Bidding Srategies of Automated Agents in Supply Chain Management

Soheil Sibdari*

Xiaoqin Zhang

University of Massachusetts Dartmouth

North Dartmouth, Massachusetts 02747

December 18, 2012

Abstract

The automated agents have profound influences on managing and performing different stages of supply chains. They play important roles in planning and coordinating the activities of organizations across the supply chain from the procurement of raw materials to the delivery negotiation for the finished goods, and the pricing of special service or advise. The use of automated agents is more effective in cases where a routine task with heavy calculations and high frequency needs to be performed, which requires significant human resources such as bidding for raw material or service outsourcing. If planned with sophistication, an automated agent significantly saves the resources of a supply chain and perform a task with low error. In this chapter, we study the bidding strategies automated agents and their impact on the supply chain effectiveness. We first review the literature, and then consider three applications of automated agents in supply chains, namely Energy Marketplace, Grid services, and the TAC SCM game.

keywords: automated agents, auction theory, supply chain management

*Corresponding author: ssibdari@umassd.edu; (508) 999-8019.

1 Introduction

Managing a supply chain consists of planning and coordinating different activities of different organizations within the chain. These activities include, but not limited to, raw material procurement, facility location, job scheduling, production planning, inventory control and finished goods delivery. Due to recent advances in information technology, most of the decisions related to these activities are made by automated agents. These activities include negotiations for raw materials procurement in manufacturers, purchase of the end product, and logistics and transportation offers. As the human interaction is minimal for such important and costly decisions, the decision-making process of these automatic decisions are crucially important. The ability of the automated agent to be able to make decisions under uncertain circumstances is an important feature of the automated agent. In addition, the sophistication level of the algorithms and procedures is another factor to be determined in designing the agent. Since the human judgement is minimal (human verification is still required for specific services), the agent is able to handle sophisticated formulation and algorithm through a computational analysis. On the other hand, as the automated agents need to handle the real world problems, implementing complex procedures that require many approximation and estimation might result in sub-optimal solutions. Having faced all these questions, the literature in automated agents is rich and considers many aspects of this design.

Automated auctions are among the popular trading mechanisms that is widely used in e-commerces, which provide fair and competitive trading situation. The auctions that traditionally were manually conducted are now performed automatically raising the concept of automated agents. Automating these auctions that involve customized products require intelligent agents programmed with different aspects of constraints from the buyers and sellers agents (see David (2000)). There are several methodologies for the automated trading agents to conduct their bidding strategies. The automated agents can forecast the price first and then using a profit maximization model determine their bids. In this method the agent ignores the inherent interaction with the other agents and that assumes that the price is determined or equivalently its bid does not have impact on the final product price. Since this method is a price taker model, it is applicable to markets where the future prices are relatively easy to forecast. In many markets where the future price is not easy to estimate the price taker models do not find many applications. The alternative method to determine the bidding strategy is the game theory approach where each agent moves depending on the predicted action of other agents. In game theory the agents consider the most recent move of their opponent or predict their moves depending on the game setting, the alternating-move games or the simultaneous-move games receptively.

Adding learning ability is another capability that can be added to an automated agents for repeated auctions. This capability increases the agents' potential for practical use specially when the auction repeats frequently. Examples of repeated auctions within a supply chain can be found in Original Equipment Designer (OEDs) and Electronic Manufacturing Service (EMS). A supplier using intelligent agent significantly benefits if the bidding strategy gets updated as new information arrives. Nicolaisen et al. (2001) employ a discriminatory double auction to develop an algorithm for learning agents in electricity markets. Richter and Sheble (1998) use a genetic algorithm simulate a marketplace with multiple generating companies as buyer an a single seller where the seller learns from previous bids. Richter et al. (1999) a combination of genetic algorithms with data structures is used to develop an adaptive strategy model for the single seller.

One area that the automated agents have been widely used are Electronic marketplaces (e-marketplaces) and online procurements. Different sectors including food service companies, retailers, energy marketplace, and wholesalers belong to different e-marketplaces. The formation and level of automation of agents varies depending on the type and nature of the marketplaces. To setup and form a automated system for e-marketplace, different e-marketplaces enable companies to trade their supply chain processes in a large scale. Examples of such web-based trading systems include World Wide Retail Exchange (WWRE) and Global Net Exchange (GNX). Eng (2004) conducts a survey in a UK food supply chain that includes food service companies, retailers, and wholesalers. He shows that the e-marketplace supply chain applications enables the majority of the companies to use an automatic transaction-based activities to conduct transactions rather than using strategic supply chain strategies. The economic impact of e-commerce in business-to-business sectors is significant and estimated six times larger than the business-to-customer sector. Ballou et al. (2000) study the interaction between different members of a supply chain and uses the e-marketplace in the supply chain activities. They show that the use of e-marketplace activities span organizational boundary through upstream and downstream linkages.

