Marginal Bidding: An Application of the Equimarginal Principle to Bidding in TAC SCM

Tyler Odean, Victor Naroditskiy, Amy Greenwald, and John Donaldson
Department of Computer Science, Brown University, Box 1910, Providence, RI 02912
{todean,vnarodit,amy,jwd}@cs.brown.edu

Abstract

We present a fast and effective bidding strategy for the Trading Agent Competition in Supply Chain Management (TAC SCM). In TAC SCM, manufacturers compete to procure computer parts from suppliers, and then sell assembled computers to customers in reverse auctions. To address the bidding problem, an agent decides how many computers to sell and at what prices to sell them. We propose a greedy solution, Marginal Bidding, inspired by the Equimarginal Principle, which states that revenue is maximized among possible uses of a resource when the return on the last unit of the resource is the same across all areas of use. We show experimentally that Marginal Bidding performs as well as a computationally intensive integer linear programming approach on small problem instances. Moreover, unlike our ILP solution, Marginal Bidding can cope with large problem instances. Hence, it can incorporate Lookahead, that is, it can effectively reason about predicted future as well as current demand.

Introduction

A supply chain is a network of autonomous entities engaged in *procurement* of raw materials, *manufacturing*—converting raw materials into finished products—and *distribution* of finished products. The Trading Agent Competition in Supply Chain Management (TAC SCM) is a simulated computer manufacturing scenario in which software agents operate a dynamic supply chain (Arunachalam & Sadeh 2005). We study the TAC SCM bidding problem, which is to decide upon prices at which to offer to sell computers to customers, balancing the tradeoff between maximizing revenue—by placing high bids—and maximizing the quantity of customer orders secured—by placing low bids, within the constraints of component availability and production capacity.

In a dynamic market setting such as TAC SCM there are often conditions under which the optimal bidding/production decisions are greatly influenced by future demand. For example, in an accelerating market it may be worth reserving factory capacity for future, more profitable demand. Conversely, in a bear market it may be optimal to bid more aggressively early on, claiming a larger share of today's demand to fulfill with future production. As such, incorporating information about predicted future demand can positively impact revenues. However, doing so increases the size of bidding problem, and hence the computational resources necessary to solve it. This is true even in an idealized setting where future demand is known with certainty. But in reality, future demand is uncertain, and this stochasticity further increases the computational resources necessary to make effective bidding decisions.

In this paper, we precisely formulate bidding in TAC SCM as a recursive stochastic program, and we propose two heuristic solutions: 1. Marginal Bidding, a greedy algorithm that is motivated by the Equimarginal Principle; 2. a computationally intensive integer linear programming (ILP) solution to "expected bidding," a deterministic approximation of the (stochastic) TAC SCM bidding problem. These heuristics are compared both with and without Lookahead, that is, predicted future demand. We show experimentally that Marginal Bidding performs as well as our ILP approach on small problem instances. Moreover, Marginal Bidding can cope with large problem instances. Hence, it can effectively reason about predicted as well as current demand.

This paper is organized as follows. First we describe the Equimarginal Principle of marginal utility theory, originally posited in the mid 1800's. We note that this principle applies to a generalization of the classic knapsack problem, the so-called *nonlinear knapsack problem* (NLK), and we note that under certain conditions a greedy algorithm can approximate an optimal solution to NLK. Then we formalize the TAC SCM bidding problem as a stochastic program, and argue that expected bidding, a deterministic approximation, is an instance of NLK. Next we describe our key idea—Marginal Bidding—an application of the aforementioned greedy approach to expected bidding. Finally, we compare experimentally the performance of two heuristics, Marginal Bidding and an ILP approach, in simulations of the TAC SCM bidding problem.

The Equimarginal Principle

The Prussian economist H. H. Gossen is credited with observing two fundamental laws of utility: the Law of Diminishing Marginal Returns:

> The amount of any pleasure is steadily decreasing as we continue until the last saturation is reached.

and the Equimarginal Principle:

> If a man is free to choose among several pleasures but has not time to afford them all to their full extent, then in order to maximize the sum of his pleasures he must engage in them all to at least some extent before enjoying the largest one fully, so that the amount of each pleasure is the same at the moment when it is stopped; and this however different the absolute magnitude of the various pleasures may be.

The Equimarginal Principle applies to problems where a limited resource needs to be distributed among a set of independent possible uses. Such problems are ubiquitous. Two problems commonly cited in economics textbooks include: a consumer allocating her income among different commodities to maximize her utility; and a firm deciding how to proportion its labor and capital to reach its desired output.

The Equimarginal Principle states that total value is maximized when marginal values per unit of resource—"marginal value densities"—are equated across all areas of use: $MV_1/P_1 = \ldots = MV_i/P_i = \ldots = MV_n/P_n$, where MV_i is the marginal value of i and P_i is the amount of resource required for one unit of i (e.g., a price). The principle relies on the assumption that the marginal values associated with each use i are decreasing in the amount of the resource allocated.

