European Journal of Operational Research 221 (2012) 99–109

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

Reinforcement learning for joint pricing, lead-time and scheduling decisions in make-to-order systems

Xueping Li *, Jiao Wang, Rapinder Sawhney

Department of Industrial and Information Engineering, University of Tennessee, Knoxville, TN 37996, USA

ARTICLE INFO

Article history:
Received 11 October 2011
Accepted 9 March 2012
Available online 20 March 2012

Keywords:
Pricing
Reinforcement learning (RL)
Scheduling
Q-learning
Simulation-based optimization
Semi-Markov Decision Problem (SMDP)

ABSTRACT

The paper investigates a problem faced by a make-to-order (MTO) firm that has the ability to reject or accept orders, and set prices and lead-times to influence demands. Inventory holding costs for early completed orders, tardiness costs for late delivery orders, order rejection costs, manufacturing variable costs, and fixed costs are considered. In order to maximize the expected profits in an infinite planning horizon with stochastic demands, the firm needs to make decisions from the following aspects: which orders to accept or reject, the trade-off between price and lead-time, and the potential for increased demand against capacity constraints. We model the problem as a Semi-Markov Decision Problem (SMDP) and develop a reinforcement learning (RL) based Q-learning algorithm (QLA) for the problem. In addition, we build a discrete-event simulation model to validate the performance of the QLA, and compare the experimental results with two benchmark policies, the First-Come-First-Serve (FCFS) policy and a threshold heuristic policy. It is shown that the QLA outperforms the existing policies.

1. Introduction

Traditionally, manufacturers usually do not contact end customers directly, which makes it difficult to influence demands by quoted price for manufactures. Hence, the integration of decisions on pricing and scheduling has less meaning in practice. However, during the past decade, direct selling has become more common due to advancements in technology, particularly with the Internet development. As reported by Direct Selling Association, the direct selling becomes a significant business sector, which had 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 many computer companies. For instance, Dell Computers divides its customers into different classes and may charge them different prices. Other companies, such as IBM and HP, also adopt the same strategy. To name another example in a different industry, between 1995 and 1999, Ford Motor Co. gained $3 billion in growth by applying pricing strategies to match supply, demand and target specific customer segments (Leibs, 2000). Generally, direct selling makes it easy to observe or even influence end customers' behavior (Boyd and Bilegan, 2003). Under this circumstance, joint pricing and production scheduling problems arise in the manufacturing firm and draw an increasing interest.

Price is used to be considered as a key factor to win competition. However, the success of the Japanese manufacturing Just-In-Time (JIT) philosophy has highlighted the importance of short lead-times. In today's time-based competitive market, practitioners recognize that customer demand increases not only along with the decrease of price but also the decrease of lead-time (So and Song, 1998; Stalk and Hout, 1990; So, 2000). Lead-time (due-date) decisions depend on several factors, such as manufacturers' capacity, customers' demands, and due-date preferences. With recognition of lead-time sensitive demands, many companies, specifically the make-to-order (MTO) manufacturing sectors, offer all customers a uniform lead-time within which they guarantee to satisfy "most" orders (So and Song, 1998; Rao et al., 2000). The risk that the demand may exceed the capacity may arise due to a larger number of customers attracted by this strategy. If the demand cannot be satisfied, a lateness penalty cost for the manufacturers or customer dissatisfaction may occur, and future business may suffer. Firms then must balance the price, the lead-time and the potential for increased demand against capacity constraints.

By adopting the MTO system, firms benefit by eliminating finished goods inventory carrying and obsolescence costs (Nicholas, 1998). However, this benefit comes at the cost of increased response time to satisfy customer orders and/or the cost of keeping higher production capacity to accommodate demands variations (Hopp and Spearman, 2000). If spare production capacities and flexible order due-dates are not profitable, it becomes very critical for an MTO manufacturer to selectively accept and schedule customer orders. One of the common characteristics of MTO companies is the rigidity of production capacity. Although manufacturing capacity can be

* Corresponding author. Address: Department of Industrial and Information Engineering, University of Tennessee, 408 East Stadium Hall, Knoxville, TN 37996-0700, USA. Tel.: +1 865 974 7648; fax: +1 865 974 0588.

E-mail addresses: Xueping.Li@utk.edu (X. Li), jiao.wang@utk.edu (J. Wang), sawhney@utk.edu (R. Sawhney).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.03.020

considered as flexible in the long term, increasing the capacity significantly in the short term is not possible. Furthermore, accepted orders require the resource availability to guarantee order promises, which potentially displaces more profitable orders to come due to 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).

In this paper, we examine a situation in which a firm must jointly determine the price, the lead-time and the production schedule for each order. The goal is to maximize the total profit gained in an infinite planning horizon under constraints of production capacities, lead-times and demands. This situation is common in MTO firms, which make joint price and lead-time decisions and have power to influence customers' decisions for order quantities. But it can not predict customers' demands. In this paper, the lead-time is the time period from the moment the manufacturer receives an order to the moment this order is shipped, and the customers are assumed to be responsible for the delivery costs. We consider a multi-machine production system with perfect reliability. Manufacturing fixed costs (including the overhead cost, setup cost and quality cost) should be considered in the 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 quoted prices and lead-times, order sizes, 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 that the customer is willing to pay or the quoted lead-time is longer than that the customer is willing to wait. However, the rejection incurs the order rejection cost. For accepted orders, the manufacturer will set the production and delivery schedule. 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 a 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.

In the MTO system 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 make the real-time decisions in an infinite time horizon. The RL is derived from dynamic programming (DP) and stochastic approximation, which will be introduced in details in Section 3. An order completed before its quoted lead-time is considered early and incurs inventory holding cost. 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 cost (including costs of raw material, direct labor and maintenance) is a linear function of production quantity in addition to a fixed cost for each production period. Production cost, inventory holding cost and lateness penalty cost are known as time-dependent. The total cost consists of fixed cost, production cost, inventory holding cost and lateness penalty cost. Therefore, the total profit is the total revenue subtracted by the total cost.

