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

Amy Greenwald, Victor Naroditskiy, Tyler Odean, Mauricio Ramirez, Eric Sodomka, Joe Zimmerman, and Clark Cutler

Department of Computer Science, Brown University, Box 1910, Providence, RI 02912

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 (the procurement problem), and then sell assembled computers to customers in reverse auctions (the bidding problem). This paper is concerned only with bidding, in which an agent must decide 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 certain variations of Marginal Bidding can compute bids faster than our ILP solution, which enables Marginal Bidders to consider future demand as well as current demand, and hence achieve greater revenues when knowledge of the future is valuable.

1 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 [1].

In this paper, we study the TAC SCM bidding problem, where the goal is to choose prices at which to offer to sell computers to customers today, balancing the tradeoff between maximizing revenue per order—by placing high bids—and maximizing the quantity of customer orders won—by placing low bids, within the constraints of current and future component availability and production capacity. Ideally, these decisions should be made taking into account future demand: in a bull market it may be advantageous to reserve today's production capacity for future, more profitable demand; in a bear market it may be preferable to bid more aggressively early on, claiming a larger share of current demand to be fulfilled with products manufactured in the future.

To model these tradeoffs, we formulate bidding in TAC SCM as an N-day recursive, stochastic, integer linear program (ILP). The mathematical program is recursive because the agent faces the same decision variables day after day, namely the prices at which to set its current bids so as to maximize the sum of its

J. Collins et al. (Eds.): AMEC/TADA 2007, LNBIP 13, pp. 217–239, 2008.

current revenue and its expected future revenue. It is stochastic in part because of the inherent uncertainty in future demand. However, we also use stochasticity to model the game-theoretic dynamics of bidding in a reverse auction, thereby reducing what is truly a game-theoretic problem to a decision-theoretic one. This is an important simplifying assumption that permeates our study.

A tractable approximation of 1-day bidding, called *expected bidding*, was considered in Benisch *et al.* [2]. We revisit this problem in paper, and show that it reduces to a generalization of the classic knapsack problem, the so-called *nonlinear knapsack problem* (NLK). Then, 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 propose a greedy solution to the expected bidding problem, which we call Marginal Bidding. We advocate for Marginal Bidding in this paper because it scales linearly with the number of days, and can hence more easily solve an N-day extension of expected bidding than traditional ILP solutions.

To analyze the performance of various heuristics designed for TAC SCM, we built a simulator that generates decision-theoretic simplifications of the game-theoretic problems TAC SCM agents face, such as bidding. Using our simulator, we compared the performance of several variants of Marginal Bidding with an ILP solution. We show that certain variations of Marginal Bidding can compute bids faster than our ILP solution; hence, incorporating a Marginal Bidder into a TAC SCM agent would allow for more time to be spent on other decision problems (e.g., procurement). Moreover, this speedup enables Marginal Bidders to reason about future demand as well as current demand, and hence achieve greater revenues when knowledge of the future is valuable. While the gains to be realized by reasoning about future demand in TAC SCM appear modest, we demonstrate that more substantial gains can be realized under more volatile or seasonal conditions that generate more extreme market swings.

This paper is organized as follows. We begin by describing the Equimarginal Principle of marginal utility theory, originally posited by Gossen in the mid 1800's. We note that this principle can be applied to solve the nonlinear knapsack problem. Then, we present a discretization technique coupled with a greedy algorithm, which we prove approximately solve the NLK. (Technically, we prove that our approach yields a Fully Polynomial Time Approximation Scheme—a FPTAS—for the NLK.) Next, we formalize TAC SCM bidding as an N-day recursive stochastic program, and argue that expected bidding, a 1-day deterministic approximation, can be reduced to solving an instance of the NLK. Then, we present Marginal Bidding, a heuristic for solving an N-day extension of expected bidding that incorporates the aforementioned discretization technique and greedy approach to solving the NLK. Finally, we compare experimentally the performance of two heuristics, Marginal Bidding and an ILP, in simulations of the TAC SCM bidding problem.

2 The Equimarginal Principle

The Prussian economist H. H. Gossen is credited with observing two fundamental laws of utility. The first is 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 in which a limited resource (in the above quote, time, but later, means) is 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 (fixed) income among different commodities to maximize her utility; and a firm deciding how to proportion its (finite) labor and capital to maximize its profits.

The second of Gossen's laws is the Law of Diminishing Marginal Returns:

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

A key assumption underlying both of Gossen's laws is that one cannot enjoy all pleasures indefinitely because a pleasure is not free—rather, it comes at some expense. Indeed, when Gossen writes the "amount of pleasure" he means the additional value that derives from enjoying a bit more of the pleasure at a bit more expense. In modern terms, this quantity—the ratio of a pleasure's marginal value to its marginal cost—can be construed as *marginal return.*

Assuming diminishing marginal returns, it is easy to see that in an optimal solution to such a resource allocation problem, marginal returns are equal.[1] Indeed, if the marginal returns were unequal, a better allocation could be achieved by redistributing a unit of the resource from the use with a lower marginal return to the use with a higher marginal return. Gossen's claim is less obvious: that equal marginal returns imply an optimal solution. For a proof, see Mas-Colell *et al.* [10] (Theorem M.K.3 on page 961), for example.

2.1 The Nonlinear Knapsack Problem

The problem domains in which the Equimarginal Principle applies have the flavor of the *knapsack problem.* In this problem, we are given a set of n items, each with a value v_i and a weight w_i, together with a knapsack of finite capacity $C \geq 0$. Our objective is to pack a variety of items in the knapsack such that the sum of the values of the items packed is maximized, but their total weight does not exceed the capacity of the knapsack. Formally,

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} v_i x_i \tag{1}$$

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

[1] For ease of exposition, we assume that in an optimal solution, a strictly positive amount of the resource is allocated to each use: i.e., there exists an interior solution.

In the continuous version of the problem, the x_is are in the range $[0, 1]$; in the 0/1 version (which is NP-hard), they are in the set $\{0, 1\}$. In either case, the x_is are bounded; otherwise, the problems would be unbounded.

In the aforementioned sample economics problems, the decision faced is one of choosing not only the best uses for the resource (i.e., which items to pack), but the quantity $x_i \geq 0$ of the resource to allocate to each use, where, in general, the value of a use can depend on its quantity. This final consideration creates a knapsack problem with a potentially nonlinear objective function: i.e., a *nonlinear knapsack problem* (NLK) problem (see, for example, Hochbaum [8]). Specifically,

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} f_i(x_i) \tag{3}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} g_i(x_i) \leq B \quad \text{and} \quad \forall i \;\; x_i \geq 0 \tag{4}$$

In NLKs, the f_is are value functions; the g_is are cost functions; and the knapsack's capacity C is typically re-interpreted as a budget B.