It is easy to see that in an optimal solution to such a resource allocation problem, marginal value densities are equal. Indeed, if the marginal value densities were unequal, a better allocation could be achieved by redistributing a unit of the resource from the use with a lower marginal value density to the use with a higher marginal value density. Gossen's claim is less obvious: that equal marginal value densities imply an optimal solution, assuming diminishing marginal values. For proof, see, for example, (Varoufakis 2002).

As an example, suppose Alice has $12 to spend, and further, suppose she derives her pleasure from apples and oranges. An apple costs $2 and an orange costs $3. Alice's marginal values of the fruit and her marginal value densities (per dollar) are shown in Table 1. Note that marginal value densities are diminishing. Alice can attempt to find an optimal solution by allocating her money in a greedy manner to uses with the highest marginal value densities. Doing so, she would allocate her $12 as follows: spend $2 on apple 1, spend $2 on apple 2, spend $3 on orange 1, spend $3 on orange 2, spend $2 on apple 3. This procedure equates marginal value densities: the marginal value density of buying apples is 4, as is the marginal value density of buying oranges. Hence, by the Equimarginal principle, this solution—buy 3 apples and 2 oranges—is optimal.

Note that it is not always possible to equate marginal value densities in discrete settings. Indeed, marginal value densities across uses can be arbitrarily far apart, rendering this greedy approach arbitrarily bad.

	Apples			Oranges		
Fruit #	Price	MV	MVD	Price	MV	MVD
1	2	14	**7**	3	15	**5**
2	2	12	**6**	3	12	**4**
3	2	8	**4**	3	6	2
4	2	2	1	3	3	1

Table 1: Apples and Oranges. MV denotes marginal value. MVD denotes MV density per dollar spent.

The Nonlinear Knapsack Problem

The problem domains in which the Equimarginal Principle applies have the flavor of the knapsack problem. In this classic problem, we are given a set of n items, each with a value v_i and a weight w_i, and our objective is to choose a subset of the items that maximizes the sum of the weights but does not exceed the capacity C of the knapsack. Formally,

$$\max_x \sum_{i=1}^{n} v_i x_i \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_i \leq C \tag{2}$$

More generally, in the aforementioned sample economics problems, the decision faced is one of choosing not only the best uses for the resource, but the quantity of the resource to be allocated to each use as well. Moreover, *the value of each use depends on the quantity selected.* This latter difference creates a knapsack problem with a nonlinear objective function: i.e., a *nonlinear knapsack problem* (NLK) problem (see, for example, (Hochbaum 1995)). Specifically,

$$\max_x \sum_{i=1}^{n} f_i(x_i) x_i \tag{3}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} g_i(x_i) \leq C \tag{4}$$

Typically, the value functions f_i are assumed to be real-valued, concave, and nondecreasing, and the weight functions g_i are assumed to be real-valued, convex, and nondecreasing. The convexity and concavity assumptions ensure that marginal values (and hence, the corresponding densities) are diminishing.

Like the traditional knapsack problem, NLK comes in various flavors: discrete (binary or integer) and continuous. In the former, the resource can be allocated to uses only in discrete quantities (e.g., $x_i \in \{0, 1\}$); in the latter the resource can be allocated to uses in any real-valued quantity (i.e., $x_i \in \mathbb{R}$). We pose and solve a special case of the continuous NLK, and to solve it we reformulate it as a very special discrete (linear) knapsack problem for which the greedy approach is optimal.

An Approximately Optimal Greedy Solution in a Special Case

Consider a continuous nonlinear knapsack problem with f_i concave, $g_i(x_i) = c_i x_i$ for some $c_i \in \mathbb{R}$, and $x_i \in \mathbb{R}$, for all $i = 1, \ldots, n$. Given $K \in \mathbb{N}$, we discretize this problem as follows: let $k = \frac{1}{K}$; for $j = 1, \ldots, K$, let

$$v_{ij} = f_i\left(\frac{jk}{c_i}\right) - f_i\left(\frac{(j-1)k}{c_i}\right) \quad (5)$$

be the marginal value of the jth "piece" of i and let $w_{ij} = k$ be the weight associated with this piece. Rewriting the objective and the constraints yields:

$$\max_x \sum_{ij} v_{ij} x_{ij} \quad (6)$$

$$\text{s.t.} \quad \sum_{ij} x_{ij} \leq C' \quad (7)$$

where $C' = CK$ and $x_{ij} \in \{0, 1\}$, for all $i = 1, \ldots, n$ and $j = 1, \ldots, K$. This problem is a very special 0/1 (linear) knapsack problem in which weights are constant. Since the value functions f_i are concave, marginal value densities are guaranteed to be diminishing. Hence, this problem can be solved greedily by including pieces in order of their marginal value densities, from highest to lowest, until the knapsack's capacity is reached. As in the corresponding continuous problem, a greedy solution to this discrete knapsack problem never includes the ith piece without first including the $i-1$st piece.