The paper contributes to the literature in three aspects. Firstly, to the best of our knowledge, this is the first to investigate the joint pricing, lead-time and scheduling decisions simultaneously in MTO systems by developing a Semi-Markov Decision Process (SMDP) model. Secondly, we propose a reinforcement learning based Q-learning algorithm (QLA) to solve the problem. Thirdly, a simulation model in ARENA® is developed to validate the QLA, compared with the other two benchmarking policies, the FCFS policy and a threshold heuristic policy.

The rest of the paper is organized as follows. Section 2 reviews the relevant literature on joint pricing and production scheduling problems. Section 3 describes basic concepts and working mechanisms of the RL, and the related literature on the RL. Section 4 provides the control state-action space of the problem, the development of the SMDP model, and the QLA. Section 5 describes the numerical experiments and compares the performance of the QLA with two benchmarking polices as well as the simulation method. Section 6 summaries the study and points out directions for future research.

2. Literature review and related work

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

Optimal control approaches incorporate both static and dynamic pricing, and deterministic production quantities. Linear, convex and non-smooth functions are used for production and inventory costs, while demand functions are always assumed to be linear. Pekelman (1974) developed an optimal control model for profit maximization with both price and production quantities as decision variables over a planning horizon. Vanthienen (1975) incorporated a capacity constraint into Pekelman's model. Pekelman's model was further extended by allowing backlogging in Feichtinger and Hartl (1985). Chen and Chu (2003) found the optimal production rate and sales rate at each time to maximize profit under different pricing strategies The research above did not consider constraints on capacity or storage.

Mathematical programming models divide the planning horizon into discrete time buckets, and are closely related to production planning models. Kim and Lee (1998) addressed the problem of joint pricing, lot sizing and capacity expansion decision over a planning horizon. Gilbert (1999) studied a model in which a constant price was maintained for a seasonal product in advance of the demand observation. Deng and Yano (2006) gave the optimal solution of price and production quantities for a manufacturer with price-sensitive demand, in which both capacity constraint and setup costs were taken into consideration in an finite discrete time horizon. They presented a model that integrated pricing, production and order selection decisions for MTO manufacturer who could set price to influence demand and set lead times for accepted orders. Ahn et al. (2007) considered a joint production and pricing decision problem, in which the portion of the demand realized in each period is induced by the interaction of pricing decisions in the current period and in the previous period. All of the above models did not consider the influence of lead-time on demand functions.

The second body of research related to our work focuses on the production scheduling and due-date quotation. Chryssolouris et al. (2004) introduced the concept of chaos theory for production scheduling in manufacturing systems. 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 summaries of recent work, see Philipoom et al. (1994), Lawrence (1995) and Easton and Moodie (1999). Duenya and Hopp (1995) and Duenya (1995) firstly incorporated the customer order selection into the firm's lead-time policy where the prob-

ability that a customer placed an order decreased as the quoted lead-time increased. Some researchers considered the order acceptance problem, where a manufacturer had the option to reject orders and accepted orders must be finished by specified due-dates with the objective to maximize the revenue or the profit (Hall and Magazine, 1994; Keskinocak et al., 2001; Charnsirisakskul et al., 2006).

The third body of research related to our research is the Revenue Management (RM) based order acceptance subject to limited manufacturing capacity. The earliest models for selective order acceptance policies that can be applied in MTO systems were developed by Miller (1969) and Lippman and Ross (1971). Their models assume that order service times are exponentially distributed, and there are no due-date constraints or lateness penalties associated with producing the orders. Miller (1969) considered the order acceptance problem as an admission control problem with a queue. Lippman and Ross (1971) extended Miller's model by allowing service times to depend on the customer classes. 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, and 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, the total expected return is maximized. Recently, Spengler et al. (2007) developed a RM approach for companies in the iron and steel industry. The aim is to improve the short-term order selection. Due to the complexity of obtaining opportunity costs from dynamic programming approaches, a multi-dimensional knapsack problem is formulated to obtain static bid-price based approximation scheme for selecting profitable orders.

This research is more relevant to the literature that considers price, lead-time quotation, production quantities and capacity investment simultaneously. So and Song (1998) presented 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 the price was sensitive to both the price and the lead-time. However, they did not consider lateness penalty costs in the objective function, and quoted uniform delivery time for every order. Palaka et al. (1998) modeled the firm's operations as a M/M/1 queue system, and treated demand as a linear function of quoted price and lead-time. The objective is to maximize revenues subtracting total variable production costs, congestion related costs and lateness penalty costs, subject to a service level constraint. Webster (2002) developed dynamic pricing and lead-time policies for an MTO system using the similar linear demand function.

In this study, we model the customer demand as a linear function of price and lead-time as in So and Song (1998), and we consider the production cost, lateness patently cost, inventory holding cost as in Charnsirisakskul et al. (2006). In addition, we also consider the setup costs incurred in each production period mentioned in the study of Deng and Yano (2006) and the order rejection cost. Our major contribution is that we firstly incorporate the order acceptance decision, setup costs and order rejection costs into the joint pricing, lead-time and production decision problem with lead-time and price sensitive demands and capacity constraints. However, in order to fit the real-world manufacturing settings, we replace the setup cost and the production cost by the manufacturing fixed cost and the variable cost.