In a typical instance of the NLK, the x_is are unbounded above, the f_is are real-valued, concave, and nondecreasing, and the g_is are real-valued, convex, and nondecreasing. Concavity (convexity) of the value (cost) function implies the derivative of the value (cost) function, i.e., marginal value (marginal cost), is nonincreasing (nondecreasing). When we divide nonincreasing marginal values by nondecreasing marginal costs, the result is "diminishing marginal returns." Hence, by the Equimarginal Principle, total value is maximized in a NLK when marginal returns are equated across all areas of use: i.e.,

$$\mu_1(x_1) = \frac{f_1'(x_1)}{g_1'(x_1)} = \ldots = \frac{f_i'(x_i)}{g_i'(x_i)} = \ldots = \frac{f_n'(x_n)}{g_n'(x_n)} = \mu_n(x_n) \tag{5}$$

The NLK can be solved exactly in polynomial time when f is quadratic and g is linear (see, for example, Tarasov, *et al.* [11]). The approach we take in this paper can be applied more generally; in particular, it can be used for arbitrary nondecreasing concave value and convex cost functions.

2.2 A Discretization Technique

In this section, we propose a strategy for approximating the solution to the NLK. This strategy involves discretizing the problem, and reformulating it as a very special 0/1 (linear) knapsack problem that can be solved greedily. In the next section, we prove that, with finer and finer discretization, (the value of) an optimal solution to our discrete problem becomes a better and better approximation of (the value of) an optimal solution to the original NLK.

Consider a nonlinear knapsack problem with the f_is satisfying the typical assumptions, and $g_i(x_i) = c_i x_i$ for $c_i, x_i \in \mathbb{R}$, for all $i = 1, \ldots, n$. We discretize this problem by assuming the limited resource can be allocated to each use in

$K \in \mathbb{N}$ equal parts, so that the size of each is $k = \frac{B}{K}$. By spending k on use i, the incremental quantity $s_i = \frac{k}{c_i}$ of i is consumed. We refer to s_i as the unit size of use i, k as the unit cost, and K as the discretization factor.

Suppose we have consumed the quantity $x_i - s_i$ of use i. Consuming an additional unit of size s_i yields the following, which we call *unit marginal return*:

$$\nu_i(x_i) = \frac{\int_{x_i - s_i}^{x_i} f_i'(t)dt}{\int_{x_i - s_i}^{x_i} g_i'(t)dt} = \frac{f_i(x_i) - f_i(x_i - s_i)}{g_i(x_i) - g_i(x_i - s_i)} = \frac{f_i(x_i) - f_i(x_i - s_i)}{c_i s_i} = \frac{f_i(x_i) - f_i(x_i - s_i)}{k} \quad (6)$$

Observe that our assumptions on f ensure that unit marginal returns are non-increasing, just like marginal returns themselves.

Now, for use i and $j = 1, \ldots, K$, let $v_{ij} = \nu_i(js_i)$ be the value of the jth unit of use i, and let $w_{ij} = k$ be the cost of this unit. We rewrite the objective function (1) and the constraint (2) to pose a 0/1 (linear) knapsack problem:

$$\max_{x_{ij}} \sum_{ij} v_{ij} x_{ij} \quad (7)$$

$$\text{s.t.} \quad \sum_{ij} w_{ij} x_{ij} \leq B \quad (8)$$

Here, $x_{ij} \in \{0, 1\}$, for all $i = 1, \ldots, n$ and $j = 1, \ldots, K$.

Constraint 8 ensures that the budget B is not exceeded. Since weights are constant and equal to k, this constraint can be restated as follows: $\sum_{ij} x_{ij} \leq K$. Hence, our problem is in fact a very special 0/1 (linear) knapsack problem that can be solved greedily by consuming units of the various uses in sorted order by value, from highest to lowest, until the budget is exhausted, breaking ties by including the jth unit of use i before the $j + 1$st.

Further, a near-optimal solution to the (original) continuous NLK can be constructed from an optimal solution to our discrete problem, precisely because our greedy solution to the latter never includes the jth unit without first including the $j - 1$st. In Section 2.3 below, we derive a bound on the quality of this greedy solution as an approximate solution to the continuous NLK, but first we demonstrate the use of our discretization technique by example.

Example. Suppose Alice is shopping at a bulk food store and has \$8 to spend on oats and granola. Oats cost \$2 per pound (i.e., $g_o(x_o) = 2x_o$). Granola costs \$6 per pound (i.e., $g_g(x_g) = 6x_g$). Alice's value functions for oats and granola are $f_o(x_o) = 20x_o - 2x_o^2$ and $f_g(x_g) = 24x_g - 3x_g^2$, respectively. The optimal quantities that Alice should buy can be calculated analytically. She should spend $\$\frac{44}{7}$ on oats and $\$\frac{12}{7}$ on granola. This solution has total value ~ 49.71.

Suppose this bulk food store does not accept denominations less than \$2. In other words, Alice must pay with \$2 bills. Alice now faces a discretized knapsack problem of the form just described, with $K = 4$ (the discretization factor) and $k = \frac{\$8}{4} = \$2$ (the unit cost). A unit of oats is of size $s_o = \frac{\$2}{\$2 \text{ per pound}} = 1$ pound, and a unit of granola is of size $s_g = \frac{\$2}{\$6 \text{ per pound}} = \frac{1}{3}$ of a pound.

Alice's marginal returns for all units are listed in Table 1. Because her unit marginal returns are decreasing, Alice can find an optimal solution to this discretized problem by allocating her money in a greedy manner to uses in this decreasing order. In this situation, Alice should allocate her four \$2 bills as follows: spend her first \$6 on oats, spend her last \$2 on granola.

Table 1. Oats and Granola at a bulk food store. UC stands for unit cost, UMV for unit marginal value, and UMR for unit marginal return.

	Oats				Granola			
UC	lbs	Value	UMV	UMR	lbs	Value	UMV	UMR
k	x_o	$f_o(x_o)$	$f_o(x_o) - f_o(x_o - s_o)$	$\nu_o(x_o)$	x_g	$f_g(x_g)$	$f_g(x_g) - f_g(x_g - s_g)$	$\nu_g(x_g)$
2	1	18	18	**9**	0.333	7.67	7.67	**3.83**
2	2	32	14	**7**	0.667	14.67	7	3.5
2	3	42	10	**5**	1	21	6.33	3.16
2	4	48	6	3	1.333	26.67	5.67	2.83

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 has to be allocated in discrete amounts (e.g., one dollar or one cent). If the store accepts half dollars, then Alice should spend \$6.50 on oats and \$1.50 on granola, which yields total value ~ 49.69; if the store accepts quarters, then Alice should spend \$6.25 on oats and \$1.75 on granola, which yields total value ~ 49.71. The value of the latter solution is within one cent of optimal. We formalize this intuition presently.