As an example, suppose Alice is shopping at a bulk food store and has \$8 to spend on oats and granola. Oats cost \$2 per pound and granola costs \$6 per pound. Alice's utility from oats and granola is given by the following functions of quantity, respectively: $u_o(q_o) = 20q_o - 2q_o^2$ and $u_g(q_g) = 24q_g - 3q_g^2$. The optimal quantities that Alice should buy can be calculated analytically. The solution is to spend $\$\frac{44}{7}$ on oats and $\$\frac{12}{7}$ on granola. The value of this solution has utility ~ 49.71.

Suppose this bulk food store does not accept denominations less than one dollar. Alice pays with single dollar bills. Her marginal value densities (per dollar) are shown in Table 2. Because her marginal value densities are diminishing, Alice can find an optimal solution to this discretized problem by allocating her money in a greedy manner to uses in order of marginal value densities. Alice would allocate her \$8 as follows: spend her first \$6 on oats, spend her last \$2 on granola.

Note that this optimal solution to the discretized problem is nearly an optimal solution to the corresponding continuous problem: its value is $42 + 7.67 = 49.67$. In this situation, as in most real-life problems, the resource (\$8) has to be allocated in discrete amounts (e.g., one dollar or one cent). If the store accepted pennies, and if Alice had \$8 in pennies, this approach would find a solution that is even closer to the optimal solution. We formalize this intuition presently.

Let $OPT_{con}(B)$ denote the optimal value of the continuous problem given a budget (i.e., a knapsack capacity) of B. Let $OPT_{dis}(B)$ denote the optimal value of the analogous discretized problem. In addition, let v_i^* denote use i's marginal value density in an optimal solution to $OPT_{dis}(B)$. (NB: Marginal value densities need not be equated in optimal solutions to discrete knapsack problems.)

\$	Oats			Granola		
	lbs	Utility	MVD	lbs	Utility	MVD
1	0.5	9.5	**9.5**	0.167	3.92	**3.92**
2	1	18	**8.5**	0.333	7.67	**3.75**
3	1.5	25.5	**7.5**	0.5	11.25	3.58
4	2	32	**6.5**	0.667	14.67	3.42
5	2.5	37.5	**5.5**	0.833	17.92	3.25
6	3	42	**4.5**	1	21	3.08
7	3.5	45.5	3.5	1.167	23.92	2.92

Table 2: Oats and Granola at a bulk food store. MVD denotes marginal value density (i.e., MV per dollar).

Theorem 1 Let $v^*_{\text{min}} = \max(0, \min_i v_i^*)$. Given $K \in \mathbb{N}$ and $k = \frac{1}{K}$, $OPT_{con}(B) \leq OPT_{dis}(B) + \epsilon(k)$ where $\epsilon(k) = k \sum_{i|v_i^*>0}(v_i^* - v^*_{\text{min}})$.

Proof If marginal value densities are equated in the solution to the discretized problem, this solution is an optimal solution to the continuous problem as well by the equimarginal principle.

Suppose marginal value densities are not equal in the discretized solution. Choose the lowest marginal value density and add additional budget Δ to the other uses until marginal value densities are equated across all uses in the continuous problem. Again, by the equimarginal principle, this solution is optimal.

Of course, $OPT_{con}(B) \leq OPT_{con}(B + \Delta)$, because Δ is additional budget. Also, $OPT_{con}(B + \Delta) = OPT_{dis}(B) + \epsilon$, because ϵ is the extra value derived from equating marginal value densities across uses. Therefore, $OPT_{con}(B) \leq OPT_{dis}(B) + \epsilon$.

In some sense, the above theorem is quite weak, since as noted above, in discrete NLK problems, marginal value densities across uses can be arbitrarily far apart. However, the following theorem shows that taking a discretized approach to solving a continuous NLK problem of the form stated above can be valid nonetheless.

Corollary 2 Via the above procedure, as $K \to \infty$, the value of an optimal solution to the discretized 0/1 (linear) knapsack problem approaches the value of an optimal solution to the continuous NLK problem, assuming the value functions f_i are bounded.

Proof By Theorem 1, it suffices to show that $\epsilon \to 0$ as $K \to \infty$. This follows immediately from the fact that the values based on which ϵ is computed are bounded.

Next we define the TAC SCM bidding problem and a tractable approximation called expected bidding. We note that the latter is a continuous NLK problem with g_i linear and diminishing marginal returns. Hence, the discretization procedure described above, followed by an application of the greedy algorithm, yields decent approximate solutions to this problem.

Bidding in TAC SCM

In TAC SCM, six software agents compete in a simulated sector of a market economy, specifically the personal computer (PC) manufacturing sector. Each agent can manufacture 16 different products (i.e., types of computers), characterized by different *stock keeping units* (SKUs). Building each SKU requires a different combination of components, of which there are 10 different types. These components are acquired from a common pool of suppliers at costs that vary as a function of agent demand. At the end of each day, each agent converts a subset of its components into SKUs according to a production schedule that it generates for its factory, within a maximum capacity of 2000 cycles. It also reports a delivery schedule assigning the SKUs in its inventory to outstanding customer orders.