Another area that the agent bidding found applications is in the Computational Grid systems where the price resources allocation needs to be determined through a bidding system. The Computational Grid systems provide the necessary infrastructure to dynamically aggregate resources to draw extra computational power from pooling the "generators." The power gets amplified through the resource pooling similar to the way appliances draw electrical power from a set of cooporating and interconnected power utilities. The resource allocation problems through automated agents are widely studied in economic and computing literature through either commodity markets or auctions. Wolski et al. (2001) uses the commodity market approach in order to address the resource allocation problem in Grid. Swany and Wolski (2004) studies the integrating dynamic performance from the Network Weather Service using a Grid setting. They describe the Network Weather Service's system provide the rationale behind the structure of the system. Pourebrahimi et al. (2010) addresses the resource allocation problem using the market-based approach for a local Grid. They use a dynamic pricing strategy for an agent based Grid economy where the decision-making process about the task and resource allocation is distributed among both the resource owners and the users. The pricing is dynamic in the sense that the mismatch between supply and demand is directly addressed into the prices that the users and the resource owners offer. Another approach to determine the resource prices in Grids is through auctions. Nisan et al. (1998) and Waldspurger et al. (1992) use the one-to-many approach (e.g. English auction and Dutch auction) where an agent (a resource owner or a user) creates an auction and let other agents to make a bid. The many-to-many auctions (e.g. double auction) are also been studied by Ogston and Vassiliadis (2002) and Preist and Van Tol (1998) where several agents initiate and auction and let other agents to bid.

For the rest of the paper, we provide three applications where automated agents play important roles on the operations of an organization. The entities involved in such systems along with their tasks and interactions with other entities will be explored in this chapter. We also describe the formation process of the agents and the role of each entity in this format on process. The process of decision making of an automated agent will be addressed including the details of the inputs and outputs associated with this process. In addition, we discuss available methods to evaluate each decision to revise and improve the decision process of the agent. In the last part of the chapter, we will introduce an application of automated agents in supply chain management that appears in simulation game format, namely Trading Agent Competition for Supply

Chain Management (TAC SCM). TAC SCM is an e-commerce simulation application that consists of several manufacturer agents that compete in a reverse auction in order to sell assembled product to a limited number of customers. Due to its applicability, TAC SCM is well studied in literature and many improvements have been made in its performance. In this chapter we explain and review the innovative optimization and simulation techniques that were employed in the advancement of the TAC SCM.

In this chapter, we review the bidding strategies that use the game theoretic approach with different applications including the energy marketplace and the the Trading Automated Competition for Supply Chain Management (TAC SCM).

2 Energy Marketplace

The automated agent bidding is being used extensively in the electricity markets where the bidding strategies are crucially important due to high level of transactions. The automated agents submit their bids to the opponent generation companies (GENCOs, herein *agents*) based on their own cost and the estimation of the opponents' bids. The agents use different methods to develop their bidding strategies including the price-taker strategy. They can assume that the future market clearing price is possible to estimate and then, using this market clearing price, they apply a profit maximization procedure to submit their bids. As the electricity markets are not in perfect competition situation, the price-taker methods do not usually fit and and therefore, the game theoretic approach are better fit for this market.

As the price-setting models do not fit, the agents in electricity market use the *reverse auctions* to calculate their bids. Bandyopadhyay et. al. ((2006) and (2005)) consider an auction problem when two identical sellers with production capacity k and variable costs c seek business from a single buyer with reserve price r while the total demand is Q. To avoid trivial solutions, they assume that $Q > k$ and $2k > Q$. Among the sellers, the one with lowest bid sells at its capacity and the residual demand is fulfilled by the higher bid seller. They determined the prices at which the sellers are willing to sell their products is given by:

$$p = c + \frac{(r - c)(Q - k)}{k} \tag{1}$$

Bandyopadhyay et. al. ((2006) and (2005)) showed that there is no Nash equilibrium in pure strategies for this problem and instead they provide the following probability density function over the price range that addresses the mixed strategy equilibrium.

$$F(g) = \frac{(g-c)k - (r-c)(Q-k)}{(g-c)(2k-Q)} \tag{2}$$

and that the Nash payoff is:

$$\text{Nash payoff} = \frac{(r-c)(r-p)(Q-k)^2}{(2k-Q)(p-c)} \tag{3}$$

Sikora and Sachdev (2008) modified the above model by studying the learning strategies of rationally bounded agents. They study the seller's learning process of the bidding strategy and since the sellers have limited information about other sellers', the bidders' best response function falls in a continuous set (in the above formulation falls in $[c, r]$). Having the continuous set for bidding strategy makes the problem intractable and therefore Sikora and Sachdev (2008) used price bands by splitting the effective price range $[c, r]$ into n equally sized bands. Using this transformation, the problem becomes a discrete two-player symmetric game with n different strategies where the sellers are represented as row and column player with associated action. In this setting, it is straightforward to calculate the values of the non-diagonal elements as the one seller bids in lower range than the other seller. The expected profit of seller x bidding in the price band i against seller y that bids in price band i, $a_{i,i}$ can be calculated as follows.