3. Reinforcement learning (RL)

3.1. Introduction on RL

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 named 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. On the one hand, the learning agent in RL keeps interacting with its dynamic environment and chooses actions based on the feedback from the environment according to certain policies; on the other hand, the environment responds to 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 Fig. 3.1, a single-agent learning model contains the following elements: the system environment, 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, 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 transits to a new state. Then, the environment assigns the immediate reward or punishment 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 are less. This learning process ends when the agent receives 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 the large-scale Markov decision process (MDP) and is widely used as a promising method in the area of machine learning. Various simulation-based RL methods are derived from the stochastic approximation, such as the method of Temporal Differences, TD (λ) (Sutton, 1988) and Q-learning (Eliashberg and Steinberg, 1993). Stochastic approximation methods attempt iteratively to find zeros or extrema of functions which cannot be solved directly but only estimated via noisy observations. The early studies on RL only aimed to maximize the cumulative reward. Mahadevan (1996) firstly undertook a detailed study on the average reward RL. Later, Das et al. (1999) developed a new model-free average-reward algorithm called SMART for continuous-time semi-Markov decision

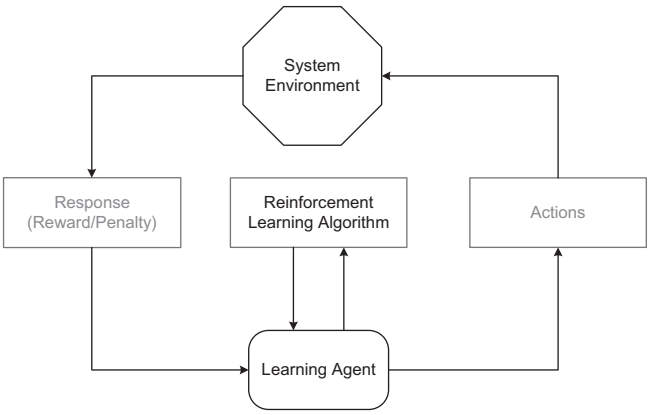

Fig. 3.1. The learning mechanism of an agent.

processes (SMDPs). Compared with the classical DP, RL avoids the need for computing transition probabilities and reward matrices. RL only requires the probability distributions of the process random variables since it is based on the discrete event simulation (Law and Kelton, 1991). RL not only overcomes some of the drawbacks of the classical DP but also can be used in the large state space, e.g., with 10^{15} states (Das et al., 1999). There are strong theoretical guarantees on convergence of most RL algorithms (Szepesvari and Littman, 1996), such as the 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 literatures

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

There are a few applications of RL in solving production scheduling problems in the literature. Zeng and Sycara (1995) firstly applied a RL approach for a job-shop scheduling problem. They developed a repair-based scheduler which was trained using the temporal difference RL algorithm. The algorithm starts with a critical-path schedule and incrementally repairs constraint violations. Mahadevan et al. (1997a,b) and Das et al. (1999) applied the SMART algorithm to the problem of optimizing preventative maintenance in a production inventory system, where a single machine was capable to produce multiple types of products with multiple buffers for storing each of the different products. The SMART outperformed another two heuristics in determining proper maintenance schedules regarding to costs. Mahadevan and Theocharous (1998) also employed SMART to optimize the policy for a three-machine transfer line system in which a single type of products is produced. They found that the policy obtained from SMART was better than that achieved by the kanban heuristic, with respect to maximizing the throughput while minimizing work-in-process (WIP) and failures. Paternina-Arboleda and Das (2001) extended the previous work to deal with a four-machine serial line and compared SMART with existing control policies. They examined the system with a constant demand rate and a Poisson demand rate. Under these two circumstances, SMART outperformed those heuristic policies on average WIP level and average WIP cost.

Some researchers have worked on applying RL methods in RM, and most of the applications focus on airline industry. The first application of RL in a single leg airline RM problem (e.g., pricing and seat allocation) is presented in Gosavi et al. (2002). They proposed an approximate RL approach to solve a continuous-time semi-Markov decision process with random demands arrival and cancelations. This approach is developed based on value iteration and employ function approximation with neural networks for estimating the value function of DP within a simulator. Gosavi (2004a) further developed a RL algorithm based on policy interaction for the same problem. He tested the proposed RL algorithm with a nearest-neighbor approach to tackle a large state space on an airline RM problem.

Although a great deal of work in algorithmic development and applications of RL have already existed, RL continues attracting research attention. To the best of our knowledge, this paper is the first one that applies RL to solve joint pricing, lead-time, order acceptance and production scheduling decision problem. We develop an infinite horizon model with the average reward performance criterion and use one of the RL algorithms, Q-learning algorithm, to solve this model.

4. Mathematical model and algorithm

The firm considered in this paper has price and lead-time dependent demands and must jointly determine the price, lead-time and production schedule for an infinite planning horizon under the fixed production capacity constraints. It is assumed that there are multiple types of customers whose demands are dependent on the quoted lead-time and price. Whenever a customer places an order, the firm needs to make a decision whether to accept or reject the order. It is assumed that the arrivals of orders from different customers are independent Poisson processes. 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 produced. The inventory holding cost is incurred if an order is 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 cost and the variable cost 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 a SMDP model for the problem.

4.1. SMDP model

In order to model this joint decision problem as a SMDP, we first define the system state-space. If there are n types of customers in the system, the system state ($\boldsymbol{\psi}$) can be denoted by the vector

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

where m represents the most recent order type (among n possible types) placed by a customer, θ is the order quantity of the most recent order, μ_1, μ_2, …, μ_n represent the order quantities 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 $pt^i_{j,q} = d^i_{j,q}$. It can be shown that the cardinality of the state-space has an upper bound of $n \times M^{(n+2)}$, where M is the maximum number of customers in any given type of order within this period, which is the production period capacity.