The next day, the agents compete in first-price reverse auctions to sell their finished products to customers: i.e., an agent secures an order by *under*bidding the other agents. More specifically, each day the customers send RFQs to the agents. Each RFQ contains a SKU, a quantity, a due date, a penalty rate, and a reserve price—the highest price the customer is willing to pay. Each agent sends an *offer* in response to each RFQ, representing the price at which it is willing to satisfy that RFQ. After each customer receives all its offers, it selects the agent with the lowest-priced offer and awards that agent with an *order*. After 220 simulated days of procurement, production, delivery, and bidding each of which lasts a total of 15 seconds, the agents are ranked based on their profits.

The Stochastic Bidding Problem

The decision problem faced by a TAC SCM agent can be divided into three central subproblems (Benisch *et al.* 2004): *procurement* of components from suppliers, *bidding* on customer requests for quotes (RFQs), and *scheduling* of factory production and deliveries. Here we focus on the bidding problem, which subsumes the scheduling problem. A study of how our methods extend to procurement remains for future work.

For simplicity, we assume all due dates are set past the end of the game, making penalties irrelevant. Also, as we are concerned only with bidding and not with procurement in this paper, all components are assumed to be infinitely available at no cost.

Agents are assumed to have perfect price prediction, that is, they know the probability of winning an order as a function of any bid they submit. We encode this information in "price-probability models." They are also assumed to have access to an accurate stochastic model of future demand (i.e., the number and variety of RFQs that will arrive each day).

A decision-theoretic version of the TAC SCM bidding problem, under the aforementioned assumptions, can be formulated as a recursive stochastic program. We do so here, using the notation explained in Figure 1.

The recursive function takes five inputs: today's product inventory, today's outstanding orders, today's RFQs, the history of RFQs received on previous days, and today's date. The objective is to choose bids on today's RFQs and to decide upon today's production and delivery schedules in such a way as to maximize today's revenue plus expected future revenue.

Variables

$x_r \geq 0$	bidding policy: bid price for RFQ r
$y_j \geq 0$	production schedule: quantity of SKU j
$z_i \in \{0,1\}$	delivery schedule: 1 if order i is delivered; 0 otherwise

Indexes

t	day index
j	SKU index

Functions

$p(r, x_r)$	probability of winning RFQ r with bid x_r

Constants

a_j	number of units of SKU j delivered
b_j	number of units of SKU j in inventory
c_j	cycles expended to produce one unit of SKU j
d_{ij}	1 if order i is for SKU j; 0 otherwise
π_i	revenue for delivering order i
q_i	quantity of order i
N	total number of days
C	daily production capacity in cycles
O	set of outstanding orders
Q	set of (today's) orders
R	set of (today's) RFQs
R'	set of tomorrow's RFQs
h	history of RFQs received until now

Figure 1: Notation for Recursive Stochastic Program

Bids on day t are placed on RFQs received that day. The set of RFQs R' received on day $t+1$ is a random variable that is independent of any decisions but depends on the history of past RFQs received.

The bids placed on day t determine the likelihoods of receiving various sets of orders on day $t+1$. Each set of new orders is called a *scenario*. Each scenario Q is weighted by probability $\Pr(Q)$ as determined by the given price-probability model. Specifically, $\Pr(Q)$ equals the product of the probabilities of winning all RFQs that are part of Q and the probabilities of not winning RFQs that are not part of Q (Equation 9).

Delivery and production scheduling decisions today affect what will remain in product inventory tomorrow. Indeed, tomorrow's product inventory equals today's product inventory b minus any product inventory depleted by today's deliveries a plus any additional inventory produced today y.

Each day capacity and allocation constraints are enforced. Equation 10 ensures that there are enough products in inventory for today's delivery schedule. Equation 11 ensures that today's production schedule does not consume more cycles than the daily capacity.

The base case (Equation 12) of the recursion pertains to the last day. Orders can be scheduled for delivery but there is no production or bidding.

if $0 \leq t < N$,

$$F(b,O,R,h,t) = \max_{x,y,z} \sum_{i \in O} z_i \pi_i +$$

$$\sum_{Q \in 2^{|R|}} \Pr(Q) E_{R'|h} \left[F(b - a + y, O \cup Q, R', h \cup R, t+1) \right] \tag{8}$$

subject to:

$$\Pr(Q) = \prod_{r \in Q} p(r, x_r) \prod_{r \notin Q} (1 - p(r, x_r)) \tag{9}$$

$$a_j = \sum_{i | i \in O, d_{ij}=1} z_i q_i \quad \forall j; \quad a \leq b \tag{10}$$

$$\sum_j y_j c_j \leq C \tag{11}$$

if $t = N$,

$$F(b,O,R,h,t) = \max_{z} \sum_{i \in O} z_i \pi_i \tag{12}$$

To find the set of optimal bids on day 0, ideally one would solve $F(0, \{\}, R, \{\}, 0)$, where R is the set of RFQs received on day 0. However, this recursive stochastic program is intractable because of an exponentially increasing number of scenarios after each recursive call.

