Bidding under Uncertainty in Simultaneous Auctions

Amy Greenwald
Department of Computer Science
Brown University, Box 1910
Providence, RI 02912
amy@cs.brown.edu

Abstract

This paper describes an experimental study of bidding strategies that were utilized in the international Trading Agent Competitions, TAC 2000–02. Three bidding principles are advocated: (i) combinatorial decision-making—reasoning about *sets* of goods, rather than individual goods; (ii) *coherent* marginal utility bidding: *i.e.*, bidding marginal utilities on a coherent subset of goods; and (iii) *hedging*, thereby explicitly accounting for the inherent uncertainty in e-commerce applications.

1 Bidding Decisions

One of the key challenges autonomous bidding agents face is to determine how to bid on complementary and substitutable goods—i.e., goods with combinatorial valuations—in simultaneous, or parallel, auctions (SAs).

Complementary goods are goods with superadditive valuations: $v(A\bar{B}) + v(\bar{A}B) < v(AB)$. Substitutable goods are goods with subadditive valuations: $v(A\bar{B}) + v(\bar{A}B) > v(AB)$. It is difficult (if not impossible) to assign independent valuations to complementary goods, which can be worthless in isolation, or substitutable goods, which can be worthwhile only in isolation. Thus, the simple bidding strategy "for each good x in auction x, bid its (independent) valuation" is inapplicable in this framework. This paper investigates strategies that generate alternative *bidding policies*—mappings from goods to bids—in SAs, assuming combinatorial valuations.

Rather than attempt to reason about the valuations of goods independently, bidding agents that operate in this framework can reason about *marginal* valuations, or the valuation of a good x relative to a set of goods X. In particular, if an agent holds the goods in X, it can ask questions such as: "what is the marginal benefit of buying x?" or, "what is the marginal cost of selling x?" To do so, the agent must reason about the *set* of goods $X \cup \{x\}$ or $X \setminus \{x\}$, relative to the set X—the valuations of which are well-defined. In reasoning about sets of goods bidding agents may pose and solve questions such as the following [Boyan and Greenwald, 2001]:

1. "Given only the set of goods I already hold, what is the maximum valuation I can attain, by arranging my individual goods into sets of goods?"
2. "Given the set of goods I already hold, and given market prices and *supply*, on what set of additional goods should I place *bids* so as to maximize my utility: *i.e.*, valuation less purchase costs?"
3. "Given the set of goods I already hold, and given market prices, supply, and *demand*, on what set of goods should I place bids or *asks* so as to maximize my utility: *i.e.*, valuation plus proceeds less costs?"

The third, and most general, of these problems, which we call *completion*, provides a foundation for bidding strategies in simultaneous single- and double-sided auctions. The second problem, which we term *acquisition*, provides a foundation for bidding strategies in single-sided auctions. The first problem, *allocation*, is a special case of the others in which all goods cost either 0 or ∞, and all goods are worth either 0 or $-\infty$. These so-called bid determination problems are formally defined in [Boyan and Greenwald, 2001], where it is shown that completion can be reduced to acquisition. In view of this result, bidding decisions in this paper are discussed solely in terms of the acquisition problem.

2 Bidding Principles

This paper advocates three bidding principles for agents participating in simultaneous auctions for complementary and substitutable goods: (i) combinatorial decision-making—reasoning about *sets* of goods, rather than individual goods; (ii) *coherent* marginal utility bidding: *i.e.*, bidding marginal utilities on a coherent subset of goods; and (iii) *hedging*, thereby explicitly accounting for the inherent uncertainty in real-world e-commerce applications. To validate these principles, this paper compares several combinatorial bidding strategies in the context of the Trading Agent Competition (TAC) (see, for example, [Wellman *et al.*, 2002a]), which challenged its entrants to build autonomous bidding agents that trade in SAs for complementary and substitutable goods. These strategies, which incorporate elements of two TAC agents, RoxyBot and ATTac, include:

1. two strategies that compute the marginal utility of each good independently, much like ATTAC, one of which explicitly handles stochasticity;
2. two coherent marginal utility strategies, which solve the acquisition problem before bidding marginal utilities, inspired by ROXYBOT, one of which explicitly handles stochasticity;
3. one Monte-Carlo-simulation-based strategy, which computes an approximation of an optimal bidding policy, by generating candidate policies according to strategies 1 and 2, evaluating these candidates under random samples, and choosing the best policy.