2.3 Main Theorem

Given an instance of a NLK, let $OPT_{con}(B)$ denote the value of an optimal solution to this problem, given a budget of B; and let $OPT_{dis}(B, K)$ denote the value of an optimal solution to the corresponding discretized problem with discretization factor K. We prove that $OPT_{dis}(B, K)$ approximates the value of $OPT_{con}(B)$. Specifically, $OPT_{dis}(B, K)$ is within a factor of $1 - \epsilon$ of $OPT_{con}(B)$.

Theorem 1. *Assuming the f_is are concave and nondecreasing, the g_is are convex and nondecreasing, and the f_i's and g_i's are continuous,*

$$OPT_{dis}(B) \geq OPT_{con}(B, K) \left(1 - \frac{2n}{K}\right)$$

A proof of this theorem appears in Greenwald, *et al.* [6]. The intuition for the proof is as follows. We introduce an intermediate solution that optimally solves a continuous NLK with a slightly different budget B'. The crucial property of this intermediate solution is that it has a value $OPT_{con}(B')$ that is close to both $OPT_{dis}(B, K)$ and $OPT_{con}(B)$. A bound on the distance between $OPT_{dis}(B, K)$ and $OPT_{con}(B)$ is then obtained by adding the distance between $OPT_{dis}(B, K)$ and $OPT_{con}(B')$ to the distance between $OPT_{con}(B')$ and $OPT_{con}(B)$.

Inputs:

discretization factor K

value functions f_i

cost functions g_i

Outputs:

a vector q of quantities consumed, one per use

1. for each use i
   (a) initialize $q_i = 0$
   (b) insert i with priority $\nu_i(s_i) = \frac{f_i(s_i)}{g_i(s_i)}$ into a priority queue Q
2. for $t = 1$ to K
   (a) pop off of Q a use j with the highest priority
   (b) increment q_j by s_j
   (c) insert j into Q with priority $\nu_j(q_j + s_j) = \frac{f_j(q_j+s_j)-f_j(q_j)}{g_j(q_j+s_j)-g_j(q_j)}$
3. return q

Fig. 1. A FPTAS for NLK. The algorithm runs in time $O(\frac{1}{\epsilon} n \log n)$.

A maximization problem admits a Fully Polynomial Time Approximation Scheme if for any $\epsilon > 0$ there exists an algorithm whose run time is polynomial in the input size and $\frac{1}{\epsilon}$ that finds a solution whose value is within a factor of $1 - \epsilon$ of the optimal. Our theorem implies that the NLK admits a FPTAS with $\epsilon = \frac{2n}{K}$ and running time $O(\frac{1}{\epsilon} n \log n)$. The algorithm is shown in Figure 1. The first loop runs in time $O(n)$ and the second in time $O(K \log n)$; hence the entire algorithm runs in time $O(n + K \log n) = O(n + \frac{2n}{\epsilon} \log n) = O(\frac{1}{\epsilon} n \log n)$.

In the next section, we define a tractable approximation of the TAC SCM bidding problem called expected bidding. We note that this problem reduces to a NLK problem with g_i linear and the f_is satisfying the usual assumptions. Hence, our discretization technique, followed by an application of the greedy algorithm, can be used to compute an approximate solution to this problem.

3 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, 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 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

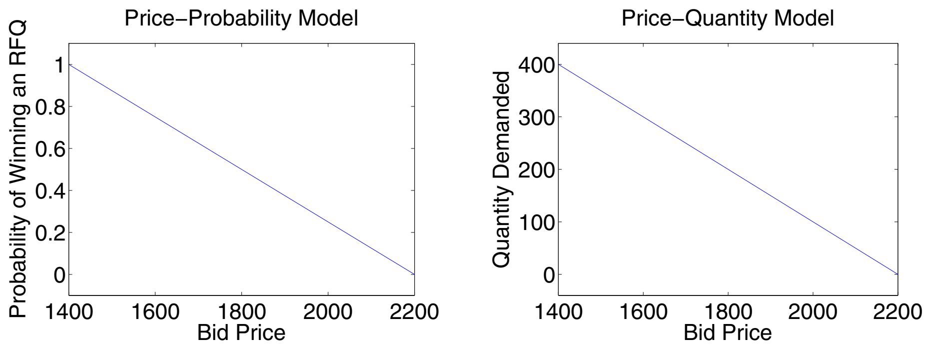

Fig. 2. (a) Sample price-probability model. (b) Sample price-quantity model.

the other agents. More specifically, each day the customers send *requests for quotes* (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.

Assuming a suitable model of market dynamics—in particular, the current and future prices at which components can be bought and finished products sold—a TAC SCM agent faces three core decision problems [2]: *procurement* of components from suppliers, *bidding* on customer requests for quotes (RFQs), and *scheduling* of factory production and deliveries. In this paper, we focus on the bidding problem, which subsumes the scheduling problem. A study of how our methods extend to procurement remains for future work. Before detailing our approach to bidding in TAC SCM, we discuss the model of market dynamics on which our formulation of this decision problem is based.

3.1 Price-Probability Models

In a marketplace with indistinguishable products, a seller hoping to adjust its market share can do so only by changing its price. Such a seller is likely to gather relevant historical data for use in predicting the market shares that correspond to various price settings. Following Benisch *et al.* [2], we assume that this prediction task has already been completed, and the agent is already endowed with a *price-probability* model that reports the probability of winning an order for each possible bid on current and future RFQs.

Rather than specifying a price-probability model for each individual RFQ, we partition the set of RFQs according to their defining characteristics so that we can obtain a richer set of price-probability models (we are assuming that models built using more data can make more accurate predictions). In TAC SCM, a natural partitioning of the set of RFQs is by SKU type and due date. We refer to each element of such a partition as a *market segment*.

Figure 2(a) depicts the price-probability model defined by this equation:

$$p(x) = \frac{2200 - x}{800} \qquad 1400 \leq x \leq 2200 \tag{9}$$

This model asserts that a bid of 2200 has no chance of winning (it is the reserve price above which there is no demand), whereas a bid of 1400 is guaranteed to win (it is the price below which there is no supply). In between, at a price of 1800, say, a bid wins 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 in a market segment.

