5-2011

Revenue Management for Make-to-Order and Make-to-Stock Systems

Jiao Wang
jwang37@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Jiao Wang entitled "Revenue Management for Make-to-Order and Make-to-Stock Systems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Xueping Li, Major Professor

We have read this dissertation and recommend its acceptance:

Rapinder Sawhney, Frank M. Guess, Xiaoyan Zhu

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Jiao Wang entitled "Revenue Management for Make-to-Order and Make-to-Stock Systems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Xueping Li, Major Professor

We have read this dissertation and recommend its acceptance:
Rapinder Sawhney, Frank M. Guess, Xiaoyan Zhu

Accepted for the Council:
Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

To the Graduate Council:

I am submitting herewith a dissertation written by Jiao Wang entitled "Revenue Management for Make-to-Order and Make-to-Stock Systems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Xueping Li, Co-Chair

Rapinder Sawhney, Co-Chair

We have read this dissertation
and recommend its acceptance:

Frank M. Guess

Xiaoyan Zhu

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Revenue Management for Make-to-Order and Make-to-Stock Systems

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Jiao Wang
May 2011

Dedication

This dissertation is dedicated to my parents, Shujun Zhang and Xinhai Wang, and my husband, Yuan Hou, for their love, support and encouragement.

Acknowledgments

First and foremost, I am deeply indebted to my two advisors Dr. Rapinder Sawhney and Dr. Xueping Li. Their willingness to support my work and the guidance throughout my studies has allowed me to develop my skills as a researcher within a supportive team environment. I thank them for the precious opportunity. To my thesis committee members, Dr. Frank M. Guess and Dr. Xiaoyan Zhu, who generously offered their time and provided their insights. Your inputs in the dissertation are highly appreciated.

I also would like to thank my fellow graduate students: Dengfeng Yang, Laigang Song, Yuerong Chen, Cong Guo, Zhaoxia Zhao and Yu Huang. Thank them for sharing the graduate school life together here and the supportive team environment. In addition, I would like to thank my friends at UT, for their generous help in my life and study: Jianying Chen, Yueying Wu and Shengnan Li.

Abstract

With the success of Revenue Management (RM) techniques over the past three decades in various segments of the service industry, many manufacturing firms have started exploring innovative RM technologies to improve their profits. This dissertation studies RM for make-to-order (MTO) and make-to-stock (MTS) systems.

We start with a problem faced by a MTO firm that has the ability to reject or accept the order and set prices and lead-times to influence demands. The firm is confronted with the problem to decide, which orders to accept or reject and trade-off the price, lead-time and potential for increased demand against capacity constraints, in order to maximize the total profits in a finite planning horizon with deterministic demands. We develop a mathematical model for this problem. Through numerical analysis, we present insights regarding the benefits of price customization and lead-time flexibilities in various demand scenarios.

However, the demands of MTO firms are always hard to be predicted in most situations. We further study the above problem under the stochastic demands, with the objective to maximize the long-run average profit. We model the problem as a Semi-Markov Decision Problem (SMDP) and develop a reinforcement learning (RL) algorithm-Q-learning algorithm (QLA), in which a decision agent is assigned to the machine and improves the accuracy of its

action-selection decisions via a "learning" process. Numerical experiment shows the superior performance of the QLA.

Finally, we consider a problem in a MTS production system consists of a single machine in which the demands and the processing times for N types of products are random. The problem is to decide when, what, and how much to produce so that the long-run average profit. We develop a mathematical model and propose two RL algorithms for real-time decision-making. Specifically, one is a Q-learning algorithm for Semi-Markov decision process (QLS) and another is a Q-learning algorithm with a learning-improvement heuristic (QLIH) to further improve the performance of QLS. We compare the performance of QLS and QLIH with a benchmarking Brownian policy and the first-come-first-serve policy. The numerical results show that QLIH outperforms QLS and both benchmarking policies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Motivations

In a global competitive market, companies are always striving to maximize their profitability. Revenue Management (RM) has been applied successfully to many industries in order to achieve this goal. However, RM techniques are mostly used in the service industries to allocate limited resources, such as airplane seats or hotel rooms, among a variety of customers, such as business or leisure travelers. With the tremendous success of RM techniques over the past three decades in various segments of the service industry such as airlines, hotels, car rentals, cargos etc., there is an increasing interest in applying RM in the manufacturing industries. The general applicability of RM concepts to manufacturing can be found in the studies of Harris and Pinder (1995) and Rehkopf and Spengler (2004).

An important trend in manufacturing industry is customized production, the make-to-order (MTO) environment. By producing in an MTO system, firms benefit by eliminating

finished goods inventory carrying and obsolescence costs (Nicholas, 1997). However, this benefit comes at the cost of increased response time in satisfying customer orders and/or cost of keeping higher production capacities to accommodate variations in demand (Hopp and Spearman, 2000). If it is not profitable to keep spare production capacities and due-dates of orders are inflexible, it becomes very critical for the MTO manufacturer to selectively accept and schedule customer orders. One of the common characteristics of MTO companies is the rigidity of available capacity. With high variable demand, the MTO systems imply that periods where the facility is idle and other time in which a large number of orders are waring for production. Furthermore, since accepted orders require the availability of resources to guarantee the order promising and hence potentially displace more profitable orders to come under limited production capacity, firms are confronted with the problem to decide, which orders to accept and which orders to reject, in order to maximize overall profit (Spengler et al., 2007).

To make efficient pricing, lead-time, order acceptance and scheduling decisions in face of deterministic/stochastic demands, limited production capacity and manufacturing costs, RM techniques can be adopted in MTO environment. This is precisely the focus of Chapter 2 and 3 in this dissertation.

While some customers enjoy customizing their products, other customers would accept a standard configuration in order to receive their products immediately. To satisfy these customers, manufacturing companies also produces the standard products to stock, the make-to-stock (MTS) environment. For instance, Dell also frequently provides some low-end products to attract more customers. For these low-end products, customer usually have

little flexibility in customize their products. For manufacturers, however, the problem of setups is prevalent in many industries because facilities that operate in a MTS mode typically produce standardized products that require setups. In most situations, the setup time problem is more realistic than the setup cost problem, but is also less amenable to analysis. The setup cost problem, however, may be relevant for manufacturing systems that have internalized their setup times; that is, they incur significant material, labor, and/or capital costs to greatly reduce their setup times (Markowitz et al., 1995). The presence of setup costs and times in combination with the stochastic environment adds to the difficulties of the problem. On the one hand, short cycle lengths are desirable because they provide frequent production opportunities for the various products to react to the stochastic environment. But, on the other hand, short cycle lengths increase the setup frequency, which not only reduce capacity available for production and hinders the timely fulfillment of demand, but also increases setup costs.

These developments call for RM models that integrate production sequencing and lot size decisions to dynamically respond to the stochastic demands for a MTS production system. This is exactly the focus of Chapter 4 in this dissertation.

In Section 1.2, we present a brief overview of RM. Section 1.3 states the special features of manufacturing RM models by comparing with the airline RM models, which highlights that solutions developed for airline RM cannot be directly applied to MTO manufacturing RM problem. This motivates us to explore efficient solutions to a version of the manufacturing RM model. Section 1.4 outlines the contributions and document organization of this dissertation.

1.2 Overview of Revenue Management (RM)

RM is an area of applying operations research techniques to deal with the demand management by finding at what price, how much of a limited resource should be made available to the customers (Talluri and van Ryzin, 2004). RM was the term originally used to describe the process of achieving maximum revenue from a fixed or "perishable" asset, such as a seat on a plane or hotel accommodation. It originates from the airline industry's "Yield Management", a set of sophisticated techniques that were developed initially in the 80s to maximize "bums in seats", then adopted in the accommodation, car rental, and other service segments. From the perspective of demand management decisions (Talluri and van Ryzin, 2004), the RM decisions can be divided into three categories: structural decisions (e.g., which selling format should be used?), price decisions (e.g., how to set the posted price, individual-offered price and reserve price?) and quantity decisions (e.g., whether reject or accept an order?).

After the deregulation of the airlines industry in the 1970s, the established airline faced with the difficult situation of competing with the newer low-priced airline. During this time, the adoption of RM helped that airlines to stay competitive (Cross, 1997). According to a report from SIAM news (Davis, 1994), yield management saved American Airlines $ 1.4 billion dollars in the period from 1989 to 1992, about 50% more than its net profit of $892 million for the same period. The successful application of RM in airline industry led to interest in the study, research, and application of RM.

RM applications can be classified into two categories: traditional and nontraditional categories (Boyd and Bilegan, 2003). Traditional applications are similar to the airline RM

models. Examples of traditional applications would include hotel and car rental industries. The nontraditional applications use models that are sufficiently different from the airline model (Boyd and Bilegan, 2003). Examples of nontraditional applications include retail, media, theaters, manufacturing, cruise ships, passenger railways, electricity generation and transmission, air cargo, freight and so on. The various nontraditional applications of RM are described in the study of Talluri and van Ryzin (2004).

1.3 RM in Manufacturing Industry

Compared with the RM in service industries, the research and applications of RM in manufacturing industry are relatively new. Although RM has been applied extensively and made significant contributions to the profitability of service enterprise, such as airline (Talluri and van Ryzin, 2004). However, certain features of RM problem in manufacturing make it difficult to adapt techniques developed for service RM to this problem.

First, the goal of service industry applications, i.e. the maximization of revenue, has to be replaced by the maximization of contribution margin due to significant variable costs encountered in manufacturing (Spengler et al., 2007). The term contribution margin thereby refers to the difference between revenue and variable costs for a specific order. Consequently, the goal of manufacturing RM is to maximize the profit.

Second, in the case of manufacturing RM, the manufacturer can sell future capacity by offering a lead-time exceeding one planning period. If the inventory is available, the manufacturer can produce to stock items that are not necessarily consumed in the current

period. In contrast, in the typical service-industry RM model (see McGill and van Ryzin (1999) and Talluri and van Ryzin (2004)) overbooking results in a financial penalty, but its impact on future capacity is not modeled.

Third, there is more flexibility in scheduling customer orders in manufacturing industry, as long as it satisfies customer demands within customer's acceptable due-date. While in the case of the airline industry, there is limited flexibility in substituting a seat reserved in a flight with another seat in some other flight. Such substitutions are done to meet overbooking in airline, but incur significant penalty costs. Therefore, the aspect of order scheduling decisions under the constraint of production capacity presents new challenges in RM.

Due to the special features of RM in manufacturing industry, it is impossible to apply the existing RM models in service industry. In other words, we need to extend the research area of RM from traditional industries with perishable products, such as airlines and hotels, to the manufacturing industry with non-perishable products.

1.4 Contributions and Document Organization

In Chapter 2, we examine a situation in which a MTO manufacturing firm faces price and lead-time dependent demand and must jointly determine the price, lead-time and production schedule for a finite planning horizon. The goal is to maximize the total profit gained in a finite planning horizon under the production capacity, lead-time and demand constraint. We develop a mathematical model for the joint decision on pricing, lead-time and production

decisions with a goal to maximize profit in the face of a demand function that links price and lead-time to demand for MTO firm. We also develop a rounding heuristic to provide the initial feasible solution to the MIP solver, so that the solution quality and time has been improved. From the numerical analysis, we get some useful insights regarding the price and lead-time flexibilities.

However, in MTO manufacturing it is generally not possible to obtain accurate forecast information about the timing and attributes of expected future orders over the planning horizon (Wu and Chiang, 2009). Therefore, to deal with the stochastic demands, we propose a reinforcement learning (RL) algorithm to solve the same problem in an infinite time horizon. The RL is derived from dynamic programming (DP) and stochastic approximation, which will be introduced in details in in Chapter 3. This stochastic version problem have the same conditions as the deterministic problem which we studied in Chapter 2 except that the stochastic demands and the objective is to maximize the long-run average total profit. In this Chapter 3, different scenarios in MTO system have been evaluated to compare the proposed RL with the First-Come-First-Serve (FCFS) policy and the threshold heuristic developed based on the order acceptance threshold heuristic proposed by Hing et al. (2007). The results shows that the proposed algorithm outperforms the other two policies in all scenarios. To the best of our knowledge, this is the first that uses RL as a method for solving joint pricing, lead-time, order acceptance and production scheduling decision problem.

In Chapter 4, we consider a RM problem in a MTS manufacturing system which consists of a single machine in which the demands and the processing times for N types of products are random. The problem decide when, what, and how much (the lot size) to produce so that the

long-run average total profit is maximized. We develop a mathematical model and propose two reinforcement learning (RL) algorithms, QLS and QLIH, for real-time decision-making. The QLS outperforms the two benchmarking policies, Brownian and First-Come-First-Serve (FCFS) policies in most cases. The QLIH outperforms QLS and the two benchmarking policies in all cases.

In Chapter 5, we summarize the work and demonstrate its potential applications. Future research directions are also pointed out.

Chapter 2

Joint Optimization on Pricing, Lead-time and Scheduling Decisions for Revenue Management in a Make-to-Order System

2.1 Introduction

During the past decade, there has been increasing interest in the integration of pricing and scheduling decisions. However, such problems may not have been practical before since manufacturers have not traditionally been in direct contact with end customers, making it difficult to influence demands by quoted price. Nowadays direct selling becomes more common due to advancements in technology, particularly with the Internet and thus makes

it easy to observe and influence end customer behavior (Boyd and Bilegan, 2003). Under this circumstance, joint pricing and production scheduling problems arise in the manufacturing firm. As reported by Direct Selling Association, the direct selling becomes a significant business sector, which have grown 2.6% from 1999 to 2008 and reached $29.6 billion in 2008. Due to the direct selling, manufacturing firms can maintain close customer relationship, as is currently done by the computer companies, such as Dell. For instance, Dell Computers separates its customers into different classes and may charge them different prices. Other companies, such as IBM and HP, are also adopting the same strategy. To name another example for a different industry, between 1995 and 1999, Ford Motor Co. credits $3 billion in growth for the effort of using pricing strategies to match supply, demand and target specific customer segments (Leibs, 2000).

Price is used to be considered as a key factor to win competitive advantage. However, the success of the Japanese manufacturing Just-In-Time (JIT) philosophy has highlighted the importance of short lead-times and further increased the level of competition. In today's time-based competitive market, practitioners recognized that customer demand increases not only with lower prices but also shorter lead-times (So, 2000; So and Song, 1998; Stalk and Hout, 1990). Lead-time (due-date) decisions depend on several factors, such as the manufacturer's capacity and the customers' demands and due-date preferences. With recognition of the lead-time sensitive demands, many companies, specifically the make-to-order (MTO) manufacturing sectors, are offering a uniform lead-time for all customers within which they guarantee to satisfy "most" orders (Rao et al., 2000; So and Song, 1998). While this strategy may attract many customers, there is a risk that the increasing demand may exceed capacity

limitation. This can incur a lateness penalty cost for the manufacturers or customer dissatisfaction and may lead to loss in future business. Firms then must trade-off the price, lead-time and potential for increased demand against capacity constraints.

In this chapter, we examine a situation in which a firm faces price and lead-time dependent demand and must jointly determine the price, lead-time and production schedule for a finite planning horizon. The goal is to maximize the total profit gained in a finite planning horizon under the production capacity, lead-time and demand constraint. This situation is common in MTO firms. The firm is a joint price and lead-time decision maker who has power to influence the customers' decisions for order quantities. But they can not predict customers' demands. In this chapter, the lead-time is the time period from the moment the manufacturer receives an order to the moment the order is shipped as the customers are assume to be responsible for the delivery costs. We consider a single machine production system with perfect reliability. Manufacturing fixed costs (including the overhead cost, setup cost and quality cost) should be considered in MTO production lines because of the customized orders for different customers. The fixed costs are time-independent and order-independent. Customer orders differ in their arrival times, possible prices and lead-times that can be quoted, quantities demanded and latest acceptable due-dates.

The manufacturer has the option to accept or reject an order. The order can be rejected if either the quoted price is higher than the customers' willingness to pay or the quoted lead-time is longer than the customers' willingness to wait (i.e., customer's latest acceptable due-date). However, the rejection may lead to the loss of future business which incurs the lost opportunity cost. For all of the accepted orders, the manufacturer sets the production and

delivery schedule within the planning horizon. The production of an order can be completed in different periods. However, we assume that the delivery of an accepted order must be integrated in an one-time shipment and the order must be shipped between its arrival time plus quoted lead-time and the customer's latest acceptable due-date. This assumption is reasonable when customers are responsible for the delivery costs and prefer to receive the order at one time.

An order completed before its quoted lead-time is considered early and incur inventory holding costs. The firm also pays lateness penalty cost whenever the actual lead-time (i.e., delivery date) exceeds the quoted lead-time. We assume that customer is patient enough to accept a late order as long as the order delivery date is within its latest acceptable due-date. Manufacturing variable costs are linear variable cost in addition to a manufacturing fixed cost for each production run. Production capacity, inventory holding cost and lateness penalty cost is known in each period and may be time varying. Manufacturing variable cost (including costs of raw material, direct labor and maintenance) is a linear variable cost in addition to a fixed cost for each production period. Variable cost, inventory holding cost and lateness penalty cost is known and time-independent. Therefore, The total profit is the total revenue minus the total cost.

The above model of the firm is a good representation of any MTO manufacturing firms that offer customized orders. An example of this type of firm can be found in iron and steel industry as described by Spengler et al. (2007). A few raw materials are converted into a variety of products which are made to order in terms of steel grade, dimensions, and surface treatment.

2.2 Literature Review

This research is related to the literature on joint production and pricing decision making. The extensive review of this literature can be found in the surveys of Eliashberg and Steinberg (1993) and Yano and Gilbert (2004). One of the earliest paper in which price and production quantity are both the decision variables in an EOQ framework is made by Whitin (1955). The literature can be classified into two categories: optimal control approaches (Chen and Chu, 2003; Feichtinger and Hartl, 1985; Pekelman, 1974; Vanthienen, 1975); and mathematical programming approaches (Ahn et al., 2007; Charnsirisakskul et al., 2006; Deng and Yano, 2006; Gilbert, 1999, 2000; Kim and Lee, 1998).

The optimal control approaches incorporate both static and dynamic pricing, and determination of production quantities. They use linear, convex and non-smooth functions for production and inventory costs while the demand functions are always assumed to be linear. Pekelman (1974) develops an optimal control model for profit maximization with both price and production quantities as decision variables over a planning horizon. Vanthienen (1975) incorporates a capacity constraint into Pekelman's model. Pekelman's model is further extended by allowing backlogging in the paper of Feichtinger and Hartl (1985). Chen and Chu (2003) find the optimal choice between production rate which may affect the inventory level and the optimal sales rate at each time through different pricing strategies in order to achieve the maximum profit for a given planning horizon. The research above do not consider limits on capacity or storage.

Mathematical programming models divide the time horizon into discrete time buckets and are closely related to production planning models. Kim and Lee (1998) address the problem of joint pricing, lot sizing and capacity expansion decision over a planning horizon. Gilbert (1999) studies version of a model in which a constant price is to be maintained for a seasonal product in advance of demand observation. Gilbert (1999) extends his model to consider a multi-product scenario in which different products share the common production capacity. Deng and Yano (2006) give the optimal solution of price and production quantities for a manufacturer facing price-sensitive demand, in consideration of both capacity constraint and setup costs in finite discrete time horizon. Deng and Yano (2006) present a model that integrates pricing, production and order selection decisions for MTO manufacturer who can set price to influence demand and set lead times for accepted orders. Ahn et al. (2007) considered a joint production and pricing decision problem, accounting for that portion of demand realized in each period that is induced by the interaction of pricing decision in the current period and in previous period. All of the models fail to incorporate the influence of the lead-time on demand function.

