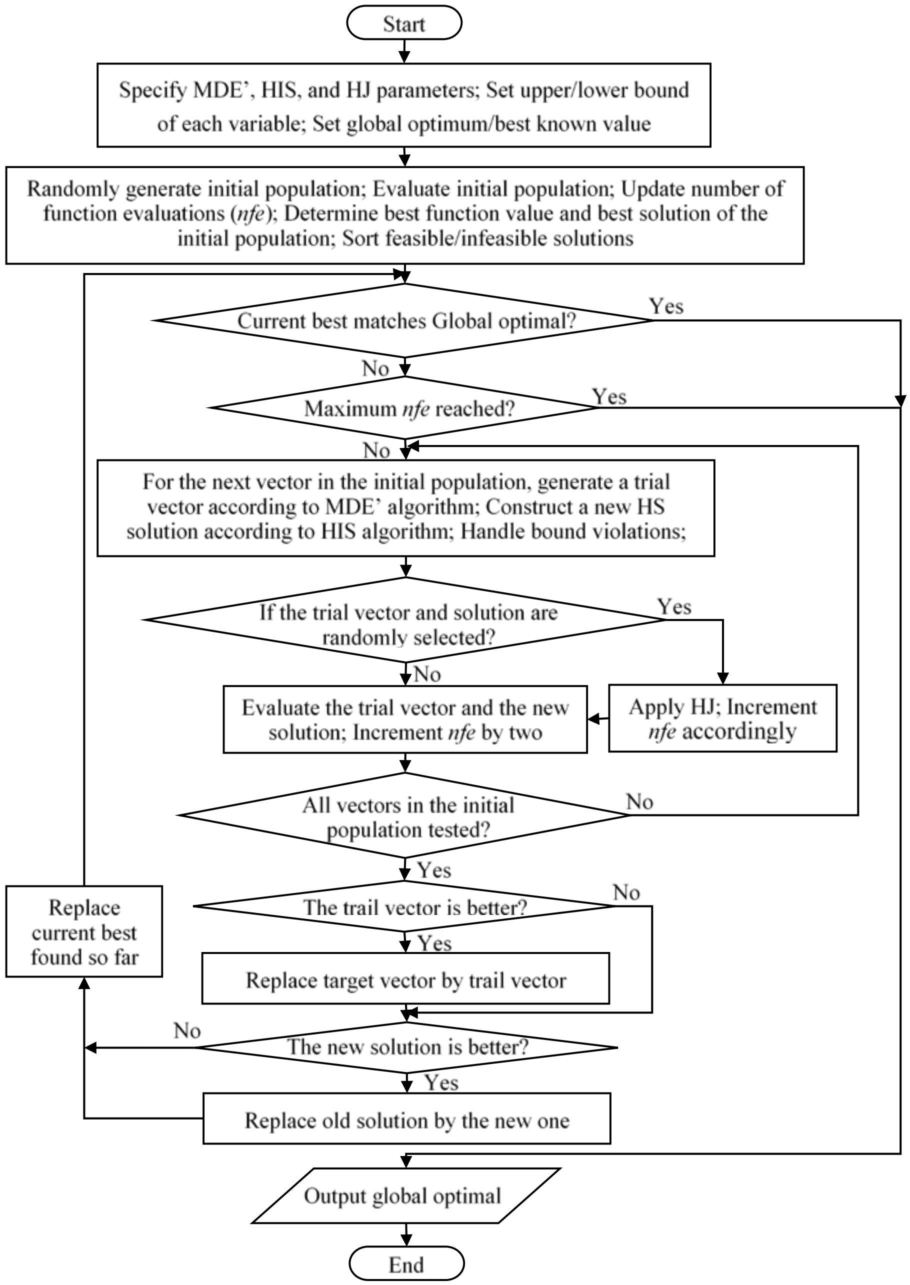


Figure 3.2 The flow chart for MDE'+IHS+HJ Hybrid Algorithm

The PSO algorithm simulates the swarm by calculating the flying velocity of each particle at each iteration according to the location of the global best particle (*gbest*) among the swarm and its personal best location (*pbest*) it remembers, according to the following formulas:

$$V_i^t = \omega V_i^{t-1} + \varphi_1 rand(P_i^{t-1} - X_i^{t-1}) + \varphi_2 rand(G^{t-1} - X_i^{t-1}), \tag{3.4}$$

$$X_i^t = X_i^{t-1} + V_i^t. \tag{3.5}$$

In the above two Equations, $X_i^t = \{x_{i,1}^t, x_{i,2}^t, ..., x_{i,j}^t, ..., x_{i,N}^t\}$ denotes the position of the *i*th particle at the *t*th iteration and $x_{i,j}^t$ denotes the *j*th dimension of the position of the *i*th particle at the *t*th iteration; $V_i^t = \{v_{i,1}^t, v_{i,2}^t, ..., v_{i,j}^t, ..., v_{i,N}^t\}$ denotes the velocity of the *i*th particle at the *t*th iteration and $v_{i,j}^t$ denotes the *j*th dimension of the velocity of *i*th particle at the *t*th iteration; ω is the inertia weight, a parameter tuning the influence of the velocity of a particle at the last iteration; φ_1 and φ_2 are two parameters determining the impact of global best and particle best, G^{t-1} is the global best at *t-1*th iteration and P_i^{t-1} is the particle best for the *i*th particle at *t-1*th iteration; rand is a random number that is uniformly distributed in [0, 1].

The specific PSO used in this study is the one proposed by Iwasaki, *et al*. (2006) in which the inertia weight is adaptive. As mentioned earlier, $v_{i,j}^t$ is the velocity of particle *i* (*i=1,…, M*) along dimension *j* (*j=1,…, N*) at iteration *t*, the average absolute value of velocity of all the particles at iteration *t*, v_{avg}^t, is computed according to the follow formula:

$$v_{avg}^t = \frac{1}{M - N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| v_{ij}^t \right|. \tag{3.6}$$

The ideal average velocity is set to a start value, v_{start}, and then linearly decreased to zero at T_{end} before reaching the end of the search indicated by Maxnfe. v_{start} is set as the maximum range of each variable for each problem. T_{end} is set as 80% of Maxnfe. Their

adaptive strategy operates as follows: if the current average velocity of the particles is larger than the ideal velocity, the parameters of inertia weight are shifted to convergent values. Otherwise, parameters are shifted to divergent values. The shift amount, $\Delta\omega$, maximal inertia weight, $\omega_{\max}$, and minimal inertia weight, $\omega_{\min}$, must be pre-specified. Mathematically,

$$If\ v_{avg}^{t} > v_{ideal}^{t},\ \omega = \max\{\omega - \Delta\omega, \omega_{\min}\} \tag{3.7a}$$

$$If\ v_{avg}^{t} < v_{ideal}^{t},\ \omega = \min\{\omega + \Delta\omega, \omega_{\max}\} \tag{3.7b}$$

This version of adaptive inertia weight is adopted here as an alternative to prevent a sharp reduction in search capabilities at higher dimensions. Similarly, the HJ local search exploits the candidate solutions generated by MDE’ and PSO if they are randomly selected based on a pre-specified percentage, *p*=0.1. The flow chart of the PSO+MDE’+HJ hybrid is shown by Figure 3.3 and the pseudo code is given in Appendix B-4.

3.3 COMPUTATIONAL RESULTS

The performances of all three hybrid meta-heuristics as described in Section 2 were verified experimentally using a set of 18 constrained mixed integer optimization problems. All programs were coded in Matlab and all executions were made on a HP Pavilion dv8 with Intel® Core ™ i7 CPU Q720 @ 1.60 GHz.

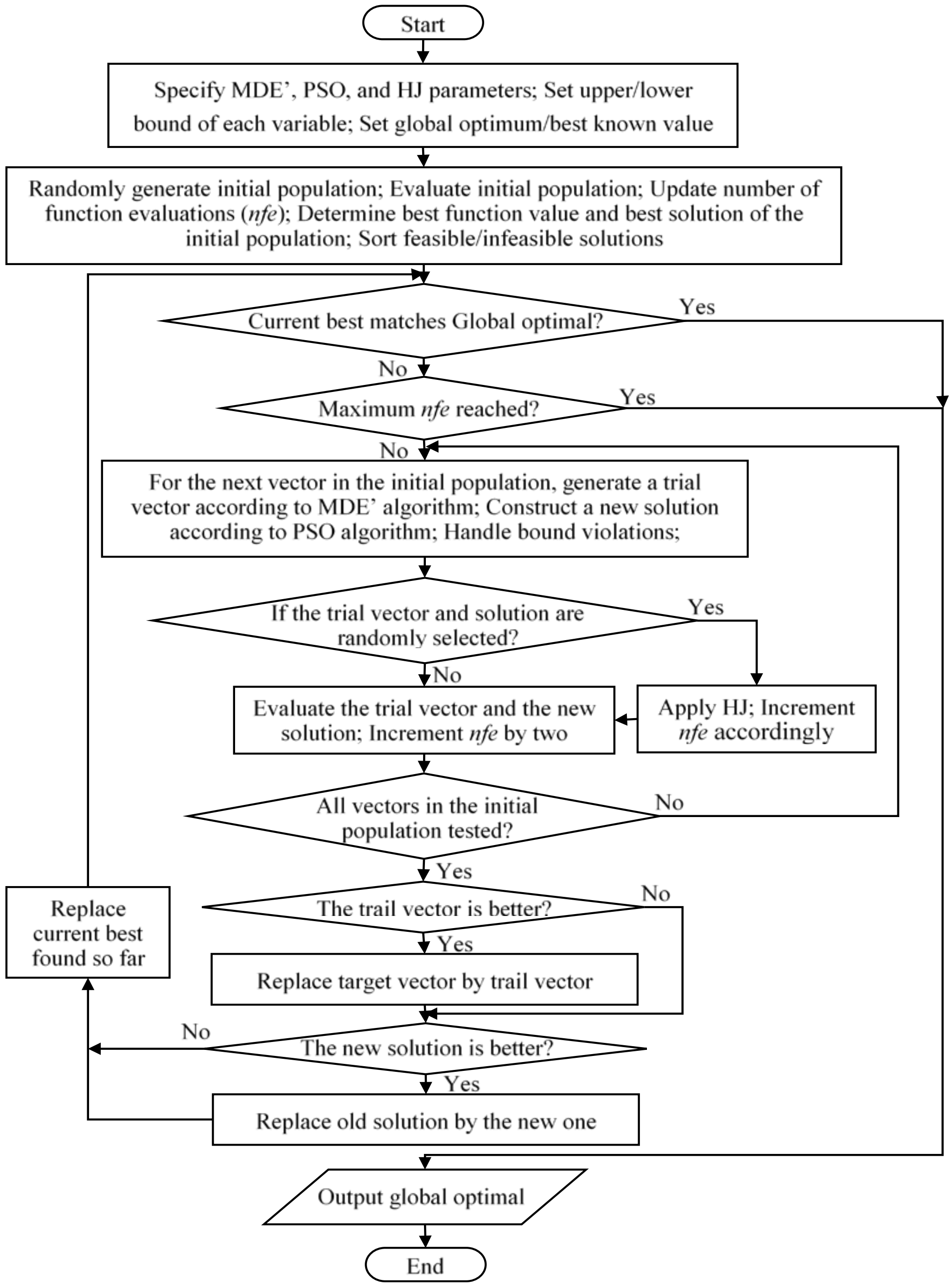


Figure 3.3 The flow chart for MDE'+PSO+HJ Hybrid Algorithm

3.3.1 Problems used to test the efficiency of the proposed algorithms

A total of six meta-heuristic algorithms, including the three improved hybrids proposed here (MDE'+HJ, MDE'+IHS+HJ, and PSO+MDE'+HJ) and the three original hybrid

algorithms (MDE', MA+MDE', and MDE'+IHS), were tested with a set of 18 different engineering problems. All these problems have previously appeared in literature. Specifically, problems 1–3 were used in Mathur, *et al*. (2000) whereas problems 4–9 were used in Angira and Babu (2006), all in the field of process synthesis and design. Functions 10–12 were used in Yokota, *et al*. (1996), in the field of system reliability. Function 13 was used in Kitayama and Yasuda (2006), in the field of pressure vessel design. Function 14 was used in Lee, *et al*. (2007), in the field of manufacturing process design. Functions 15-18 were taken from Chen (2006), in the field of system reliability redundancy allocation problems. Since all those functions have been used in literature, the best solution for each function is known.

Our test results show that three hybrid algorithms proposed here can generate more accurate results than previous studies in less time without being trapped in local minima. For fair comparison, all parameter settings are fixed for all testing problems. Those settings were found to work well based on previous research (Liao, 2010). Table 3.1 summarizes the settings of all relevant algorithmic parameters (organized into groups related to MDE', IHS, PSO and HJ, respectively). The parameter settings for RWDE follow those in Liao (2010). However, the maximal number of function evaluations, Maxnfe, is varied from problem to problem depending upon its difficulty. A run is declared as successful if the global optimum (or the known best) was found within 10^{-6} error. For each test problem, 100 runs were made and the mean and standard deviation of objective values, number of function evaluations taken and elapsed CPU time were recorded in Tables 3.2-3.4, respectively.

Table 3.1 Global algorithmic parameters for MDE', IHS, PSO and HJ

MDE'	
Population Size	NP=10×Dimensions of the problem
Scale Factor	F=0.5
Crossover Rate	CR=0.95
IHS	
Harmony memory size	HMS=10×Dimensions of the problem
Harmony considering rate	HMCR=0.99
Number of improvisations	NI= MaxFE
Minimum pitch adjusting rate	PAR_min=0.45
Maximum pitch adjusting rate	PAR_max=0.99
Minimum bandwidth	bw_min=1×10^{-5}
Maximum bandwidth	bw_max=4
PSO	
Population size	PSO=10×Dimensions of the problem
Maximal inertia weight	$\omega_{max} = 0.9$
Minimal inertia weight	$\omega_{min} = 0.4$
Shift amount	$\Delta\omega = 0.1$
Impact factor 1	$\varphi_1 = 1.3$
Impact factor 2	$\varphi_2 = 1.3$ $\lambda = 0.1$
HJ local search	
Step size	$\lambda = 0.1$
Maximum number of iterations	$M = 10$
Probability to perform HJ local search	$p = 0.1$

3.3.2 Results obtained by previous algorithms

The three hybrid algorithms used as the base of this study were retested here and their results serve as the baseline so that the improvement of the three proposed hybrid algorithms can be clearly quantified. As shown in Table 3.2, the results obtained by algorithms presented by previous researchers are listed in the first three columns, and the results obtained by algorithms proposed in this research are listed in the last three columns. Using the previous algorithms to all of the 18 problems tested, it is found that MA+MDE' is the best among the three previous algorithms in terms of total success rate with a value of

13.58. However, MDE'+IHS is shown here to be the one with a perfect success rate in the most number of problems (5 out of 18). This finding is slightly different from that of Liao (2010) in which he showed that MDE'+IHS was slightly better than MA+MDE' (by 0.3 successful rate) in their problem setting. The difference is probably due to the fact that the pool of testing problems has been extended to cover four more problems; our results are based on the average of more runs (100 vs. 30); and the stochastic nature of the meta-heuristic algorithms produces different results in different runs.

Table 3.2 Summary of success rate obtained by all six algorithms

		Previous Algorithms			Algorithms Proposed by this study		
Problem	Maxnfe	MDE'	MA+MDE'	MDE'+IHS	MDE'+HJ	MDE'+IHS+HJ	PSO+MDE'+HJ
1	15,000	0.54	0.91	0.84	1.00	0.96	1.00
2	100,000	0.04	0.13	0.17	0.74	0.94	0.73
3	15,000	0.91	0.93	0.97	0.00	0.48	0.00
4	5,000	0.91	0.88	0.96	0.99	0.95	0.88
5	5,000	0.65	0.84	1.00	0.79	0.86	0.49
6	50,000	0.42	0.65	0.38	0.9	0.83	0.71
7	1,797	0.97	0.96	0.75	0.91	0.99	1.00
8	50,000	0.54	0.85	0.72	0.03	0.81	0.28
9	5,495	1.00	1.00	1.00	1.00	1.00	1.00
10	50,000	0.93	0.94	1.00	0.47	0.92	0.89
11	1,000	1.00	1.00	0.88	1.00	1.00	1.00
12	14,000	1.00	1.00	1.00	1.00	1.00	1.00
13	50,000	0.17	0.17	0.08	0.76	0.50	0.99
14	10,000	0.99	0.98	1.00	1.00	1.00	1.00
15	400,000	0.46	0.52	0.27	0.7	0.47	0.46
16	10,000	0.04	0.00	0.54	1.00	1.00	0.99
17	200,000	0.76	0.85	0.81	0.98	1.00	0.99
18	50,000	0.98	0.97	0.93	0.78	1.00	0.88
Total		12.31	13.58	13.30	14.05	15.71	14.29

Table 3.3 Summary of average numbers of evaluations taken by all six algorithms

		Previous Algorithms			Algorithms Proposed by this Study		
Problem	Maxnfe	MDE'	MA+MDE'	MDE'+IHS	MDE'+HJ	MDE'+IHS+HJ	PSO+MDE'+HJ
1	15,000	7,696	3,901	3,731	5,859	6,589	4,596
2	100,000	96,070	87,422	85,048	28,389	10,522	20,910
3	15,000	7,912	13,254	6,259	15,795	15,116	15,511
4	5,000	1,057	1,677	3,290	1,787	1,211	1,863
5	5,000	1,984	1,241	1,290	1,721	1,251	2,776
6	50,000	30,030	23,462	45,764	15,964	21,890	22,929
7	1,797	426	670	642	994	458	412
8	50,000	27,329	20,546	19,947	50,210	45,821	49,206
9	5,495	1,023	1,913	403	495	453	555
10	50,000	17,567	30,951	3,955	43,090	13,152	24,484
11	1,000	222	388	241	285	221	288
12	14,000	1,460	2,524	1,070	1,704	1,762	1,414
13	50,000	42,108	42,632	46,451	30,138	32,618	18,265
14	10,000	1,603	2,856	2,977	3,058	1,747	2,419
15	400,000	225,865	212,619	296,556	351,082	317,981	349,881
16	10,000	9,936	10,100	7,178	2,935	3,283	3,060
17	200,000	63,276	64,018	51,268	181,044	110,522	130,366
18	50,000	6,283	11,755	7,423	27,190	15,398	20,984
Total		541,847	531,930	583,494	761,740	599,995	669,919

The performance of each algorithm in terms of average numbers of function evaluations, and CPU time taken over 100 runs is given in Tables 3.3 and 3.4, respectively. In each run, the numbers of function evaluations will be lower than the Maxnfe value only if it satisfies the global optimum within 10^{-6} error before Maxnfe is reached. Again, it is found that MA+MDE' is the best among the three previous algorithms in terms of both total average numbers of function evaluations and total average CPU time taken. As expected, numbers of function evaluations taken generally increase with the difficulty of problem and the maximal number of function evaluations selected. Problem 15 is the most difficult problem in which it requires significantly large number of function evaluations and more CPU time.

Table 3.4 Summary of CPU time (in seconds) taken by all six algorithms

		Previous Algorithms			Algorithms Proposed by this Study		
Problem	Maxnfe	MDE’	MA+MDE’	MDE’+IHS	MDE’+HJ	MDE’+IHS+HJ	PSO+MDE’+HJ
1	15,000	7.87	1.58	1.31	0.58	0.74	0.33
2	100,000	69.32	35.56	31.58	4.03	1.47	1.70
3	15,000	8.59	8.32	3.90	1.80	0.90	1.27
4	5,000	0.63	0.71	1.15	0.28	0.16	0.19
5	5,000	1.21	0.75	0.81	0.30	0.18	0.27
6	50,000	18.22	11.54	26.48	2.42	2.71	2.25
7	1,797	0.27	0.27	0.22	0.17	0.05	0.04
8	50,000	17.93	12.17	11.78	5.84	4.01	3.63
9	5,495	0.63	0.78	0.19	0.07	0.07	0.05
10	50,000	12.21	22.25	2.37	9.95	2.49	3.31
11	1,000	0.15	0.18	0.11	0.09	0.07	0.06
12	14,000	1.02	1.16	0.47	0.22	0.26	0.15
13	50,000	27.44	27.13	32.33	4.51	3.60	1.90
14	10,000	0.98	1.15	1.22	0.48	0.26	0.23
15	400,000	147.90	133.59	166.76	27.66	32.11	24.02
16	10,000	11.35	7.55	5.63	0.44	0.36	0.23
17	200,000	71.66	41.03	34.75	32.43	14.06	12.83
18	50,000	6.75	5.00	3.76	3.98	1.75	1.52
Total		404.11	310.73	324.83	95.25	65.25	53.95

3.3.3 Results obtained by the three new hybrid algorithms

Based on the test results, our first observation is that RWDE cannot be regarded as an efficient local search method because it generates a sequence of approximations of the optimizer by assuming a random vector as a search direction. Compared to RWDE, Hooke and Jeeves (*HJ*) local search is a more powerful local descent algorithm and it makes no use of the objective function derivatives. All three newly proposed hybrid algorithms make use of the Hooke and Jeeves local search.

From Table 3.2, it can be observed that the performance has been significantly improved by employing an efficient local search method. The total success rate over 18

problems for the three new hybrids arranged in descending order are MDE'+IHS+HJ, PSO+MDE'+HJ, and MDE'+HJ with values of 15.71, 14.29, and 14.05, respectively. Note that all of them are better than 13.58, the best of the three previous algorithms. Comparing MDE'+HJ with MA+MDE', by replacing RWDE with HJ improves the total success rate by 0.47 or 3.46%. Comparing MDE'+IHS+HJ with MDE'+IHS, by adding the HJ local search improves the total success rate by 2.41 or 18.12%. By comparing PSO+MDE'+HJ with MDE'+IHS+HJ, by replacing IHS with PSO reduces the total success rate by 1.42 or 9.04%. Therefore, PSO does not serve as well as IHS with MDE'. Nevertheless, PSO+MDE'+HJ is still better than MA+MDE' and MDE'+IHS by 0.71 (or 5.23%) and 0.99 (or 7.44%). By examining Tables 3.3 and 3.4, it is observed that all three new hybrids require higher number of evaluations but take less CPU time than the three previous algorithms. In terms of required number of evaluations, among the three new hybrids the best is MDE'+IHS+HJ, followed by PSO+MDE'+HJ, and then MDE'+HJ. In terms of CPU time taken, among the three new hybrids the best is PSO+MDE'+HJ, followed by MDE'+IHS+HJ, and then MDE'+HJ. This is an interesting observation because solo IHS-based algorithms may be faster than PSO. Possible explanation can due to the cooperative searching strategy. Even though the solo IHS-based algorithm may work faster than PSO algorithm, it does not necessarily apply to their corresponding hybrids. According to our results, the efficiency of hybrid meta-heuristic does not strictly follow its solo components. The cooperative framework, if designed properly, will increase the robustness of the global search

substantially and perform more efficient search, even for a relatively slow solo complementary algorithm. Algorithm designers should try different combinations and tailor different coordination mechanism for various meta-heuristics. Overall, it can be concluded that integrative cooperative hybrids such as MDE'+IHS+HJ and PSO+MDE'+HJ are better than integrative hybrids such as MDE'+HJ and MA+MDE' and cooperative hybrids such as MDE'+IHS and PSO+MDE' (not shown due to worse results).

One would expect that there is a positive correlation between required number of function evaluations and CPU time taken, but the results given in Tables 3.3 and 3.4 show the contrary. In order to further examine the reason behind this, the number of function evaluations taken is divided into two categories: function evaluations inside and outside the local search. The function evaluations outside the local search represent the number of the subject function evaluated by MDE' or IHS. Table 3.5 summarizes the results. It can be seen that most function evaluations occur inside the local search. This is due to the fact that HJ local search method performs coordinate search efficiently. It performs exploration search and pattern search on each dimension of a trial solution. Therefore, possible direction of improvement is generated quickly and introduced into the subject function to compute its objective value. This procedure is N times faster than RWDE (depends on the problem dimension N) because in RWDE, only one search direction is used at a time. Comparing to the solution generation mechanism for MDE' or IHS, which is more complicated and requires more steps, HJ local search reduces the computing burden in a way that does not

compromise accuracy. This explains why the HJ hybrids are able to perform more function evaluations with high efficiency and achieve more accurate results in less time. The overall performance of HJ-based hybrids are encouraging, especially for the MDE'+IHS+HJ hybrid. For seven out of eighteen problems, the MDE'+IHS+HJ hybrid find the global optima within 10^{-6} error in all 100 runs and it is the best in terms of total success rate.

Table 3.5 Separate counting of function evaluations on two HJ-based hybrids

Problem	Maxnfe	MDE'+HJ		MDE'+IHS+HJ	
		Inside HJ	Outside HJ	Inside HJ	Outside HJ
1	15,000	5,313.70	546.15	5,873.37	715.56
2	100,000	25,356.12	3,032.77	9,406.33	1,115.36
3	80,000	48,662.92	2,817.47	27,468.28	1,749.90
4	5,000	1,458.10	329.36	980.35	230.58
5	10,000	1,854.59	566.46	846.93	227.54
6	100,000	18,907.64	3,681.26	17,376.40	3,473.30
7	1,797	805.20	188.92	383.62	74.82
8	100,000	77,968.80	4,847.74	47,838.23	3,294.64
9	5,495	399.42	95.43	353.75	99.48
10	100,000	45,734.49	9,098.38	13,370.21	3,056.88
11	1,000	166.98	117.98	125.24	95.60
12	1,4000	1,545.25	158.49	1604.6	157.32
13	150,000	39,105.92	3,832.58	62,085.44	5,919.64
14	10,000	2,573.14	485.27	1,457.65	289.16
15	1,000,000	456,670.84	20,228.96	564,324.39	23,619.66
16	10,000	2,656.06	278.47	3,061.95	221.36
17	200,000	173,873.73	7,170.47	105,840.79	4,680.80
18	50,000	25,974.48	1,215.50	14,673.85	723.92

3.3.4 Convergence profiles of all algorithms for selected problems

Among all algorithms considered, MDE'+IHS+HJ and PSO+MDE'+HJ are the two successful hybrids with the largest margin of improvement. Especially, MDE'+IHS+HJ is the

best one in terms of success rate, better than the best hybrid in previous algorithms by over 15% and better than the second best, PSO+MDE'+HJ in this study, by nearly 10%. In order to observe how each hybrid performs its search and gain some useful insights, the convergence profiles of the average best solution over all 100 runs of each hybrid algorithm for three selected problems (2, 3, and 13) are shown in Figures 3.4-3.6, respectively. These three problems are selected for detailed analysis because the differences in success rate between hybrid strategies are generally more significant. It will assist in finding the difference between different hybrids because different searching patterns for these hybrid strategies can be observed more clearly. In addition, they cover problems of different levels of complexity. Convergence profiles of the average of the average objective value of the entire population over all 100 runs can also be shown. They are, however, not shown because of the difficulty in plotting them all in one figure where some strategies have infinite values. The infinite value is the result of the Deb's constraint handling method (Deb, 2000) used here, which sets the objective value of an infeasible solution to be infinite and the average is rendered to be infinite if any strategy does not find a feasible solution in any one run.

Figure 3.4 shows the convergence profiles of all six algorithms for Problem 2. Note that Problem 2 is a relatively difficult problem in terms of successful rate, ranging from 4% to 94% between algorithms with average value of 45.8% over all algorithms. Firstly, the starting point of each profile marks the time when the solution becomes feasible. Secondly, the final magnitude of the convergence profile is kind of an indirect indicator of the % successful rate.

Note that Problem 2 has a global optimum of 7.667. Generally speaking, the lower the final magnitude, the higher the successful rate. All newly proposed hybrids are able to converge to lower objective values comparing to those previous hybrids with an improvement around 0.2.

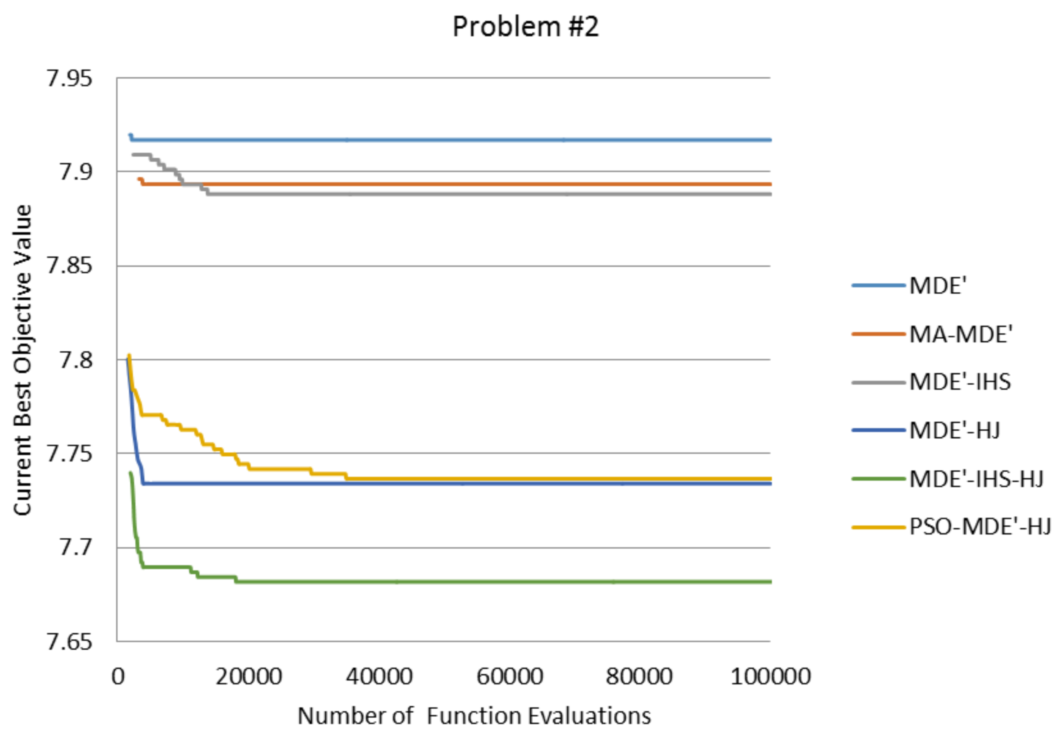


Figure 3.4 Average convergence rates of the best solutions of problem 2 over 100 runs

Among those new hybrids, MDE'+IHS+HJ converged to the lowest value, which matches with the fact that it achieves the highest accuracy of 94% on this problem. The convergence profiles of MDE'+HJ and PSO+MDE'+HJ differ in the early stage of searching process but later they converge to be close to each other with MDE'+HJ slightly better, which is consistent with the results of success rate with MDE'+HJ at 74% and PSO+MDE'+HJ at 73%. The success rates achieved by the three previous algorithms are considerably lower, ranging from 4%~17%. Their convergence profiles are very flat which indicates their inability

to converge to lower values. It appears that whether an algorithm is successful in finding the optimum depends on how low the algorithm can go in the early phase of searching process. If trapped at local optima, it is difficult for the subject algorithms to escape to find better solutions. In this respect, the newly proposed hybrids are more successful in guiding the search into the right direction without getting trapped in local optima at very early stage of the searching process.

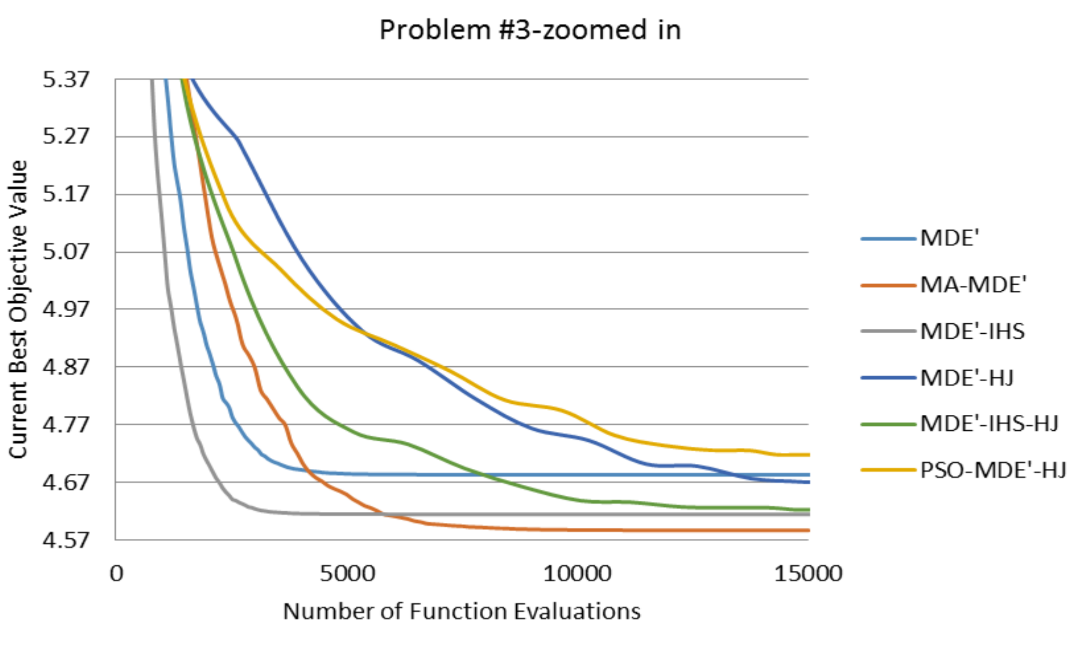


Figure 3.5 Average convergence rates of the best solutions of problem 3 over 100 runs (zoomed-in view)

Figure 3.5 shows the corresponding convergence profiles of all six algorithms for Problem 3. Note that Problem 3 is a bit easier than Problem 2 with success rate ranging from 0 to 97% and average of 54.8%. In order to show the end parts of the convergence profiles more clearly, Figure 3.5 provides a zoomed-in view of the actual profiles. Note that Problem 3 is one of a few problems for which all the newly proposed hybrids fail to perform better than those

previous algorithms. This result is also reflected in those profiles. Among all hybrids, MA+MDE' ranks first in terms of average of best solutions (the lowest magnitude of profile), followed by MDE'+IHS. The ranking is not consistent with their corresponding success rates probably because the profile is a measurement of the average of best solutions over 100 runs. The best solutions found by MA+MDE' may not be within the 10^{-6} error of the truly optimum as many as MDE'+HIS, but for those unsuccessful runs, their objective values may be close to the true optimum, much closer than those generated by MDE'+IHS. Similarly, MDE'+IHS+HJ ranks third in the convergence profiles while its success rate is only 48%, lower than the success rate obtained by MDE'. MDE'+HJ also achieve slightly lower magnitude in convergence profiles than MDE'. Overall, the newly proposed three hybrids are not efficient in finding the optimal solution for Problem 3 based on their success rates. Two of them, MDE'+HJ and PSO+MDE'+HJ, do not find the optimal solution at all in all 100 runs according to Table 3.2. However, the ranking of convergence profiles do not strictly follow their corresponding success rates, implying that their objective values are close to the truly optimum. The trend seems to indicate that better results might be obtainable if the maximum number of function evaluations is increased. On this point, further discussion is given in Section 3.4.1.

Similarly, Figure 3.6 shows the convergence profiles of all six algorithms for Problem 13. Note that Problem 13 is the most difficult problem among the three selected with the average success rate of only 44.5% over all algorithms. To better distinguish the differences, Figure

3.6 also provides a zoomed-in view of the actual profiles focusing on the end part of the search of each algorithm. It can be observed that different algorithms perform rather differently on this difficult problem. Overall, the three newly proposed hybrids are all more efficient than the three previous algorithms. Among the three new hybrids, the PSO+MDE'+HJ hybrid performs especially well for this problem.

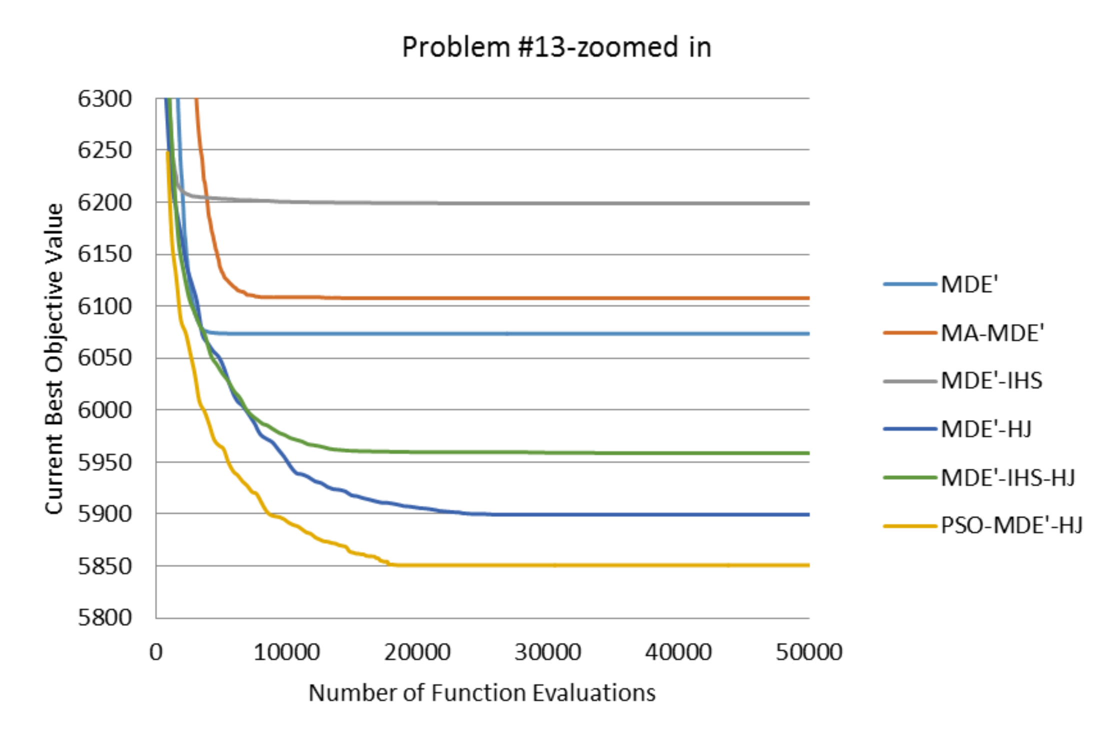


Figure 3.6 Average convergence rates of the best solutions of problem 13 over 100 runs (zoomed-in view)

3.4 Discussion

The results presented in the previous Section show that the overall performances of three newly proposed hybrids, MDE'+IHS+HJ, PSO+MDE'+HJ and MDE'+HJ (ranked by their success rates), are all better than the three recent developed hybrids. In this Section, some experiments are carried out to provide more detailed information.

3.4.1 Effect of increased number of maximum evaluations

The results presented in Section 3.3 indicate that all three new hybrids require more evaluations but generally take less CPU time. As mentioned in Section 3.3.4, it is possible that the newly proposed hybrids will perform even better when allowing more evaluations to run. Therefore, it is of interest to investigate the effect of maximum evaluations allowed (Maxnfe) on the performance of each new hybrid. Experiments on selected problems are run and the detailed results are listed in Table 3.6. The selection of problem is based on observation of their corresponding convergence profiles. For example, the profile for a specific problem should show possible room for improvement (the profile is usually not flat and there is a downward trend, as shown in Figure 3.5).

For ease of comparison, the upper part of Table 3.6 (Pre-determined maximum number of evaluations) is just a reorganization of the corresponding results in Table 3.5 and the bottom part of Table 3.6 (Increased maximum number of evaluations) is the results from the new experiments. The first and second columns indicate the problem selected for testing and the maximum evaluation number allowed. The last row of each part shows the total success rate summed over all problems considered. By comparing those numbers, it can be observed that increasing Maxnfe generally improves the performance for each algorithm, except for MA+MDE’. For MA+MDE’, the total success rate decreases by 0.01 probably because the stochastic nature of meta-heuristics. Furthermore, the overall improvement for the three newly proposed algorithms is more significant than those three previous

algorithms: 1.23 to 2.54 versus 0.3. Therefore, the improvement of the three newly proposed hybrids over the three previous algorithms is even higher when higher Maxnfe values are used.

Table 3.6 Effect of increasing maximum number of function evaluations on success rate for selected problems

Pre-determined maximum number of evaluations							
Problem	Maxnfe	MDE’	MA+MDE’	MDE’+IH	MDE’+HJ	MDE’+IHS+HJ	PSO+MDE’+HJ
3	15,000	0.91	0.93	0.97	0.00	0.48	0.00
5	5,000	0.65	0.84	1.00	0.79	0.86	0.49
6	50,000	0.42	0.65	0.38	0.90	0.83	0.71
8	50,000	0.54	0.85	0.72	0.03	0.81	0.28
10	50,000	0.93	0.94	1.00	0.47	0.92	0.89
13	50,000	0.17	0.17	0.08	0.76	0.50	0.99
15	400,000	0.46	0.52	0.27	0.70	0.47	0.46
Total		4.08	4.90	4.42	3.65	4.87	3.82
Increased maximum number of evaluations							
3	80,000	0.91	1.00	1.00	0.95	1.00	0.98
5	10,000	0.76	0.82	1.00	0.82	0.96	0.47
6	100,000	0.45	0.55	0.44	0.89	0.97	0.72
8	100,000	0.51	0.81	0.64	0.94	1.00	1.00
10	100,000	0.92	0.94	1.00	0.96	0.97	0.92
13	150,000	0.22	0.19	0.06	0.85	0.62	1.00
15	1,000,000	0.54	0.58	0.36	0.78	0.58	0.49
Total		4.31	4.89	4.50	6.19	6.10	5.58

Figures 3.7 and 3.8 show the convergence profiles of all six algorithms for Problem 3 and 13, respectively, for increased Maxnfe. Given sufficient number of function evaluations, one of our newly proposed hybrids (MDE’+IHS+HJ) is also capable of achieving perfect accuracy as the two previous hybrids for Problem 3. The performance of other two new hybrids, MDE’+HJ and PSO+MDE’+HJ, have also improved (from 0% to 95% and 98%, respectively). For Problem 13, which is shown in Figure 3.8, each algorithm seems to

follow similar trends as in Figure 3.6. No significant change in search pattern has been observed for this problem, probably because of its difficulty.

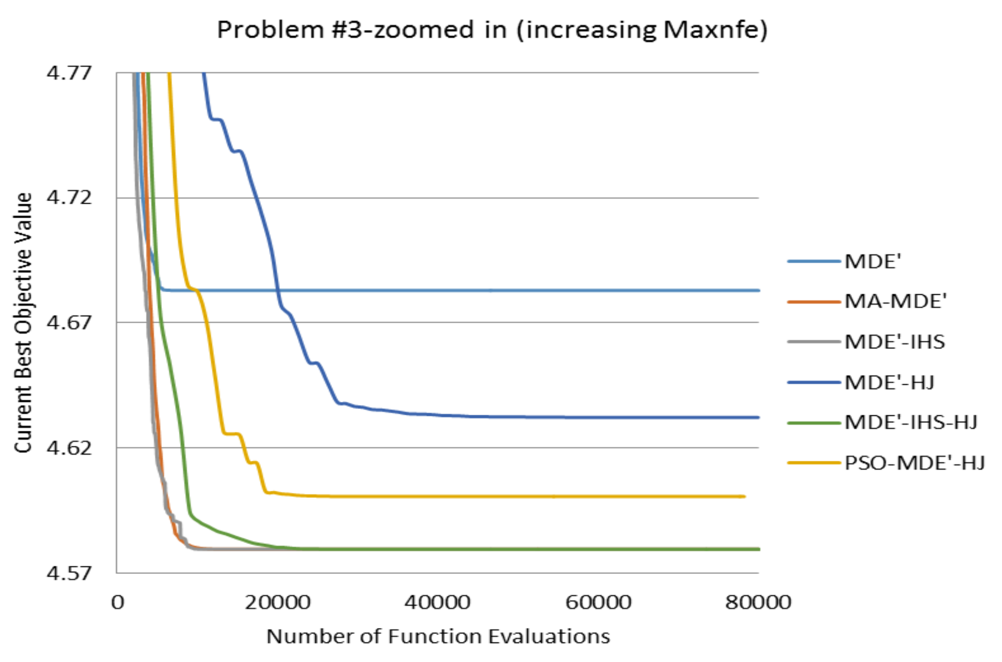


Figure 3.7 Average convergence rates of the best solutions of problem 3 over 100 runs Maxnfe=80000 (zoomed-in view)

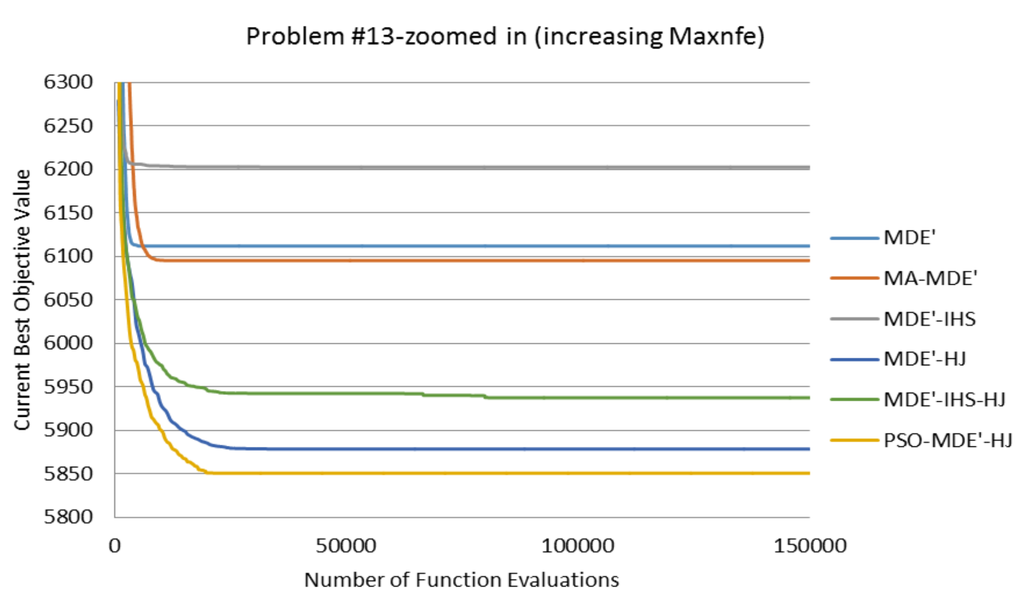


Figure 3.8 Average convergence rates of the best solutions of problem 13 over 100 runs Maxnfe=150000 (zoomed-in view)

3.4.2 Effect of adapting step size in HJ

Another parameter that can be tuned is the step size, λ. In all our tests above, λ is set to be adaptive throughout the searching process. At the beginning, it is set as 10% of the domain range and later gradually reduced since more exploitation of the neighborhood nearby is desirable. To show the difference between using adaptive step size and using fixed step size, tests were run on two HJ-based hybrids for fixed step size in which λ is constantly set as 10% of the domain range. Tables 3.7-3.9 record the detailed results in terms of success rate, number of function evaluations, and CPU time taken, respectively.

Table 3.7 Success rate with different step sizes for two HJ-based hybrids

Problem	Maxnfe'	MDE'+HJ		MDE'+IHS+HJ	
		Fixed step size	Adaptive step size	Fixed step size	Adaptive step size
1	15,000	1.00	1.00	0.95	0.96
2	100,000	0.76	0.74	0.97	0.94
3	80,000	0.97	0.95	1.00	1.00
4	5,000	0.99	0.99	0.96	0.95
5	10,000	0.77	0.82	0.95	0.96
6	100,000	0.91	0.89	0.92	0.97
7	1,797	1.00	0.91	0.98	0.99
8	100,000	0.83	0.94	1.00	1.00
9	5,495	1.00	1.00	1.00	1.00
10	100,000	0.97	0.96	0.93	0.97
11	1,000	1.00	1.00	0.98	1.00
12	14,000	1.00	1.00	1.00	1.00
13	150,000	0.85	0.85	0.56	0.62
14	10,000	1.00	1.00	1.00	1.00
15	1,000,000	0.70	0.78	0.62	0.58
16	10,000	1.00	1.00	1.00	1.00
17	200,000	0.99	0.98	1.00	1.00
18	50,000	1.00	1.00	1.00	1.00
Total		16.74	16.81	16.82	16.94

Table 3.8 Function evaluations with different step sizes for two HJ-based hybrids

Problem	Maxnfe	MDE'+HJ		MDE'+IHS+HJ	
		Fixed step size	Adaptive step size	Fixed step size	Adaptive step size
1	15,000	6,493	5,860	6,394	6,589
2	100,000	26,748	28,389	5,393	10,522
3	80,000	53,600	51,480	29,956	29,218
4	5,000	1,801	1,787	1,232	1,211
5	10,000	2,893	2,421	1,278	1,074
6	100,000	20,197	22,589	29,599	20,850
7	1,797	364	994	405	458
8	100,000	85,898	82,817	53,108	51,133
9	5,495	484	495	493	453
10	100,000	55,723	54,833	19,628	16,427
11	1,000	286	285	254	221
12	14,000	1,593	1,704	1,651	1,762
13	150,000	44,356	42,939	75,576	68,005
14	10,000	2,867	3,058	1,640	1,747
15	1,000,000	548,848	476,900	556,658	587,944
16	10,000	2,927	2,997	3,040	3,283
17	200,000	175,885	181,044	110,307	110,522
18	50,000	18,529	22,815	17,136	17,120
Total		1,049,493.45	983,406.54	913,746.95	928,539.28

From Table 3.7, it can be seen that adapting step size generally yields higher total success rate but the difference is not significant. Tables 3.8 and 3.9 show that adapting the step size does not necessarily increase the corresponding function evaluations needed and the computing time. Based on these results, it is recommended to use adaptive step size in HJ search method to achieve higher success rate.