Experimentally, it is established that algorithm 2 outperforms algorithm 1, in both certain and uncertain environments. Thus, constraining marginal utility bidding is an effective bidding heuristic. But 3 dominates both 1 and 2, since 3 generates candidate policies according to 1 and 2, and chooses the best candidate policy. The insights gained from these experiments about the design of autonomous bidding agents are applicable beyond the scope of TAC: *e.g.*, eBay is home to numerous SAs for complementary and substitutable goods.

Overview

This paper is organized as follows. Next, we describe an architecture for bidding agents, emphasizing our approach to solving the bidding problems that agents face. We proceed to show that bidding marginal utilities is suboptimal, but bidding *coherent* marginal utilities is optimal, if prices are known with certainty. Since constrained bidding is proven effective, if prices are known with certainty, we describe two generalizations of constrained bidding that are suited to decision-making under uncertainty. We present experiments with TAC agents in the final section, reinforcing the conclusions that we draw via proof and example in earlier sections.

3 Bidding Under Certainty

In constructing bidding policies—mappings from goods to prices—for multi-unit SAs, such as those that characterize TAC, we propose the following natural breakdown of bidding decisions:

1. how many copies of each good do I want?
2. for the goods I want, how much am I willing to pay?

One straightforward approach to answering these questions, which was employed by ATTAC [Stone *et al.*, 2002], is for the agent to skip question 1, and simply compute how much it is willing to pay for each copy of each good. An alternative approach, employed by ROXYBOT, is to explicitly answer question 1 before question 2; in this way, the agent is certain to bid on a "coherent" set of goods: *i.e.*, a set of goods which together comprise an optimal solution to the acquisition problem.

In this section, we compare these two approaches under two assumptions: (i) prices are exogenously determined: *i.e.*, we ignore the impact of agent behaviors on prices; and (ii) prices are known with certainty. These assumptions are clearly applicable when prices are posted. In an auction setting, we interpret these assumptions as follows: the payment rule is "pay the known price;" the winner determination rule is "the winner is the first to bid equal to or more than the price." It is as if an agent is capable of predicting all other agents' bids, and the impact of its own bids on clearing prices.

We begin by investigating ATTAC's approach. One formula for determining willingness to pay is to compute marginal utilities (MUs). This direct MU approach to answering the aforementioned bidding questions explicitly answers only question 2, but implicitly answers question 1: a willingness to pay 0 suggests that an agent wants 0 additional copies of that good.

As alluded to in the introduction, computing MUs depends on solving the acquisition problem. Let us introduce some notation, and formally define acquisition and MU. Let X denote a set of goods; let $v(X)$ denote the (combinatorial) valuation of X; let $p(x)$ denote the price of $x \in X$ and $p(X) = \sum_{x \in X} p(x)$; finally, let $u(X) = v(X) - p(X)$ denote the utility of X.

Definition 3.1 Given a set of goods X, a combinatorial valuation function $v : 2^X \to \mathbb{R}^+$, and an exogenous pricing mechanism $p : X \to \mathbb{R}^+$, an optimal solution to the acquisition problem is a subset $A^* \subseteq X$ *s.t.*

$$A^* \in \arg\max_{Y \subseteq X} v(Y) - p(Y) \tag{1}$$

Definition 3.2 Given a set of goods X, the marginal utility of good x is defined as follows: $\mu(x|X) = u(X^*) - u(Y^*)$, where

$$X^* \in \arg\max_{Y \subseteq X \cup \{x\}} v(Y) - p(Y \setminus \{x\}) \tag{2}$$

and

$$Y^* \in \arg\max_{Y \subseteq X \setminus \{x\}} v(Y) - p(Y) \tag{3}$$

In words, the marginal utility of good x, relative to the set X, is simply the difference between the utility of X, assuming x costs 0, $u(X|p(x) = 0)$, and the utility of X, assuming x costs ∞, $u(X|p(x) = \infty)$.

As is evident from the definition of MU, computing marginal utilities requires two calls to a combinatorial optimization solver. If an agent explicitly answers question 1 before question 2, it might determine that it is not necessary to compute marginal utilities for all copies of all goods (although in the worst case it solves one additional optimization problem). ROXYBOT explicitly answers question 1, before question 2, and usually achieves computational savings over direct MU calculation. But the real motivation behind ROXYBOT's approach is that reasoning about individual goods, rather than coherent sets of goods, when valuations are combinatorial, is suboptimal. In the following example, ROXYBOT's approach is optimal, but the direct MU approach is not.