$$a_{i,i} = \int_{g=l_i}^{u_i} \int_{h=l_i}^{g} \frac{(Q-K)(g-c)}{(u_i - l_i)^2} dgdh + \int_{g=l_i}^{u_i} \int_{h=g}^{u_i} \frac{K(g-c)}{(u_i-l_i)^2} dgdh \tag{4}$$

where $[l_i, u_i]$ is the price band i. Using the above expected function for each player they assigned a probability distribution to each action and found the Nash equilibrium in mixed strategies and calculated the value of the unique Nash equilibrium for a two-player game with 10 price band using numerical studies.

An interesting analysis of the Sikora and Sachdev (2008) is the focus of the paper on the dynamics of the results rather than the statics of the Nash equilibrium. They studied the equilibrium concept by exploring how the agents arrive at such an equilibrium specially under the bounded rationality assumption. The concept of stability is used for the equilibrium point in a way that an equilibrium is said to be stable if when the system moves from the stable point, it tends to return to the stability and do not keep moving form the equilibrium. To this end, they used the Liapunov stability in the case of bounded rationality and used numerical study to show this stability. Sikora and Sachdev (2008) show that the Nash equilibrium in their situation is not stable in the case where a rationally bounded agent tries to maximize its payoff by learning from past payoffs. In this situation, they used an agent-based modeling to provide a natural setting for the dynamic of the game with the aforementioned desired properties of rational behavior and convergence in order to evaluate the agents' performance. They measured the effectiveness of the agents by using the Nash payoff as the benchmark.

Sikora and Sachdev (2008) also addressed the learning behavior of a simple evolutionary and reinforcement agent using Genetic Algorithms (GA). The GA forms a search algorithm by combining the survival of the fittest among string structure with a structured and randomized information exchange. In this paper the GA considers the population of bidders and implements a mixed strategy for an individual member of this population. Therefore, for each member of this population a vector of probabilities is formed for each action that adds up to one. Using numerical techniques, they let each strategy to play against other strategies for a finite number of times and they calculate the fitness of each strategy as the average payoff from all one-to-one games. Using this average payoff for each strategy, they used the Reinforcement learning (RL) method in order to select the action that maximizes the future payoffs. This method is based on estimating action-values that is the estimated reward for taking each action. In this problem each action refers to the price band to bid and the values are referred to the weighted average of past reward with the most recent values receiving more weights. In this paper, they used different action-selection methods that the RL agent can choose in deciding the next action. These methods assign a probability distribution over the actions based on the estimated values. Using this method the agent learns about its bidding strategy by evaluating previous actions.

Another approach in in emphasizing the dynamics of strategy changes is through Evolutionary Game Theory (EGT). EGT takes into account the personal knowledge, risk factors, and market perception of the agents in finding and characterizing the equilibrium. It differs than the traditional game theory by considering the dynamics of the agents' learning abilities that is more appealing in practice. EGT is widely studied in economics and computer science literature and is being applied to analyze various gaming behavior such as firm and industry, economic growth theory, and dynamic systems (see Smith (1982) and Axerlrod (1997)).

Wang et. al. (2011) employed an evolutionary game theory to model the agent bidding strategy in the electricity market when the agents' information is incomplete. This paper considers the adaptive behavior of the agents where the agents belief about their rivals bidding strategies and probabilities get update as new information arrives.

By applying evolutionary game theory, in each round, the agents make the bidding decisions based on their own updated strategy that is affected by their updated belief on opponents' bidding strategy. They also studied the concept of evolutionary stable strategies (ESS) in this game, where ESS arises if it performs better than any new and invading strategy. In their model setting, a strategy is evolutionary stable if not agent can increase its profit by either changing its belief or own bidding strategy.

In the context of Genetic Algorithm (GA), Wang et. al. (2011) used a coevolutionary approach by encoding the optimal bidding strategies as the representative of each generator (i.e. bidder). In each generation, every agent uses the standard GA to choose its best bidding strategy for a given bid of the opponent. To calculate the fitness value, a standard genetic operations such as reproduction, crossover and mutation is used and all the possible strategies for the bidder is encoded as the GA individuals. In order to decode the other opponent bidding strategy, this paper uses the online decoding method that is to update the opponents' strategies immediately after they finish with their respective evolution. The alternative method is to decode the other agents' bidding strategies after all the agents finish evolving in tho generation. Finally the agents' profits are calculated after all the bids are submitted and market is clear.