Table 3.9 CPU time (in seconds) with different step sizes for two HJ-based hybrids

Problem #	Maxnfe	MDE'+HJ (s)		MDE'+IHS+HJ (s)	
		Fixed step size	Adaptive step size	Fixed step size	Adaptive step size
1	15,000	0.896694	0.577360	0.713549	0.744593
2	100,000	3.941365	4.034342	0.724625	1.470465
3	80,000	6.037083	5.467679	3.281481	4.090034
4	5,000	0.397023	0.276746	0.163957	0.162709
5	10,000	0.767057	0.484383	0.133693	0.149605
6	100,000	3.482566	3.181952	3.408466	3.005359
7	1,797	0.061620	0.171757	0.053664	0.052104
8	100,000	10.836297	8.200037	5.641932	5.791225
9	5,495	0.078625	0.070200	0.065832	0.068172
10	100,000	18.303129	15.623812	4.637286	3.422662
11	1,000	0.121837	0.091885	0.074724	0.066300
12	14,000	0.263330	0.215749	0.234782	0.257090
13	150,000	9.281903	8.258693	12.320647	11.684007
14	10,000	0.508719	0.482355	0.247730	0.258338
15	1,000,000	66.301361	43.07172	54.182579	50.578488
16	10,000	0.314498	0.269102	0.335090	0.362858
17	200,000	32.611853	32.428552	16.203356	14.056314
18	50,000	2.496016	2.345631	1.504318	1.434117
Total		156.700976	125.251955	103.927711	97.65444

3.5 CONCLUSION

In hybrid meta-heuristics, choosing an adequate combination of complementary algorithms can be very challenging. However, it is probably the key for achieving top performance in solving many complicated optimization problems. This Chapter has presented three new hybrid meta-heuristics, aiming to improve the two hybrids developed by Liao (2010), i.e., MA+MDE' and MDE'+IHS. The three new improved algorithms are:

(i) MDE'+HJ, which is a modification of MA+MDE' by replacing the original local search method, i.e. random walk with direction exploitation, with a more efficient direct search

method, i.e. the Hooke and Jeeves method;

(ii) MDE'+IHS+HJ, which is constructed by adding the Hooke and Jeeves to the original cooperative hybrid, i.e. MDE'+IHS; and

(iii) PSO+MDE'+HJ, which is a variation of MDE'+IHS+HJ by replacing IHS with PSO.

A comprehensive comparative study was carried out to show that the three new hybrids improves over the two previous hybrids in terms of average success rate while taking less average elapsed CPU time. Among these three hybrids, MDE'+IHS+HJ is the best one, better than the best previous hybrid by over 15% and better than the second best new hybrid, PSO+MDE'+HJ, by nearly 10%. It was also shown that for some problems the performance of those new hybrids can be further improved by increasing the maximal number of function evaluations. Given the good performance of the three new hybrid meta-heuristic algorithms on the 18 different engineering problems, they are used as a general solution procedure for solving the complicated, multiple variable inventory models provided in this research.

CHAPTER IV
ONE-FACTOR CS MODEL: CONTROLLABLE LEAD TIME

As mentioned in Sections 1.1.2 and 2.3.1, the lead time is an important system controlling factor. However, its effects on a CS inventory model are not clear. In this Chapter the one-factor CS model, which considers the effect of a controllable lead time to a buyer-vendor integrated inventory system under CS policy, is presented. The objective of this model is to jointly decide the optimal ordering size, number of shipments within each production cycle, and the lead time, that minimize the annual joint total expected cost (*JTEC*) of the system.

4.1 THE PROBLEM

The basic CS model is defined by Braglia and Zavanella (2003) as the CS-k case, where *k* stands for the number of delayed deliveries. The model considers a single-vendor, single-buyer integrated inventory system, under long term CS policy agreement, in which the vendor produces at a finite rate, and the outputs are shipped to the buyer in a number of equal-sized lots. The shipments are made without having to wait until the buyer's inventory reaches the reorder point, thus increasing buyer's inventory. Once the buyer's inventory is close to its capacity limitation, the rest of the shipments are delayed until the replenishment does not lead to an increase in the buyer's stock level.

The one-factor CS model is based on the CS-k case, but adding one factor, the lead time, as a decision variable. The length of lead time is reducible in this model with an extra investment. In this integrated system, all unit holding costs consist of a storage component

and a financial component. Table 4.1 indicates which party should bear the associated portion of the holding cost; specifically, the buyer bears only the storage part of the holding cost of the products stored in its warehouse. All other holding costs are considered to be borne by the vendor because of the CS case considerations. Moreover, the buyer's annual unit storage holding costs could be either greater or less than that of the vendor in the model developed and the annual unit storage holding cost in transit is considered to be greater than that of the vendors and/or the buyers. Meanwhile, the lead time is considered to be controllable under an extra investment which shall be borne by both of the trading parties under a long-term agreement.

Table 4.1 Responsibility of bearing the relevant portion of holding costs

		Position of Goods		
		Vender	In Transit	Buyer
Relevant costs	Vender	F[a], S[b]	F, S	F
	Buyer	0	0	S

a: F is the pure financial portion of the holding costs.
b: S is the pure storage portion of the holding costs.

The remainder of this Chapter is organized as follows. Section 4.2 defines parameters and assumptions of the model. A four-variable integrated EOQ/EPQ model is developed in Section 4.3. Section 4.4 presents a procedure to find the optimal solution of the model. Then, two numerical examples are used to illustrate the solution procedure in Section 4.5 and the impacts of some important parameters are discussed in Section 4.6. Conclusions are made in Section 4.7.

4.2 NOTATION AND ASSUMPTIONS

The *JETC* of the one-factor CS model is developed here by associating the costs of individual elements of the system relative to the physical location of the products. In order to develop the integrated model, the notations and assumptions are listed below. Note that the notations and assumptions used in this Chapter are also used in the following Chapters, so they are named as general definitions and general assumptions. New notations/assumptions and modified notations/assumptions will be given in the later Chapters as local definitions/assumptions, wherever they are needed.

Notation Summary:

Variables: n, q, L

General Parameters: $r,\ I_s, P, D$

Vendor related parameters: $A_v, H_v, h_v,\ h_v^f,\ h_v^s, I_v, p_v$

Buyer related parameters: $A_b, H_b, h_b,\ h_b^s, I_b$

Delivery related parameters: $H_d,\ h_d^s,\ h_d, a_i, b_i, c_i, L_i$

General Global Notations:

A_v: Vendor’s batch setup cost ($/setup),

A_b: Buyer’s ordering cost ($/order),

a_i: Minimum duration of the *i*th segment of lead time S_i (year),

b_i: Maximum duration of the *i*th segment of lead time S_i (year),

c_i: Crushing cost to reduce one time unit of the *i*th segment of lead time S_i

($/unit time),

D: Yearly demand rate at the buyers' level (units/year),

h: Annual unit holding cost ($/item/year),

H_v: Vendor's average annual inventory holding cost ($/year),

h_v: Unit annual holding cost for inventory at the vendor ($/item/year), $h_v = h_v^s + h_v^f$,

h_v^f : Financial components of h_v ($/item/year), $h_v^f = rp_v$,

h_v^s : Storage component of h_v ($/item/year),

H_b: Buyer's average annual inventory holding cost ($/year),

h_b: Unit annual holding cost for inventory at the buyer ($/item/year), $h_b = h_b^s + h_v^f$,

h_b^s : Storage component of h_b ($/item/year),

H_d: Average annual holding cost for inventory in transit ($/year),

h_d : Unit annual holding cost in transit ($/item/year), $h_d = h_d^s + h_v^f$,

h_d^s : Storage component of h_d ($/item/year),

r: Opportunity cost of capital (%/year),

I: Average inventory (units),

I_b: Buyer's average inventory (units),

I_s: Average system inventory (units),

I_v: Vendor's average inventory (units),

k: Number of delayed deliveries due to buyer's stock capacity,

L: Length of lead time (year) $L = \sum S_i$ (years),

L_i: Lead time where the ith component S_i was crushed to its minimum duration a_i (years),

n: Number of delivery operations per production batch,

P: Vendor's production rate (units/year),

p_v: Vendor's unit production cost ($/item),

q: Size of each delivery or shipped lot (batch size $Q = nq$),

$R(L)$: Lead time crushing cost per replenishment cycle ($/shipment),

S_i: ith segment of lead time (year), $a_i \le S_i \le b_i$.

General Assumptions:

The following assumptions are necessary to model this problem:

(1) The annual average demand of the end product is considered to be stable and is assumed a fixed value D.

(2) The yearly production rate P is finite, and $P > D$ to avoid any shortage.

(3) The products are shipped to the buyer at n batches each of equal size q.

(4) The extra investment incurred to reduce the lead-time is borne by the system under a long-term agreement between the vendor and the buyer.

(5) The inventory is continuously reviewed and shortage is not allowed.

(6) The inventory is continuously reviewed.

4.3 MODEL FORMULATION

In this system, the vendor produces within each cycle n batches, each of size q, with a fixed setup cost A_v, at a finite production rate P. In order to avoid stock out, P is assumed to be greater than the buyer's demand rate D (i.e., $P > D$). Under CS case, the vendor is responsible to maintain a reasonable stock level in the buyer's warehouse, thus each batch is delivered before the buyer's inventory decreases to the reorder point within one production cycle. Each shipment incurs an ordering cost, A_b, to the buyer and the buyer builds up stocks while the vendor maintains the minimum amount of inventory [Figure 4.1(a)]. In addition, the last k number of the n shipments is delayed so that the buyer's inventory does not go beyond its capacity limitation. Therefore, the vendor's inventory also increases to a certain level [Figure 4.1(b)]. The number of delayed shipments k has an implicit upper bound n-1 because the first shipment cannot be delayed, or there will be shortage, which is not allowed. Furthermore, according to Table 4.1, the unit holding cost h is split into two mutually independent portions: a pure financial portion h^f and a pure storage portion, h^s. Therefore, the three unit holding costs are given as: $h_b = h_b^s + h_b^f$, $h_v = h_v^s + h_v^f$, and $h_d = h_d^s + h_d^f$. The pure storage portion h^s has nothing to do with the opportunity cost, r, while the pure financial portion h^f is r times the money tied up by investing on the product, that is, $h_v^f = rp_v$, where p_v is the unit production cost of the product for the vendor. In consignment stock setting, all the opportunity costs are borne by the vendor. The three unit holding costs can, thus, be rewritten as: $h_v = h_v^s + h_v^f$, $h_d = h_d^s + h_v^f$, and $h_b = h_b^s + h_v^f$.

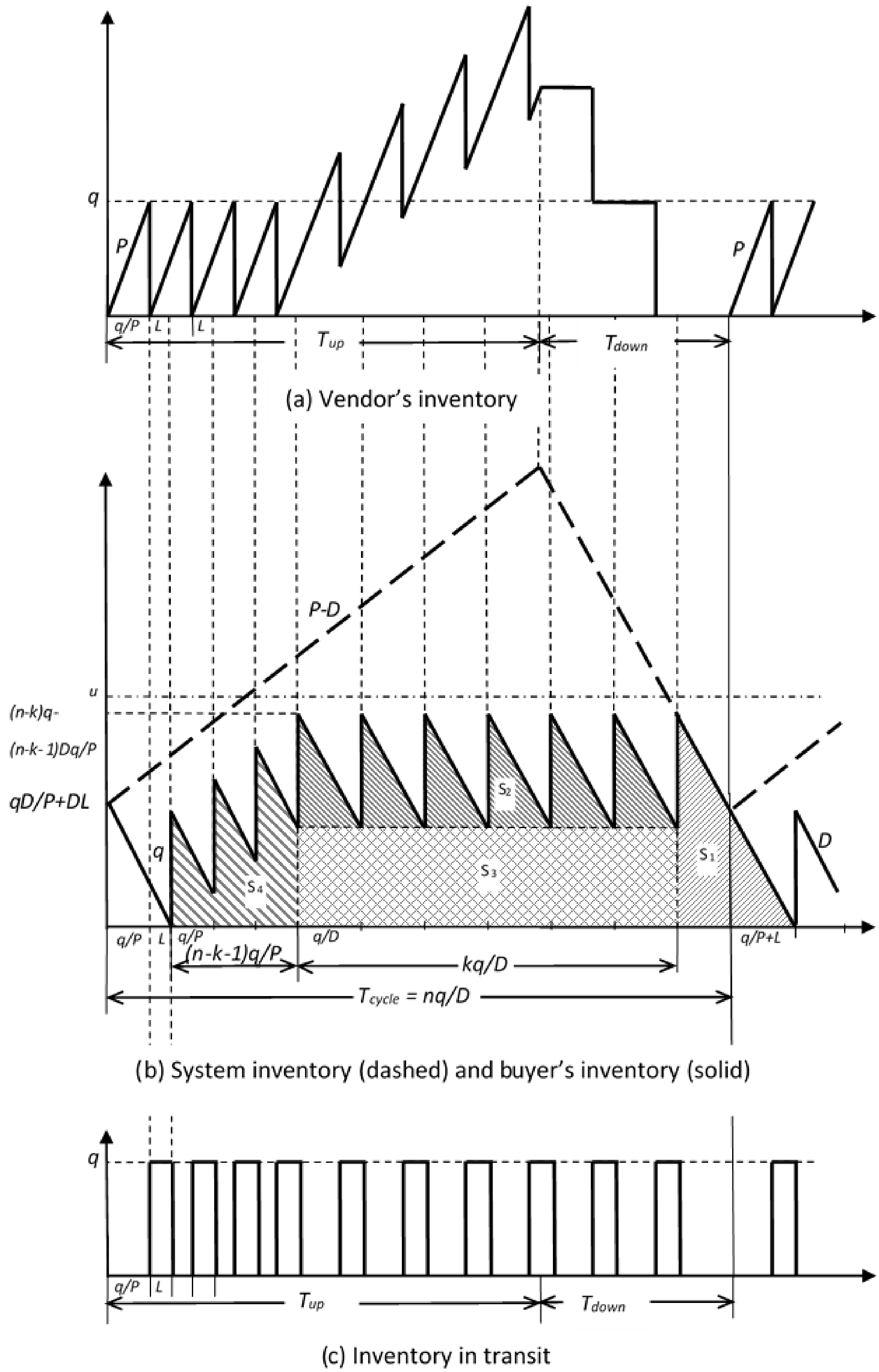


Figure 4.1 Composition of the inventory in the system

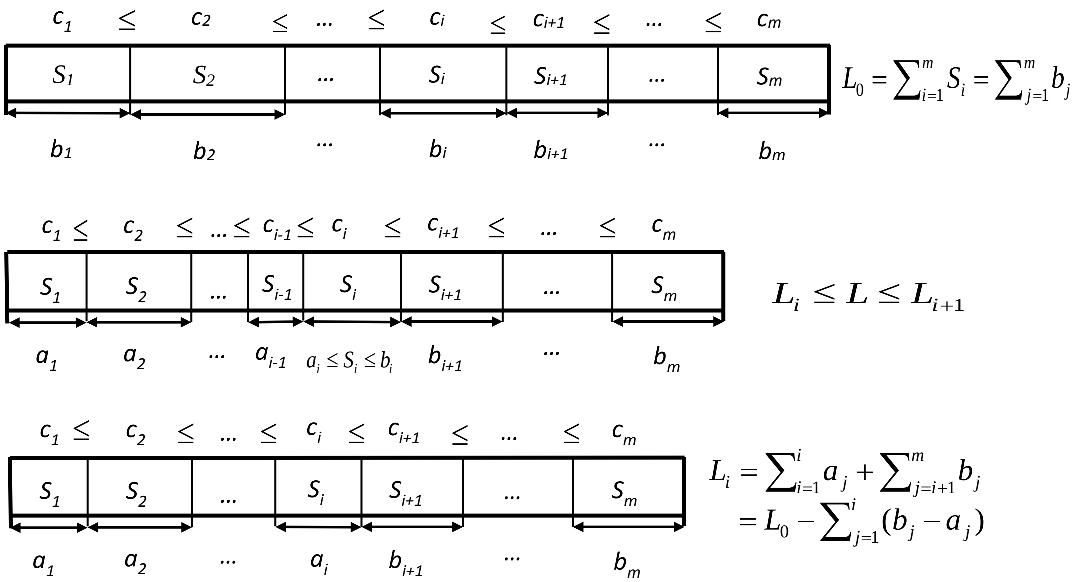


Figure 4.2 Composition of Lead time

Figure 4.2 illustrates the composition of the lead time L, before and after crushing. The lead time L is divided into m mutually independent components and labeled as S_i in such a way that their respective crushing cost per time unit, c_i, follows a non-decreasing order $c_1 \le c_2 \le c_3 \le \ldots \le c_m$ for $i = 1, 2, \ldots, m$. Thus, $L = \sum_{i=1}^{m} S_i$ and $a_i \le S_i \le b_i, \forall i$, where a_i and b_i are the minimum and maximum (i.e., *normal*) duration of S_i, respectively. The lead time L is crushed one time-unit at a time (for example, one day at a time) starting with the cheapest crushing cost, c_1.

Let L_0 denote the normal lead time before crushing, that is, $L_0 = \sum_{j=1}^{m} b_j$ which is the maximum normal duration. Also, let L_i denote the lead time when the ith component was crushed to its minimum duration. i.e., $L_i = \sum_{j=1}^{i} a_j + \sum_{j=i+1}^{m} b_j = \sum_{j=1}^{m} b_j - \sum_{j=1}^{i} (b_j - a_j) = L_0 - \sum_{j=1}^{i} (bj$

$-a_j)$ $i=1,2,...,m$. Thus, in order to reduce the lead time, the investment should be at least c_1; that is, with an investment c_1, the first lead time segment S_1 can be reduced for one time unit (say, one day). When the reduced lead time is less than $b_1 - a_1$, the crushing cost can be calculated as $c_1(b_1 - S_1)$. The second lead time segment S_2 will not be reduced unless the investment is greater than $c_1(b_1 - a_1)$ because sticking to the segment S_1 with lower investment cost will be economically beneficial. When the *i*th segment of lead time is crushed, then L should be somewhere between L_{i-1} and L_i. The lead time crushing cost per replenishment cycle can thus be written as $R(L) = c_i(L_{i-1} - L) \quad + \sum_{j=1}^{i-1} c_j(b_j - a_j)$, $i = 1,2,...,m$ for $L \in (L_i, L_{i-1})$.

4.3.1 Finding the joint total expected cost *JTEC*

The *JTEC* of the system is the sum of vendor's annual production setup cost, buyer's annual ordering cost, annual lead time crushing cost, and annual system holding cost, as shown in Figure 4.3. The annual system holding cost is, in turn, the sum of the annual holding cost incurred in the vendor's warehouse, the annual holding cost incurred in the buyer's warehouse, and the annual holding cost in transit. The vendor's annual production setup cost can be written as $A_v D / nq$. The buyer's yearly ordering cost is given by $A_b D / q$. Meanwhile, the yearly lead time crushing cost is $R(L)D/q$, where, the lead time crushing cost per replenishment cycle is $R(L) = c_i(L_{i-1} - L) \; + \sum_{j=1}^{i-1} c_j(b_j - a_j)$.

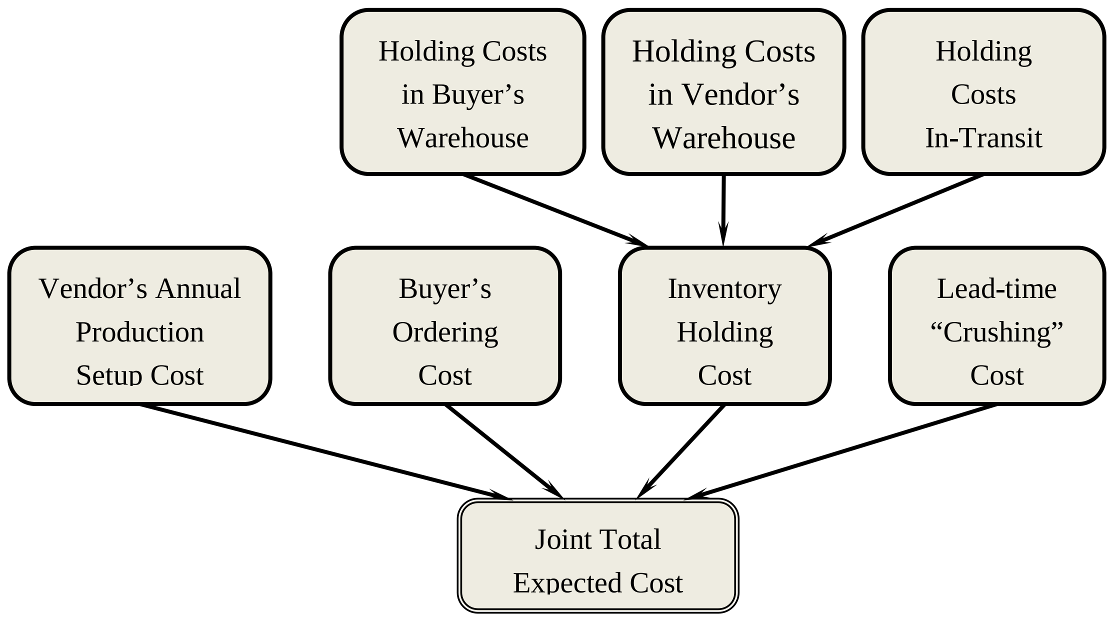


Figure 4.3 Composition of the *JTEC* of the one-factor CS model

To obtain the annual system holding cost, it is necessary to calculate the average inventory for the vendor, buyer, and "in transit". Figure 4.1 illustrates the pattern of the inventory of all parties in the integrated system within one production cycle. The average system inventory I_s can be written as

$$I_s = \frac{nq(P-D)}{2P} + \frac{qD}{P} + DL\,.$$

The buyer's average inventory can be derived by the area *S* [the shaded area in Figure 4.1(b)] divided by the cycle time T_{cycle}, where, the area *S* can be written as (see Appendix A-1)

$$S = q^2\left(\frac{n^2-k^2-k}{2D} + \frac{n-n^2+k^2+k}{2P}\right). \tag{4.1}$$

Thus, the buyer's average inventory, I_b, is given by

$$I_b = S/T_{cycle} = \frac{qD}{2P} + nq\frac{(P-D)}{2P} - q\frac{P-D}{nP}\frac{(k+1)k}{2}\,. \tag{4.2}$$

The average inventory in transit is the sum of *n* equal area under the solid line in Figure

4.1(c) divided by the cycle time T_{cycle}, that is,

$$I_d = \frac{nqL}{nq/D} = DL\,. \tag{4.3}$$

The vendor's average inventory can be derived by subtracting the buyer's average inventory and the average inventory in transit from the system's average inventory and it is thus given as

$$I_v = I_s - I_b - I_d == \frac{qD}{2P} + q\frac{P-D}{nP}\frac{(k+1)k}{2}\,. \tag{4.4}$$

Finally, the system annual holding cost can be written as

$$H = H_v + H_d + H_b = (h_v^s + h_v^f)q\frac{D}{2P} + (h_b^s + h_v^f)q\frac{nP-(n-1)D}{2P}$$
$$+ (h_v^s - h_b^s)q\frac{(P-D)(k+1)k}{2nP} + (h_d^s + h_v^f)DL\,. \tag{4.5}$$

Hence, the annual joint total expected cost $JTEC(q,n,k,L)$ for given $L \in (L_i, L_{i-1})$ can be written as

$$JTEC(q,n,k,L) = (A_v + nA_b)\frac{D}{nq} + (h_v^s + h_v^f)q\frac{D}{2P} + (h_b^s + h_v^f)q\frac{nP-(n-1)D}{2P}$$
$$+ (h_v^s - h_b^s)q\frac{(P-D)(k+1)k}{2nP} + (h_d^s + h_v^f)DL + \frac{D}{q}R(L)\,. \tag{4.6}$$

Equation (4.6) is a non-linear integer problem with four decision variables *n*, *q*, *L*, and *k*. It is a modified version of Huang and Chen's (2009) general model to take controllable lead time into consideration.

In the literature, much attention has been paid to the case where the buyer's unit holding cost is greater than that of the vendor's, that is, $h_b^s > h_v^s$. However, some

researchers, such as Zanoni and Grubbstrom (2004), Hill and Omar (2006), and Huang and Chen (2009), also pointed out that the unit storage cost may either increase or decrease down the supply chain, depending on different circumstances. In this research, we divide the problem into two distinguished cases: Case 1, $h_v^s > h_b^s$, and Case 2, $h_v^s < h_b^s$. Based upon these two cases, problem described by Equation (4.6) can be divided into the following two problems:

One-factor CS model 1: No delay. When $h_v^s > h_b^s$, the optimum occurs when $k = 0$. Thus, the problem of minimizing Equation (4.6) can be rewritten as

$$Min: \quad JTEC(q,n,L) = (A_v + nA_b)\frac{D}{nq} + (h_v^s + h_v^f)q\frac{D}{2P} + (h_b^s + h_v^f)q\frac{nP-(n-1)D}{2P}$$

$$+\frac{D}{q}R(L) + (h_d^s + h_v^f)DL. \tag{4.7}$$

One-factor CS model 2: Maximum delay. When $h_v^s < h_b^s$, the optimums occur when $k = n-1$. Thus, the problem becomes

$$Min: \quad JTEC(q,n,L) = (A_v + nA_b)\frac{D}{nq} + (h_v^s + h_v^f)q\frac{D}{2P} + (h_b^s + h_v^f)q\frac{nP-(n-1)D}{2P}$$

$$+(h_v^s - h_b^s)q\frac{(P-D)(n-1)}{2P} + \frac{D}{q}R(L) + (h_d^s + h_v^f)DL. \tag{4.8}$$

When the controllable lead time is removed from the above two models, they also reduce to Huang and Chen's (2009) two reduced models. In Equations (4.7) and (4.8), the lead time is a discrete variable with limited integer choice of values. Therefore, it is feasible to fix the value of the L and look for the optimal solutions for the other two variables n and

q. It would be ideal if one can prove that the objective functions are strictly convex over n and q. However, based on the Hessian Matrix in Appendix A-2.1 and A-2.2, the difference between the multiplication of the two main diagonal elements and that of the two off-diagonal elements is not guaranteed to be greater than zero. Therefore, it is difficult to prove the convexity of the objective function on both n and q over the entire domain. It is, thus, also difficult to develop a solution procedure based on the property of a convex function. In Section 4.4, we shall discuss some properties of the two Equations and based upon which, we shall present a solution procedure that does not require the objective function to be strictly convex simultaneously in both n and q.

4.4 SOLUTION PROCEDURE

In order to develop a solution procedure for the two reduced CS models, some properties of the objective functions are necessary and are discussed as below.

Property 4.1: *For given n and q, the joint total expected cost* $JTEC(L)$ *is linear on L.*

Proof: Fixing the value of q and n and taking the first derivatives of Equations (4.7) and (4.8) will lead to

$$\frac{\partial JTEC(L)}{\partial L} = \left(h_d^s + h_v^f - \frac{c_i}{q} \right) D. \tag{4.9}$$

This result is a constant, the value of which can be positive or negative or even zero depending on the difference between the unit annual holding cost in transit, ($h_d = h_d^s + h_v^f$), and the ratio c_i/q. If its value is positive, then the *JTEC* is linear increase on L. If it is

negative, then the *JTEC* is linear decrease on *L*. If it is zero, then the *JTEC* is flat on *L*. Under each scenario, the *JTEC* is linear on *L*, which completes the proof of Property 1. □

As the result, for a given value of *n* and *q*, the minimum *JTEC* always occur at one of the end points of the segments of the lead time. This property further reduced the computational burden to search over the dimension of lead time since the number of segments of lead time is usually much less than the possible number of lead time. Next, we focus on the two remaining variables *n* and *q* and study the convexity of the objective function *JTEC(n, q)* on them. Unfortunately, like other researchers, we cannot prove that the *joint total expected cost* function is convex over the entire feasible range of *n* and *q*. However, we do not need this condition to have a global optimum of the developed model. In fact, it is sufficient to show that the objective function *JTEC(n, q)* has only one stationary point in the whole feasible range and the function is convex at that point. In other words, if we can prove that there is only one local optimum over the entire feasible range of the decision variables, this local optimum would also be the global optimum.

Property 4.2: *For given* $L \in (L_i, L_{i-1})$, *the joint total expected cost* $JTEC(q,n)$ *is strictly convex at the point* (*n**, *q**), *where n* and q* are the solutions of* $\partial JTEC(n,q)/\partial n = 0$ *and* $\partial JTEC(n,q)/\partial q = 0$.

See Appendix A-2 for the proof of Property 4.2. This property only guarantees the point (*n**, *q**) to be a local minimum, but not necessarily the global minimum. However, considering the fact that there is only one stationery point within the entire feasible region

of *n* and *q* (see Appendix A-2), this sole point has to be the global minimum as well.

Now, we are able to locate the global optimal solution of *n* and *q* when they are assumed to be real values. However, both of them are integer variable in reality, and the closest integer point to the global solution point of a real variable may not be the global solution point to an integer variable. Therefore, it is necessary to develop a method to locate the global integer optima starting from the known global optimal real solution.

Property 4.3: *For fixed* $L \in (L_i, L_{i-1})$ *and n, the JTEC(q) is strictly convex in q, and the solution of q can be written as*

$$q^* = \sqrt{\frac{2PD[A_v + nA_b + nR(L)]}{n\{(h_v^s + h_v^f)D + (h_b^s + h_v^f)[np - (n-1)D]\}}}\,, \tag{4.10}$$

for case 1, and

$$q^* = \sqrt{\frac{2PD[A_v + nA_b + nR(L)]}{n\{(h_v^s[(n-1)P - (n-2)D] + h_v^f[np - (n-2)D] + h_b^s P\}}}\,, \tag{4.11}$$

for case 2.

See Appendix A-3 for the proof of Property 4.3.

Property 4.4: *For fixed* $L \in (L_i, L_{i-1})$ *and q, the JTEC(n) is strictly convex in n, and the solution of n can be written as*

$$n^* = \frac{1}{q}\sqrt{\frac{2A_v DP}{(h_b^s + h_v^f)(P - D)}}\,, \tag{4.12}$$

for case 1, and

$$n^* = \frac{1}{q}\sqrt{\frac{2A_v DP}{(h_v^s + h_v^f)(P - D)}}\,, \tag{4.13}$$

for case 2.

See Appendix A-4 for proof of Property 4.4.

Based on Properties 4.2, 4.3, and 4.4, we are now able to develop an iterative searching algorithm to find the integer optimal solution point (n, q) from the real global optimal solution point (n^*, q^*). This searching algorithm is embedded into the following two full algorithms to jointly determine the optimal integer solution of n, q, and L, for the two cases:

Algorithm 4.1: (for Case 1)

Step 1: Let $i = 0$.

Step 2: Let $L = L_i$.

Step 3: Use Equation (A-2.7) and (A-2.8) to compute $n_i^{real\_opt}$ and $q_i^{real\_opt}$.

Step 4: Use Equation (4.7) to compute a start point n_i^{start} and q_i^{start} which satisfies

$$JTEC_i(L, n_i^{start}, q_i^{start}) = Min\{JTEC(L, \lceil n_i^{real\_opt} \rceil, \lceil q_i^{real\_opt} \rceil), JTEC(L, \lfloor n_i^{real\_opt} \rfloor, \lceil q_i^{real\_opt} \rceil),$$

$$JTEC(L, \lceil n_i^{real\_opt} \rceil, \lfloor q_i^{real\_opt} \rfloor), JTEC(L, \lfloor n_i^{real\_opt} \rfloor, \lfloor q_i^{real\_opt} \rfloor)\}.$$

Step 5: Starting from n_i^{start}, calculate $q_i^{real\_trail}$ using Equation (4.10).

Step 6: Use Equation (4.7) to compute q_i^{trail} that satisfies

$$JTEC_i(L, n_i^{start}, q_i^{trail}) = Min\{JTEC(L, n_i^{start}, \lceil q_i^{real\_trail} \rceil), JTEC(L, n_i^{start}, \lfloor q_i^{real\_trail} \rfloor)\}.$$

Step 7: If $JTEC_i(L, n_i^{start}, q_i^{trail}) \le JTEC_i(L, n_i^{start}, q_i^{start})$, let $q_i^{start} = q_i^{trail}$, otherwise, let $q_i^{start} = q_i^{opt}$.

Step 8: Starting from q_i^{start}, calculate $n_i^{real\_trail}$ using Equation (4.12).

Step 9: Use Equation (4.7) to compute n_i^{trail} that satisfies

$$JTEC_i(L, n_i^{trail}, q_i^{start}) = Min\{JTEC(L, \lceil n_i^{real\_trail} \rceil, q_i^{start}), JTEC(L, \lfloor n_i^{real\_trail} \rfloor, q_i^{start})\}.$$

Step 10: If $n_i^{trail} = n_i^{start}$, let $q_i^{opt} = q_i^{start}$, and $n_i^{opt} = n_i^{start}$, go to *Step 12*.

Step 11: Let $n_i^{start} = n_i^{trail}$, go to *Step 5*.

Step 12: If $i \le m$, where m is the number of segments of lead time, let $i = i + 1$, go to *Step* 2.

Step 13: The optimum solutions are $q*$, $n*$ and $L*$ such that

$$JTEC(L*, n*, q*) = Min\{JTEC_i(L, n_i^{opt}, q_i^{opt}), i = 1,..., m\}.$$

Algorithm 4.2: (for Case 2)

Step 1: Let $i = 0$.

Step 2: Let $L = L_i$.

Step 3: Use Equation (A-2.17) and (A-2.18) to compute $n_i^{real\_opt}$ and $q_i^{real\_opt}$.

Step 4: Use Equation (4.8) to compute a start point n_i^{start} and q_i^{start} which satisfies

$$JTEC_i(L, n_i^{start}, q_i^{start}) = Min\{JTEC(L, \lceil n_i^{real\_opt} \rceil, \lceil q_i^{real\_opt} \rceil), JTEC(L, \lfloor n_i^{real\_opt} \rfloor, \lceil q_i^{real\_opt} \rceil),$$

$$JTEC(L, \lceil n_i^{real\_opt} \rceil, \lfloor q_i^{real\_opt} \rfloor), JTEC(L, \lfloor n_i^{real\_opt} \rfloor, \lfloor q_i^{real\_opt} \rfloor)\}.$$

Step 5: Starting from n_i^{start}, calculate $q_i^{real\_trail}$ using Equation (4.11).

Step 6: Use Equation (4.8) to compute q_i^{trail} that satisfies

$$JTEC_i(L, n_i^{start}, q_i^{trail}) = Min\left\{JTEC\left(L, n_i^{start}, \lceil q_i^{real\_trail} \rceil\right), JTEC\left(L, n_i^{start}, \lfloor q_i^{real\_trail} \rfloor\right)\right\}.$$

Step 7: If $JTEC_i(L, n_i^{start}, q_i^{trail}) \le JTEC_i(L, n_i^{start}, q_i^{start})$, let $q_i^{start} = q_i^{trail}$, otherwise, let $q_i^{start} = q_i^{opt}$.

Step 8: Let $q_i^{start} = q_i^{trail}$, starting from q_i^{start}, calculate $n_i^{real\_trail}$ using Equation (4.13).

Step 9: Use Equation (4.8) to compute n_i^{trail} that satisfies

$$JTEC_i(L, n_i^{trail}, q_i^{start}) = Min\{JTEC(L, \lceil n_i^{real\_trail} \rceil, q_i^{start}), JTEC(L, \lfloor n_i^{real\_trail} \rfloor, q_i^{start})\}.$$

Step 10: If $n_i^{trail} = n_i^{start}$, let $q_i^{opt} = q_i^{start}$, and $n_i^{opt} = n_i^{start}$, go to *Step 12*.

Step 11: Let $n_i^{start} = n_i^{trail}$, go to *Step 5*.

Step 12: If $i \le m$, where *m* is the number of segments of lead time, let $i = i + 1$, go to *Step* 2.

Step 13: The optimum solutions are *q**, *n** and *L** such that

$$JTEC(L^*, n^*, q^*) = Min\{JTEC_i(L, n_i^{opt}, q_i^{opt}), i = 1, \ldots, m\}.$$

The algorithm first finds the global optima with real values. Then, amongst the four integer points closest to the real global optimal point, it chooses the point that leads to the minimum *JTEC* as the starting point and uses Properties 4.3 and 4.4 to conduct an alternative and iterative search over the two dimensions of *n* and *q*. The property that there is only one stationary point will guarantee that the search converges to the global optima of integer value. The reason is that, compared to any other location, the global optima will improve the objective function at least at one of the two dimensions of *n* and *q*. The searching will not stop until it reaches the final global optimal point.

4.5 COMPUTATIONAL RESULTS

Next, we use two numerical examples to illustrate the solution procedures of the two algorithms given in Section 4.4.

Example 4.1: Case 1 ($h_v^s > h_b^s$)

For the convenience of comparing with other researchers, the following data used for this example is taken from Braglia and Zavanella (2003) and Huang and Chen (2009): $A_v = \$400$ /setup, $A_b = \$25$ /order, $D = 1000$ units/year, $P = 3200$ units/year, $r = 10\%$, $p_v = \$20$ /item, $h_v^s = \$3$ /item/year, and $h_b^s = \$1.50$ /item/year. The lead time unit annual

holding cost is considered to be higher than both h_v^s and h_b^s with a value $h_d^s = \$4$/item/year. In addition, the lead time data are borrowed from Goyal (1988), Ouyang, *et al.* (1999), Pan and Yang (2002). The duration of each lead time component and associated unit crushing costs are shown on Table 4.2.

Table 4.2 Lead time data for Example 4.1

Lead time component i	Normal duration b_i (years)	Minimum duration a_i (years)	Unit crushing cost c_i (\$/year)
1	20/365 = 0.05479	6/365 = 0.01644	0.10(365) = 36.5
2	20/365 = 0.05479	6/365 = 0.01644	1.20(365) = 438
3	16/365 = 0.04384	9/365 = 0.02466	5.00(365) = 1,825

The controllable lead time has three segments, each with a maximum duration b_i and a minimum duration a_i. Therefore, the four end points of these lead time segments are: $L_0 = 20 + 20 + 16 = 56$ days = 0.1534 years, $L_1 = 6 + 20 + 16 = 42$ days = 0.1151 years, $L_2 = 6 + 6 + 16 = 28$ days = 0.0767 years, and $L_3 = 6 + 6 + 9 = 21$ days = 0.0575 years. The associated lead time crushing cost $R(L_i)$ are: $R(L_0) = \$0$, $R(L_1) = 36.5(0.05479 - 0.01644)$ $= \$1.40$, $R(L_2) = \$18.20$, and $R(L_3) = \$53.20$.

Table 4.3 illustrates the computation results of Algorithm 4.1. There are four rows in Table 4.3. Each row shows the computation results corresponding to one of the end points of the controllable lead time. It can be seen that the third row shows the optimal solution. The integer optimal solution for Example 4.1 is thus given by: $n^* = 3$, $q^* = 181$, and L^* = 28 days = 0.0767 years. The associated annual joint total expected cost is $JTEC^* = \$2{,}327.50$/year. Compared to the first row, which is the *JTEC* that the integrated

system will expend without investing on reducing the lead time, the saving is 2672.78 - 2327.50 = $345.28/year, which account for 345.28 *100 % / 2327.5 = 14.84 % of the annual total cost.

The *JTEC*s are higher in the tenth column than in the eighth column, but they are the highest in the ninth column. This is natural since the integer optimal solutions cannot outperform the real optimal solutions. However, they do yield better results than the start points. A starting point is one of the four closest corner points next to the real global optimal point and is the one that yields the minimum *JTEC*. The fact that the tenth column outperforms the ninth column illustrates how the integer global optima may not be very close the real global optima, which justifies the needs to develop a method to determine the integer solution based upon the real solution.

Table 4.3 Computational Summary for Example 4.1

L (days)	n			q			*JTEC* ($/year)		
	real opt	start	integer opt	real opt	start	integer opt	real opt	start	integer opt
56[a]	4.20	4	**4[a]**	137.20	138	**143[a]**	2672.43	2673.75	**2672.78[a]**
42	4.09	4	4	140.99	141	143	2452.35	2452.69	2452.44
28[b]	3.20	3	**3[b]**	180.35	181	**189[b]**	2326.78	2329.28	**2327.50[b]**
21	2.38	2	2	242.65	243	273	2377.19	2397.51	2383.74

a: The optimal solution of the modified Huang and Chen's model considering lead time.

b: The optimal solution of the reduced model 4.1.

Example 4.2: Case 2 ($h_v^s < h_b^s$)

For consistency, the data of Example 4.2 are also adapted from Braglia and Zavanella (2003) and Huang and Chen (2009). Specifically, all the data used in Example 4.2 are the

same as in Example 4.1 except for the following: $p_v = \$15$/item, $h_v^s = \$2.50$/item/year, and $h_b^s = \$3$/item/year. Table 4.4 shows the computation results of the Algorithm 4.2. The optimal results of Algorithm 4.2 occur at $n^* = 3$, $q^* = 170$, $L^* = 28$ days = 0.0767 years, and the corresponding *JTEC* = $2,416.60/year. Again, compared with the base model where the lead time is not reduced, the Example 4.2 also favors the choice to invest on reducing the lead time, and the saving is 2714.67 – 2416.6 = $298.07/year, which is approximately 12.33% of the annual total inventory cost.

Table 4.4 Computational Summary for Example 4.2

L (days)	*n*			*Q*			*JTEC*($/year)		
	real opt	start	integer opt	real opt	start	integer opt	real opt	start	integer opt
56[a]	4.18	4	**4[a]**	129.10	130	**134[a]**	2714.37	2715.37	**2714.67[a]**
42	4.07	4	4	132.67	133	134	2514.11	2514.25	2514.16
28[b]	3.18	3	**3[b]**	169.71	181	**170[b]**	2414.27	2329.28	**2416.60[b]**
21	2.36	2	2	228.33	229	256	2484.66	2504.54	2491.16

a: The minimum *JTEC* of modified Huang and Chen's model considering lead time.

b: The minimum *JTEC* of the model.

Under what condition will it be beneficial to invest on reducing lead time is an interesting problem. Although both the numerical examples are in favor of doing the investment, whether it is wise to do so is actually highly parameter dependent. It can be seen from Equation (4.9) that whether the *JTEC* is increasing or decreasing on the length of lead time is dependent on the difference of $\Delta_i = h_d^s + h_v^f - \frac{c_i}{q}$. When Δ_i is greater than zero, *JTEC* will increase on *L*, thus the optimal solution favors a shorter lead time. Note that the parameter c_i have *m* values and each is associated with a q_i value. We could use the value of

the two examples to verify the above statement. In Example 1, we have $h_d^s + h_v^f = \$6$/year, $c_0 = 0$, $c_1 = \$36.5$/year, $c_2 = \$438$/year, $c_3 = \$1,825$/year; $q_0 = 143$, $q_1 = 143$, $q_2 = 189$, and $q_3 = 273$. Hence, the differences are: $\Delta_0 = 6 - 0 = 6$, $\Delta_1 = 6 - 36.5/143 = 5.74$, $\Delta_2 = 6 - 438/189 = 3.68$, and $\Delta_3 = 6 - 1825/273 = -0.68$. It follows that the *JTEC* keeps decreasing as the lead time is crushed to the minimum duration of the second segment, that is, the optimal lead time is $L^* = L_2 = 0.0767$ years = 28 days. This verified the computation results. Using the data of Example 4.2, we get the same result of lead time. It is, thus, convenient to make a decision of when to invest on reducing lead time.

4.6 SENSITIVITY ANALYSIS

The developed model has several parameters. A change in the value of one or some of the parameters may sometimes change the solution. It is, thus, very important to study what the effects are that the change in the parameters may have on the model. It is also necessary to know the relationship between these parameters and the objective function, the relationship among different parameters, and to determine which parameters or relationships are more influential on the current model. Primary study shows that the relationship between any single parameter and our *JTEC* is simple.

Now, we focus on the effect of some combinations of the parameters on our solutions. The combinations includes: the ratio of the average annual demand rate over the annual production rate (D/P), the ratio of buyer's ordering cost over vendor's batch setup cost (A_b / A_v), and the ratio of the buyer's unit annual storage component of holding cost over

the vendor's unit annual storage component of holding cost (h_b^s / h_v^s). Because we considered a controllable lead time in our model, it is important to study the effect of the unit crushing cost c_i and the unit holding cost $h_d^s + h_v^f$ on the solutions. As far as the two parts of this unit holding cost are concerned, the role of h_d^s is the same as that of h_v^f. Thus, studying the effect of any of them is equivalent to studying the other. Also, since $h_v^f = i(p_v)$, where i is the opportunity cost which does not fluctuate. We can, therefore, focus on the study of the effects of c_i and p_v on the solutions. We conduct a sensitivity analysis on these effects as follow.

4.6.1 Effect of *D/P* on *JTEC*, *n*, and *q*

The ratio between demand rate and production rate turns out to be the most influential factor among all those parameters as can be seen from Figure 4.4 (for Example 4.1) and Figure 4.5 (for Example 4.2), which illustrate the effect of D/P on the optimal solution *n* and *q*, and the minimum *JTEC*, that almost all of these results changed as the ratio *D/P* increases. Specifically, note that the total cost *JTEC* increases relatively rapidly as *D* approaches *P*, and after *D* nears *P* (the ratio of *D/P* greater than 90%), the total cost decreases. Also, the optimal number of deliveries in each cycle *n* is not sensitive to *D/P* when the ratio is low. It becomes highly sensitive to *D/P* when the ratio is greater than 80%. It is interesting that in Example 4.1 the batch size is not affected by *D/P*, whereas in Example 4.2 the ratio does affect batch size.