Example 3.3 Given goods x, y, z, with combinatorial valuations as follows: $v(xyz) = v(xy) = v(yz) = 500$ and $v(x) = v(y) = v(z) = v(xz) = 0$. Each good in isolation is of no value, but together with y, either x or

z is worth 500. Assume all goods are priced equivalently at 100. The optimal sets of goods in this example are xy and yz, each of which yields utility of 300.

Now let us compute marginal utilities: $\mu(x|xyz) = u(xyz|p(x) = 0) - u(xyz|p(x) = \infty) = 400 - 300 = 100$; $\mu(y|xyz) = u(xyz|p(y) = 0) - u(xyz|p(y) = \infty) = 400 - 0 = 400$; and $\mu(z|xyz) = u(xyz|p(z) = 0) - u(xyz|p(z) = \infty) = 400 - 300 = 100$. Thus, a MU bidder bids on all three goods, and wins all three goods, obtaining utility $v(xyz) - (p(x) + p(y) + p(z)) = 200$.

A bidder that first determines the set of goods it wants bids on either x and y, or y and z, but not both. Computing marginal utilities for (say) x and y yields: $\mu(x|xy) = u(xy|p(x) = 0) - u(xy|p(x) = \infty) = 400 - 0 = 400$ and $\mu(y|xy) = u(xy|p(y) = 0) - u(xy|p(y) = \infty) = 400 - 0 = 400$. This bidder earns the optimal utility $v(xy) - (p(x) + p(y)) = 300$. □

The two theoretical observations reported in this paper are the following: (i) for all goods x in an optimal acquisition, the price of x does not exceed the marginal utility of x; and (ii) for all goods x *not* in an optimal acquisition, the marginal utility of x does not exceed the price of x. Formal statements and proofs follow.

Observation 3.4 *If $A^* \subseteq X$ is an optimal solution to the acquisition problem, then $\mu(x|A^*) \geq p(x)$, $\forall x \in A^*$.*

Proof 3.5

$$\begin{aligned} & \mu(x|A^*) \\ = \; & [v(X^*) - p(X^* \setminus \{x\})] - [v(Y^*) - p(Y^*)] && (4) \\ \geq \; & [v(X^*) - p(X^*) + p(x)] - [v(Y^*) - p(Y^*)] && (5) \\ = \; & [v(A^*) - p(A^*) + p(x)] - [v(Y^*) - p(Y^*)] && (6) \\ = \; & u(A^*) + p(x) - u(Y^*) && (7) \\ \geq \; & p(x) && (8) \end{aligned}$$

Eq. 4 is the definition of marginal utility (X^* is defined in Eq. 2 and Y^* is defined in Eq. 3). Eq. 5 follows from the assumption that $x \in A^*$: if $v(X^*) - p(X^* \setminus \{x\}) < v(X^*) - p(X^*) + p(x)$, then $x \notin A^*$. Eq. 6 follows from the fact that $X^* \subseteq A^* \cup \{x\} = A^*$. Eq. 7 is the definition of utility. Finally, Eq. 8 follows from the fact that $u(A^*) \geq u(Y^*)$ since $Y^* \subseteq A^* \setminus \{x\}$. □

Observation 3.6 *If $A^* \subseteq X$ is an optimal solution to the acquisition problem, then $\mu(x|A^*) \leq p(x)$, $\forall x \notin A^*$.*

Proof 3.7 The proof of Obs. 3.6 mimics that of Obs. 3.4. □

RoxyBot's strategy answers questions 1 and 2 in turn, and bids marginal utilities on all the goods it wants. This strategy is optimal, under our assumptions, by the previous observation. In particular, RoxyBot wins all the goods it wants by bidding marginal utilities. Of course, bidding $p(x)$, or ∞, $\forall x \in A^*$, are also optimal policies in this setting. But bidding the marginal utility of good x is a reasonable heuristic, since it is in fact an optimal bidding policy if the prices of all goods other than x are exogenous and certain.

4 Bidding Under Uncertainty

In TAC, as in most auctions, clearing prices are *not* known with certainty. On the contrary, agents make predictions about clearing prices, and rather than predict point estimates, it can be advantageous to agents to predict price distributions. RoxyBot-01 and RoxyBot-02 extend RoxyBot by predicting price distributions, rather than point estimates.