3.2 The Expected Bidding Problem

The N-day stochastic bidding problem is formulated as a recursive stochastic program in Appendix A. A tractable approximation of 1-day stochastic bidding, called *expected bidding*, was considered in Benisch *et al.* [2]. 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 bids on $|R|$ RFQs results in exactly one set of (partial) orders, instead of $2^{|R|}$, as in Equation 16.

Collapsing the stochastic content of a price-probability model into deterministic statistics in the form of partial orders is achieved by scaling the model by the demand in the corresponding market segment. We call the ensuing models *price-quantity* models. Recall the price-probability model depicted in Figure 2(a). Assume this market segment consists of 80 RFQs of 5 SKUs each, 400 SKUs in total. Since a price of 1800 wins with probability 0.50, at this same price, an agent can expect to win 200 SKUs worth of demand (see Figure 2 (b)).

The objective in expected bidding is to find a set of bids x, one per market segment i, that maximizes expected revenue, subject to the constraint that expected production does not exceed available capacity, given, (i) for each market segment, a price-quantity model $h_i(x_i)$ that maps bid prices into quantities—i.e., expected market share; (ii) the total available production capacity C; and (iii) 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:

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} h_i(x_i)\, x_i \tag{10}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} c_i\, h_i(x_i) \leq C \tag{11}$$

where $x_i \in \mathbb{R}$ is the bid price in market segment i. Observe that this problem is an instance of the NLK with $f_i = h_i(x_i)\, x_i$ and $g_i = c_i\, h_i(x_i)$.

Assuming h is invertible, so that the price-quantity model is a 1 to 1 mapping between bid prices and expected market shares, selecting a bid is equivalent to selecting a quantity. In this case, by renaming variables (in particular, letting $x'_i = h_i(x_i)$), we can solve the expected bidding problem as follows:

1. Invert h.
2. Solve this mathematical program:

$$\max_{x'_1,\ldots,x'_n} \sum_{i=1}^{n} x'_i \, h_i^{-1}(x'_i) \tag{12}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} c_i x'_i \leq C \tag{13}$$

   where $x'_i \in \mathbb{R}$ is the desired share of market segment i.
3. Bid $h^{-1}(x')$.

Hence, we have reduced the expected bidding problem to solving an instance of the NLK in which uses are market segments, the knapsack's capacity (or the budget) is the factory's capacity, the value functions $f_i(x'_i) = x'_i \, h_i^{-1}(x'_i)$, and the cost functions $g_i(x'_i) = c_i x'_i$. Assuming the f_is are concave and nondecreasing, the results we derived in Section 2 are directly applicable. In particular, our Theorem 1 bounds the quality of a solution to Equations 12 and 13; since the value of such a solution is equal to value of a solution to Equations 10 and 11, Theorem 1 similarly bounds the quality of a solution to expected bidding. An example of this reduction follows.

Example. Consider an instance of the expected bidding problem in which $c_i = 5$ cycles and

$$h_i(x_i) = \frac{2200 - x_i}{2} \qquad 1400 \leq x_i \leq 2200 \tag{14}$$

for some market segment i. We invert this price-quantity model, which yields the following "quantity-price" model:

$$h_i^{-1}(x'_i) = 2200 - 2x'_i \qquad 0 \leq x'_i \leq 400 \tag{15}$$

(Note that $f_i(x'_i) = x'_i \, h_i^{-1}(x'_i) = x'_i \, (2200 - 2x'_i)$ is concave and nondecreasing on the interval $0 \leq x'_i \leq 400$; hence, our results from Section 2 apply.)

Next, we apply the discretization technique to market segment i. If the factory capacity $C = 4000$ cycles and the discretization factor $K = 10$, then the unit cost $k = 400$ cycles and the unit size $s_i = \frac{400 \text{ cycles}}{5 \text{ cycles per SKU}} = 80$ SKUs. By querying the quantity-price model in increments of 80 SKUs, we can generate a list of prices at various incremental quantities. Each revenue is then the product of a price and a corresponding quantity. Unit marginal revenues are the incremental differences in revenue corresponding to the incremental quantities. Finally, unit marginal returns are unit marginal revenues divided by unit costs.

Table 2. Unit Marginal Returns on Market Segment i

Unit Cost k	Quantity x'_i	Price $h_i^{-1}(x'_i)$	Revenue $f_i(x'_i)$	Unit Marginal Revenue $f_i(x'_i) - f_i(x'_i - s_i)$	Unit Marginal Return $\nu_i(x'_i)$
400	80	2040	163200	163200	408
400	160	1880	300800	137600	344
400	240	1720	412800	112000	280
400	320	1560	499200	86400	216
400	400	1400	560000	60800	152

The complete list of unit marginal returns in this example is shown Table 2. These unit marginal returns could have been computed directly using Equation 6. For example, the marginal return on the second unit in market segment i is:

$$v_{i2} = \frac{f_i(2s_i) - f_i(s_i)}{g_i(2s_i) - g_i(s_i)} = \frac{h_i^{-1}(2s_i)\,2s_i - h_i^{-1}(s_i)\,s_i}{k} = \frac{h_i^{-1}(160)\,160 - h_i^{-1}(80)\,80}{(5)(80)} = 344$$

Based on the unit cost c_i and the quantity-price model $h_i^{-1}(x_i)$, we can create such a list of unit marginal returns in each market segment.[2] After doing so for all market segments (i.e., after discretizing the problem), we compute a greedy solution to the ensuing discrete problem. We input the output of this solution, namely a vector of quantities x', to the quantity-price model to obtain a vector of bids, which is our solution to the expected bidding problem.

3.3 Marginal Bidding in TAC SCM

There is one important aspect of the TAC SCM bidding problem that we have not thoroughly emphasized, namely that the bidding problem spans multiple days. In this section, we describe how we extend our solution to the 1-day expected bidding problem to the multi-day setting. We call the resulting heuristic *Marginal Bidding*. One of the strengths of a greedy approach to bidding in TAC SCM is that it is natural, and hence easily extensible.

The extension of a greedy solution from the 1-day to a multi-day problem requires an additional parameter. The number of days in the multi-day problem may be too large for even a greedy bidder to reason about within the available time frame. We define the bidder's *window size* W to be the number of days of demand and production considered when making decisions. For example, a window size of 17 means that the bidder can schedule production on 17 days, namely today and on 16 future days. In doing so it considers the current set of RFQs as well as an anticipated set of RFQs for 16 future days. These RFQs are partitioned into market segments by SKU and due date.

When the window size W is large, a large value of K can increase the Marginal Bidder's run time to an unacceptable level. On the other hand, a small value of

[2] Note that in our implementation we do not explicitly create lists of all unit marginal returns. Since unit marginal returns are nonincreasing, we need only identify the next highest unit marginal return in each market segment. See Figure 3 for details.