4.6.2 Effect of A_b/A_v on *JTEC*, *n*, and *q*

Figures 4.6 and 4.7 illustrate the effect of A_b / A_v on the objective function and model variables. We can see that both *JTEC* and *q* have a nearly positive linear relationship with A_b / A_v whereas *n*, although not changed much, has a negative relationship with A_b / A_v. It is also noticeable that the effect of A_b / A_v is consistent for both Example 4.1 and Example 4.2 and, compared with *D/P*, the ratio A_b / A_v is more influential on the shipping size *q* but less

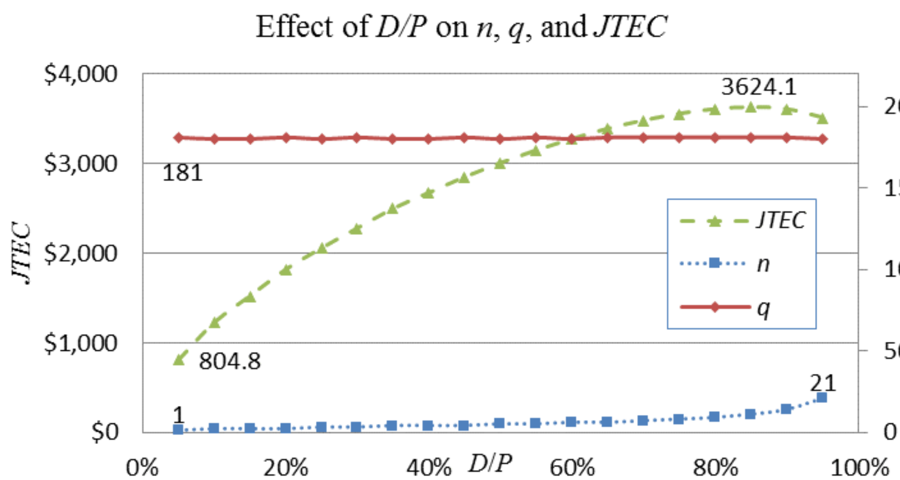


Figure 4.4 Effect of change in *D/P* on optimal solutions of Example 4.1

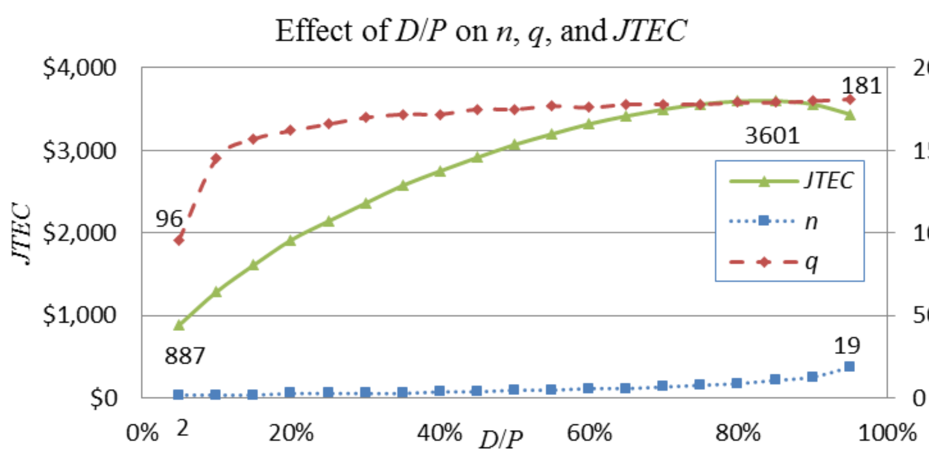


Figure 4.5 Effect of change in *D/P* on optimal solutions of Example 4.2

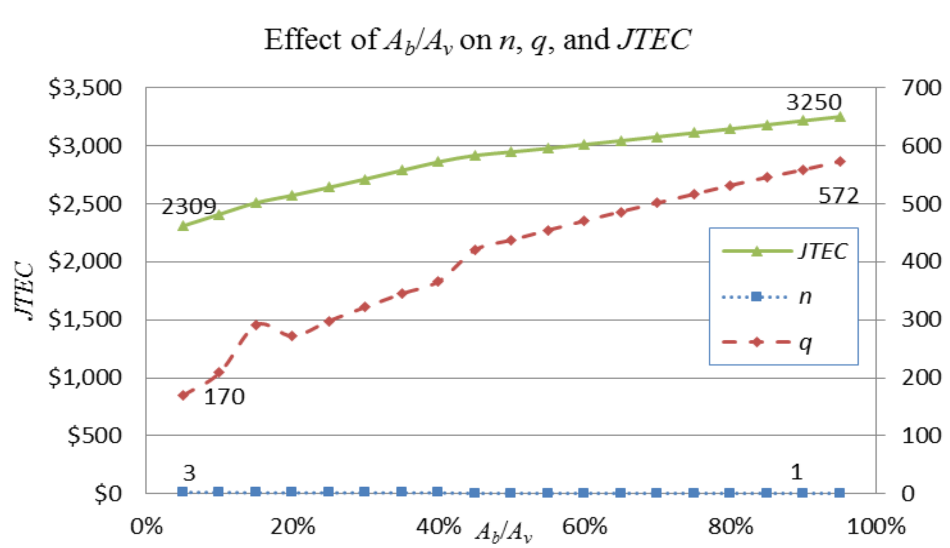


Figure 4.6 Effect of change in A_b/A_v on optimal solutions of Example 4.1

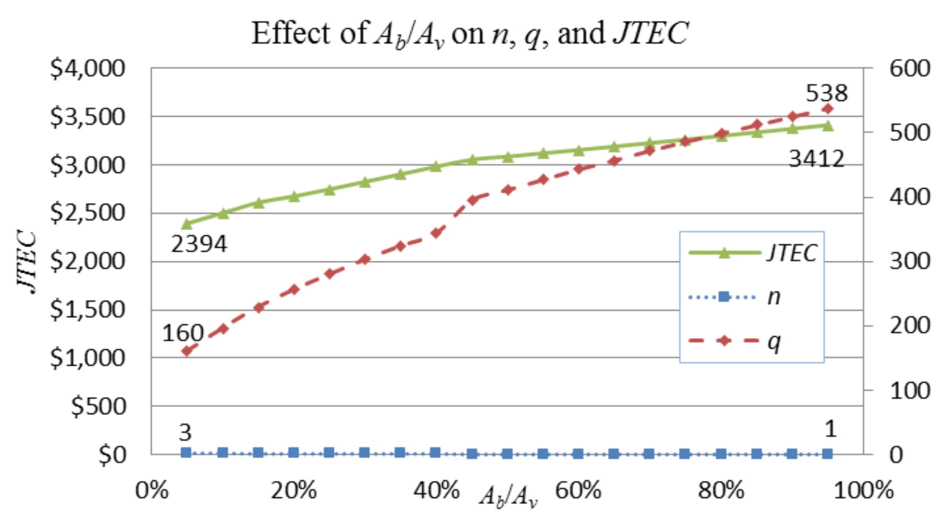


Figure 4.7 Effect of change in A_b/A_v on optimal solutions of Example 4.2

4.6.3 Effect of h_b^s/h_v^s on *JTEC*, *n*, and *q*

The effect of h_b^s and h_v^s on the model is shown in Figures 4.8 and 4.9. Note that Figure 4.8 shows the effect of h_b^s / h_v^s, whereas Figure 4.8 shows the effect of h_v^s / h_b^s. The ratio is so arranged (the larger one in the denominator) that we could see the consistency of the effect. The results implies the greater the difference between h_b^s and h_v^s, the better the system.

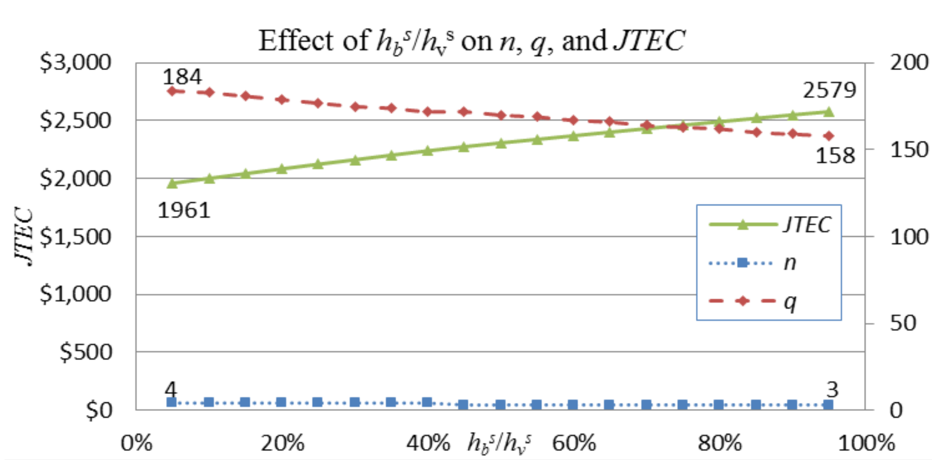


Figure 4.8 Effect of change in h_b^s / h_v^s on optimal solutions of Example 4.1

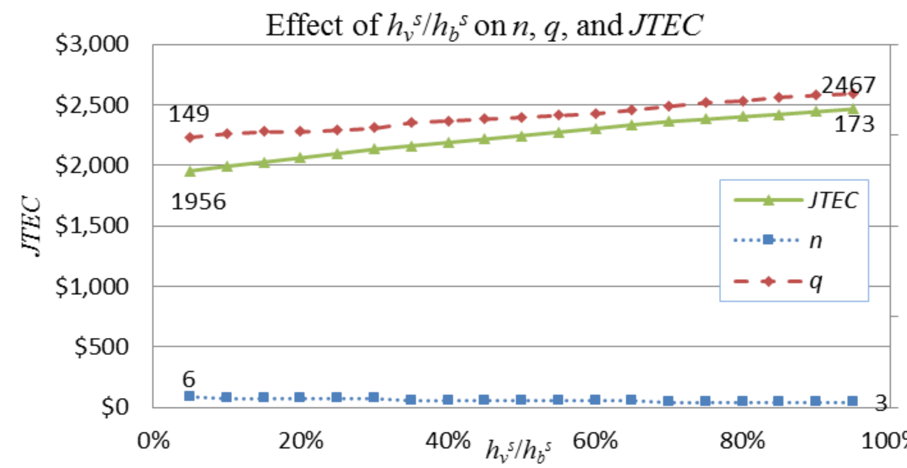


Figure 4.9 Effect of change in h_v^s / h_b^s on optimal solutions of Example 4.2

4.6.4 Effect of c_i and p_v on *JTEC*, *n*, *q*, and *L*

The parameter c_i have several values. It seems logical to study the impact of all these c_i's changing simultaneously. Figures 4.10 and 4.11 show the effect of c_i on the model for Example 4.1 and 4.2, respectively. Note that the horizontal axis is c_3/365, which is the daily crushing cost of the last segment of the lead time. The values of the rest of the c_i's were not listed, but they are also changing, accordingly. Generally speaking, both of the examples follow a similar pattern. Specifically, the *JTEC* and *q* are piecewise increases on c_i, while

the number of shipment of each production cycle n, and the lead time L is relatively stable. The figures indicate that largest change occurs when $c_3/365$ increases from \$10/day to \$15/day. Within that range, all three decision variables changed significantly. Minor perturbations in solution can be found when $c_3/365$ moves from \$35/day to \$40/day in Example 4.2, and from \$90/day to \$95/day in both examples 4.1 and 4.2, where only n and q changed.

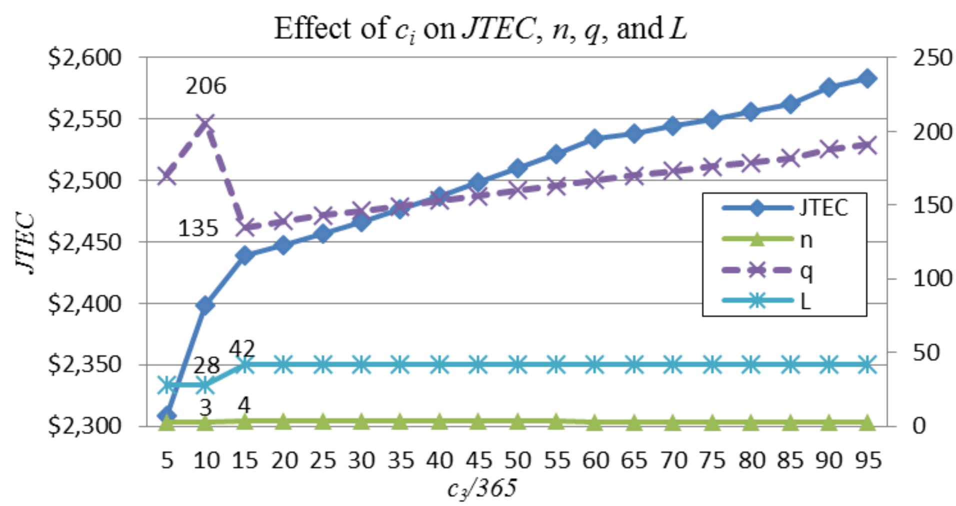


Figure 4.10 Effect of change in c_i on optimal solutions of Example 4.1

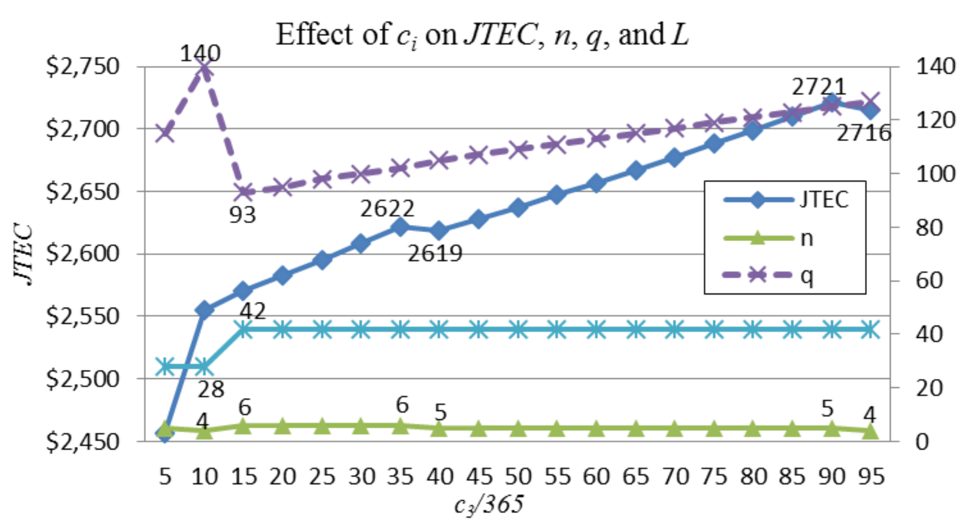


Figure 4.11 Effect of change in c_i on optimal solutions of Example 4.2

Figures 4.12 and 4.13 show the effect of p_v on the solutions. It can be seen that the *JTEC* is more sensitive to the changes in p_v than to the changes in c_i, whereas the other two variable n and q are more stable to the changes in p_v, as there are less vibrations on these figures. Again, the most useful information is that the optimal solutions of the three variables n, q, and L will all be changed significantly when p_v increases from \$150/unit to \$160/unit for Example 4.1 and from \$140/unit to \$150/unit for Example 4.2.

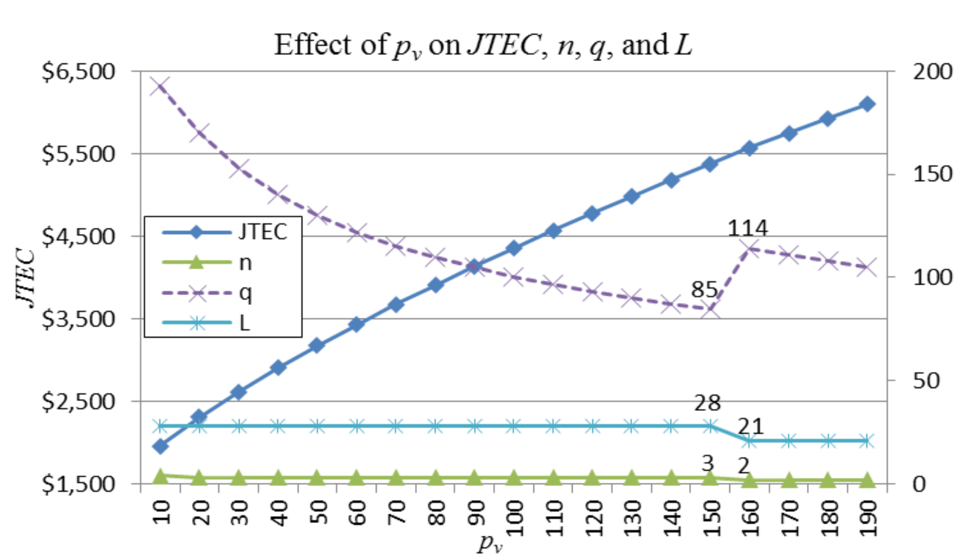


Figure 4.12 Effect of change in p_v on optimal solutions of Example 4.1

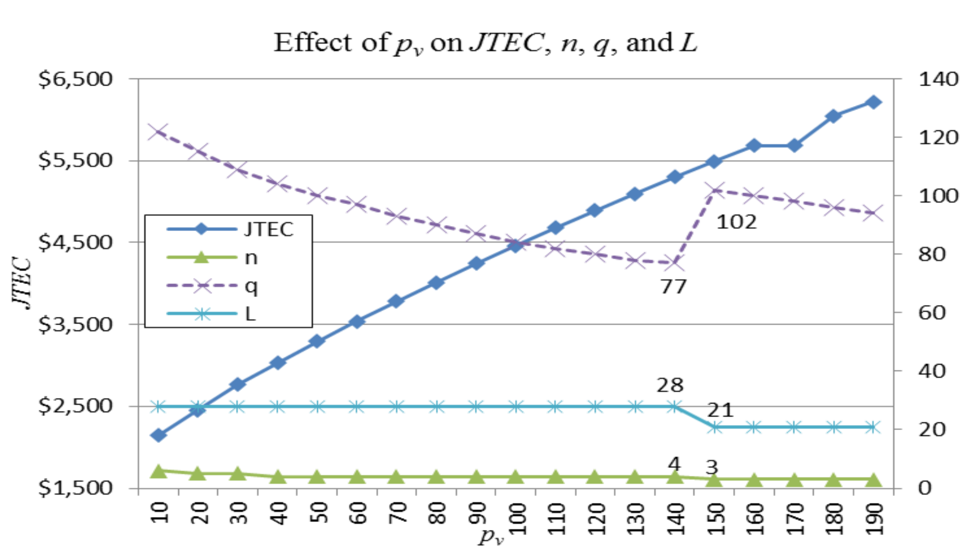


Figure 4.13 Effect of change in p_v on optimal solutions of Example 4.2

4.7 CONCLUSION

A four-variable EOQ/EPQ model is developed here and a solution procedure to find the optimal solution is provided for an integrated one-vendor, one-buyer consignment stock replenishment system with a controllable lead time. First, the research studies the impacts of controllable lead time, which is an important and practical factor that has never been studied under the CS cases. Second, the developed solution procedures are developed based on some properties of the objective function of proposed CS model. Third, the proposed algorithm does not have to search over the dimension of the number of shipments within each production cycle, *n;* rather, it directly starts the search from the real global optimal point with values of n and q, and is, therefore, more efficient, and lastly, this research addressed an important issue of finding out the integer global optima from the real global optima. The numerical example showed that considering the combined effect of adopting a consignment stock policy and lead time crushing opportunities may in some cases lead to a better result than any of these two policies, considered separately. Under what condition will it be favorable to invest on reducing lead time is also discussed and the results are illustrated in the sensitivity analysis Section. The results obtained here help understand the CS mechanism better and reveal the role of controllable lead time in the integrated system.

CHAPTER V
TWO-FACTOR CS MODEL: CONTROLLABLE LEAD TIME AND BUYER'S SPACE LIMITATION

The study of the buyer's space limitation to the integrated Consignment Stock (CS) system is important in that it forces both the vendor and the buyer to deviate from their optimal operational decisions. To further extend the study of the CS policy, the two-factor CS model that considers both the effects of controllable lead time and buyer's space limitation is developed in this Section. The objective of this model is to jointly decide the optimal ordering size, number of shipments within each production cycle, the number of delay shipments within each cycle, and the lead time, that minimize the annual joint total expected cost (*JTEC*) of the system.

5.1 THE PROBLEM

This Chapter considers an integrated vendor-buyer system, under CS policy, where the buyer places a space limitation to the vendor and the lead-time is controllable with added investment. Within any production cycle, the vendor produces at a finite rate and ships to the buyer with a number of equal-sized lots. With a long-term consignment stock agreement, the vendor takes responsibility to maintain a certain inventory level in the buyer's warehouse. Some of the shipments are delayed so that the buyer's inventory does not go beyond the capacity limitation. The buyer compensates the vendor after the complete consumption of the products. The holding cost consists of a storage component and a financial component. Two constraint four-variable, non-linear integer optimization models

are developed. The first model (two-factor CS model) adopts a replenishment policy that was described in Chapter 4, that is, when the buyer's maximum inventory level $I_{\max}$ is reached, all the following shipments are delayed for a certain period such that the arrival of new shipments to the buyer bring the buyer's inventory level up to $I_{\max}$. The second model (modified Two-factor CS model) uses another policy: when $I_{\max}$ is reached, the following shipments are delayed for a period so that the arrival of new shipments bring the buyer's inventory up to the buyer's space limitation *U*.

The remainder of this Chapter is organized as follows. Section 5.2 first defines local parameters and assumptions for the two-factor CS models. Then, the two two-factor CS models are formulated in Section 5.3. Computational results of the Doubly-hybrid Meta-Heuristic methods as well as that of some other algorithms are illustrated and compared in Section 5.4. The effects of important parameters are analyzed in Section 5.5. Conclusions are made in Section 5.6.

5.2 NOTATION AND ASSUMPTIONS

The two-factor CS models consider the same problem as the one-factor CS models, except that the two factor CS models treat the buyer's space limitation as an additional decision variable. Therefore, no additional assumption is needed. The following are some of the local notations that are added in this Chapter:

Additional Variable: k

Local Notations:

I_{max}: Buyer's maximum inventory level (units).

U: Buyer's space limitation (unit).

5.3 MODEL FORMULATION

In this Section, the formulations of the two two-factor CS model are given separately in Sections 5.3.1 and 5.3.2 as below.

5.3.1 The two-factor CS Model

In the two-factor CS model we consider a single-vendor single-buyer integrated system in which the vendor produces at a finite production rate P, which is greater than the buyer's demand rate D to ensure no backorder. Under the agreement, the vendor satisfies the buyer's demand in each production cycle with n shipments of equal lot size q. In the beginning, the shipments are made whenever the vendor's inventory reaches the level q. Each shipment takes the length L to reach the buyer but the decision maker has an option to reduce the lead time with a "crushing" cost, which is a function of reduced time. Since the buyers offers a space limitation U to the vendor, when the buyer's inventory reaches the point $I_{\max} = (n-k)q - (n-k-1)qD/P \leq u$, all the following k shipments are delayed for a period such that the arrival of a new shipment always bring the buyer inventory back to I_{max} (Figure 4.1b). Note that the number of delayed shipments k has an implicit upper bound $n-1$ since the first delivery cannot be delayed in order to guarantee that there is no shortage.

The two-factor CS model uses the same setting as the one-factor CS model. Therefore, the *JTEC* of the two-factor CS model is the same as that of the one-factor CS model which was given in Equation (4.6). The only difference here is the buyer's space limitation, which affects the system as a constraint. Using Equation (4.6), together with the system constraints, the two-factor CS model can therefore be written as:

$$\text{Min} \quad JTEC(q,n,k,L) = (A_v + nA_b)\frac{D}{nq} + (h_v^s + h_v^f)q\frac{D}{2P} + (h_b^s + h_v^f)q\frac{nP-(n-1)D}{2P}$$

$$+ (h_v^s - h_b^s)q\frac{(P-D)(k+1)k}{2nP} + (h_d^s + h_v^f)DL + \frac{D}{q}R(L), \qquad (5.1)$$

Subject To: $$(n-k)q - (n-k-1)qD/P \le U, \qquad (5.2)$$

$$k \le n-1, \qquad (5.3)$$

q, *n*, *k*, *L* are positive and integers. (5.4)

Note that the buyer space limitation *U* only appears in the constraint Equation (5.2), but not in the objective function in Equation (5.1). Due to the integer nature of the variables *n*, *k*, and *q*, the constraint in Equation (5.2) apply when it takes equality sign, which implies that in this model, *U* only affect the system when it takes some discrete values.

It is also noticeable that the number of delayed shipments, *k*, only appears in the fourth term of the Equation (5.1), which is the only term in the function that may have negative values that depend on the sign of $h_v^s - h_b^s$. It follows that, when the unit inventory holding cost is greater to the buyer than to the vendor, i.e., when $h_v^s - h_b^s < 0$, the *JTEC* is minimized when *k* is maximized. Observe that increasing *k* will not violate the constraint in Equation

(5.2). Hence, k can take the maximum value of $n-1$. Therefore, when the unit holding cost is greater at the buyer, the two-factor CS model reduces to the one-factor CS model 2, the CS-($k=n-1$) model. The pattern of the vendor's inventory, the buyer's inventory, the inventory in transit and the system inventory of the CS-($k=n-1$) model is given by Figure 5.1. Under this case, the buyer's average inventory is kept at the lowest level of $q/2$.

Conversely, when $h_v^s - h_b^s$ is positive, the *JTEC* is minimized when k is minimized. Therefore, if the buyer space limitation U is higher than the maximum inventory level, which is the buyer's maximum inventory level when there is not space limitation and can be calculated as $nq-(n-1)qD/P$, k can be as small as zero. The two-factor CS model is then reduced to the one-factor CS model 1, the CS-($k=0$) model. The pattern of all the inventories can be seen by Figure 5.2. Under this scenario, the vendor's average inventory is kept at the lowest level of $q/2$. However, if the buyer space limitation U is less than $nq-(n-1)qD/P$, then k has to satisfy constraint in Equation (5.2). In other words, the one-factor CS models 1 and 2 are two special cases of the two-factor CS model.

5.3.2 The modified two-factor CS Model

For the two-factor CS model, when the unit holding cost is higher to the vendor than to the buyer, the *JTEC* is minimized by minimizing the number of delayed shipments k. However, the maximum buyer's inventory is $I_{\max}=(n-k)q-(n-k-1)qD/P$, which is less than or equal to U. The two-factor CS model does not make the most use of the buyer's warehouse space and thus, this replenishment policy is not the optimal solution.

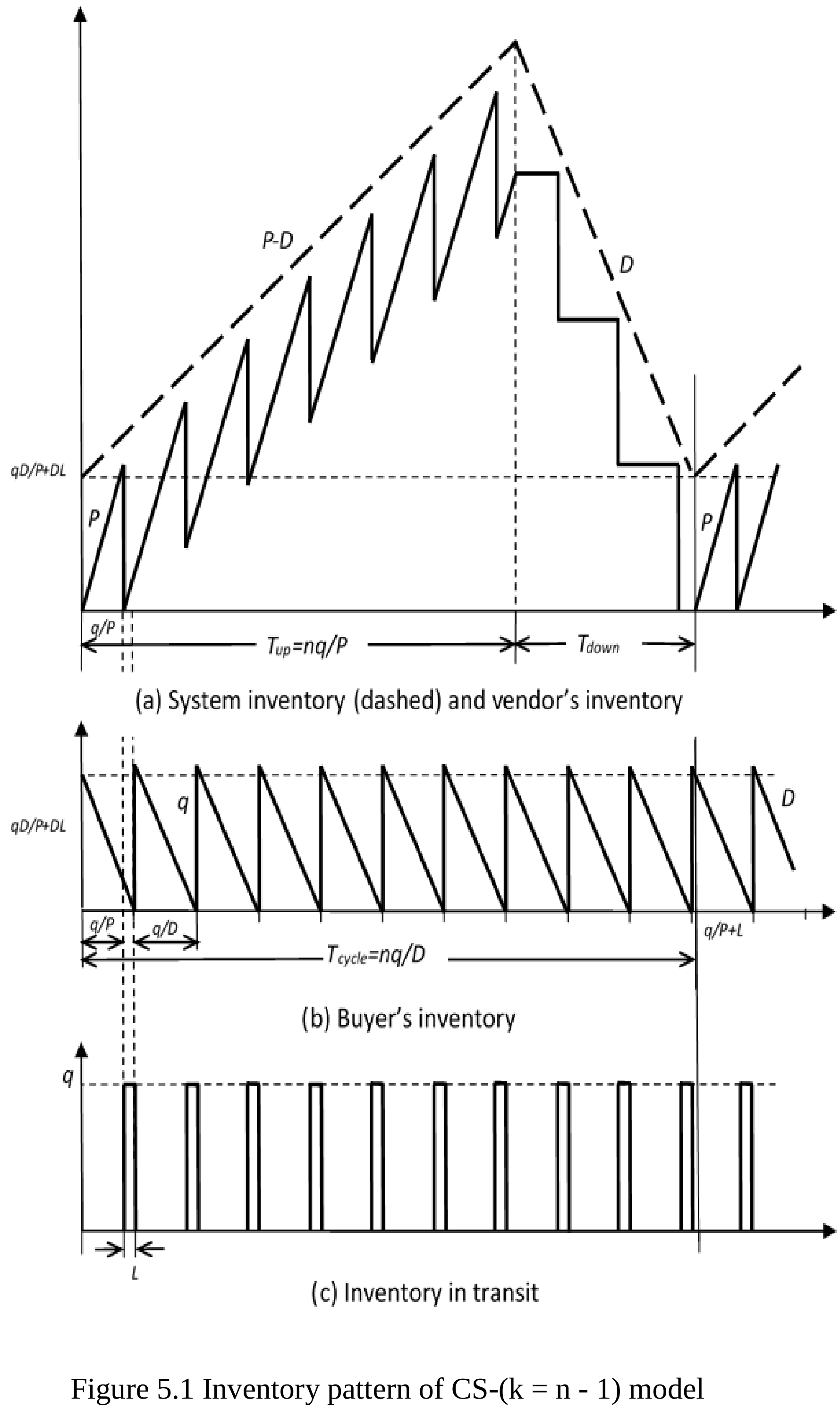


Figure 5.1 Inventory pattern of CS-(k = n - 1) model

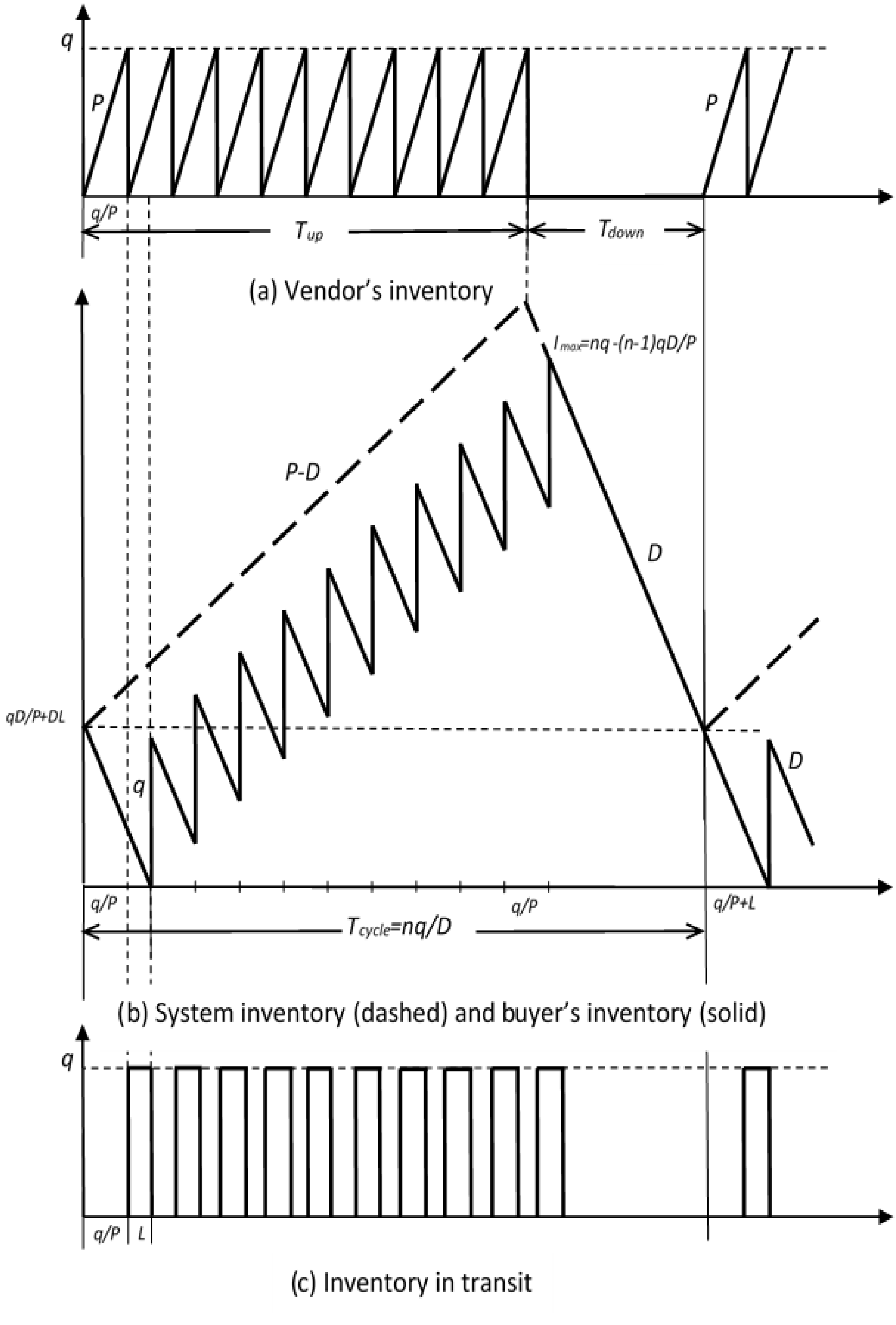


Figure 5.2 Inventory pattern of CS-(k = 0) model

A modification of the replenishment policy can be made to improve the use of the buyer's space. The model that makes the most uses of all of the buyer's space limitation U is thus defined as the modified two-factor CS model. The pattern of the system inventory, the vendor inventory, the inventory in transit, and the buyer inventory of the modified two-factor CS model are shown on Figure 5.3. In the modified two-factor CS model, when the buyer inventory reaches the point that the new arrival of another shipment will result in the exceeding of the buyer's space limitation, the ensuing shipments are delayed so that the arrival of each shipment brings the buyer inventory to the space limitation U instead of to the maximum inventory level I_{max}.

According to Figure 5.3, the buyer's average inventory I_b can be calculated as the shaded area S (Figure 5.3b) divided the cycle time T_{cycle}. That is,

$$I_b = S / T_{cycle} = \left[\frac{n^2q^2 + k^2q^2 - 2nkq^2 - kq^2 + 2ukq}{2D} - \frac{(n-k)(n-k-1)q^2}{2P} \right] \frac{D}{nq}$$

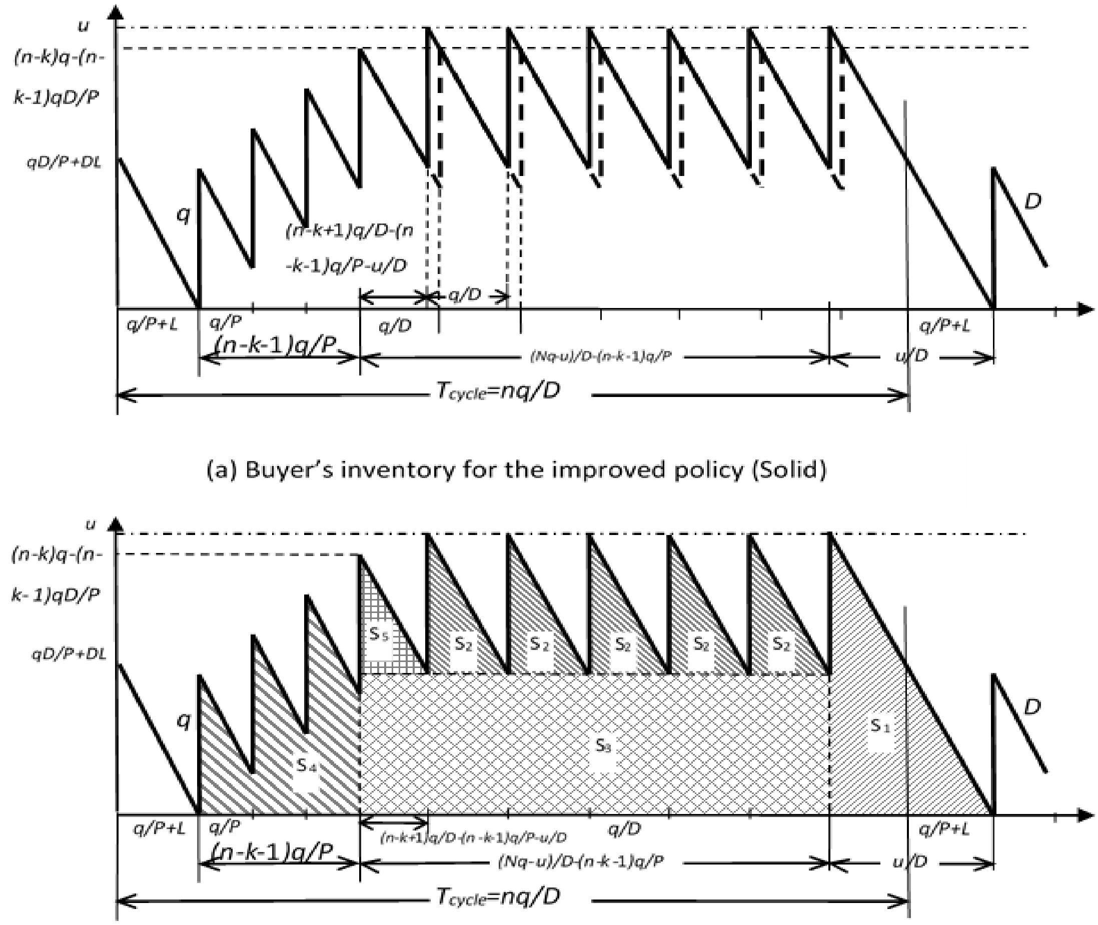
$$= \frac{n^2q + k^2q - 2nkq - kq + 2uk}{2n} - \frac{(n-k)(n-k-1)qD}{2nP} . \tag{5.5}$$

The calculation of S is given by Appendix A-5. The average system inventory, I_s, can be calculated as $I_s = \frac{nq(P-D)}{2P} + \frac{qD}{P} + DL$. The average inventory in transit is $I_d = DL$.

The average vendor inventory, I_v, can be given as

$$I_v = I_s - I_d - I_b = \frac{nq(P-D)}{2P} + \frac{qD}{P} - \frac{n^2q + k^2q - 2nkq - kq + 2uk}{2n} + \frac{(n-k)(n-k-1)qD}{2nP}$$

$$= \frac{2nkq - k^2q + kq - 2uk}{2n} + \frac{(k^2 + k + n - 2nk)qD}{2nP} . \tag{5.6}$$



(a) Buyer's inventory for the improved policy (Solid)

(b) Average buyer's inventory (shaded area)

Figure 5.3 Buyer's inventory in the modified two-factor CS model

Hence, the annual holding cost occurred at the buyer warehouse is given by

$$H_b = (h_b^s + h_v^f)\left[\frac{n^2q + k^2q - 2nkq - kq + 2uk}{2n} - \frac{(n-k)(n-k-1)qD}{2nP}\right]. \tag{5.7}$$

The annual holding cost in transit is

$$H_d = (h_d^s + h_v^f)DL\,. \tag{5.8}$$

The annual holding cost occurred at the vendor warehouse can be written as

$$H_v = (h_v^s + h_v^f)\left[\frac{2nkq - k^2q + kq - 2uk}{2n} + \frac{(k^2 + k + n - 2nk)qD}{2nP}\right]. \tag{5.9}$$

Therefore, the system annual holding cost is computed as

$$H = H_v + H_d + H_b = (h_v^s + h_v^f)\left[\frac{2nkq - k^2q + kq - 2uk}{2n} + \frac{(k^2 + k + n - 2nk)qD}{2nP}\right]$$

$$+ (h_b^s + h_v^f)\left[\frac{n^2q + k^2q - 2nkq - kq + 2uk}{2n} - \frac{(n-k)(n-k-1)qD}{2nP}\right] + (h_d^s + h_v^f)DL \,.$$

Upon simplification, it can be rewritten as

$$H = (h_v^s + h_v^f)\frac{qD}{2P} + (h_v^s - h_b^s)k\left[\frac{q(2n-k-1)(P-D)}{2nP} + \frac{q-u}{n}\right]$$

$$+ (h_b^s + h_v^f)q\left[\frac{n}{2} - \frac{(n-1)D}{2P}\right] + (h_d^s + h_v^f)DL \,. \tag{5.10}$$

Therefore, the *JTEC* of the **modified two-factor CS model** can be written as

$$\text{Min} \quad JTEC(q,n,k,L) = (A_v + nA_b)\frac{D}{nq} + (h_v^s + h_v^f)q\frac{D}{2P} + (h_b^s + h_v^f)q\frac{nP-(n-1)D}{2P}$$

$$+ (h_v^s - h_b^s)k\left[\frac{q(2n-k-1)(P-D)}{2nP} + \frac{q-u}{n}\right] + (h_d^s + h_v^f)DL + \frac{D}{q}R(L)\,, \tag{5.11}$$

Subject To: $$(n-k)q - (n-k-1)qD/P \le U\,, \tag{5.12}$$

$$k \le n-1\,, \tag{5.13}$$

q, n, k, L are positive and integers. (5.14)

The two-factor CS model and the modified two-factor CS model are both constrained continuous non-linear integer optimization problem. It is complicated to solve these models through traditional optimization methods. In this Chapter, the three hybrid meta-heuristic algorithms that were developed in Chapter 3 are applied to solve them.

5.4 COMPUTATION RESULTS AND DISCUSSIONS

In this Section, we use two numerical examples to test the efficiency of the three doubly-hybrid algorithms and illustrate the effects of the buyer space limitation. The outcomes of these three algorithms are compared to that of exhaustive search algorithm to verify that the solutions are global optimum. All the algorithms were coded in Matlab and were executed on a HP Pavilion Dv8 notebook PC with an Intel® QuadCore i7 CPU and Q 720@ 1.6 GHz processor. Each of the three algorithms is executed for 30 runs. If the optimal *JTEC* value, and the optimal solutions found by the doubly-hybrid algorithm match the outcome of the exhaustive search algorithm, we say that the doubly-hybrid algorithm succeed in that run. The mean CPU time, mean number of function evaluations (MNFE), and success rate of these 30 runs were recoded and compared to show the overall performance of each algorithm. The value the parameters of these two examples are adopted from former researchers (Braglia and Zavanella, 2003; Huang and Chen, 2009). The composition of the lead time and the associated unit crushing cost are as shown on Table 4.2. Other settings of the values and the computational results are summarized as below.

5.4.1 Numerical Example 5.1: $h_v^s > h_b^s$

The parameter values of numerical Example 5.1 are: $A_v = \$400$/setup, $A_b = \$25$/order, $D = 1000$ units/year, $P = 3200$ units/year, $r = 10\%$, $p_v = \$20$ /item, $h_v^s = \$3$ /item/year, $h_b^s = \$1.50$ /item/year, and $h_d^s = \$4$ /item/year. To test the effects of the buyer's space

limitation U, we set its value to be from 100 to 500 with a step size of 50. The value 500 is used as the buyer's space limitation level because preliminary computation showed that the maximum potential buyer's inventory level will not go beyond this level.

Computational results of the two-factor CS model

The optimal *JTEC* and solution are first found by exhaustive search and the results are shown in Table 5.1.

Table 5.1 Results of exhaustive search for the 2-factor CS model Example 5.1

U	*JTEC*	n	q	k	L(days)	NFE	CPU(s)
100	2632.90	5	100	4	28	29630916	1,402
150	2532.48	3	150	2	28	29630916	1,717
200	2489.69	4	118	2	28	29630916	2,307
250	2434.69	3	148	1	28	29630916	2,176
300	2392.42	3	177	1	28	29630916	2,149
350	2375.00	4	147	1	28	29630916	2,019
400	2340.57	3	168	0	28	29630916	1,920
450	2327.50	3	189	0	28	29630916	2,186
500	2327.50	3	189	0	28	29630916	2,017

In order to test the proposed hybrid algorithms, all the three algorithms are applied to Example 5.1 and their results listed in Table 5.2. To compare the efficiency and accuracy, six other meta-heuristic algorithms, MDE', PSO, IHS, MDE'+HJ, PSO+HJ, and IHS+HJ, were also utilized to this example. Their results are not shown here because of the unsatisfactory performance.

It can be found, from Table 5.2, that all the three doubly-hybrid algorithms require less mean CPU time than do the exhaustive search methods since the maximum number of

function evaluation (Max_NFE) allowed by the hybrid methods are set to be lower than that of exhaustive search method which has to run all possible solution outcomes.

Table 5.2 Results of doubly-hybrid algorithms for Example 5.1

Algorithm	Max_NFE	100,000	200,000	300,000	100,000	200,000	300,000
	U	Mean CPU	Mean CPU	Mean CPU	Success Rate	Success Rate	Success Rate
PSO+MDE'+HJ	100	16.518	35.048	50.765	0.500	0.433	0.500
	150	18.365	39.532	54.712	0.333	0.167	0.167
	200	19.928	40.367	59.878	0.067	0.300	0.067
	250	19.910	42.690	58.205	0.167	0.167	0.200
	300	22.163	45.113	58.570	0.000	0.100	0.100
	350	20.541	43.538	63.498	0.000	0.133	0.100
	400	20.981	48.378	54.595	0.067	0.067	0.067
	450	22.008	48.902	65.831	0.100	0.067	0.100
	500	20.687	46.226	62.908	0.200	0.200	0.133
Total		181.1	389.79	528.96	1.434	1.634	1.434
IHS+MDE'+HJ	100	22.396	36.593	74.123	0.867	1.000	1.000
	150	22.934	44.754	74.446	0.967	0.967	1.000
	200	25.832	39.991	81.292	0.100	0.433	0.167
	250	26.434	39.924	68.171	0.367	0.433	0.367
	300	26.286	39.399	56.105	0.833	1.000	0.967
	350	24.385	39.286	51.506	0.833	0.967	1.000
	400	22.885	36.432	42.506	0.867	1.000	1.000
	450	22.258	37.974	42.497	0.867	1.000	1.000
	500	22.040	37.929	42.607	0.867	1.000	1.000
Total		215.450	352.28	533.25	6.568	7.8	7.501
PSO+IHS+HJ	100	6.922	14.017	20.653	1.000	1.000	1.000
	150	7.255	15.212	21.941	1.000	1.000	1.000
	200	7.249	15.132	19.486	0.933	1.000	1.000
	250	7.405	15.555	19.056	1.000	1.000	1.000
	300	7.502	17.485	19.064	1.000	1.000	1.000
	350	7.956	18.510	19.213	0.467	0.900	1.000
	400	8.559	25.044	18.900	0.933	1.000	1.000
	450	8.889	27.193	19.035	0.8	0.867	1.000
	500	9.237	22.402	19.743	0.933	0.967	1.000
Total		70.97	170.55	177.09	8.066	8.734	9.000

Amount the three hybrid methods, PSO+IHS+HJ requires the least computation time and has the best success rate. Another finding is that the success rate of all three hybrid algorithms improved when the value of Max_NFE increases from 100,000 to 200,000. However, although further increasing the Max_NFE to 300,000 does improve the PSO+IHS+HJ to reach the perfection (perfection means that the success rate becomes all 100%), it does not further improve the performance of PSO+MDE'+HJ and IHS+MDE'+HJ.

In short, among the three hybrid methods, the PSO+MDE'+HJ do not work well. The speed of IHS+MDE'+HJ is about the same as PSO+MDE'+HJ, but the accuracy of IHS+MDE'+HJ is better than PSO+MDE'+HJ, so the overall performance of IHS+MDE'+HJ algorithm is acceptable. The most important conclusion is that the results of PSO+IHS+HJ algorithm are satisfactory both in the sense of success rate and mean CPU time, considering that this algorithm is based on search and not on derivative information, it has, therefore, a potential to be used to solve other complicated inventory models.

Effects of buyer space limitation *U*

It can be seen from Table 5.1 that the *JTEC* decreases as the value of *U* increases. This is natural since when h_v^s is greater than h_b^s, the *JTEC* decreases when more inventory are kept in the buyers warehouse. Another observation is that the decreasing rate of the *JTEC* decreases when *U* is increasing (i.e., the rate is more flat at higher space availability). So, the negative effect of *U* on the system cost is more at tighter limitation which suggests the decision maker consider a larger *U* when s/he could. Note that there is an exception for

this negative relationship in Table 5.1: when *U* increases from 450 to 500, the system cost does not further reduce. This is because the buyer's maximum inventory level, when without a space limitation, is $nq-(n-1)qD/P=$ 448.875, which is less than 450. This means that any space limitation that is greater than 449 does not affect the system at all. Another observation is that some decision variables such as *n*, *q*, and *k*, do not seem to have linear relationship with *U*. In order to further investigate the effects of *U* on the whole system in more detail, we used the doubly-hybrid meta-heuristics method, the IHS+MDE'+HJ, to numerical example 5.1 with the integer value of *U* from 100 to 450. Figure 5.5 and 5.6 illustrate, respectively, the detailed effects of *U* on *JTEC* and *q* and the effects of *U* on *n* and *k*.