The second body of research related to our research focuses on due-date quotation. For an overview of due-date management policies, see Keskinocak and Tayur (2004). A large number of studies are conducted on due-date assignment in the production scheduling field. For the summary of recent works, see Philipoom et al. (1994), Lawrence (1995) and Easton and Moodie (1999). Duenya and Hopp (1995) and Duenya (1995) firstly incorporate the customer order selection to the firm's lead-time policy where the probability that a customer places the order decreases as the quoted lead-time increases. Some researchers consider the

order acceptance problem, where the manufacturer has options to reject orders and accepted orders must be finished by specified due-dates with the objectives to maximize revenue or profit (Charnsirisakskul et al., 2006; Hall and Magazine, 1994; Keskinocak et al., 2001).

The third body of research related to our research is revenue management (RM) based order acceptance with the limited manufacturing capacity. Some examples of the earliest models for selective order acceptance policies that can be applied in MTO systems are developed by Miller (1969) and Lippman and Ross (1971). Their models assume that the order service times are exponentially distributed and there are no due-date constraints or lateness penalties in producing the orders. Miller (1969) studies the order acceptance problem as an admission control problem to a queue. Lippman and Ross (1971) extend Miller's model by allowing service times that are dependent on the customer classes and a general arrival process. One of the key insights from these models is that in an MTO system with exponentially distributed processing times, a $c\mu$ policy gives optimal results. c is the revenue earned after a job is completed while the job's mean processing time is $1/\mu$. A $c\mu$ policy states that if the job with largest value of $c\mu$, among all available jobs, is chosen for scheduling, it maximizes the total expected returns. In the recent study, Spengler et al. (2007) develop a RM approach for companies in the iron and steel industry. The aim is to improve short-term order selection. Due to the complexity of obtaining opportunity costs from dynamic programming approaches, a static bid-price based approximation scheme for selecting profitable orders is implemented to determines bid-prices from a multi-dimensional knapsack problem formulation.

This research is more relevant to the literature that consider price, lead-time quotation, production quantities and capacity investment simultaneously. So and Song (1998) present an optimization model to determine the joint optimal selection of price, lead-time and capacity investment with an objective of maximizing the average net profit where price is sensitive to both price and lead-time. However, they do not consider lateness penalty costs in the objective function and quote uniform delivery times for every order. Palaka et al. (1998) model the firm's operations as an M/M/1 queue and treat the demand as being linear in quoted price and lead-time. The objective is to maximize revenues less total variable production costs, congestion related costs and lateness penalty costs subject to a service level constraint. Webster (2002) develops dynamic pricing and lead-time policies for a MTO system using the similar linear demand function. In the later research, use a RM approach for companies in the iron and steel industry was developed. The aim was to improve short-term order selection. Due to the complexity of obtaining opportunity costs from dynamic programming approaches, a static bid-price based approximation scheme for selecting profitable orders was implemented to determines bid-prices from a multi-dimensional knapsack problem formulation.

In this study, we model customer demand as a linear function of price and lead-time as in So and Song (1998), but we consider the production cost (incorporated in the manufacturing variable cost), lateness patently cost, inventory holding cost as in Charnsirisakskul et al. (2006). In addition to these costs, we also consider the setup costs (incorporated in the manufacturing fixed cost) incurred in each production period mentioned in the study of Deng and Yano (2006). Our major contribution is that we firstly incorporate the order

acceptance decision and manufacturing fixed cost on joint pricing, lead-time and production decision problem with lead-time and price sensitive demands and capacity constraints.

2.3 Problem Definition

In this chapter, we assume customer demands are known and predictable. We consider a firm that serves customers in a MTO fashion. In this model, we assume that the manufacturer can charge different prices to each customers since customer will have the customized order. Our customers are delay sensitive and thus the order quantity is a function quoted lead-times as well as price. We assume that the demand is downward sloping in both quoted lead-times and prices and the demand to be a linear function of quoted lead-time as in constraint (4.2). The decision process is illustrated in Section 2.3.1 and the firm's cost structure is given in Section 2.3.2.

2.3.1 The decision process of the problem

The figure 2.1 illustrates the decision process of the problem. Customer orders differ in their arrival times, possible prices and lead-times that can be quoted, quantities demanded and latest acceptable due-dates. The demand quantities are determined by a demand function which is dependent on the quoted lead-time and price. The accepted orders are scheduled for production and delivery within the manufacturer's capacity. Since for this problem, the order/demand arrival times are known in advance, the manufacturer makes the decisions on

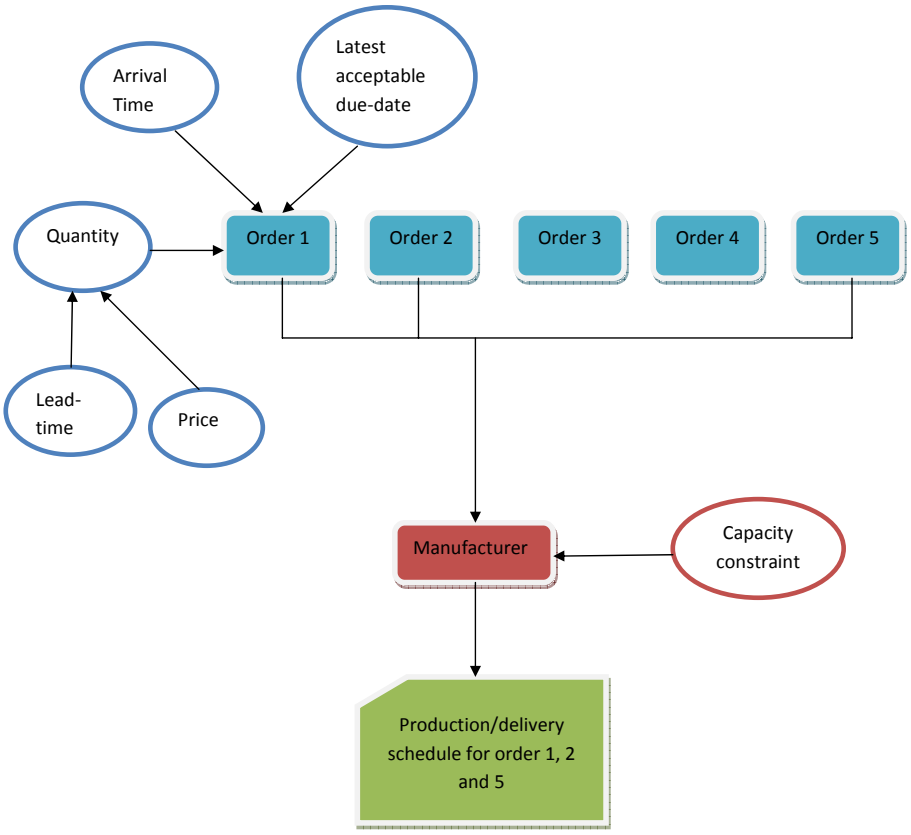

Figure 2.1: The decision process of the problem

order acceptance, price, lead-time and scheduling simultaneously in order to maximize the total profit over a given planning horizon.

2.3.2 Cost structure

As shown in figure 2.2, the firm's cost structure is composed of manufacturing cost, inventory holding cost, lateness penalty cost and lost opportunity cost.

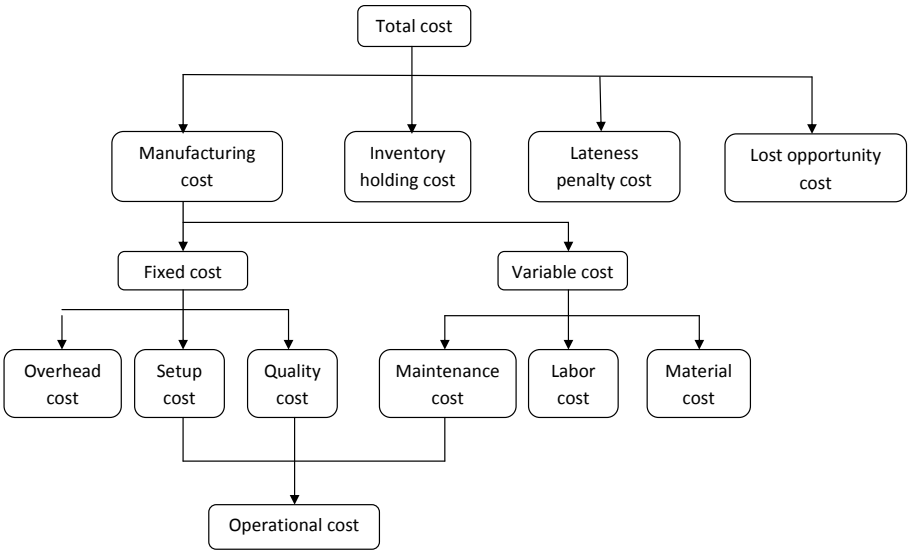

Figure 2.2: Cost structure

2.3.3 Manufacturing cost

The manufacturing costs include costs of material, labor, fixed overhead, and operational cost such as maintenance, setup and quality cost. The manufacturing cost can be divided into fixed cost and variable cost.

Fixed cost

In our research, manufacturing fixed costs are these costs incurred whenever the production takes place in each production period. The cost includes overhead cost, setup cost and quality cost. The mathematical model is a deterministic model that maximizes the total profit over a given planning horizon. Each planning period is considered as a work shift (e.g.,three shifts per day), in which workers need to take a break and be replaced by another shift. At the same time, the production also needs to resume for the new shift.

Variable cost

Manufacturing variable cost combines costs of raw material and direct labor and maintenance incurred in producing products, which is proportional to production quantities. The products have a broad definition in our research: it can be a real product (e.g., a lamp) or a virtual product (e.g., packing service). It is clear that it spends limited resources (raw material and labor) on producing any kind of products in a manufacturing firm. On the other hand, the provision of any type of services also consumes the limited labor (and raw material in some cases), thus the manufacturing variable cost can be considered as a labor cost. There are vast literature in joint pricing, lead-time and scheduling decisions, which incorporate the manufacturing variable cost (Charnsirisakskul et al., 2006; Deng and Yano, 2006; Palaka et al., 1998; So and Song, 1998).

Inventory holding cost

Inventory holding cost is defined as these costs that include warehousing and storage costs and labor to operate the space, insurance, security, interest on money invested in the inventory and space, and other direct expenses. We assume that the costs is the number of products completed before its quoted lead-time multiple the number of periods in inventory multiple the unit holding cost. Inventory holding cost has been incorporated in many researches on scheduling problem (Anupindi and Tayur, 1998; Paternina-Arboleda and Das, 2005; Qiu and Loulou, 1995; Sox and Muckstadt, 1997). However, due to the difficulties in obtaining the optimality, there are few related research incorporating the holding costs, except for the study of So and Song (1998) and Charnsirisakskul et al. (2006).

Lateness penalty cost

The lateness penalty cost is defined as these penalties that reflect direct compensation paid to the customers for not meeting the quoted lead-times. We assume that these costs are proportional to the number of periods and the quantity delay. The latest acceptable lead-time and the tardiness penalty may depend upon negotiations between each customer and the firm. The lateness penalty cost is a common cost that can be found in the real-world applications. However, in the related literature, it only can be found in the study of Charnsirisakskul et al. (2006).

Lost opportunity cost

The lost opportunity cost is defined as a penalty due to loss of goodwill, and the potential loss of future business because the orders is rejected. The order may be rejected due to the firm's limited capacity. The rejected order also can be considered as the quoted lead-time is longer than the customers' willingness to wait (i.e., the latest acceptable due-date). However, since the customer's order is reject, the firm loses customer loyalty. It is most likely that the firm loses its long-term profits because the customer may switch to other suppliers in the future.

2.4 Model Formulation

2.4.1 Notation

Sets:

$\mathcal{T}$={1,2,...,T} planning periods

$\mathcal{O}$={1,2,...,n} Customer order

$\mathcal{P}(i)$=$\{p_1^i, p_2^i, ...p_{n_i}^i\}$ prices (per unit) the firm can charge for order i

$\mathcal{L}(i)$=$\{l_1^i, l_2^i, ...l_{m_i}^i\}$ the lead-time the firm can quote for order i

Parameters:

a_i: arrival time of order i, ($i \in \mathcal{O}$)

c_i: variable cost per unit of order i, ($i \in \mathcal{O}$)

h_i: holding cost per period per unit of order i, ($i \in \mathcal{O}$)

b_i: lateness penalty per period per unit of order i, ($i \in \mathcal{O}$)

s_i: fixed cost of order i, ($i \in \mathcal{O}$)

r_i: rejection cost for order i ($i \in \mathcal{O}$)

k_t: units of production capacity available in period t, ($t \in \mathcal{T}$)

e_i: the latest acceptable due-dates of order i, ($i \in \mathcal{O}$)

v: demand corresponding to zero quoted price and zero quoted lead-time

w: price sensitivity of demand

u: lead-time sensitivity of demand

2.4.2 Model

In this model, we assume that the manufacturer can customize price and lead-time, i.e., charge a different price and lead-time to different customers. This might be the case if customers can be differentiated by geographical differences. We formulate the manufacturer's profit maximization problem as a Mixed Integer Program (MIP) as follows.

Decision variables:

$x^i_{t,k}$: quantity produced (in units of capacity) for order i in period t and delivered in period k ($t = a^i, ..., k, k = a_i, ...e_i, i \in \mathcal{O}$)

$d^i_{j,q}$: implicit decision variable, order quantity of order i corresponding to price p^i_j and lead-time l^i_q ($j = 1, ..., n_i, q = 1, ..., m_i, i \in \mathcal{O}$)

I^i_k: 1 if order i is accepted and delivered in period k; 0, otherwise ($k = a_i, ..., e_i$)

$Z^i_{j,q}$: 1 if price p^i_j and lead-time l^i_q are quoted for order i 0, otherwise ($j = 1, ..., n_i, q = 1, ..., m_i, i \in \mathcal{O}$)

δ^i_t: fixed cost indicator in period t for order i (1 if $x^i_{t,k} > 0$; 0, otherwise)

ϕ^i order i rejection indicator (1 if $\sum_{j=1}^{n_i} \sum_{q=1}^{m_i} Z^i_{j,q} = 0$; 0, otherwise)

$H^i_{j,q}$: total quantity-period holding inventory of order i corresponding to price p^i_j and lead-time l^i_q

$B^i_{j,q}$: total quantity-period tardiness of order i corresponding to price p^i_j and lead-time l^i_q

Objective function

maximize total profit:

$$\sum_{i\in\mathcal{O}}\sum_{j=1}^{n_i}\sum_{q=1}^{m_i}p_j^i d_{j,q}^i Z_{j,q}^i - \sum_{i\in\mathcal{O}}\sum_{k=a_i+l_q^i}^{e_i}\sum_{t=a_i}^{k}c_i x_{t,k}^i - \sum_{i\in\mathcal{O}}\sum_{t=1}^{T}\delta_t^i s_i - \sum_{i\in\mathcal{O}}\sum_{j=1}^{n_i}\sum_{q=1}^{m_i}h_i H_{j,q}^i - \sum_{i\in\mathcal{O}}\sum_{j=1}^{n_i}\sum_{q=1}^{m_i}b_i B_{j,q}^i - \sum_{i\in\mathcal{O}}r_i\phi^i \tag{2.1}$$

Constraints:

Order quantity constraints:

$$d_{j,q}^i = v - wp_j^i - ul_q^i \qquad \forall i\in\mathcal{O}, j=1,...,n_i, q=1,...,m_i \tag{2.2}$$

Demand constraints:

$$\sum_{k=a_i+l_q^i}^{e_i}\sum_{t=a_i}^{k}x_{t,k}^i = \sum_{j=1}^{n_i}\sum_{q=1}^{m_i}d_{j,q}^i Z_{j,q}^i \qquad \forall i\in\mathcal{O} \tag{2.3}$$

Capacity constraints:

$$\sum_{i\in\mathcal{O}|t\geq a_i;t\leq e_i}\sum_{k=a_i+l_q^i}^{e_i}x_{t,k}^i \leq K_t \qquad \forall t\in\mathcal{T} \tag{2.4}$$

Price and lead-time selection constraints:

$$\sum_{j=1}^{n_i}\sum_{q=1}^{m_i}Z_{j,q}^i \leq 1 \qquad \forall i\in\mathcal{O} \tag{2.5}$$

$$a_i + l_q^i \leq e_i \qquad \forall i\in O \tag{2.6}$$

Delivery quantity and time constraints:

$$\sum_{t=a_i}^{k} x_{t,k}^i \leq M_1 I_k^i \qquad k = a_i + l_q^i, ...e_i, \forall i \in \mathcal{O} \tag{2.7}$$

$$\sum_{k=a_i+l_q^i}^{e_i} I_k^i \leq \sum_{j=1}^{n_i}\sum_{q=1}^{m_i} Z_{j,q}^i \qquad \forall i \in \mathcal{O} \tag{2.8}$$

$$\sum_{k=e_{j,q}^i+1}^{e_i} I_k^i \leq 1 - Z_{j,q}^i \qquad j = 1, ..., n_i, q = 1, ..., m_i, \forall i \in \mathcal{O} \tag{2.9}$$

Manufacturing fixed cost (e.g. setup cost) indicator constraint:

$$x_{t,k}^i \leq M_2 \delta_t^i \qquad t = a_i, ...k, k = a_i + l_q^i, ...e_i, \forall i \in \mathcal{O} \tag{2.10}$$

Order rejection indicator constraint:

$$1 - \sum_{j=1}^{n_i}\sum_{q=1}^{m_i} Z_{j,q}^i = \phi^i \qquad \forall i \in \mathcal{O} \tag{2.11}$$

Quantity-period holding inventory:

$$H_{j,q}^i \geq \sum_{k=a_i+l_q^i}^{e_i}\sum_{t=a_i}^{k}(k-t)x_{t,k}^i + M_3(Z_{j,q}^i - 1) \qquad j = 1, ..., n_i, q = 1, ..., m_i, \forall i \in \mathcal{O} \tag{2.12}$$

Quantity-period tardiness:

$$B^i_{j,q} \geq \sum_{k=a_i+l^i_q}^{e_i} \sum_{t=a_i}^{k} (k - a_i - l^i_q) x^i_{t,k} + M_4(Z^i_{j,q} - 1) \qquad j = 1, ..., n_i, q = 1, ..., m_i, \forall i \in \mathcal{O} \quad (2.13)$$

Variable constraints:

$$x^i_{t,k} \geq 0 \qquad t = a_i, ...k, k = a_i + l^i_q, ...e_i, \forall i \in \mathcal{O} \quad (2.14)$$

$$B^i_{j,q}, H^i_{j,q} \geq 0 \qquad j = 1, ..., n_i, q = 1, ..., m_i, \forall i \in \mathcal{O} \quad (2.15)$$

$$I^i_k \in \{0, 1\} \qquad k = a_i + l^i_q, ...e_i, \forall i \in O \quad (2.16)$$

$$Z^i_{j,q} \in \{0, 1\} \qquad j = 1, ..., n_i, q = 1, ..., m_i, \forall i \in \mathcal{O} \quad (2.17)$$

$$\delta^i_t \in \{0, 1\} \qquad j = 1, ..., n_i, q = 1, ..., m_i, \forall i \in \mathcal{O} \quad (2.18)$$

In the objective function (2.1), the total profit is composed of five elements: the total revenue, manufacturing variable cost, manufacturing fixed cost, inventory holding cost and lateness penalty cost. In constraint (2.2), we assume order quantity or customer demand is downward- sloping in both price and quoted lead-time, which is a linear function of the quoted price and lead-time. Demand constraints (2.3) ensures that if price p^i_j and lead-time

l_q^i are quoted, the exact number of demand $d_{j,q}^i$ must be produced and delivered. Capacity constraint (2.4) ensures that the production capacity in each period will not exceed. Constraint (2.5) ensures that the order is either be accepted or rejected. If accepted, the order can be only quoted one price and one lead-time. Constraint (2.6) states that the order arrival time a_i plus the quoted lead-time l_q^i is less that the latest acceptable order due-date. This implies that customers refuse to accept the orders which are later than their acceptable latest due-date corresponding to quoted price and lead-time. Constraint (2.7) states that every production must be delivered in a specific period. M_1 is a sufficient large number, such as $\max_{j,q}\{d_{j,q}^i\}$. Constraint (2.8) ensure that an order can be only delivered once between $(a_i + l_q^i)$ to e_i. Constraints (2.9) guarantees that the order can not be delivered after the latest acceptable due-date if the price and lead-time are quoted. Constraints (2.10) states that if production for order i begins in a period t, the setup indicator δ_t^i is 1; otherwise, zero. M_2 is a sufficient large number, such as $\max_{j,q}\{d_{j,q}^i\}$. Constraint (2.11) states that if the order is rejected, the order rejection indicator is 1. Constraint (2.12) calculates the total quantity-period inventory of each order at price p_j^i and lead-time l_q^i. The $H_{j,q}^i$ is positive only if $Z_{j,q}^i$ equals to 1 when price p_j^i and lead-time l_q^i are selected. M_3 is a sufficient large number, such as $(e_i - a_i)(\max_{j,q}\{d_{j,q}^i\})$. Similarly, constraint (2.13) calculates the total quantity-period tardiness and $B_{j,q}^i$ is positive only if the price and lead-time are selected. M_4 is a sufficient large number, such as $(e_i - a_i - l_q^i)(\max_{j,q}\{d_{j,q}^i\})$. Constraints (2.14) to (2.16) are non-negativity constraints and constraints (2.17) to (2.18) specify binary variables.