K can result in a unit size s_i so large that it hinders the algorithm's ability to make short-term decisions at a fine enough granularity. Since we are interested in invoking the Marginal Bidder with large window sizes, we implicitly vary K across market segments (although the theorem presented in Section 2.3 is only applicable when K is constant across market segments). More specifically, the Marginal Bidder also takes as input a unit size s_i for each market segment i, with each s_i proportional to the size of i's range of due dates.

A detailed description of the Marginal Bidder appears in Figure 3. At a high level, first it greedily fulfills outstanding orders in nonincreasing order of revenue per cycle; second it greedily schedules production of units of the various market segments in nonincreasing order of unit marginal returns; third it bids the price associated with the quantity of demand met in each market segment. Note that bids on all RFQs in a single market segment are equal.

For simplicity, the algorithm we present does not consider component constraints, but it can easily be extended to do so. The Marginal Bidder would have to take as input current component inventory and anticipated daily component arrivals, and could only schedule units for production when sufficiently many components were predicted to be on hand. After scheduling, the corresponding components would be removed from inventory by decrementing the daily component inventory backwards from the production date.

Scheduling. To schedule outstanding orders and incremental quantities of market segments for production, there are two natural approaches. First, we can schedule *as soon as possible*, meaning that production is scheduled forwards from the current day. Because the Marginal Bidder schedules greedily, using this method, the most profitable products are produced on the current day, and less profitable products are scheduled for production on subsequent days.

An alternative approach is to schedule for production *as late as possible*, which means that production for an order or an incremental quantity of a market segment is scheduled backwards from its due date. While this approach allows for production decisions to be postponed until more of the uncertainty in the market is resolved, it also allows for empty or near-empty production schedules on the current day, which can be risky. In particular, if demand or prices unexpectedly increase, the Marginal Bidder may wish it had more finished goods on hand.

The Marginal Bidder uses both of these approaches, scheduling outstanding orders as soon as possible because of the penalties incurred for defaulting, and scheduling incremental quantities of market segments as late as possible in order to allow for greater flexibility in bidding decisions.

While the Marginal Bidding algorithm is easy to understand and implement, it behooves us to demonstrate that its performance is acceptable, particularly with market segments and hence units of varying sizes, which renders our theory inapplicable. This is the subject of the remaining sections of this paper.

Inputs:

a window size W
the factory production capacity C
M market segments, each one i characterized by:
a product, a quantity, a range of due dates, a unit size s_i,
an invertible price-quantity model $h_i(x')$, the number of cycles c_i
required to manufacture 1 of i's product, and a "successfully-scheduled-quantity" q_i initialized to 0
a set of outstanding orders
product inventory

1. sort outstanding orders in nonincreasing order by revenue per cycle
2. for each outstanding order (traversing the list of orders in sorted order)
   (a) use product inventory to fulfill as much of the order as possible
   (b) schedule the rest of the order for production as soon as possible within the scheduling window W
   (c) if the order still cannot be satisfied entirely, undo the inventory and production schedule changes made in the last two steps
3. set j to be the market segment with the highest unit marginal return: i.e.,

$$\begin{aligned} j &= \operatorname*{argmax}_i \left(\nu_i(q_i + s_i)\right) \\ &= \operatorname*{argmax}_i \left(\frac{f_i(q_i + s_i) - f_i(q_i)}{g_i(q_i + s_i) - g_i(q_i)}\right) \\ &= \operatorname*{argmax}_i \left(\frac{(q_i + s_i)h_i^{-1}(q_i + s_i) - q_i h_i^{-1}(q_i)}{c_i s_i}\right) \end{aligned}$$

4. while $\nu_j > 0$
   (a) take up to s_j units of the product associated with j from product inventory
   (b) schedule the remaining units for production as late as possible but before the median due date associated with j and within the scheduling window W
   (c) if s_j units cannot be supplied, then set $\nu_j(q_j+s_j) = -1$ and undo the inventory and production schedule changes made in the last two steps
   (d) otherwise, if s_j units can be supplied, increment q_j by s_j
   (e) set j to be the market segment with the highest unit marginal return
5. for each market segment i
   (a) bid the price at which the agent expects to win the quantity it successfully scheduled: i.e., bid $h_i^{-1}(q_i)$

Outputs: A bid for each market segment, and hence for all the RFQs that comprise that market segment. Note that bids on all RFQs in a single market segment are equal.

Fig. 3. Marginal Bidding Algorithm

4 Experimental Results

In this section, we report on experiments designed to compare the performance of four bidding algorithms with varying abilities to reason about the future, an ILP bidding heuristic (see Benisch *et al.* [2]) and three variations on the Marginal

Bidding heuristic developed in this paper. We expect the Marginal Bidders to compute bids faster than the ILP, and we expect this speed to enable them to consider larger windows into the future, which should lead to higher revenues than the ILP under some market conditions (and never lead to lower revenues). We test these conjectures on instances of TAC SCM bidding in a simulator we built that tests individual agents in isolation by generating decision-theoretic simplifications of the game-theoretic problems TAC SCM agents face.

4.1 Test Suite

We tested an integer linear programming solution with a 1 day window (ILP), meaning it did not reason about any future demand beyond the current RFQs and outstanding orders arriving each day. We compared this ILP with three variations of the marginal bidder: a marginal bidder with a 17-day[3] window (MB-17), a marginal bidder with a full-game window (MB-Full), and a marginal bidder with a hybridization of the two that considers the full game window, but does so at a coarser granularity as it reasons further into the future in order to keep its run time in check (MB-Coarse).

The 17 Day (MB-17) and full-game (MB-Full) bidders partition demand (i.e., the set of current and future RFQs) into market segments by SKU type and due date, and the size of a unit in each market segment is 1 product. The hybrid full-game bidder (MB-Coarse) also divides demand up by SKU type and due date. For the first 17 days, it considers each due date separately, but beyond the initial 17 days it divides demand into increasingly larger chunks, whose due-date ranges grow by powers of 1.8.[4] For the coarse bidder, each market segment's unit size is 1 product multiplied by the number of days in that segment.

Since each TAC SCM day is 15 seconds, and a bidding policy is one of many decisions an agent must make each day, it may not be wise for an agent to allot too much of its daily run time to bidding alone. We thus study a likely TAC SCM situation in which the bidder is only given 5 seconds to formulate its daily bidding policy. The full-game Marginal Bidder often requires more than 5 seconds per day to compute its policy, so it is not a feasible TAC SCM bidder, but we include it in this discussion for benchmarking purposes.