There are a number of ways an agent can make use of price distributions. One obvious extension of the MU bidder introduced in Sec. 3 is a bidder that takes many samples from price distributions, computes marginal utilities, and bids *average* marginal utilities. (This strategy more accurately reflects that which was employed by ATTac [Stone *et al.*, 2002].)

But it is well-known that averaging can lead to nonsensical behavior. Recall the example of the robot that comes to a fork in the road. The robot reasons as follows, "50% of the time it is optimal to go 40° to the right, but 50% of the time it is optimal to go 40° to the left. Aha! I shall go straight ahead." Lo and behold, the robot walks directly into a tree.

Verifying this intuition, the following example demonstrates that bidding average marginal utility is not an optimal bidding strategy. In this example, we compare an average MU strategy ($\overline{\text{MU}}$) with RoxyBot and MU, assuming RoxyBot and MU base their decisions on the mean of the price distributions that characterize the example. Prices are exogenous, as above.

Example 4.1 Assume goods and combinatorial valuations are as in Ex. 3.3. But assume that the prices of each good are distributed as follows: with probability 0.5, $p(a) = 0$ and with probability 0.5, $p(a) = 200$, for all $a = x, y, z$. Now let us consider the behavior of the three bidding strategies.

RoxyBot and MU take as input the mean price of each distribution, namely $p(a) = 100$, for all $a = x, y, z$, and proceed as if prices are known with certainty. RoxyBot chooses an optimal set of goods on which to bid, say $\{x, y\}$, and bids MUs (400) on each good in this set. MU bids $\mu(x) = 100$, $\mu(y) = 400$, and $\mu(z) = 100$.

The $\overline{\text{MU}}$ bidder determines its bidding policy as follows: it obtains many samples of the price distributions, and averages the MUs under each sample. The table below depicts this reasoning. The average MU bidder bids 100 on x, 450 on y, and 100 on z.

X	Y	Z	$\mu(X)$	$\mu(Y)$	$\mu(Z)$
0	0	0	0	500	0
200	0	0	0	500	200
0	200	0	0	500	0
0	0	200	200	500	0
200	200	0	0	500	200
200	0	200	200	300	200
0	200	200	200	500	0
200	200	200	200	300	200
Average MU			100	450	100

To evaluate the bidding policies generated by these three strategies, we determine their average scores, given the price distributions. RoxyBot outperforms both MU and $\overline{\text{MU}}$ in this example, although it does not dominate their behavior in all circumstances. □

X	Y	Z	MU	$\overline{\text{MU}}$	RoxyBot	Oracle
0	0	0	500	500	500	500
200	0	0	500	500	300	500
0	200	0	300	300	300	300
0	0	200	500	500	500	500
200	200	0	300	300	100	300
200	0	200	0	0	300	300
0	200	200	300	300	300	300
200	200	200	−200	−200	100	100
Scores			275	275	300	350

Rather than bidding average marginal utilities in answering bidding questions 1 and 2, RoxyBot-01 and RoxyBot-02 take two alternative approaches to bidding in uncertain environments. Both of these approaches insist upon reasoning about coherent sets of goods, rather than independent goods, extending the RoxyBot's philosophy, hereafter RoxyBot-00.

4.1 RoxyBot Under Uncertainty

In preparing for TAC-2000, most development efforts were concentrated on solving the acquisition problem, since an efficient solution to this problem is essential to answering bidding questions 1 and 2. Once this problem was solved [Greenwald and Boyan, 2001; Stone *et al.*, 2001], agent designers went on to solve other problems in the TAC domain. For example, ATTac's developers focused on building a machine learning algorithm that predicts clearing price distributions [Schapire *et al.*, 2002]. Walverine's developers focused on (among other things) computing competitive equilibrium prices, which they used to estimate auction clearing prices [Cheng *et al.*, 2003]. In designing RoxyBot, we chose to focus on the design of algorithms that operate under uncertainty. We now present an example that motivates our approach.