3 Grid Services

Another application where automated bidding is employed is the allocation of Grid services. A Grid is a distributed system that includes multiple number of computers that are connected through a fast network that share different devices in order to facilitate large scale computing. The devices and software packages are shared in a network to reach a common goal or to reach a single task and therefore the Grid computing is cost efficient and therefore the allocation of tasks and resources requires an allocation mechanism. A demand modeling and economic modeling tools in being used to determine the pricing of the requested time-slot. The customers describe their economics preferences including the valuation of required services, the expiration date of a bid, and the preferred bidding strategy. On the other hand the provider determines the mechanisms under which the bidding are submitted and the spaced offers are allocated. This include the pricing policy that indicated whether the customers get charged for the time-slot or for the amount of space usage. Finally, the description of bids offered by both the customers and the providers becomes available in the market to initiate transactions. All these processes take place autonomously using different generators of the system.

Borissov and Wirstrom (2008) presented a bidding strategy to implement artificial bidding agents that support the customers and the providers in the Grid computing business. In addition, they also provided a bidding strategy for the customer side that enables the strategic customers to submit their service requests and select the right service provider. They named this policy as Q-strategy and used two other mechanisms namely *Truth-Telling* and the *Zero Intelligence Plus* strategies to evaluate their performances.

The Truth-Telling strategy agents are myopic methods and only consider the current situation as they do not remember the outcome of earlier market interactions. The agents report their true type to the system and choose the machine that be maximize their immediate utility. This bidding strategy places a bid equal to the either provider or the customer true valuation of a certain service. The outcome of this policy does not depend on the policy of other agents and guarantees obtaining optimal payoffs. However, it is been shown that this policy is not dominant in the budget-balanced double auction mechanism (Phelps (2007)).

In the Zero-Intelligences Plus (ZIP) the bids are generated using a reinforcement learning method to learn the price of a particular market. Using this method, the rule for updating the profit margins gets updates depending on the difference between the agent's valuation and the generated bid. This method updates the profit margins based on whether the last event was an offer or a bid and whether the agent is an active agent and has service to sell or buy. Using an experiment, Das et. al. (2001) showed that ZIP agents perform better than human trader bidding.

The Q-Strategy develops rational agents with learning capabilities that may report false information about its true valuation based on previous experiences. This strategy consists of two algorithms where the first one describes the case where an agent generates a bid for the desired configuration service and the second one refers to the case where an agent receives many offers for a given configuration and select the right one. Both algorithms are based on a reinforcing learning approach that employs a ϵ-greedy selection policy, which selects an action with probability ϵ and exploits its obtained knowledge with probability $1-\epsilon$. The objective of the agent is to learn function $Q(s, a)$ that is its expected value of being in state s and taking action a, where s refers to job specific requirement. The agent desires that given any job specification such as tech-

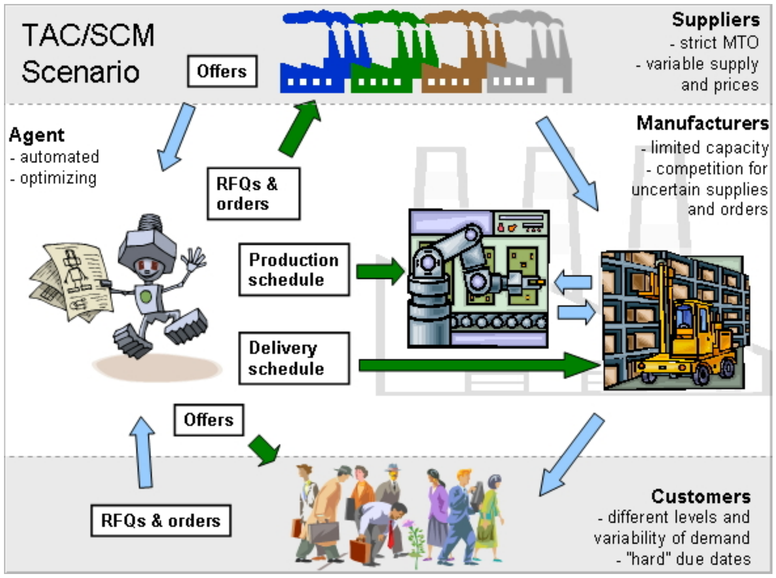

Figure 1: TAC SCM Game Scenario

nical requirements, duration, and valuation, a price can be selected so that the utility is maximized. Das et. al. (2001) showed that the reinforcement learning algorithms learn the environment and converge to an optimal action.