Clearly, a change of system state is caused by any one the following three events: (1) a new customer arrives and places an order; (2) the production is completed for a specific unit of an order, and (3) the production period is ended. Whenever a customer places an order, a decision whether to accept or reject the order needs to be made. 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 $\boldsymbol{\psi}_m$ and $\boldsymbol{\sigma}_m$ denote the system state at the mth decision-making epoch. We define two stochastic processes: $\boldsymbol{\psi} = \{\boldsymbol{\psi}_m : m \in \mathcal{N}\}$ and $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_m : m \in \mathcal{N}\}$, where $\mathcal{N}$ is the set of natural numbers. We also define a joint process $(\boldsymbol{\psi}, \boldsymbol{\sigma}) = \{\boldsymbol{\psi}_m, \boldsymbol{\sigma}_m : m \in \mathcal{N}\}$. Assuming that the demand arrival process for the order type i is Poisson, it can be easily shown that

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

which means that the process ($\boldsymbol{\psi}$,$\boldsymbol{\sigma}$) is a Markov renewal process. Then the decision process related to ($\boldsymbol{\psi}$,$\boldsymbol{\sigma}$) is a SMDP, and $\boldsymbol{\psi}$ 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 is referred to

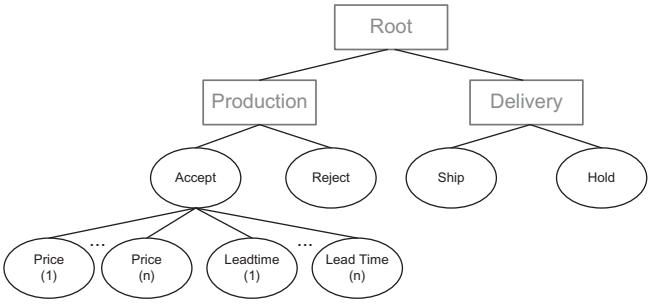

Fig. 4.1. The action space of the SMDP model.

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) $\mathcal{A}$ is defined as the set of all possible actions that can be chosen at any decision-making state (epoch). The action space can be divided into two sets: production ($\mathcal{C}$) and delivery ($\mathcal{D}$). The action "production" corresponds to the action of whether to accept or reject the arrival order, while the action set "delivery" corresponds to the action of whether to ship or hold the completed orders, i.e., $\mathcal{C}$ ={accept, reject} and $\mathcal{D}$ ={ship, hold}. Thus, the action space is $\mathcal{A} = \mathcal{C} \cup \mathcal{D}$. If the action "accept" in the set "production" is selected, the price and lead-time is chosen from the specific price and lead-time sets. Fig. 4.1 shows the hierarchy of the action space. At each decision-making epoch, the agent decides to take action a from the action space $\mathcal{A}$. If we define the current and the next state as $\mathbf{s}$ and $\mathbf{s}'$ ($\mathbf{s},\mathbf{s}' \in \boldsymbol{\psi}$), respectively, the current state-action pair is defined as $(\mathbf{s},a)$.

4.2. Q-learning algorithm (QLA)

Our problem can be modeled as a SMDP with the following characters: state space, action space and immediate rewards (Qiu and Loulou, 1995). We use one of the RL algorithms, the Q-learning algorithm (QLA). The Q-learning algorithm developed by Watkins (1989), is a temporal difference (TD) method that approximates $Q(s,a)$ directly, where (s,a) is a state-action pair. Q-learning algorithm updates 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 to 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.

The following notation is used to describe our QLA algorithm.

m: the current iteration (step) index
t_m: the simulation time at the mth decision-making epoch
maxtime: the maximum time of iterations
$\mathcal{A}$: the action space
a: an action in the action space $\mathcal{A}$
$Q_m(\mathbf{s},a)$: the action value (Q function value) for the state-action pair $(\mathbf{s},a)$ at the mth decision-making epoch
a_m: the action taken at the mth decision-making epoch
c_m: the cumulative reward at the mth decision-making epoch
ρ_m: the average reward at the mth decision-making epoch
α_m: the learning rate at the mth decision-making epoch
p_m: the probability of exploratory at the mth decision-making epoch
$\tau(\mathbf{s},\mathbf{s}',a)$: the transition time from the current state $\mathbf{s}$ to the next state $\mathbf{s}'$ due to the action a
$r(\mathbf{s},\mathbf{s}',a)$: the immediate rewards earned as a result of taking the action a in the current state $\mathbf{s}$ and leading to the next state $\mathbf{s}'$

At each epoch m, a learning rate denoted by α_m is used. In addition, we use the exploratory rate p_m which decreases with the simulation iterations in order to guarantee the convergence of the algorithm. 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 the 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, with the vlaue of 0.1, 0.1 and 10^{15}, respectively. In the early stages of adaptation, the value of learning rate and exploratory rate are approximately the same as the constant α_0 and p_0 as the algorithm mainly explores. As the iteration number m approaches the constant χ, the algorithm converges. If m is sufficiently large compared to the search time constant χ, the learning rate operates as a traditional stochastic approximation algorithm.

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 the probability $(1 - p_m)$, choose an action a^{min} that minimizes $Q_m(\mathbf{s},a)$; otherwise randomly choose an action from $\mathcal{A}$. Denote the chosen action as $\tilde{a}$
4 **if** the order can be inserted into the current schedule **then**
5 $a_m \leftarrow \tilde{a}$
6 **else**
7 **while** the order can not be inserted into the current schedule **do**
8 Randomly choose an action from $\{\mathcal{A} \setminus \tilde{a}\}$. 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 $\mathbf{s}$ and $\mathbf{s}'$, respectively.
13 $Q_{m+1}(\mathbf{s},a_m) \leftarrow (1-\alpha_m)Q_m(\mathbf{s},a_m) + \alpha_m(r(\mathbf{s},\mathbf{s}',a_m) - \rho_m\tau_m(\mathbf{s},\mathbf{s}',a_m) + \max_{b\in\mathcal{A}} Q_m(\mathbf{s}',b))$
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 $m \leftarrow m + 1$

In the QLA, step 1 (line 1) initializes the simulation parameters and chooses the initial state randomly. Then, at the iteration m, lines 2–3 choose the action a^{min} that minimizes the value of $Q_m(\mathbf{s},a)$ with a probability $(1-p_m)$; otherwise an action is randomly chosen from $\mathcal{A}$ with equal probabilities. RL involves a conflict between exploitation and exploration. When deciding which action to take, the RL agent has to balance two conflicting goals: exploiting what it has already learned in order to have a high reward, and exploring new ways to learn more. In order to balance these two objectives, there is an exploratory rate in RL algorithms. The exploratory rate p_m decreases with the simulation iterations to guarantee the convergence of QLA as in line 21. Step 2 (lines 4–11) checks the feasibility to make sure that the order can be inserted 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, the action will be executed; otherwise, another action will be chosen randomly from the action space.