In this model, there are $(7n + T + 2\sum_{i=1}^{n}(e_i - a_i - l_q^i) + 6\sum_{i=1}^{n}(n_i m_i) + 2\sum_{i=1}^{n}(e_i - a_i - l_q^i)(e_i - a_i))$ constraints, $(nT^2 + 2\sum_{i=1}^{n}(n_i m_i))$ nonnegative decision variables and $(n + 2nT +$

$\sum_{i=1}^{n}(n_i m_i)$) binary decision variables. Therefore, the formulation is of size $O(nT^2)\times O(nT^2)$, where n is the number of arrival orders and T is the number of time periods of the planning horizon.

2.5 Numerical Analysis

The numerical analysis presented in this section is designed to develop insights regarding the benefits of price and lead-time flexibilities, under various demand environments.

2.5.1 Experiment Design

In the numerical experiment, we assume that each customer is a retailer who faces various demand scenarios. The price and the lead-time follow Uniform [20, 80] and Uniform [10, 20] distributions, respectively. We assume the order quantity function is determined by Equation 2.2, where u=600, v=2 and w=3, respectively. The the manufacturing fixed cost, variable cost, unit holding and unit tardiness costs per period are generated from Uniform [2, 5], Uniform [10, 15], Uniform[2, 5] and [2, 4] for arrival orders , respectively. Since the lost sale may have the negative influence the manufacturer's future sale, we assume the lost opportunity cost is 20. We assume one unit of production capacity ($K_t = 1$) in each period. Therefore, the processing time of order i is $pt^i_{j,q} = d^i_{j,q}$. To ensure that the latest acceptable due-dates of all orders are within the planning horizon, the length of the planning horizon is extended to $|\mathcal{T}|$=max$_i\{e_i\}$.

In the model, there are three factors that affect the demand environment: order size (S), demand load (L), and order arrival time distribution (D). We focus on two forms of flexibilities which have impact on the manufacturing environment: price and lead-time flexibilities.

Order Size indicates the amount of production resources (in units of capacity) required to complete an order. The order size is determined by the price distribution and lead-time distribution.

Demand Load indicates the congestion in the system and is defined as the expected ratio of total requested production capacity and total available capacity over the forecast horizon. The demand load is directly related to the order size and the number of orders that arrive during the forecast horizon. Since the demanded quantity of an order depends on the price and lead-time, we use the average of quantities of the orders in defining the demand load. In this numerical study, we use three levels of demand load, Low: 0.5, Medium: 1.0, and High: 1.5. The number of orders that arrive during the forecast horizon is determined by the expression: $N = \frac{L \cdot |\mathcal{T}|}{S_{medium}}$.

Order Arrival Time Distribution represents seasonality or the variations of demands. Three types of arrival time distributions are considered: high arrivals near the beginning (LT: triangular[1, 20, 80]), the middle (MT: triangular[1, 40, 80]), and the end (RT: triangular[1, 60, 80]) of the planning horizon.

Price Flexibility describes the flexibility to charge different prices to different customers. In some cases, for example, when manufacturer wants to sell the low-end product in similar price in a promotion or the customers cannot be segmented due to reasons such as legal

regulations or difficulties in identifying geographic differences, the manufacturer may not be able to price discriminate among the customers. In this case, the manufacturer has to quote the best single price to all customers. When there is no price flexibility, a single (fixed) price is charged to all customers.

Lead-time Flexibility refers to the flexibility to extend their latest acceptable due-dates. The lead-time flexibility indicator (W) denotes the ratio of the length of the lead-time window (the time between quoted lead-time and latest acceptable due-dates) to the processing time of the order $pt^i_{j,q}$. To illustrate the effect of lead-time flexibility, the latest acceptable due-date is $e_i = \max_{j,q}\{l^i_q + Lpt^i_{j,q}\}$, in which l^i_q is quoted lead-time of order i. In each experiment, We use the same (L) for all orders. In the analysis, we compare systems with different lead-time flexibility levels: no lead-time flexibility ($L = 0$), low lead-time flexibility ($L = 3$), and high lead-time flexibility ($L = 5$), denoted by $L0$, $L3$, and $L5$, respectively.

2.5.2 Computational Issues

The problem size becomes larger with more price and lead-time selection, higher demand load, larger order numbers and longer planning horizon. It takes a long time to find a feasible MIP solution in CPLEX when the problem size increases. However, the solution time can be reduced considerably when an initial feasible MIP solution is input into the solver. We propose three LP-based Binary Rounding Heuristics (BRH) for finding good initial solutions. We solve the LP relaxation and then interactively round the fractional binary integer variables in LP relaxation.

Binary Rounding Heuristics

In the model formulation in Section 2.4, there are three sets of binary variables $Z^i_{j,q}$, I^i_k, δ^i_t and ϕ^i. However, it is difficult to find a feasible solution by rounding the four sets of binary variables at the same time. For example, if a rounding heuristic fixes $Z^i_{j,q} = 1$, the heuristic may not find a feasible solution by rounding I^i_k. Therefore, to simplify finding a feasible solution in the binary rounding heuristic, we introduce an alternative model for formulation. The decision variables $Z^i_{j,q}$, I^i_k and $B^i_{j,q}$ are removed and introduce new binary variable $Z^i_{k,j,q}$ which equals to 1 if order i is quoted price p^i_j and l^i_q and delivered in period k and 0 otherwise. However, we can always round δ^i_t to 1 after a rounding heuristic fixes $Z^i_{k,j,q} = 1$ because this will not lead to infeasibility as shown in constraint (2.10). The $Z^i_{j,q}$ in objective function (2.1), constraints (2.3), (2.5), (2.12) and (2.13) is replaced by $\sum_{k=a_i+l^i_q}^{e_i} Z^i_{k,j,q}$. In addition, the ϕ^i in the objective function can be replaced by $(1 - \sum_{j=1}^{n_i}\sum_{q=1}^{m_i} Z^i_{k,j,q})$. The constraints (2.8), (2.9), (2.11), (2.13) and (2.16) are removed. $B^i_{j,q}$ in the objective function is replaced by $d^i_{j,q}(\sum_{k=a_i+l^i_q}^{e_i}(k - a_i - l^i_q)Z^i_{k,j,q})$.

Algorithm 1: Binary Rounding Heuristic-1 (BRH-1)

1 **while** *not all binary variable are integers* **do**
2 | Solve the LP relaxation;
3 | Find "order" rounding index $\tilde{i} \leftarrow$ Pick orders not yet fixed with highest p_j^i;
4 | Find "delivery period" index $k \leftarrow$ find $Z_{k,j,q}^{\tilde{i}}$ with value closest to 0.5;
5 | Round $Z_{k,j,q}^{\tilde{i}}$ up (down) with probability equal to value 1 (otherwise, 0);
6 | Fix $Z_{k,j,q}^{\tilde{i}}$ in the LP ;
7 | **if** $Z_{k,j,q}^{\tilde{i}} = 1$ **then**
8 | | Find "production period" index $t \leftarrow$ find $X_{t,k}^{\tilde{i}}$ with value bigger than 0;
9 | | **if** *There is available capacity in period* t **then**
10 | | | Round δ_t^i up to value 1;
11 | | **else**
12 | | | Round δ_t^i down to value 0;
13 | | **end**
14 | | Fix δ_t^i in the LP ;
15 | **end**
16 **end**

The key point in the the BRH-1 is to firstly fix the order with highest price. However, in consideration of lead-time and profit, we propose the other two heuristics which inherit the most parts in BRH-1. In BRH-2 and BRH-3, we replace the priority selection rule to shortest l_j^i and highest $p_q^i d_{j,q}^i$ in step 4 of BRH-1, respectively.

The heuristics can be run repetitively with different seeds for randomization to obtain a better solution, we use a different random number sequence generated for the probabilistic rounding. In the numerical experiments, we will repeat the heuristic for 90 times and run BRH-1, BRH-2 and BRH-3 for 30 iterations, respectively. The best solution will be selected as the initial feasible solution and input into the solver.

For each demand scenario, five replications were simulated and the corresponding models were solved using CPLEX 11.0. The computational time varies from less than 10 seconds for easy instances to more than 12 hours for hard instances. The quality of the solution

varies from less than 0.1% to 5.0% optimality gap. Because of the problem complexity, we were not able to solve some of the instances to optimality. In our experiment, 92% of the tested cases achieve optimality and 5% of the cases have an optimality gap less than 0.1%. The instances where optimality cannot be achieved are those with high demand load and high lead-time flexibility. In such cases where optimality is not achieved, the analysis of the benefit of flexibility is based on the lower bound (the profit obtained from the best solution).

2.5.3 Numerical Results and Insights

We consider six different combinations of price and lead-time flexibility: S-L0 (single price and no lead-time flexibility), S-L3 (single price and low lead-time flexibility), S-L5 (single price and high lead-time flexibility), M-L0 (multiple price and no lead-time flexibility), S-L3 (multiple price and low lead-time flexibility) and S-L5 (multiple price and high lead-time flexibility).

With the binrary rounding heuristic, the CPU time required to solve the problems to optimality in CPLEX is compared to the CPU time required for the default CPLEX settings. For the hard instances, the CPU times are compared based on the computational time used to obtain the same feasible solutions. Table 2.1 shows the average percentage CPU time reduction under each demand load scenario for different levels of price and lead-time flexibility compared with the default setting of CPLEX. For example, in Table 2.1, the column corresponding to (Single price, L0) reports the average percentage CPU time reduction under Single price-no lead-time flexibility. First, the method combined with the binary rounding heuristic on average requires 11.12% less CPU time than when using the default CPLEX

settings, and the solution obtained from the binary rounding heuristic is close to the optimal solution. Second, the reduction in CPU time increases as the demand load and lead-time flexibility increase.

Table 2.1: Percentage CPU time reduction under each demand load scenario for different levels of price and lead-time flexibility

Demand load	Single price (S)			Multiple price (M)		
Scenario	L0	L3	L5	L0	L3	L5
Low	-5.08	-3.15	-1.8	-1.65	0.05	5.19
Medium	5.33	9.12	10.82	8.28	10.67	24.07
High	13.19	22.95	25.64	15.01	22.83	38.72
Average	4.48	9.64	11.55	7.21	11.18	22.66

Price Flexibility

Table 2.2 shows the percentage profit increases due to charging multiple prices in stead of single price under different levels of lead-time flexibility. Comparing the benefit of price flexibility under L0, L3 and L5, we find that under L0, the benefit of price flexibility is higher in L3 and L5 in all environments. However, on average, the percentage of price increased is not significantly different under L3 and L5. That is, in most environments with no lead-time flexibility, we observe a substitution effect between price and lead-time flexibilities.

Lead-time Flexibility

Table 2.3 reports the percentage profit increases due to L3 and L5 over L0. For instance, in Table 2.3, the column corresponding to (Single price, L3) reports the percentage increase in S-L3 over S-L0.

Table 2.2: Percentage profit increases over baseline (single price) under each environment for different levels of lead-time flexibility

Demand scenario	L0	L3	L5
(LT, L)	24.82	4.36	3.94
(LT, M)	27.73	7.53	6.98
(LT, H)	27.17	9.45	8.15
(MT, L)	30.61	6.77	9.03
(MT, M)	25.23	6.46	6.11
(MT, H)	31.82	13.02	12.03
(RT, L)	25.17	5.77	6.78
(RT, M)	21.13	3.69	6.43
(RT, H)	21.36	5.40	5.25
Average	26.12	6.94	7.19

Table 2.3: Percentage profit increases over baseline (L0) under each environment for different levels of price and lead-time flexibility

Demand	Single price (S)		Multiple price (M)	
scenario	L3	L5	L3	L5
(LT, L)	21.45	21.43	1.55	1.12
(LT, M)	23.70	25.48	4.13	5.09
(LT, H)	24.74	26.88	7.36	7.90
(MT, L)	23.94	20.81	1.32	0.85
(MT, M)	24.19	25.23	5.57	6.11
(MT, H)	24.13	27.85	6.43	8.65
(RT, L)	21.71	20.63	2.84	2.91
(RT, M)	23.34	23.79	5.58	8.77
(RT, H)	25.35	26.37	8.87	9.60
Average	23.62	24.27	4.85	5.67

First, we discuss the benefits of lead-time flexibility under environments without price flexibility (single price). If there is no price flexibility, L3 leads to higher profits than L0 in all environments. In comparison with L3, L5 increases profits over L0 in most environments, except in (MT, L) and (RT, L), where the demand load is low. If the demand load is low, medium lead-time L3 provides sufficient flexibility to satisfy most of the orders. If the the lead-time extends to L5, the order quantity will decrease according to our demand function. Therefore, the average profit will not increase too much compared with L3.

Second, when there is price flexibility (multiple price), the percentage profit increases less than that without price flexibility, which confirms our observation on the substitution effect between price and lead-time flexibility. Interestingly, with the price flexibility, L3 and L5 are more useful when the demand load is high, such as in (LT, M) and (MT, H). However, when the demand load is low, such as in (LT, L) and (MT, L), most orders can be satisfied by their best prices and corresponding order quantity.

2.5.4 Summary of Results

We compared five combinations of lead-time and price flexibility with the baseline (no lead-time flexibility and single price). Figure 2.3 and 2.4 show the impact of different combinations of price and lead-time flexibility on the profits and the order acceptance rate, respectively. Order acceptance rate is the number of accepted orders divided by the total number of arrival orders. Although the manufacturer would primarily be interested in maximizing profits, the order acceptance rate is also of interest as it reflects customer satisfaction, which is important for long-term profitability.

As shown in Figure 2.3, lead-time flexibility leads to higher profits under all combinations of price flexibility; however, it exhibits diminishing returns. In addition, Price flexibility is more useful when there is no lead-time flexibility. Under L0, the average percentage increase in profit over single price is 26% whereas they are 7% and 6% under L3 and L5, respectively. From Figure 2.4, higher price and lead-time flexibility leads to a higher order acceptance rate; However, the effects are diminishing as lead-time flexibility increases.

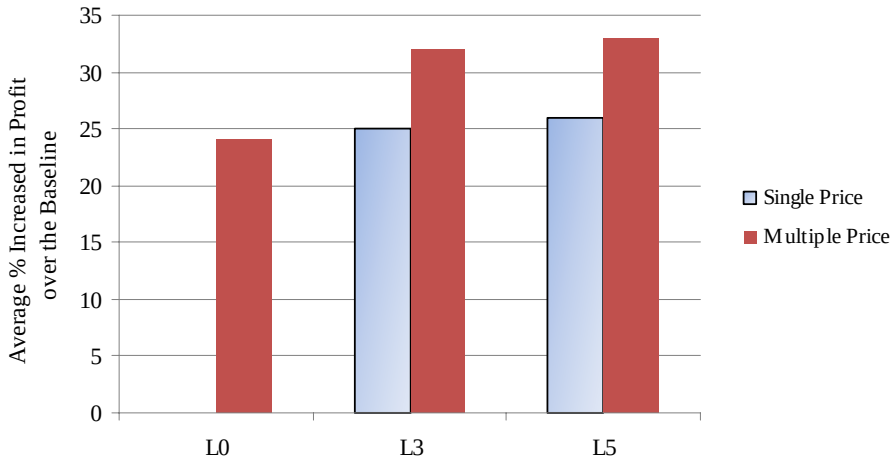

Figure 2.3: Average percentage profit increases over the baseline under different combinations of lead-time and price flexibility

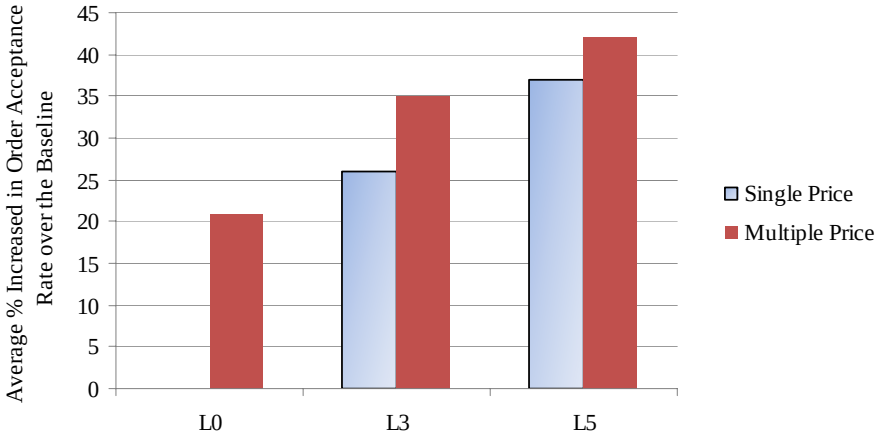

Figure 2.4: Average percentage order acceptance rate increase over the baseline under different combinations of lead-time and price flexibility

2.6 Summary

In this chapter, we study the joint optimization decisions on pricing, lead-time and scheduling in a MTO production system under a price and lead-time sensitive demand environment, where the manufacturer can quote different prices and lead-times to the customers. The goal of the research is to maximize the profit, which is the revenue minus costs. We develop

a mathematical model, which can be solved by optimization software CPLEX. We also develop a rounding heuristic to provide the initial feasible solution to the MIP solver so that the solution time and quality are considerably improved.

In the numerical experiment, we study the problem under different demand scenarios with price and lead-time flexibilities. Both of the price and lead-time flexibilities are useful to improve the profit and order acceptance rate. Furthermore, price and lead-time flexibilities have a substitution effect. That is, without the price flexibility, the effect of lead-time flexibility is more useful and vice versa. In general, when the demand load is high, the high level of lead-time flexibility is more useful.

Chapter 3

A Reinforcement Learning (RL) Approach to Revenue Management in a Make-to-Order System

3.1 Introduction on Reinforcement Learning (RL)

Reinforcement Learning (RL) was initially proposed in the early 1990s in the area of machine learning and has attracted a lot of interest from the research community. Different from another popular approach of supervised learning, in which the agent learns from examples provided by a knowledgeable external supervisor, in RL the learning agent learns by directly interacting with the environment in two ways. On the one hand, the learning agent in RL keeps interacting with its dynamic environment and uses a policy to choose actions based on the feedback from the environment; on the other hand, the environment responds to the

agent's actions by assigning rewards or punishment to the agent in a way that tends to increase the long-run average reward (Kaelbling et al., 1996).