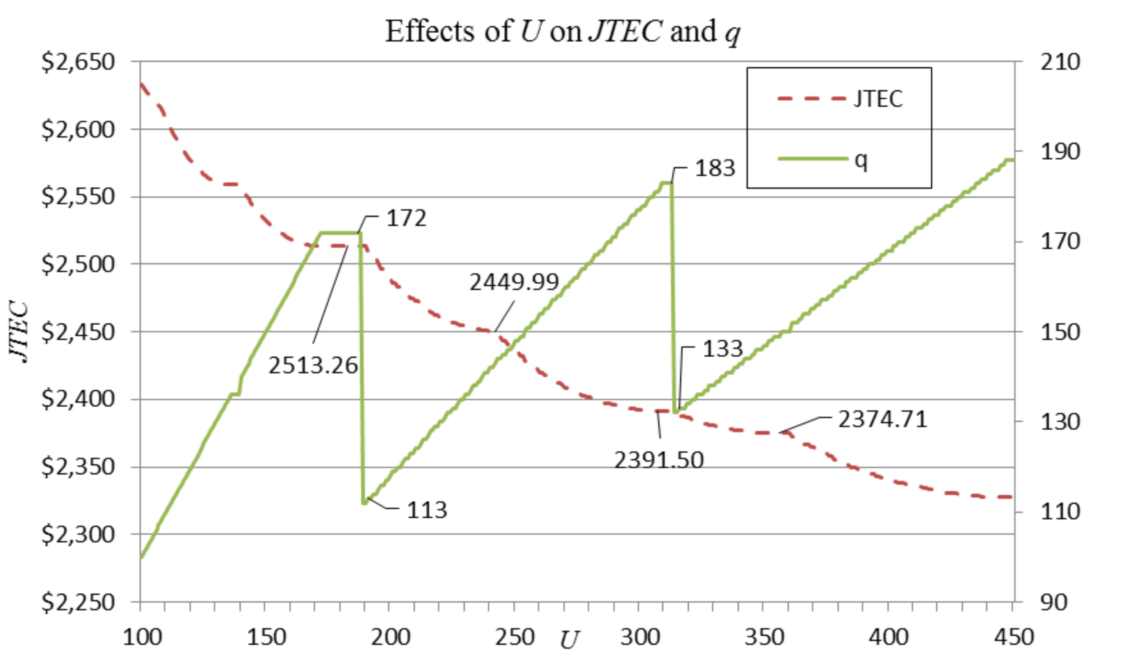


Figure 5.4 Effects of *U* on *JTEC* and *q* for Example 5.1 of the two-factor CS model

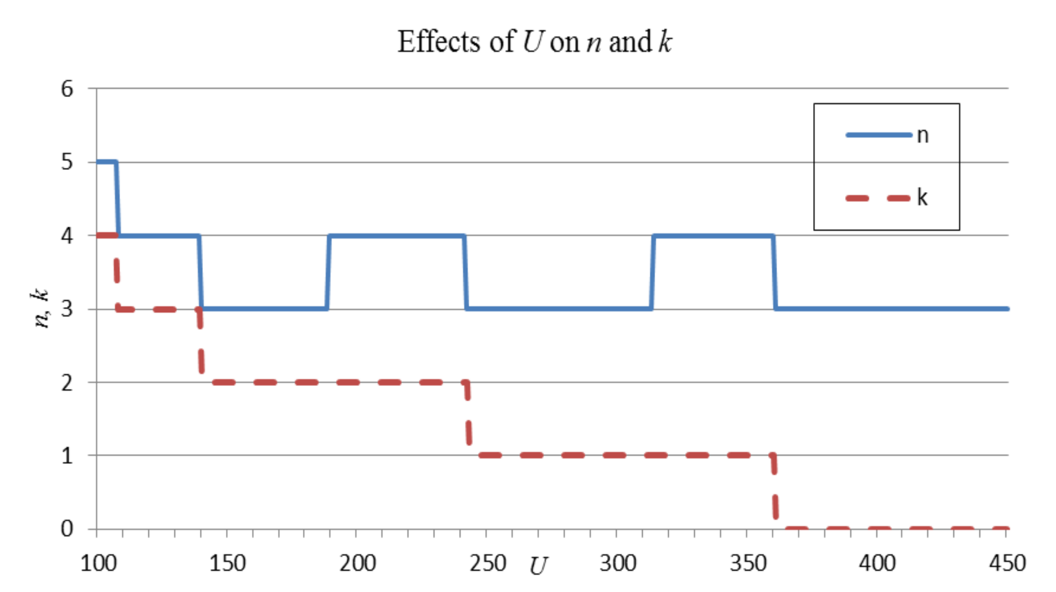


Figure 5.5 Effects of U on n and k for Example 5.1 of the two-factor CS model

Table 5.3 Part of the effects of U on the system

U	$JTEC$	n	q	K
…	…	…	…	…
170	2513.39	3	170	2
171	2513.29	3	171	2
172	2513.26	3	172	2
…	…	…	…	…
188	2513.26	3	172	2
189	2513.22	4	112	2
190	2513.22	4	112	2
191	2508.82	4	113	2
192	2508.82	4	113	2
193	2504.62	4	114	2
…	…	…	…	…

Figures 5.4 and 5.5 show more clearly the effects of U to the system. The total cost $JTEC$, n, and k, are actually piecewise decreasing functions of U, while as q is a piecewise increasing function of U. At the end of each of these pieces, there is a platform within which, changing in U does not affect the total cost and the solutions. This can be seen from the

original table from which we draw Figures 5.4 and 5.5. Because of the size, the whole table is not showing here. Instead, a portion of it is shown in Table 5.3. To better illustrate the effects of *U* on *JTEC* and *q*, we further enlarged a portion of Figure 5.4 with Figure 5.6 which uses the value of *U* as from 160 to 200.

It is clear from Table 5.3 and Figure 5.6 that none of the value of *JTEC*, *q*, *n*, or *k* changes when *U* takes the value between 172 and 188. This result is useful to the decision maker because a smaller *U* = 172 is better than any values from 173 to 188 if s/he happens to have a space limitation within that range. Moreover, there are a lot of small platforms in which the *JTEC* does not change, but the *q*, *n*, and *k*, may or may not change (Table 5.3). This information may provide the manager flexibility in choosing the value of *U*.

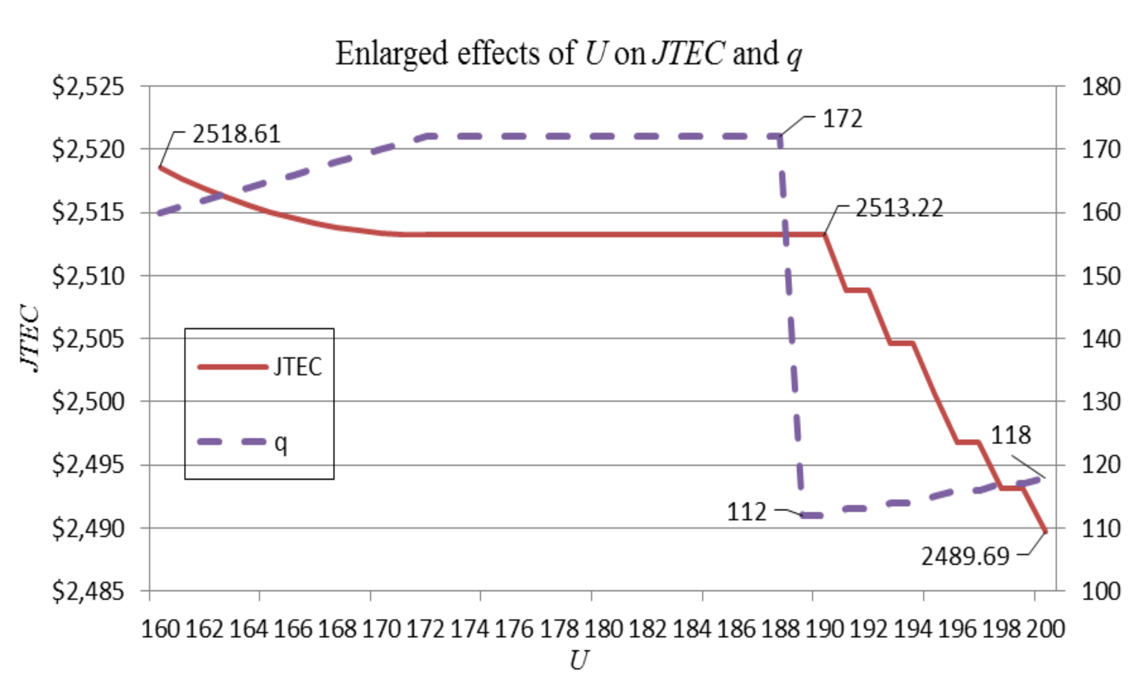


Figure 5.6 Effects of *U* on the two-factor CS model when *U* is between 160 and 200

Computational results of the modified two-factor CS model

From the analysis to the Table 5.1, we know that the maximum buyer's space limitation that affects the system is 449, meaning that there is no way to improve the performance of the system if the value of *U* is set to be greater than 449. As a result, we set the value of *U* to be from 100 to 400 with a step size 50 to illustrate the results of the modified two-factor CS model. The optimal *JTEC* and solution found by exhaustive search are shown in Table 5.4. The results of the three hybrid meta-heuristic algorithms are listed in Table 5.5.

Table 5.4 Results of exhaustive search for modified 2-factor CS model Example 5.1

U	*JTEC*	*n*	*q*	*k*	*L*(days)	NFE	CPU(s)
100	2632.899	5	100	4	28	29630916	1,172
150	2532.475	3	150	2	28	29630916	1,237
200	2478.578	3	159	2	28	29630916	1,279
250	2425.771	4	127	3	28	29630916	1,731
300	2378.578	3	159	2	28	29630916	1,498
350	2338.567	3	175	2	28	29630916	1,576
400	2336.691	3	200	2	28	29630916	1,773

All the observations that applied to the two-factor CS model also apply to the modified two-factor CS model. Comparing Table 5.4 with Table 5.1, we notice that the *JTEC* of Table 5.4 is always less or equal to that of Table 5.1. This observation verifies the conclusion that the modified model is superior to the original one given that $h_v^s > h_b^s$. Again, Table 5.5 illustrates the reliability and the efficiency of the PSO+IHS+HJ algorithm.

Table 5.5 Results of 3 hybrid algorithms for modified 2-factor CS model Example 5.1

Algorithm	Max_NFE	100,000	200,000	300,000	100,000	200,000	300,000
	U	Mean CPU	Mean CPU	Mean CPU	Success Rate	Success Rate	Success Rate
PSO+MDE'+HJ	100	19.872	35.434	54.205	0.433	0.500	0.400
	150	23.608	43.031	52.341	0.467	0.333	0.400
	200	23.465	40.160	52.197	0.400	0.333	0.400
	250	19.945	45.399	52.669	0.3	0.167	0.367
	300	18.177	43.782	56.279	0.133	0.300	0.100
	350	14.851	50.567	59.049	0.133	0.167	0.100
	400	16.116	43.298	53.534	0.133	0.167	0.133
Total		136.030	301.670	380.270	1.999	1.967	1.900
IHS+MDE'+HJ	100	22.751	34.778	63.736	0.867	0.900	1.00
	150	24.548	36.999	73.003	0.767	0.933	0.933
	200	23.346	33.675	75.538	0.933	0.967	1.000
	250	21.804	35.696	76.447	0.800	1.000	1.000
	300	21.384	37.900	80.551	0.900	0.967	0.967
	350	22.285	35.982	66.099	0.533	0.733	0.867
	400	19.319	37.225	51.190	0.367	0.900	0.867
Total		155.440	252.250	486.560	5.167	6.40	6.634
PSO+IHS+HJ	100	8.801	18.221	38.442	1.000	**1.000**	**1.000**
	150	8.990	17.508	36.727	0.900	**1.000**	**1.000**
	200	8.948	18.200	31.677	1.000	**1.000**	**1.000**
	250	7.672	19.175	25.528	0.900	**1.000**	**1.000**
	300	7.023	21.765	25.647	0.900	**1.000**	**1.000**
	350	7.969	22.304	25.212	0.700	**0.933**	**1.000**
	400	10.965	23.067	31.941	0.867	**0.933**	**1.000**
Total		60.370	140.240	**215.170**	6.267	**6.866**	**7.000**

A detailed computation of the effects of U on the modified two-factor CS model is also conducted and the result is shown in Figures 5.7 and 5.8. To show the comparison, we used a prime mark in Figures 5.7 and 5.8 to indicate that it is the result of the modified two-factor CS model. It is clear that the modified model is always at least as good as the two-factor CS model. It is noticeable that, within the platform, where the increasing of U cannot improve the two-factor CS model, the *JTEC* of the modified two-factor CS model still improves.

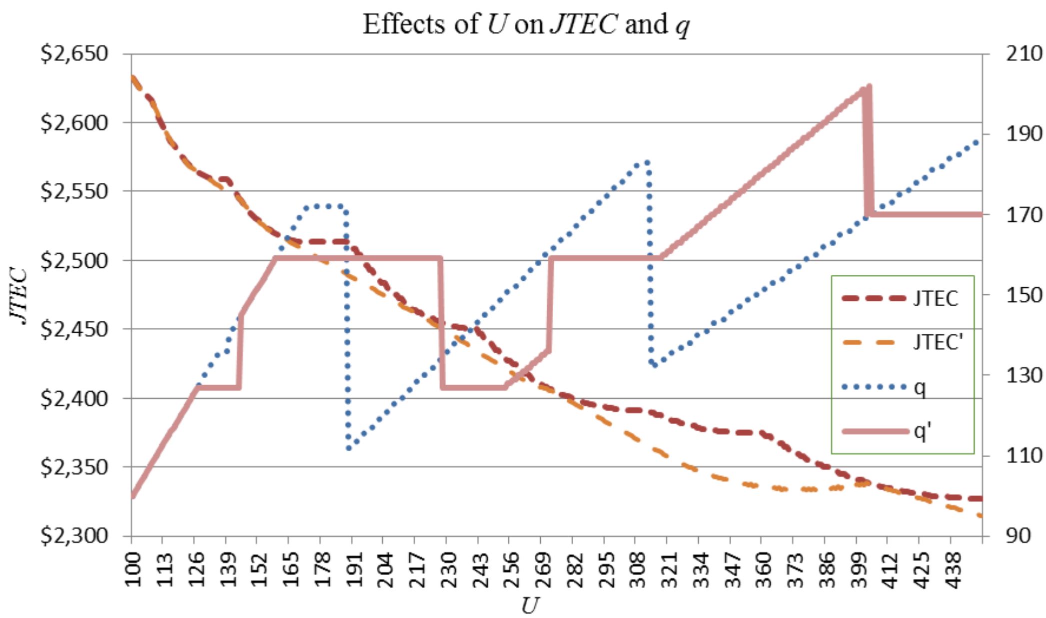


Figure 5.7 Comparison of effects of U on $JTEC$ and q of both models

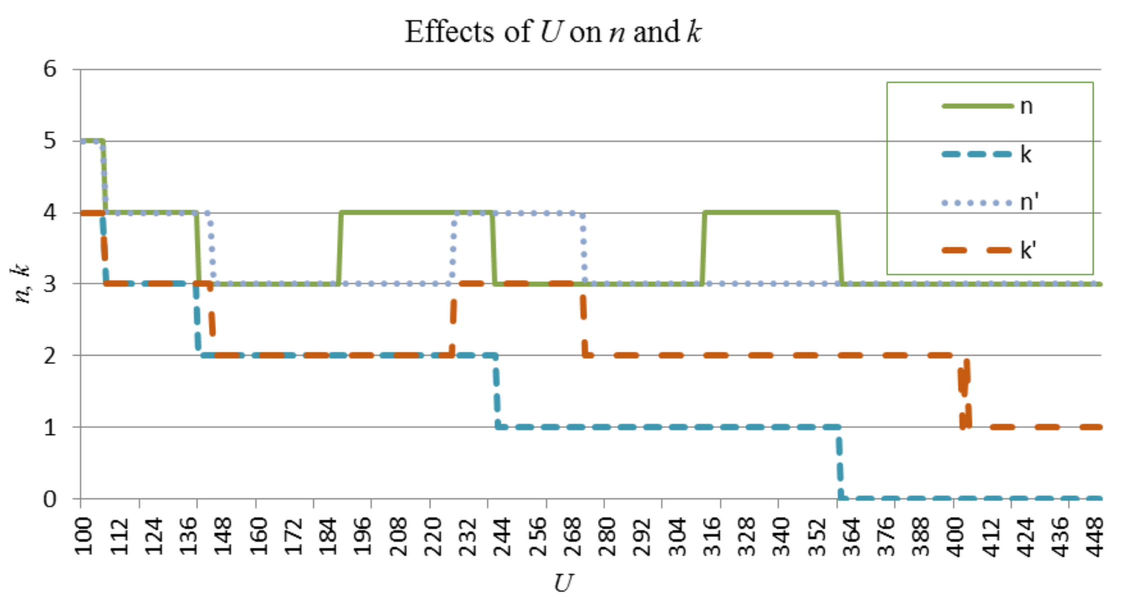


Figure 5.8 Comparison of effects of U on n and k of both models

5.4.2 Numerical Example 5.2: $h_v^s < h_b^s$

The parameter values of numerical example 5.2 are: $A_v = \$400$/setup, $A_b = \$25$/order, $D = 1000$ units/year, $P = 3200$ units/year, $r = 10\%$, $p_v = \$15$ /item, $h_v^s = \$2.5$ /item/year,

$h_b^s = \$3$/item/year, and $h_d^s = \$4$/item/year. The buyer's space limitation U is still set to be from 100 to 500 with a step size of 50.

Note that in this case, the optimum solution requires the maximum number of delayed shipments so that the buyer only takes an average inventory of $q/2$. It follows that the modified two-factor CS model does not improve the performance of the system. Therefore, in Example 5.2, only the two-factor CS model is needed. The results found by exhaustive search and three doubly-hybrid algorithms are given by Tables 5.6 and 5.7, respectively.

Table 5.6 Results of exhaustive search for two-factor CS model Example 5.2

U	*JTEC*($)	*n*	*q*	*k*	*L*(days)	NFE	CPU(s)
100	2,491.42	5	100	4	28	29,630,916	1,395
150	2,424.32	4	143	3	28	29,630,916	2,110
200	2,414.91	3	177	2	28	29,630,916	2,741
250	2,414.91	3	177	2	28	29,630,916	2,447
300	2,414.91	3	177	2	28	29,630,916	2,443
350	2,414.91	3	177	2	28	29,630,916	2,250
400	2,414.91	3	177	2	28	29,630,916	2,511
450	2,414.91	3	177	2	28	29,630,916	2,258
500	2,414.91	3	177	2	28	29,630,916	1,951

Table 5.6 first verified the conclusion that when $h_v^s < h_b^s$, the optimal solution requires $k = n - 1$. Other observations to Table 5.1 also apply to Table 5.6. Table 5.7 reveals that both algorithms MDE'+IHS+HJ and PSO+IHS+HJ perform well in Example 5.2. The algorithm MDE'+PSO+HJ is still not acceptable.

Table 5.7 Results of the three hybrid algorithms for two-factor CS model Example 5.2

Algorithm	Max_NFE	100,000	200,000	300,000	100,000	200,000	300,000
	U	Mean CPU(s)	Mean CPU(s)	Mean CPU(s)	Success Rate	Success Rate	Success Rate
PSO+MDE'+HJ	100	14.185	39.189	45.808	0.933	0.933	1.000
	150	14.210	47.158	69.186	0.567	0.467	0.233
	200	14.499	47.488	68.700	0.433	0.300	0.167
	250	14.416	40.279	62.965	0.233	0.133	0.167
	300	14.655	37.667	68.538	0.333	0.133	0.200
	350	14.593	35.998	68.324	0.400	0.267	0.100
	400	14.900	39.574	55.125	0.300	0.133	0.267
	450	15.580	40.157	50.856	0.233	0.067	0.267
	500	16.754	36.418	76.278	0.367	0.133	0.133
Total		133.790	363.930	565.780	3.799	2.566	2.534
IHS+MDE'+HJ	100	23.859	32.389	64.849	0.833	**1.000**	**1.000**
	150	27.066	38.785	61.783	1.000	**1.000**	**1.000**
	200	27.392	43.149	48.501	0.967	**1.000**	**1.000**
	250	24.971	41.453	46.637	1.000	**1.000**	**1.000**
	300	29.148	41.954	46.610	1.000	**1.000**	**1.000**
	350	29.056	45.148	46.794	1.000	**1.000**	**1.000**
	400	29.230	41.409	46.379	0.967	**1.000**	**0.967**
	450	27.817	44.711	46.377	0.967	**1.000**	**0.967**
	500	25.480	49.489	46.622	1.000	**1.000**	**1.000**
Total		244.020	378.49	454.550	8.734	**9.000**	**8.934**
PSO+IHS+HJ	100	8.690	18.248	32.837	1.000	**1.000**	**1.000**
	150	8.968	20.859	33.420	0.833	**1.000**	**1.000**
	200	8.986	23.290	33.273	0.800	**1.000**	**1.000**
	250	9.369	23.147	32.076	0.867	**1.000**	**1.000**
	300	9.803	20.874	32.563	0.967	**1.000**	**1.000**
	350	9.787	25.277	30.751	0.933	**1.000**	**1.000**
	400	9.582	25.143	28.395	0.700	**0.933**	**1.000**
	450	9.750	25.887	32.363	0.733	**1.000**	**1.000**
	500	10.247	26.011	36.361	0.767	**1.000**	**1.000**
Total		85.180	208.740	**292.040**	7.600	**8.933**	**9.000**

Table 5.6 shows that *U* only affects the system when it is less than 200. Therefore, we conduct a detailed computation finding the optimal solutions and the corresponding *JTEC* when *U* is chosen from 100 to 200. Figures 5.9 and 5.10 illustrate the results.

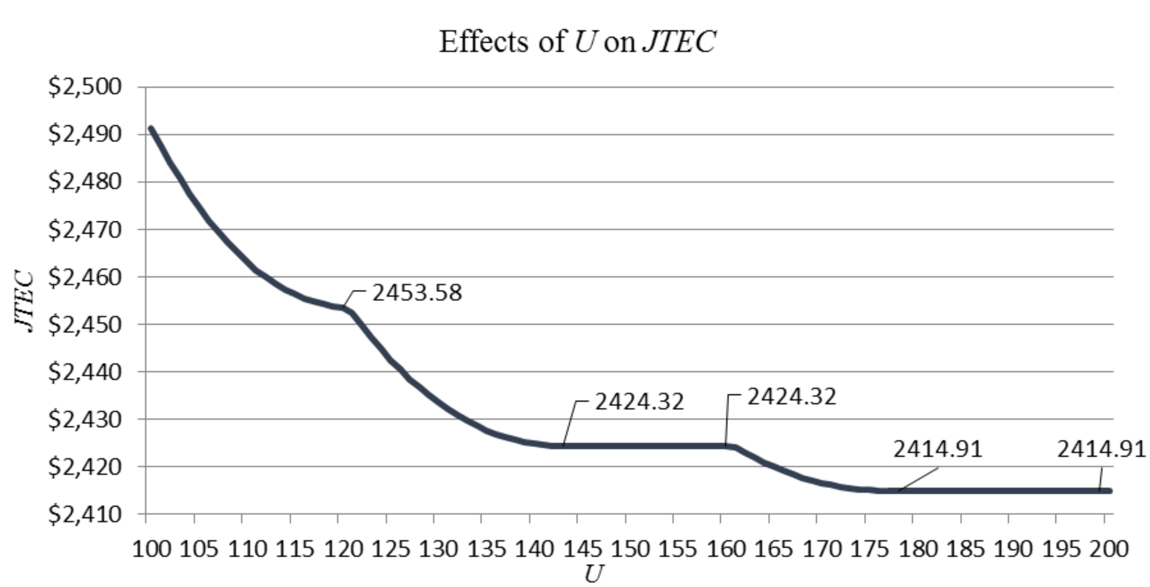


Figure 5.9 Effects of *U* on *JTEC* for Example 5.2

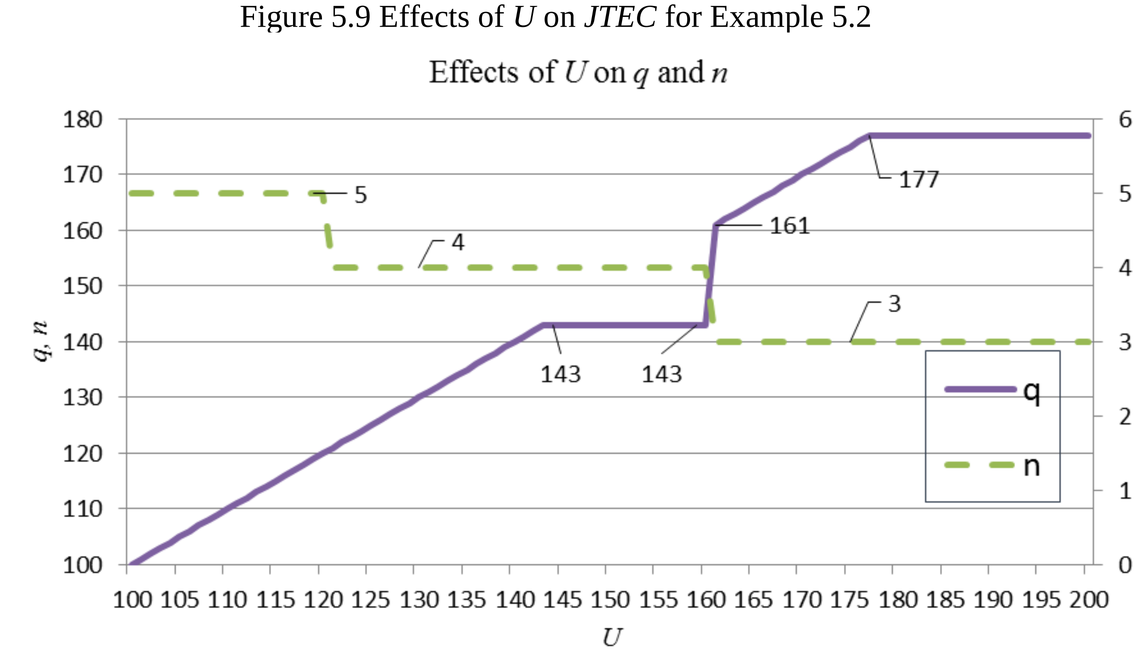


Figure 5.10 Effects of *U* on *n* and *q* for Example 5.2

The manner in which the buyer influences the vendor's inventory decisions is an interesting problem. There are different ways that the buyer can control, to a certain degree, the vendor's inventory decision. For example, the buyer can ask for a shorter/longer lead time to reduce/increase the entire inventory. The buyer may also offer an incentive in the

unit price to the buyer to encourage the buyer to keep an inventory that is more preferable to the buyer. In this integrated system, it is supposed that the vendor will always be willing to follow a policy that is best to the entire system. Therefore, when the buyer want to provide more space to other items or other companies, he can simply place a tighter space limitation on the vendor to force the vendor to modify his inventory decision.

5.5 CONCLUSION

This Section studied the effects of the buyer's space limitation in an integrated lead time controllable CS inventory system. Two integrated inventory models were developed to jointly determine the optimal value of four decision variables that minimize the annual *Joint Total Expected Cost* (*JTEC*) of the system. Due to the difficulty of the problem, analytical solutions of the models are not presented. Instead, three novel doubly-hybrid meta-heuristic algorithms are utilized to find the global optimum. Two numerical examples showed that at least one of these doubly-hybrid algorithm works very well both in the sense of the CPU time and the success rate. The computational analysis revealed how the buyer space limitation affects the *JTEC* and the best solution of the four decision variables. The results obtained in the Chapter helps understand the role of buyer space limitation and the CS mechanism better.

CHAPTER VI
THREE-FACTOR CS MODEL: CONTROLLABLE LEAD TIME, BUYER'S SPACE LIMITATION, AND VARIABLE DEMAND

The buyer's demand rate is seldom deterministic in reality. More often, it may follow a stochastic distribution. Variable demand incurs the risk of stock outs. When there is a space limitation in the warehouse, an uncertain demand may also incur the needs for extra space. Therefore, the study of the effects of stochastic demand rate is useful. Toward this end, the three-factor CS model that considers the effects of controllable lead time, buyer's space limitation, and a normally distributed buyer demand rate is developed in this Chapter. The objective of this model is to jointly decide the optimal ordering size, number of shipments within each production cycle, the number of delay shipments within each cycle, the lead time, and the safety stock, that minimizes the annual *JTEC* of the system.

6.1 THE PROBLEM

This Chapter considers an integrated vendor-buyer system, under Consignment Stock (CS) policy, where the demand is uncertain, but follows a normal distribution with known mean and standard deviation. Moreover, the buyer places a space limitation to the vendor and the lead-time is controllable with an extra investment. With a long-term CS agreement, the vendor takes responsibility to maintain a certain inventory level in the buyer's warehouse. Therefore, within any production cycle, the vendor produces at a finite rate and delivers the products to the buyer in small shipments. As a result, the buyer's inventory level increases gradually. When the buyer's inventory level reaches its maximum level, I_{max},

which is close or equal to the buyer's space limitation, all later shipments from the vendor are delayed for a certain period so that the buyer's inventory does not go beyond the capacity limitation. Because of the demand is uncertain, both backorder costs and extra space costs are considered and the system has to provide a safety stock to against the risk of stock out and the needs for extra spaces. The buyer compensates the vendor after the complete consumption of the products. The holding cost consists of a storage component and a financial component. In this Section, two constraint five-variable non-linear integer optimization models are established. The first model (three-factor CS model) adopts a replenishment policy that was described in Chapter 4, that is, when the buyer's maximum inventory level $I_{\max}$ is reached, all the following shipments are delayed for a certain period such that the arrival of new shipments to the buyer brings the buyer's inventory level up to $I_{\max}$. The second model (modified three-factor CS model) uses another policy: when $I_{\max}$ is reached, the following shipments are delayed for a period so that the arrival of new shipments brings the buyer's inventory up to the buyer's space limitation *U*.

The remainder of this Chapter is organized as follows. Section 6.2 first defines local parameters and assumptions for the three-factor CS models. The two three-factor CS models are then formulated in Section 6.3. Computational results of the Doubly-hybrid Meta-Heuristic methods, as well as that of the Exhaustive search method, are illustrated and compared in Section 6.4. The effects of important parameters are analyzed in Section 6.5. Section 6.6 summarizes this Chapter.

6.2 NOTATION AND ASSUMPTIONS

The three-factor CS models are different than the ones developed in Chapters IV and V. In order to develop the integrated models, some additional notations and assumptions other than that defined in Chapter IV are needed and are given below:

Additional Variable: k, s.

Local Notations:

I_{max}: Buyer's maximum inventory level (units).

c_b: Unit backorder cost (\$/unit),

C_b: Expected annual backorder cost (\$/year),

c_{il}: Unit crushing cost for reducing one time unit of the ith segment of lead time S_i when the ordering quantity q is between q_{l-1} and q_l (\$/year),

c_o: Unit outsourcing cost (\$/unit),

C_o: Expected annual extra space cost (\$/year),

D: Yearly demand rate at the buyers' level (units/year), $D \sim N(\mu, \sigma)$.

E(▪): Mathematical expectation of ▪,

$R(q, L)$: Lead time crushing cost per replenishment cycle (\$/shipment),

s: Safety factor,

s_s: Safety stock level,

U: Space limitation placed by the buyer to the vendor (units),

μ: Expectation value of annual demand rate D (units/year), $\mu = \int_{-\infty}^{+\infty} Df(D)dD$,

σ: Standard deviation of annual demand rate D (units),

v: Number of price segments associated with ordering quantity,

x^+: Maximum value of x and 0, i.e. $x^+ = \max\{x, 0\}$,

X_1: The demand during the period ($q/P + L$), having a mean $\mu(q/P+L)$ and standard deviation $\sigma\sqrt{q/P+L}$,

X_2: The demand during the period q/P, having a mean $q\mu/P$ and standard deviation $\sigma\sqrt{q/P}$,

X_3: The demand during the period q/μ, having a mean q and standard deviation $\sigma\sqrt{q/\mu}$.

Local Assumptions:

The following additional assumptions are necessary to the two models developed in this Chapter:

(1) The demand of the buyer D follows a normal distribution with a mean μ and a standard deviation σ, i.e., $D \sim N(\mu, \sigma)$.

(2) The demand during lead-time L also follows normal distribution with a mean μL and a standard deviation $\sigma\sqrt{L}$.

(3) Due to the demand uncertainty, shortage is allowed in the system and is backordered with a shortage cost.

(4) Also due to the demand uncertainty, extra inventories beyond the buyer's capacity are allowed. The extra products still may be stored in the buyer's warehouse, but in

a space reserved for other products/suppliers or they may be stored by a third party. In both cases the vendor will be charged an extra penalty cost.

(5) Unlike most other research which consider the shipping cost to be a function of the guaranteed lead time, this study assume that it is an incremental function of both the guaranteed lead time and the shipment quantity.

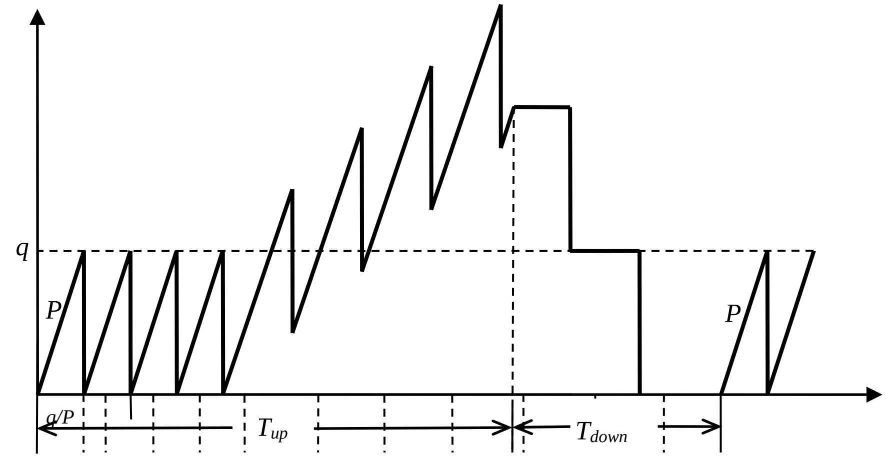
6.3 MODEL FORMULATION

In this Section, a three-factor CS model and a modified three-factor CS model are developed separately in Sections 6.3.1 and 6.3.2, respectively.

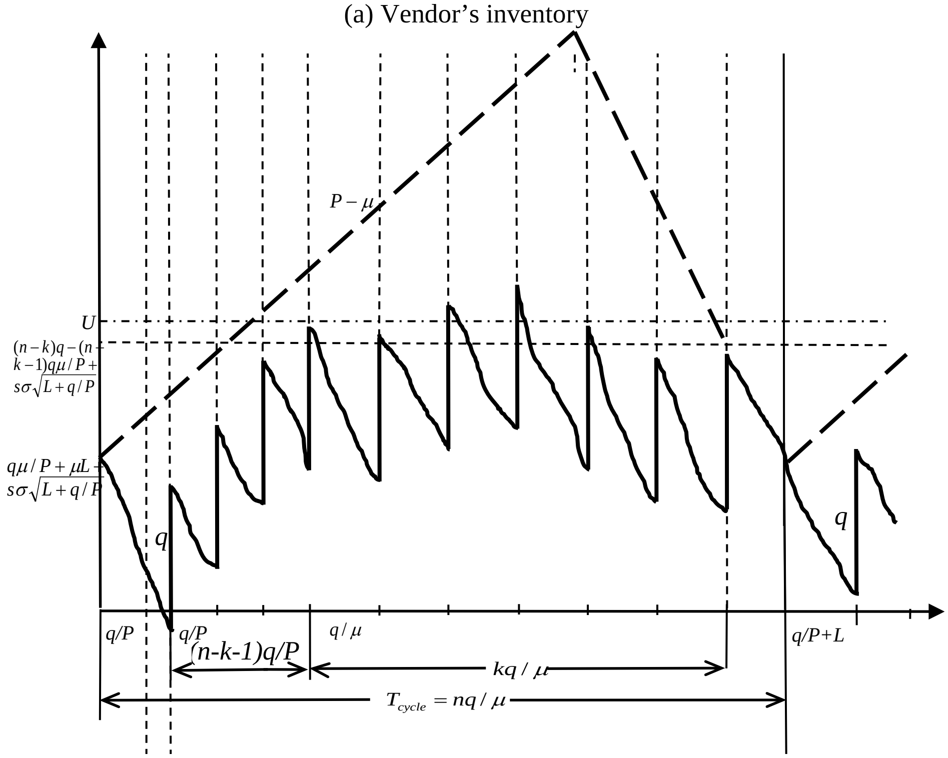
6.3.1 The Three-factor CS Model

Here, we develop a CS-*k* model considering three practical factors: controllable lead time (CLT), buyer's capacity limitation (BCL), and stochastic demand (SD). The pattern of the system inventory, the vendor inventory, the inventory in transit, and the buyer inventory of this model is shown on Figure 6.1.

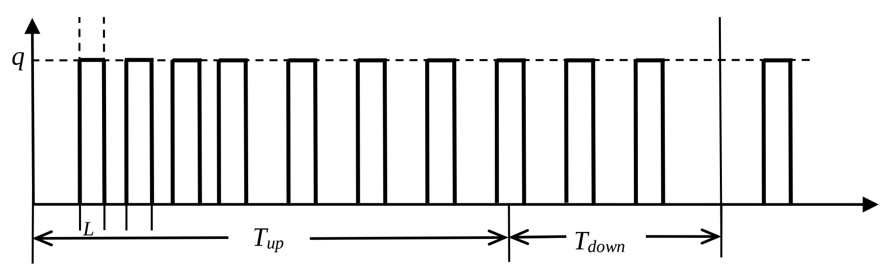
The vendor produces at a finite production rate P, and the annual demand D follows a normal distribution with mean μ and standard deviation σ. Under a long term CS agreement, the vendor satisfies the buyer's demand in each production cycle with n shipments of equal lot size q. In this model, the safety factor, s, is a decision variable used to decide the safety stock level, s_s, so as to mitigate the risk of stock outs due to the uncertain demand.



(a) Vendor's inventory



(b) System inventory (dashed) and buyer's inventory (solid)



(c) Inventory in transit

Figure 6.1 Composition of the inventory in the three-factor CS model

At the beginning of each production cycle, the buyer has an average initial inventory that equals to $\mu(L+q/P)+s\sigma\sqrt{L+q/P}$ which is left over from the last production cycle. The relationship between the safety factor and the safety stock level can be given by $s_s = s\sigma\sqrt{L+q/P}$. The first several shipments are made whenever the vendor's inventory reaches the level *q*. Each shipment reaches the buyer within a fixed length *L* that is reducible from the normal length L_0 with a "crushing" cost, which is a function of the reduced time $L - L_0$ and the ordering quantity *q*. Due to the uncertain nature of the demand, the lead time demand has a mean of μL and a standard deviation of $\sigma\sqrt{L}$ and the expected cycle length is $T_{cycle} = nq/\mu$. Meanwhile, the buyer places a space limitation *U* to the vendor. Therefore, when the buyer's inventory reaches the point $I_{\max}$ = $(n-k)q-(n-k-1)\ q\mu/P+s\sigma\sqrt{L+q/P} \le U$, all the following *k* shipments are delayed for a period such that the arrival of a new shipment always bring the buyer's average inventory up back to I_{max} (Figure 6.1b).

The unit holding cost *h* consists of a pure financial component, h^f, and a pure storage component, h^s. Under CS policy, all the financial component of holding costs is carried by the vendor which can be calculated by $h_v^f = rp_v$. There are three different unit holding costs with different physical locations of products: vendor unit holding cost, $h_v = h_v^s + h_v^f$, unit holding cost in transit, $h_d = h_d^s + h_v^f$, and buyer unit holding cost $h_b = h_b^s + h_v^f$.

Past research considered the lead time crushing cost to be a single function of the time period that has been reduced (Braglia and Zavanella, 2003; Huang and Chen, 2009).

However, in practice, the lead time cost often depends on the quantity being shipped. Therefore, we consider the composition of the lead time crushing cost different than others. In this system, the lead time has m mutually independent components, each with a minimum duration a_i and a maximum duration b_i, $i = 1, 2, \ldots, m$. The crushing cost per unit time c_{il} is an incremental function of both reduced lead time and shipping quantity, and is arranged such that $c_{il} \le c_{(i+1)l}$, and $c_{il} \ge c_{i(l+1)}$, $\forall i, l$, where $i = 1, 2, ..., m$, $l = 1, 2, ..., v$, and v denotes the total number of quantity discount segments. Within each quantity range, the lead time L is crushed one segment at a time starting with the least c_{ij}, and so on. Let L_0 denote the normal lead time before crushing, i.e., $L_0 = \sum_1^m b_j$ and L_i denote the lead time where the ith component was crushed to its minimum duration. Thus, the lead time L_i can be written as $\sum_{j=1}^{i} a_j + \sum_{j=i+1}^{m} b_j = \sum_{j=1}^{m} b_j - \sum_{j=1}^{i} (b_j - a_j) = L_0 - \sum_{j=1}^{i} (b_j - a_j)$, $i = 1, 2, ..., m$. The lead time crushing cost per replenishment cycle can thus be calculated by $R(q, L) = qc_{il}(L_{i-1} - L)$ $+ q\sum_{j=1}^{i-1} c_{jl}(b_j - a_j)$, $i = 1, 2, ..., m$, $l = 1, 2, ..., v$ for $q \in (q_{l-1}, q_l)$ and $L \in (L_i, L_{i-1})$.

The joint total expected cost of the system, *JTEC*, can be written as the sum of the vendor's expected setup cost, the buyer's expected ordering cost, the buyer's expected backorder cost, the buyer's expected extra space cost, the extra expected lead time crushing cost, and the expected system holding cost. The vendor's expected annual setup cost can be written as $A_v \mu / nq$. The buyer's expected annual ordering cost is given by $A_b \mu / q$. Also, the yearly expected lead time crushing cost can be written as $R(q, L)\mu / q = \mu c_{il}(L_{i-1} - L)$

$+\ \mu\sum_{j=1}^{i-1} c_{jl}(b_j - a_j)$, $i = 1, 2, ..., m$, $l = 1, 2, ..., r$ for $q \in (q_{l-1}, q_l)$ and $L \in (L_i, L_{i-1})$. The expected system holding cost consists of three parts: the expected holding cost incurred in the vendor's warehouse, the expected holding cost in transit, and the expected holding cost incurred in the buyer's warehouse.

According to Figure 6.1, the expected annual holding cost associated with the buyer's inventory can be written as

$$H_b = (h_b^s + h_v^f)\left[\frac{q\mu}{2P} + nq\frac{P-\mu}{2P} - q\frac{P-\mu}{nP}\frac{(k+1)k}{2} + s\sigma\sqrt{L + q/P}\right], \tag{6.1}$$

and the expected annual holding cost in transit is given by

$$H_d = (h_d^s + h_v^f)\mu L\,. \tag{6.2}$$

The annual holding cost at the vendor's warehouse is

$$H_v = (h_v^s + h_v^f)\left[\frac{q\mu}{2P} + q\frac{P-\mu}{nP}\frac{(k+1)k}{2}\right]. \tag{6.3}$$

The system annual holding cost can thus be computed as

$$\begin{aligned} H = H_v + H_d + H_b = (h_v^s + h_v^f)q\frac{\mu}{2P} + (h_b^s + h_v^f)\Bigg[q\frac{nP-(n-1)\mu}{2P} \\ + s\sigma\sqrt{L + \frac{q}{P}}\Bigg] + (h_v^s - h_b^s)q\frac{(P-\mu)(k+1)k}{2nP} + (h_d^s + h_v^f)\mu L\,. \end{aligned} \tag{6.4}$$

The expected annual backorder cost can be written as

$$\begin{aligned} C_b = \frac{\mu c_b}{nq}\Bigg\{E\left(X_1 - \frac{q\mu}{P} - \mu L - s\sigma\sqrt{L + \frac{q}{P}}\right)^+ + E\left(X_2 - q - s\sigma\sqrt{L + \frac{q}{P}}\right)^+ \\ + E\left(X_2 - 2q + \frac{q\mu}{P} - s\sigma\sqrt{L + \frac{q}{P}}\right)^+ + \ldots \end{aligned}$$

$$+E\left[X_2-(n-k-1)q+(n-k-2)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right]^+$$

$$+E\left[X_3-(n-k)q+(n-k-1)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right]^+k\Bigg\}, \tag{6.5}$$

where, the random variable X_1 is the demand during the first shipment, X_2 is the demand after the first shipment but before the first delayed shipment, and X_3 is the demand during the last k number of delayed shipments. Note that in the Equation (6.5), $E(\cdot)^+ = \max\{0, E(\cdot)\}$. As a result, its value could be zero or above zero under different situations, depending on the value of the random variable X_1, X_2, and X_3. Also note that the Equation (6.5) is complicated and the number of terms in it varies depending on the parameter values n and k. However, it cannot be further simplified by moving the constant portions inside the expectation brackets to the outside.

The expected annual extra space cost is given as

$$C_o=\frac{\mu k c_o}{nq}E\left[U-(n-k-1)q\left(1-\frac{\mu}{P}\right)-s\sigma\sqrt{L+\frac{q}{P}}-X_3\right]^+. \tag{6.6}$$

Hence, the annual joint total expected cost $JTEC(q,n,k,s,L)$ for given $L\in(L_i, L_{i-1})$ and $q\in(q_{l-1}, q_l)$ can be written as

$$JTEC(q,n,k,s,L)=(A_v+nA_b)\frac{\mu}{nq}+(h_v^s+h_v^f)\frac{q\mu}{2P}$$

$$+(h_b^s+h_v^f)\left[q\frac{nP-(n-1)\mu}{2P}+s\sigma\sqrt{L+\frac{q}{P}}\right]$$

$$+(h_v^s-h_b^s)q\frac{(P-\mu)(k+1)k}{2nP}+(h_d^s+h_v^f)\mu L+\frac{\mu}{q}R(q,L)$$

$$+\frac{\mu c_b}{nq}\left\{E\left(X_1-\frac{q\mu}{P}-\mu L-s\sigma\sqrt{L+\frac{q}{P}}\right)^+ +E\left(X_2-q-s\sigma\sqrt{L+\frac{q}{P}}\right)^+\right.$$

$$+E\left(X_2-2q+\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right)^+ + \ldots$$

$$+E\left[X_2-(n-k-1)q+(n-k-2)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right]^+$$

$$\left.+E\left[X_3-(n-k)q+(n-k-1)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right]^+ k\right\}$$

$$+\frac{\mu k c_o}{nq}E\left[U-(n-k-1)q\left(1-\frac{\mu}{P}\right)-s\sigma\sqrt{L+\frac{q}{P}}-X_3\right]. \tag{6.7}$$

Model 6.1: (Three-factor CS model)

The three-factor CS model can, therefore, be written as:

Min $JTEC(q,n,k,s,L)$ (6.8)

Subject to: $(n-k)q-(n-k-1)q\mu/P+s\sigma\sqrt{L+q/P}\le U$, (6.8a)

$k\le n$, (6.8b)

q, n, k, L are positive and integers. (6.8c)

The problem is to jointly decide the optimal ordering quantity q, the safety factor s, the number of shipments within a production cycle n, the number of delayed shipments k, and the lead time L, that minimize the *JTEC* as expressed by Equation (6.8), under the constraints in (8a), (8b) and (8c).

6.3.2 The modified Three-factor CS Model

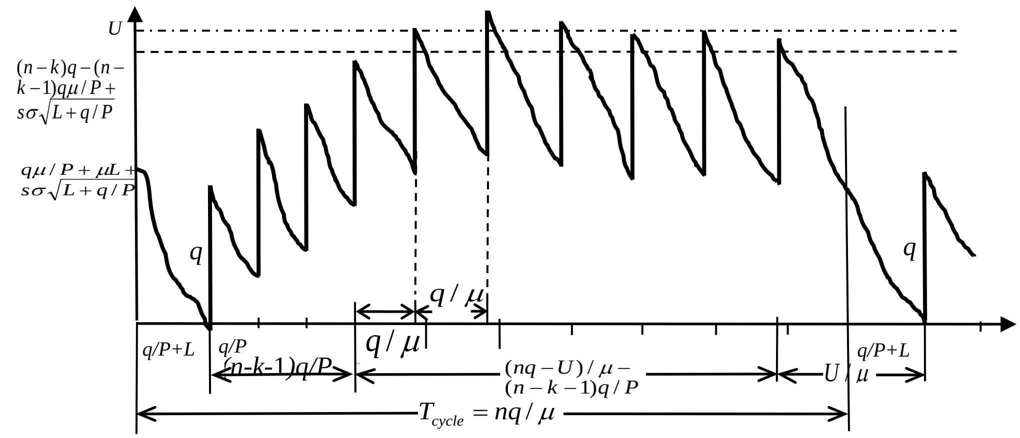
The Three-factor CS model does not make the most use of the buyer's capacity because, once the buyer's inventory reaches I_{max}, which may be less than the space limitation U, the policy always brings the buyer's inventory back to the level of I_{max}. It is, thus, possible to modify this policy by trying to bring the buyer's inventory up to the space limitation U. As a result, the buyer's average inventory increases while the vendor's decreases. This modification might sometime be favorable to the system, especially when the buyer's unit holding cost is less than that of the vendor's. Therefore, a modified three-factor CS model is developed in this Section.