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

If a nonexploratory (nonrandom) action is chosen, lines 14–17 update the total time t_m, the total cost c_m, and the average cost ρ_m. Otherwise, as in lines 19–21, the total time t_m, total cost c_m, and average cost ρ_m remain the same as they are in the $m-1$ decision-making epoch. The last step (lines 23–26) sets the current state $\mathbf{s}$ to be the 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.

4.3. 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, so 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.

The 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 of a small number of scalers (weights of the neural network). A primitive class of NN, consisting of a single neuron which is operated 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 nodes. Our input nodes denote the variables used to define the state-action space and the output nodes are the Q-values. 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-value for the taken action 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 changing the corresponding weights of the NN. The rule (algorithm) we used to update the weights is the delta-rule or the *Widrow–Hoff* rule (Widrow and Hoff, 1960). The algorithm is based on instantaneous estimate 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 notation and updating algorithm is given below:

y_m: the actual output unit (the old Q-value) at the mth decision-making epoch
$\hat{y}_m$: the target output unit (the new Q-value) at the mth decision-making epoch
δ_m: the error between the actual output and the target output at the mth decision-making epoch
w_m^i: estimate of the ith input unit weight at the mth decision-making epoch
η: the learning rate
x_m^i: the ith input unit at the mth decision-making epoch
p: the number of input units that undergoes updating during the learning process (equivalent to the number of partitions for the problem
m_{max}: the maximum simulation epoch

Algorithm 2. 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, 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 reinforcement learning. Line 7–9 update each network weight w^i. In the numerical problems that we studied, η was decayed according the DCM scheme.

5. Numerical experiment

In the numerical experiments, there are different types of customer orders for an MTO manufacturer. Each order can be quoted three levels of prices and three lengths of lead-times. The price set and the lead-time set 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 is a linear function of the quoted lead-time and price. The available

Table 5.1
Parameters for the baseline scenario.

Demand rate	Holding cost	Lateness penalty cost	Variable cost	Fixed cost	Rejection cost
1/10	1	3	2		
1/20	2	5	1	2	20
1/30	3	2	3		

production capacity in each production period is 80 hour. Type i customer orders arrive following a Poisson process with the mean λ_i. The parameters of baseline case are shown in Table 5.1.

Different scenarios in the MTO system have been evaluated to compare the QLA with the FCFS policy and the heuristic developed based on the order acceptance threshold heuristic proposed by Hing et al. (2007). The policies are evaluated based on the average reward, order acceptance rate, and quantity acceptance rate. Order acceptance rate is the number of accepted orders divided by the total number of arrived orders. Quantity acceptance rate is the number of accepted order quantity divided by the total quantity of arrived orders. The performance of QLA, FCFS and the Heuristic are tested in five scenarios. The results are presented in Table 5.2. The detailed designs and outcomes of the five simulation scenarios are illustrated in the following sections.

5.1. Scenario 1: baseline

In Table 5.1, the input parameter values for this baseline scenario are given. In Fig. 5.1, 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 taking average of ten independent simulation runs. The comparison is made among the QLA, FCFS, and the heuristic. 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.

As shown in Fig. 5.1, it is clear that the QLA outperforms both the Heuristic and FCFS policies on profitability. However, it is quite interesting that the average reward in the learning phase is much greater than that after the learning phase. This phenomenon can be explained by the high order rejection cost and the limited production capacity. At the the beginning of the simulation, almost every arrived order is 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 5.2, the QLA is very selective on orders because it has the lowest order acceptance rate and quantity acceptance rate. Although the QLA has the very low acceptable rate, it generates the highest average reward compared with the Heuristic and FCFS policies, which indicates the superior performance of the QLA in the order selection, pricing and lead-time decisions.

5.2. 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 the same as those in the baseline scenario.

Variations in demand rates are quickly detected by the QLA and it tends to accept more orders. This situation triggers a resetting of the exploration strategy of the QLA, and a new leaning curve is presented in Fig. 5.2. After a learning phase, the QLA outperforms the Heuristic and FCFS, and 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 are reduced slightly. The QLA demonstrates the adaptability on the demand variation. It is worth noting that most of the arrived orders can be accepted when the demand rates are lower. In Table 5.2, the order acceptance rates and quantity acceptance rates for three policies are increased compared to the baseline scenario.

5.3. 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 responsive to the capacity variation. For example, if the capacity is increased, the policy needs to select more orders from the rejection set. In this capacity variation scenario, we increase the available production capacity to 160 hour. The rest of the input parameters are the same as those in baseline scenario. The learning curves are shown in Fig. 5.3.

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

5.4. Scenario 4: single price

Market conditions constantly change order profitability. If the price for each type of order is fixed, the order acceptance and lead-time selection policy should be able to detect and 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, if the price set in baseline scenario is (80,90,100) for order type 1, 80 is chosen as the single price. The rest of the input parameters are the same as these in baseline scenario. The learning curves are shown in Fig. 5.4.