Pourebrahimi et al. (2010) studiy the allocation of the Grid services using a dynamic pricing strategy and implemented it for a market-based resource allocation mechanism in a local Grid. A Continuous Double Auction is used as a matchmaking mechanism for the consumers and producers where the decision-making process is distributed across both the users and the resource owners. Using the local Grid experiment, they described how the consumer and producer agent can influence the process of assigning the resources.

4 The TAC SCM Game

The Trading Agent Competition Supply Chain Management (TAC SCM) game simulates a real world supply chain management scenario. There are three type of agents in the game: customers, manufacturers and suppliers, as shown in Figure 1(except from (TAC SCM n.d.)). Each game consists 220 simulated days, 15 seconds for each day. Six manufacturer agents compete against each other and the agent with the most money at the end of the game wins.

Customers order PCs from the manufacturers starting with sending RFQs (Request For Quote) to all manufacturers. The RFQ consists of the following information.

1. **PC type** There are 16 types of PCs, which fall into three market ranges, namely low, medium and high range.
2. **Quantity** The number of PC units ordered.
3. **Reserve price** The maximum unit price.
4. **Due date** The date by then the orders must be shipped to the customer.
5. **Penalty amount** The amount the manufacturer must pay per day for late delivery.

A **manufacturer** agent produces PCs ordered by customers. It buys components from suppliers, sends production requests to the factory and then delivers finished PCs to customers. The manufacturer receives multiple RFQs from customers every day. For each RFQ, the manufacturer decides whether to bid for it; if so, what the bidding price would be. If the customer is satisfied with the bidding price, it places an order with the manufacturer agent. When there are multiple bids for an RFQ, the lowest bid is selected by the customer. Should the manufacturer gets selected by the customer, it analyzes what types of components and how many of them are needed to fulfill those orders. With the components in inventory considered, it then creates RFQs for component suppliers for additional components needed. The supplier sends a bid for corresponding RFQ from the manufacturer agents. The manufacturer places an order for the components if it is satisfied with the offer. After receiving the components, the manufacturer provides a production and delivery schedule to the factory. The completed PCs are shipped to the customer according to the delivery schedule. Each manufacturer owns a factory and there are 2000 production cycles per day. The products are only manufactured if there are sufficient components in inventory. Manufacturer agent can store both components and finished PCs in a a warehouse by paying a daily storage fee.

Suppliers produce the components required to build a PC. They are revenue-maximizing agents and they work on make-to-order basis. Each supplier has a fixed production capacity. When the supplier receives an RFQ from the manufacturer, it checks if it can offer a price less than the reserve price. The supplier can also offer a reduced quantity and negotiate on the due-date. Next we will review three approaches for finding the bidding strategy for the manufacturer agent in TAC SCM game.

4.1 Constrained Based Optimization with Learning

Burke and Brown (2005) presented a constraint based agent for TAC-SCM problem. They viewed the game as three major decision problems for the agent: what offers should be made to the customers? what offers should be made to the supplies? and how to schedule the production? In this work, rather simple approaches are adopted for the later two problems and the focus is to solve the first problem with constraint-based approach combined with learning.

Production Scheduling Productions are only scheduled for confirmed orders, which are sorted by due dates. Production schedule is limited by the availability of components and production capabilities. Any order that cannot be processed in current day are scheduled on the day before their due date.

Ordering Components form Suppliers The agent orders components from suppliers in advance so that the cost of components are available when deciding the offers to customers. The amount of each type of component to order is based on the expected orders that is adjusted when the actual orders are available.

Making Offers to Customers The agent receives multiple requests for quotes (RFQ) from customers each day and it needs to decide for each RFQ, whether it should bid, that is to make an offer ($bid_r \in \{0, 1\}$), and if so, what is the bidding price ($price_r$). A fixed price for each product is assumed first to make the problem is easier to solve. Now the only decision to make is on the decision whether to bid bid_r.

Two constraints are considered in making the bidding decision: component availability and the production capability. A binary matrix is employed to model the bill of material (BOM) that describes what components are needed for each product. $bom_{i,j} = 1$ indicates that component j is used for product i. Because the components are ordered in advance, and it is known how much components will arrive and when. In the following formula, that is excepted from (2005), the availability constraint is modeled.

$$\sum_{k=1}^{t} comp_{k,j} \leq \sum_{k=1}^{t} inv_{k,j}, \text{ where } comp_{t,j} = \sum_{i=1}^{n} prod_{t,i} \times bom_{i,j} \tag{5}$$

here $comp_{t,j}$ represents the number of component j used at time t, $inv_{t,j}$ represents the number of component j's inventory at time t, and $prod_{t,i}$ is the number of product i produced on day t, which is calculated based on the quantity specified in RFQs and the schedules of RFQs:

$$prod_{t,i} = \sum quantity_r \times bid_r, \forall RFQ\ r \,\text{where}\, product_r = i, sch_r = t \tag{6}$$