The pattern of the system inventory, the vendor inventory, the inventory in transit, and the buyer inventory of the modified three-factor CS model is shown in Figure 6.2. In the modified model, when the buyer inventory reaches the point that the new arrival of another shipment will result in the exceeding of the buyer's space limitation, the ensuing shipments are delayed so that the arrival of each shipment brings the buyer expected inventory to the space limitation U instead of to the maximum inventory level I_{max}.

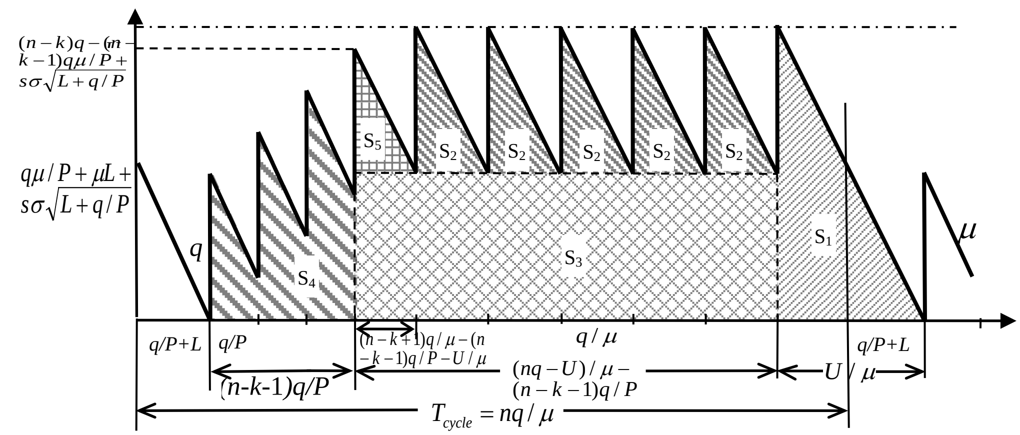
According to Figure 6.2, the buyer's average inventory I_b can be calculated as the shaded area S (Figure 6.2b) divided the cycle time T_{cycle}. That is,

$$I_b = S / T_{cycle} = \left[\frac{n^2q^2 + k^2q^2 - 2nkq^2 - kq^2 + 2ukq + s\sigma\sqrt{L + q/P}(nq - kq + q - U)}{2\mu} \right.$$

$$\left. - \frac{(n-k)(n-k-1)q^2 - qs\sigma\sqrt{L+q/P}(n-k-1)}{2P} \right] \frac{\mu}{nq}$$



(a) Buyer's inventory for the modified model



(b) Average buyer's inventory (shaded area)

Figure 6.2 Buyer's inventory in the modified three-factor CS model

$$= \frac{n^2 q + k^2 q - 2nkq - kq + 2Uk + s\sigma\sqrt{L + q/P}(n - k + 1 - U/q)}{2n}$$

$$- \frac{(n-k)(n-k-1)q\mu - \mu s\sigma\sqrt{L + q/P}(n-k-1)}{2nP}.$$

The calculation of S is given in Appendix A-6. The average system inventory I_s can be calculated as $I_s = \frac{nq(P-\mu)}{2P} + \frac{q\mu}{P} + \mu L + s\sigma\sqrt{L + q/P}$ while the average inventory in transit is $I_d = \mu L$.

The average vendor inventory I_v can also be given as

$$I_v = I_s - I_d - I_b = \frac{nq(P-\mu)}{2P} + \frac{q\mu}{P} + s\sigma\sqrt{L+q/P}$$

$$-\frac{n^2 q + k^2 q - 2nkq - kq + 2Uk + s\sigma\sqrt{L+q/P}(n-k+1-U/q)}{2n}$$

$$+\frac{(n-k)(n-k-1)q\mu - \mu s\sigma\sqrt{L+q/P}(n-k-1)}{2nP}$$

$$= \frac{2nkq - k^2 q + kq - 2Uk + s\sigma\sqrt{L+q/P}(n+k-1+U/q)}{2n}$$

$$+\frac{(k^2+k+n-2nk)q\mu - \mu s\sigma\sqrt{L+q/P}(n-k-1)}{2nP}.$$

Hence, the annual holding cost occurred at the buyer warehouse is given by

$$H_b = (h_b^s + h_v^f)\left[\frac{n^2 q + k^2 q - 2nkq - kq + 2Uk + s\sigma\sqrt{L+q/P}(n-k+1-U/q)}{2n}\right.$$

$$\left.-\frac{(n-k)(n-k-1)q\mu - \mu s\sigma\sqrt{L+q/P}(n-k-1)}{2nP}\right]. \tag{6.9}$$

The annual holding cost in transit is

$$H_d = (h_d^s + h_v^f)\mu L. \tag{6.10}$$

The annual holding cost occurred at the vendor warehouse can be written as

$$H_v = (h_v^s + h_v^f)\left[\frac{2nkq - k^2 q + kq - 2Uk + s\sigma\sqrt{L+q/P}(n+k-1+U/q)}{2n}\right.$$

$$\left.+\frac{(k^2+k+n-2nk)q\mu - \mu s\sigma\sqrt{L+q/P}(n-k-1)}{2nP}\right]. \tag{6.11}$$

Therefore, the system annual holding cost is computed as

$$H = H_v + H_d + H_b = (h_v^s + h_v^f)\left[\frac{-k^2 q + 2nkq + kq - 2Uk - s\sigma\sqrt{L+q/P}(-n-k+1-U/q)}{2n}\right.$$

$$+\frac{(k^2+k+n-2nk)q\mu-\mu s\sigma\sqrt{L+q/P}(n-k-1)}{2nP}\Bigg]$$

$$+(h_b^s+h_v^f)\left[\frac{n^2q+k^2q-2nkq-kq+2Uk+s\sigma\sqrt{L+q/P}(n-k+1-U/q)}{2n}\right.$$

$$\left.-\frac{(n-k)(n-k-1)q\mu-\mu s\sigma\sqrt{L+q/P}(n-k-1)}{2nP}\right]+(h_d^s+h_v^f)\mu L.$$

Upon simplification, it can be rewritten as

$$H=(h_v^s+h_v^f)\left(\frac{q\mu}{2P}+s\sigma\sqrt{L+\frac{q}{P}}\right)$$

$$+(h_v^s-h_b^s)\left\{\frac{[2nqk-k^2q-kq-s\sigma\sqrt{L+q/P}(n+k-1)](P-\mu)}{2nP}\right.$$

$$\left.+\frac{2qk-2Uk+s\sigma\sqrt{L+q/P}(U/P)}{2n}\right\}$$

$$+(h_b^s+h_v^f)q\left[\frac{n}{2}-\frac{(n-1)D}{2P}\right]+(h_d^s+h_v^f)\mu L. \tag{6.12}$$

The expected annual backorder cost can be written as

$$C_b=\frac{\mu c_b}{nq}\left\{E\left(X_1-\frac{q\mu}{P}-\mu L-s\sigma\sqrt{L+\frac{q}{P}}\right)^+ +E\left(X_2-q-s\sigma\sqrt{L+\frac{q}{P}}\right)^+\right.$$

$$+E\left(X_2-2q+\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right)^+ +\ldots+E\left[X_2-(n-k-1)q+(n-k-2)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right]^+$$

$$\left.+E[X_3-U]^+k\right\}. \tag{6.13}$$

It can be shown that the expected annual extra space cost is

$$C_o=\frac{\mu kc_o}{nq}E[q-X_3]^+. \tag{6.14}$$

Model 6.2: (Modified Three-factor CS model)

Therefore, the *JTEC* of the modified three-factor CS model can be written as

$$\text{Min } JTEC(q,n,k,s,L) = (A_v + nA_b)\frac{\mu}{nq} + (h_v^s + h_v^f)\left(\frac{q\mu}{2P} + s\sigma\sqrt{L+\frac{q}{P}}\right)$$

$$+ (h_v^s - h_b^s)\left\{\frac{[2nqk - k^2q - kq - s\sigma\sqrt{L+q/P}(n+k-1)](P-\mu)}{2nP}\right.$$

$$\left. + \frac{2qk - 2Uk + s\sigma\sqrt{L+q/P}(U/P)}{2n}\right\} + (h_b^s + h_v^f)q\left[\frac{n}{2} - \frac{(n-1)\mu}{2P}\right] + (h_d^s + h_v^f)\mu L$$

$$+ \frac{\mu c_b}{nq}\left\{E\left(X_1 - \frac{q\mu}{P} - \mu L - s\sigma\sqrt{L+\frac{q}{P}}\right)^+ + E\left(X_2 - q - s\sigma\sqrt{L+\frac{q}{P}}\right)^+\right.$$

$$+ E\left(X_2 - 2q + \frac{q\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}}\right)^+ + \ldots + E\left[X_2 - (n-k-1)q + (n-k-2)\frac{q\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}}\right]^+$$

$$\left. + E[X_3 - U]^+ k\right\} + \frac{\mu k c_o}{nq}E[q - X_3]^+ + \frac{\mu}{q}R(q,L) \tag{6.15}$$

Subject to: $$(n-k)q - (n-k-1)q\mu/P \le U\,, \tag{6.15a}$$

$$k \le n\,, \tag{6.15b}$$

q, n, and k are positive and integers. (6.15c)

L is positive integer values (in days) (6.15d)

In the developed two models, the calculation of the system cost requires the calculation of the expected backorder cost and the expected extra space cost, which, in turn, relies on the specific distribution of the demand rate. In this study, a normally distributed demand is used to demonstrate the generic solution methodology that can be applied for any type of demand distributions. Here the yearly demand rate follows a normal distribution with mean

μ and standard deviation σ. It follows that the lead time demand also follows normal distribution with mean μL and standard deviation $\sigma\sqrt{L}$. Hence, the random variable *X1*, *X2*, *X3* (see Figures 6.1 and Figure 6.2) are all normally distributed and satisfy $X_1 \sim N(\mu_{X_1}, \sigma_{X_1})$, $X_2 \sim N(\mu_{X_2}, \sigma_{X_2})$, and $X_3 \sim N(\mu_{X_3}, \sigma_{X_3})$, where $\mu_{X_1} = \mu(q/P + L)$, $\mu_{X_2} = \mu q/P$, $\mu_{X_3} = q$, and $\sigma_{X_1} = \sigma\sqrt{q/P+L}$, $\sigma_{X_2} = \sigma\sqrt{q/P}$, $\sigma_{X_1} = \sigma\sqrt{q/\mu}$, respectively.

It can be shown that, when a random variable *X* follows a normal distribution with mean μ and standard deviation σ, i.e., $X \sim N(\mu, \sigma)$, then $E(X-r)^+$ is the loss function $\bar{b}(r)$ and can be expressed by $E(X-r)^+ = \bar{b}(r) = \sigma[v\Phi(v) + \phi(v) - v]$ (See Appendix A-7), where $v = (r-\mu)/\sigma$, $\phi(\cdot)$ is the probability density function (pdf) of the standard normal distribution, and $\Phi(\cdot)$ is the cumulative density function (cdf) of the standard normal distribution. Similarly, it can be shown that $E(r-X)^+ = \sigma[v\Phi(v) + \phi(v)]$. Therefore, the expected annual backorder cost for the three-factor CS model (Equation 6.5) can be rewritten as

$$
\begin{aligned}
C_b = \frac{\mu c_b}{nq}\Bigg\{ & \sigma\sqrt{\frac{q}{P}+L}\cdot[s\cdot\Phi(s)+\phi(s)-s] + \sigma\sqrt{\frac{q}{P}}\cdot\Bigg[\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}} \\
& \cdot\Phi\left(\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right) + \phi\left(\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right) \\
& - \frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}} + \frac{2q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}} \\
& \cdot\Phi\left(\frac{2q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right) + \phi\left(\frac{2q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)
\end{aligned}
$$

$$-\frac{2q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}+\ldots+\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}$$

$$\cdot\Phi\left(\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)+\phi\left(\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)$$

$$\left.-\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right]+k\sigma\sqrt{\frac{q}{\mu}}\left[\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right.$$

$$\cdot\Phi\left(\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right)+\phi\left(\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right)$$

$$\left.\left.-\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right]\right\}, \tag{6.16}$$

and the expected annual extra space cost for the three-factor CS model [Equation (6.6)] can be rewritten as

$$C_o=\frac{\mu k c_o}{nq}\left[\frac{U-(n-k)q+(n-k-1)q\mu/P-s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right.$$

$$\cdot\Phi\left(\frac{U-(n-k)q+(n-k-1)q\mu/P-s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right)$$

$$\left.+\phi\left(\frac{U-(n-k)q+(n-k-1)q\mu/P-s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/\mu}}\right)\right]. \tag{6.17}$$

Similarly, the expected annual backorder cost for the modified three-factor CS model [Equation (6.13)] can be rewritten as

$$C_b=\frac{\mu c_b}{nq}\left\{\sigma\sqrt{\frac{q}{P}+L}\cdot[s\cdot\Phi(s)+\phi(s)-s]+\sigma\sqrt{\frac{q}{P}}\cdot\left[\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right.\right.$$

$$\cdot\Phi\left(\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)+\phi\left(\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)$$

$$-\frac{q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}+\frac{2q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}$$

$$\cdot\Phi\left(\frac{2q(1-D/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)+\phi\left(\frac{2q(1-D/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)$$

$$-\frac{2q(1-D/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}+\ldots+\frac{(n-k-1)q(1-D/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}$$

$$\cdot\Phi\left(\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)+\phi\left(\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right)$$

$$-\left.\frac{(n-k-1)q(1-\mu/P)+s\sigma\sqrt{L+q/P}}{\sigma\sqrt{q/P}}\right]+k\sigma\sqrt{\frac{q}{\mu}}\left[\frac{U-q}{\sigma\sqrt{q/\mu}}\cdot\Phi\left(\frac{U-q}{\sigma\sqrt{q/\mu}}\right)\right.$$

$$\left.\left.+\phi\left(\frac{U-q}{\sigma\sqrt{q/\mu}}\right)-\frac{U-q}{\sigma\sqrt{q/\mu}}\right]\right\}, \tag{6.18}$$

and the extra space cost for the modified three-factor CS model [Equation (6.14)] can be rewritten as

$$C_o=\frac{\mu kc_o}{nq}\phi(0). \tag{6.19}$$

With Equations (6.16), (6.17), (6.18), and (6.19), the annual system cost can be evaluated.

6.3.3 Problems of optimality

The three-factor CS model and the modified three-factor CS models developed in this Chapter are both constrained nonlinear mixed-integer optimization problems. Unlike most other inventory models which only consider two or three decision variables, these two models contain five decision variables. Moreover, one of the decision variables, the demand rate *D*, is stochastic and follows a distribution that can be decided upon historical data. Due

to the number of variables in the model, it is difficult to prove the convexity of the objective function. Also, due to the stochastic nature of the decision variable, traditional optimization methods that rely on differentiation may not be appropriate to solve the two models. Therefore, apparently there is no closed-form solution for the models in consideration.

In this research, an exhaustive search algorithm (ESA) is adopted at first in Section 6.4 to locate the global optimal solutions of an instance of the problem. Since the ESA is time consuming, it is necessary to find another suitable general solution procedure. In this Chapter, the two doubly-hybrid meta-heuristic algorithms (DHMHA) by Yi, *et al*. (2013) are used to provide a solution procedure to the problems. The computation results show that both the DHMHA and the ESA yield the same solutions (See Section 6.4.1 for details) but the DHMHA takes less time. Because of efficiency, the two DHMHAs are adopted later in Section 6.5 to perform the analysis of the effects of some important parameters to the system.

6.4 COMPUTATIONAL RESULTS

A numerical example is framed here to illustrate the optimal solutions (ESA) along with other heuristics mentioned earlier for models 6.1 and 6.2.

Example 6.1: Illustration of the doubly-hybrid solution procedure

Most of the values of the parameters of this example are adopted from Braglia and Zavanella (2003), and Huang and Chen (2009). The composition of the lead time is one of such parameters and is shown on Table 4.2.

As mentioned earlier in Section 6.2.1, the unit lead time crushing cost is a function of both the reduced period and the shipping size. In Example 6.1, a unified quantity discount is used to represent the relationship between the lead time crushing cost and the quantity. Table 6.1 illustrates the detailed composition of this cost.

The values of other parameters are: $A_v = \$400$ /setup, $A_b = \$25$ /order, $\mu = 1000$ units/year, $\sigma = 100$ units, $P = 3200$ units/year, $r = 10\%$, p_v \$20/unit, $h_v^s = \$3$/unit/year, $h_b^s = \$1.50$/unit/year, $h_d^s = \$4$/unit/year, c_o = \$10/unit, c_b = \$50/unit, and U = 150 units.

Table 6.1 The composition of the unit lead time crushing cost c_{ij}

	$q \geq 100$	$100 > q \geq 20$	$q < 20$
Lead time component i	c_{i1}(\$/unit/year)	c_{i2}(\$/unit/year)	c_{i3}(\$/unit/year)
1	(0.8)(0.1)(365)=29.2	(0.9)(0.1)(365)=32.85	(1)(0.1)(365)=36.5
2	(0.8)(1.2)(365)=350.4	(0.9)(1.2)(365)=394.2	(1)(1.2)(365)=438
3	(0.8)(5.0)(365)=1,460	(0.9)(5.0)(365)=1,642.5	(1)(5.0)(365)=1,825

In the models developed, all other decision variables are integers except for the safety factor *s*, which is continuous. Two decimal values are allowed for *s* so that the ESA can be used to find the optimal solutions and verify whether the solutions found by the two Doubly-hybrid Meta-heuristic Methods are global optimal. Table 6.2 shows the optimal solutions of the decision variables and expected system cost found by the six algorithms for the two models.

It is observed that all the six algorithms (including ESA that gives the optimal solution exhaustively) lead to the same solutions, which is guaranteed to be the global optimum

since the ESA is used as a comparison. While the ESA takes more than 30 days to find the optimal solutions (exhaustively), all the other approaches take less than four minutes. Although the meta-heuristic approaches seem to be more efficient (faster) than ESA, the accuracy (success rate) of the MDE', IHS, and PSO are not as satisfactory.

Table 6.2 Optimal solutions of the six algorithms

Ex6.1	Algorithm	*JTEC* ($/yr)	*n*	*q*	*k*	*L* *(day)*	*s*	CPU/ M_CPU (s)	NFE/M_NFE	Suc. Rate
Model 6.1	ESA	4,480.22	5	85	4	56	1.53	2,653,545	1,084,343,544	-
	MDE'	4,480.22	5	85	4	56	1.53	238.07	300,000	0.80
	IHS	4,480.22	5	85	4	56	1.53	120.87	300,000	0.10
	PSO	4,480.22	5	85	4	56	1.53	6.22	300,000	0.50
	PSO+IHS+HJ	4,480.22	5	85	4	56	1.53	137.25	300,127	1.00
	MDE'+IHS+HJ	4,480.22	5	85	4	56	1.53	163.36	300,066	1.00
Model 6.2	ESA	4,441.71	5	85	4	56	1.53	2,639,145	1,084,343,544	-
	MDE'	4,441.71	5	85	4	56	1.53	224.39	300,000	0.90
	IHS	4,441.71	5	85	4	56	1.53	82.81	300,000	0.80
	PSO	4,441.71	5	85	4	56	1.53	5.48	300,000	1.00
	PSO+IHS+HJ	4,441.71	5	85	4	56	1.53	132.76	300,102	1.00
	MDE'+IHS+HJ	4,441.71	5	85	4	56	1.53	139.83	300,073	1.00

Of the three algorithms (MDE', IHS, and PSO), the one that forms the base of the hybrid-algorithm, the MDE' takes the longest mean CPU time (nearly four minutes) and the PSO is the quickest methods (a few seconds). However, the success rate of IHS and PSO is not good enough for the first model (0.1 and 0.5, respectively). On the contrary, the performance of MDE' is relatively more consistent for both the models (0.8 and 0.9, respectively). Compared to that, the two hybrid-algorithms (PSO+IHS+HJ and MDE'+IHS+HJ) requires only minutes with a satisfactory success rate (100% for both models). Therefore, the two doubly-hybrid methods are more accurate than the three meta-

heuristic methods alone. In order to further compare the time of the two doubly-hybrid algorithms and the ESA, a graphical representation is shown in Figure 6.3 as to how these three algorithms approach the global optimum. It can be seen that the MDE'+IHS+HJ reaches the global optimum within 30 seconds and the IHS+PSO+HJ requires less than 90 seconds. Compared to that, the ESA does not reach the global optimum within the 30 minutes of run time. This result shows that the accuracy (success rate) of both the hybrid algorithms is as satisfactory as that of the ESA, but the efficiency (time) of the doubly-hybrid algorithms are superior. The efficiencies of the hybrids are comparable to but more accurate than those of the three individual meta-heuristic algorithms. That is, the two doubly-hybrid approaches are superior to the other four for the two models.

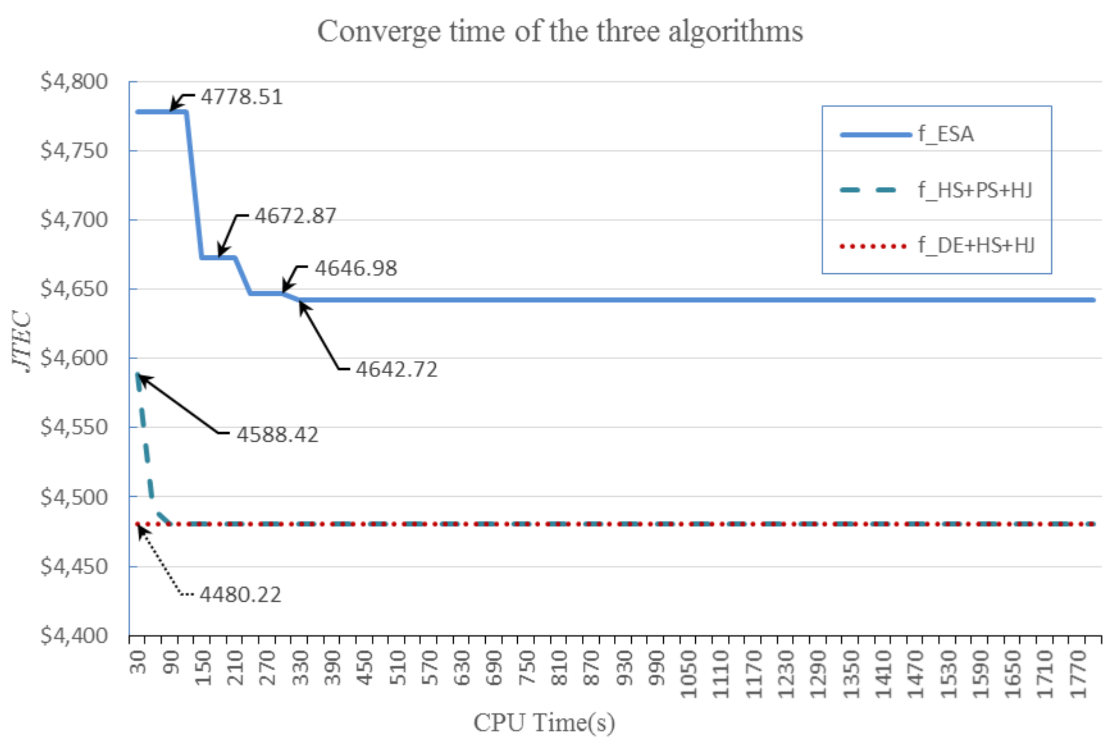


Figure 6.3 Converge time of the ESA and the tow doubly-hybrid methods

Due to the fact that the ESA is time consuming and the performances of the three individual meta-heuristic algorithms are not satisfactory enough, only the two doubly-hybrid approaches are used in the remainder of this Section. Meanwhile, preliminary testing computation shows that increasing the precision of the results of the hybrid methods by allowing more decimal digits to s does not increase the computation time to the two hybrid algorithms. Therefore, four decimal values are from now on allowed to the safety factor s to make the results more precise.

6.5 SENSITIVITY ANALYSIS

There are several important parameters in the models, such as the buyer space limitation U, the unit penalty for backorders c_b, the unit penalty for extra space c_o, and the quantity discount coefficient c_i. A change in their values might influence the optimal solutions. The combination of the values of h_v^s and h_b^s is also important, previous research has shown that $h_v^s > h_b^s$ or $h_v^s < h_b^s$ may bring different solutions. We therefore discuss the effects of these important parameters to the system in this Section.

6.5.1 Effects of buyer space limitation *U*

The buyer space limitation U is one of the most influential parameters of the entire system. All operational decisions and system cost will change according to different values of U. Table 6.3 illustrates the results of both hybrid algorithms for the two developed models with the value of U changing from 50 to 500. It can be seen that the overall performance of both the hybrid algorithms are satisfactory. They both have satisfactory

success rate and convergent to the optimum within a relatively short period. The solutions of both algorithms agree with each other. Compared to the PSO+IHS+HJ hybrid algorithm, the MDE'+IHS+HJ is more efficient. Comparing the solutions of the rows where *u* is 150 in Table 6.3 to that of Table 6.2, it is apparent that, with two more decimal values allowed to the safety factor *s*, the *JTEC* improved. More decimal digits to the *s* are not necessary since more testing computation shows little improving in the *JTEC*.

Table 6.3 Effects of *U* to the system

	Optimal Solutions						PSO+IHS+HJ		MDE'+IHS+HJ	
U	*JTEC*($)	*n*	*q*	*K*	*L* *(day)*	*s*	Mean CPU(s)	Success Rate	Mean CPU(s)	Success Rate
Model I										
50	7,256.59	26	22	26	56	1.0771	121.98	1.00	131.79	0.90
100	5,211.54	10	51	9	56	1.1906	193.04	1.00	136.75	1.00
150	**4,475.13**	**5**	**85**	**4**	**56**	**1.5321**	**276.08**	**0.50**	**148.29**	**1.00**
200	4,110.18	3	125	2	56	1.7094	276.62	0.70	133.68	0.90
250	3,765.77	3	109	1	56	1.5257	450.60	0.30	139.60	0.90
300	3,429.15	4	76	0	56	1.5976	631.21	1.00	144.47	0.70
350	3,217.16	4	91	0	56	1.6722	558.41	0.90	230.78	0.70
400	3,097.56	4	107	0	56	1.6728	517.74	0.60	215.09	0.90
450	3,035.03	4	122	0	56	1.7450	649.90	0.80	244.71	0.90
500	3,007.87	5	113	0	56	1.7551	443.17	0.90	273.35	0.90
Total	-	-	-	-	-	-	**4118.75**	**7.70**	**1798.52**	**8.80**
Model II										
50	7,212.79	26	22	26	56	1.0771	76.03	1.00	125.39	1.00
100	5,173.27	10	51	9	56	1.1906	101.77	1.00	134.98	1.00
150	**4,441.42**	**5**	**85**	**4**	**56**	**1.5321**	**300.10**	**1.00**	**138.59**	**1.00**
200	4,094.55	4	119	3	56	1.7467	362.13	1.00	147.34	1.00
250	3,798.50	3	110	1	56	1.4854	277.63	0.70	159.62	0.60
300	3,503.92	4	77	0	56	1.5235	409.76	1.00	199.30	0.50
350	3,295.53	4	92	0	56	1.5990	465.90	0.90	201.83	0.90
400	3,178.44	4	108	0	56	1.6006	457.11	1.00	210.88	0.80
450	3,115.82	5	102	0	56	1.5680	484.11	0.80	225.67	0.50
500	3,087.33	5	115	0	56	1.5798	538.75	1.00	230.74	0.90
Total	-	-	-	-	-	-	**3473.29**	**9.40**	**1774.34**	**8.20**

The *JTEC* decreases as the space limitation, *U*, increases, as expected. It is interesting, however, that the *JTEC*s of the two models do not decrease at the same rate with respective to *U*. In fact, the modified model performs better when the space limitation is tight. Note that in Table 6.3, when *U* is between 50 and 150, the modified model is superior. On the contrary, the original model performs better when *U* is greater than or equal to 200.

The differences in performance of the two models can possibly be explained by the tradeoff of putting more inventories in the buyer's place, which is exactly a modification of the modified model to the original one. Under a deterministic background, this modification would have not brought any extra charge to the system. Therefore, when the unit holding cost is lower to the buyer, the modified model performs better. However, with a stochastic demand, there are both shortage cost and extra space cost in the system. A higher level of inventory in the buyer's warehouse decreases, on one hand, the possibility of stock out which, in turn, decreases the expected stock out penalty. It also, conversely, increases the probability of violating the space limitation which, in turn, increases the expected over space penalty. How does the tradeoff affect the system is an interesting issue that remains unclear and needs further study. The useful conclusion we could draw from this example is that the introduction of uncertainty of demand can lead to a different operational decisions. The developed models, together with the developed hybrid algorithms, as shown in this example, can provide the decision makers a quantitative tool to assist in making the right choice of replenishment policies and to protect them from the demand variations.

6.5.2 Effects of the back order penalty coefficient c_b

Another important parameter in the system is the back order penalty coefficient c_b. A change to c_b will directly change the *JTEC* and the safety factor s, and all other decision variables will vary accordingly. In Table 6.4, we show the effects of c_b to the system.

Table 6.4 Effects of c_b to the system

	Optimal Solutions						PSO+IHS+HJ		MDE'+IHS+HJ	
c_b	*JTEC*($)	*n*	*q*	*k*	*L* *(day)*	*s*	Mean CPU(s)	Success Rate	Mean CPU(s)	Success Rate
Model I										
5	3,863.33	4	88	2	56	0.0352	157.90	1.00	180.82	0.60
10	4,062.82	5	87	3	56	0.0750	184.30	0.60	271.67	0.70
15	4,213.00	6	79	4	56	0.3954	180.28	0.40	247.19	0.60
20	4,303.48	4	98	3	56	1.2120	170.43	0.70	223.55	0.70
25	4,346.13	5	93	4	56	1.3343	166.22	0.90	216.18	1.00
30	4,377.74	5	91	4	56	1.3835	169.57	0.90	213.43	1.00
35	4,405.73	5	89	4	56	1.4328	166.79	0.90	231.32	1.00
40	4,430.84	5	88	4	56	1.4576	164.15	0.80	214.20	0.90
45	4,453.89	5	87	4	56	1.4824	162.36	1.00	210.92	1.00
50	**4,475.13**	**5**	**85**	**4**	**56**	**1.5321**	**140.22**	**0.90**	**213.46**	**1.00**
Total	-	-	-	-	-	-	**1662.21**	**8.10**	**2222.74**	**8.50**
Model II										
5	3,863.42	4	88	2	56	0.0352	152.36	1.00	145.88	0.80
10	4,062.45	5	87	3	56	0.0750	163.34	0.90	132.37	0.30
15	4,208.52	6	79	4	56	0.3954	143.13	0.10	129.65	0.10
20	4,280.46	5	94	4	56	1.2914	150.02	1.00	129.03	1.00
25	4,316.08	5	92	4	56	1.3588	138.16	1.00	128.74	1.00
30	4,346.77	5	90	4	56	1.4081	136.59	1.00	132.20	1.00
35	4,373.98	5	89	4	56	1.4328	147.39	1.00	128.69	1.00
40	4,398.41	5	87	4	56	1.4824	158.06	1.00	127.83	1.00
45	4,420.74	5	86	4	56	1.5072	150.47	1.00	129.05	1.00
50	**4,441.42**	**5**	**85**	**4**	**56**	**1.5321**	**138.18**	**1.00**	**131.27**	**1.00**
Total	-	-	-	-	-	-	**1477.70**	**9.00**	**1314.70**	**8.20**

Table 6.4 shows the effects of c_b on all decision variables and on the *JTEC*. The values of other variables are relatively stable or without a certain pattern, whereas that of the safety

factor s and the *JTEC* have a pattern. Figure 6.4 illustrates the relation between s, *JTEC*, and c_b. It can be seen that both *JTEC* and s increase as c_b increases. However, the *JTEC* increases rapidly when c_b is from 5 to 20. After that, the increasing rate seems to be stabilized. Moreover, the increasing rates of the two models are not similar. In the beginning, the *JTEC* of both models are approximately the same and they also increase at approximately the same rate, starting from c_b equals 6.15, the increasing rate of the modified model is apparently slower than that of the original model. As to the safety factor s, both models follow a similar increasing pattern with the modified model having a slightly higher safety factor. Specifically, the s increases rapidly and at a nearly exponential rate when c_b is between 10 and 20. On the contrary, it increases at a nearly linear rate when c_b is outside that range. In short, through this example, it is clear that the two models are more or less the same when c_b is less than or equals to 15. With c_b greater than 15, the modified model is slightly better. This analysis also helps to find out the range of c_b within which the system is sensitive.

Figure 6.4 illustrates the optimized *JTEC* and the corresponding optimal service level s at given values of the back order penalty c_b. In practice, however, it is difficult for the decision makers to directly manage the safety factor. Instead, it is easier to manage the safety stock level s_s. Therefore, it is necessary to demonstrate how the total cost *JTEC*, the safety stock level s_s, and the ordering quantity q vary with the back order penalty c_b. Figure 6.5 illustrates this relationship, wherein the safety stock level is calculated by

$s_s = s\sigma\sqrt{L + q/P}$. Note that Figure 6.5 does not provide more information of the system than does Figure 6.4. It is actually a translated version of Figure 6.4 so as to assist future users of the three-factor CS model and the modified Three-factor CS model to determine the optimal safety stock level in consignment agreement between suppliers and buyers. Therefore, all findings through Figure 6.4 that were mentioned in the last paragraph also apply to Figure 6.5.

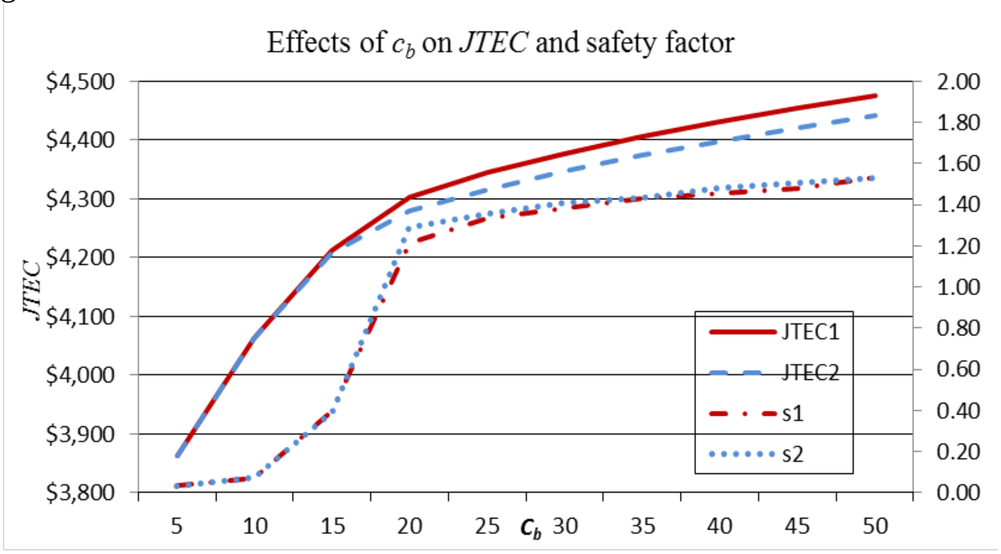


Figure 6.4 Effects of c_b on s and *JTEC*

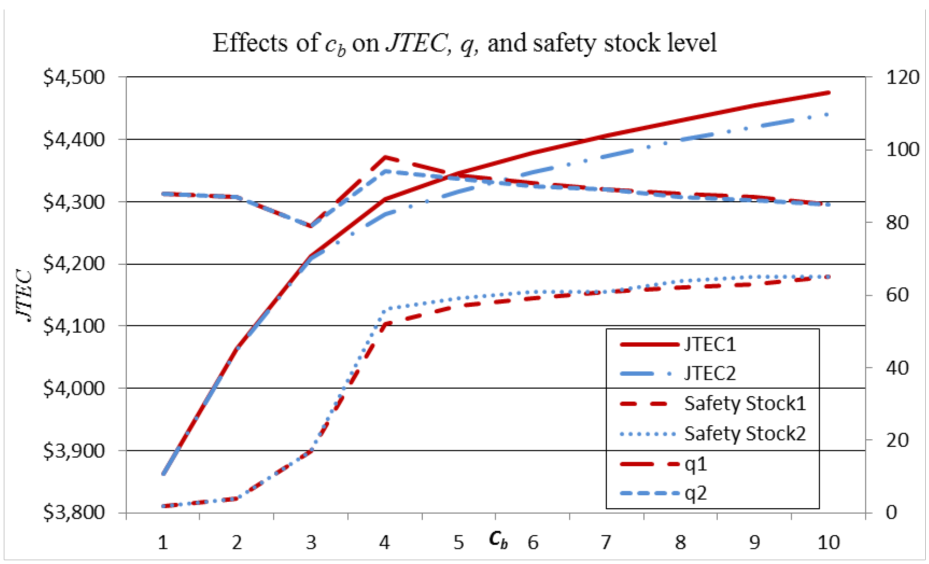


Figure 6.5 Effects of c_b on *JTEC*, safety stock, and ordering quantity

6.5.3 Effects of the extra space penalty coefficient c_o

The extra space penalty coefficient c_o is as important as c_b. The study of its effects to the system is useful and is illustrated by Table 6.5.

Table 6.5 Effects of c_o to the system

	Optimal Solutions						PSO+IHS+HJ		MDE'+IHS+HJ	
c_o	*JTEC*($)	*n*	*q*	*k*	*L* (*day*)	*s*	Mean CPU(s)	Success Rate	Mean CPU(s)	Success Rate
Model I										
5	3,909.71	6	81	5	56	1.6320	159.32	1.00	171.63	1.00
10	4,475.13	5	85	4	56	1.5321	201.88	0.90	157.12	0.90
15	5,017.75	5	87	4	56	1.4824	218.09	0.80	144.08	0.90
20	5,525.42	4	91	3	56	1.3835	205.50	1.00	138.03	1.00
25	5,923.09	3	43	0	56	1.1720	359.24	1.00	155.57	0.40
30	5,923.09	3	43	0	56	1.1720	275.53	1.00	178.61	0.80
35	5,923.09	3	43	0	56	1.1720	269.04	1.00	187.25	0.60
40	5,923.09	3	43	0	56	1.1720	274.70	1.00	208.00	0.40
45	5,923.09	3	43	0	56	1.1720	248.87	1.00	180.07	0.60
50	5,923.09	3	43	0	56	1.1720	209.16	1.00	178.96	0.70
Total	-	-	-	-	-	-	**2421.33**	**9.70**	**1699.32**	**7.30**
Model II										
5	3,867.96	6	81	5	56	1.6320	149.61	0.20	136.94	1.00
10	4,441.42	5	85	4	56	1.5321	176.71	0.10	145.98	1.00
15	4,985.10	5	87	4	56	1.4824	173.17	0.30	142.89	1.00
20	5,501.92	4	90	3	56	1.4081	157.93	1.00	143.08	1.00
25	5,975.48	4	34	0	56	1.1326	173.86	0.50	144.40	0.10
30	5,975.48	4	34	0	56	1.1326	223.42	0.80	166.25	0.10
35	5,975.48	4	34	0	56	1.1326	229.12	0.70	156.24	0.20
40	5,975.48	4	34	0	56	1.1326	216.40	0.70	182.24	0.30
45	5,975.48	4	34	0	56	1.1326	223.55	0.90	176.13	0.20
50	5,975.48	4	34	0	56	1.1326	219.95	0.80	186.25	0.50
Total	-	-	-	-	-	-	**1943.71**	**6.00**	**1580.42**	**5.40**

It can be seen that the *JTEC* has a positive relation with c_o while the safety factor *s* has a negative relation with c_o, as expected. However, there are two useful observations. First, there seems to have a certain threshold value (somewhere between \$20/item/year to

\$25/item/year) over which the further increasing in c_o does not further influent the system. This effect of c_o is different from that of c_b. Second, when c_o is below the threshold value, the second model is superior to the first one since it yields a lower *JTEC*. On the contrary, the first model is a better choice when c_o is greater than the threshold value.

6.5.4 Effects of the lead time crushing cost coefficient c_i

Different from Braglia and Zavanella (2003), and Huang and Chen (2009), who assume the lead time crushing cost as a function of the period to be reduced, this research considers the lead time crushing cost as a function of both the time period to be reduced and the shipping quantity. In numerical Example 6.1, we used the same parameters that were used in Braglia and Zavanella's (2003), and Huang and Chen's (2009) examples, multiplied by a quantity discount c_{ij} (see Table 6.1). However, in our system, this c_{ij} is considered as a unit cost, so, to compute the total lead time reducing cost, the c_{ij} is multiplied by the shipping size q, which makes the lead time reducing cost in our system greater than that in theirs, which might not be true in practice. This fact can explain why the solutions in the tables 6.2-6.5 show that the optimal lead time is 56 days, meaning that it is not wise to invest on reducing the lead time. It follows that, to test the performance of the developed model, especially to test whether the model and the algorithms can help the decision maker to make the right choice of the lead time L, it is necessary to test a scenario wherein the unit lead time reducing cost c_{ij} is lower. Therefore, we further introduce another discount coefficient c_i that is less than one and we multiply it by c_{ij} to obtain the new unit lead time reducing

cost. We, then, vary c_i from 0.1 to 1 to test its effects on the system. The result is shown by Table 6.6.

Table 6.6 Effects of c_i to the system

	Optimal Solutions						PSO+IHS+HJ		MDE'+IHS+HJ	
c_i	*JTEC*($)	*n*	*q*	*k*	*L (day)*	*s*	Mean CPU(s)	Success Rate	Mean CPU(s)	Success Rate
Model I										
0.1	4,308.86	5	87	4	**42**	1.6703	171.69	0.40	180.75	0.80
0.2	4,434.86	5	87	4	**42**	1.6703	185.54	0.10	159.62	0.60
0.3	4,475.13	5	85	4	56	1.5321	180.22	1.00	154.99	1.00
0.4	4,475.13	5	85	4	56	1.5321	175.51	1.00	158.15	0.90
0.5	4,475.13	5	85	4	56	1.5321	178.00	0.90	160.45	0.90
0.6	4,475.13	5	85	4	56	1.5321	172.92	0.80	156.42	0.90
0.7	4,475.13	5	85	4	56	1.5321	174.60	1.00	156.48	1.00
0.8	4,475.13	5	85	4	56	1.5321	177.44	1.00	158.60	1.00
0.9	4,475.13	5	85	4	56	1.5321	171.55	1.00	155.71	0.90
1	**4,475.13**	**5**	**85**	**4**	**56**	**1.5321**	**177.41**	**0.90**	**156.28**	**1.00**
Total	-	-	-	-	-	-	**1764.88**	**8.10**	**1597.45**	**9.00**
Model II										
0.1	4,275.88	5	86	4	**42**	1.6987	147.25	1.00	115.91	1.00
0.2	4,401.88	5	86	4	**42**	1.6987	139.17	1.00	132.61	1.00
0.3	4,441.42	5	85	4	56	1.5321	139.13	1.00	124.20	1.00
0.4	4,441.42	5	85	4	56	1.5321	137.73	1.00	126.19	1.00
0.5	4,441.42	5	85	4	56	1.5321	138.18	1.00	120.87	1.00
0.6	4,441.42	5	85	4	56	1.5321	141.30	1.00	122.77	1.00
0.7	4,441.42	5	85	4	56	1.5321	150.97	1.00	128.40	1.00
0.8	4,441.42	5	85	4	56	1.5321	148.15	1.00	134.08	1.00
0.9	4,441.42	5	85	4	56	1.5321	138.86	1.00	137.07	1.00
1	**4,441.42**	**5**	**85**	**4**	**56**	**1.5321**	**128.84**	**1.00**	**136.25**	**1.00**
Total	-	-	-	-	-	-	**1409.58**	**10.00**	**1278.34**	**10.00**

It can be seen that when the additional discount coefficient c_i is 0.1 or 0.2, the hybrid algorithm presented in this research does suggest investing in reducing the lead time from 56 days to 42 days.

6.5.5 Effects of the ratio of the buyer's/vendor's storage unit holding cost h_b^s/h_v^s

Some other researchers, such as Braglia and Zavanella (2003), and Huang and Chen (2009), pointed out that the relation between the storage component of unit holding cost to the vendor, h_v^s, and to the buyer, h_b^s, are important factors. Whichever is greater will greatly affect the optimal solutions, especially to the number of delayed shipment *k*. In fact, they proved that, under a deterministic demand, when h_v^s is greater than h_b^s, the optimal solution will include a minimized number of delayed shipment *k*. On the contrary, when h_v^s is less than h_b^s, the optimal solution of *k* would be maximized. In order to verify the effects of this relationship under a stochastic demand background, we consider another example.

Example 6.2: Effect of holding cost

The parametric values of the Example 6.2 are also borrowed from Braglia and Zavanella (2003), and Huang and Chen (2009), and the values that are different from Example 6.1 are: $p_v = \$15$ /item, $h_v^s = \$2.5$ /item/year, and $h_b^s = \$3$ /item/year, which is greater than h_v^s. Table 6.7 shows the optimal solutions found by both the DHMAs while the space limitation is from 50 to 500.

Comparing Table 6.7 with Table 6.3, it can be seen that the *k* value in both examples gradually goes to zero when the space limitation becomes tighter. It seems that under a stochastic demand rate, the value of *k* is independent on the fact that whether h_v^s is greater than h_b^s or not. Rather, it depends, at least partly, on the value of *U*. A further investigation on the effects of the relation between h_v^s and h_b^s was conducted in which the value of h_b^s

was set from 0.5 to 5 with a step size 0.5. The results show no apparent relationship between k and the ratio of h_b^s / h_v^s. This finding is useful in that it reveals a different working performance of a deterministic model and a stochastic one.

Table 6.7 Effects of U to the system for numerical Example 6.2

	Optimal Solutions						PSO+IHS+HJ		MDE'+IHS+HJ	
U	*JTEC*($)	*n*	*q*	*k*	*L*(day)	*s*	Mean CPU(s)	Success Rate	Mean CPU(s)	Success Rate
Model I										
50	7,003.67	29	22	29	56	1.0771	98.88	0.90	138.12	0.90
100	5,038.55	11	51	10	56	1.1906	131.91	0.80	167.14	1.00
150	**4,361.31**	**6**	**85**	**5**	**56**	**1.5321**	**171.42**	**0.70**	**210.13**	**1.00**
200	4,044.27	4	123	3	56	1.7579	183.49	0.60	150.18	1.00
250	3,816.81	3	110	1	56	1.4854	324.22	0.40	267.67	0.40
300	3,524.19	4	76	0	56	1.5976	463.23	0.70	220.31	0.20
350	3,336.19	4	92	0	56	1.5990	507.10	1.00	238.58	0.70
400	3,240.28	4	107	0	56	1.6728	504.89	0.90	231.49	0.70
450	3,201.19	4	123	0	56	1.6737	557.53	1.00	237.06	0.90
500	3,196.62	4	132	0	56	1.6694	468.81	1.00	254.83	0.90
Total	-	-	-	-	-	-	**3411.48**	**8.00**	**2115.50**	**7.70**
Model II										
50	7,018.03	29	22	29	56	1.0771	78.85	1.00	126.89	1.00
100	5,051.35	11	51	10	56	1.1906	133.59	1.00	141.27	1.00
150	**4,374.34**	**6**	**85**	**5**	**56**	**1.5321**	**147.33**	**1.00**	**129.05**	**0.90**
200	4,054.02	4	123	3	56	1.7295	156.69	1.00	151.00	1.00
250	3,805.94	3	110	1	56	1.4854	257.31	0.80	165.27	0.40
300	3,498.84	4	76	0	56	1.5976	390.44	0.80	208.82	0.50
350	3,309.94	4	91	0	56	1.6722	470.67	0.90	202.34	0.30
400	3,212.88	4	107	0	56	1.6728	400.87	0.90	225.22	0.50
450	3,173.34	4	123	0	56	1.6737	534.11	0.60	238.59	0.80
500	3,168.19	4	131	0	56	1.7150	527.89	1.00	232.44	1.00
Total	-	-	-	-	-	-	**3097.76**	**9.00**	**1820.88**	**7.40**

Briefly, there are several important findings in this Section. First, the two doubly-hybrid approaches is useful to solve the newly developed two models. This finding may provide the decision makers of such a system a choice of solution methods. The second

useful finding is that, for the two models developed in this Chapter, one could be more satisfactory than the other, given different parametric values. For example, when the buyer's space limitation is tight, when the backorder penalty is high, or when the extra space penalty is below some certain threshold, the modified three-factor CS model is a better policy. Based on the real parametric values of a system, the mathematical models and the doubly-hybrid approaches can assist the manager to make the right choice. Some conclusions made by researchers based upon deterministic models may not apply for variable demand cases; such as when the unit holding cost is higher to the vendor, the number of delayed shipments should be equal to zero which may not be true in the case of variable demand.