In order to reach a reasonable solution within the allotted 5 seconds, the ILP dynamically calculates an appropriate degree of discretization using a formula that was empirically determined to minimize the ILP's distance from optimality within a 5 second window. The equation for the number of price points is 2300/(# of RFQs + # of Orders). An ILP with a run time of up to 15 seconds and additional price points was also tested, but did not yield significant gains.[5]

[3] We chose 17 as the default window size because it is the last day on which a current RFQ with the latest possible due date can be filled in TAC SCM.

[4] For example, SKUs due on days 18-19 are grouped together ($1.8 \approx 2$), as are SKUs due on days 20-22 ($1.8^2 \approx 3$), and days 23-28 ($1.8^3 \approx 6$), and so on.

[5] An ILP with a 2-day window was also tested, as was one with a 17-day window and constrained capacity (2000 cycles on day 1 and 2000 cycles on days 2 through 17). Again, these variants did not yield significant gains.

4.2 Experimental Design

Recall that in TAC SCM each agent submits its bids to a reverse auction, so that an RFQ is awarded to the agent that bids the lowest price below the reserve price. Using our simulator, we tested our bidding algorithms in isolation, not against other bidding agents, as would be the case in a true reverse-auction setting. The simulator simply awarded contracts by transforming each offer into an order with a certain probability, namely that which is associated with the bid price under the price-probability model for the relevant market segment. Hence, we simulated the stochastic bidding problem, although our heuristic solutions are approximate solutions to the expected bidding problem.

In our experiments, agents were endowed with perfect price prediction: i.e., the various price-probability models (one per market segment per simulation day) were shared between the agent and the simulator. Regarding demand, the number of customer RFQs of each SKU type scheduled to arrive each day was broadcast before the simulations began. Then, on each simulation day, the agents received a set of current RFQs whose quantities and due dates were sampled from the distributions outlined in the TAC SCM game specifications, and they assumed that the quantity and due date associated with each of the future RFQs were the means of the same distributions.[6] Reserve prices were also known to the agents; they were built in to the price-probability models.

We tested our bidders by running 25 simulations of 100 day games under three families of market conditions: (i) constant: i.e., conditions on one day are reflective of the conditions on the next; (ii) gradually changing conditions; and (iii) sudden shifts, including demand or price shocks. Under the non-constant conditions we examine situations of rising demand and price. Falling demand and price conditions are not presented, but produce similar results.

For simplicity, in these simulations we assume infinitely many components. Introduction of component constraints does not appear to significantly alter the relative performance of our bidders.

4.3 Constant Conditions

In our first set of market conditions, we compare the bidders under constant demand and price. Presented here are steady conditions of high demand, defined as 20 RFQs per SKU type per day, which is the maximum possible according to the TAC SCM game specification, and low demand, defined as 5 RFQs per SKU type per day, the lowest possible. Prices in this experimental setup range linearly from 50% to 125% of the SKU base price.

Under such conditions, we should expect to see no particular advantage to planning for the future, since an optimal solution to the entire game can be contructed by concatenating a sequence of optimal solutions, one per day, computed

[6] The reason for drawing a distinction between the quality of the predictions of the number of RFQs of each SKU type and their attributes is: the former is somewhat predictable in TAC SCM—it is dependent on history (see, for example, Kiekintveld *et al.* [9])—while the latter is not.

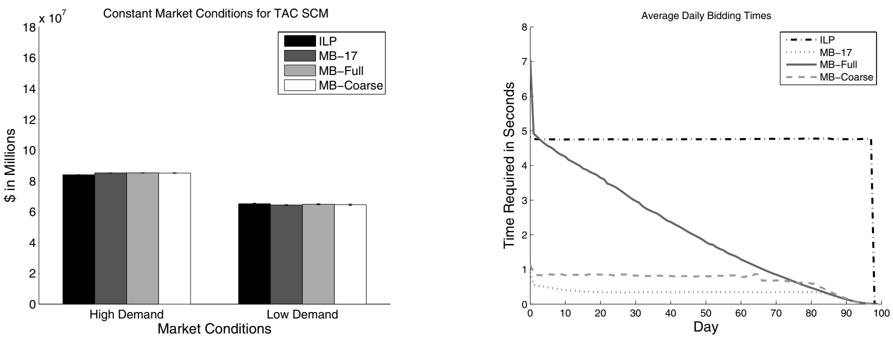

Fig. 4. (a) Revenue from deliveries under constant market conditions. (b) Average daily bidder times in high demand conditions. Low demand bidder times were similar.

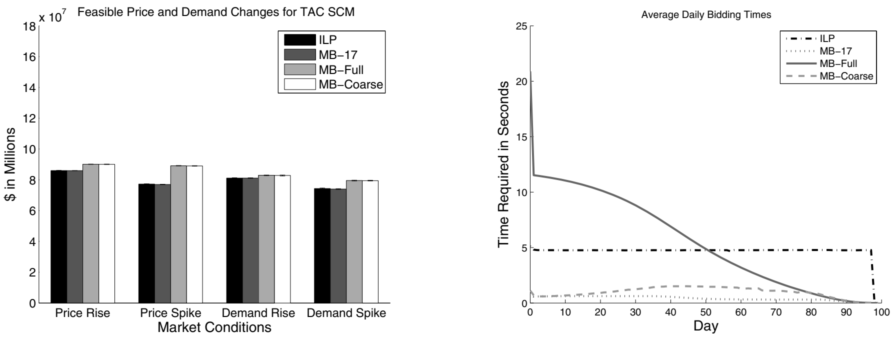

Fig. 5. (a) Revenue from deliveries under feasible SCM market conditions. (b) Average daily bidder times in Price Rise conditions. Other market conditions had similar run times.

for each day in isolation. Indeed, in terms of revenue, all the bidders are competitive with one another under these conditions (see Figure 4(a)). Note however that MB-17 and MB-Coarse arrive at their solutions an order of magnitude faster than the ILP or the MB-Full bidding algorithms (see Figure 4(b)).

4.4 Shifting Conditions

More interestingly, market conditions can change over the course of a TAC SCM game, either steadily as in a market adjustment or suddenly as in a demand or price shock. In our next experimental setup, demand is initialized to 5 RFQs per SKU per day, and prices range linearly from 50% to 75% of the SKU base price. We then considered shifts to 20 RFQs per SKU per day and prices ranging from 100% to 125% of the base prices by day 50. These shifts are representative of the magnitude of changes an agent might observe while playing a typical TAC SCM game. These changing market conditions were tested both as steady linear accumulations from day 1 to day 100 and as abrupt surges on day 50. In our price-shifting simulations demand is held constant; in our demand-shifting simulations price is held constant.