The production capability constraint is represented as:

$$\sum_{i=1}^{n} prod_{t,i} \times cycles_i \leq cap_t \tag{7}$$

Each type of product i requires a certain number of production cycles, $cycles_i$, the total number of required production cycles is limited by the factory capability at day t that is cap_t. The objective function is to maximize the profit, which is the sum of profit for each RFQ r:

$$profit = \sum_{r=1}^{m} profit_r \times bid_r \tag{8}$$

where the profit of each REQ, $profit_r$, is calculated based on the price in the bid $price_r$, component cost and potential delay penalty depending on the scheduling production time sch_r and the due date.

However, not each bid for RFQ will be accepted by customer, and the probability of acceptance depends on the bidding price $price_r$ and the market situation. To ensure a certain probability of bid acceptance, an online learning approach is proposed. Initially, the bidding price is calculated as a default weight w times the based price, which is provided by the game designer for each type of product. The agent keeps track of the actual acceptance rate a/o (the number of accepted bids a over the total number of bids made by this agent o). If this ratio a/o is greater than the target, then weight w is increased, otherwise, w is decreased. A decreasing step-size factor is used for updating w to allow quick learning in the beginning of the game and then converging to a stable weight value. Using the constraint optimization model makes it possible to adopt the existing methods of solving such problem. However, the bidding price decision problem was not addressed in this approach. Moreover, the learning of the probability of acceptance also affects the agent's performance. Before the learning coverages, the agent has to make decision based on in-perfect information.

4.2 Marginal Bidding Strategy using Equimarginal Principle

Greenwald et. al. (2007) developed a greedy solution named marginal bidding strategy. Based on the law of Diminishing Marginal Returns, in order to maximize utility, the limited resource should be allocated among two or more independent users with equal expected marginal return from each (Mas-Colell, Whinston and Green. 1995). This Equimarginal Principle applies to the agent bidding problem in TAC-SCM since the agent is looking to maximize its utility when allocating its limited production capability to different RFQs.

The set of RFQs are partitioned as market segments according to the SKU type and due date. To simply the problem, a price-probability model is used to predict the probability of wining a bid, $p(x)$ at price x: $p(x) = \frac{2200-x}{800}$ for $1400 \leq x \leq 2200$. A bid for a RFQ with quantity q and winning probability p is approximated as winning a partial order with quantity pq deterministically.

For RFQs in segment i, a price-quantity model $h_i(x_i)$ maps the bid price x_i into the quantities that is called *expected market shares*. The bidding problem for each RFQ is then converted as an expected bidding problem: selecting bidding price (and hence the quantity) for each market segment so as to maximize the overall profit with the constrain of the total available production capacity C:

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} h_i(x_i)x_i \textbf{ s.t. } \sum_{i=1}^{n} c_i h_i(x_i) \leq C \tag{9}$$

Assuming the price-quantity model $h_i(x_i)$ is invertible, $h_i^{-1}(x_i')$ maps the quantity (market share) x_i' for segment i into the corresponding bidding price, then this problem is converted to a Nonlinear Knapsack Problem (NLK) with the value function $f_i = x_i' h_i^{-1}(x_i')$ and the cost function $g_i = c_i x_i'$:

$$\max_{x_1',\ldots,x_n'} \sum_{i=1}^{n} x_i' h_i^{-1}(x_i') \textbf{ s.t. } \sum_{i=1}^{n} c_i x_i' \leq C \tag{10}$$

The knapsacks capacity is the factorys capacity, and the objective is to choose the market share for each segment in order to maximize the total utility. A list of unit marginal returns in each market segment is created based on the unit cost c_i and the quantity-price model $h_i^{-1}(x_i')$. A greedy solution is developed by applying the equimarginal principle.

The above 1-day greedy solution is extended for a multi-day problem by adding an additional parameter bidder's *window size* W. With window size as W (number of days), the bidder will consider today's RFQ and the anticipated RFQs in the future $W - 1$ days. A heuristic Marginal Bidding algorithm is developed for the decision-making within the W-day window. First, it greedily fulfills outstanding orders according to a nonincreasing order of revenue per production cycle; then it greedily schedules production of units of different market segments according to a nonincreasing order of unit marginal returns. For each market segment, the agent bids the price associated with the quantity of demand.

There are three limitations for this approach. First, the component constraints are not considered in this algorithm, which are important in the real TAC-SCM game setting. Secondly, the component procurement and pricing problem are not considered either, which may affect the bidding decision because the revenue depends on both the bidding price and the component prices. Thirdly, the greedy algorithm cannot be scaled to handle the entire game length, so the decision has to be made for a limited time window instead of finding a long-term solution for the whole game.