As depicted in Figure 3.1, a single-agent learning model contains four elements: the system environment model, the learning agent, the policy set (a set of actions), and the response (intermediate reward or punishment) from the environment. The process is conducted iteratively; each iteration starts from a decision-making epoch (i.e., system state) and ends in another new epoch (state). In each decision-making epoch, the learning agent chooses an action according to its policy and based on the information about the current state: immediate reward or punishment obtained from the environment in the last iteration and the time of the last state-transition. As new demands arrive or the production completes, the state will transit to a new state. Then, the environment assigns the immediate reward or punishment to react to the action taken by the learning agent, which completes a state transition (iteration). At each iteration, the learning agent updates its knowledge using the RL algorithm and chooses the action according to its policy with the goal of maximizing the long-run average reward or minimizing the long-run average cost/punishment. This leads the system evolution to the next decision-making epoch and a state transition (iteration) is completed. As "good" actions are rewarded and "bad" actions are punished over time, some actions tend to be more and more preferable and some less. This learning process ends when the agent has a steady average reward, a stable trend appears in the knowledge, and a near-optimal policy on action selection is obtained.

RL has been proposed to learn near-optimal policies for large scale Markov decision process (MDP) and is widely used as a promising method in the area of machine learning.

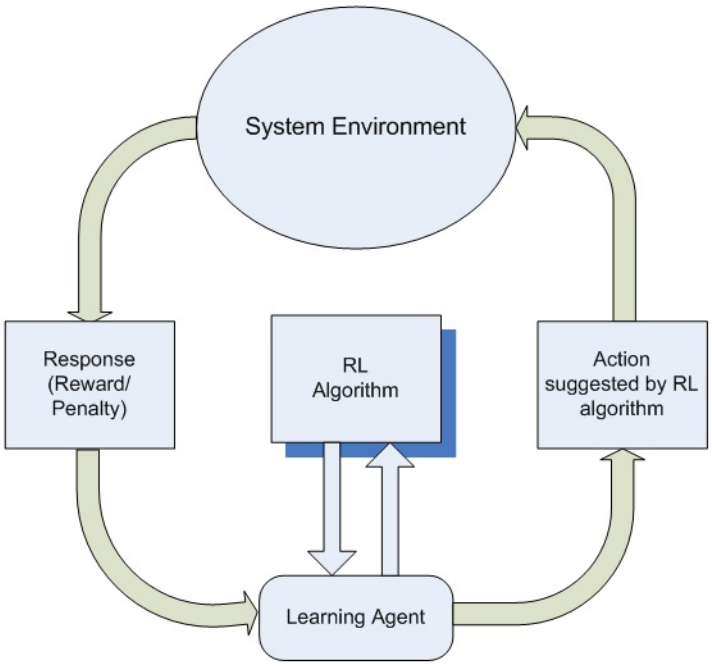

Figure 3.1: The mechanism of reinforcement learning

Various simulation-based RL methods are derived from stochastic approximation, such as the method of Temporal Differences, TD(λ) (Sutton, 1988) and Q-learning (Eliashberg and Steinberg, 1989). Stochastic approximation method is iteratively attempts to find zeros or extrema of functions which cannot be computed directly, but only estimated via noisy observations. The early studies on RL only consider maximizing the cumulative reward. Mahadevan (1996) firstly undertakes a detailed study on the average reward RL. Later, Das et al. (1999) develop a new model-free average-reward algorithm called SMART for continuous-time semi-Markov decision processes (SMDPs). Compared with classical Dynamic Programming (DP), RL avoids the need for computing the transition probability and the reward matrices. RL only requires the probability distributions of the process random

variables since it is based on discrete event simulation (Law and Kelton, 1991). RL not only overcomes some of the drawbacks of classical DP but also can be used in large state space, e.g., with 10^{15} states (Das et al., 1999). There are strong theoretical guarantees on convergence in most of RL algorithms (Szepesvari and Littman, 1996), such as Q-learning algorithm (Gosavi, 2004b). In addition, RL provides model-free learning of adequate control strategies. Thus, RL is an ideal method to solve the joint pricing, lead-time and production scheduling problem in a stochastic environment.

3.2 RL for Joint Pricing, Lead-time, Order Acceptance and Production Scheduling Decision Problem

Our problem is closely related to the production scheduling, revenue management (RM) and pricing problems. Here we enumerate a few cases that are related to our topic.

In literature, there are some applications of RL in production scheduling problem. Zeng and Sycara (1995) firstly apply a RL approach for a job-shop scheduling problem: They develop a repair-based scheduing that is trained using the temporal difference RL algorithm and that starts with a critical-path schedule and incrementally repairs constraint violations. Mahadevan et al. (1997a), Mahadevan et al. (1997b) and Das et al. (1999) apply the SMART algorithm to the problem of optimal preventative maintenance in a production inventory system, where there is a single machine capable of producing multiple types of products with multiple buffers for storing each of the different products. The SMART outperforms the two heuristics in determining proper maintenance schedules regarding to costs. Mahadevan

and Theocharous (1998) also apply SMART to the problem of optimizing a three-machine transfer line producing single type of products. They find that the policy from SMART is better than the kanban heuristic to maximize the throughput while minimizing WIP and failures. Paternina-Arboleda and Das (2001) extend the previous work to deal with a four-machine serial line and compared SMART to more existing control policies. They examined the system with constant demand rate and Poisson demand rate. Under these two circumstances, SMART outperformed those heuristic policies on average WIP level and average WIP costs.

Some researchers have worked on applying RL methods in RM, in which RM in airline industry is considered as the the first application area. The first paper related to this problem is a single leg airline RM problem (e.g., pricing and seat allocation) solved by Gosavi et al. (2002). They present an approximate DP or RL approach that is based on value iteration and employ function approximation with neural networks for estimating the value function of DP within a simulator to solve a continuous-time SMDP with random demands arrival and cancellations. Gosavi (2004a) further develops a RL algorithm base on policy interaction for the same problem. He tests the proposed RL algorithm with a nearest-neighbor approach on an airline RM problem to tackle a large state space.

There are several advantages of using RL in our problem. Firstly, RL is a simulation-based approach, thus, it only requires the probability distributions of the process random variables. Secondly, a combined approach of RL and discrete-event simulation approach avoids the difficulties of determining the price, lead-time and production sequence in a stochastic environment. Thirdly, many RL algorithms have been proved to converge and

allow a free choice of action (e.g., any action may be chosen from an action set according to a RL algorithm) in the learning process, such as in a Q-learning algorithm.

The Q-learning algorithm developed by (Eliashberg and Steinberg, 1989), is a temporal difference (TD) method that approximates $Q(s, a)$ directly, where (s, a) is a state-action pair. TD methods such as Q-learning are incremental algorithms that update the estimates of Q(s, a) on each time-step. The update formula is $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$. The parameter α is referred as the learning rate, which determines the size of the update made on each step. γ is referred as the discount rate, which determines the value of future rewards. r is the immediate reward obtained. The Q-learning algorithm is well suited for on-line applications and generally performs well in practice.

Although a great deal of work in algorithmic development and applications of RL has already occurred, RL continues to attract research attention. To the best of our knowledge, this is the first that uses RL as a method for solving joint pricing, lead-time, order acceptance and production scheduling decision problem. We adopt an infinite horizon model with the average reward (profit) performance criterion using one of the RL algorithm, Q-learning algorithm.

3.3 Problem Description

The firm considered in this problem faces price and lead-time dependent demand and must jointly determine the price, lead-time and production schedule for a given planning horizon under time-dependent production capacity constraints. It is assumed that there are multiple

types of customers who place an order with the demands dependent on the quoted lead-time and price. Whenever a customer place an order, the firm needs to make a decision whether to accept or reject the order. The order from different customers are assumed to arrive according to independent Poisson processes. The different orders have different sets of lead-times and prices to be quoted, arrival times and latest acceptable due-dates. An order can be delivered immediately only after the entire order is completed. The inventory holding cost is incurred if an order completed before its quoted lead-time and the lateness penalty cost is incurred whenever the actual lead-time (i.e., delivery date) exceeds the quoted lead-time. The manufacturing fixed costs and variable costs are incurred in each production period. Our goal is to develop a strategy for pricing, lead-time quotation and production scheduling so as to maximize the average profit earned by the firm per unit period. In the following section, we present an SMDP model for the problem.

3.3.1 SMDP Model

In order to model this joint decision problem as a SMDP, we first defined the system state-space. If there are n type of customer in the system, the system state (ψ) can be denoted by the vector

$$\psi = (m, \theta, \mu_1, \mu_2, \ldots, \mu_n, \sigma),$$

where m represents the type the most recent order type (among n possible types) was placed by a customer, θ is the order quantity of the most recent order, $\mu_1, \mu_2, \ldots, \mu_n$ represent the order quantity sold in n types of orders, and σ denotes the production capacity remaining

in this period. We assume the processing time for order i is $pt^i_{j,q} = d^i_{j,q}$, so σ also denote the production capacity remaining in this period. The cardinality of the state-space can be shown to have an upper bound of $n \times M^{(n+2)}$, where M is the maximum number of customers in any give type of order within this period, which is the production period capacity.

Clearly, a change in system state is caused by any of the following three events: (1) a new customer arrives and places an order; (2) the production is completed for a specific unit in an order and (3) the production period is ended. Whenever a customer places an order, a decision needs to be made whether to accept or reject the order. The completed order will not be delivered until its quoted lead-time. Hence, the time epochs of customer demand arrivals form the decision-making epochs.

Let ψ_m and σ_m denote the system state and time respectively at the m^{th} decision-making epoch. We define two stochastic processes: ψ=$\{\psi_m\colon\ m \in \mathcal{N}\}$ and σ=$\{\sigma_m\colon\ m \in \mathcal{N}\}$ where $\mathcal{N}$ is the set of natural numbers. We also defined a joint process (ψ, σ)=$\{\psi_m, \sigma_m\colon\ m \in \mathcal{N}\}$. Assuming that the demand arrival processes for the order type i are Poisson, it can be easily shown that

$$P[\psi_{\mathbf{m+1}} = j, \sigma_{\mathbf{m+1}} - \sigma_{\mathbf{m}} <= \sigma | \psi_{\mathbf{0}}, \ldots, \psi_{\mathbf{m}}; \sigma_{\mathbf{0}}, \ldots, \sigma_{\mathbf{m}}] = P[\psi_{\mathbf{m+1}} = j, \sigma_{\mathbf{m+1}} - \sigma_{\mathbf{m}} <= \sigma | \psi_{\mathbf{m}}, \sigma_{\mathbf{m}}]$$

which means that the process (ψ, σ) is a Markov renewal process. Then the decision process related to (ψ, σ) is a SMDP, and ψ is a Markov chain underlying the Markov renewal process. It is clear that for SMDPs, decision epochs are not restricted to discrete time epochs (e.g., in MDPs) but are all time epochs at which the system transits to a new decision-making

state. That is, the system state may change several times between two decision epochs. In our SMDP, the process that tracks every state change referred to as the natural process, and the process embedded at the decision-making epochs is referred to as decision process. Between two consecutive decision epochs, the system state can be changed for many times because it is possible that one unit product for an accepted order is completed or a specific production period is ended.

The action space (set) $\mathcal{A}$ can be defined as the set of all possible actions that can be chosen at any decision-making state (epoch). The actions to be taken can be described broadly as production ($\mathcal{C}$) and delivery ($\mathcal{D}$). The action set "production" corresponds to the action of whether accept or reject the arrival order while the action set "delivery" corresponds to the action of whether ship or hold the completed orders. $\mathcal{C}$={accept, reject} and $\mathcal{D}$={ship, hold}, Thus, the action space is $\mathcal{A}$=$\mathcal{C} \cup \mathcal{D}$. Under the action set of production, if action "accept" is selected, we will choose the price and lead-time from the specific price and lead-time sets. Figure 3.2 shows the hierarchy of the action space. At each decision-making epoch, the agent decides to take action a from an action set or action space $\mathcal{A}$. If we define the current and next state is **s** and **s'** (**s, s'** $\in \psi$), respectively, the current state-action pair is defined as $(\mathbf{s}, a)$.

3.3.2 Q-learning Algorithm (QLA)

As stated in Section 3.3.1, the problem can be modeled as a SMDP with the following characters: state space, action space and immediate rewards (Qiu and Loulou, 1995). For our SMDP, we use one of RL algorithms, Q-learning algorithm for SMDP.

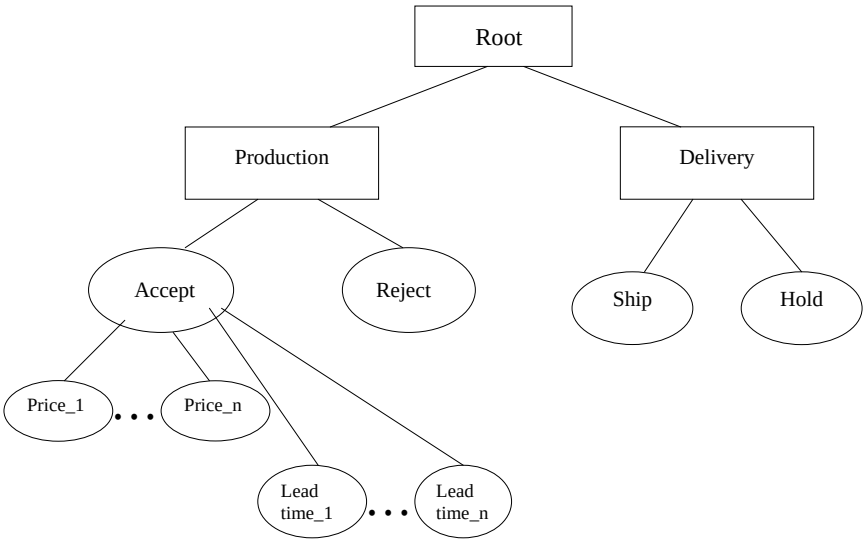

Figure 3.2: Hierarchical action space for the SMDP (The root node is the decision to be made for the main problem)

The following notation is used in describing our QLA algorithm.

m : current iteration(step) index

t_m : simulation time at the m^{th} decision-making epoch

$maxtime$: maximum time of iterations

$\mathcal{A}$: action space

a: an action in action space $\mathcal{A}$

$Q_m(\mathbf{s}, a)$: action value (Q function value) for state-action pair $(\mathbf{s}, a)$ at the

m^{th} decision-making epoch

a_m : the action taken at the m^{th} decision-making epoch

c_m : cumulative reward at the m^{th} decision-making epoch

ρ_m : average reward at the m^{th} decision-making epoch

α_m : the learning rate at the m^{th} decision-making epoch

p_m : probability of exploratory at the m^{th} decision-making epoch

$\tau(\mathbf{s}, \mathbf{s'}, a)$: the transition time from current state **s** to next state **s'** due to action a

$r(\mathbf{s}, \mathbf{s'}, a)$: the immediate rewards earned as a result of taking action a in current state **s** and leading to next state **s'**

Algorithm 2: QLA

1 $m = 0, Q_m(\mathbf{s}, a) = 0$, Choose the initial state arbitrarily.
  $t_m = 0, c_m = 0, \rho_m = 0, p_m = \alpha_m = 0.1$.
2 **while** $t_m <$ *maxtime* **do**
3 With probability $(1 - p_m)$, choose an action a^{max} that maximize $Q_m(\mathbf{s}, a)$; otherwise randomly choose an action from $\mathcal{A}$. Denote the chosen action as $\tilde{a}$
4 **if** *the order can be insert into the current schedule* **then**
5 $a_m \leftarrow \tilde{a}$
6 **else**
7 **while** *the order can not be insert into the current schedule* **do**
8 Randomly choose an action from $\{\mathcal{A} \backslash \tilde{a}\}$. Denote the chosen action as $\tilde{a}$
9 **end**
10 $a_m \leftarrow \tilde{a}$
11 **end**
12 Let the current and next state be **s** and **s'**, respectively.
13 $Q_{m+1}(\mathbf{s}, a_m) \leftarrow$
  $(1 - \alpha_m) Q_m(\mathbf{s}, a_m) + \alpha_m \big(r(\mathbf{s}, \mathbf{s'}, a_m) - \rho_m \tau_m(\mathbf{s}, \mathbf{s'}, a_m) + \max_{b \in \mathcal{A}} Q_m(\mathbf{s'}, b) \big)$
14 **if** $a_m = a^{max}$ **then**
15 $t_{m+1} \leftarrow t_m + \tau(\mathbf{s}, \mathbf{s'}, a_m)$
16 $c_{m+1} \leftarrow c_m + r(\mathbf{s}, \mathbf{s'}, a_m)$
17 $\rho_{m+1} \leftarrow c_{m+1} / t_{m+1}$
18 **else**
19 $t_{m+1} \leftarrow t_m$
20 $c_{m+1} \leftarrow c_m$
21 $\rho_{m+1} \leftarrow \rho_m$
22 **end**
23 $\alpha_m = \frac{\alpha_0}{1 + (m/\chi)}$
24 $p_m = \frac{p_0}{1 + (m/\chi)}$
25 $m \leftarrow m + 1$
26 **end**

At each epoch m, a learning rate is used, which is denoted by α_m. In addition, we use the exploratory rate p_m which decreases with the simulation iterations in order to guarantee its convergence. As in line 23 and 24, α_m and p_m are decayed according the DCM scheme (Darken and Moody, 1992). The DCM scheme is also called *search-then-converge* schedule. In DCM scheme,

$$\alpha_m = \frac{\alpha_0}{1 + (m/\chi)}$$

$$p_m = \frac{p_0}{1 + (m/\chi)}$$

where α_0, p_0 and χ are constants, which are 0.1, 0.1 and 10^{15}, respectively. In the early stages of adaptation, the value of learning rate and exploratory rate are approximately same as the constant α_0 and p_0 as the algorithm mainly explores. As the number of time steps approaches the constant χ, the algorithm converges. For a number of steps m sufficiently large compared to the search time constant χ, the learning rate operates as a traditional stochastic approximation algorithm.

In the QLA (Algorithm 2), step 1 (line 1) initializes the simulation parameters and chooses the initial state randomly. Then, at the iteration m, lines 2 to 3 choose an action a^{min} that minimizes the value of $Q_m(\mathbf{s}, a)$ with a probability $(1-p_m)$; otherwise a random (exploratory) action from $\mathcal{A}$ with equal probabilities is chosen. RL involves a conflict between exploitation and exploration. When deciding which action to take, the RL agent has to trade-off two conflicting goals: it has to exploit what it has already learned in order to have a high reward, and it has to behave in new ways, which are to explore and to learn more. In order to balance these two objectives, there is an exploratory rate in RL algorithms. We use the exploratory rate p_m which decreases with the simulation iterations to guarantee the convergence of QLA as in line 24. Step 2 (lines 4 to 11) checks the feasibility to make sure if the order can be insert into the current schedule with the chosen action. The order can be inserted into the current schedule if all the accepted orders can be completed within their latest acceptable

due-dates. If it is feasible, then it will be executed; otherwise, another one from the action set will be chosen randomly.

Then, line 13 updates the reward value $Q_{m+1}(\mathbf{s}, a)$ for the pair of $(\mathbf{s}, a)$. $\tau(\mathbf{s}, \mathbf{s'}, a)$ is the transit time between two states, that is recorded and stored in the simulation model. For example, if the chosen action is "produce order i", it records the production start time and completion time as the simulation is running. Therefore, the transition time is the processing time, which is updated at each decision-making epoch. Regarding the immediate reward $r(\mathbf{s}, \mathbf{s'}, a)$, the simulation model calculates and stores the total rewards at each decision-making epoch and the immediate reward is the difference in total costs between two consecutive decision-making epochs.

If a non-exploratory (nonrandom) action is chosen, lines 14 to 17 update total time t_m, update total reward c_m and update average reward ρ_m as in lines 14 to 17. Otherwise, as in lines 19 to 21 the total time t_m, update total reward c_m and update average reward ρ_m remain the same values as they are in the $m-1$ decision-making epoch. The last step (lines 23 to 26) sets current state $\mathbf{s}$ to new state $\mathbf{s'}$ and updates the decision epoch number m, learning rates α_m and β_m, and probability of exploratory p_m. The simulation continues until the termination criterion (line 2) is met.