The Expected Bidding Problem

A tractable approximation of 1-day stochastic bidding called expected bidding was considered in (Benisch *et al.* 2004). In the expected bidding problem, it is assumed that a bid that has probability p of winning an order for quantity q wins a partial order for quantity pq with probability 1. In this deterministic setup, a set of $|R|$ bids results in exactly one set of partial orders, that is, one scenario instead of $2^{|R|}$ scenarios in Equation 8.

Unlike (Benisch *et al.* 2004), where there was no model (stochastic or deterministic) of future demand, here, we study an N-day version of expected bidding. To do so, we collapse the stochastic information contained in the price-probability models into partial orders as in (Benisch *et al.* 2004), and we collapse the stochastic information contained in the stochastic model of future demand into a single statistic (e.g., the mean). Generally speaking, this approach to solving stochastic optimization problems is called the expected value method (Birge & Louveaux 1997).

Define by "market segment" any subset in any partitioning of the customer demand.[1] The objective in expected bidding is to find a set of bids x_i, one per market segment i, that maximizes expected revenue, subject to the constraint that expected production does not exceed available capacity, given, for each market segment i, a demand curve $f_i(x_i)$ that maps bid prices into expected quantities together with the number of cycles $c_i \in \mathbb{N}$ required to produce one unit of i.

Expected bidding can be stated formally as a mathematical program:

$$\max_x \sum_{i=1}^{n} f_i(x_i) x_i \tag{13}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} c_i f_i(x_i) \leq C \tag{14}$$

where $x_i \in \mathbb{R}$ is the bid in market segment i and $f_i(x_i)$ is the expected quantity of i at price x_i. Equivalently, we can state expected bidding as follows:

$$\max_{x'} \sum_{i=1}^{n} f_i^{-1}(x'_i) x'_i \tag{15}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} c_i x'_i \leq C \tag{16}$$

where $x'_i \in \mathbb{R}$ is the expected quantity of i desired. Assuming f_i^{-1} is concave (so that marginal revenues are diminishing), this latter formulation is equivalent to Equations 3 and 4 with $g_i(x'_i) = c_i x'_i$.

To solve this continuous knapsack problem, we reformulate it as a discrete one as in the oats and granola example and we solve this latter problem greedily. That is, we develop an approximate solution to a deterministic approximation of the bidding problem: i.e., an approximate solution to the approximate problem! Ultimately, we test both an ILP bidder (feeding an N-day version of the expected bidder studied in (Benisch *et al.* 2004) to CPLEX) and our greedy approach on simulated instances of stochastic bidding.

A Greedy Algorithm

Since expected bidding is a continuous NLK problem with g_i linear where the assumption of diminishing marginal values holds, and since our discussion above shows that a greedy algorithm yields a decent approximation for such instances of the continuous NLK, we now describe a greedy algorithm that relies on this latter assumption about the market structure to solve the expected bidding problem in TAC SCM. At a high level, our algorithm first fulfills outstanding orders; second, it greedily schedules SKU production for a given segment of the overall market in decreasing order of marginal revenue per cycle; third and last, it determines its bids by computing the percentage of demand met in this market segment, and then bidding the price associated with this percentage according to the corresponding price-probability model.

Price-Probability Models

A price-probability model is a mapping from prices to the probability of winning an RFQ in a given market segment. An example of a linear price-probability model is (see Figure 2(a)):

$$\frac{2200 - \text{bid price}}{800} = \text{probability of winning the RFQ} \tag{17}$$

[1] In our experiments, we partition the customer RFQ market by SKU type.

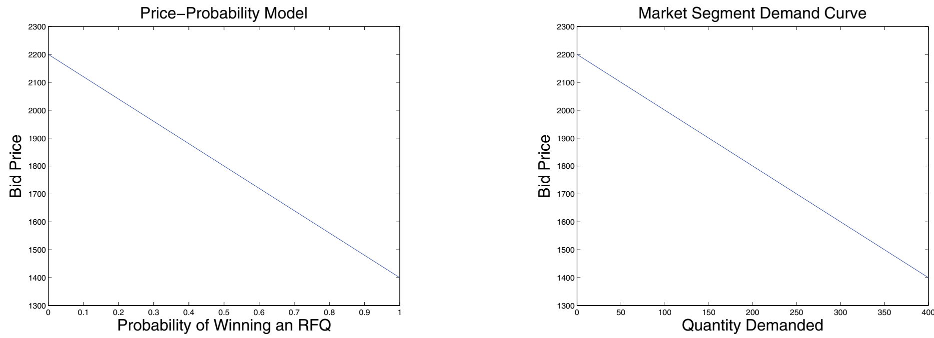

Figure 2: (a) Sample price-probability model. (b) Sample market segment demand curve.

Here, it is assumed that a price of 2200 will have no chance of winning the RFQ, whereas a price of 1400 is guaranteed to win. At a price of 1800, then, a seller would win with probability 0.50 Price-probability models need not be linear, but can incorporate whatever techniques necessary to model the likelihood of a bid price being the lowest offered on an RFQ.