As expected, those bidders with more extensive knowledge of the future (MB-Full, MB-Coarse) are able to exploit the mid-game surges by dedicating production from today to future demand when conditions are more favorable. Bidders with a shorter window (ILP, MB-17) are unable to plan far enough ahead to take advantage of the upcoming shifts, and hence accumulate less revenue over the course of the game. In addition to the additional revenue gained by exploiting its knowledge of the future, the MB-Coarse bidder continues to run in substantially less time than the ILP. See Figure 5.

The advantages of a larger window are more pronounced under those market conditions in which the shift in demand or price comes as a sudden spike rather than as a steady rise. When demand or prices rise gradually, even an agent with a small window is aware that tomorrow's market conditions are slightly more profitable than today's, and can reserve some inventory for future sales. However, when demand or price spikes suddenly, an agent is not aware of more desirable future market conditions until the spike falls within its window.

Because one of our simplifying assumptions for these simulations is that agents have perfect models of future demand and price, it is encouraging that MB-Coarse performs just as well as MB-Full. Their similar performance suggests that the benefits of looking into the future may still be realized by agents with more realistic but less accurate models.

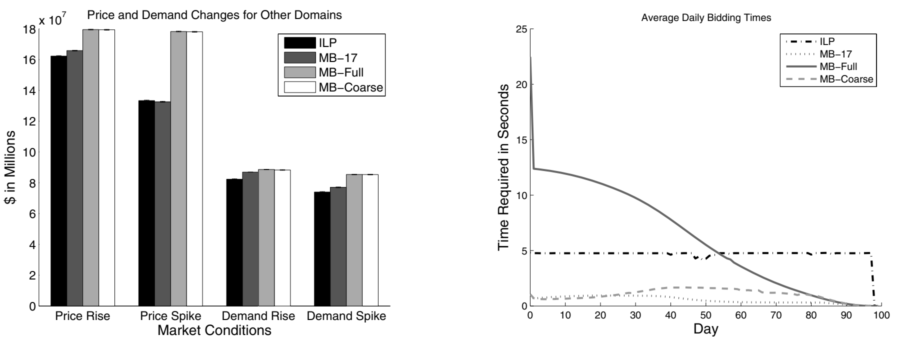

Fig. 6. (a) Revenue from deliveries in extreme market conditions. (b) Average daily bidder times in Price Rise conditions. Other market conditions had similar run times.

4.5 Extreme Conditions

Within the context of TAC SCM, the previous experimental setup characterizes shifts from one extreme set of realistic conditions to another, and the gains resulting from knowledge of the future are modest. However, it is easy to envision markets that are more naturally volatile or are subject to large seasonal trends in demand. The greater the extent to which market conditions vary across time,

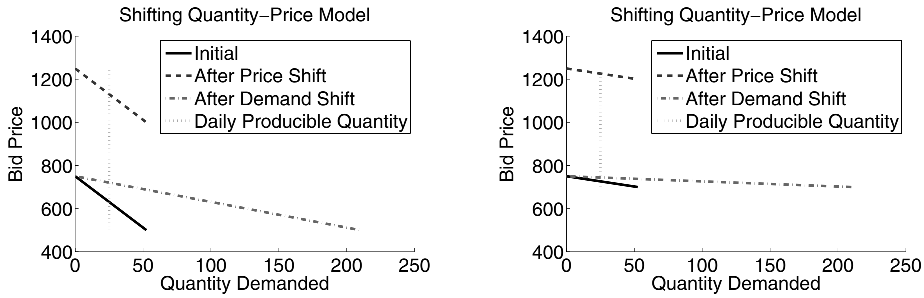

Fig. 7. (a) Sample quantity-price models before any shift, after a price shift, and after a demand shift. To illustrate the constraining effects of production capacity, also shown is a sample daily producible quantity. In our experiments, price shifting conditions result in higher revenues than demand shifting conditions, and thus knowledge of a future price shift is more valuable than knowledge of a future demand shift. (b) For quantity-price models with flat slopes, predicting future demand is not very important.

the greater the opportunity for bidders able to consider a larger window into the future to earn greater profits. In order to demonstrate this effect, we present a second set of simulations assuming shifting market conditions, but the shifts are of greater magnitudes. In particular, demand surges from 5 to 40 RFQs per SKU per day, and price rises from $[50\%, 75\%]$ to $[200\%, 250\%]$ of the base prices, again as both an interpolated steady rise and as an overnight jump.

With no significant changes in run time (compare Figures 5(b) and 6(b)), the marginal bidders are able to exploit the extreme changes in market conditions, and in particular the bidders with larger windows (MB-Full and MB-Coarse) are able to earn even greater profits (see Figure 6(a)). Also of interest is the relative impact of demand changes versus price changes. We observe a more pronounced impact when considering knowledge of the future under price-changing conditions for two reasons.

First, because of capacity constraints, an agent can only produce a limited quantity of each product on each day. Hence, an increase in demand does not necessarily translate into an increase in the number of finished products. So even if a demand shift results in higher prices, revenues need not increase substantially, particulary in comparison to the revenue increase associated with a price increase (see Figure 7(a)). If the magnitude of the price shifts in our experiments were reduced, or if production capacity were increased, stockpiling products until a demand shift could be as worthwhile as stockpiling products until a price shift.

The second factor that mitigates the advantage of knowledge of the future in conditions of shifting demand is the relatively flat slopes of our quantity-price curves. With flatter slopes, the difference in revenue between prices on the initial curve and prices on the curve after a demand shift is small (Figure 7(b)). Thus it matters less if the agent stockpiles products for the future, and in turn it matters less if the agent has any knowledge of the future. If the quantity-price curves had steeper slopes, knowledge of the future in conditions of shifting demand would likely prove more valuable than our current experiments suggest.

5 Related Work

The NLK problem, also known as the Nonlinear Resource Allocation problem, is well-studied. The interested reader is referred to Patrikson [14] for a recent survey, which includes a number of algorithms that solve various formulations of the NLK. One feature of the approach described here is that we construct a solution incrementally; this makes it easy to check if the solution remains feasible under more general conditions: e.g., when not only capacity but also scheduling or component constraints are present. Also, unlike other techniques (e.g., Bretthaur and Shetty [4]), our algorithm does not rely on value or cost functions being differentiable or having closed-form representations (although our theoretical results do not hold under these more general assumptions).

Most closely related to our work is the work of Hochbaum [8]. In her study of the NLK, she employs a discretization technique that generalizes the one presented here. Correspondingly, the discretized problem we construct is a special case of her *simple allocation problem*; in our problem, all variables are binary rather than non-negative and integer-valued. Like us, she solves her discrete problem greedily, and, invoking work in a related paper [7], she connects this greedy solution back to an optimal solution to the original (continuous) NLK. Her results apply in the special case in which the g_is are linear. Our main theorem applies more generally; in particular, the g_is may be convex.