4.3 Finding Bidding Strategy Using Dynamic Programming

Sibdari et al. (2012) proposed using a dynamic programming method to find the agent's optimal bidding strategy. The problem is modeled as follows. Consider an agent who produces a finite number of PC products. To assemble a PC, depending on the product type, the agent needs to spend a specific amount of time called *production cycle*. The total game duration can be divided into equal intervals (say T time periods) such that at most one RFQ can be received in each period. Each RFQ_i, consists of *producID*, *quantity*, *reservePrice*, *dueDate*, and *penalty*. The RFQs are categorized as *RFQ type* that differentiates between RFQs by their productIDs, reserve prices, due dates, and penalty costs. In total M RFQ types are identified. P_i is the probability of receiving a RFQ of type i in each period. At the beginning of each period, upon a RFQ arrival the agent should decide whether or not to bid and if decides to bid at what price. A dynamic programming method is used to solve this problem in order to maximize its total expected profit from period t until period T.

They use historical experimental data to estimate the profit r_i that can be generated from a RFQ of type i. The number of production cycles c_{ij} that are needed to assemble j units of PCs as specified in RFQ type i is also estimated based on historical data. The probability of bid acceptance by customers for RFQ of type i is modeled as a function of the bid price x made by the agent, $g_i(x)$. The bidding price x is determined using a heuristic function that considering the average product price from the market report, hence $g_i(x)$ can be simplified as g_i.

$J(c,t)$ is defined as the agent's expected profit at beginning of period t when its production capacity is c. The following dynamic program equation is used to calculate the optimal decision of whether to bid or to ignore a RFQ of type i.

$$J(c,t) = \sum_{i=1}^{M}\sum_{j=1}^{K} P_i * Q_{ij} * max(J(c,t+1), g_i*(j*f_i+J(c-c_{ij},t+1))+(1-g_i)*J(c,t+1)) \tag{11}$$

with boundary conditions of $J(0,t) = J(c,T) = 0$ for all values of $c \geq 0$ and $0 \leq t \leq T$, and Q_{ij} is the probability of requesting j units of products in a RFQ of type i.

In order to maximize its expected profit, an agent with c available production cycles at time t should accept a RFQ of type i if the profit it makes from this RFQ plus the maximum profit it can make at time $t+1$ with the rest of production cycles after satisfying the requests in this RFQ is greater than the maximum profit it can make at time $t+1$ with c available production cycles. Otherwise, the agent should not bid for this RFQ. In theory the solution brought out by this approach is optimal, however, it subjects to the following limitations. First, the values for P_i, Q_{ij}, g_i and f_i were all gathered from the past experiments. Since the market situation is dependent on the participating manufacturer agents and their strategies, so the data collecting from the past experiments may different from the game that the agent will participate. Secondly, the inventory and storage costs were assumed to be zero, hence the solution may not be optimal in a real game considering the storage cost.

4.4 Summary

In this section, we introduced TAC-SCM game and reviewed three bidding strategies using different models. The original TAC-SCM bidding problem is a very complicated problem because it interleaves with several related problems: scheduling the production and purchasing components. Uncertainty of the market situation makes the problem even harder. Hence all three approaches have made simplified assumptions to make the problem tractable. Learning also has been utilized in different formats to deal with uncertainties. The first approach models the problem as a constraint satisfaction problem with the objective to maximize the profit. Online learning is used to find the relationship between the probability of bid acceptance and the bidding price. The bidding price is adjusted to maintain a targeted acceptance probability. The second approach models the problem as a Nonlinear Knapsack Problem (NLK) and solves it with a greedy algorithm. Component constraints and pricing problem are not considered, and the greedy algorithm can only work for a limited time window rather than the whole game length. The third approach uses a dynamic programing model and finds the optimal strategy for the whole game length window, with learning necessary parameter values from previous experimental data. The difference between the learning environment and the execution environment affects the agent's performance.

TAC-SCM is actually a multi-agent environment, the strategy of each participating agent affects other agent's performance. None of the above approach has model for other agents' strategy, which is a challenging issue. It is also hard to compare the performance of these three approaches since no direct competition has been conducted among the agents implementing these approaches. It will be interesting to observe such competition. In real supply chain problems with large number of players, modeling individual competitors would be infeasible, so simplified assumptions and model of market are needed for making rational biding decisions. All three approaches can be applied to real bidding problems with modifying their assumptions to fit the real problem and learning from the real execution environment.

References

Axelrod, R. 1997, *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*, Princeton University Press, Princeton, NJ.