3.3.3 Function approximation scheme

Traditionally, the lookup tables are used to represent Q-values as described. However, it can only be feasible for small-scale problems. For our problem, the state-action space is too large-scale and complex to tackle via lookup tables, and that some sort of function

approximation scheme will be necessary. This section examines the use of neural networks (NN) to represent the Q-functions in the same economic models studied previously.

Artificial Neural Networks (ANN), usually called NN, is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. NN provides a general and practical method for learning real-values and discrete-values functions from examples. In order to approximate the Q-values over the state-action space, we use a function approximation scheme by which a large number of Q-values are stored in the form a small number of scalers (weights of the neural network). A primitive class of NN consisting of a single neuron operating under the assumption of linearity is presented next. This class of network is known as the Least Mean Square (LMS) algorithm. A linear neuron is a two-layered NN where the input layer contains the input nodes and the output layer contains the output node. Our input nodes denote the variables used to define the state-action space and the output node is the Q-value. The input is normalized between zero and one. When an action is executed and the state transits to the new decision-making epoch, the Q-values for the action taken in the old decision-making epoch has to be updated. The net stores the Q-value for each state-action pair in form of weights. Consequently, the Q-value is updated by updating the weights that store this particular Q-value. The rule (algorithm) we used to update the weights is the delta-rule or the *Widrow-Hoff* rule. The algorithm is based on the use of instantaneous estimates of the environment (simulation) response to the learning agent. Whenever an action is chosen, Q-values for all possible actions are obtained from the network and an action is chosen based its Q-value. As the system transits to the new state due to the chosen action, the new Q-value for the most

recent state-action pair is calculated from the QLA algorithm. The new Q-values serve as the target in the updating algorithm and update the weights of the network. The notations and updating algorithm is given below:

y_m : the actual output unit (the old Q-value) at the m^{th} decision-making epoch

$\hat{y}_m$: the target output unit (the new Q-value) at the m^{th} decision-making epoch

δ_m : the error between the actual output and the target output at the m^{th} decision-making epoch

w_m^i : estimates of the input unit weights at the m^{th} decision-making epoch

η : the learning rate

x_m^i : the i^{th} input unit at the m^{th} decision-making epoch

p : the number of input units that will undergo updating during the learning process (equivalent to the number of partitions for the problem

m_{max} : the maximum simulation epoch

Algorithm 3: Function approximation algorithm

```
1  for i ← 1 to p do
2  |  w_1^i = 0.0001
3  end
4  for m ← 1 to m_max do
5  |  y_m = Σ_{i=1}^{p} w_m^i x_m^i
6  |  δ_m = ŷ_m − y_m
7  |  for i ← 1 to p do
8  |  |  w_{m+1}^i = w_m^i + ηδx_m^i
9  |  end
10 end
```

In the function approximation algorithm (Algorithm 3), step 1 (from line 1 to 3) initializes the values of input unit weights w_m^i to 0.001. Step 2 is from line 4 to line 10. Line 5 calculates the network output (old Q-value) y_m. Line 6 calculate the error between the target value $\hat{y}_m$ and the actual output y_m. The target value is the new Q-value, which is obtained from RL. Line 7 to 9 update each network weight w^i. In the numerical problems that we studied, η was decayed according the DCM scheme.

3.4 Numerical Experiments

In the numerical experiments, there are three types of customer orders for a MTO manufacturer. Each type of order can be quote three levels of prices and three lengths of lead-times. Price set and lead-time are generated using Uniform[60,100] and Uniform[5, 15]. Each order has a specific latest acceptable due-date which is generated from Uniform[20,30]. Customer order quantity depends on the quoted lead-time and price, which is a linear function. The available production capacity in each production period is 80 hours. Type i customer orders

arrive following a Poisson process with parameter λ_i. The parameter of baseline case is shown in Table 3.1.

Table 3.1: Parameters for the baseline scenario

Demand Rate	Holding cost	Lateness penalty cost	Variable cost	Fixed cost	Lost opportunity cost
1/10	1	3	2		
1/20	2	5	1	2	20
1/30	3	2	3		

Different scenarios in MTO system have been evaluated to compare the QLA with the First-Come-First-Serve (FCFS) policy and the threshold heuristic developed based on the order acceptance threshold heuristic proposed by Hing et al. (2007). The evaluation of performance is based on the average reward and order acceptance rate. Order acceptance rate is the number of accepted orders divided by the total number of arrival orders. The performance of QLA, FCFS and threshold heuristic are tested in five scenario. The results are shown in Table 3.2. The detailed design and outcome of the five simulation scenarios are illustrated in the following sections.

Table 3.2: Comparison results of average reward and acceptance rate earned by QLA vs. Heuristic and FCFS for the five scenarios at 95% confidence interval

	QLA		Heuristic		FCFS	
Scenario	Average reward	Acceptance rate	Average reward	Acceptance rate	Average reward	Acceptance rate
1	61.37±1.81	0.31±0.02	54.36±0.18	0.42 ±0.01	52.92±0.23	0.42± 0.01
2	63.25±1.17	0.53±0.03	52.88±0.26	0.78±0.02	51.79±0.36	0.79±0.03
3	130.54±2.12	0.53±0.03	106.17±0.52	0.72±0.02	102.69±0.37	0.78±0.01
4	58.39±1.19	0.43±0.01	40.59±0.48	0.50±0.02	38.79±0.6	0.51±0.01
5	72.63±1.05	0.52±0.02	61.93±0.32	0.69±0.05	56.42±0.25	0.70 ±0.02

3.4.1 Scenario 1: Baseline

In Table 3.1, the input parameter values for this baseline scenario are given. In Figure 3.3, the learning curve for this baseline scenario is shown for the QLA policy using the average reward obtained from the accepted orders. The learning curves are obtained by averaging ten independent simulation runs. The comparison is made among the QLA, FCFS and the modified threshold heuristic $b \geq 6$. Parameter b is defined as the revenue per requested unit of capacity for an accepted order. In the Heuristic, we accept the order with b larger than 6 and give the priority to the order with the bigger b value.

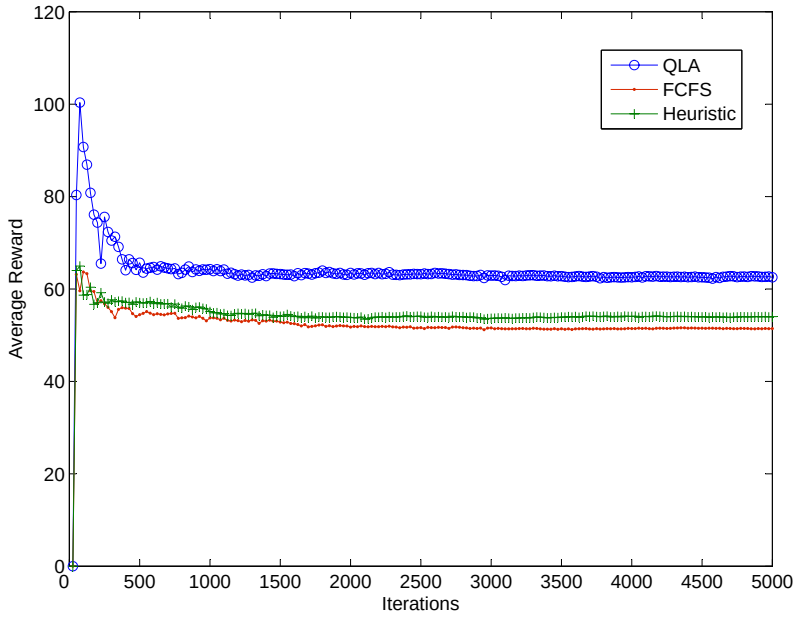

Figure 3.3: Average reward leaning curves for baseline scenario (QLA compared to FCFS and a threshold heuristic with b=6)

As shown in Figure 3.3, it is clear that the QLA outperforms both the Heuristic and FCFS policies on profitability. However, it is quite interesting that in the learning phase, the average reward are much greater than the average reward obtained after the learning

phase. This phenomenon can be explained by the high lost opportunity cost and limited production capacity. At the the beginning of simulation, almost every arrival order can be accepted. But after several iterations, the learning agent has to reject some orders due to the limited production capacity. According to the results in Table 3.2, the QLA is very selective on orders because it has the lowest order acceptance rate. Although the QLA has a very low order acceptable rate, it generates the highest average reward compared with the Heuristic and FCFS policies, which indicates the superior performance of the QLA in order selection, pricing and lead-time decisions.

3.4.2 Scenario 2: Demand variation

In order to test the adaptability of the proposed QLA, we make a variation on the customer demands. In the demand variation scenario, the demand rates of order type 1, 2 and 3 are reduced to 1/20, 1/40 and 1/60, respectively. The rest of the input parameters are same as these in baseline scenario.

Variations in demand rates are quickly detected by the QLA by accepting more orders. This situation triggers a resetting of the exploration strategy of the QLA and a new leaning curve is shown in Figure 3.4. After a learning phase, the QLA outperforms the Heuristic and FCFS, which obtains the highest average reward. However, compared to the baseline scenario, the average reward of the QLA increases while the average rewards of the Heuristic and FCFS reduce slightly. It is interesting that with the increase in demand, the average rewards gained from the Heuristic and FCFS decrease because many profitable orders are rejected. The QLA demonstrates the adaptability on the demand variation. It is worth

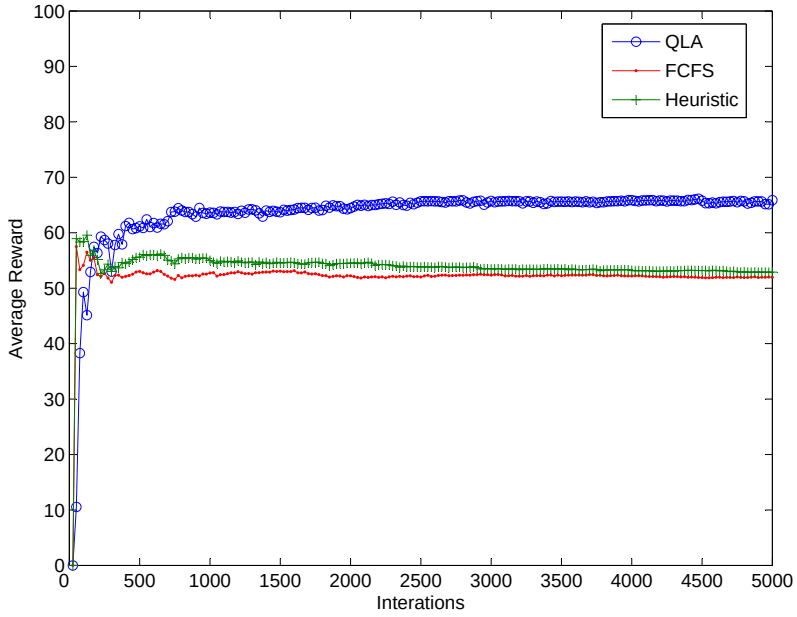

Figure 3.4: Average reward leaning curves for demand variation scenario (QLA compared to FCFS and a threshold heuristic with b=6)

noting that most of the arrival orders can be accepted when the demand rate are lower. In Table 3.2, the order acceptance rates for three policies are increased compared to the baseline scenario.

3.4.3 Scenario 3: Capacity variation

In each production period, the available production capacity is very critical for the MTO manufacture in light of profitability. The order selection, pricing and lead-time selection policy should be response to the capacity variation. For example, if the capacity increases, the policy needs to select more orders from the rejection set. In this capacity variation scenario, we increase the available production capacity to 160 hours. The rest of the input

parameters are same as these in baseline scenario. The learning curves are shown in Figure 3.5.

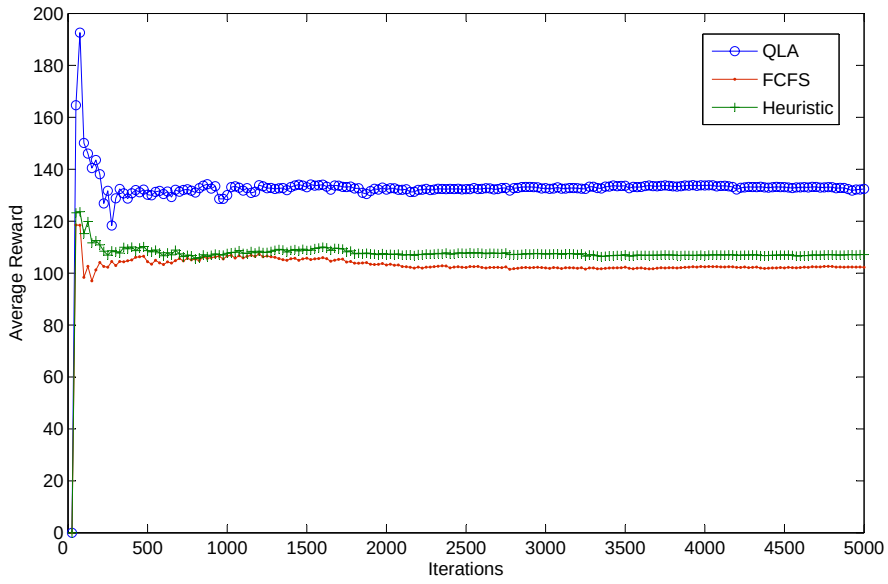

Figure 3.5: Average reward leaning curves for capacity variation scenario (QLA compared to FCFS and a threshold heuristic with b=6)

With the increase in capacity, the average rewards generated by three policies increase significantly. More specifically, the average rewards obtained by the QLA, Heuristic and FCFS increase by 111%, 95% and 94%, respectively. Therefore, the QLA gains the highest improvement in profit generation compared with the other two policies. It also indicates that the QLA is more sensitive to the increase in capacity and takes full advantage of the capacity improvement by its learning ability.

3.4.4 Scenario 4: Single price

Market conditions constantly change order profitability. If there is only a single price can be chosen for each type of order. The order acceptance and lead-time selection policy should be able to detect to adapt to these changes. According to the study of Charnsirisakskul et al. (2006) , the multiple price is more profitable than a single price for a order. We choose the lowest price level as the single price for each order. For instance,the different types of order can be quoted different price sets in baseline scenario while different order only can be quoted one price set. The rest of the input parameters are same as these in baseline scenario. The learning curves are shown in Figure 3.6.

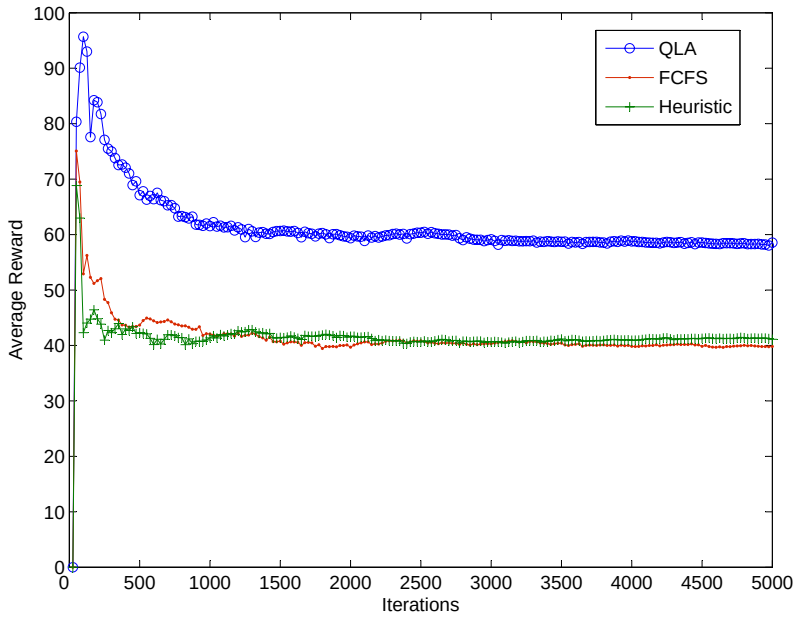

Figure 3.6: Average reward leaning curves for single price scenario (QLA compared to FCFS and a threshold heuristic with b=6)

As shown in Table 3.2, the change to single low price cause the average rewards reduce and the acceptance rates increase compared with the baseline scenario. Clearly, the QLA

consistently outperforms the other two policies with highest average reward and acceptance rate.

3.4.5 Scenario 5: Due-date negotiation

Due-date negotiation is modeled here as offering customer some compensation on order late delivery (i.e., lateness penalty cost for the manufacturer) if customers postpone their latest acceptable due-dates. In this scenario, the latest acceptable due-date for all orders are generated from Uniform[25, 35]. The rest of the input parameters are same as these in baseline scenario. The learning curves are shown in Figure 3.7.

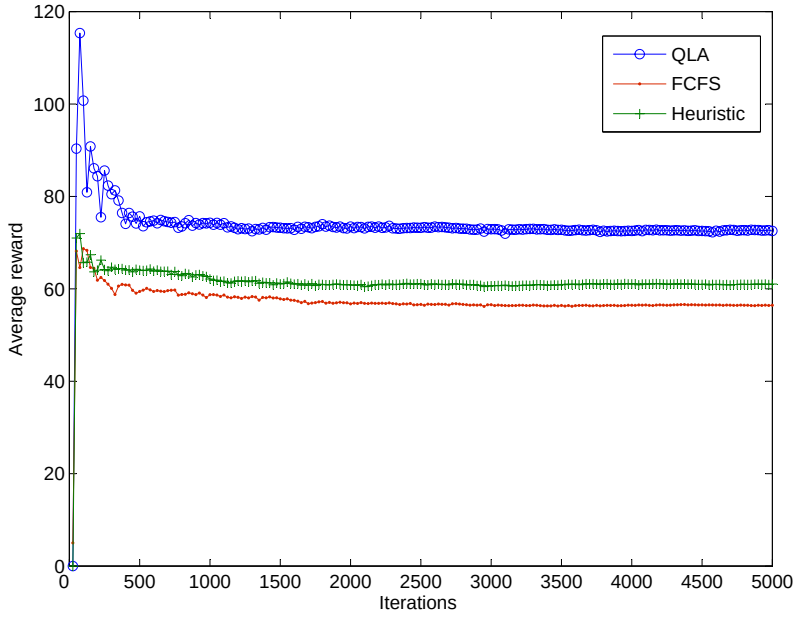

Figure 3.7: Average reward leaning curves for due-date negotiation scenario (QLA compared to FCFS and a threshold heuristic with b=6)

With the due-date negotiation, the average rewards and the order acceptance rates for three policies increase. More specifically, the average reward of the QLA, Heuristic and

FCFS increase by 18%, 14% and 7%, respectively. It is clear that the QLA obtains the highest improvement in profit compared with the other two policies. It indicates that the QLA makes better use of the advantages of the due-date negotiation and generate more profit than the other two policies.

3.5 Summary

In this chapter, we investigate the pricing, lead-time, scheduling and order acceptance decisions in a MTO manufacturing system with stochastic demands. The goal of this study is to maximize the long-run average profit (reward). We proposed a RL-based algorithm, the QLA, to solve this problem. To deal with many sources of uncertainty in a MTO manufacturing system, we set up five scenarios including the baseline, the demand variation, capacity improvement, single price selection and due-date negotiation. In all these five scenarios, the QLA outperforms the other two benchmarking policies, the Heuristic and FCFS. Although the QLA has the lowest order acceptance rate in all five scenarios, it can generate the highest average reward among three policies.

The importance of developing joint optimization on the pricing, lead-time, scheduling and order acceptance decisions under uncertainty using the QLA have been highlighted for MTO. It is obvious that the QLA not only is highly selective on order acceptance but also adapts quickly to the environmental changes. The QLA is very effective to quickly learn which order is worth accepting in order to accept more profitable orders within the limited

production capacity. Therefore, the QLA can be implemented in a MTO manufacturing firm for revenue management.