Example 4.2 Consider only one good a of value \$100. Suppose that 90% of the time a costs \$1, but that 10% of the time a costs \$1 million. In this setting, RoxyBot-00 predicts that the price of good a is its expected price, roughly \$100,000. Thus, RoxyBot-00 bids \$0, and scores \$0. But now consider the bidding policy "bid \$100." This policy scores \$99 90% of the time, and \$0 10% of the time. Thus, on average, this policy scores ∼\$90. "Bid 100" dominates RoxyBot-00's bidding policy in this example. We seek an agent strategy that exploits the information inherent in price distributions. □

Example 4.3 Let us revisit the previous example, and compare RoxyBot-00's score with that of an average MU bidder. $\overline{\text{MU}}$ bids according to the policy "bid ∼\$90" in this example, which scores ∼\$90 on average. In other words, average MU outperforms RoxyBot-00 in this example. Thus, neither RoxyBot-00 nor average MU dominates the other, in general (see Ex. 4.1). □

4.2 RoxyBot-01

Like RoxyBot-00, RoxyBot-01 answers questions 1 and 2 in turn. But in answering question 1, RoxyBot-01 takes account of distributional information. Intuitively, if a good is wanted in sufficiently many circumstances, it is added to the agent's want list; on the other hand, if a good is not wanted in sufficiently many circumstances, it is subtracted from the agent's want list.

More specifically, RoxyBot-01's acquisition scheme operates as follows: sample the price distributions many times; solve the acquisition problem for each sample; if a copy of a good is included in sufficiently many solutions, include this good in the final solution; if a copy of a good is included in sufficiently few solutions, eliminate this good from the final solution; conditioned on the new set of goods in and out of the final solution, repeat.

Due to the (logical) complexity of RoxyBot-01's core, we defer analysis to a longer version of this paper.

4.3 RoxyBot-02

RoxyBot-02

SAMPLE price distributions n times to fix test set
WHILE !(TIME_OUT)
1. GENERATE candidate policy
2. score = SIMULATE this policy

RETURN the highest-scoring policy

GENERATE

1. SAMPLE price distributions
2. COMPUTE bidding policy using sample prices *à la* RoxyBot-00, RoxyBot-01, MU, or $\overline{\text{MU}}$
3. RETURN bidding policy

SIMULATE

1. $sum = 0$
2. for $i = 1$ to n samples in test set
   (a) compute winnings and *cost*
   (b) allocate winnings to find *value*
   (c) add $value - cost$ to sum
3. return $(sum\ /\ n)$

Table 1: RoxyBot-02: Monte Carlo Simulation.

RoxyBot-02 directly generalizes RoxyBot-00. The underlying idea of this agent design is to generate and test numerous candidate policies, and then to place bids and asks according to the best policy seen. Candidates are generated as follows: sample from the price distributions; answer questions 1 and 2. Candidates are evaluated over n samples: for each sample, the candidate's score is computed, and scores are averaged over all samples (as in the policy evaluation process in Ex. 4.1). This algorithm is depicted in Table 1.

In theory, RoxyBot-02 approximates an optimal bidding policy. If RoxyBot-02 were to evaluate all possible candidate policies, under an infinitely many samples, it would output an optimal bidding policy, given the (exogenously determined) price distributions. In practice, RoxyBot-02 can generate policies according to any of the aforementioned algorithms: RoxyBot-00, RoxyBot-01, MU, or average MU. By including in the space of candidates bidding policies generated by all of these alternative strategies, we can ensure (probabilistically) that RoxyBot-02 dominates the others.

In TAC-02, RoxyBot-02 generated its candidate policies using RoxyBot-00's internals. In Ex. 4.2, this instantiation of RoxyBot-02 generates two policies: if the sample price is $1, its bidding policy is "bid $100;" if the sample price is $1 million, its bidding policy is "bid $0." "Bid $100" scores $90, on average, whereas "bid $0" scores $0. Thus, RoxyBot-02 employs the policy "bid $100", and scores $90, on average. RoxyBot-01 generates only one policy in this example, either "bid $100" (if it bids MUs) or "bid $90" (if it bids average MUs). Thus, RoxyBot-01 also scores $90, on average.

5 Experiments

Our analyses of agent bidding strategies in the previous two sections was based on the assumption that prices are determined exogenously. In particular, RoxyBot-00 is optimal if prices are certain and determined exogenously; similarly, RoxyBot-02 is approximately optimal if prices are uncertain and determined exogenously. In this section, we discuss experiments designed to ascertain the power of these strategies in TAC, where prices are *not* determined exogenously.