6.6 CONCLUSION

This Chapter studies the effects of the introduction of a stochastic demand to an integrated lead time controllable CS inventory system with buyer's space limitation. Two integrated inventory models are developed to jointly determine the optimal values of five decision variables that minimize the annual *JTEC* of the system. Due to the complexity of the problem, closed-form solutions of the models are not presented. Instead, two doubly-hybrid meta-heuristic algorithms are used to solve the models. Numerical examples showed that both the doubly-hybrid algorithms are satisfactory both in the sense of the CPU time and the success rate. The computational analysis first revealed how the stochastic models work different from the deterministic ones and then disclosed how some important

parameters affect the solutions and the objective functions. The results obtained in this research help understand the role of stochastic demand rate, the buyer's space limitation, controllable lead time, and the CS mechanism better. Moreover, the successful use of doubly-hybrid meta-heuristic algorithms to inventory problems provides a way of solving more difficult and complicated models.

CHAPTER VII
FOUR-FACTOR CS MODEL: CONTROLLABLE LEAD TIME, BUYER'S SPACE LIMITATION, VARIABLE DEMAND, AND OBSOLESCENCE

The risk of obsolescence is important in practice. Failure to consider the impact of obsolescence, resulting in unsold products, will decrease the profit. However, it is rare in the literature that that factor is considered, especially in the context of an integrated CS system. Traditionally, the risk of obsolescence is borne solely by the buyer of a two-echelon supply chain. On the contrary, under a CS scenario, this risk is shared by both the vendor and the buyer, since the unsold products remain owned by the vendor. As a result, the replenishment policies of both the vendor and the buyer have to be modified to accommodate this risk. In order to provide the decision makers with such a system the optimal decisions against the obsolescence risk, a four-factor CS model considers obsolescence, variability in demand rate, buyer's space limitation, and controllable lead time, is developed in this Chapter. The objective of this model is to jointly decide the optimal ordering size, number of shipments within each production cycle, the number of delay shipments within each cycle, the lead time, and the safety stock, that minimize the annual joint total expected cost (*JTEC*) of the system.

7.1 THE PROBLEM

Under a traditional CS policy, either the vendor keeps most of the inventory (when the buyer's unit inventory cost is much higher than that of the vendors) while maintaining a minimum amount of inventory in the buyer's warehouse, or the buyer stores the majority of

the products (when the buyer's unit inventory cost is lower to the vendor), keeping the lower amount at the vendor's site. This is defined by Yi and Sarker (2013a, 2013b) as the CS (k = 0) and CS ($k = n - 1$) policy. Based on the four-factor CS model, there are several system constraints to be considered.

Firstly, the buyer may want to place a space limitation to each of his/her supplier for each of the product. When there is an upper limit capacity in the space, the vendor cannot put as much inventory as s/he wants to the buyer. In the beginning, when the buyer's space limitation is not reached, the vendor is obliged, based on a CS agreement, to keep the buyer's inventory above a certain safety level. Toward this end, the products are shipped to the buyer in small quantities without having to wait until the up-time of each production cycle is ended. As a result, the buyer's inventory level gradually increases in the beginning of each cycle. In the meanwhile, the vendor maintains a minimum level of average inventory (equal to half the shipping size) during this period. However, when the inventory level in buyer's warehouse reaches its maximum level, I_{max}, which is close or equal to the buyer's space limitation, all later shipments from the vendor are delayed for a certain period so that the late arrival of a new shipment brings the buyer's inventory level back to I_{max}. As a result, the vendor's inventory level is forced to be increased to a certain level.

Secondly, the shipments are taking a certain period to reach the buyer, which incurs a shipping cost and a holding cost of the products in transit. However, the lead time can be reduced, often required by the buyer, with an extra charge, which is also shared by both the

vendor and buyer under a long-term agreement. The lead time crushing cost is a function of both the time to be reduced and the quantity of the products to be shipped. The tradeoff is that, while the shipping cost might be increased, the holding cost in transit is reduced, and the safety stock level is also reduced accordingly because of a shorter lead time.

The third constraint is that the demand rate may be uncertain. Based upon historical data; however, it may follow a stochastic distribution with known mean and variance. Because of the variation in demand, there might be back orders and/or some extra space may be needed in the buyer's warehouse. Both of the two cases cost the system an extra.

Finally, the product may be obsolescent sometime, which may occur at any time within the last production cycle. Thus, the last cycle may be incomplete. After that point, the unsold products and the material are considered to be lost. Hence, it incurs an obsolescence cost to the system.

The four-factor model to be developed later in this Chapter needs to address all the issues so as to help the decision makers find the best combination of the decision variables: (a) the number of shipments made within any production cycle, (b) the shipping size, (c) the number of delayed shipment, (d) the lead time, and (e) the safety stock level in the buyer's warehouse, so as to minimized the annual *JTEC* of the system.

The remainder of this Chapter is organized as follows. Section 7.2 first defines local parameters and assumptions for the four-factor CS model. The model is then formulated in Section 7.3. Computational results of the Doubly-hybrid Meta-Heuristic methods as well as

that of the Exhaustive search method are illustrated and compared in Section 7.4. The effects of important parameters are analyzed in Section 7.5 and finally, conclusions are make in Section 7.6.

7.2 NOTATION AND ASSUMPTIONS

The four-factor CS model is different from the previous ones. In order to develop the integrated models, some additional notations and assumptions other than what was defined in Chapter IV are needed and are given below:

Additional Variable: *k, s*.

Local Notations:

I_{max}: Buyer's maximum inventory level (units),

U: Buyer's space limitation (unit),

c_b: Unit backorder cost (\$/unit),

C_b: Expected annual backorder cost (\$/year),

c_{il}: Unit crushing cost for reducing one time unit of the *i*th segment of lead time S_i when the ordering quantity q is between q_{l-1} and q_l (\$/year),

c_o: Unit outsourcing cost (\$/unit),

C_o: Expected annual extra space cost (\$/year),

D: Yearly demand rate at the buyers' level (units/year), $D \sim N(\mu, \sigma)$,

E(▪): Mathematical expectation of ▪,

n^*: Number of full production cycles during the entire planning horizon,

$n^* = \lfloor \mu T / nq \rfloor$,

$R(q, L)$: Lead time crushing cost per replenishment cycle ($/shipment),

s: Safety factor,

s_S: Safety stock level (units),

t: Length of the last incomplete production cycle (year), $t = T - n^* \times nq / \mu$,

t_{li}^{I}: Lower bound of the ith scenario (year), where i is a positive integer less or equal to four,

t_{ui}^{I}: Upper bound of the ith scenario (year), where i is a positive integer less or equal to four,

T: Item life period (year),

U: Space limitation placed by the buyer to the vendor (units),

μ: Expectation value of annual demand rate D (units/year), $\mu = \int_{-\infty}^{+\infty} Df(D)dD$,

σ: Standard deviation of annual demand rate D (units),

v: Number of price segments associated with ordering quantity,

x^+: Maximum value of x and 0, i.e. $x^+ = \max\{x, 0\}$,

X_1: The demand during the period $(q/P + L)$, having a mean $\mu(q/P + L)$ and standard deviation $\sigma\sqrt{q/P + L}$,

X_2: The demand during the period q/P, having a mean $q\mu/P$ and standard deviation $\sigma\sqrt{q/P}$,

X_3: The demand during the period q/μ, having a mean q and standard deviation

$\sigma\sqrt{q/\mu}$,

X4: A random variable associated with the four-factor CS model, i.e., the during the period $(n-k)q/\mu-(n-k)q/P-L$, having a mean $(n-k)q \quad -(n-k)q\mu/P-\mu L$ and standard deviation $\sigma\sqrt{(n-k)q/\mu-(n-k)q/P-L}$,

X5: The random demand during the last incomplete production cycle *t*, having a mean μt and standard deviation $\sigma\sqrt{t}$,

X6: The random demand during the period $t-L-q/P-\lfloor(t-L-q/P)P/q\rfloor q/P$, having a standard deviation $\sigma\sqrt{t-L-q/P-\lfloor(t-L-q/P)P/q\rfloor q/P}$ and a mean $\mu(t-L-q/P) \; -\mu\lfloor(t-L-q/P)P/q\rfloor \, q/P$,

X7: A random variable associated with scenarios 3 and 4 of the four-factor CS model, i.e., the demand during the period $t-L-(n-k)q/P-\lfloor[t-L-(n-k)q/P]\mu/q\rfloor q/\mu$, having a mean $\mu[t-L-(n-k)q/P]-\lfloor[t-L-(n-k)q/P]\mu/q\rfloor q$ and a standard deviation $\sigma\sqrt{t-L-(n-k)q/P-\lfloor[t-L-(n-k)q/P]\mu/q\rfloor q/\mu}$.

Local Assumptions*:*

The following additional assumptions are necessary for the four-factor CS model developed in this Chapter:

(1) The demand of the buyer *D* follows a normal distribution with a mean μ and a standard deviation σ, i.e., $D \sim N(\mu, \sigma^2)$.

(2) The demand during lead-time L also follows normal distribution with a mean μL and a standard deviation $\sigma\sqrt{L}$.

(3) Due to the demand uncertainty, shortage is allowed in the system and is backordered with a shortage cost.

(4) Due to the demand uncertainty, extra inventories beyond the buyer's capacity are allowed. The extra products may be still stored in the buyer's warehouse but in a space reserved for other products/suppliers of the buyer or they may be stored by a third party. In both cases the vendor will be charged an extra penalty cost.

(5) The shipping cost is an incremental function of both the guaranteed lead time and the shipment quantity.

(6) The predicted time between the beginning of the first production cycle and the date of obsolescence is deterministic (Persona, *et al.*, 2005, Battini, *et al.*, 2010a, 2010b).

7.3 MODEL FORMULATION

In this Section, we develop a CS-k model considering four practical factors: (a) obsolescence (Obs), (b) controllable lead time (CLT), (c) buyer's capacity limitation (BCL), and (d) stochastic demand (SD). The pattern of the system inventory, the vendor inventory, the inventory in transit, and the buyer inventory in a full production cycle of this model is shown in Figure 7.1.

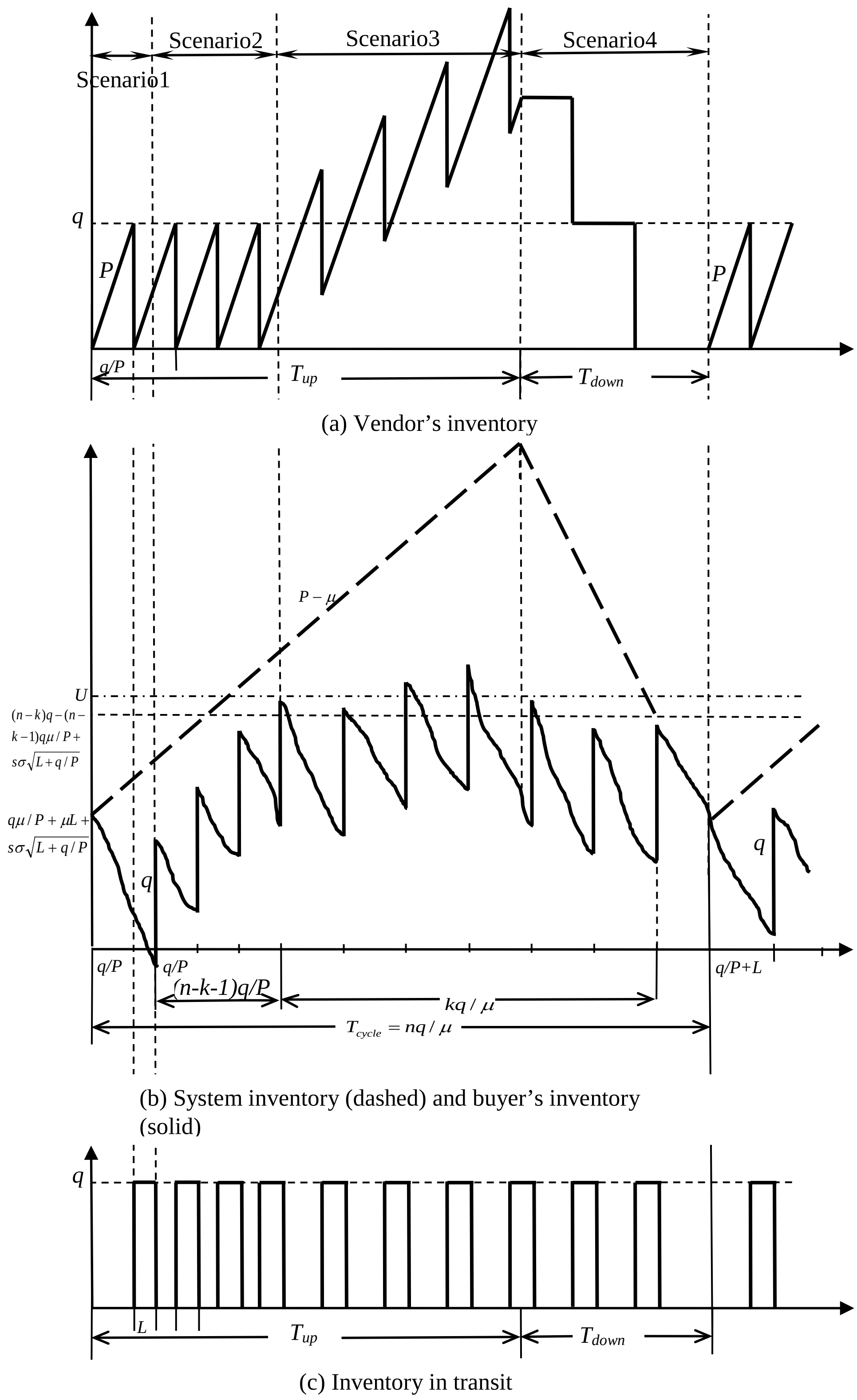


Figure 7.1 Composition of the inventory in the four-factor CS model

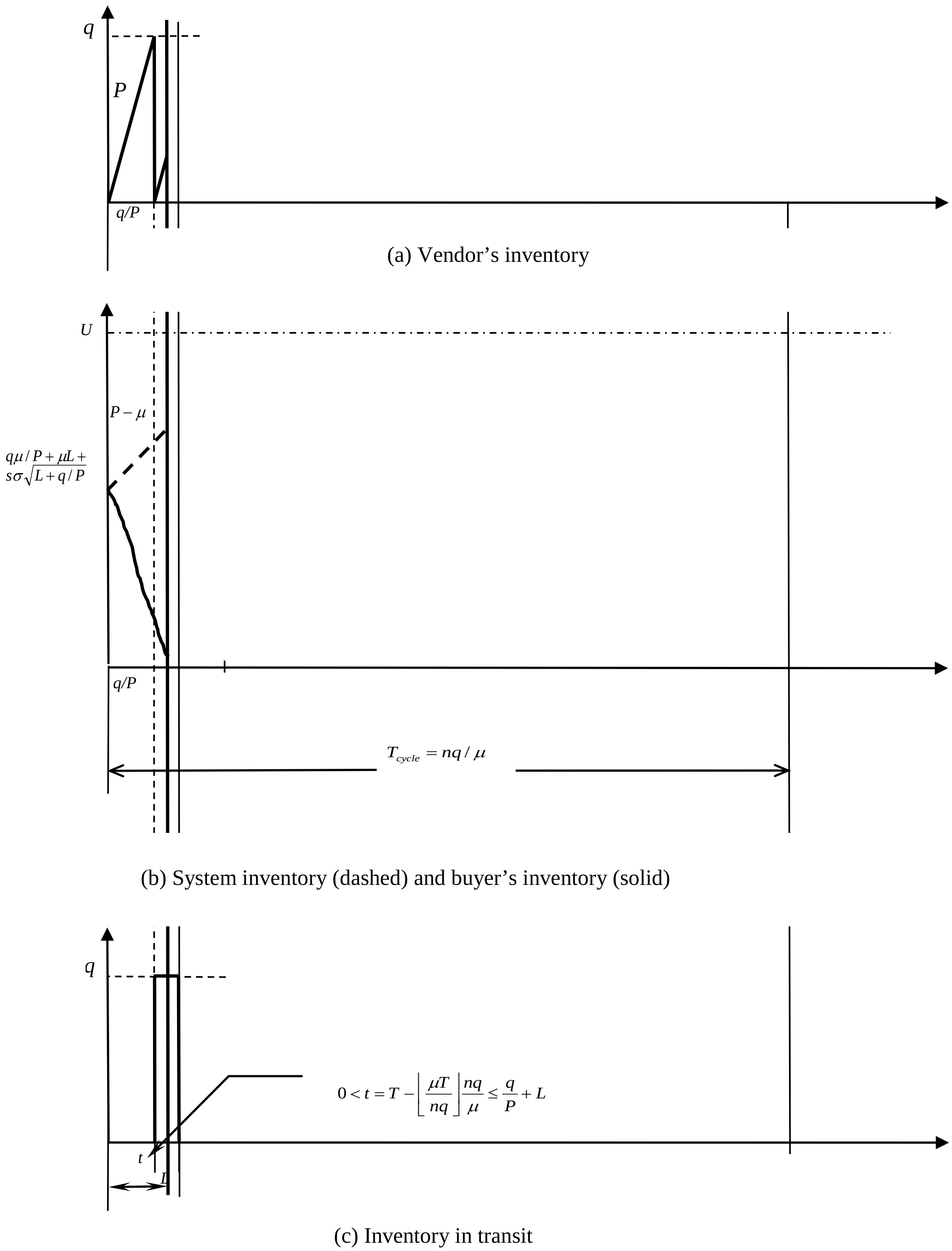
Within a full production cycle, the vendor produces at a finite production rate P, and the annual demand D follows a normal distribution with mean μ and standard deviation σ. Under a long term CS agreement, the vendor satisfies the buyer's demand in each production cycle with n shipments of equal lot size q. At the beginning of each production cycle, the buyer has an average initial inventory that equals to $\mu(L+q/P)+s\sigma\sqrt{L+q/P}$ which remains from the last production cycle, where, the safety factor s is one of the decision variables used to decide the safety stock level s_s, so as to mitigate the risk of stock outs due to the uncertain demand. The relationship between the safety factor and the safety stock level can be given by $s_s = s\sigma\sqrt{L+q/P}$. The first several shipments are made whenever the vendor's inventory reaches the level q. Each shipment reaches the buyer within a fixed length L that is reducible from the normal length L_0 with a "crushing" cost, which is a function of the reduced time $L - L_0$ and the ordering quantity q. Due to the uncertain nature of the demand, the lead time demand has a mean of μL and a standard deviation of $\sigma\sqrt{L}$ and the expected cycle length is $T_{cycle} = nq/\mu$. Meanwhile, the buyer places a space limitation U to the vendor. Therefore, when the buyer's inventory reaches the point $I_{\max} = (n-k)q - (n-k-1)q\mu/P + s\sigma\sqrt{L+q/P} \le U$, all of the following k shipments are delayed for a period such that the arrival of a new shipment always bring the buyer's average inventory up back to I_{max} (Figure 1b).

The unit holding cost h consists of a pure financial component, h^f, and a pure storage component, h^s. Under CS policy, all the financial component of holding costs is carried by

the vendor which can be calculated by $h_v^f = rp_v$. There are three different unit holding costs, accordingly, with different physical locations of products: vendor unit holding cost, $h_v = h_v^s + h_v^f$, unit holding cost in transit, $h_d = h_d^s + h_v^f$, and buyer unit holding cost $h_b = h_b^s + h_v^f$.

Past research considered the lead time crushing cost as a single function of the time period being reduced [Braglia and Zavanella (2003), and Huang and Chen (2009)]. However, in practice, the lead time cost quite often depends on the quantity being shipped. Therefore, we consider the composition of the lead time crushing cost different from them. In this system, the lead time has *m* mutually independent components, each with a minimum duration a_i and a maximum duration b_i, $i = 1, 2, \ldots, m$. The crushing cost per unit time c_{il} is an incremental function of both reduced lead time and shipping quantity, and is arranged such that $c_{il} \le c_{(i+1)l}$, and $c_{il} \ge c_{i(l+1)}$, $\forall i, l$, where $i = 1, 2, ..., m$, $l = 1, 2, ..., v$, and v denotes the total number of quantity discount segments. Within each quantity range, the lead time L is crushed one segment at a time starting with the least c_{ij}, and so on. Let L_0 denote the normal lead time before crushing, i.e., $L_0 = \sum_1^m b_j$ and L_i denote the lead time where the *i*th component was crushed to its minimum duration. Thus, the lead time L_i can be written as $\sum_{j=1}^{i} a_j + \sum_{j=i+1}^{m} b_j = \sum_{j=1}^{m} b_j - \sum_{j=1}^{i} (b_j - a_j) = L_0 - \sum_{j=1}^{i} (b_j - a_j)$, $i = 1, 2, ..., m$. The lead time crushing cost per replenishment cycle can thus be calculated by $R(q, L) = qc_{il}(L_{i-1} - L) + q\sum_{j=1}^{i-1} c_{jl}(b_j - a_j)$, $i = 1, 2, ..., m$, $l = 1, 2, ..., v$ for $q \in (q_{l-1}, q_l)$ and $L \in (L_i, L_{i-1})$.

Unlike normal full production cycles, the last cycle may be incomplete due to the product obsolescence. To avoid further lost caused by the obsolescence, the vendor is assumed to cease its production at the moment when the obsolescence occurs. The remaining stocks are considered to be lost and will be cleared from the warehouses immediately. The in-transit inventory, if any, will be cleared when it reaches the buyer. Because the obsolescence can occur anytime within the last cycle, the inventory patterns can be categorized into four scenarios according to the time that the obsolescence may occur (Scenarios 1, 2, 3, and 4 in Figure 7.1a). Figures 7.2, 7.3, 7.4 and 7.5 illustrate the inventory patterns of all parties (vendor, buyer, in transit, and system) associated with each of the four scenarios, respectively. Specifically, Scenario 1 (Figure 7.2) is a situation when the obsolescence occurs before the arrival of the first shipment. Scenario 2 (Figure 7.3) reflects when the obsolescence occurs after the arrival of the first shipment, but before the maximum inventory level I_{max} is reached. Scenario 3 (Figure 7.4) describes when obsolescence occurs after the beginning of the first delayed shipment, but before the end of the up-time $T_{up} = nq / P$. Scenario 4 (Figure 7.5) illustrates when obsolescence occurs during the down-time.



(a) Vendor's inventory

(b) System inventory (dashed) and buyer's inventory (solid)

(c) Inventory in transit

Figure 7.2 Inventory patterns of the last production cycle associated with Scenario 1

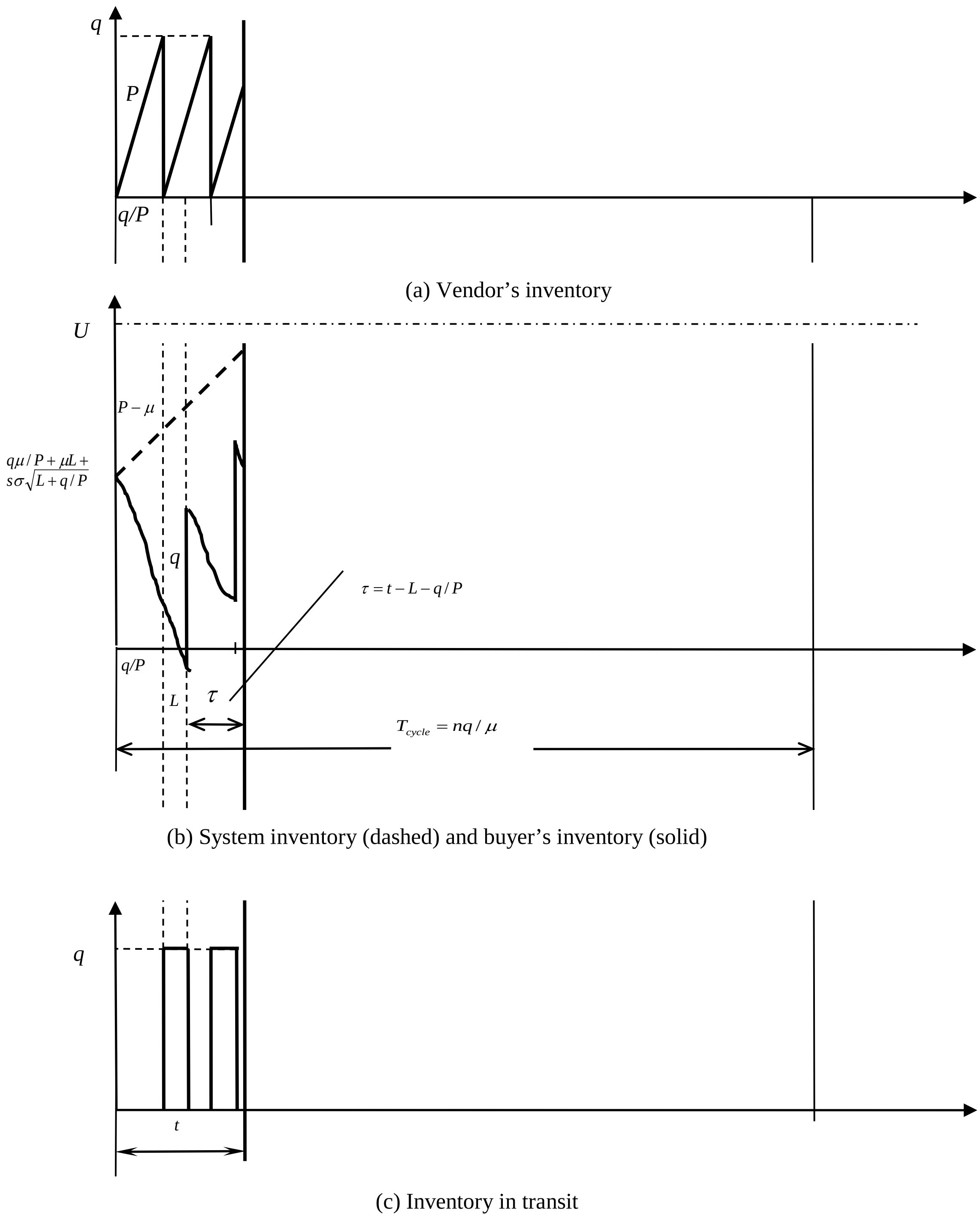
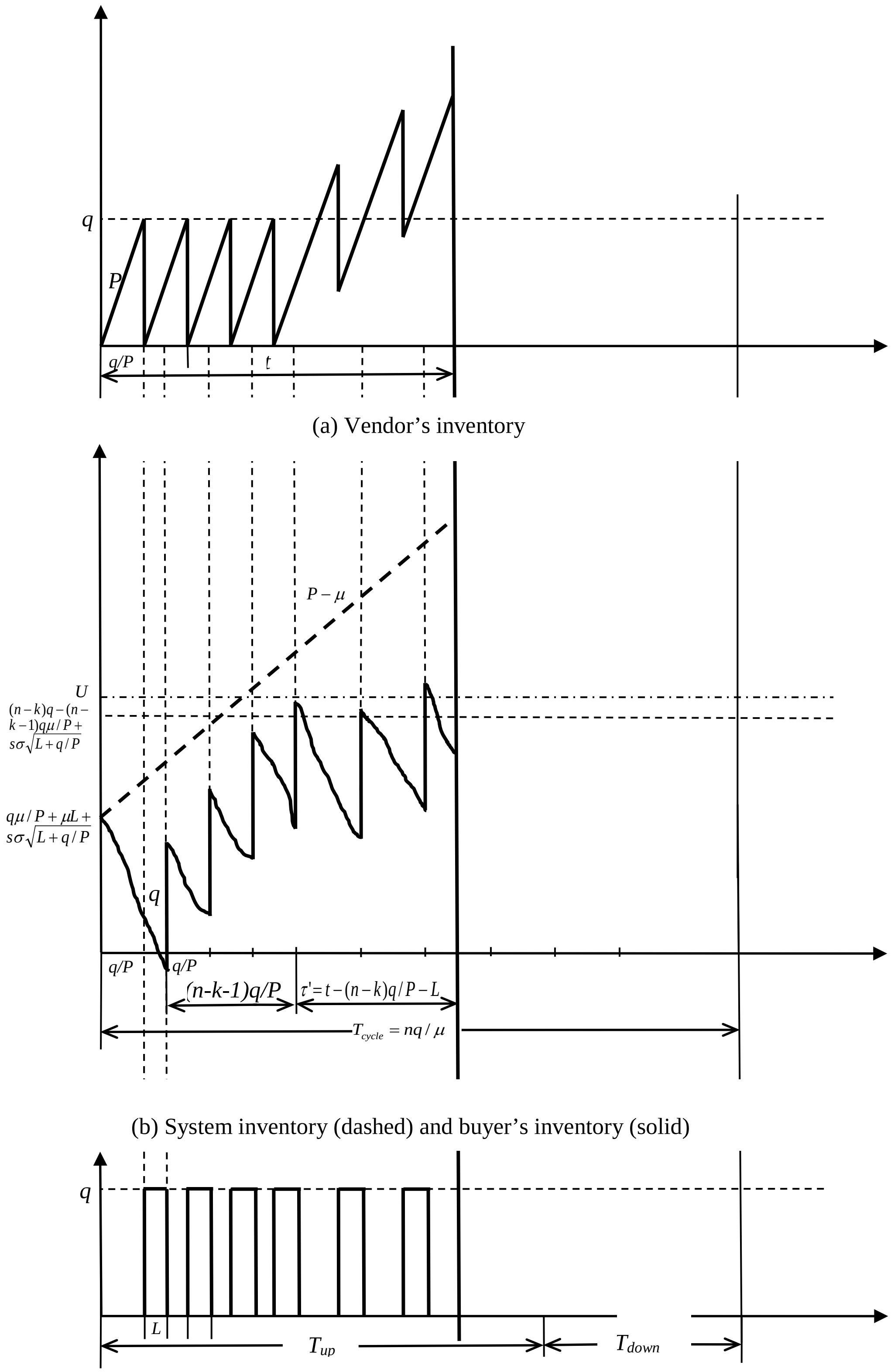


Figure 7.3 Inventory patterns of the last production cycle associated with Scenario 2

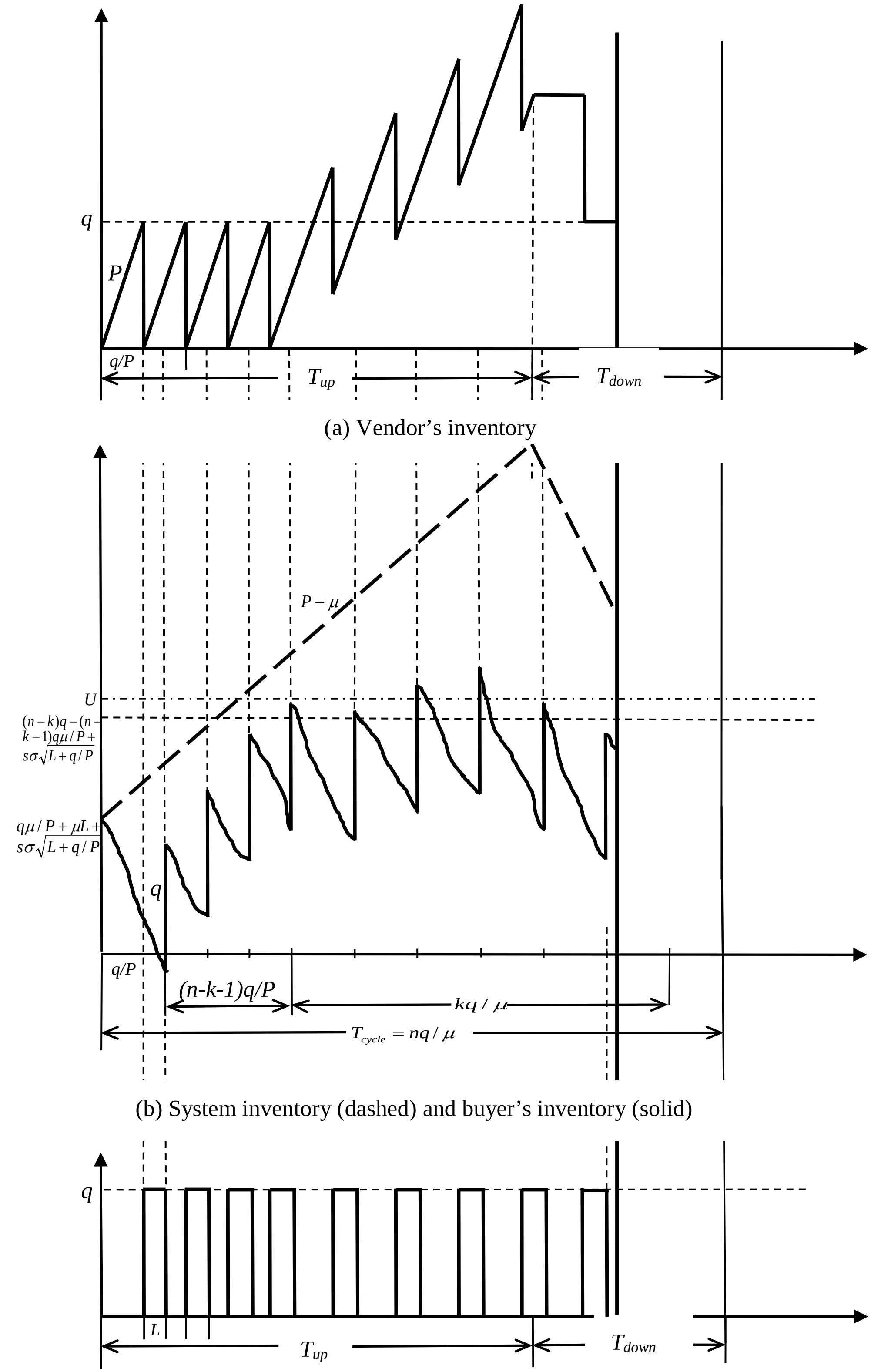


(a) Vendor's inventory

(b) System inventory (dashed) and buyer's inventory (solid)

(c) Inventory in transit

Figure 7.4 Inventory patterns of the last production cycle associated with Scenario 3



(a) Vendor's inventory

(b) System inventory (dashed) and buyer's inventory (solid)

(c) Inventory in transit

Figure 7.5 Inventory patterns of the last production cycle associated with Scenario 4

According to Valentini and Zavanella (2003), the buyer's ordering cost is zero under a CS policy. As the result, the joint total expected cost of the system, *JTEC*, can be written as the sum of the vendor's expected setup cost, the buyer's expected backorder cost, the buyer's expected extra space cost, the extra expected lead time crushing cost, the expected system holding cost, and the obsolescence cost. The calculations of all these costs are discussed below.

7.3.1 The vendor's expected annual setup cost C_s^v

The vendor's expected annual setup cost C_s^v can be written as

$$C_s^v = \frac{A_v}{T}\left\lceil \frac{\mu T}{nq} \right\rceil. \tag{7.1}$$

7.3.2 The expected annual lead time crushing cost C_L

The lead time crushing cost per replenishment cycle has been described earlier in this Section as $R(q,L) = qc_{il}(L_{i-1} - L) + q\sum_{j=1}^{i-1} c_{jl}(b_j - a_j)$, $i = 1, 2, ..., m$, $l = 1, 2, ..., v$, for $q \in (q_{l-1}, q_l)$ and $L \in (L_i, L_{i-1})$. Therefore, the expected annual lead time crushing cost can be written as

$$C_L = \frac{R(q,L)}{T}\left\lceil \frac{\mu T}{q} \right\rceil = \frac{qc_{il}(L_{i-1} - L) + q\sum_{j=1}^{i-1} c_{jl}(b_j - a_j)}{T}\left\lceil \frac{\mu T}{q} \right\rceil, \tag{7.2}$$

where, $i = 1, 2, ..., m$, $l = 1, 2, ..., r$ for $q \in (q_{l-1}, q_l)$ and $L \in (L_i, L_{i-1})$.

7.3.3 The expected system holding cost *H*

The expected system holding cost *H* consists of three parts: the expected holding cost incurred in the vendor's warehouse H_v, the expected holding cost in transit H_t, and the expected holding cost incurred in the buyer's warehouse H_b. Moreover, each of these three parts has four cases associated with the four scenarios as shown in Figure 7.1(a). The formulas used to calculate H_v, H_t, and H_b are thus different according to different cases and are given, in turns, as below:

(a) The expected annual holding cost incurred in the buyer's inventory H_b.

Case 1: Case 1 is associated with the Scenario 1 in Figure 7.1(a). Let *t* be the duration of the last production cycle before the obsolescence occurs, then it can be calculated as $t = T - \lfloor \mu T / nq \rfloor nq / \mu$. Therefore, the condition for Case 1 can be expressed as $t_{l1} = 0 < t \le t_{u1} = q/P + L$. Under this case, the inventory patterns of all three parties (vendor, buyer, and in transit) of the last production cycle are shown in Figure 7.2. According to Figure 7.1(b) and Figure 7.2(b), the expected annual holding cost incurred in the buyer's inventory can be written as:

$$H_b = (h_b^s + h_v^f)\left\{\left[\frac{q\mu}{2P} + nq\frac{P-\mu}{2P} - q\frac{P-\mu}{nP}\frac{(k+1)k}{2} + s\sigma\sqrt{L+q/P}\right]n * \frac{nq}{\mu}\right.$$

$$\left. + \left[\left(\frac{q\mu}{P} + \mu L + s\sigma\sqrt{L+q/P}\right) - \frac{\mu t}{2}\right]t\right\} \Big/ T, \tag{7.3a}$$

where, the expression $\frac{q\mu}{2P} + nq\frac{P-\mu}{2P} - q\frac{P-\mu}{nP}\frac{(k+1)k}{2} + s\sigma\sqrt{L+q/P}$ is the buyer's average inventory level of a full production cycle (Yi and Sarker, 2013c). The expression

$\left(\frac{q\mu}{P}+\mu L+\ s\sigma\sqrt{L+\frac{q}{P}}\right)-\frac{\mu}{2}t$ is the buyer's average inventory level of the last production cycle under Case 1. Moreover, $n^*=\lfloor \mu T/nq \rfloor$ is the number of full production cycles.

Case 2: Similarly, the condition of Case 2 can be written as $q/P+L<t\leq (n-k+1)q/P+L$. The inventory patterns of Case 2 are given in Figure 7.3. According to Figure 7.1(b) and Figure 7.3(b), the H_b can be written as:

$$H_b=(h_b^s+h_v^f)\left\{\left[\frac{q\mu}{2P}+nq\frac{P-\mu}{2P}-q\frac{P-\mu}{nP}\frac{(k+1)k}{2}+s\sigma\sqrt{L+q/P}\right]n^*\frac{nq}{\mu}\right.$$

$$+\frac{1}{2}\left(\frac{q\mu}{P}+\mu L+s\sigma\sqrt{L+\frac{q}{P}}\right)\left(L+\frac{q}{P}\right)+\frac{i\mu q^2}{2P}+\frac{i(i-1)}{2}\left(1-\frac{\mu}{P}\right)\frac{q^2}{P}$$

$$\left.+q(i+1)\left(\tau-\frac{iq}{P}\right)-\frac{iq\mu}{P}\left(\tau-\frac{iq}{P}\right)-\frac{\mu}{2}\left(\tau-\frac{iq}{P}\right)^2\right\}\Big/T\ , \tag{7.3b}$$

where, $n^*=\lfloor \mu T/nq \rfloor$, $\tau=t-L-q/P$, and $i=\lfloor \tau P/q \rfloor$.

Case 3: the condition of Case 3 is $(n-k+1)q/P+L<t\leq T_{up}=nq/P$. The inventory patterns of this scenario are illustrated in Figure 7.4. According to Figure 7.1(b) and Figure 7.4(b), the H_b can be written as:

$$H_b=(h_b^s+h_v^f)\left\{\left[\frac{q\mu}{2P}+nq\frac{P-\mu}{2P}-q\frac{P-\mu}{nP}\frac{(k+1)k}{2}+s\sigma\sqrt{L+q/P}\right]n^*\frac{nq}{\mu}\right.$$

$$+\frac{1}{2}\left(\frac{q\mu}{P}+\mu L+s\sigma\sqrt{L+\frac{q}{P}}\right)\left(L+\frac{q}{P}\right)+\frac{(n-k)(2P-\mu)q^2+(n-k)(n-k-1)(P-\mu)q^2}{2P^2}$$

$$+\frac{i'q}{\mu}\left[\left(n-k-\frac{1}{2}\right)q-\frac{(n-k-1)q\mu}{P}+s\sigma\sqrt{L+\frac{q}{P}}\right]+\left[(n-k)q-\frac{(n-k-1)q\mu}{P}+s\sigma\sqrt{L+\frac{q}{P}}\right.$$

$$\left.\left.-\frac{\mu}{2}\left(\tau'-\frac{i'q}{\mu}\right)\right]\left(\tau'-\frac{i'q}{\mu}\right)\right\}\Big/T\ , \tag{7.3c}$$

where, $n^* = \lfloor \mu T / nq \rfloor$, $\tau' = t - L - (n-k)q/P$, and $i' = \lfloor \tau' \mu / q \rfloor$.

Case 4: the condition of Case 4 is $T_{up} = nq/P < t \le T_{cycle} = nq/\mu$. The inventory patterns of this scenario are presented in Figure 7.5. According to Figure 7.1(b) and Figure 7.5(b), the H_b can be written as:

$$H_b = (h_b^s + h_v^f)\left\{\left[\frac{q\mu}{2P} + nq\frac{P-\mu}{2P} - q\frac{P-\mu}{nP}\frac{(k+1)k}{2} + s\sigma\sqrt{L+q/P}\right]n^*\frac{nq}{\mu}\right.$$

$$+\frac{1}{2}\left(\frac{q\mu}{P} + \mu L + s\sigma\sqrt{L+\frac{q}{P}}\right)\left(L+\frac{q}{P}\right) + \frac{(n-k)(2P-\mu)q^2 + (n-k-1)(n-k)(P-\mu)q^2}{2P^2}$$

$$+\frac{i''q}{\mu}\left[\left(n-k-\frac{1}{2}\right)q - \frac{(n-k-1)q\mu}{P} + s\sigma\sqrt{L+\frac{q}{P}}\right] + \left[(n-k)q - \frac{(n-k-1)q\mu}{P} + s\sigma\sqrt{L+\frac{q}{P}}\right.$$

$$\left.\left.-\frac{\mu}{2}\left(\tau' - \frac{i''q}{\mu}\right)\right]\left(\tau' - \frac{i''q}{\mu}\right)\right\}\Big/T, \qquad (7.3d)$$

where $n^* = \lfloor \mu T / nq \rfloor$, $\tau' = t - L - (n-k)q/P$, and $i'' = \min\left(k, \lfloor \tau' \mu / q \rfloor\right)$.

(b) The expected holding cost incurred in the vendor's warehouse H_v.

Similarly, as H_b, the expected holding cost incurred in the vendor's warehouse H_v also can be categorized into four cases and are to be given as below.

For Cases 1 and 2:

$$H_v = (h_v^s + h_v^f)\left\{\left[\frac{q\mu}{2P} + q\frac{P-\mu}{nP}\frac{(k+1)k}{2}\right]n^*\frac{nq}{\mu} + \frac{q^2}{2P}\left\lfloor\frac{tP}{q}\right\rfloor + \frac{P}{2}\left(t - \left\lfloor\frac{tP}{q}\right\rfloor\frac{q}{P}\right)^2\right\}\Big/T, \qquad (7.4a, 7.4b)$$

in which, $n^* = \lfloor \mu T / nq \rfloor$.

For Case 3:

$$H_v = (h_v^s + h_v^f)\left\{\left[\frac{q\mu}{2P} + q\frac{P-\mu}{nP}\frac{(k+1)k}{2}\right]n^*\frac{nq}{\mu} + \frac{(n-k)q^2}{2P} + \frac{n'Pq^2}{2\mu^2} + \frac{n'(n'+1)(P-\mu)q^2}{2\mu^2}\right.$$

$$+\frac{n'q(P-\mu)}{2\mu}\left[t-\frac{(n-k)q}{P}-\frac{n'q}{\mu}\right]+\frac{P}{2}\left[t-\frac{(n-k)q}{P}-\frac{n'q}{\mu}\right]^2\Bigg\}\Bigg/T, \qquad (7.4c)$$

where, $n^*=\lfloor \mu T/nq \rfloor$, and $n'=\left\lfloor \frac{t\mu}{q}-\frac{(n-k)\mu}{P}\right\rfloor-1$.

For Case 4: that is, when $t_{l4}^I = nq/P < t \le t_{u4}^I = nq/\mu$, the H_v becomes

$$H_v=(h_v^s+h_v^f)\left\{\left[\frac{q\mu}{2P}+q\frac{P-\mu}{nP}\frac{(k+1)k}{2}\right]n^*\frac{nq}{\mu}+\frac{(n-k)q^2}{2P}+\frac{n''Pq^2}{2\mu^2}+\frac{n''(n''+1)(P-\mu)q^2}{2\mu^2}\right.$$

$$+\frac{n''q(P-\mu)}{2\mu}\left(\frac{kq}{P}-\frac{n''q}{\mu}\right)+\frac{P}{2}\left(\frac{kq}{P}-\frac{n''q}{\mu}\right)^2$$

$$+\min\left(t-\frac{nq}{P},\frac{n''+1}{\mu}q-\frac{kq}{P}\right)(k-n'')q$$

$$\left.+\sum_{j=1}^{n''-k}\max\left[0,\min\left(t-\frac{n-k}{P}q-\frac{n''+j}{\mu}q,\frac{q}{\mu}\right)\right](k-n''-j)q\right\}\Bigg/T, \qquad (7.4d)$$

where, $n^*=\lfloor \mu T/nq \rfloor$, and $n''=\left\lfloor \frac{k\mu}{P}\right\rfloor$ is the number of the delayed shipments during the up time. Note that n'' is an integer and is less than k, hence the term $(k-n''-j)$ will finally converge to zero.

(c) The expected annual holding cost in transit H_d

The expected annual holding cost in transit H_d is given by:

For Cases 1 and 2: $t_{l1}^I = 0 < t \le t_{u2}^I = (n-k+1)q/P+L$,

$$H_d=(h_d^s+h_v^f)\left(nqLn^*+\left\lfloor \frac{tP}{q}\right\rfloor qL\right)\Bigg/T, \qquad (7.5a,\ 7.5b)$$

where, $n^*=\lfloor \mu T/nq \rfloor$.

and for Case 3 and 4: $t_{l3}^I=(n-k+1)q/P+L<t\le t_{u4}^I = nq/\mu$,

$$H_d = (h_d^s + h_v^f)\left(nqLn^* + (n-k)qL + \left\lfloor \frac{t\mu}{q} - (n-k)\frac{\mu}{P} \right\rfloor qL \right) \Big/ T\,, \qquad (7.5c,\ 7.5d)$$

in which, $n^* = \lfloor \mu T / nq \rfloor$.

The system annual holding cost can thus be computed as

$$H = H_v + H_d + H_b\,. \qquad (7.6)$$

7.3.4 The expected annual backorder cost C_b

The expected annual backorder cost C_b can be written as:

For Case 1: $t_{l1}^{I} = 0 < t \le t_{u1}^{I} = q/P + L$,

$$C_b = c_b \Bigg\{ n^* \Bigg[E\left(X_1 - \frac{q\mu}{P} - \mu L - s\sigma\sqrt{L + \frac{q}{P}} \right)^+ + \sum_{i=0}^{n-k-2} E\left(X_2 - (i+1)q + \frac{iq\mu}{P} - s\sigma\sqrt{L + \frac{q}{P}} \right)^+$$

$$+ E\left(X_3 - (n-k)q + (n-k-1)\frac{q\mu}{P} - s\sigma\sqrt{L + \frac{q}{P}} \right)^+ (k-1)$$

$$+ E\left(X_4 - (n-k)q + (n-k-1)\frac{q\mu}{P} - s\sigma\sqrt{L + \frac{q}{P}} \right)^+ \Bigg]$$

$$+ E\left(X_5 - \frac{q\mu}{P} - \mu L - s\sigma\sqrt{L + \frac{q}{P}} \right)^+ \Bigg\} \Big/ T\,, \qquad (7.7a)$$

where, $n^* = \lfloor \mu T / nq \rfloor$. The random variable X_1 is the demand during the period $q/P + L$, which is the time window from the start of a new production cycle to the arrival of the first shipment to the buyer. X_2 is the demand during q/P. X_3 is the demand during q/μ. X_4 is the demand during $(n-k)q/\mu - (n-k)q/P - L$, which is the time window between the arrival of the last shipment and the end of the production cycle. X_5 is the demand during the last incomplete production cycle under Case 1. Note that in the expression (7.7a),

$E(\cdot)^+ = \max\{0, E(\cdot)\}$. As a result, its value could be zero or above zero under different situations, depending on the value of the random variable X_1, X_2, X_3, X_4, and X_5. Also note that the expression (7.7a) is a little complicated and the number of terms in it varies depending on the parameter values n and k. However, it cannot be further simplified by moving the constant portions inside the expectation brackets to the outside.