Market Segment Demand Curves & Marginal Revenue Lists

In a competitive marketplace with indistinguishable products, a seller hoping to adjust its market share can do so only by changing its offer price. To assist a seller in making such pricing decisions, a price-probability model for a market segment can easily be converted into a representation of that segment's demand curve: the expected quantity associated with a given price is determined by multiplying the probability associated with that price in the price-probability model by the total quantity demanded in that market segment.

For example, suppose we are using the same model as in the previous section to represent a market segment consisting of 80 RFQs of 5 SKUs each, 400 SKUs in total. By aggregating the quantities of each SKU demanded by all the RFQs in a market segment, we can use the probability specified by the price-probability model to calculate the expected quantity of each SKU demanded at a given price (see Figure 2(b)). In our example, a price of 1800 wins with probability 0.50. Hence, if an agent wishes to capture 50% of the market segment, it should make offers at a price of 1800. Conversely, in a market segment with 400 SKUs, an agent could expect to win 200 SKUs worth of demand.

By traversing the market segment demand curve at a constant, exogenously declared incremental quantity (elsewhere referred to as step size), we generate the marginal revenue (per cycle) list that is input to our greedy bidding algorithm. Corresponding to the sample market segment demand curve shown in Figure 2(b), assuming a step size of 20% (i.e., 80 SKUs), a sample marginal revenue list is shown in Table 3. The prices in the first and second rows were generated by querying the price-probability model for the prices corresponding to the quantities 160 and 80. Then, the marginal revenue per cycle in the second row was computed as $\frac{160*1880-80*2040}{80*5} = 344$, assuming that five cycles are required to produce SKUs in this market segment.

Quantity	Price	Marginal Revenue / Cycle
80	2040	408
160	1880	344
240	1720	280
320	1560	216
400	1400	152

Table 3: Market Segment Marginal Revenue List

Marginal Bidding in Detail

In more detail, the Marginal Bidder proceeds as follows:

Inputs: Current and Future RFQs, Market Segment Marginal Revenue Lists, Price-Probability Models, Outstanding Orders, Product Inventory, Component Inventory

1. for each order
   - if the order is fulfillable using product inventory, schedule it for delivery
   - if the order is not fulfillable using product inventory, schedule it for production
   - reduce product inventory or component inventory and production capacity accordingly
2. repeat while there exist positive marginal revenues per cycle, remaining production capacity and components in inventory
   - take from inventory or schedule production of one unit of the product from the market segment with the highest marginal revenue per cycle, ignoring products that are not in product inventory and cannot be built from component inventory
   - remove the first entry from the product's market segment marginal revenue list and reduce product inventory or production capacity and component inventory accordingly
3. for each product

- bid the price at which the agent expects to win with probability:

$$\frac{\text{quantity of the product scheduled}}{\text{quantity of the product requested}}$$

Outputs: A bid corresponding to each current RFQ

Experiments

Here, we report on experiments designed to compare the performance of two bidding algorithms, one based on our greedy algorithm (MB, for marginal bidding), the other on our integer linear programming solution (ILP). Both are tested with (MB-L,ILP-L) and without lookahead (MB,ILP). The marginal bidders are run with both a 1% and 5% step size, and the ILP bidders are tested with a 1% discretization (100 possible price points) and 5% discretization (20 possible price points).

Lookahead

One way to incorporate future demand is a technique we refer to as Lookahead. With Lookahead, a bidder schedules the game's remaining demand in one long day with capacity equal to *Daily Capacity* x *Remaining Game Days*. A daily production schedule is generated from this game long schedule by calculating the ratios at which each SKU is produced, and then scaling the production schedule down proportionally to one day's capacity (using only the capacity remaining after the production associated with orders is scheduled).

Setup

Recall that in TAC SCM an RFQ is awarded to the agent presenting the lowest offer below the reserve price. We tested our bidding algorithms in isolation, not against other bidding agents, as in a true reverse-auction setting. The awarding of contracts to RFQs was determined solely by the simulator, which transforms an offer into an order with the probability associated with the bid price under the price-probability model for the relevant market segment. Agents were endowed with perfect price prediction: i.e., the price-probability model was shared between the agent and the simulator. Moreover, agents were also endowed with complete and perfect knowledge of future demand. In other words, the set of customer RFQs scheduled to arrive each day was broadcast before the simulations began.

We tested our bidders in four setups, which differed only in the level of customer demand (i.e., the number of RFQs) each day. In all setups, demand was assumed to be uniformly distributed across SKUs. For each setup 25 trials each lasting 25 days were run under identical conditions. The only randomness arose in the awarding of orders, which was done based on the linear price-probability model specifed in Equation 17.

In our results tables, Revenue is reported in millions; Runtime is reported in seconds on a per day basis; and Cycles refers to the average number of factory cycles (necessarily ≤ 2000) used per day.

Constant Demand

In our first simulation, a constant customer demand of 100 RFQs per day was given to the bidding algorithms. Assuming constant demand, there exist no particular advantage to planning for future demand, since the optimal solution for the whole game is a concatenation of the optimal solutions on each of the individual days. Indeed, the revenues of all the bidding algorithms under constant demand (Table 4) are within one standard deviation of each other.