Ballou, R. H., Gilbert, S. M. and Mukherjee, A. 2000, New managerial challenges from supply chain opportunities, *Industrial Marketing Management* **29**(1), 7–18.

Bandyopadhyay, S., Barron, J. and Chaturvedi, A. 2005, Competition among sellers in online exchanges, *Information Systems Research* **16**(1), 47–60.

Bandyopadhyay, S., Rees, J. and Barron, J. 2006, Simulating sellers in online exchanges, *Decision Support Systems* **41**(2), 500–513.

Borissov, N. and Wirstrom, N. 2008, Q-strategy: A bidding strategy for market-based allocation of grid services, *Grid computing, high-performAnce and Distributed Applications (GADA'08*.

Burke, D. and Brown, K. 2005, A constraint based agent for tac–scm, *in* P. Beek (ed.), *Principles and Practice of Constraint Programming - CP 2005*, Vol. 3709 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 839–839.

David, A. and Wen, F. 2000, Building optimal bidding strategies for competitive building optimal bidding strategies for competitive power suppliers, *Proceedings of the VII Symposium of Specialists in Electric Operational and Expansion Planning (VII SEPOPE)*.

Eng, T.-Y. 2004, The role of e-marketplaces in supply chain management, *Industrial Marketing Management* **33**, 97–105.

Greenwald, A. R., Naroditskiy, V., Odean, T., Ramirez, M., Sodomka, E., Zimmerman, J. and Cutler, C. 2007, Marginal bidding: An application of the equimarginal principle to bidding in tac scm, *AMEC/TADA'07*, pp. 217–239.

Mas-Colell, A., Whinston, M. and Green., J. 1995, *Microeconomic Theory*, Oxford University Press, New York.

Nicolaisen, J., Petrov, V. and Tesfatsion, L. 2001, Market power and efficiency in a computational electricity market with discriminatory double-auction, *IEEE Transactions on Evolutionary Computation* **5**(5), 504–523.

Nisan, N., London, S., Regev, O. and Camiel, N. 1998, Globally distributed computation over the internet - the popcorn project, *in* I. C. Society (ed.), *In ICDCS '98: Proceedings of the The 18th International Conference on Distributed Computing Systems*, p. 592.

Ogston, E. and Vassiliadis, S. 2002, A peer-to-peer agent auction, *In Proceedings of the first international joint conference on Autonomous agents and multiagent systems Part I*, pp. 151–159.

Phelps, S. 2007, *Evolutionary mechanism design*, Ph.D. Thesis, Liverpool.

Pourebrahimi, B., Bertels, K., Vassiliadis, S. and Alima, L. O. 2010, *A Dynamic Pricing and Bidding Strategy for Autonomous Agents in Grids (a book chapter in: Agents and Peer-to-Peer Computing)*, Springer-Verlag Berlin, Heidelberg.

Preist, M. and Tol, C. V. 1998, Adaptive agents in a persistent shout double auction, *in* 11-17 (ed.), *In Proc. of 1st International Conference on the Internet Computing and Economics*.

Richter, C. and Sheble, G. 1998, Genetic algorithm evolution of utility bidding strategies for the competitive marketplace, *IEEE Transactions on Power IEEE Transactions on Power Systems* **13**(1), 256–261.

Richter, C., Sheble, G. and Ashlock, D. 1999, Comprehensive bidding strategies with genetic programming finite state automata, *IEEE Transactions on Power Systems* **14**, 1207–1212.

Sibdari, S., Zhang, X. S. and Singh, S. 2012, *International Journal of Operational Research (IJOR)* **14**(2), 121–134.

Sikora, R. and Sachdev, V. 2008, Learning bidding strategies with autonomous agents in environments with unstable equilibrium, *Decision Support Systems* **46**(1), 101–114.

Smith, M. 1982, *Evolution and the Theory of Games*, Cambridge University Press, Cambridge.

Swany, M. and Wolski, R. 2004, Building performance topologies for computational grids, *International Journal of High Performance Computing Applications* **18**(2), 255–265.

TAC SCM n.d., TAC SCM game description, http://www.sics.se/tac/page.php?id=13.

Waldspurger, C. A., Hogg, T., Huberman, B. A., Kephart, J. O. and Stornetta, W. S. 1992, Spawn: A distributed computational economy, *Software Engineering* **18**(2), 103–117.

Wang, J., Zhou, Z. and Botterud, A. 2011, An evolutionary game approach to analyzing bidding strategies in electricity markets with elastic demand, *Energy* **36**(5).

Wolski, R., Plank, J., Brevik, J. and Bryan, T. 2001, G-commerce: Market formulations controlling resource allocation on the computational grid, *In Proc. International parallel and Distributed Processing Symposium (IPDPS).*