Benisch *et al.* [3] reduce a probabilistic pricing problem (akin to 1-day expected bidding) to the NLK, and present an ϵ-optimal solution to this problem assuming diminishing marginal returns. Although they demonstrate that their algorithm can be efficient in practice, they provide no theoretical guarantees on its run time. Also, their algorithm is not incremental, so it is not immediately obvious how to extend it to apply to problems with additional constraints.

The TacTex team developed a greedy bidding agent for TAC SCM along the lines of the Marginal Bidder presented here, with a few subtle distinctions [13]. TacTex is initialized to bid reserve prices on each RFQ, and then it iteratively reduces its bids according to a 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). No theoretical guarantees validating their approach are discussed.

Finally, researchers at the Cork Constraint Computation Center implemented an ILP approach to bidding in a constraint-based TAC SCM agent, Foreseer [5]. Not unlike the expected bidder posited in Benisch *et al.* [2], Foreseer uses profit as the objective function, bid prices as the decision variables, and constraints based on factory capacity, component availability, and reserve prices.

6 Summary and Future Work

In this paper, we described a technique for solving the NLK by converting it into a (discrete) simple allocation problem that can be solved greedily. Our theoretical results establish that the greedy solution to the resulting simple allocation problem is a FPTAS for the NLK. Although more complicated algorithms with better run times are known, our simple incremental solution affords us extra flexibility. In particular, the greedy algorithm extends easily into the Marginal Bidding heuristic, which solves an extended version of the NLK with natural scheduling and component constraints.

Our ultimate goal is to develop a scalable bidding algorithm that 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 Bidders avoid this complexity and appear to be more readily extensible to the procurement problem. However, 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 return of another. Interdependencies arise naturally in procurement because components are shared among SKU types.

Despite the game-theoretic nature of bidding in TAC SCM, our focus here was simply on a decision-theoretic (stochastic) optimization problem, not on game-theoretic equilibrium calculations. The enormity of the decision space in TAC SCM renders 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. In the near future, we plan to test the robustness of our algorithms to imperfect modeling of future prices and demand. 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

The authors would like to thank Warren Schudy for his input on the statement of our main theorem. We are also grateful to an anonymous reviewer for his or her comments. This research was supported by NSF Career Grant #IIS-0133689.

References

1. Arunachalam, R., Sadeh, N.M.: The supply chain trading agent competition. Electronic Commerce Research and Applications 4(1), 66–84 (2005)
2. Benisch, M., Greenwald, A., Grypari, I., Lederman, R., Naroditskiy, V., Tschantz, M.: Botticelli: A supply chain management agent. In: Third International Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1174–1181 (July 2004)
3. Benisch, M., Andrews, J., Sadeh, N.: Pricing for customers with probabilistic valuations as a continuous knapsack problem. In: ACM International Conference Proceeding Series, vol. 156, pp. 38–46 (2006)
4. Bretthauer, K.M., Shetty, B.: The nonlinear resource allocation problem. Operations Research 43, 670–683 (1995)

5. Burke, D.A., Brown, K.N., Koyuncu, O., Hnich, B., Tarim, A.: Foreseer: A constraint based agent for tac-scm. In: Trading Agent Competition - Supply Chain Management, IJCAI 2005, Edinburgh, Scotland (August 2005)
6. Greenwald, A., Naroditskiy, V., Odean, T., Ramirez, M., Sodomka, E., Zimmerman, J., Cutler, C.: Marginal bidding: An application of the equimarginal principle to bidding in tac scm. Technical Report CS-08-04, Brown University (2008)
7. Hochbaum, D.S., Shanthikumar, J.G.: Convex separable optimization is not much harder than linear optimization. J. ACM 37(4), 843–862 (1990)
8. Hochbaum, D.S.: A nonlinear knapsack problem. Operations Research Letters 17, 103–110 (1995)
9. Kiekintveld, C., Wellman, M.P., Singh, S., Estelle, J., Vorobeychik, Y., Soni, V., Rudary, M.: Distributed feedback control for decision making on supply chains. In: Fourteenth International Conference on Automated Planning and Scheduling (to appear, 2004)
10. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, New York (1995)
11. Tarasov, S.P., Kozlov, M.K., Khachian, L.G.: Polynomial solvability of convex quadratic programming. Doklady Akad. Nauk SSR 5, 1051–1053 (1979)
12. Odean, T., Naroditskiy, V., Greenwald, A., Donaldson, J.: Marginal bidding: An application of the equimarginal principle to bidding in tac SCM. In: AAA-2007 Workshop on Trading Agent Design and Analysis (July 2007)
13. Pardoe, D., Stone, P.: Bidding for customer orders in tac SCM. Agent Mediated Electronic Commerce 4, 143–157 (2005)
14. Patriksson, M.: A survey on the continuous nonlinear resource allocation problem. European Journal of Operational Research 127(1), 1–46 (2008)

A The Stochastic Bidding Problem

The bidding problem posed here is intended to model the bidding problem that agents face in TAC SCM. 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 the number and variety of RFQs that will arrive on each future day of the game.

A decision-theoretic version of the TAC SCM bidding problem, under the aforementioned assumptions, can be formulated naturally as a recursive stochastic program. We have not seen this program appear elsewhere in the literature (except in Odean *et al.* [12]), so we present it here, using the notation explained in Figure 8.

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
$\pi_i(t)$ revenue (minus penalty) for delivering order i on day t
zero if t is past order's due date
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

Fig. 8. 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 17).

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 18 ensures that there are enough products in inventory for today's delivery schedule. Equation 19 ensures that today's production schedule does not consume more cycles than the daily capacity.

The base case (Equation 20) of the recursion pertains to the last day. Orders can be scheduled for delivery but there is no production or bidding.

if $0 \leq t < N$,

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

$$\sum_{Q \in 2^{|R|}} \Pr(Q) E_{R'|h} \left[F(b - a + y, O \cup Q, R', h \cup R, t + 1) \right] \tag{16}$$

subject to:

$$\Pr(Q) = \prod_{r \in Q} p(r, x_r) \prod_{r \notin Q} (1 - p(r, x_r)) \tag{17}$$

$$a_j = \sum_{i | i \in O, d_{ij} = 1} z_i q_i \quad \forall j; \quad a \leq b \tag{18}$$

$$\sum_j y_j c_j \leq C \tag{19}$$

if $t = N$,

$$F(b, O, R, h, t) = \max_{z} \sum_{i \in O} z_i \pi_i(t) \tag{20}$$