For Case 2: $t_{l2}^I = q/P + L < t \le t_{u2}^I = (n-k+1)q/P + L$,

$$C_b = c_b \left\{ \left[(n^*+1)E\left(X_1 - \frac{q\mu}{P} - \mu L - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ + n^* \sum_{i=0}^{n-k-2} E\left(X_2 - (i+1)q + \frac{iq\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ \right.\right.$$

$$+ n^* E\left(X_3 - (n-k)q + (n-k-1)\frac{q\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ (k-1)$$

$$\left. + n^* E\left(X_4 - (n-k)q + (n-k-1)\frac{q\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ \right]$$

$$+ \sum_{i=0}^{\lfloor \tau P/q \rfloor} \frac{i}{\max(1,i)} E\left(X_2 - (i+1)q + \frac{iq\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+$$

$$\left. + E\left(X_6 - (\lfloor \tau P/q \rfloor + 2)q + \frac{(\lfloor \tau P/q \rfloor + 1)q\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ \right\} \bigg/ T, \tag{7.7b}$$

where, $n^* = \lfloor \mu T / nq \rfloor$ and $\tau = t - L - q/P$. The random variable X_6 is the demand during the period $\tau - \lfloor \tau P/q \rfloor q/P$, which is the duration of the last shipment in this case.

For Cases 3 and 4: $t_{l3}^I = (n-k+1)q/P + L < t \le t_{u4}^I = nq/\mu$,

$$C_b = c_b \left\{ (n^*+1)E\left(X_1 - \frac{q\mu}{P} - \mu L - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ + (n^*+1) \sum_{i=0}^{n-k-2} E\left(X_2 - (i+1)q + \frac{iq\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+ \right.$$

$$+ \left[n^*(k-1) + \left\lfloor \frac{\tau'\mu}{q} \right\rfloor \right] E\left(X_3 - (n-k)q + (n-k-1)\frac{q\mu}{P} - s\sigma\sqrt{L+\frac{q}{P}} \right)^+$$

$$+E\left(X_4-(n-k)q+(n-k-1)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right)^+$$

$$+E\left(X_7-(n-k)q+(n-k-1)\frac{q\mu}{P}-s\sigma\sqrt{L+\frac{q}{P}}\right)^+\Bigg\}\Bigg/T\ , \qquad (7.7c,\ 7.7d)$$

where, $n^*=\lfloor \mu T/nq\rfloor$ and $\tau'=t-L-(n-k)q/P$. The random variable X_7 is the demand during the period $\tau'-\lfloor \tau'\mu/q\rfloor q/\mu$, which is the time window from the arrival of the last shipment to the moment of obsolescence.

7.3.5 The expected annual extra space cost C_o

Again, associated with four cases, the expected annual extra space cost can be given as:

For Cases 1 and 2: $t_{l1}^I=0<t\le t_{u2}^I=(n-k+1)q/P+L$, there is no extra space cost since the maximum inventory level is not reached. Hence, the expected annual extra space cost for Cases 1 and 2 can be written as:

$$C_o=0\ . \qquad (7.8a,\ 7.8b)$$

For Cases 3 and 4: $t_{l3}^I=(n-k+1)q/P+L<t\le t_{u4}^I=nq/\mu$,

$$C_o=\frac{c_o}{T}\left(kn^*+\frac{\tau'\mu}{q}\right)E\left[(n-k+1)q-U-(n-k-1)q\mu/P+s\sigma\sqrt{L+\frac{q}{P}}-X_3\right]^+, \quad (7.8c,\ 7.8d)$$

where, $n^*=\lfloor \mu T/nq\rfloor$ and $\tau'=t-L-(n-k)q/P$.

7.3.6 The expected annual obsolescence cost C_{ob}

The expected annual obsolescence cost C_{ob} can be calculated as:

For Case 1: that is, when $t_{l1}^I=0<t\le t_{u1}^I=q/P+L$:

$$C_{ob}=p_vP\left(t-\left\lfloor\frac{tP}{q}\right\rfloor\frac{q}{P}\right)+p_b\left(\frac{q\mu}{P}+\mu L+s\sigma\sqrt{L+\frac{q}{P}}-\mu t\right)+p_bq\left\lceil\frac{Max(0,t-q/P)}{L}\right\rceil. \qquad (7.9a)$$

In expression (7.9a), the first term is the obsolescence cost incurred at the vendor; the second term is that of the buyer's; and the last term is the obsolescence cost in transit. Inside the third term, the expression $\left\lceil \frac{Max(0, t - q/P)}{L} \right\rceil$ is a determinant whose value is one when there is in transit inventory at the moment that obsolescence occurs and is zero when there is no in transit inventory at that moment.

For Case 2: $t_{l2}^{I} = q/P + L < t \le t_{u2}^{I} = (n-k+1)q/P + L$,

$$C_{ob} = p_v P\left(t - \left\lfloor \frac{tP}{q} \right\rfloor \frac{q}{P}\right) + p_b\left[\left\lfloor \frac{\tau P}{q} \right\rfloor \frac{P-\mu}{P} q + q - \mu\left(\tau - \left\lfloor \frac{\tau P}{q} \right\rfloor \frac{q}{P}\right)\right]$$
$$+ p_b q \left\lceil \frac{Max\left(0, L + \lfloor tP/q \rfloor q/P - t\right)}{L} \right\rceil, \tag{7.9b}$$

where, $\tau = t - L - q/P$. The expression $\left\lceil \frac{Max\left(0, L + \lfloor tP/q \rfloor q/P - t\right)}{L} \right\rceil$ is used to determine whether or not there is in transit inventory at the moment of obsolescence.

For Case 3: $t_{l3}^{I} = (n-k+1)q/P + L < t \le t_{u3}^{I} = nq/P$,

$$C_{ob} = p_v\left[\frac{(P-\mu)q}{\mu}\left\lfloor \frac{\mu t}{q} - \frac{(n-k)\mu}{P} \right\rfloor + P\left(t - \frac{n-k}{P} q - \left\lfloor \frac{\mu t}{q} - \frac{(n-k)\mu}{P} \right\rfloor \frac{q}{\mu}\right)\right]$$
$$+ p_b\left[(n-k)q - (n-k-1)\frac{q\mu}{P} + s\sigma\sqrt{L + \frac{q}{P}} - \mu\left(\tau' - \left\lfloor \frac{\tau' \mu}{q} \right\rfloor \frac{q}{\mu}\right)\right]$$
$$+ p_b q \left\lceil \frac{Max\left[0, t - t_{l3}^{I} - \left\lfloor \left(t - t_{l3}^{I}\right)\mu/q \right\rfloor q/\mu - q/\mu + L\right]}{L} \right\rceil, \tag{7.9c}$$

where, $\tau' = t - L - (n-k)q/P$. The expression $\left\lceil \frac{Max\left[0, t - t_{l3}^{I} - \left\lfloor \left(t - t_{l3}^{I}\right)\mu/q \right\rfloor q/\mu - q/\mu + L\right]}{L} \right\rceil$ is used to determine whether or not there is in transit inventory at the moment of obsolescence.

For Case 4: $t_{l4}^{I} = nq/P < t \le t_{u4}^{I} = nq/\mu$,

$$C_{ob} = p_v Max\left(0, kq - \left\lfloor \frac{t\mu}{q} - \frac{n-k}{P}\mu \right\rfloor q\right)$$

$$+ p_b\left[(n-k)q - (n-k-1)\frac{q\mu}{P} + s\sigma\sqrt{L + \frac{q}{P}} - \mu\left(t - t_{l3}^I - \left\lfloor \frac{t - t_{l3}^I}{q}\mu \right\rfloor q/\mu\right)\right]$$

$$+ \left\lceil \frac{Max\left[0, k - \left(t - t_{l3}^I\right)\mu/q\right]}{n} \right\rceil p_b q \left\lceil \frac{Max\left[0, t - t_{l3}^I - \left\lfloor \left(t - t_{l3}^I\right)\mu/q \right\rfloor q/\mu - q/\mu + L\right]}{L} \right\rceil, \qquad (7.9d)$$

where, the expression $\left\lceil \frac{Max\left[0, t - t_{l3}^I - \left\lfloor \left(t - t_{l3}^I\right)\mu/q \right\rfloor q/\mu - q/\mu + L\right]}{L} \right\rceil$ is still used to judge whether or not there is in transit inventory at the moment of obsolescence and the expression $\left\lceil \frac{Max\left[0, k - \left(t - t_{l3}^I\right)\mu/q\right]}{n} \right\rceil$ is used to judge whether or not all the shipments have arrived to the buyer. After that point, the value of the expression becomes zero, meaning there would be no more inventory in transit.

Finally, the annual joint total expected cost $JTEC(q,n,k,s,L)$ for given $L \in (L_i, L_{i-1})$ and $q \in (q_{l-1}, q_l)$ can be written as

$$JTEC(q,n,k,s,L) = C_s^v + C_L + H + C_b + C_o + C_{ob}. \qquad (7.10)$$

The four-factor CS model can, therefore, be written as:

Min $\quad JTEC(q,n,k,s,L)$, (7.11)

Subject To: $\quad (n-k)q - (n-k-1)q\mu/P + s\sigma\sqrt{L + q/P} \le U$, (7.11a)

$k \le n$, (7.11b)

$(n-k)q/P + L + kq/\mu \le nq/\mu$, (7.11c)

q, n, k, L are positive and integers. (7.11d)

The problem is to jointly decide the optimal ordering quantity *q*, safety factor *s*, number of shipments within a production cycle *n*, number of delayed shipments *k*, and lead time *L* that minimize the *JTEC* as expressed by Equation (7.11), under the constraints in (7.11a), (7.11b) and (7.11c). In which, the constraint in formula (7.11a) ensures that the buyer's space limitation is greater than the buyer's potential maximum inventory level, and the constraint in formula (7.11d) ensures the arrival of all shipments to the buyer within each full production cycle.

Note that the general four-factor CS model given by Equation (7.11) doesn't assume any specific distribution for the random demand. However, in order to perform a numerical study, a specific distribution is needed so that the expected backorder cost and the expected extra space cost can be evaluated. In this study, a demand that follows normal distribution is used as an example to demonstrate the generic solution methodology that can be applied for any type of demand distributions. In particular, the yearly demand rate is assumed to follows a normal distribution with mean μ and standard deviation σ. It follows that the random variables X_1, X_2, … X_7, are all normally distributed since they all represent the demand during some certain period of time. According to the length of time each of these random variable represents, their mean and standard deviation can be decided. The list can be seen in Section 7.2.

As shown by Appendix A-7, that if a normal random variable X has mean μ and standard deviation σ, i.e., $X \sim N(\mu,\sigma)$, then $E(X-r)^+$ represents the expected number of

occurrences that X goes beyond the value of r. Its value can be expressed by

$$E(X-r)^{+}=\bar{b}(r)=\sigma[v\Phi(v)+\phi(v)-v], \tag{7.12}$$

where $v=(r-\mu)/\sigma$, the notation $\phi(\cdot)$ is the probability density function (pdf) of the standard normal distribution, and the notation $\Phi(\cdot)$ is the cumulative density function (cdf) of the standard normal distribution.

Similarly, it can be shown that

$$E(r-X)^{+}=\sigma[v\Phi(v)+\phi(v)]. \tag{7.13}$$

Therefore, substituting Equation (7.12) and Equation (7.13) into Equation (7.7) and Equation (7.8), and use the standard normal table, the expected annual backorder cost, the expected annual extra space cost can be evaluated, and finally, the joint total expected cost of the four-factor CS model can also be evaluated.

The four-factor CS model developed in this study is a constrained nonlinear mixed-integer optimization problem. Similar to Persona, *et al.* (2005), Battini, *et al.* (2010a, 2010b) models, the cost function of the four-factor CS model is also characterized by several discontinuity points. As the result, the five decision variables that minimize the total expected cost cannot be evaluated by deriving Equation (7.11) (Persona, *et al.*, 2005). Moreover, because of the number of variables considered in the model, it is difficult to prove the convexity of the objective function. Also, due to the stochastic nature of the decision variable, traditional optimization methods that rely on differentiation may not be appropriate to solve the two models. Therefore, a newly developed doubly-hybrid meta-

heuristic algorithm (DHMHA) (Yi, *et al.*, 2013), which has been shown by Yi and Sarker (2013b, and 2013c) to be a satisfactory tool to solve complicated inventory models with multiple variables, is adopted in the next Section to locate the solutions. Specifically, we used the doubly-hybrid meta-heuristic algorithms of Particle Swamp algorithm, Harmony Search algorithm, and Hooke and Jeeve local search method (PSO+IHS+HJ), and the hybrid of Harmony Search algorithm, Deferential Evolution algorithm, and Hooke and Jeeve's local search methods (MDE'+IHS+HJ) to conduct the calculation and to perform numerical analyzes. Interested readers may refer to Yi, *et al.* (2013) and Yi and Sarker (2013b and 2013c) for details.

7.4 COMPUTATIONAL RESULTS

In this Section, a numerical example is first used to demonstrate the developed algorithms. It is then followed by a sensitivity analysis to show the parametric evaluation of the systems. The outcomes of the numerical example, found the two doubly-hybrid algorithms, are compared to that of an exhaustive search algorithm (ESA) to verify that the solutions found by the doubly-hybrid are global optimum. Because of the complexity of the model, it is not feasible to use the ESA to get a solution. Hence, only the DHMHAs are used to perform the analysis of the important parameters in the model. All the algorithms were coded in Matlab and were executed on a HP Pavilion Dv8 notebook PC with an Intel® QuadCore i7 CPU and Q 720@ 1.6 GHz processor. For each setting of the test parameters, 10 runs were made by both doubly-hybrid algorithms (PSO+IHS+HJ and MDE'+IHS+HJ).

A run is declared as ‘successful’ when the global optimum (or the best-known) was found within 10^{-6} error.

A numerical example is framed here to illustrate the optimal solutions (ESA) along with PSO+IHS+HJ and MDE’+IHS+HJ. Most of the values of the parameters used in this example are adopted from Braglia and Zavanella (2003), and Huang and Chen (2009). For example, the lead time in our study also consists of three segment the composition of which is shown on Table 4.1 [the original time data are in days].

As mentioned earlier in Section 7.2, the unit lead time crushing cost is a function of both the reduced period and the shipping size. In this example, a unified quantity discount is used to represent the relationship between the lead time crushing cost and the quantity. Table 4.2 illustrates the detailed composition of this cost.

In the models developed, all other decision variables are integers except for the safety factor *s*, which is continuous. Two decimal values are allowed for *s* so that the ESA can be used to determine the optimal solutions and verify whether the solutions found by the two doubly-hybrid meta-heuristic methods are global optimal. Table 7.1 shows the optimal solutions of the decision variables and expected system cost found by the three algorithms.

It is observed that all the three algorithms lead to the same solutions, which is guaranteed to be the global optimum since the ESA is used as a comparison. The ESA requires more than 30 days to find the optimal solutions (exhaustively), whereas the two doubly-hybrid approaches take a few minutes. This result shows that the accuracy of both

the hybrid algorithms is as good as that of the ESA, but the efficiency of the doubly-hybrid algorithms are superior.

Table 7.1 Optimal solutions of the three algorithms

Algorithm	*JTEC*($/yr)	*n*	*q*	*k*	*L* *(day)*	*s*	CPU/M_CPU (s)	NFE/M_NFE	Suc_ Rate
ESA	3,297.64	123	3	87	56	1.90	3,199,302.72	1,023,814,404	-
PSO+IHS+HJ	3,297.64	123	3	87	56	1.90	432.89	100,131	0.50
MDE'+IHS+HJ	3,297.64	123	3	87	56	1.90	258.80	100,252	0.60

The four-factor CS model is developed on top of the three-factor CS model 1. The main different between this two models is that the planning horizon of the former is finite, whereas the later one is not. Moreover, the last production cycle of the newer model might be of great different than that of the older one. It is, thus, interesting to compare the outcomes of the two models since the comparison may provide the decision maker with some insights on what changes the new factors considered may bring. Comparing the optimal solutions of the four-factor CS model with that of the three-factor CS model, i.e., Table 7.1 versus Table 6.3, it is obviously that the total cost of the four-factor CS model is smaller (3297.64 vs. 4480.22). This is mainly because of a new assumption, the buyer's ordering cost is eliminated under the CS agreement (Valentini and Zavanella, 2003, and Battini, *et al.*, 2010a, 2010b). Under the new assumption, one part of the system cost is directly removed. Furthermore, a replenishment policy that takes a higher number of orders is also encouraged.

Another noticeable difference between the solutions of the two models is that the new solution tends to have a high number of shipments (n = 123) and a small shipping size (q = 3), whereas the old optimal solution tends to have a higher shipping size (q = 85) and a small number of shipments (n = 5). This may be explained partly by the removal of ordering cost of the new model, and partly by the newly considered obsolescence cost. The more inventory at hand, the more lost to the system when the obsolescence occurs.

Due to the fact that the ESA is too time consuming, the two doubly-hybrid approaches are used in the remainder of this Section to conduct the analysis.

7.5 SENSITIVITY ANALYSIS

There are several important parameters in our model such as the buyer space limitation U, the obsolescence occurring time T, the unit penalty for backorders c_b and for over the space limitation c_o, and the quantity discount coefficient c_i. A change in their values might appreciably influence the optimal solutions. The combination of the values of h_v^s and h_b^s is also important, past research has shown that $h_v^s > h_b^s$ or $h_v^s < h_b^s$ may bring different solutions. We, therefore, discuss the effects of these important parameters to the system in this Section.

7.5.1 Effects of obsolescence timing *T*

The timing when obsolescence will occur is one of the most important parameters of the four-factor CS model, since it is the only parameter that the new factor, the obsolescence,

brings to the system. Therefore, it is necessary to study its effects to the entire system. Table 7.2 and Figure 7.6 illustrate this effect.

Note that the last column of Table 7.2 shows the translated safety stock level S_s instead of the safety factor s. This is because it is more meaningful and easier for the manager to manipulate the safety stock level than the safety factor. The relation between S_s and s is given by ceiling operator as $S_s = \left\lceil s\sigma\sqrt{L+q/P} \right\rceil$.

Table 7.2 Effects of T on the system

T(year)	$JTEC$($/yr)	n	q	k	L(day)	S_s
1	3,677.34	106	4	79	55	75
2[a]	**3,297.64**	**123**	**3**	**87**	**56**	**75**
3	3,329.68	79	6	60	56	70
4	3,397.59	148	2	90	55	70
5	3,293.25	147	3	110	54	73
6	3,275.84	150	3	111	54	69
7	3,247.78	107	4	78	55	69
8	3,233.42	109	4	80	55	69
9	3,130.25	134	3	97	56	73
10	3,248.26	107	4	80	56	74

a: This row was also shown in Table 7.1

It can be seen that both the *JTEC* and all the decision variables are sensitive to the value of T. However, none has an apparent relation with T. In general, the *JETC* is relatively lower when T is longer. The shipping size and the lead time are less sensitive to T. This irregular characteristic is probably due to the discontinuous nature of the obsolescence timing. Because there is no obvious relation between the decision variables and the obsolescence timing, it has, therefore, no obvious optimal decisions for all possible obsolescence timing. How the operational policies perform depends on how precise is the

prediction of *T*. This result shows how important the four-factor model is to the decision maker, since there is no rule-of-thumb policy and the manager can only rely on the results of the model to make their decisions.

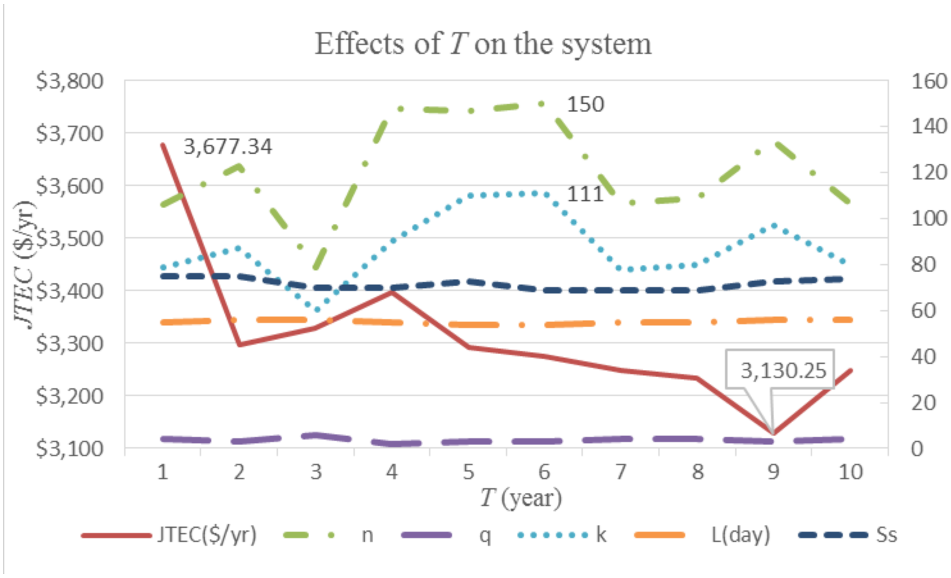


Figure 7.6 Effects of the obsolescence time *T* to the system

7.5.2 Effects of buyer space limitation *U*

The buyer space limitation *U* is an influential parameter of the entire system. All operational decisions and system cost will change according to different values of *U*. Table 7.3 and Figure 7.7 illustrate the results of both hybrid algorithms for the four-factor CS model with the value of *U* changing from 50 to 500.

In general, the system is sensitive to changes in *U*. The total cost and most of the decision variables are monotonic functions of *U* except for the safety stock level. *n* and *k* have a positive relation with *U*, whereas the *JTEC* decreases as the *U* increases. It is interesting that the *q* also has a negative relationship with *U*. However, all these

relationships cease as the U goes beyond 250; that is to say, the buyer's inventory level will never be above that level. This finding is useful in that it provide the decision makers a quantitative idea about how to set the suitable space limitation.

Table 7.3 Effects of U to the system

U	$JTEC$($)	N	q	k	L(day)	S_s
50	4,073.03	41	9	40	55	42
100	3,649.40	77	4	69	55	77
150[a]	**3,297.64**	**123**	**3**	**87**	**56**	**75**
200	3,253.57	154	3	92	55	72
250	3,181.64	205	3	132	56	72
300	3,181.64	205	3	132	56	72
350	3,181.64	205	3	132	56	72
400	3,181.64	205	3	132	56	72
450	3,181.64	205	3	132	56	72
500	3,181.64	205	3	132	56	72

a: This row was also shown in Table 7.1

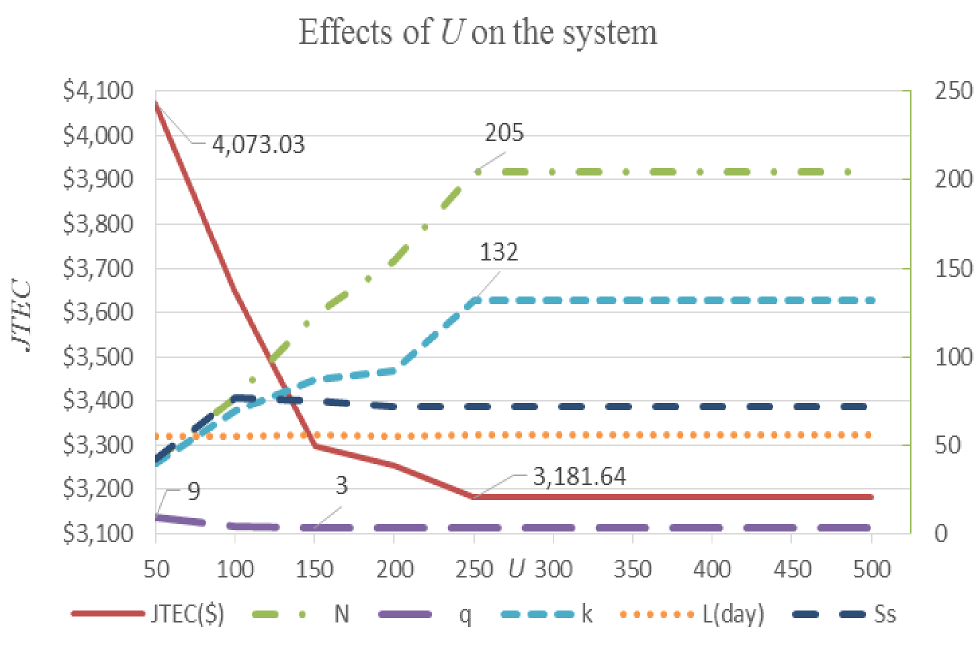


Figure 7.7 Effects of the buyer's space limitation U to the system

7.5.3 Effects of the back order penalty coefficient c_b

Another important parameter in the system is the back order penalty coefficient c_b. A change to c_b will directly change the *JTEC* and the safety factor *s*, and all other decision variables will vary, accordingly. Table 7.4 and Figure 7.7 show the effects of c_b on all decision variables and on the objective function *JTEC*.

Table 7.4 Effects of c_b on the system

c_b($/unit)	*JTEC*($/yr)	*n*	*q*	*K*	*L*(day)	S_s
5	3,116.83	123	3	62	56	24
10	3,179.28	123	3	71	56	42
15	3,211.82	123	3	76	56	53
20	3,233.65	123	3	79	56	59
25	3,249.90	123	3	81	56	63
30	3,262.93	123	3	82	56	65
35	3,273.51	123	3	84	56	69
40	3,282.59	123	3	85	56	71
45	3,290.50	123	3	86	56	73
50	**3,297.64**	**123**	**3**	**87**	**56**	**75**

a: This row was also shown in Table 7.1

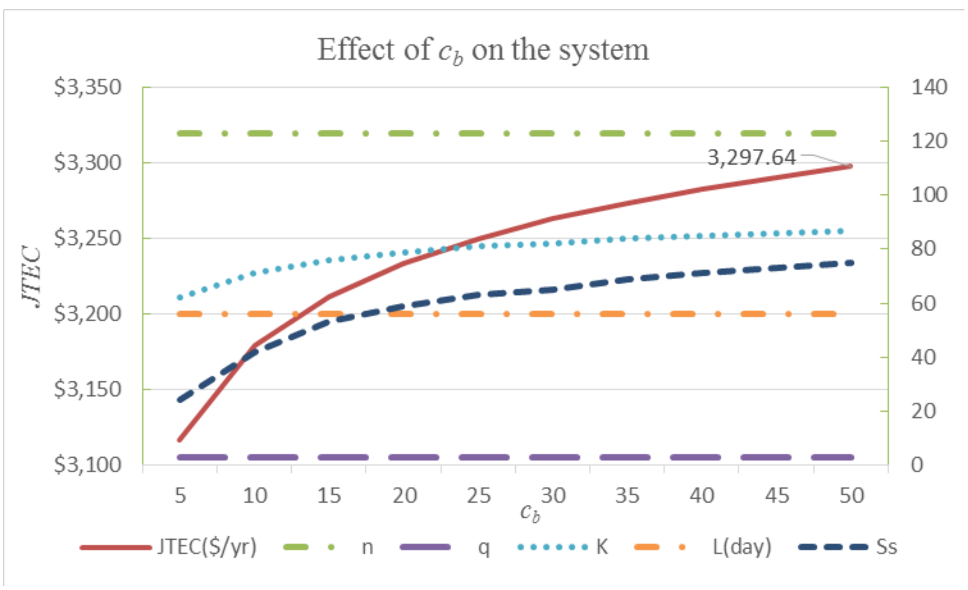


Figure 7.8 Effects of c_b on the system

Unlike the three-factor CS model, in which the c_b seems to have a positive relationship with the *JTEC* and *s*, but have no obvious relationship with *n*, *q*, and *k*. The four-factor CS model does show an apparent relation among c_b and the *JTEC* and the decision variables. In particular, the *JTEC*, *k*, and S_s all seem to be monotonically increasing as c_b increases. On the contrary, the *n*, *q*, and *L* are not sensitive to the changes in c_b.

It can be seen that both *JTEC*, S_s, and *k* increases as c_b increases. However, the *JTEC* increases more rapidly than the other two. Moreover, the increasing rate of *JTEC* also increases when the value of c_b is lower from 5 to 20. After that, the increasing rate seems to stabilize. This analysis also helps find out the range of c_b within which the system is sensitive.

7.5.4 Effects of the extra space penalty coefficient c_o

As shown in Chapter 6, the extra space penalty coefficient c_o is also important. The study of its effects to the system is important and is illustrated by Table 7.5.

Table 7.5 Effects of c_o to the system

c_o($/unit)	*JTEC*($/yr)	*n*	*q*	*k*	*L*(day)	S_s
5	3,297.48	123	3	87	56	75
10	**3,297.48**	**123**	**3**	**87**	**56**	**75**
15	3,297.48	123	3	87	56	75
20	3,297.48	123	3	87	56	75
25	3,297.48	123	3	87	56	75
30	3,297.48	123	3	87	56	75
35	3,297.48	123	3	87	56	75
40	3,297.48	123	3	87	56	75
45	3,297.48	123	3	87	56	75
50	3,297.48	123	3	87	56	75

a: This row was also shown in Table 7.1

Unlike the three-factor CS model, the new model developed in this Chapter does not change for all the changes in c_o, since the optimal policy of the four-factor CS model is characterized by a small shipping size and a large number of shipments. This policy makes the system needless extra storage space.

7.5.5 Effects of the lead time crushing cost coefficient c_i

As mentioned in Section 6.5.4, the value of the unit lead time crushing cost used to conduct the numerical example is somewhat high, which makes the optimal lead time always at its highest possible value under most circumstances. Therefore, we tested again the lead time crushing cost coefficient c_i and the results are shown by Table 7.6 and Figure 7.8.

Table 7.6 Effects of c_i to the system

c_i(%)	*JTEC*($/yr)	*n*	*q*	*k*	*L*(day)	S_s
0.1	3,139.22	125	3	85	45	67
0.2	3,249.27	125	3	85	45	67
0.3	3,350.75	116	4	82	52	56
0.4	3,362.52	116	4	87	52	69
0.5	3,297.64	123	3	87	56	75
0.6	3,297.64	123	3	87	56	75
0.7	3,297.64	123	3	87	56	75
0.8	3,297.64	123	3	87	56	75
0.9	3,297.64	123	3	87	56	75
1	**3,297.64**	**123**	**3**	**87**	**56**	75

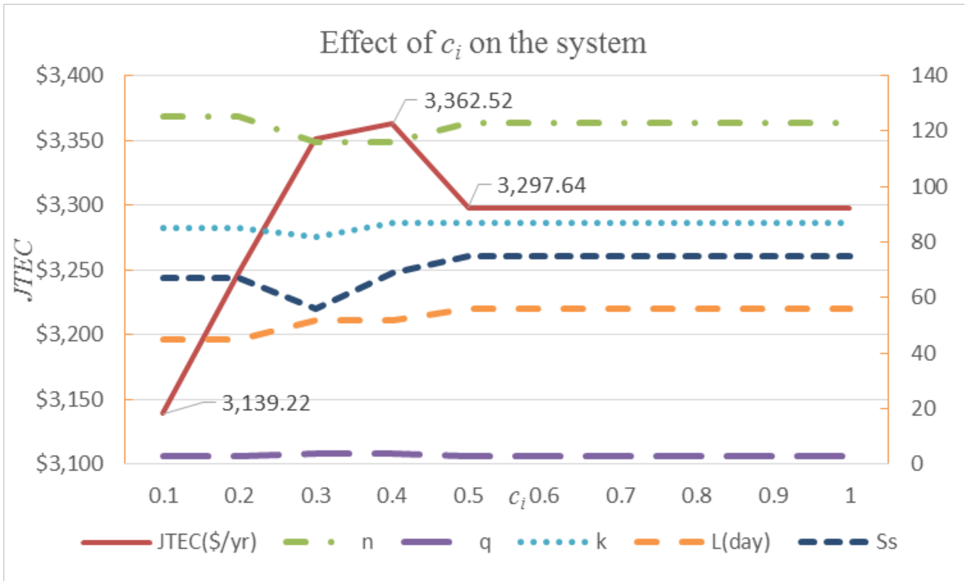


Figure 7.9 Effects of c_i on the system

It can be seen that when the additional discount coefficient c_i is less than 0.4, the doubly hybrid algorithms suggest reducing the lead time from 56 days to 52 days, then to 45 days.

7.6 CONCLUSION

This Chapter studies the effects of introducing the risk of obsolescence into an integrated lead time controllable consignment stock inventory system with buyer's space limitation and stochastic demand. A four-factor inventory model is developed to jointly determine the optimal value of five decision variables that minimize the annual *JTEC* of the system. Due to the complexity of the problem structure, analytical solutions are not presented. Instead, two novel doubly-hybrid meta-heuristic algorithms are utilized to find the global optimum of the models. Numerical examples showed that both the doubly-hybrid algorithms perform well both in the sense of the CPU time and the success rate. The

computational analysis revealed how the model with the risk of obsolescence performs differently than in the previous ones and also discloses how some important parameters affect the solutions and the objective functions. The results obtained in this Chapter help understand the role of obsolescence, stochastic demand rate, the buyer's space limitation, controllable lead time, and the CS mechanism. Moreover, the successful use of doubly-hybrid meta-heuristic algorithms to inventory problems provides a possible way of solving more difficult and complicated models in the future.

CHAPTER VIII
GENERAL DISCUSSION AND CONCLUSIONS

The motivation of this dissertation is to fill the gap between the current research on integrated CS system and the realistism in the following three ways: (1) relax some unrealistic assumptions that most current research made, (2) provided a general solution approach for many different CS inventory models, and (3) simutaneously consider multiple decision variables in one model. Toward this end, this research presented three improved doubly-hybrid meta-heuristic algorithms and developed four sets of CS models, each of which considered one more realistic factor than previous studies. This Chapter concludes the dissertation with a summary of the models developed, a brief research results and conclusions, and a short discusion on portential future studies.

8.1 SUMMARY OF THE FOUR SETS OF CS MODELS

This study, based upon the general CS-k model defined by Braglia and Zavanella (2003), improves the current research on integrated CS inventory system by: (1) provided three doubly-hybrid meta-heuristic methods as a possible choice of solution approach for many different inventory models; (2) relaxed four unrealistic assumptions that most current literature made: zero and uncontrollable lead time, infinite storage space at the buyer, deterministic demand, and infinite product life cycel, respectively; and (3) developed four sets of mathematical models in Chapters 4, 5, 6 and 7.

The later Chapters are the logical extensions of the former ones that consider more important and practical issues which have never been considered before or have been tried

by previous researchers, but their results/methods are insufficient. Compared to earlier ones, the models developed here are more complex, more realistic, and considers larger numbers of decision variables. Table 8.1 briefly summarizes the major issues and system parameters that are addressed here.

Initially, the ordering quantity, q, is the only decision variable in the basic CS-K model. Chapter 4 included the number of shipments within each production cycle, n, and the lead time, L, as two other decision variables. This one-factor CS model is then divided into two cases by adding another assumption: the unit holding cost is lower/higher at the vendor. Associated with these two cases, two mathematical models are developed and are solved analytically. Because the decision variables are all integers, a procedure to identify the optimal integer solutions starting from the optimal real solutions is also provided.

In Chapter 5, another important factor that often faced in reality by most of the companies along the supply chain, the buyer's space limitation, is taken into consideration. As a result, the number of delayed shipments, k, is added into the model as the fourth decision variable. Other than the traditional replenish policy that the basic CS-k model adopted, a modified replenishment policy that makes more usage of the buyer's space limitation is introduced. The two new models developed based on the two replenishment policies are difficult to solve analytically. As a result, the doubly-hybrid meta-heuristic method introduced in Chapter 3 are used to solve them.

One unrealistic assumption of former models, the deterministic demand rate, is relaxed in Chapter 6. An uncertain demand rate that follows a normal distribution is considered. It follows that the fifth decision variable, the safety factor s, is introduced into the two new models. The backorder penalty and extra space penalty are all taken into consideration in this Chapter to quantify the impacts of the uncertainty. To make the model even more realistic, the unit lead time crushing cost is considered to be a function of both n and q, instead of only a function of n.

Finally, the assumption of an infinite product infinite life cycle is seldom the case in practice. The finite life cycle of the product, characterized by obsolescence, is discussed in detail in Chapter 7. Based on the three-factor CS model developed in Chapter 6, a four-factor CS model is formulated. The quantitative effects of the planning horizon T, together with some other important system parameters, are illustrated in this Chapter with numerical examples.

Table 8.1 Summary of the four CS models developed in this research

Descriptors		Chapter 4 1-factor model	Chapter 5 2-factor model	Chapter 6 3-factor model	Chapter 7 4-factor model
Number of Decision Variables		Three: n, q, L	Four: n, q, k, L	Five: n, q, k, L, s	Five: n, q, k, L, s
Factors considered	Controllable lead time	Yes	Yes	Yes	Yes
	Buyers space limitation	No	Yes	Yes	Yes
	Deterministic/ Stochastic demand?	Deterministic	Deterministic	Stochastic	Stochastic
	Obsolescence	No	No	No	Yes
Assumptions	Lead time crushing cost is	A function of n	A function of n	A function of n and q	A function of n and q
	Shortage allowed?	No	No	Yes	Yes
	Extra-space penalty allowed?	No	No	Yes	Yes
	Back order penalty considered?	No	No	Yes	Yes
	Safety stock considered?	No	No	Yes	Yes
Model structure	Number of new models	One	Two	Two	One
	General CS-k model	1-factor model partitioned into two cases	2-factor model and modified 2-factor model	3-factor model and modified 3-factor model	4-factor model
Solution method	Closed-form solution?	Yes	No	No	No
	Meta-heuristic solution?	No	Yes	Yes	Yes
	Integralization method used?	Yes	No	No	No
Comparison of models developed	Which model is better (original or modified)?	Original model (one)	Depends on unit holding costs at the vendor and/or buyer	Depends on unit holding cost, extra-space penalty, back order penalty, and safety inventory levels	Compared 4-factor model with 3-factor model
Discussion of key parameters	Effects of backorder cost discussed?	No	No	Yes	Yes
	Effects of extra-space cost discussed?	No	No	Yes	Yes
	Effects of lead time crushing cost discussed?	Yes	Yes	Yes	Yes
	Effects of product life cycle discussed?	No	No	No	Yes

n : Number of deliveries/Batch, q : Shipping size, k : Number of delayed shipments, L : Lead time, s : Safety factor.

8.2 RESEARCH RESULTS

All the three improved hybrid meta-heuristics, proved by using 18 different engineering problems, improve over the two hybrids described in Liao (2010) in terms of average success rate while taking less average elapsed CPU time. Among these three hybrids, the overall performance of MDE'-IHS-HJ is the best, better than the best hybrid in Liao (2010) by more than 15% and better than the second best new hybrid, PSO-MDE'-HJ by nearly 10%. It was also shown that for some problems the performance of those new hybrids can be further improved by increasing the maximal number of function evaluations.

The three variable one-factor CS model (both under Cases 1 and 2) reveal the quantitative impact of the controllable lead time. The models are sensitive to the ratio D/P when D is close to P but is not sensitive to the ratios A_b/A_v and h_b^s/h_v^s. Also, the optimal lead time is found to be sensitive to the unit lead time crushing cost c_i.

The four variable two-factor CS model and the modified two-factor CS model were shown to be sensitive to the value of the buyer's space limitation when the value is under a certain level. This is useful for the decision maker at a planning phase. Specifically, the total cost, the number of shipments in a cycle, and the number of delayed shipments are all piecewise decreasing as the space limitation getting freer, whereas the shipping size is piecewise increasing. Moreover, the modified two-factor CS model is better than the two-factor CS model when the unit holding cost is lower to the buyer.

The five variable three-factor CS model and the modified three-factor CS model mainly reveal the impact of a stochastic demand rate to the system. In fact, because of the uncertainty in demand, the joint total cost increased for both the three-factor CS model and the modified three-factor CS model. The numerical example also shows that the modified three-factor CS model performs better when the space limitation is tight, the backorder penalty is high, or the extra space penalty is below a certain threshold, whereas the original three-factor CS model is better when the conditions are reverse.

The five variable four-factor CS model studies the impact of the obsolescence. The computational results show that the most dominant parameter is the product life cycle time T. The system is sensitive to its variation. However, there is no apparent relationship between them. Unlike the three-factor CS model, the four-factor model is not sensitive to the extra space penalty.

8.3 CONCLUSIONS

This work studied a one-vendor one-buyer integrated consignment stock inventory system. The research developed four sets of models that address the impacts of four important system controlling factors: the controllable lead time, the buyer's space limitation, the variable demand, and the risk of obsolescence to the system. The models developed in the study are quantitative tools that can assist the manager of such a system to make decisions on the optimal ordering quantity, number of shipments within each production cycle, number of delayed shipments within each production cycle, safety stock level, and

the lead time, so that the annual joint total expected cost of the system is minimized. The research also presented three improved doubly hybrid meta-heuristic algorithms that are able to efficiently locate the optimal solutions for complicated, multiple variable, non-linear optimization inventory models. More comprehensive conclusions were made in each Chapter. The most important conclusions are generalized below:

(1) At least two doubly-hybrid meta-heuristic algorithms (MDE'+IHS+HJ and PSO+IHS+HJ) developed here perform well in solving complicated, multiple variable inventory models. They are capable of coping with complicate inventory models with multiple variables and, thus, may be used to solve even more complicated systems.

(2) Controllable lead time is an influential factor that is worth further study. Under what cases it is desirable to consider reducing lead time can be quantified by the one-factor CS model. The lead time crushing cost, as a function of both the shipping size and the duration, is a better approximation.

(3) Buyer's space limitation is another important issue to be considered when making decisions. The two-factor CS model is a tool helping the managers to make operational decisions when there is an upper limitation in the buyer's warehouse.

(4) The modified two-factor CS model performs better than the original two-factor CS model when the unit holding cost is lower to the buyer and the buyer's space limitation is tight.

(5) Under a normally distributed demand rate, the three-factor CS model and the modified three-factor CS model perform differently than the two two-factor CS models. Which one of the three-factor CS model is better depends on different system parameters such as the unit hold cost in the vendor/buyer, the extra space penalty, and the backorder penalty.

(6) The risk of obsolescence is important to the system. The performance of the four-factor CS model depends on how precise is the prediction of the product life cycle.

The models developed in this study are more realistic than previous models. The results generated here are important enhancements to the study of supply chain and inventory management.

8.4 FUTURE RESEARCH

While performing this research, many problems have been encountered and several issues relating to the problem structures and solution methods and improvements have surfaced, the action to which may lead to various research agenda. Such potential research issues that may be further pursued concerning the supply chain system studied here are given two different phases below:

(A) System characteristics and structures:

(1) Variable Production Rate**:** In many cases, the production rate of a manufacturing facility is variable (Glock, 2010, 2011), either due to the changing in the demand or from the internal changes such as the improvement or development in the producing technology.

Therefore, taking a variable production rate into consideration is important. Adding the production rate as a decision variable will enhance the understanding on the integrated CS inventory system.

(2) Multiple Products: Most business transactions between vendor and buyer involve more than one product. Those products may be different in many ways including: their volume, weight, size, stocking requirements, deterioration rate, demand, price etc. Different products bring different profitability. Therefore, how to jointly make the optimal decisions for producing, transporting, storing, and selling multiple independent items, is a concern in industrial situations. Consideration of this issue can make any study more realistic.

(3) Multiple Vendors/Buyers**:** The operational strategies for a multiple vendor/buyer systems are different from that of a system with single vendor and single buyer. They are much more complicated. For example, if there is more than one vendor that can provide the same products, then the vendors are the competitors within themselves. What would be the best policy to satisfy multiple vendors is an interesting research topic. In other cases, there might be transshipments between the vendors when one is temporarily out of stock. In this scenario, one vendor plays the role of a supplier to the other. How the system accommodates transshipments also deserves further study.

(4) Price Discount: When there is backorder, the unsatisfied demand may be backordered or may be lost. In many business practices, the vendor often offers a price discount to the

buyer to encourage the buyer to accept a backorder. How this issue applies to the CS context is also an area to study.

(5) Variable Shipping Size: In reality, according to variable demand, the shipping size needs also to change. Therefore, how the CS policy changes to cope with this issue is important and deserves further study.

(B) Solution methodologies

While searching for potential methods capable of solving different complicated, multi-variable inventory models, a number of possibilities have been encountered. Such possibilities include integer programming, non-linear programming, geometric programming, stochastic programming, etc. Even though some were not appropriate and/or successful, some did seem to be applicable, among other possible ways of acting as general solution methods:

(1) New innovative meta-heuristic methods**:** It is believed that hybrid meta-heuristics research is an area that will attract more attention in the years to come. As it has been shown, this study is successful in finding new hybrids which improve over the previous work. It is possible to come up with even better hybrids than the three presented here. For example, there are continuing research work on the improvement of IHS algorithm by different approaches. One possible topic for future study can be to evaluating the effectiveness of those new IHS-based algorithms and develop more efficient hybrids.

(2) Geometric Programming (GP) Approach: Many CS inventory models, such as the one-factor and two-factor CS models developed here can be categorized into generalized geometric programming problems with non-positive degrees of difficulty. Previously, the GP methods were not be used in solving inventory problems, mainly because there is no satisfactory way of solving these generalized GP models. However, recent studies (Wang, *et al*., 2009, and Li and Lu 2009) present solution methods to solve such kind of problems. These new methods or their variants may be used to solve those integrated CS inventory problems that can be categorized into generalized GP approaches.

The study of the supply chain management under the context of CS policy is still young. Conducting any of the above mentioned research will improve the understanding of the CS model, helping the managers make the best operational decisions under different scenarios.

REFERENCES

1. Al-Sultan, K. S., and Al-Fawzan, M. (1997). A tabu search Hooke and Jeeves algorithm for unconstrained optimization. *European Journal of Operational Research*, *103(1)*, 198-208.

2. Angira, R., and Babu, B. V. (2006). Optimization of process synthesis and design problems: a modified differential evolution approach. *Chemical Engineering Science*, *61*(14), 4707–4721.

3. Aytug, H. K., and Vergara, F. (2003). Use of genetic algorithms to solve production and operations management problems: a review . *International Journal of Production Research*,, *41*(17), 3955-4009.

4. Banerjee, A. (1986). A joint economic-lot-size model for purchaser and vendor. *Decision Science*, *17*(3), 292-311.

5. Battini, D., Grassi, A., Persona, A., and Sgarbossa, F. (2010a). Consignment stock inventory policy: methodological framework and model . *International Journal of Production Research*, *48*(7), 2055-2079.

6. Battini, D., Gunasekaran, A., Faccio, M., Persona, A., and Sgarbossa, F. (2010b). Consignment stock inventory model in an integrated supply chain. *International Journal of Production Research*, *48*(2), 477-500.

7. Ben-Daya, M., and Hariga, M. (2004). Integrated single vendor single buyer model with stochastic demand and variable lead time. *International Journal of Production Economics*, *92*(1), 75-80.

8. Bergh, F. V., and Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *8*(3), 225-239.

9. Blum, C. (2010). Hybrid metaheuristics. *Computers and Operations Research*, *37*(3), 430-431.

10. Bolen, W. H. (1988). *Contemporary Retailing* (3rd ed.). New Jersey.: Prentice Hall, New Jersey.

11. Braglia, M., and Zavanella, L. (2003). Modeling an industrial strategy for inventory management in supply chains: the "Consignment Stock" case. *International Journal of Production Research*, *41*(16), 3793-3808.

12. Brest, J., Greiner, S., Boskovic, B., Mernik, M., and Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, *10*(6), 646-657.

13. Cadenas, J. M., Garrido, M. C., and Muñoz, E. (2009). Using machine learning in a cooperative hybrid parallel strategy of metaheuristics. Information Sciences, 179(19), 3255–3267.

14. Cadenas, J. M., Garrido, M. C., and Muñoz, E. (2011). Facing dynamic optimization using a cooperative metaheuristic configured via fuzzy logic and SVMs,. Applied Soft Computing, 11(8), 5639-5651.