Chapter 4

Revenue management for a Make-to-Stock System

4.1 Introduction

We consider a problem that is motivated by a situation commonly found in make-to-stock (MTS) manufacturing. The system consists of a single machine in which the demands and the processing times for N types of products are random. The prices, sequence-independent setup times and costs are explicitly considered and may have different values for various types of products. If stock is on-hand, the demands can be fulfilled immediately. In shortage situations, orders can be backordered to be fulfilled from future arriving supply. The problem is to decide when, what, and how much (the lot size) to produce so that the long-run average profit. The total profit equals to the total revenue minus the total cost including setup, holding and backorder costs.

The problem is related to the stochastic economic lot scheduling problem (SELSP). The objective in SELSP, however, is to minimize the long-run average total cost. The SELSP is the stochastic version of economic lot scheduling problem (ELSP), which is NP-hard (Hsu, 1983). The typical ELSP is concerned with scheduling the production of several products on a single machine with the objective of minimizing the long-run average holding and setup costs. Despite the vast literature devoted to the ELSP (see the survey paper by (Elmaghraby, 1978; Raza et al., 2006)), the ELSP has not been solved in general. Not surprisingly, the SELSP appears to be analytically intractable when the demands arrive in a random fashion and the production rates also are random. The main challenge in the SELSP is how to dynamically schedule the products on a single machine in the face of unpredictable, dynamic and random demands so that the long-run average cost is minimized. In contrast to the ESLP, it is hard to get the predetermined cyclical production sequence and lot sizes, for the SELSP, due to the stochastic environment (e.g., stochastic demands and processing rates) since it is important to be responsive to the dynamic changes in this environment. In our cost structure, the presence of setup costs and times in combination with the stochastic environment adds to the difficulties of the problem. On the one hand, short cycle lengths are desirable because they provide frequent production opportunities for the various products to react to the stochastic environment. But, on the other hand, short cycle lengths increase the setup frequency, which not only reduce capacity available for production and hinders the timely fulfillment of demand, but also increases setup costs.

Despite its difficulty, this problem has been the subject of a recent flurry of activity. A range of quantitative techniques have been applied to the SELSP in the literature. Some

authors have adopted a deterministic analysis of the ELSP to construct simple control rules for the SELSP. For example, Leachman and Gascon (1988) developed a dynamic cycle lengths heuristics (DCL), for the problem with both setup costs and times. Their control policy is based on the solution of an ELSP, using the basic period formulation of the ELSP and assuming equal lot sizes for every run of a particular item to determine the target cycle times. Gallego (1990) employed a cyclic production sequence that was computed using the results of (Gallego and Roundy, 1992) for the deterministic ELSP with backorder costs. Bourland and Yano (1994) use a static continuous review control policy in a cyclical schedule and developed a two-level approach including a planning model and a control model. However, other authors have directly incorporated the stochastic elements of the problem and applied non-linear optimization, queueing analysis, or simulation to construct a control approach. Qiu and Loulou (1995) model the SELSP with Poisson demand and deterministic processing times as a semi-Markov decision process (SMDP) and use a successive approximation scheme over a truncated state space to compute near-optimal policies in the two-product case. Sox and Muckstadt (1997) address the stochastic lot-size model directly, developing heuristic solutions. Anupindi and Tayur (1998) consider a class of periodic base stock policies and used a simulation-based approach to obtain good base stock levels for a variety of performance measures. The strength of their approach was that it is more general than the control policies described previously. They consider a given cyclic production sequence and use an optimization-based search procedure to determine base stock levels for different cost and service objectives. Paternina-Arboleda and Das (2005) first apply the reinforcement learning (RL) algorithm named relaxed-SMART (Semi-Markov Average Reward Technique) to the

SELSP problem with stochastic demands, constant processing times and setup times. In their work, the relaxed-SMART algorithm outperformed the various cyclical policies.

Our problem is more challenging than the SELSP in that we consider not only the setup time, setup, holding and backorder costs but also the prices so that the objective is to maximize the long-run average total profit. There is only limited research available on the revenue management (RM) in MTS manufacturing. RM applications in manufacturing largely focus on make-to-order (MTO) production environments. This is intuitive due to the correspondence between production capacities and perishable assets, which are the difficulties in applying RM in the service industries. In the context of pricing and order acceptance control, MTS queues, which combine aspects of queueing and inventory models, have been studied. Li (1992) allows the customer arrival rate to depend on the congestion and exogenous price, and derives optimal busy/idle policies for competing manufacturers. Carr and Duenyas (2000) consider a two-class queue with a make-to-stock class and a MTO class, where the controller sequences the jobs and makes accept/reject decisions on the MTS class. A closely related paper is the paper written by Caldentey and Lawrence (2006) who consider a single-product MTS manufacturing system in which the manufacturer's control problem is to select the optimal long-term contract price as well as the optimal production (i.e., busy/idle) and electronic-order admission policies to maximize revenue minus inventory holding and backorder costs. All of these models assume that no set-ups (e.g. setup times and costs) are required to switch from one type of product to another. However, the problem of set-ups are prevalent in many industries because facilities that operate in a MTS mode typically produce standardized products that require set-ups. In most situation, the setup

time problem is more realistic than the setup cost problem, but is also less amenable to analysis. The setup cost problem, however, may be relevant for manufacturing systems that have internalized their setup times; that is, they incur significant material, labor, and/or capital costs to greatly reduce their switchover times (Markowitz et al., 1995). Therefore, in order to represent the real-world problem, we consider the setup times and costs. In order to make real-time decisions that react to the stochastic demands and processing times, we use a discrete control approach with the dynamic sequencing based on a combination of discrete event simulation and reinforcement learning (RL).

This chapter offers three contributions. Firstly, we develop a mathematical model to describe the problem for further investigation. Secondly, we propose two RL algorithms. One of our RL algorithms is a Q-learning algorithm for the Semi-Markov decision process (QLS), which is developed from Q-learning algorithm (Gosavi, 2004b). In order to improve the performance of QLS, we develop QLIH by integrating a heuristic with QLS. Thirdly, a simulation model in Arena® is developed to test the QLIH and QLS compared with the other two benchmarking policies, Brownian policy and FCFS policy in the study of (Wein, 1992).

The rest of the chapter is organized as follows. Section 4.2 develops a mathematical model for the problem. Section 4.3 addresses the control state-action space of the problem and proposes two RL algorithms. Section 4.4 presents two benchmarking policies, numerical examples and simulation results. Section 4.5 gives the summary and proposes future research.

4.2 Model Formulation

The problem in this study involves a single machine and N types of products. The setup times are separate from the processing times and independent of job sequences. We assume that raw materials are always sufficient and available for production (e.g., raw materials for product i can be obtained whenever the production of product i occurs) and we do not consider the procurement and holding costs for the raw materials. Such an assumption is common in the literature (Paternina-Arboleda and Das, 2001; Wein, 1992); for example, Wein (1992) assumes there are enough raw materials for production in a multiclass queue. Each product may have a unique demand pattern and processing rate distribution. Demands for various products are stationary and mutually independent stochastic processes. Let $\mathbf{D(t)} = [D_i(t)]$, for $i = 1, \cdots, N$. The demand for product i follows the renewal process $D_i(t)$, which is the number of demands for type i up to time t. The demands that cannot be satisfied directly from stock are backordered and will be filled whenever the products are ready. The processing times for product i are expressed by the renewal process $P_i(t)$, which is the number of finished products i completed up to time t assuming the machine is continuously producing product i in the interval $[0, t]$.

We present the decisions in our model in the form of cumulative control process. The system state is a vector of the inventory position of each product. The inventory position could be positive, implying that the on-hand inventory is available, or it could be negative, implying there is no inventory and its absolute value is at the level of backorder. Let $L_i(t)$ denote the inventory position of type i product in inventory at time t. The vector

$\mathbf{L(t)} = [L_i(t)]$ for $i = 1, \cdots, N$ denotes the system state at time t, representing the dynamics of the inventory process. The inventory position for product i, $L_i(t)$ cannot exceed the basestock level L_i^{max}, that is,

$$L_i(t) \leq L_i^{max} \tag{4.1}$$

where L_i^{max} represents basestock inventory level. If the inventory level $L_i(t)$ reaches the basestock level L_i^{max}, the production for product i is stopped. If $L_i(t)$ reaches zero, future demands are backordered. The agent must make a scheduling decision whenever the system state changes either because of production completion or demand arrival.

The allocation process $T_i(t)$ is the cumulative amount of time that the machine produces type i product in $[0, t]$. The vector $\mathbf{T(t)} = [T_i(t)]$ denotes a scheduling policy which is the allocation of the time over $[0, t]$ on processing each type of product. As in the study of Harrison (1988), a scheduling policy $\mathbf{T(t)}$ has two properties: (1) $\mathbf{T(t)}$ is nondecreasing and continuous. (2) $\mathbf{T(t)}$ is nonanticipating with respect to $\mathbf{L}$, which implies that the agent cannot predict future demands or processing times. Assuming $\mathbf{T}(0) = \mathbf{0}, \mathbf{D}(0) = \mathbf{0}$ and $\mathbf{L}(0) = \mathbf{0}$. For $i = 1, \cdots, N$ and $t \geq 0$,

$$L_i(t) = P_i(T_i(t)) - D_i(t) \tag{4.2}$$

Let $Q_i(t)$ denote the number of setups for processing product i during the time interval $[0, t]$. $Q_i(t)$ is determined by the scheduling policy $\mathbf{T}$. Considering the machine idle times

and setup times, we have the following constraint:

$$\sum_{i=1}^{N} T_i(t) + \sum_{i=1}^{N} Q_i(t) st_i \leq t \tag{4.3}$$

where st_i denotes the setup time for product i. In our model, the decision variables are $L_i(t)$, $T_i(t)$ and $Q_i(t)$ for $i = 1, \cdots, N$.

The objective of the problem is to minimize long-run average profit, i.e., the total profit minus the sum of setup, holding and backorder costs, per unit time over a time horizon. Let ϕ denote the planning time horizon. Let r_i denotes the unit price of product i. Let h_i and b_i denote the unit holding and backorder cost of product i per unit time, respectively, for $i = 1, \cdots, N$. Let st_i and sc_i denote the sequence-independent setup time and the corresponding cost, respectively, that are incurred when the production switches from products i' to i for $i' \neq i$ and $i = 1, \cdots, N$. The problem is to find a policy $\mathbf{T}$ to minimize the long-run average profit. Recalling that if $L_i(t) < 0$, we have the backorder cost $|bL_i(t)|$ and if $L_i(t) > 0$, we have the holding cost $hL_i(t)$; thus the problem can be formulated as

$$\max \lim_{\phi \to \infty} \frac{1}{\phi} \int_0^{\phi} \sum_{i=1}^{N} \big(r_i D_i(t) - h_i \max\{0, L_i(t)\} + b_i \min\{0, L_i(t)\} - sc_i Q_i(t) \big) dt \tag{4.4}$$

subject to constraints (4.1) to (4.3).

4.3 The RL Algorithms

In our RL algorithms, the environmental model includes the N types of products that need to be processed, associated with a basestock inventory level and a maximal backorder level, so as to perform the simulation in a limited state space. The learning agent is the single machine that decides which type of product to produce whenever the machine is available. The response from the environment is the immediate rewards (profits) incurred by executing the chosen action from an action set. In order to design the RL algorithms, we first define the control state-action space in subsection 4.3.1. Then, subsections 4.3.2 and 4.3.3, we present the proposed RL algorithms, QLS and QLIH.

4.3.1 The control state-action space

In order to simulate our model formulated in Section 4.2, we discretize the continuous time system to decision-making epoch. At each decision-making epoch (i.e., simulation iterations), the agent decides to take action a from an action set or action space $\mathcal{A}$ ={no production, produce product 1, $\cdots$, produce product N}. The control state-action space is defined as a set $\mathcal{LA} = \{(\mathbf{L}, a)\}$ $(a \in \mathcal{A})$, in which each element is a combination of inventory levels of N types of products and the current action. We define the minimum inventory level or maximum backorder level as L_i^{min}. Since the inventory level $L_i(t)$, is allowed to be in the range of $[L_i^{min}, L_i^{max}]$ in the simulation, there are $(N+1)\prod_{i=1}^{N}(L_i^{max} - L_i^{min} + 1)$ vectors in the state-action space. For example, suppose that there are three types of products, and the target basestock levels and maximum backorder levels are $(50, 50, 50)$ and $(49, 49, 49)$,

respectively. Then, the total number of vectors in the control state-action space is $(3+1)\prod_{i=1}^{3}(50-(-49)+1)= 4\cdot(50+49+1)^3 = 4\times 10^6$.

4.3.2 Q-learning Algorithm for Semi-Markov Decision Processes (QLS)

Many practical sequential decision making problems have underlying probability structures that cannot simply be characterized by Markov chains. For SMDPs, decision epochs are not restricted to discrete time epochs (e.g., in MDPs) but are all time epochs at which the system transits to a new decision-making state (Das et al., 1999). That is, the system state may change several times between two consecutive decision epoch. In our problem, between two consecutive decision epoch (e.g., two decision points upon the completion of the production), the system state can be changed many times as the new demands arrive. Therefore, the problem can be modeled as an SMDP. For our SMDP, we use one of RL algorithms, the Q-learning algorithm for SMDP.

The following notation is used in describing our QLS in Algorithm 4:

m : current decision epoch index

t_m : simulation time at the m^{th} decision-making epoch

$maxtime$: maximum time of an iteration

$\mathcal{A}$: action space {produce product 1, $\cdots$, produce product N, no production}

a: an action in action space $\mathcal{A}$

$Q_m(\mathbf{L}, a)$: action value (Q function value) for state-action pair $(\mathbf{L}, a)$ at the m^{th} decision-making epoch

a_m : the action taken at the m^{th} decision-making epoch

c_m : cumulative reward at the m^{th} decision-making epoch

ρ_m : average reward at the m^{th} decision-making epoch

α_m : the first learning rate at the m^{th} decision-making epoch

β_m : the second learning rate at the m^{th} decision-making epoch

p_m : probability of exploratory at the m^{th} decision-making epoch

L_i^{min} : the lower bound for inventory of product i (its absolute value is the maximal backorder level

L_i^{max} : the upper bound for inventory of product i (i.e., the basestock level)

$\tau(\mathbf{L}, \mathbf{L'}, a)$: the transition time from the current state $\mathbf{L}$ to the next state $\mathbf{L}'$ due to action a

$r(\mathbf{L}, \mathbf{L'}, a)$: the immediate reward earned as a result of taking action a in current state $\mathbf{L}$ and leading to next state $\mathbf{L'}$

Algorithm 4: QLS

1 $m = 0, Q_m(\mathbf{L}, a) = 0, \forall \mathbf{L} \in \mathcal{LA}$. Choose the initial state (inventory levels of N types of products) arbitrarily. $t_m = 0, c_m = 0, \rho_m = 0, p_m = \alpha_m = \beta_m = 0.1$.
2 **while** $t_m < maxtime$ **do**
3 With probability $(1 - p_m)$, choose an action a^{max} that maximize $Q_m(\mathbf{L}, a)$; otherwise randomly choose an action from $\mathcal{A}$. Denote the chosen action as $\tilde{a}$
4 **if** $L_i^{min} \leq L_i \leq L_i^{max}, \forall i$ **then**
5 $a_m \leftarrow \tilde{a}$
6 **else**
7 **while** $L_i < L_i^{min}$ *or* $L_i > L_i^{max}, \forall i$ **do**
8 Randomly choose an action from $\{\mathcal{A} \backslash \tilde{a}\}$. Denote the chosen action as $\tilde{a}$
9 **end**
10 $a_m \leftarrow \tilde{a}$
11 **end**
12 Let the current and next state be **L** and **L'**, respectively.
13 $Q_{m+1}(\mathbf{L}, a_m) \leftarrow$
$(1 - \alpha_m) Q_m(\mathbf{L}, a_m) + \alpha_m \big(r(\mathbf{L}, \mathbf{L'}, a_m) - \rho_m \tau_m(\mathbf{L}, \mathbf{L'}, a_m) + \max_{b \in \mathcal{A}} Q_m(\mathbf{L'}, b) \big)$
14 **if** $a_m = a^{max}$ **then**
15 $t_{m+1} \leftarrow t_m + \tau(\mathbf{L}, \mathbf{L'}, a_m)$
16 $c_{m+1} \leftarrow c_m + r(\mathbf{L}, \mathbf{L'}, a_m)$
17 $\rho_{m+1} \leftarrow (1 - \beta_m)\rho_m + \beta_m \frac{t_m \rho_m + r(\mathbf{L}, \mathbf{L'}, a_m)}{t_{m+1}}$
18 **else**
19 $t_{m+1} \leftarrow t_m$
20 $c_{m+1} \leftarrow c_m$
21 $\rho_{m+1} \leftarrow \rho_m$
22 **end**
23 $\alpha_{m+1} \leftarrow \alpha_m / t_{m+1}$
24 $\beta_{m+1} \leftarrow \beta_m / (m + 1)$
25 $p_{m+1} \leftarrow p_m / (m + 1)$
26 $m \leftarrow m + 1$
27 **end**

The learning rate α_m at any decision epoch for each state-action pair depends on $1/t_m$, which is the reciprocal of the number of times the state-action pair tried before that decision-making epoch, as seen in line 19 in Algorithm 4. As seen in line 24, β_m depends on $1/m$, which is the reciprocal of the number of decision-making epochs according to (Gosavi, 2004b).

In the QLS, step 1 (line 1) initializes the simulation parameters and chooses the initial state randomly. Then, at the iteration m, lines 2 to 3 choose an action a^{max} that maximizes the value of $Q_m(\mathbf{L}, a)$ with a probability $(1 - p_m)$; otherwise a random (exploratory) action from $\mathcal{A}$ with equal probabilities is chosen. RL involves a conflict between exploitation and exploration. When deciding which action to take, the RL agent has to trade-off two conflicting goals: it has to exploit what it has already learned in order to have a high reward, and it has to behave in new ways, which are to explore and to learn more. In order to balance these two objectives, there is an exploratory rate in RL algorithms. We use the exploratory rate p_m which decreases with the simulation iterations to guarantee the convergence of QLS as in line 21. Step 2 (lines 4 to 11) checks the feasibility of the chosen action. If it is feasible, then it will be executed; otherwise, another one from the action set will be chosen randomly.

Then, line 13 updates the reward value $Q_{m+1}(\mathbf{L}, a)$ for the pair of $(\mathbf{L}, a)$. $\tau(\mathbf{L}, \mathbf{L'}, a)$ is the transit time between two states, that is recorded and stored in the simulation model. For example, if the chosen action is "produce product i", it records the production start time and completion time as the simulation is running. Therefore, the transition time is the processing time, which is updated at each decision-making epoch. Regarding the immediate reward $r(\mathbf{L}, \mathbf{L'}, a)$, the simulation model calculates and stores the total rewards at each decision-making epoch and the immediate rewards is the difference in total rewards between two consecutive decision-making epochs.

If a non-exploratory (nonrandom) action is chosen, lines 14 to 17 update total time t_m, update total reward c_m and update average reward ρ_m as in lines 14 to 17. Otherwise, as in lines 19 to 21 the total time t_m, update total reward c_m and update average reward ρ_m remain

the same values as they are in the $m-1$ decision-making epoch. In the SMDP version of the algorithm, the update equation for reward value $Q_{m+1}(\mathbf{L}, a)$ and average reward value ρ_{m+1} has a Robbins-Monro version that also makes use of the renewal reward theorem (Gosavi, 2004b). The last step (lines 23 to 26) sets current state **L** to new state **L'** and updates the decision epoch number m, learning rates α_m and β_m, and probability of exploratory p_m. The simulation continues until the termination criterion (line 2) is met.

4.3.3 Q-learning with Learning-Improvement Heuristic (QLIH)

In the QLS, the learning agent has no knowledge on the action choice rule at the beginning of learning because the reward value matrix that stores the reward values is a zero matrix; thus in the QLIH, we train its knowledge and consequently improve the learning process through a heuristic. In the other words, we combine the heuristic with QLS.