Table 4: 25-day simulation under constant demand.

Agent	Rev.	S.Dev.	Time	S.Dev.	Cycles
MB 1%	16.91	0.77	0.38	0.73	1999.0
ILP 1%	16.92	0.64	1.82	1.38	1998.7
MB-L 1%	16.95	0.77	0.40	0.78	1997.3
ILP-L 1%	16.91	0.64	22.7	63.6	1997.5
MB 5%	16.91	0.71	0.08	0.37	1995.8
ILP 5%	16.94	0.70	0.55	0.71	1998.3
MB-L 5%	16.95	0.63	0.08	0.37	1997.2
ILP-L 5%	16.91	0.66	4.16	3.18	1997.3

High/Low Demand

High/Low demand is an artificial demand setup designed to highlight the advantages of taking future demand into account. In our simulation, demand on even numbered days is quite high (100 RFQs) whereas demand on odd numbered days is quite low (0 RFQs).

Bidders with Lookahead are able to exploit this bimodal demand by bidding aggressively on days when there is a surplus in demand, and fulfilling orders with excess inventory produced on the days when demand is low. Bidders without Lookahead, because they are only able to consider a single day's demand, starve on the days when demand is lower than their factory's capacity. As expected, the revenues for Bidders with Lookahead are substantially higher than their no Lookahead counterparts in this setup (Table 5). In particular, the Marginal Bidder with Lookahead earns more revenue than the ILP without Lookahead. Moreover, the former is also more computationally efficient, finding it's solution between 5 and 18 times faster than the ILP.

Note that in this setup (only) the Marginal Bidder without Lookahead is performing approximately twice as fast as the Marginal Bidder with Lookahead. On low demand days, the Marginal Bidder without Lookahead has no demand to consider, and so immediately returns the empty offer set. In contrast, the Lookahead Bidder always has future demand to consider, and so consumes runtime even on empty demand days.

Decreasing Demand

A more realistic demand setup in TAC SCM is one of gently decreasing demand. This is representative of supply effects that are artifacts of the start of a typical game, when component constraints cause the agents to leave an initially large chunk of customer demand unfulfilled. As the game progresses and agents begin to manufacture products in greater

Table 5: 25-day simulation under high/low demand.

Agent	Revenue	S.Dev.	Time	S.Dev.	Cycles
MB 1%	10.19	2.30	0.44	0.59	1243.2
ILP 1%	10.19	2.21	0.84	1.24	1244.1
MBL-L 1%	14.41	1.22	0.95	0.75	1992.2
ILP-L 1%	14.40	1.21	17.1	22.0	1997.6
MB 5%	10.18	2.26	0.09	0.31	1242.5
ILP 5%	10.17	2.36	2.15	4.79	1242.8
MB-L 5%	14.40	1.30	0.20	0.44	1991.8
ILP-L 5%	14.40	1.28	2.31	2.19	1992.6

numbers, this unfulfilled demand diminishes and then disappears when the market reaches a competitive steady state, subject to drift in customer demand.

Our simulation of decreasing demand begins with 120 RFQs decreasing by 5 RFQs each day until no RFQs arrive on the last day. Again, unable to compensate for future demand, the Bidders without Lookahead initially bid for enough RFQs to fill one day of production and then starve in later days when demand diminishes and a single day's RFQs no longer constitutes a full day's worth of production. The Bidders with Lookahead, knowing that future demand will be insufficient to keep their factories running, bid for a higher percentage of the excess early demand and are able to sustain themselves for longer through the dry spell at the end of the game. As above, the Marginal Bidder with Lookahead outperforms the ILP without Lookahead, both in revenue and in runtime. (See Table 6). The mirror case of increasing demand (from 0 to 120 RFQs) is omitted here but produces symmetric results.

Table 6: 25-day simulation under decreasing demand.

Agent	Revenue	S.Dev.	Time	S.Dev.	Cycles
MB 1%	13.31	0.77	0.45	0.83	1724.9
ILP 1%	13.35	0.92	1.41	1.44	1721.3
MB-L 1%	15.46	0.89	0.49	0.83	1997.5
ILP-L 1%	15.42	0.92	16.8	34.6	1997.2
MB 5%	13.34	0.84	0.10	0.42	1723.0
ILP 5%	13.30	1.16	0.75	2.06	1711.7
MB-L 5%	15.46	0.94	0.10	0.40	1997.4
ILP-L 5%	15.41	0.89	2.40	2.85	1997.1

Related Work

Researchers at the Cork Constraint Computation Center implemented an integer linear approach to bidding in a constraint based agent, Foreseer (Burke *et al.* 2005). Similar to the expected bidder posited in (Benisch *et al.* 2004), Foreseer uses profit as the objective function, bid prices as the decision variables, and constraints based on factory capacity, component availability, and reserve prices.