15. Chakraborty, T., and Giri, B. C. (2012). Joint determination of optimal safety stocks and production policy for an imperfect production system. *Applied Mathematical Modeling*, *36*(2), 712-722.

16. Chaudhury, S. S., and Luo, W. (2005). Application of genetic algorithms in production and operations management: a review,". *International Journal of Production Research*, *43*(19), 4083-4101.

17. Chen, J., Lin, I., and Cheng, H. (2010). Channel coordination under consignment and vendor-managed inventory in a distribution system. *Transportation Research Part E*, *46*(6), 831-843.

18. Chen, T. C. (2006). IAs based approach for reliability redundancy allocation problems. *Applied Mathematics and Computation*, *182*(2), 1556-1567.

19. Chen, Z. X., and Sarker, B. R. (2010). Multi-vendor integrated procurement-production system under shared transportation and just-in-time delivery system. *Journal of the Operational Research Society*, *61*(11), 1654-1666.

20. Clerc, M., and Kennedy, J. (2002). The particle swarm: Explosion, stability, and convergence in multidimensional complex space,". *IEEE Transactions on Evolutionary Computation*, *6*(1), 58-73.

21. Corbett, C. (2001). Stochastic inventory systems in a supply chain with asymmetric information: Cycle stocks, safety stocks, and consignment stock. *Operations Research*, *49*(4), 487-500.

22. Coughlan, A., Anderson, E., Stern, L., and El-Ansary, A. (2001). *Marketing Channels* (6th ed.). New Jersey, NJ: Prentice Hall.

23. Crainic, T. G., Gendreau, M., Hansen, P., and Mladenovic, N. (2004). Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, *10(3)*, 293–314.

24. Daniel, J., and Rajendran, C. (2005). A simulation-based genetic algorithm for inventory optimization in a serial supply chain. *International Transactions in Operational Research, 12*(1), 101-127.

25. Das, S., and Suganthan, P. N. (2011). Differential Evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*(1), 27-54.

26. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, *186*(2-4), 311-338.

27. Digalakis, J., and Margaritis, K. (2004). Performance comparison of memetic algorithms. *Applied Mathematics and Computation*, *158(1)*, 237-252.

28. Dimopoulos, G. G. (2007). Mixed-variable engineering optimization based on evolutionary and social metaphors. *Computer Methods in Applied Mechanics and Engineering*, *196(4-6)*, 803-817.

29. Dorigo, M., Caro, G. D., and Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, *5(2)*, 137-172.

30. Duarte, A., Marti, R., Glover, F., and Gortazar, F. (2011). Hybrid scatter tabu search for unconstrained global optimization. *Annals of Operations Research*, *183(1)*, 95-123.

31. Foote, B., Kebriaei, N., and Kumin, H. (1988). Heuristic policies for inventory ordering problems with long and randomly varying lead times,”. *Journal of Operations Management*, *7*(4), 115-124.

32. Geem, Z., Kim, J. H., and Loganathan, G. (2001). “A new heuristic optimization algorithm: harmony search. *Simulation*, *76*(2), 60-68.

33. Gerchak, Y., and Khmelnitsky, E. (2003). A consignment system where suppliers cannot verify retailer's sales reports. *International Journal of Production Economics*, *83(1)*, 37-43.

34. Gerchak, Y., and Wang, Y. (2004). Revenue-sharing vs. wholesale-price contracts in assembly systems with random demand. *Production and Operations Management*, *13*(1), 23-33.

35. Giri, B. C. (2011). Managing inventory with two suppliers under yield uncertainty and risk aversion,. *International Journal of Production Economics*, *133*(1), 80-85.

36. Giri, B. C., and Dohi, T. (2009). Cost-effective ordering policies for inventory systems with emergency order. *Computers and Industrial Engineering*, *57*(4), 1336-1341.

37. Glock, C. (2010). Batch sizing with controllable production rates. *International Journal of Production Research*, *48*(20), 5925–5942.

38. Glock, C. (2011). Batch sizing with controllable production rates in a multi-stage production system. *International Journal of Production Research*, *49*(20), 6017-6039.

39. Glover, F. (1986). Future paths for integer programming and links to artificial-intelligence. *Computers and Operations Research*, *13*(5), 533-549.

40. Glover, F., and Kochenberger, G. (2003). *Handbook of metaheuristics*. Massachusetts: Kluwer Academic Publishers.

41. Golhar, D. Y., and Sarker, B. R. (1992). Economic Manufacturing Quantity in a Just in Time Delivery System. ," *International Journal of Production Research*, *30*(5), 961-972.

42. Goyal, S. K. (1976). An Integrated Inventory Model for a Single Supplier-Single customer Problem. *International Journal of Production Research*, *15*(1), 107-111.

43. Goyal, S. K. (1977). Determination of optimum production quantity for a two-stage production system. *Operational Research Quarterly*, *28*(4), 865-870.

44. Goyal, S. K. (1988). A joint economic lot size model for purchaser and vendor: a comment. *Decision Science*, *19*(1), 236-241.

45. Goyal, S. K. (1995). A one-vendor multi-buyer integrated inventory model: a comment. *European Journal of Operational Research*, *82*(1), 209-210.

46. Goyal, S. K., Huang, C., and Chen, K. (2003). A simple integrated production policy of an imperfect item for vendor and buyer. *Production Planning and Control*, *14*(7), 596-602.

47. Gumus, M., Jewkes, E., and Bookbinger, J. (2008). Impact of consignment inventory and vendor-managed inventory for a two-part supply chain. *International Journal of Production Economics*, *113*(2), 502-517.

48. Harris, F. W. (1913). How many parts to make at once. *Factory: The Magazine of Management*, *10(2)*, February, 135-136,152.

49. He, Q., and Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20(1),* 89-99.

50. Hill, R. M. (1997). The single-vendor single-buyer integrated production-inventory model with a generalized policy,". *European Journal of Operational Research*, *97*(3), 493-499.

51. Hill, R. M. (1999). The optimal production and shipment policy for the single-vendor single-buyer integrated production inventory problem. *International Journal of Production Research*, *37*(12), 2463-2475.

52. Hill, R. M., and Omar, M. (2006). Another look at the single-vendor single-buyer integrated production-inventory problem. *International Journal of Production Research*, *44*(4), 791-800.

53. Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* MIT Press.

54. Hooke, R., and Jeeves, T. A. (1961). Direct Search Solution of Numerical and Statistical Problems. *Journal of the ACM*, *8*(2), 212-229.

55. Huang, C. (2002). An integrated vendor-buyer cooperative inventory model for items with imperfect quality. *Production Planning and control*, *13*(4), 355-361.

56. Huang, C., Cheng, T. L., Kao, T. C., and Goyal, S. K. (2011). An integrated inventory model involving manufacturing setup cost reduction in compound Poisson process. *International Journal of Production Research*, *49*(4), 1219-1228.

57. Huang, F.-Z., Wang, L., and He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, *186(1)*, 340-356.

58. Huang, Q. Y., and Chen, J. F. (2009). A note on: Modeling an industrial strategy for inventory management in supply chains: the "consignment Stock" case. *International Journal of Production Research*, *47*(22), 6469-6475.

59. Iwasaki, N., Yasuda, K., and Ueno, G. (2006). Dynamic Parameter Tuning of Particle Swarm Optimization. *IEEJ Transactions*, *1*(4), 353-363.

60. Kao, Y.-T., and Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, *8(2)*, 849-857.

61. Karaboga, D., and Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8(1)*, 687-697.

62. Kennedy, J., and Eberhart, R. C. (1995). Particle Swarm optimization. *Proceedings of IEEE International Conference of Neural Networks*, (pp. 1942-1948). Perth, Australia.

63. Kim, D. H., and Park, K. S. (1985). (Q, r) inventory model with a mixture of lost sales and time-weighted backorders. *Journal of the Operational Research Society*, *36*(3), 231-238.

64. Kitayama, S., and Yasuda, K. (2006). A method for mixed integer programming problems by particle swarm optimization. *Electrical Engineering in Japan*, *157*(2), 40–49.

65. LaTorre, A., Muelas, S., and Pena, J. (2011). A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test. *Soft Computing*, *15(11)*, 2187-2199.

66. Lee, T. S., Ting, T. O., Lin, Y. J., and Htay, T. (2007). A particle swarm approach for grinding process optimization analysi. *International Journal of Advanced Manufacturing Technology*, *33(11-12)*, 1128–1135.

67. Lee, W., and Wang, S. P. (2008). Managing level of consigned inventory with buyer's warehouse capacity constraint. *Production Planning and control*, *19*(7), 677-685.

68. Li, H., and Lu, H., (2009). Global optimization for generalized geometric programs with mixed free-sign variables, *Operations Research*, 57(3), 701-713.

69. Li, S., and Hua, Z. (2008). A note on channel performance under consignment contract with revenue sharing. *European Journal of Operational Research*, *184(2)*, 793-796.

70. Li, S., Zhu, Z., and Huang, L. (2009). Supply chain coordination and decision making under consignment contract with revenue sharing. *International Journal of Production Economics*, *120(1)*, 88-99.

71. Liao, T. (2010). Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, *10*(4), 1188-1199.

72. Lin, Y.-C., Hwang, K. -S., and Wang, F. -S. (2001). Co-evolutionary hybrid differential evolution for mixed-integer optimization problems,. *Engineering Optimization*, *33(6)*, 663-682.

73. Liu, D., Tan, K. C., Goh, C. K., and Ho, W. K. (2007). A multiobjective memetic algorithm based on particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37*(1), 42-50.

74. Lu, L. (1995). A one-vendor multi-buyer integrated inventory model. *European Journal of Operational Research, 81*(2), 312-323.

75. Lung, R. I., and Dimitrescu, D. (2010). Evolutionary swarm cooperative optimization in dynamic environments. *Nature Computing, 9(1)*, 83-94.

76. Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*(2), 1567-1579.

77. Mathur, M., Karale, S. B., Priye, S., Jayaraman, V. K., and Kulkarni, B. D. (2000). Ant colony approach to continuous function optimization,. *Industrial and Engineering Chemistry Research, 39(10)*, 3814–3822.

78. Micheau, V. A. (2005). How Boeing and Alcoa implemented a successful vendor managed inventory program. *The Journal of Business Forecasting, 24*(1), 17-19.

79. Ong, H. L., Ang, B. W., Goh, T. N., and Deng, C. C. (1996). A model for vending machine services in the soft drink industry. *Asia-Pacific Journal of Operational Research, 13*(2), 209-224.

80. Ouyang, L. Y., Chen, C. K., and Chang, H. C. (1999). Lead time and ordering cost reductions in continuous review inventory systems with partial backorders. *Journal of the Operational Research Society, 50*(12), 1272-1279.

81. Pan, H., Wang, L., and Liu, B. (2006). Particle swarm optimization for function optimization in noisy environment. *Applied Mathematics and Computation, 181*(2), 908-919.

82. Pan, J. C., and Yang, J. S. (2002). A study of an integrated inventory with controllable lead time. *International Journal of Production Research, 40*(5), 1263-1273.

83. Parija, G. R., and Sarker, B. R. (1999). Operations planning in a supply chain system with fixed-interval deliveries to multiple customers. *IIE Transactions Special Issue on Manufacturing Logistics, 31*(11), 1075-1082.

84. Pelta, D., Cruz, C., Sancho-Royo, A., and Verdegay, J. L. (2006). Using memory and fuzzy rules in a cooperative multi-thread strategy for optimization. *Information Sciences, 176* (13), 1849–1868.

85. Persona, A., Grassi, A., and Catena, M. (2005). Consignment stock of inventories in the presence of obsolescence. *International Journal of Production Research*, *43*(23), 4969-4988.

86. Piplani, R., and Viswanathan, S. (2003). A model for evaluating supplier-owned inventory strategy. *International Journal of Production Economics*, *81-82*(SI), 565-571.

87. Qin, A. K., Huang, V. L., and Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, *13*(2), 398-417.

88. Raidl, G. R. (2006). A unified view on hybrid metaheuristics, F. Almeida et al. (Eds.): HM 2006, LNCS 4030 (2006) 1-12.

89. Ravichandran, N. (1995). Stochastic analysis of a continuous review perishable inventory system with positive lead time and Poisson demand. *European Journal of Operational Research*, *84*(2), 444-457.

90. Sarker, B. R. (2013). Consignment stock policy models for supply chain systems: a critical review and comparative perspectives. *International Journal of Production Economics* (Invited Paper, accepted: *IJPE-D-13-00642-R1*).

91. Sarker, B. R., and Coates, E. (1997). Manufacturing setup cost reduction under variable lead times and finite opportunities for investment. *International Journal of Production Economics*, *49*(3), 237–247.

92. Sarker, B. R., and Diponegoro, A. (2009). Optimal production plans and shipment schedules for a supply-chain system with multiple suppliers and multiple buyers. *European Journal of Operational Research*, *194*(3), 753-773.

93. Sarker, B. R., and Parija, G. R. (1994). An optimal batch size for a production system operating under a fixed quantity, periodic delivery policy. *Journal of the Operational Research Society*, *45*(8), 891-900.

94. Sarker, B. R., and Parija, G. R. (1996). Optimal batch size and raw material ordering policy for a production system with a fixed interval, lumpy-demand delivery system. *European Journal of Operational Research*, *89*(3), 593-608.

95. Sarker, B. R., Chawhan, A. D., and Biswas, P. (2009). An optimal policy for recovery and procurement under multiple setups. *OPSEARCH*, *46*(4), 390-417.

96. Sarker, B. R., Duan, Q., Wu, B., and Yu, J. (2011). Consignment Policy for Retailers . In B. A. Metri, J. N. Gupta, and E. B. S. Sahay, *Managing Global Services Emerging Trends (ISBN-978-81-7446-922-9), First edition*, (pp. 324-344). Naraina, Phase I, New Delhi 110 028, India.: EXCEL Books.

97. Sharma, S. (2008). On the flexibility of demand and production rate. *European Journal of Operational Research*, *190*(2), 557-561.

98. Sharma, S. (2009). A composite model in the context of a production-inventory system. *Optimization Letters*, *3*(2), 239-251.

99. Srinivas, C., and Rao, C. (2010). Optimization of supply chains for single-vendor-multibuyer consignment stock policy with genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *48*(1-4), 407-420.

100. Srinivas, C., and Rao, C. S. (2004). An improved consignment stock policy under stochastic lead times for effective inventory management in supply chains. *Proceedings of International Engineering management Conference* , (pp. 1222-1225).

101. Storn, R., and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341-359.

102. Sun, J., Zhang, Q., and Tsang, E. P. (2005). DE/EDA: a new evolutionary algorithm for global optimization. *Information Sciences, 169(3-4)*, 249-262.

103. Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, *8(5)*, 541-564.

104. Tseng, L., and Chen, C. (2008). Multiple trajectory search for large scale global optimization. *the 10th IEEE congress on evolutionary computation, CEC 2008 (IEEE World Congress on Computational Intelligence).* (pp. 3052–3059). IEEE Press.

105. Turcsik, R. (2002). Efficiency on consignment. *Progressive Grocer*, *81*(2), 29-31.

106. Valentini, G., and Zavanella, L. (2003). The consignment stock of inventories: industrial case study and performance analysis. *International Journal of Production Economics*, *81-82*(1), 215-224.

107. Wang, X., Gao, X. Z., and Ovaska, S. J. (2007). A hybrid optimization algorithm based on ant colony and immune principles. *International Journal of Computer Science and Applications*, *4*(3), 30-44.

108. Wang, Y. (2006). Joint pricing-production decisions in supply chains of complementary products with uncertain demand. *Operations Research*, *54*(6), 1110-1127.

109. Wang, Y., Li, J., and Shen, Z. J. (2004). Channel performance under consignment contract with revenue sharing. *Management Science*, *50*(1), 34-47.

110. Wang, Y., Li, T., and Liang, Z. (2009). A general algorithm for solving Generalized Geometric Programming with nonpositive degree of difficulty. *Computational Optimization and Applications*, *44*(1), 139-158.

111. Wu, K., and Ouyang, L. (2003). An integrated single-vendor single-buyer inventory system with shortage derived algebraically. *Production Planning and Control*, *14*(6), 555-561.

112. Yap, R. (1999, October 8-9). *Holding the aces in manufacturing logistics*. Retrieved October 8-9, 1999, from Manufacturing Logistics for the 21st Century: http://www.ych.com/e-conference.htm.

113. Yi, H.Z., and Sarker, B. R. (2013a). An Optimal consignment stock production and replenishment policy with controllable lead time. *International Journal of Production Research*, *51*(21), 6316-6335.

114. Yi, H.Z., and Sarker, B. R. (2013b). An operational policy for an integrated inventory system under consignment stock policy with controllable lead time and buyers' space limitation. *Computers and Operations Research*, *40*(11), 2632-2645.

115. Yi, H.Z., and Sarker, B. R. (2013c). The impacts of normally distributed demand, controllable lead time and buyer's space limitation to an integrated system under consignment stocking policy. *International Journal of Production Research* (Accepted August 21, 2013: *TPRS-2013-IJPR-0597.R2*).

116. Yi, H.Z, Duan, Q., and Liao, T. W. (2013). Three improved hybrid metaheuristic algorithms for engineering design optimization. *Applied Soft Computing*, *13*(5), 2433-2444.

117. Yin, P.-Y., Glover, F., Laguna, M., and Zhu, J.-X. (2010). Cyber swarm algorithms-improving particle swarm optimization using adaptive memory strategies. *European Journal of Operational Research*, *201(2)*, 377-389.

118. Yokota, T., Gen, M., and Li, Y.-X. (1996). Genetic algorithm for non-linear mixed integer programming problems and its application. *Computers and Industrial Engineering, 30*(4), 905–917.

119. Yu, J., Sarker, B. R., Duan, Q., and Wu, B. (2012). Single-manufacturer, multi-retailer consignment policy for retailers' generalized demand distributions. *Journal of the Operational Research Society, 63*(12), 1708-1719.

120. Zanoni, S., and Grubbstrom, R. W. (2004). A note on an industrial strategy for stock management in supply chains: modelling and performance evaluation. *International Journal of Production Research*, *42*(20), 4421-4426.

121. Zanoni, S., Ferretti, I., and Zavanella, L. (2005). Multi echelon spare parts inventory optimization: a simulative study. *Proceedings 19th European Conference on Modelling and Simulation*, *ECMS 2005*, (pp. 393-398). Riga, Latvia.

122. Zavanella, L., and Zanoni, S. (2009). A one-vendor multi-buyer integrated production-inventory model: The"'Consignment Stock" case. *International Journal of Production Economics*, *118*(1), 225-232.

123. Zhang, J., and Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optimal external archive. *IEEE Transactions on Evolutionary Computation*. *13*(5), 945-958.

124. Zhao, X., and Wu, F. (2011). Coordination of agri-food chain with revenue-sharing contract under stochastic output and stochastic demand. *Asia-Pacific Journal of Operational Research*, *28*(4), 487-510.

APPENDIX A: EQUATION DEDUCTIONS AND PROOF OF PROPERTIES

APPENDIX A-1: CALCULATION OF AREA *S* ON FIGURE 4.1

The shaded area S in Figure 4.1(b) is composed of 4 parts, namely S_1, S_2, S_3, and S_4. Note that, for convenience of computation, the leftmost triangle is moved to the right side acting as part of S_1. Applying simple geometry, $S_1 = \frac{D}{2}\left[\frac{nq}{D} - (n-k-1)\frac{q}{P} - \frac{kq}{D}\right]^2$,

$S_2 = kq^2/2D$, $S_3 = (n-k-1)(1-D/P)kq^2/D$, and

$S_4 = qTD/2P + q(1-D/P)T \ + qTD/2P \ + 2q(1-D/P)T \ + ... + qTD/2P$

$+ q(n-k-1)(1-D/P)T \ = (n\text{-}k\text{-}1)qTD/2P \ + q(n-k-1)(n-k)(1-D/P)T/2$.

Hence, the total area S can be given as

$$S = S_1 + S_2 + S_3 + S_4 = \frac{D}{2}\left[\frac{nq}{D} - (n-k-1)\frac{q}{P} - \frac{kq}{D}\right]^2 + (n-k-1)(1-D/P)\frac{kq^2}{D}$$

$$+\frac{kq^2}{2D} + (n\text{-}k\text{-}1)\frac{qDT}{2P} + \frac{(n-k-1)(n-k)}{2}(1-\frac{D}{P})qT .$$

This, through rearrangement and simplification, can be rewritten as

$$S = q^2\left(\frac{n^2-k^2-k}{2D} + \frac{n-n^2+k^2+k}{2P}\right) \qquad \text{(A-1.1)}$$

APPENDIX A-2: CONVEXITY OF THE OBJECTIVE FUNCTION IN EQUATIONS (4.7) AND (4.8)

Here, we study the convexity of the objective functions in Equations (4.7) and (4.8) for variables *n* and *q* under Case 1 and Case 2.

A-2.1 ***The objective function*** $JTEC(q,n,0,L)$ ***in Equation (4.7) is strictly convex at point (n*, q*) under Case 1 (*** $h_v^s > h_b^s$ ***):***

Taking the first partial derivative of Equation (4.7) with respect to *n* and *q*, respectively, we have:

$$\frac{\partial JTEC}{\partial q} = \frac{-(A_v + nA_b)D}{nq^2} + h_v \frac{D}{2P} + h_b \frac{D + n(P-D)}{2P} - \frac{DR(L)}{q^2}, \tag{A-2.1}$$

and

$$\frac{\partial JTEC}{\partial n} = \frac{-A_v D}{n^2 q} + h_b \frac{q(P-D)}{2P}. \tag{A-2.2}$$

The second partial derivatives with respect to *n* and *q* are

$$\frac{\partial^2 JTEC}{\partial q^2} = \frac{2(A_v + nA_b)D}{nq^3} + \frac{2DR(L)}{q^3} = \frac{2D[A_v + nA_b + nR(L)]}{nq^3}, \tag{A-2.3}$$

and

$$\frac{\partial^2 JTEC}{\partial n^2} = \frac{2A_v D}{n^3 q}. \tag{A-2.4}$$

Meanwhile, the second partial derivative of Equation (A-2.2) with respect to *q* is given as

$$\frac{\partial^2 JTEC}{\partial q \partial n} = \frac{A_v D}{n^2 q^2} + h_b \frac{(P-D)}{2P}. \tag{A-2.5}$$

It can be easily verified that taking partial derivative of Equation (A-2.1) with respect to *n* yields the same result as Equation (A-2.5).Therefore, the Hessian matrix can be written as

$$H(JTEC)=\begin{bmatrix}\dfrac{\partial^2 JTEC}{\partial q^2} & \dfrac{\partial^2 JTEC}{\partial q\partial n}\\ \dfrac{\partial^2 JTEC}{\partial n\partial q} & \dfrac{\partial^2 JTEC}{\partial n^2}\end{bmatrix}=\begin{bmatrix}\dfrac{2D[A_v+nA_b+nR(L)]}{nq^3} & \dfrac{A_vD}{n^2q^2}+h_b\dfrac{(P-D)}{2P}\\ \dfrac{A_vD}{n^2q^2}+h_b\dfrac{(P-D)}{2P} & \dfrac{2A_vD}{n^3q}\end{bmatrix}.$$

Note that the main diagonal elements are positive. It follows that the necessity conditions for *JTEC(n, q)* to be convex are:

$$\begin{cases}\dfrac{\partial JTEC}{\partial q}=\dfrac{-(A_v+nA_b)D}{nq^2}+h_v\dfrac{D}{2P}+h_b\dfrac{D+n(P-D)}{2P}-\dfrac{DR(L)}{q^2}=0\\ \dfrac{\partial JTEC}{\partial n}=\dfrac{-A_vD}{n^2q}+h_b\dfrac{q(P-D)}{2P}=0\end{cases}. \quad \text{(A-2.6)}$$

Solving Equation (A-2.6) we have

$$n^*=\sqrt{\frac{A_v(h_v+h_b)D}{h_b(P-D)[R(L)+A_b]}}, \quad \text{(A-2.7)}$$

$$q^*=\sqrt{\frac{2P[R(L)+A_b]}{(h_v+h_b)}}. \quad \text{(A-2.8)}$$

The sufficient condition for the function $JTEC(q,n,0,L)$ to be convex at point (n*, q*) is to test whether the following inequality holds or not:

$$H(JTEC)\big|_{(n^*,q^*)}=\left\{\frac{2D[A_v+nA_b+nR(L)]}{nq^3}\frac{2A_vD}{n^3q}-\left[\frac{A_vD}{n^2q^2}+h_b\frac{(P-D)}{2P}\right]^2\right\}\Bigg|_{(n^*,q^*)}>0. \quad \text{(A-2.9)}$$

After substituting Equations (A-2.7) and (A-2.8) into Equation (A-2.9) and simplifying, we have

$$\sqrt{\frac{(h_v+h_b)[R(L)+A_b]h_b^{\,3}(P-D)^3}{A_v^{\,3}P^4D^3}}>0. \quad \text{(A-2.10)}$$

Since $P > D$, the inequality always holds; that is, the objective function *JTEC(n, q)* is strictly convex at point *(n*, q*)* under Case 1 ($h_v^s > h_b^s$).

A-2.2 ***The objective function is strictly convex at point (n*, q*) under Case 2(*** $h_v^s < h_b^s$ ***):***

Similarly, the first partial derivative of Equation (4.8) with respect to q and n, leads respectively, to

$$\frac{\partial JTEC}{\partial q} = \frac{-(A_v + nA_b)D}{nq^2} + h_v \frac{(n-1)P-(n-2)D}{2P} + \frac{h_b}{2} - \frac{DR(L)}{q^2}, \tag{A-2.11}$$

and

$$\frac{\partial JTEC}{\partial n} = \frac{-A_v D}{n^2 q} + h_v \frac{q(P-D)}{2P}. \tag{A-2.12}$$

The second partial derivative of Equation (4.8) with respect to n and q can be given as

$$\frac{\partial^2 JTEC}{\partial q^2} = \frac{2(A_v + nA_b)D}{nq^3} + \frac{2DR(L)}{q^3} = \frac{2D[A_v + nA_b + nR(L)]}{nq^3}, \tag{A-2.13}$$

and

$$\frac{\partial^2 JTEC}{\partial n^2} = \frac{2A_v D}{n^3 q}. \tag{A-2.14}$$

Similarly, from Equation (A-2.11), we have:

$$\frac{\partial^2 JTEC}{\partial q \partial n} = \frac{A_v D}{n^2 q^2} + h_v \frac{(P-D)}{2P}. \tag{A-2.15}$$

The Hessian matrix for this case stands as

$$H(JTEC) = \begin{bmatrix} \frac{\partial^2 JTEC}{\partial q^2} & \frac{\partial^2 JTEC}{\partial q \partial n} \\ \frac{\partial^2 JTEC}{\partial n \partial q} & \frac{\partial^2 JTEC}{\partial n^2} \end{bmatrix} = \begin{bmatrix} \frac{2D[A_v + nA_b + nR(L)]}{nq^3} & \frac{A_v D}{n^2 q^2} + h_v \frac{(P-D)}{2P} \\ \frac{A_v D}{n^2 q^2} + h_v \frac{(P-D)}{2P} & \frac{2A_v D}{n^3 q} \end{bmatrix}.$$

where the necessary condition for convexity is

$$\begin{cases} \frac{\partial JTEC}{\partial q} = \frac{-(A_v + nA_b)D}{nq^2} + h_v \frac{(n-1)P-(n-2)D}{2P} + \frac{h_b}{2} - \frac{DR(L)}{q^2} = 0 \\ \frac{\partial JTEC}{\partial n} = \frac{-A_v D}{n^2 q} + h_v \frac{q(P-D)}{2P} = 0 \end{cases}. \tag{A-2.16}$$

Solving it, we have

$$n^* = \sqrt{\frac{A_v(Ph_b - Ph_v + 2Dh_v)}{h_v(P-D)[R(L)+A_b]}}\,, \tag{A-2.17}$$

and

$$q^* = \sqrt{\frac{2PD[R(L)+A_b]}{(Ph_b - Ph_v + 2Dh_v)}}\ . \tag{A-2.18}$$

Now, we test the sufficient condition for our objective function to be convex at point (*n**, *q**)

$$H(JTEC)\Big|_{(n^*,q^*)} = \left\{\frac{2D[A_v + nA_b + nR(L)]}{nq^3}\frac{2A_vD}{n^3q} - \left[\frac{A_vD}{n^2q^2} + h_v\frac{(P-D)}{2P}\right]^2\right\}\Bigg|_{(n^*,q^*)} > 0\,. \tag{A-2.19}$$

Note that the only difference between Equations (A-2.19) and (A-2.9) is that we have h_b in Equation (A-2.9) but instead have h_v in Equation (A-2.19). Again, through substituting Equations (A-2.17) and (A-2.18) into Equation (A-2.19) and simplifying, we have

$$\sqrt{\frac{(Ph_b - Ph_v + 2Dh_v)[R(L)+A_b]{h_v}^3(P-D)^3}{{A_v}^3P^4D^3}} > 0\,. \tag{A-2.20}$$

All the terms in (A-2.20) are positive, the inequality always holds. In other words, the objective function *JTEC(n, q)* is strictly convex at point *(n*, q*)* under Case 2 ($h_v^s < h_b^s$).

APPENDIX A-3: CONVEXITY OF THE OBJECTIVE FUNCTION IN EQUATIONS (4.7) AND (4.8)

It can be easily seen from Equations (A-2.3) and (A-2.13) that the second partial derivatives of $JTEC(q,n,0,L)$ and $JTEC(q,n,n-1,L)$ with respect to q are greater than zero. Therefore, the objective function is convex on q.

Letting Equations (A-2.1) and (A-2.11) equal to zero and solve. We get

$$q^* = \sqrt{\frac{2PD[A_v + nA_b + nR(L)]}{n\{(h_v^s + h_v^f)D + (h_b^s + h_v^f)[np - (n-1)D]\}}},$$

for case 1, and

$$q^* = \sqrt{\frac{2PD[A_v + nA_b + nR(L)]}{n\{(h_v^s[(n-1)P - (n-2)D] + h_v^f[np - (n-2)D] + h_b^s P\}}},$$

for case 2. This completes the proof of Property 4.3.

APPENDIX A-4: CONVEXITY OF THE OBJECTIVE FUNCTION IN EQUATIONS (4.7) AND (4.8)

It can be easily seen from Equations (A-2.4) and (A-2.14) that the second partial derivatives of $JTEC(q,n,0,L)$ and $JTEC(q,n,n-1,L)$ with respect to n are greater than zero. Therefore, the objective function is convex on n.

Letting Equations (A-2.2) and (A-2.12) equal to zero and solve. We get

$$n^* = \frac{1}{q}\sqrt{\frac{2A_v DP}{(h_b^s + h_v^f)(P-D)}},$$

for case 1, and

$$n^* = \frac{1}{q}\sqrt{\frac{2A_v DP}{(h_v^s + h_v^f)(P-D)}},$$

for case 2. This completes the proof of Property 4.4.

APPENDIX A-5: CALCULATION OF AREA *S* OF FIGURE 5.4

The area S in Figure 5.4b is composed of 5 parts, namely S_1, S_2, S_3, S_4, and S_5. For convenience of computation, the leftmost triangle is moved to the right side acting as part of S_1. Applying simple geometry, we have:

$$S_1 = u^2 / 2D,$$

$$S_2 = (k-1)q^2 / 2D,$$

$$S_3 = (u-q)[(nq-u)/D - (n-k-1)q/P],$$

$$S_4 = \frac{q^2 D}{2P^2} + \frac{q^2(P-D)}{P^2} + \frac{q^2 D}{2P^2} \quad + 2\frac{q^2(P-D)}{P^2} + \ldots + \frac{q^2 D}{2P^2} \; + (n-k-1)\frac{q^2(P-D)}{P^2}$$

$$= (n\text{-}k\text{-}1)\frac{q^2 D}{2P^2} + \frac{(n-k-1)(n-k)}{2}\frac{q^2(P-D)}{P^2},$$

and

$$S_5 = \frac{[(n-k+1)q - (n-k-1)\frac{D}{P}q - u]^2}{2D}.$$

Hence, the total area S can be given as

$$S = S_1 + S_2 + S_3 + S_4 + S_5 = \frac{u^2}{2D} + \frac{(k-1)q^2}{2D} + \frac{(u-q)(nq-u)}{D} - \frac{(u-q)(n-k-1)q}{P}$$

$$+ \frac{(n-k-1)q^2 D + (n-k-1)(n-k)q^2(P-D)}{2P^2} + \frac{[(n-k+1)q - (n-k-1)\frac{D}{P}q - u]^2}{2D}.$$

This, through rearrangement and simplification, can be rewritten as

$$S = \frac{n^2q^2 + k^2q^2 - 2nkq^2 - kq^2 + 2ukq}{2D} - \frac{(n-k)(n-k-1)q^2}{2P} \quad . \qquad \text{(A-5.1)}$$

APPENDIX A-6: CALCULATION OF AREA *S* OF FIGURE 6.2

The area S in Figure 6.2(b) is composed of 5 parts, namely S_1, S_2, S_3, S_4, and S_5. For convenience of computation, the leftmost triangle is moved to the right side acting as part of S_1. Applying simple geometry, we have:

$$S_1 = U^2 / 2\mu\,,$$

$$S_2 = (k-1)q^2 / 2\mu\,,$$

$$S_3 = (U-q)[(nq-U)/\mu - (n-k-1)q/P]\,,$$

$$S_4 = \frac{q^2\mu}{2P^2} + \frac{q^2(P-\mu)}{P^2} + \frac{q^2\mu}{2P^2} + 2\frac{q^2(P-\mu)}{P^2} + \ldots + \frac{q^2\mu}{2P^2}$$

$$+ (n-k-1)\frac{q^2(P-\mu)}{P^2} + (n-k-1)\frac{q}{P}s\sigma\sqrt{L+\frac{q}{P}}$$

$$= (n-k-1)\left[\frac{q^2\mu}{2P^2} + \frac{q}{P}s\sigma\sqrt{L+\frac{q}{P}}\right] + \frac{(n-k-1)(n-k)}{2}\frac{q^2(P-\mu)}{P^2}\,,$$

And

$$S_5 = \frac{[(n-k+1)q-(n-k-1)q\mu/P-U]^2 + [(n-k+1)q-(n-k-1)q\mu/P-U]s\sigma\sqrt{L+q/P}}{2\mu}$$

Hence, the total area S can be given as

$$S = S_1 + S_2 + S_3 + S_4 + S_5 = \frac{U^2}{2\mu} + \frac{(k-1)q^2}{2\mu} + \frac{(U-q)(nq-U)}{\mu} - \frac{(U-q)(n-k-1)q}{P}$$

$$+ \frac{(n-k-1)[q^2\mu + 2Pqs\sigma\sqrt{L+q/P} + q^2(n-k)(P-\mu)]}{2P^2}$$

$$+ \frac{[(n-k+1)q-(n-k-1)q\mu/P-U]^2 + s\sigma\sqrt{L+q/P}[(n-k+1)q-(n-k-1)q\mu/P-U]}{2\mu}$$

This, through rearrangement and simplification, can be rewritten as

$$S = \frac{n^2q^2 + k^2q^2 - 2nkq^2 - kq^2 + 2kqU + s\sigma\sqrt{L + q/P}(nq - kq + q - U)}{2\mu}$$

$$- \frac{(n-k)(n-k-1)q^2 - qs\sigma\sqrt{L + q/P}(n-k-1)}{2P}. \tag{A-6.1}$$

APPENDIX A-7: EVALUATION OF THE EXPECTATION $E(X-r)^+$

Suppose X is the random variable, the demand rate during lead time that follows a normal distribution which is defined as

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty \le x \le \infty. \tag{A-7.1}$$

Let $z = \dfrac{x-\mu}{\sigma}$ from which $x = \mu + z\sigma$, $x - r = z\sigma + \mu - r$, and also $dx = \sigma dz$. thus, using expression (A-7.1),

$$f(x)dx = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = \phi(z)dz, \qquad -\infty \le z \le \infty \tag{A-7.2}$$

where $\phi(z) = \dfrac{1}{\sqrt{2\pi}} e^{-z^2/2}$ is the standard normal distribution, $N(0,1)$. If $x = r$ in $z = (x-\mu)/\sigma$, then $z = (r-\mu)/\sigma$, and if $x = \infty$, then $z = \infty$ also. Therefore, the expected amount of shortage per cycle is

$$E(X-r)^+ = \bar{b}(r) = \int_r^\infty (x-r) f(x)dx = \int_{\frac{r-\mu}{\sigma}}^\infty (z\sigma + \mu - r)\phi(z)dz$$

$$= \sigma \int_{\frac{r-\mu}{\sigma}}^\infty \left(z - \frac{r-\mu}{\sigma}\right)\phi(z)dz. \tag{A-7.3}$$

Let $v = \dfrac{r-\mu}{\sigma}$; then Equation (A-7.3) transforms to

$$E(X-r)^+ = \bar{b}(r) = \sigma \int_v^\infty (z-v)\phi(z)dz = \sigma L(v), \tag{A-7.4}$$

where $L(v) = \int_{v}^{\infty}(z-v)\phi(z)dz$ is called the *unit normal loss function*, it can also be evaluated in a closed form by transforming *L(v)* in the way the standard normal table can be used:

$$L(v) = \int_{v}^{\infty}(z-v)\phi(z)dz = \int_{v}^{\infty} z\phi(z)dz - v\int_{v}^{\infty}\phi(z)dz = \int_{v}^{\infty} z\phi(z)dz - v\left[1-\int_{-\infty}^{v}\phi(z)dz\right]$$

$$= \frac{1}{\sqrt{2\pi}}\int_{v}^{\infty} ze^{-z^2/2}dz - v[1-\Phi(v)], \qquad \text{(A-7.5)}$$

where $\Phi(v) = \int_{-\infty}^{v}\phi(z)dz$, the cumulative probability distribution of $N(0,1)$ with $v = (r-\mu)/\sigma$.

Let $y = e^{-z^2/2}$. Then $dy = -ze^{-z^2/2}dz$ from which $y = -\int ze^{-z^2/2}dz$. Therefore, by reverse process, $\int_{v}^{\infty} ze^{-z^2/2}dz = \left[-e^{-z^2/2}\right]_{v}^{\infty} = e^{-v^2/2}$. Thus, Equation (A-7.5) can be written as

$$L(v) = \frac{1}{\sqrt{2\pi}}\int_{v}^{\infty} ze^{-z^2/2}dz - v[1-\Phi(v)] = \frac{1}{\sqrt{2\pi}}e^{-v^2/2} - v[1-\Phi(v)] = \phi(v) - v[1-\Phi(v)]. \text{(A-7.6)}$$

From Equations (A-7.4) and (A-7.6), the expected number of units short at the end of a cycle can be calculated as:

$$E(X-r)^{+} = \overline{b}(r) = \int_{r}^{\infty}(x-r)f(x)dx = \sigma L(v) = \sigma[v\Phi(v) + \phi(v) - v]. \qquad \text{(A-7.7)}$$

where, $v = (r-\mu)/\sigma$.

APPENDIX B: ALGORITHMS AND PSEUDO-CODES

APPENDIX B-1: PSEUDO-CODE OF THE MDE'-HJ HYBRID ALGORITHM

Specify MDE'-related parameters, i.e., population size *NP*, factor used to generate a mutated vector *F*, crossover threshold *CR* and maximal number of function evaluations *Maxnfe* and HJ related parameters, i.e., step size λ , Maximum number of iterations *M* and Probability to perform HJ local search *p*.
Give the upper/lower bound of each variable (Replace each discrete variable by a continuous position variable using the generalized discrete variable handling method)
Give the global optimum or the best known value, *g_opt*
Randomly generate the initial population.
Evaluate the initial population. Handle constraints by using the Deb's method. Set the number of function evaluations, *nfe*, to be *NP*.
Determine the best function value, *f_best*, and the best solution, *x_best*, of the initial population. Sort feasible and infeasible solutions according to Deb's constraint handling method.
While *nfe*<*Maxnfe* and *f_best*>*g_opt*+10^{-6}
    For each target vector (=1, 2, …, *NP*)
     Generate a trial vector according to the MDE' algorithm
     Handle bound violations
     If randomly selected (with a specified percentage, *p*), apply HJ (presented below) to the trial vector and increment *nfe* accordingly;
         Else
       Evaluate the trial vector and increment *nfe* by one.
     End if
  End for
  Update the current generation of population by replacing the target vector with the trial vector if the trial vector is better according to Deb's selection criteria
End while

APPENDIX B-2: PSEUDO-CODE OF THE HJ LOCAL SEARCH ALGORITHM

```
%       beat_patn is the pattern search indicator to show a pattern search succeeds (1) or not (0)
%       beat_explr is the exploration search indicator to show an exploration search succeeds
        (1) or not (0)
%       λ is the adaptive step size parameter, initially set as 10% of the domain range
%       g_obj is the global optimum
%       ε is the acceptable error
%       Main_Function
(1)     Let the candidate solution obtained from MDE' be x_base;
(2)     Evaluate x_base and store its objective function value in f_base, its constraint
        violation in g_base
(3)     While current number of cycles < maximum number of cycles or f_base>g_obj+ε
(4)         [beat_explr, x_base, x_explr]=Subfunction_ExplSrh(x_base, λ)
(5)         If beat_explr=1,
(6)             [x_base]=Subfunction_PatnSrh(x_base, x_explr, λ)
(7)         Otherwise
(8)             Reduce λ;
(9)         end
(10)        handle discrete numbers
(11)        increment cycle number by one;
(12)    end while
(13)    Output final x_base
%       Subfunction_ExplSrh: Exploration search
(1)     set beat_explr=0;
(2)     set x_explr and x_temp as x_base;
(3)     for  i=1: dimensions of x_explr
(4)        x_explr(i)=x_base(i)+λ(i);
(5)        repair boundary of x_explr if needed;
(6)        evaluate new x_explr
(7)        if  x_explr is superior to x_temp
(8)          set beat_explr=1;
(9)          x_temp(i)=x_explr(i); update f_temp and g_temp accordingly
(10)       else
(11)         x_explr(i)=x_base(i)-λ(i);
(12)         repair boundary of x_explr if needed;
(13)         evaluate new x_explr
(14)         if x_explr is superior to x_temp
(15)           set beat_explr=1;
(16)           x_temp(i)=x_explr(i); update f_temp and g_temp
(17)         end
(18)       end
(19)    end
```

```
%   Subfunction_PatnSrh: Pattern search.
initialize beat_patn=0;
x_patn_start=2*x_explr-x_base;
handle discrete numbers and boundary repair
evaluate x_patn_start and store its f_patn_start and g_patn_start
set x_patn and x_temp as x_patn_start;
for i=1:dimensions of x_patn
    x_patn(i)=x_patn_start(i)+λ(i);
    repair boundary of x_patn
    evalute x_patn
    if x_patn is superior to x_temp
        beat_patn=1;
        replace x_temp(i) as x_patn(i);
    else
        x_patn(i)=x_patn_start(i)-λ(i);
        repair boundary of x_patn
        evaluate x_patn
        if x_patn is superior to x_temp
            beat_patn=1;
            replace x_temp(i) as x_patn(i);
        end
    end
end
if beat_patn==1
    if x_patn is superior to x_explr
        replace x_base as x_explr; replace x_explr as x_patn;
        [x_base]=Subfunction_PatnSrh (x_base, x_explr, λ);
    else
        if x_patn_start is superior to x_explr
            replace x_base as x_patn_start;
            beat_patn=0;
        else
            replace x_base as x_explr;
            beat_patn=0;
        end
    end
else
    replace x_base as x_explr;
    beat_patn=0;
end
```

APPENDIX B-3: PSEUDO-CODE OF THE MDE'-IHS-HJ HYBRID ALGORITHM

Specify MDE'-related parameters, i.e., *NP*, *F*, *CR* and *Maxnfe*. Specify IHS-related parameters, i.e., *HMS*, *NI*, *HMCR*, $PAR_{\min}$, $PAR_{\max}$, $bw_{\min}$ and $bw_{\max}$. Set *HMS* and *NP* to be equal. Specify HJ related parameters, i.e., λ, *M* and *p*.

Give the upper/lower bound of each variable (Replace each discrete variable by a continuous position variable using the generalized discrete variable handling method)

Give the global optimum or the best known value, *g_opt*

Randomly generate the initial population of NP+HMS solutions

Evaluate the initial population. Handle constraints by using the Deb's method. Set the number of function evaluations, *nfe*, to be *NP+HMS*.

Determine the best function value, *f_best*, and the best solution, *x_best*, of the initial population. Sort feasible and infeasible solutions according to Deb's constraint handling method.

Retain the top 50% solutions in the HM and as the initial solutions of MDE'.

```
While nfe<Maxnfe and f_best>g_opt+10^-6
    For each target vector (=1, 2, …, NP)
        Generate a trial vector according to MDE' algorithm
        Handle bound violations
        Construct a new IHS solution according to IHS algorithm:
        Update PAR according to Eq. (3.1) and update bw according to Eq. (3.2).
        For each dimension
            If rand<HMCR
                Randomly pick a value from the HM
                If rand<PAR
                    Adjust pitch for improving the harmony x' = x' ± rand · bw
                    Check and repair violation of the bound
                End if
            Else
                Randomly generate a value within the domain range
            End if
        End for
        If the new solutions are randomly selected (with a specified percentage, p),
            apply HJ (presented above) and increment nfe accordingly;
        Else
            Evaluate the new solutions found by both algorithms and increment nfe by two.
        End if
        Replace the target vector with the new trail vector for the MDE' algorithm if there is
    an improvement, and update the HM
        Update the best solution found so far
    End for
End while
Output results
```

APPENDIX B-4: PSEUDO-CODE OF THE MDE'-PSO-HJ HYBRID ALGORITHM

Specify MDE'-related parameters, i.e., *NP*, *F*, *CR* and *Maxnfe*. Specify PSO-related parameters, i.e., *PSO*, $\omega_{\min}$, $\omega_{\max}$, $\Delta\omega$, φ_1 and φ_2. Set *NP* and *PSO* to be equal. Specify HJ related parameters, i.e., λ, *M* and *p*.

Give the upper/lower bound of each variable (Replace each discrete variable by a continuous position variable using the generalized discrete variable handling method)

Give the global optimum *g_opt*

Randomly generate the initial population of *NP+PSO* solutions

Evaluate the initial population. Handle constraints by using the Deb's method. Set the number of function evaluations, *nfe*, to be *NP+PSO*.

Determine the best function value, *f_best*, and the best solution, *x_best*, of the initial population. Sort feasible and infeasible solutions according to Deb's constraint handling method.

Retain the top 50% solutions as the initial solutions of MDE' and PSO.

While *nfe*<*Maxnfe* and *f_best*>*g_opt*+10^{-6}

    For each particle (=1, 2, …, *PSO*)

        Calculate the flying velocity of the particle according to Eq. (3.4).

        Generate new trail location for the particle according to Eq. (3.5).

        Handle bound violations.

        Construct a new trial vector according to the MDE' algorithm.

        Handle bound violations.

        If the new solutions are randomly selected (with a specified percentage, *p*),
          apply HJ (presented above) and increment *nfe* accordingly;

        Else

          Evaluate the new solutions found by both algorithms and increment *nfe* by two.

        End

        Update the position of the particle with the new position for the PSO algorithm if
          there is a improvement, and update the vector of MDE' algorithm.

        Adapt ω according to Eq. (3.6) and (3.7).

        Update the best solution found so far

    End for

End while

Output results

VITA

Hui Zhi Yi was born in Changsha, China. He received a Bachelor of Science degree in Automation Control from Central South University of Technology (CSUT, which is now known as Central South University, CSU), China, in 1994. He received his Master of Science degree in Industrial Engineering from Louisiana State University (LSU) in 2012. The degree of Doctor of Philosophy will be awarded to him by LSU in December 2013.

At LSU, Hui Zhi taught Advanced Engineering Statistics course for undergraduate and graduate students. He also worked as a Teaching Assistant and a Lab Instructor for a number of courses. As a Research Assistant, he also worked on a project funded by the U. S. Department of Homeland Security (DHS) in the area of Emergency Management.

Hui Zhi has published five papers in *Applied Soft Computing*, *Computers and Operations Research*, and *International Journal of Production Research*. Four of his recent articles are in different stages of review processes by *International Journal of Production Economics, Computers & Operations Research* and *Journal of the Operational Research Society*. He served as a reviewer for many journals including *European Journal of Operational Research*, *International Journal of Production Economics*, *International Journal of Production Research,* and *Journal of Optimization*. His current research interests are in Supply Chain Management, Manufacturing Systems, Optimization and Bio-energy Systems. He is a member of Institute For Operations Research and Management Sciences, Institute of Industrial Engineers and Decision Science Institute.