The heuristic function is used in the action selection rule, defining which action a must be executed when the agent is in state **L** according to the heuristic. The action selection rule used is a modification of the standard $\epsilon - greedy$ rule used in Q-learning. The heuristic function is defined as follows:

$$\pi(\mathbf{L}) = \begin{cases} \arg\max_a [Q(\mathbf{L}, a) + \xi H(\mathbf{L}, a)] & \text{if } p < 1 - p_m \\ a_{random} & \text{otherwise} \end{cases} \tag{4.5}$$

where $\pi(\mathbf{L})$ is the policy used in state **L**, ξ is a real variable (e.g., 0.1) used to weight the influence of the heuristic, p is a random number generated from a distribution with uniform probability in (0,1), and a_{random} is a random action selected among the other possible actions.

The value of the heuristic function $H(\mathbf{L}, a)$ must be higher than the variation among the $Q(\mathbf{L}, a)$ so that it can influence the choice of actions. According to (Bianchi et al., 2004), it is defined as follows:

$$H(\mathbf{L}, a) = \begin{cases} \max_a Q(\mathbf{L}, a) - Q(\mathbf{L}, a_m) + \eta & \text{if } a_m = \pi_H(\mathbf{L}) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where η is a small real value (e.g., 0.01) and $\pi_H(\mathbf{L})$ is the action suggested by the heuristic at state $\mathbf{L}$.

For example, if the agent can execute four different actions when in state $\mathbf{L}$, the values of $Q(\mathbf{L}, a)$ for the actions are (1.1 1.0 0.9 1.3), and the action that the heuristic suggests is the second one. If η=0.01, the values to be used are $H(\mathbf{L}, 2)$=0.31, and zero for the other actions.

Algorithm 5 presents the QLIH algorithm, which makes use of a heuristic function in line 13 to improve the Q-learning process.

Algorithm 5: QLIH

1 $m = 0, Q_m(\mathbf{L}, a) = 0, \forall \mathbf{L} \in \mathcal{LA}$. Choose the initial state (inventory levels of N types of products) arbitrarily. $t_m = 0, c_m = 0, \rho_m = 0, p_m = \alpha_m = \beta_m = 0.1$.
2 **while** $t_m <$ *maxtime* **do**
3 With probability $(1 - p_m)$, choose an action a^{max} that maximize $Q_0(\mathbf{L}, a) + \xi H_0(\mathbf{L}, a)$; otherwise randomly choose an action from $\mathcal{A}$. Denote the chosen action as $\tilde{a}$
4 **if** $L_i^{min} \leq L_i \leq L_i^{max}, \forall i$ **then**
5 $a_m \leftarrow \tilde{a}$
6 **else**
7 **while** $L_i < L_i^{min}$ *or* $L_i > L_i^{max}, \forall i$ **do**
8 Randomly choose an action from $\{\mathcal{A} \backslash \tilde{a}\}$. Denote the chosen action as $\tilde{a}$
9 **end**
10 $a_m \leftarrow \tilde{a}$
11 **end**
12 Let the current and next state be **L** and **L'**, respectively.
13 Call heuristic
14 Update the value of $H_{m+1}(\mathbf{L}, a), \forall a \in \mathcal{A}$ using equation (4.6)
15 $Q_{m+1}(\mathbf{L}, a_m) \leftarrow$ $(1 - \alpha_m) Q_m(\mathbf{L}, a_m) + \alpha_m \big(r(\mathbf{L}, \mathbf{L'}, a_m) - \rho_m \tau_m(\mathbf{L}, \mathbf{L'}, a_m) + \max_{b \in \mathcal{A}} Q_m(\mathbf{L'}, b) \big)$
16 **if** $a_m = a^{max}$ **then**
17 $t_{m+1} \leftarrow t_m + \tau(\mathbf{L}, \mathbf{L'}, a_m)$
18 $c_{m+1} \leftarrow c_m + r(\mathbf{L}, \mathbf{L'}, a_m)$
19 $\rho_{m+1} \leftarrow (1 - \beta_m)\rho_m + \beta_m \frac{t_m \rho_m + r(\mathbf{L}, \mathbf{L'}, a_m)}{t_{m+1}}$
20 **else**
21 $t_{m+1} \leftarrow t_m$
22 $c_{m+1} \leftarrow c_m$
23 $\rho_{m+1} \leftarrow \rho_m$
24 **end**
25 $\alpha_{m+1} \leftarrow \alpha_m / t_{m+1}$
26 $\beta_{m+1} \leftarrow \beta_m / (m + 1)$
27 $p_{m+1} \leftarrow p_m / (m + 1)$
28 $m \leftarrow m + 1$
29 **end**

In the heuristic in line 13 of Algorithm 5, we consider the set of products that are currently in danger of being backordered (e.g., $L_i < 1, i = 1, \cdots, N$) and gives production priority to the largest value of index $b_i \mu_i - k_i s c_i$, where μ_i is the processing rate of product i, and k_i is 0

if the last action is producing i (i.e., no setup if choosing to produce product i); otherwise, 1. If no product is in danger of backorder, we will choose no production at this decision-making epoch. Our policy gives priority to the products (among the ones that are in danger of being backordered) with the maximum value of the index $b_i\mu_i - k_i sc_i$: these products are quick to process and expensive to backorder while reducing setup costs (frequency).

QLIH uses the same α, β and p values as the QLS. According to Bianchi et al. (2004), the values of ξ and η are set to be 0.1 and 0.01, respectively. The proposed QLIH is different from QLS only in the method of carrying out the exploration. The heuristic function value has been considered in making the action choice in the decision-making epoch while QLS only considers the value of $Q_m(\mathbf{L}, a)$. QLIH inherits some good characteristics of the Q-learning algorithm used in QLS, for example, the free choice of action from an action set and its convergence. The additional advantage is that it can also learn from the heuristic to reduce the arbitrariness in selecting action and thus reduce the search space of Q-learning. If the heuristic sometimes is not appropriate due to the stochastic environment, QLIH still converges, but slowly.

4.4 Simulation Experiments

The proposed RL algorithms were tested on an example with three different products processed on a single machine. We conducted a series of simulation experiments to evaluate the performances of QLS and QLIH and compared their performances with two existing policies, Brownian and FCFS policies. None of these four policies allow preemption of service. One

policy prioritizes jobs in an FCFS manner and the other control policy was developed from Brownian approximation (Wein, 1992) in which the scheduling problem is approximated by a dynamic control problem involving Brownian motion. The four policies QLIH, QLS, Brownian and FCFS are all modeled in ARENA® simulation software with the same experiment parameters and simulation condition.

4.4.1 Experiment Design

Table 4.1 shows the setting of seven factors for our simulation experiment, including price, setup time, production rate, holding cost, backorder cost, setup cost and demand rate. Based on the values in (Wein, 1992), we choose two levels for each of six factors. Each factor was a three-dimension vector which contained parameters for three different products. Then, we set up a seven-factor two-level factorial design, as shown in Table 4.1, requiring a total of 32 ($2^{(7-2)}$) data points. Three sets of cases were picked, as our illustrated examples shown in Table 4.2.

Table 4.1: Parameters for seven-factor two-level factorial design

	Price		Production rate	Costs(dollar/unit-time)		Setup cost	
Level	(dollar/unit)	Setup time	(units/time)	Holding	Backorder	(dollar/setup)	Demand rate
	50	0.01	1	1	1	1.6	0.3
1	60	0.02	1/2	1	1	1	0.15
	70	0.03	1/3	1	1	1.5	0.1
	70	0.1	2	2	2	4.8	0.15
2	80	0.2	1	2	2	3	0.075
	90	0.3	2/3	2	2	4.5	0.05

Wein (1992) did not consider prices, setup times and costs; thus, we assume constant prices, setup times and costs. The following are the assumptions in the test examples:

Table 4.2: Tested parameters for the illustrated numerical examples

Case	Price (dollar/unit)	Setup time	Production rate (units/time)	Costs(dollar/unit-time) Holding	Costs(dollar/unit-time) Backorder	Setup cost (dollar/setup)	Demand rate
6	50	0.1	1	2	2	4.8	0.15
	60	0.2	1/2	2	2	3	0.075
	70	0.3	1/3	2	2	4.5	0.05
12	50	0.01	2	2	2	4.8	0.15
	60	0.02	1	2	2	3	0.075
	70	0.03	2/3	2	2	4.5	0.05
16	70	0.1	2	1	2	1.6	0.15
	80	0.2	1	1	2	1	0.075
	90	0.3	2/3	1	21	1.5	0.05

. The machine is modeled as a multiproduct queue in which the machine operates continuously and can only process one product at a time. Preemption is not allowed.

. The release times of all products are zero and time/sequence-independent setup cost and time exist.

. Demand for each product is an arbitrary point process which is a Poisson process.

. The processing time for each product is modeled as an Exponential distribution.

. Each product has a constant price, setup time, setup cost, holding cost and backorder cost.

. Customers (orders) are homogeneous. For the same product type, service is given on a first-come-first-serve (FCFS) rule.

A search was performed to find the optimal combination of basestock levels for all products in the simulation model to get the optimal objective function because the decision agent cannot decide the basestock target inventory levels. This was obtained by using a commercial optimization tool (OptQuest®), embedded in the ARENA® simulation software. The

optimization objective function is described in OptQuest and the basestock levels of all products were set as control variables. Then, it searched for the values of controls that minimize the long-run average total profit.

4.4.2 Computational Results

The value for performance measure, which is the long-run average total profit was based on twenty replications. Each replication ran 20,000 time units where the average total profit converged. Each replication started with an arbitrary system state (inventory position) and ended when simulation time was reached. The total profit (consisting of revenue minus holding cost, backorder cost and setup cost) incurred per unit of time was observed for each replication. The mean value of average total profit and 95% confidence interval of this quantity were calculated for each case and policy. The simulation results are shown in Table 4.3.

The results show that the proposed QLIH outperforms QLS, Brownian and FCFS policies in all cases. Furthermore, QLS is also consistently better than Brownian and FCFS policy. An interesting finding is that Brownian policy is not an effective policy with the presence of setup cost and time. The optimality of Brownian policy has been proved by (Wein, 1992) and it performs effectively when the product is only associated with the holding cost and backorder cost. Under this condition, Brownian policy is better than the FCFS in minimize the total cost. However, with the existence of the setup cost and time, it is not always the case because Brownian policy may increase the setup and thus decrease the profit (e.g., case 6, 12 and 16 in Table 4.2). It is clear that Brownian policy is not a robust policy in any case.

Table 4.3: Comparison results of average profit incurred by QLIH vs. QLS, Brownian and FCFS for the three products at 95% confidence interval

	Average profit (95% C.I.) in dollars			
Case	QLIH	QLS	Brownian	FCFS
1	36.61(±3.02)	27.50 (±3.46)	22.1 (±1.32)	19.53 (±1.42)
2	38.85 (±3.65)	30.91 (±3.07)	19.32 (±1.25)	18.51 (±1.84)
3	28.66 (±2.99)	18.50 (±1.65)	15.21 (±1.03)	12.37 (±0.97)
4	55.42 (±3.82)	47.18 (±3.69)	38.59 (±1.77)	31.73 (±1.32)
5	49.23 (±3.12)	40.03 (±3.27)	25.90 (±1.35)	23.11 (±1.67)
6	19.33 (±2.86)	15.35 (±2.06)	10.7 (±1.01)	12.81 (±0.98)
7	44.40 (±3.95)	36.31 (±3.98)	28.63 (±1.80)	25.10 (±1.35)
8	40.90 (±3.03)	31.58 (±3.35)	18.05 (±1.97)	17.20 (±1.53)
9	44.52 (±3.53)	35.60 (±3.02)	21.12 (±1.86)	22.7 (±2.01)
10	30.33 (±2.21)	25.19 (±2.03)	19.27 (±0.98)	17.09 (±0.90)
11	61.03 (±2.06)	51.89 (±2.15)	34.37 (±1.16)	29.73 (±1.04)
12	63.26 (±4.51)	54.73 (±4.03)	40.11 (±2.05)	42.11 (±2.15)
13	33.70 (±2.05)	21.57 (±1.28)	17.20 (±0.95)	15.36 (±0.82)
14	34.41 (±2.15)	28.00 (±2.32)	23.09 (±1.05)	19.21 (±0.98)
15	27.93 (±1.23)	21.53 (±2.10)	12.90 (±0.79)	10.81 (±0.97)
16	69.50 (±4.20)	58.16 (±4.06)	40.29 (±1.22)	43.60 (±1.80)
17	58.32 (±3.97)	43.10 (±3.64)	30.50 (±1.83)	27.30 (±1.38)
18	19.99 (±1.84)	13.10 (±0.91)	10.12 (±0.79)	9.23 (±0.86)
19	76.65 (±3.87)	68.02 (±3.25)	47.22 (±2.57)	42.00 (±1.53)
20	43.99 (±2.78)	35.33 (±1.87)	29.21 (±1.15)	26.35 (±1.43)
21	82.26 (±4.38)	73.40 (±3.50)	58.31 (±2.89)	52.61 (±2.38)
22	18.02 (±1.10)	14.90 (±0.87)	9.20 (±0.53)	8.32 (±0.76)
23	27.12 (±2.57)	21.10 (±1.36)	15.05 (±0.86)	12.16(±0.22)
24	38.03 (±2.26)	32.19 (±1.97)	23.20 (±1.30)	21.49 (±1.27)
25	30.15 (±1.70)	22.60 (±1.83)	18.55 (±1.21)	15.30 (±0.79)
26	68.33 (±2.33)	59.40 (±2.50)	47.91 (±1.79)	44.20 (±1.87)
27	96.76 (±4.29)	85.24 (±3.56)	76.10 (±2.67)	72.63 (±2.96)
28	46.22 (±2.86)	37.18 (±1.84)	31.10 (±1.96)	27.76 (±1.45)
29	63.72 (±2.91)	55.30 (±3.01)	39.21 (±1.57)	31.56 (±1.21)
30	39.91 (±1.86)	25.82 (±1.43)	18.30 (±0.96)	16.02 (±0.74)
31	39.74 (±1.95)	30.15 (±1.79)	25.16 (±0.86)	20.72 (±0.95)
32	50.81 (±2.73)	42.22 (±2.60)	35.65 (±1.87)	31.37 (±1.90)

Corresponding to QLS and QLIH, respectively, Figure 4.1 shows two learning curves for the decision agent for case 1, presenting the average profit obtained by the decision agent in each decision-making epoch. Computational convergence was achieved long before the simulation run ended. The convergence plot appears to be smooth. It is clear to see that

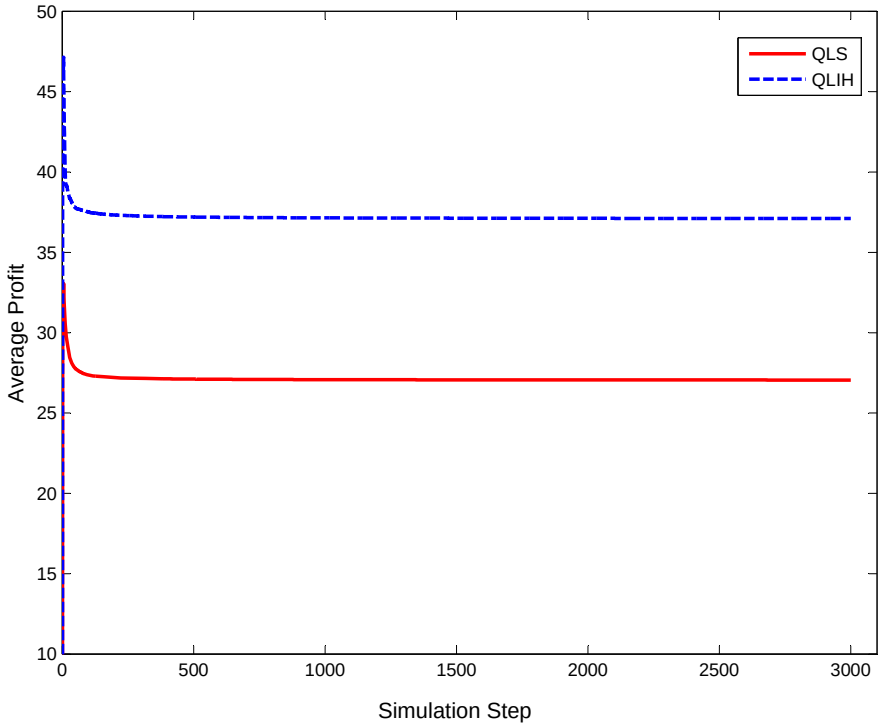

Figure 4.1: The learning curves of a decision agent for case 1

QLS had worse performance than QLIH after the convergence although they had a similar performance at the beginning of the simulation.

4.4.3 Sensitivity Analysis

Since there are several factors that can affect production scheduling decisions, it is imperative for the decision maker to know what factors most critically influence the average profit. To address this issue, we study seven factors, namely price, setup time, production rate, holding cost, backorder cost, setup cost and demand rate. The ANOVA (analysis of variance) performed on the 32 pieces of data obtained from the factorial experiment is shown in Table 4.4. From the ANOVA, it can be seen that four factors, price, producton rate, setup cost

and setup time, were significant at a 95% confidence interval with respect to the average profit obtained.

Table 4.4: Analysis of variance (ANOVA) results: Analysis performed at 95% confidence interval

Source of variation	Degree of freedom (DF)	Sums of squares (SS)	F-ratio	Probability level
Price	1	2406.52	13.11	0.0001
Setup time	1	1101.73	6	0.0219
Production rate	1	1322.91	7.21	0.0129
Holding cost	1	0.58	0	0.9558
Backorder cost	1	67.88	0.37	0.5488
Setup cost	1	1035.53	5.64	0.0259
Demand rate	1	207.05	1.13	0.2987
Error	24	4404.29		
Total (Adjusted)	31	11227.31		

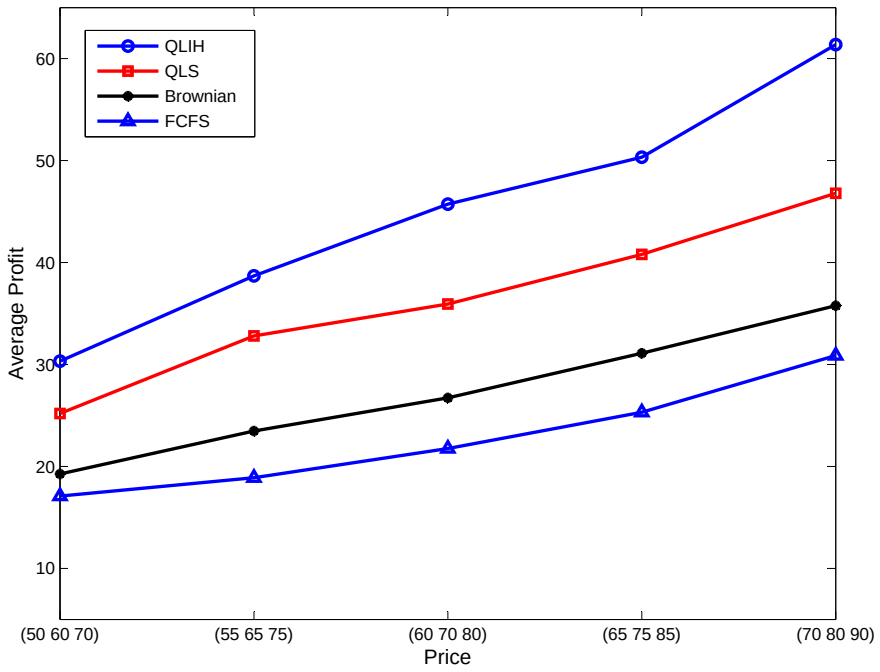

Figure 4.2: Sensitivity of QLIH, QLS, Brownian and FCFS to price for case 10

As a follow-up to the ANOVA, we studied the sensitivities of QLIH, QLS, Brownian and FCFS policies with respect to the price and production rate. For case 10 of Table 4.2, price

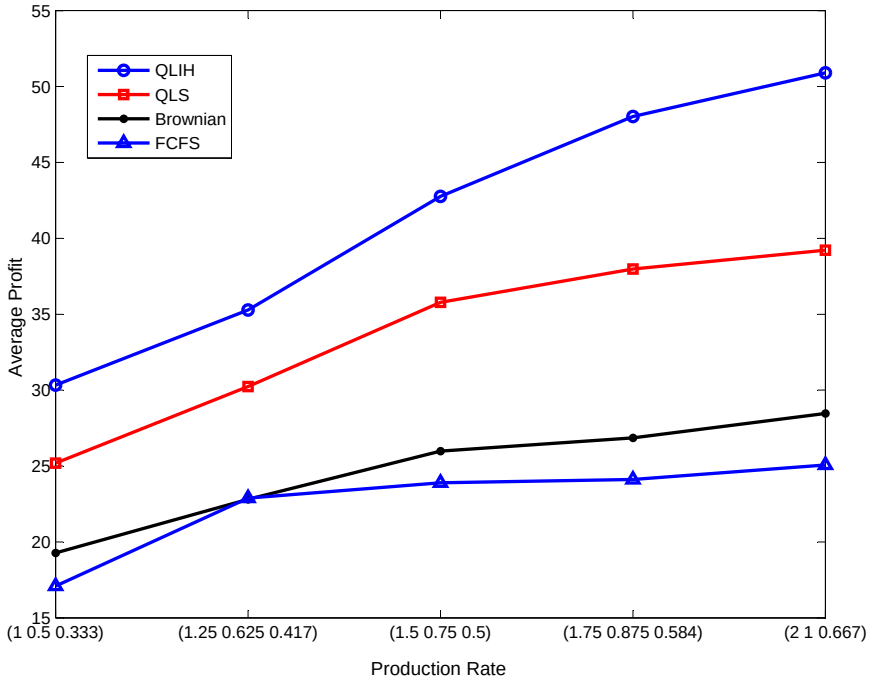

Figure 4.3: Sensitivity of QLIH, QLS, Brownian and FCFS to production rate for case 10

is varied from a low level of (50, 60, 70) to a high level of (70, 80, 90) and setup time varied from a low level of (1, 1/2, 1/3) to a high level of (2, 1, 2/3). The plots of the sensitivity of the average profit obtained for case 10 in Table 4.2 with respect to the two main factors (one at a time) are shown in Figures 4.2 and 4.3. In Figure 4.2 and 4.3, we find that QLIH increases most quickly than QLS, Brownian and FCFS policies, with the increase in price and production rate because QLIH is more sensitive to the increase in price and production rate. These also partly explain the superior performance of QLIH.