Table 5.2
Comparison results of average reward and acceptance rate earned by the QLA vs. the Heuristic and FCFS for the five scenarios at 95% confidence interval.

Scenario	QLA			Heuristic			FCFS		
	Average reward	Order acceptance rate	Quantity acceptance rate	Average reward	Order acceptance rate	Quantity acceptance rate	Average reward	Order acceptance rate	Quantity acceptance rate
1	61.37 ± 1.81	0.31 ± 0.02	0.60 ± 0.01	54.36 ± 0.18	0.42 ± 0.01	0.89 ± 0.02	52.92 ± 0.23	0.42 ± 0.01	0.89 ± 0.02
2	63.25 ± 1.17	0.53 ± 0.03	0.82 ± 0.03	52.88 ± 0.26	0.78 ± 0.02	0.95 ± 0.02	51.79 ± 0.36	0.79 ± 0.03	0.95 ± 0.02
3	130.54 ± 2.12	0.53 ± 0.03	0.86 ± 0.03	106.17 ± 0.52	0.72 ± 0.02	0.97 ± 0.03	102.69 ± 0.37	0.78 ± 0.01	0.96 ± 0.03
4	58.39 ± 1.19	0.43 ± 0.01	0.77 ± 0.02	40.59 ± 0.48	0.50 ± 0.02	0.92 ± 0.02	38.79 ± 0.6	0.51 ± 0.01	0.91 ± 0.02
5	72.63 ± 1.05	0.52 ± 0.02	0.85 ± 0.02	61.93 ± 0.32	0.69 ± 0.05	0.97 ± 0.03	56.42 ± 0.25	0.70 ± 0.02	0.98 ± 0.02

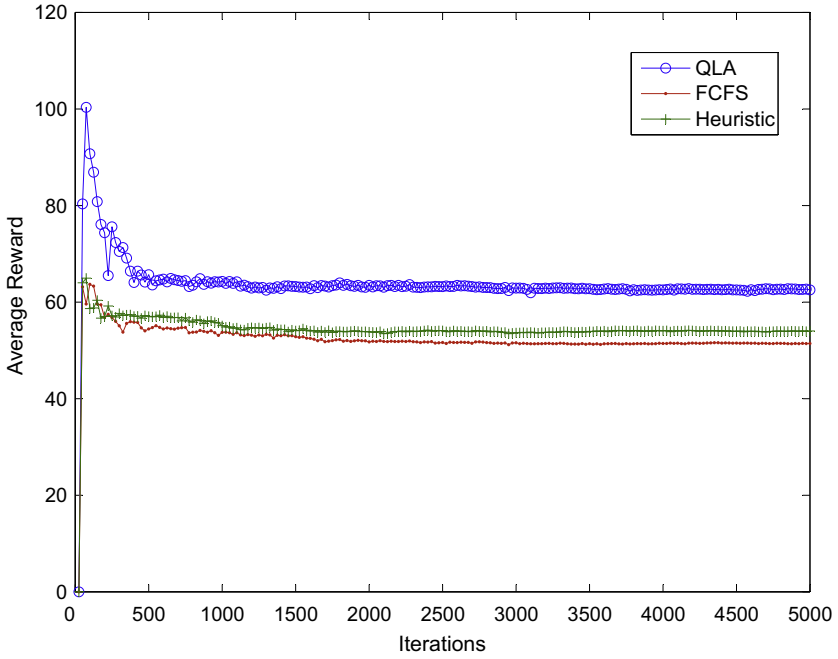

Fig. 5.1. Average reward leaning curves for baseline scenario (QLA compared to FCFS and a threshold heuristic with $b = 6$).

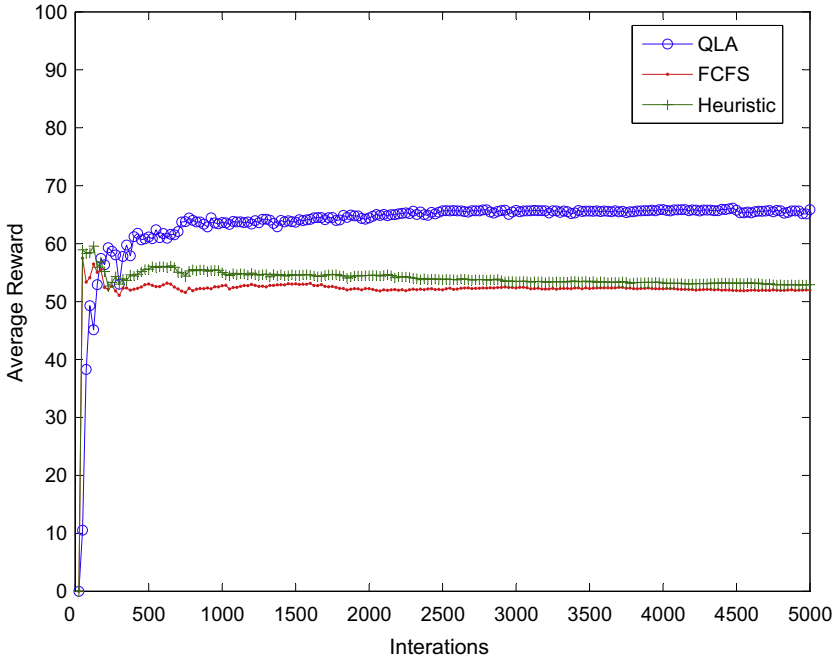

Fig. 5.2. Average reward leaning curves for demand variation scenario (the QLA compared to FCFS and a threshold heuristic with $b = 6$).

As shown in Table 5.2, the change to the single low price causes reduces of the average rewards and increases of the acceptance rates compared with the baseline scenario. Clearly, the QLA consistently outperforms the other two policies with highest average reward and lowest acceptance rate.