5.1 Setup

In our experiments, we pitted 4 TAC agents bidding according to one strategy against 4 TAC agents bidding according to another strategy (*e.g.*, 4 RoxyBot-02 agents vs. 4 RoxyBot-00 agents). We refer to each set of 4 TAC agents in one game as a team. We played numerous games between pairs of teams—exact numbers depended on which teams were participating—until we achieved statistical significance. In RoxyBot-02, we arbitarily fixed the number of samples $n = 50$. No attempt was made to optimize this parameter. None of the other algorithms used in this study—RoxyBot-00, MU, and $\overline{\text{MU}}$–have any tuneable parameters.

Before running any experiments, we played 500 training games between RoxyBot-00 and MU, initializing price estimates to the competitive equilibrium prices derived in Wellman, *et al.* [2002b]. Using data collected from these training games, we generated distributions over clearing prices for each good. The distributions were represented by a lookup table over five salient features of the domain, the details of which are beyond the scope of this paper. In each cell of the table, 10 numbers were stored, corresponding to the predictions at percentiles $5, 10, 15, \ldots, 95$. This representation was chosen for its simplicity and its weak assumptions about the shape of the underlying price distributions. We sought to capture the highly skewed and multimodal distributions that arise in practice in TAC games.

A sample output after the run of one game instance is shown in the table below.

Agent	Score	Rank
RoxyBot-02	3802	1
RoxyBot-02	3116	6
RoxyBot-02	3166	5
RoxyBot-02	3447	4
RoxyBot-00	2521	8
RoxyBot-00	3696	2
RoxyBot-00	2788	7
RoxyBot-00	3600	3

Scores in our experiments are not as high as scores in actual competitions, because bidding marginal utilities, as do all eight agents in our experiments, tends to lead to high prices.

5.2 Evaluation

To evaluate our results, we use two statistical tests: the z-test, which we use to compare scores, and the Wilcoxon test, which we use to compare rankings. In our context, the inputs to the z-test are two sample datasets of scores, one per team, over many game instances. The z-test outputs the probability that the difference between the means of these datasets is positive: *i.e.*, the probability that the mean of the second is greater than the mean of the first. Our input to the Wilcoxon test[1] is a list of pairs of average rankings, one per game. The test measures the significance of the difference in these rankings; its interpretation suggests that one team is ranked higher than the other team.

5.3 Results

Our experimental results are depicted numerically in Table 2 and graphically in Fig. 1. These results are qualitatively consistent with our prior analyses: MU bidding outperforms average MU bidding; RoxyBot-00 outperforms MU bidding; and, RoxyBot-02 outperforms RoxyBot-00. Moreover, these results are transitive: RoxyBot-00 outperforms average MU bidding; RoxyBot-02 outperforms MU bidding; and RoxyBot-02 outperforms average MU bidding.

The numbers in Table 2 reveal that with high confidence, all strategies are expected to score higher than average MU bidding. Also with high confidence, RoxyBot-02 is expected to score higher than RoxyBot-00 and MU. Finally, although RoxyBot-00 is expected to score higher than MU, the lower confidence level supporting this conclusion makes this results less credible. The outcome of all Wilcoxon tests reinforce the outcome of the z-tests. Similarly, the graphs depicted in Fig. 1 complement our numerical findings.

[1]For a description of the Wilcoxon test, visit `http://fonsg3.let.uva.nl/Service/Statistics/Signed_Rank_Test.html`.

Teams	Means		z-test	Wilcoxon	Games
$\overline{\text{MU}}$ < MU	964	1908	.999	.999	25
MU < RoxyBot-00	1508	1612	.793	.803	75
RoxyBot-00 < RoxyBot-02	1837	2031	.977	.996	50
$\overline{\text{MU}}$ < RoxyBot-00	1334	2034	.999	.999	25
MU < RoxyBot-02	1705	1987	.976	.993	50
$\overline{\text{MU}}$ < RoxyBot-02	915	1920	.999	.999	25

Table 2: Numerical Results: Means, z-test, Wilcoxon test, and Sample Size.