4.5 Summary

In this chapter, we investigated RM in a MTS manufacturing. The objective function of the problem is to maximize the long-run average total profit, which is the total revenue minus the

holding cost, backorder cost and setup cost. We first developed a mathematical model of the problem. Then, we proposed two RL algorithms, QLH and QLIH, to solve this problem. We set up a seven-factor two-level design, and 32 cases were tested using QLIH, QLS, Brownian and FCFS control policies. The results show that QLIH outperforms the other three policies in all cases. Furthermore, we conducted a sensitivity analysis of the QLIH, QLS, Brownian and FCFS policies with respect to the price and production rate. The results confirm the superior performance of our proposed QLIH, which show that QLIH policy is more sensitive to increase in significant factors, such as price and production rate.

Chapter 5

Conclusions

5.1 Summarized Work

This dissertation studies the revenue management (RM) in the manufacturing industry including make-to-order (MTO) and make-to-stock (MTS) production system. In Chapter 2, we study a situation in which a firm faces price and lead-time dependent demand and must jointly determine the price, lead-time and production schedule for a finite planning horizon. The goal is to maximize the total profit gained in a finite planning horizon under the production capacity, lead-time and demand constraint. The firm is a joint price and lead-time decision maker who has power to influence the customers' decisions for order quantities. In Section 2.5, we develop a mathematical model, which can be solved by optimization software CPLEX. In Section 2.4, we also develop a rounding heuristic to provide the initial feasible solution to the MIP solver, which can solve the large-size instance in a considerably short time with a satisfied solution. In the numerical experiment in Section 2.5, we study the

problem under different demand scenarios with price and lead-time flexibilities. We find that both of the price and lead-time flexibilities are useful to increase the average profit. In addition, price and lead-time flexibilities are complementary. That is, without the price flexibility, the effect of lead-time flexibility is more useful and vice versa. In general, when the demand load is high, the high level of lead-time flexibility is more useful. These results provide the insight regarding price and lead-time flexibilities for RM in MTO firms.

However, in MTO manufacturing, it is generally not possible to obtain accurate forecast information about the timing and attributes of expected future orders over the planning horizon. In Chapter 3, we investigate the pricing, lead-time, scheduling and order acceptance decisions in a MTO manufacturing system with stochastic demands. The goal of this study is to maximize the long-run average profit. We proposed a reinforcement learning (RL) algorithm, the QLA, to solve this problem. To deal with many sources of uncertainty in a MTO manufacturing system, we set up five scenarios including the baseline, the demand variation, capacity improvement, single price selection and due-date negotiation. In all these five scenarios, the QLA outperforms the other two benchmarking policies, the Heuristic and FCFS. Although the QLA has the lowest order acceptance rate in all five scenarios, it can generate the highest average reward among three policies. It is obvious that the QLA is not only highly selective on order acceptance but also adaptive to the environmental changes. The QLA is very effective to select more profitable orders with the limited production capacity. Therefore, the QLA can be implemented in a MTO manufacturing firm to make real-time decisions for RM.

In Chapter 4, we consider a RM problem in a MTS manufacturing system which consists of a single machine in which the demands and the processing times for N types of products are random. The problem decide when, what, and how much to produce so that the long-run average total profit, which is the total revenue minus the setup, holding and backorder costs, is maximized.We first developed a mathematical model of the problem. Then, we proposed two RL algorithms, QLH and QLIH, to solve this problem. We set up a seven-factor two-level design, and 32 cases were tested using QLIH, QLS, Brownian and FCFS control policies. The results show that QLIH outperforms the other three policies in all cases. Furthermore, we conducted a sensitivity analysis of the QLIH, QLS, Brownian and FCFS policies with respect to the price and production rate. The results confirm the superior performance of our proposed QLIH, which show that QLIH policy is more adaptive to change in significant factors, such as setup time and setup cost.

5.2 Potential Applications

The study results of this dissertation can be widely applied to the real world. Firstly, the mathematical model and proposed rounding heuristic developed for joint pricing, lead-time and scheduling decisions can be employed in RM by some MTO firms, where the customer demands are predictable. Second, for some MTO firms where the customer demand is hard to be predicted, the proposed RL algorithms can be applied to make the real-time decisions on pricing, lead-time and production scheduling. Thirdly, the RL algorithms developed for

MTS production systems can be used in RM of most MTS firms with the goal to maximize the long-run average total profit.

5.3 Future Work and Directions

In the mathematical model in Chapter 2, the capacity allocation decision has not been considered. In the future research, we may consider incorporating the capacity decision in the model and studying the joint optimization decisions on capacity, pricing, lead-time and scheduling. Moreover, in this model, we assume the demand function is a linear function, which depends on the price and lead-time. Future research directions also include incorporating other demand functions.

In Chapter 3 and 4, we employ RL approaches to make real-time RM decisions for MTO and MTS production systems. However, we only consider a single machine production system. First, further studies can be extended to apply the RL approach to the multi-machine production setting. Second, a disadvantage of RL is that the policy learned by the decision agent was hard to determine since the agent's actions depended on a black box process. In addition, the learning process was a little harder for the practitioners to interpret. Future research should focus on how to derive a parametric policy that can give the action selection, from the RL process with a combination of other techniques, such as data mining. Third, since there are only three products considered in theses two problem, future research might also focus on how to reduce the complexity of the system while incorporating more products into it. Last, in Chapter 3, we use one type of neural network approach for function

approximation. In future work, other neural network approach for function approximation can be explored to make better use of the learning information.

Bibliography

Ahn, H., Gumus, M., Kaminsky, P., 2007. Pricing and manufacturing decisions when demand is a function of price in multiple period. Operations Research 55 (6), 1039–1057.

Anupindi, R., Tayur, S., 1998. Managing stochastic multiproduct systems: model, measures, and analysis. Operations Research 46 (3), s98–s111.

Bianchi, R. A. C., Ribeiro, C. H. C., Costa, A. H. R., 2004. Heuristically accelerated q-learning: a new approach to speed up reinforcement learning. Lecture Notes in Artificial Intelligence.

Bourland, K. E., Yano, C. A., 1994. The strategic use of capacity slack in the economic lot scheduling problem with random demands. Management Science 40 (12), 181–200.

Boyd, E. A., Bilegan, I. C., 2003. Revenue management and e-commerce. Management Science 49 (10), 1363–1386.

Caldentey, R., Lawrence, M. W., 2006. Revenue manageme in make-to-stock queue. Operations Research 54 (5), 859–875.

Carr, S., Duenyas, I., 2000. Optimal admission control and sequencing in a make-to-stock/make-to-order production system. Operations Research 48 (5), 709–720.

Charnsirisakskul, K., Griffin, P., Keskinocak, P., 2006. Pricing and scheduling decisions with leadtime flexibility. European Journal of Operational Research 171, 153–169.

Chen, M., Chu, M., 2003. The analysis of optimal control model in matching problem between manufacturing and marketing. European Journal of Operational Research 150, 293–303.

Cross, R. G., 1997. Revenue Management. Broadway, New York.

Darken, C., Moody, J., 1992. Towards fasters stochastic gradient search. In: Moody, J., Hanson, S. (Eds.), Advances in Neural Information Processing Systems. Morgan Kaufmann, San Mateo, CA, pp. 1009–1016.

Das, T. K., Gosavi, A., Mahadevan, S., Marchalleck, N., 1999. Solving semi-markov decision problems using average reward reinforcement learning. Management Science 45 (4), 560–574.

Davis, P., 1994. Airline ties profitability yield to management. SIAM News 27.

Deng, S., Yano, C. A., 2006. Joint production and pricing decision with setup costs and capacity constraints. Management Science 52 (5), 741–756.

Duenya, I., 1995. Single facility due date setting with multiple customer classes. Management Science 41 (4), 608–619.

Duenya, I., Hopp, W. J., 1995. Quoting customer lead times. Management Science 41 (1), 43–57.

Easton, F., Moodie, D., 1999. Pricing and lead time decision for a make-to-order firms with contingent orders. European Journal of Operational Research 116 (2), 305–318.

Eliashberg, J., Steinberg, R., May 1989. Marketing-production joint decisions. PHD thesis, Kings College, Cambridge, England.

Eliashberg, J., Steinberg, R., 1993. Handbooks in Operations Research and Management Science: Marketing, Vol.5. North Holland, Amsterdam, The Netherland, Ch. Marketing-production joint decisions, pp. 827–880.

Elmaghraby, S. E., 1978. The economic lot scheduling problem(elsp): review and extensions. Management Science 24 (6), 587–598.

Feichtinger, G., Hartl, R., 1985. Optimal pricing and production in an inventory model. European Journal of Operational Research 19, 45–46.

Gallego, G., 1990. Scheduling the production of several items with random demands in a single facility. Management Science 36 (12), 1579–1592.

Gallego, G., Roundy, R., 1992. The economic lot scheduling problem with finite back-order costs. Naval Research Logistics Quarterly 39, 729–739.

Gilbert, S. M., 1999. Coordination of pricing and multi-period production for constant priced goods. European Journal of Operational Research 114, 330–337.

Gilbert, S. M., 2000. Coordination of pricing and multi-period production across multiple constant priced goods. Management Science 46 (12), 1602–1616.

Gosavi, A., 2004a. A reinforcement learning algorithm based on policy iteration for average reward: empirical results with yield management and convergence analysis. Machine Learning 55 (1), 5–29.

Gosavi, A., 2004b. Reinforcement learning for long-run average cost. European journal of operations research 155 (3), 654–674.

Gosavi, A., Bandla, N., Das, T. K., 2002. A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking. IIE Transactions 34 (9), 729–742.

Hall, N., Magazine, M., 1994. Maximizing the value of a space mission. European Journal of Operational Research 78, 224–241.

Harris, F., Pinder, J., 1995. A revenue management approach to demand management and order booking in assemble-to-order manufacturing. Journal of Operations Management 22, 299–309.

Harrison, J. M., 1988. Brownian models of queuing networks with heterogeneous customer populations. In: Fleming, W., Lions, P. L. (Eds.), Stochastic Differential Systems, Stochastic Control Theory and Applications. Springer-Verlag, New York, pp. 147–186.

Hing, M. M., van Harten, A., Schuur, P. C., 2007. Reinforcement learning versus heuristics for order acceptance on a single resource. Journal of Heuristic 3 (12), 167–187.

Hopp, W., Spearman, M., 2000. Factory Physics. McGraw-Hill, Inc., Columbus, OH.

Hsu, W., 1983. On the general feasibility test of scheduling lot sizes for several products on one machine. Management Science 29 (1), 93–105.

Kaelbling, L. P., Littman, M. L., Moore, A. P., 1996. Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285.

Keskinocak, P., Ravi, R., Tayur, S., 2001. Scheduling and reliable lead time quotation for orders with availability intervals and lead time sensitive revenues. Management Science 47 (2), 264–279.

Keskinocak, P., Tayur, S., 2004. Handbook of quantitative supply chain analysis: modeling in the E-business era. Kluwer Academic Publishers, Ch. Due-date management policies.

Kim, D., Lee, W. J., 1998. Optimal joint pricing and lot sizing with fixed and variable capacity. European Journal of Operational Research 109, 212–227.

Law, A. M., Kelton, W. D., 1991. Simulation modeling and analysis. McGraw-Hill, Inc., New York, NY.

Lawrence, S. R., 1995. Estimating flowtimes and setting due-dates in complex production systems. IIE Transactions 27 (5), 657–688.

Leachman, R. C., Gascon, A., 1988. A heuristic scheduling policy for multi-item, single-machine production systems with time-varying, stochastic demands. Management Science 34 (3), 377–390.

Leibs, S., 2000. Ford heads the profits. CFO The Magazine 16, 33–35.

Li, L., 1992. The role of inventory in delivery-time competition. Management Science 38, 182–197.

Lippman, S., Ross, S., 1971. The streetwalker's dilemma: A job shop model. SIAM Journal of Applied Mathematics 20, 336–342.

Mahadevan, S., 1996. Average reward reinforcement learning: foundations, algorithms and empirical results. Machine Learning 22, 159–196.

Mahadevan, S., Khaleeli, N., Marchalleck, N., 1997a. Designing agent controllers using discrete-event markov models. AAAI Fall Symposium on Model-Directed Autonomous Systems, MIT, Cambridge.

Mahadevan, S., Marchalleck, N., Das, T., Gosavi, A., 1997b. Self-improving factory simulation using continuous-time average-reward reinforcement learning. In: Proceedings of the Fourth International Machine Learning Conference. pp. 202–210.

Mahadevan, S., Theocharous, G., 1998. Optimizing production manufacturing using reinforcement learning. The Eleventh International FLAIRS Conference, AAAI Press 9, 372–377.

Markowitz, D. M., Reiman, M. I., Wein, L. M., 1995. The stochastic economic lot scheduling problem: heavy traffic analysis of dynamic cyclic policies. Working paper 3863-95-MSA.

McGill, J. I., van Ryzin, G. J., 1999. Revenue management: research overview and prospects. Transportation Science 33, 233–256.

Miller, B., 1969. A queueing reward system with several customer classes. Management Science 16 (3), 234–245.

Nicholas, J., 1997. Competitive Manufacturing Management: Continuous Improvement. McGraw-Hill, Inc., Columbus, OH.

Palaka, K., Erlebacher, S., Kropp, D. H., 1998. Lead-time setting, capacity utilization and pricing decision under lead-time dependent demand. IIE Transactions 30 (2), 151–163.

Paternina-Arboleda, C. D., Das, T. K., 2001. Intelligent dynamic control of single-product serial production lines. IIE Transaction 33 (1), 65–77.

Paternina-Arboleda, C. D., Das, T. K., 2005. A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem. Simulation Modelling Practice and Theory 13, 389–406.

Pekelman, D., 1974. Simultaneous price-production decisions. Operations Research 22 (4), 788–794.

Philipoom, P., Rees, L., Wiegmann, L., 1994. Using neural networks to determine internally-set due-date assignments for shop scheduling. Decision Science 25 (5/6), 825–851.

Qiu, J., Loulou, R., 1995. Multiproduct production/inventory control under random demands. IEEE Transactions on Automatic Control 40 (2), 350–356.

Rao, U. S., Swaminathan, J. M., Zhang, J., 2000. Integrated demand and production management in a periodic, make-to-order setting with uniform guaranteed lead time and outsourcing. Working Paper, GSIA, Carnegie Mellon University, Pittsburgh, September.

Raza, S. A., Akgunduz, A., Chen, M. Y., 2006. A tabu search algorithm for solving economic lot scheduling problem. Journal of Heuristics 12 (6), 413–426.

Rehkopf, S., Spengler, T., 2004. Revenue management in a make-to-order environment. In: Moody, J., Hanson, S. (Eds.), Operations Research Proceeding. Springer, Berlin Heidelberg New York, pp. 470–478.

So, K. C., 2000. Price and time competition for service delivery, manufacturing and service. Operations Management 2 (4), 392–409.

So, K. C., Song, J.-S., 1998. Price, delivery time guarantees and capacity selection. European Journal of Operational Research 111 (1), 28–49.

Sox, C. R., Muckstadt, J. A., 1997. Optimization-based planning for the stochastic lot scheduling problem. IIE Trasaction 29 (5), 349–357.

Spengler, T., Rehkopf, S., Volling, T., 2007. Revenue management in make-to-order manufacturing-an application to the iron and steel industry. OR Spectrum 29 (1), 158–171.

Stalk, G. J., Hout, T. M., 1990. Competing Against Time. The Free Press.

Sutton, R. L., 1988. Learning to predict by the method of temporal differences. Machine Learning 3, 9–33.

Szepesvari, C., Littman, M. L., 1996. Generalized markov decision processes: Dynamic-programming and reinforcement-learning algorithms. Tech. rep., Brown University.

Talluri, K., van Ryzin, G., 2004. The Theory and Practice of Revenue Management. Springer, New York.

Vanthienen, L. G., 1975. Simultaneous price-production decision making with production adjustment costs. In TIMS XX international meeting.

Webster, S., 2002. Dynamic pricing and lead-time policies. Decision Science 33 (4), 579–599.

Wein, M. L., 1992. Dynamic scheduling of a multiclass make-to-stock queue. Operations Research 40 (4), 724–735.

Whitin, T., 1955. Inventory control and price theory. Management Science 2 (1), 61–68.

Wu, A., Chiang, D., 2009. The impact of estimation error on the dynamic order admission policy in b2b mto environments. Expert Systems with Applications 36, 11782–11791.

Yano, C. A., Gilbert, S. M., 2004. Managing Business Interfaces: Marketing, Engineering and Manufacturing Perspectives. Kluwer Academic Publishers, Ch. Coordinated pricing and production/procurement decisions: A review.

Zeng, D., Sycara, K., 1995. Using case-based reasoning as a reinforcement learning framework for optimization with changing criteria. In: Proceedings of the 7th International Conference on Tools with Artificial Intelligence, Takamatsu, Japan. pp. 56–62.

Vita

Jiao Wang received the dual B.S. degrees in Industrial Engineering, Management and M.S. degree in Management Science and Engineering from School of Management, Xi'an Jiaotong University, Xi'an, China. In the fall of 2007, she joined as a graduate assistant the Intelligent Information Engineering and Systems Laboratory (IIESL) at the Department of Industrial and Information Engineering, The University of Tennessee at Knoxville. She is expected to complete her Doctor of Philosophy degree in 2011. She has published several peer-review journal and conference papers. Now she is working as an operations research analyst for Bank of America.