5.5. Scenario 5: due-date negotiation

Due-date negotiation is modeled here to offer customers, who postpone their latest acceptable due-dates, some compensation for late delivered orders. In this scenario, the latest acceptable due-date for all orders are generated from Uniform[25,35]. The rest of the input parameters are the same as these in baseline scenario. The learning curves are shown in Fig. 5.5.

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, the Heuristic and FCFS increase by 18%, 14% and 7%, respectively. It is clear that the QLA achieves the highest improvement in profit generation compared with the other two policies. It indicates that the QLA makes better use of

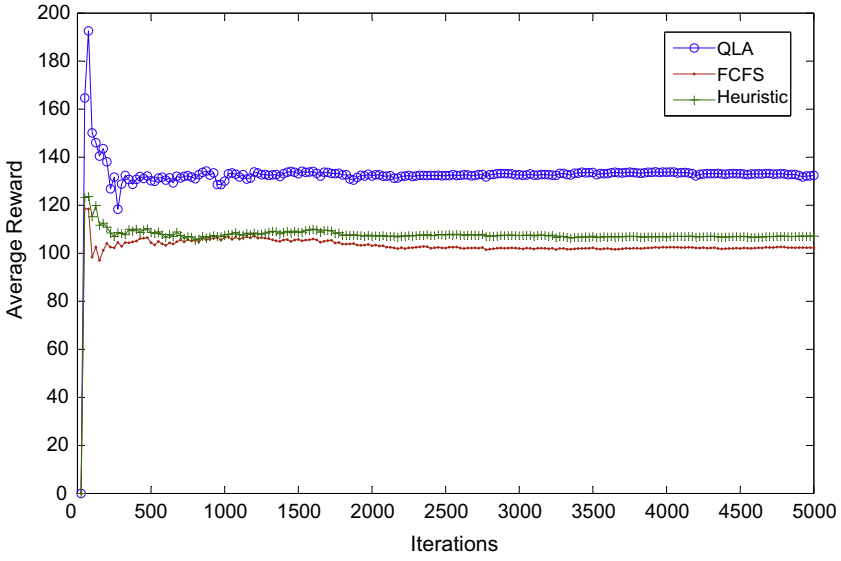

Fig. 5.3. Average reward leaning curves for capacity variation scenario (the QLA compared to FCFS and a threshold heuristic with $b = 6$).

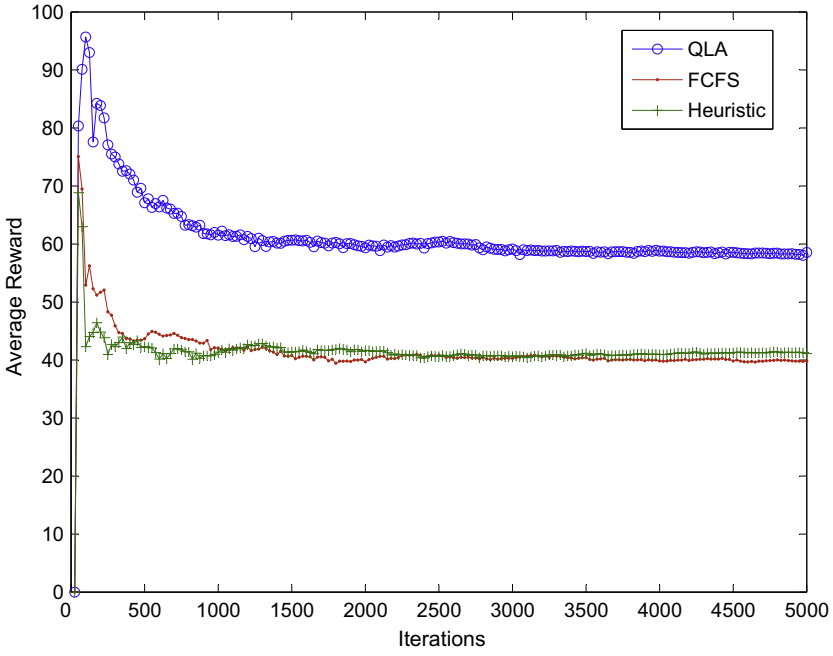

Fig. 5.4. Average reward leaning curves for single price scenario (the QLA compared to FCFS and a threshold heuristic with $b = 6$).

the advantages of the due-date negotiation and generates more profit than the other two policies.

6. Summary and future research

In this paper, we investigate the pricing, lead-time, scheduling and order acceptance decisions in an MTO manufacturing system with stochastic demands. The goal of our problem is to maximize the long-run average reward. We propose a RL-based algorithm, the QLA, to solve this problem. To deal with many sources of uncertainty in an 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 and quantity acceptance rates 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 has been highlighted for the MTO system. Due to the limited production capacity, some orders need to be rejected to save space for the more profitable orders in the near future. The QLA can effectively learn which order is worth accepting, and can distinguish it from orders that need to be rejected. The numerical experiments show that the QLA is not only highly selective on order acceptance but also adaptive to the environmental changes. Therefore, the QLA can be implemented in an MTO manufacturing firm for revenue management.