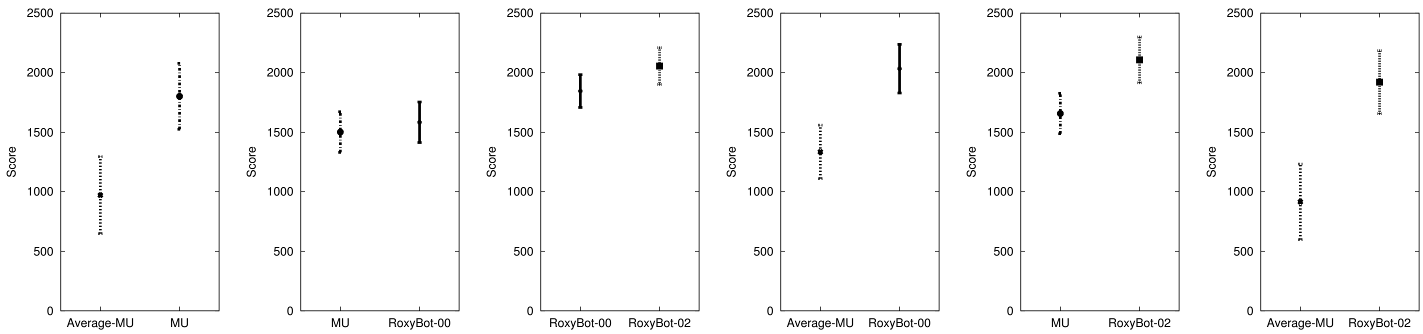

Figure 1: Graphical Results: 95% Confidence Intervals on the Means. All agent strategies clearly outperform average MU bidding, but no other graphs reveal such a clear performance distinction.

5.4 Discussion

In this section, we described experiments in which prices are endogenously determined, but whose results are consistent with our earlier analyses of agent bidding strategies, where we assumed that prices were exogenously determined. Recall that eight agents participate in each TAC game instance. Thus, the setting is not strictly competitive (an agent can sometimes detect the influence of its behavior on market prices); nor is it straightforward to analyze from a game-theoretic perspective. Wellman, *et al.* [2002b] conducted a study of price prediction techniques used in TAC-02, and found that competitive equilibrium analysis was the best proposed method. This finding is consistent with our results, since it suggests that an individual TAC agent's influence on prices is often inconsequential, although TAC agents collectively determine market prices. In future work, we intend to design bidding strategies that explictly model each agent's influence on its environment.

Finally, we note one strong assumption that permeates this work; that is, the assumption that good prices are independent. Specifically, we model one price distribution per good, and we draw independent samples from each price distribution. In future work, we intend to investigate how to compactly represent the joint price probability distributions that naturally arise in combinatorial auctions and SAs where goods have combinatorial valuations. Moreover, we plan to design agent bidding strategies capable of reasoning about such structures.

Acknowledgments

This research was supported by NSF Career Grant #IIS-0133689. The author also acknowledges the contribution of Maureen Hurtgen to this work, and the funding of the CRA's Distributed Mentor Program.

References

[Boyan and Greenwald, 2001] Justin Boyan and Amy Greenwald. Bid determination in simultaneous auctions: An agent architecture. In *Proceedings of Third ACM Conference on Electronic Commerce*, 210–212 2001.

[Cheng *et al.*, 2003] S.F. Cheng, E. Leung, K.M. Lochner, K.O'Malley, D.M. Reeves, L.J. Schvartzman, and M.P. Wellman. Walverine: A walrasian trading agent. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, page To Appear, July 2003.

[Greenwald and Boyan, 2001] Amy Greenwald and Justin Boyan. Bidding algorithms for simultaneous auctions: A case study. In *Proceedings of Third ACM Conference on Electronic Commerce*, 115-124 2001.

[Schapire *et al.*, 2002] Robert E. Schapire, Peter Stone, David McAllester, Michael L. Littman, and János A. Csirik. Modeling auction price uncertainty using boosting-based conditional density estimation. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002.

[Stone *et al.*, 2001] Peter Stone, Michael L. Littman, Satinder Singh, and Michael Kearns. ATTac-2000: An adaptive autonomous bidding agent. In *Fifth International Conference on Autonomous Agents*, 2001.

[Stone *et al.*, 2002] Peter Stone, Robert E. Schapire, János A. Csirik, Michael L. Littman, and David McAllester. ATTac-01: A learning, autonomous bidding agent. In *Workshop on Agent Mediated Electronic CommerceIV: Designing Mechanisms and Systems*, 2002.

[Wellman *et al.*, 2002a] Michael P. Wellman, Amy Greenwald, Peter Stone, and Peter R. Wurman. The 2001 Trading Agent Competition. In *Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference*, pages 935–941, July 2002.

[Wellman *et al.*, 2002b] M.P. Wellman, D.M. Reeves, K.M. Lochner, and Y. Vorobeychik. Price prediction in a Trading Agent Competition. Working Paper, 2002.