Researchers at CMU reduce a probabilistic pricing problem (akin to TAC SCM bidding) to a nonlinear continuous knapsack problem, under the assumption of diminishing marginal returns, and present an ϵ-optimal solution to this problem with arbitrary concave value functions (Benisch, Andrews, & Sadeh 2006). Their approach is efficient assuming normally distributed customer valuations (an analog of price-probability models). Our method's efficiency does not depend on the form of the price-probability model.

The TacTex team developed a greedy bidder along the lines of the marginal bidder presented here, with a few subtle distinctions (Pardoe & Stone 2004). TacTex is initialized to bid reserve prices on each RFQ and then iteratively reduces its bids according to some selection mechanism until production capacity is reached or profits are no longer increasing. The selection mechanism relies on a heuristic that determines whether the most limiting resource is production capacity (in which case it selects by profit per cycle) or component availability (in which case it selects by change-in-Profit / change-in-Probability).

Discussion and Conclusion

We have described a marginal revenue based method for bidding on customer demand in the TAC SCM environment. The greedy solutions found by the Marginal Bidders are competitive with our ILP solutions in terms of revenue, both with and without Lookahead, but take a fraction of the time to compute. Since each day in TAC SCM is simulated in 15 physical seconds, this computational savings can in turn be applied to other dimensions of the agent's decision making.

Our ultimate goal is to develop a scalable bidding algorithm so that it can be extended into a procurer capable of reasoning about long-term future demand. Because the ILP considers each RFQ as a separate decision variable, its complexity grows rapidly as a function of the number of RFQs. By reasoning about SKUs in collective market segments, the Marginal Bidder avoids this complexity and appears to be more readily extensible to the procurement problem.

As a first step towards procurement, we extended our Marginal Bidder to handle due dates. Doing so involved changing the multiday production schedule from being a monolithic block of capacity equal to the entire game's factory capacity into a collection of capacity constraints each representing a separate day, and reorganizing the set of market segment demand curves according to due date as well as SKU. Products can then be scheduled backwards from their latest possible production date, until all possible days of production have been filled.

In practice, since our Marginal Bidder prioritizes scheduling orders, due dates and penalties do not have a large impact on bidding decisions, which is why they are not considered in this paper. Consideration of due dates is crucial in procurement, however, and makes for an exponentially larger ILP.

It remains to be seen whether our Marginal Bidding approach can be extended to handle interdependent uses, where devoting resources to one use can affect the marginal value density of another. Interdependencies arise naturally in procurement because components are shared among SKU types.

Since information about market prices is provided by the TAC SCM server by SKU type, there exist no particular returns to considering RFQs of the same type separately in TAC SCM. However, one could conceive of more complicated scenarios in which such segmentation of the overall market results in a loss of information regarding specific RFQs. The Marginal Bidding algorithm is agnostic towards this market division, however, so whatever tradeoff between specificity and complexity is deemed desirable can be transparently integrated into our bidding module.

It is also worth noting that despite the game-theoretic nature of bidding in TAC SCM, our focus here is on a decision-theoretic (stochastic) optimization problem, not on game-theoretic equilibrium calculations. The enormity of the decision space in TAC SCM makes game-theoretic strategic analysis intractable with current technology. It remains to be seen whether an effective game-theoretic approach can be developed to exploit strategic opportunities in the TAC SCM game.

Finally, in the near future, we plan to test the robustness of our algorithms to imperfect modeling of future demand and trading prices. Doing so would lead to progress in addressing the challenging game-theoretic issues that arise in environments like TAC SCM that are inhabited by multiple artificially intelligent agents.

Acknowledgments

This research was supported by NSF Career Grant #IIS-0133689.

References

Arunachalam, R., and Sadeh, N. M. 2005. The supply chain trading agent competition. *Electronic Commerce Research and Applications* 4(1):66–84.

Benisch, M.; Andrews, J.; and Sadeh, N. 2006. Pricing for customers with probabilistic valuations as a continuous knapsack problem. In *ACM International Conference Proceeding Series*, volume 156, 38 – 46.

Benisch, M.; Greenwald, A.; Grypari, I.; Lederman, R.; Naroditskiy, V.; and Tschantz, M. 2004. Botticelli: A supply chain management agent. In *Third International Conference on Autonomous Agents and Multiagent Systems*, volume 3, 1174–1181.

Birge, J., and Louveaux, F. 1997. *Introduction to Stochastic Programming*. New York: Springer.

Burke, D. A.; Brown, K. N.; Koyuncu, O.; Hnich, B.; and Tarim, A. 2005. Foreseer: A constraint based agent for tac-scm. In *Trading Agent Competition - Supply Chain Management, IJCAI '05, Edinburgh, Scotland*.

Hochbaum, D. S. 1995. A nonlinear knapsack problem. In *Operations Research Letters*, volume 17, p. 103–110(8).

Pardoe, D., and Stone, P. 2004. TacTex-03: A supply chain management agent. *SIGecom Exchanges* 4(3):19–28.

Varoufakis, Y. 2002. *Foundations of Economics: A Beginner's Companion*. Taylor and Francis e-Library.