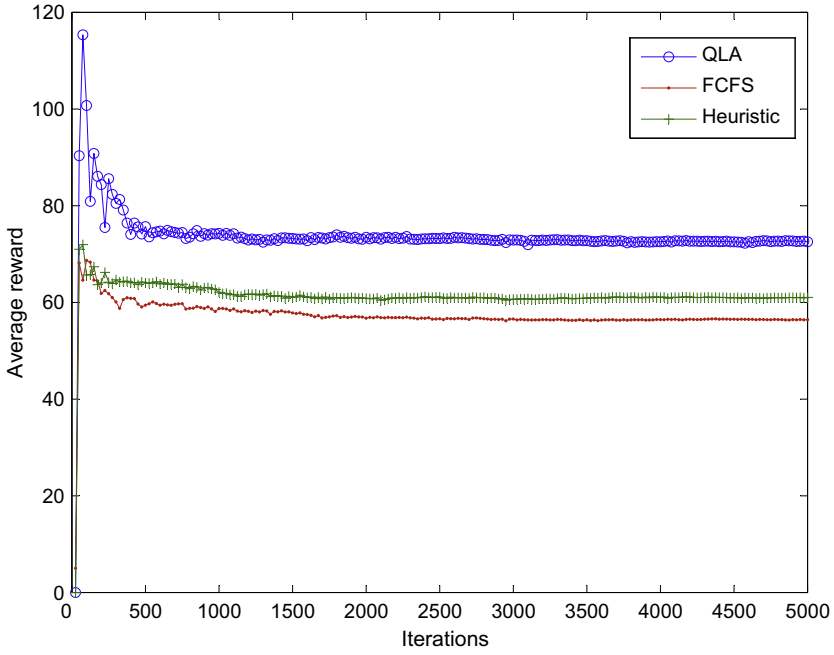

Fig. 5.5. Average reward leaning curves for due-date negotiation scenario (the QLA compared to FCFS and a threshold heuristic with $b = 6$).

Further studies can focus on the application of the RL approach to the multi-resource capacity planning setting. Another issue that could be interesting is to study whether the RL approach is suitable to solve other joint decision problems. For example, the joint optimization on pricing, scheduling and inventory level decisions in a make-to-stock manufacturing system. Furthermore, other neural network approaches to the function approximation can be explored to make better use of the information.

References

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.

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

Charnsirisakskul, K., Griffin, P., Keskinocak, P., 2006. Pricing and scheduling decisions with leadtime flexibility. European Journal of Operational Research 171 (1), 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 (2), 293–303.

Chryssolouris, G., Giannelos, N., Papakostas, N., Mourtzis, D., 2004. Chaos theory in production scheduling. CIRP Annals 53 (1), 381–383.

Darken, C., Moody, J., 1992. Towards fasters stochastic gradient search. In: Moody, J., Hanson, S., Lippmann, R. (Eds.), Advances in Neural Information Processing Systems, vol. 4. 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.

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 make-to-order firms with contingent orders. European Journal of Operational Research 116 (2), 305–318.

Eliashberg, J., Steinberg, R., 1993. Marketing-production joint decision-making. In: Eliashberg, J., Lilien, G.L. (Eds.), Handbooks in Operations Research and Management Science, Vol. 5. North Holland, Amsterdam, pp. 827–880.

Feichtinger, G., Hartl, R., 1985. Optimal pricing and production in an inventory model. European Journal of Operational Research 19 (1), 45–46.

Gilbert, S.M., 1999. Coordination of pricing and multi-period production for constant priced goods. European Journal of Operational Research 114 (2), 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 (2), 224–241.

Hing, M.M., van Harten, A., Schuur, P.C., 2007. Reinforcement learning versus heuristics for order acceptance on a single resource. Journal of Heuristic 13 (2), 167–187.

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

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

Keskinocak, P., Tayur, S., 2004. Due date management policies. In: Simchi-Levi, S., Wu, S.D., Shen, Z.M. (Eds.), Handbook of Quantitative Supply Chain Analysis: Modeling in the E-business Era. Kluwer Academic Publishers, Norwell, MA, pp. 485–553.

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.

Kim, D., Lee, W.J., 1998. Optimal joint pricing and lot sizing with fixed and variable capacity. European Journal of Operational Research 109 (1), 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.

Leibs, S., 2000. Ford heeds the profits. CFO Magazine 16 (9), 33–35.

Lippman, S., Ross, S., 1971. The streetwalkers dilemma: a job shop model. SIAM Journal of Applied Mathematics 20 (3), 336–342.

Mahadevan, S., 1996. Average reward reinforcement learning: foundations, algorithms and empirical results. Machine Learning 22 (1), 159–195.

Mahadevan, S., Theocharous, G., 1998. Optimizing production manufacturing using reinforcement learning. In: Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference. AAAI Press, pp. 372–377.

Mahadevan, S., Khaleeli, N., Marchalleck, N., 1997a. Designing agent controllers using discrete-event markov models. In: Proceeding of the 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. Morgan Kaufmann, San Mateo, CA, pp. 202–210.

Miller, B., 1969. A queueing reward system with several customer classes. Management Science 16 (3), 234–245.
Nicholas, J., 1998. Competitive Manufacturing Management: Montinuous Improvement, Lean Production, Customer-Focused Quality. McGraw-Hill, Inc., Newyork, NY.
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.
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 assinments 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.
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.
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 (1), 9–44.
Szepesvari, C., Littman, M.L., 1996. Generalized markov decision processes: dynamic-programming and reinforcement-learning algorithms. Tech. Rep., Brown University, Providence, RI.
Vanthienen, L.G., 1975. Simultaneous price-repoduction decision making with production adjustment costs. In: TIMS XX International Meeting. pp. 249–254.
Watkins, C., 1989. Learning from delayed rewards. Ph.D. Thesis, University of Cambridge, England.
Webster, S., 2002. Dynamic pricing and lead-time policies. Decision Science 33 (4), 579–599.
Whitin, T., 1955. Inventory control and price theory. Management Science 2 (1), 61–68.
Widrow, B., Hoff, M.E.J., 1960. Adaptive switching circuits. IRE WESCON Convention Record 4, 96–104.
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 (9), 11782–11791.
Yano, C.A., Gilbert, S.M., 2004. Coordinated pricing and production/procurement decisions: a review. In: Chakravarty, A., Eliashberg, J. (Eds.), Managing Business Interfaces: Marketing, Engineering and Manufacturing Perspectives. Kluwer Academic Publishers, Boston, MA.
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.