Self-Confirming Price Prediction for Bidding in Simultaneous Second-Price Sealed-Bid Auctions

Dong Young Yoon and Michael P. Wellman
University of Michigan
Ann Arbor, MI 48109-2121, USA
{dyoon,wellman}@umich.edu

Abstract

Bidding strategies for simultaneous interdependent markets typically rely on price predictions to manage uncertainty across markets. Previous work on simultaneous ascending auctions has found self-confirming price predictions particularly effective in dealing with the exposure problem. We adapt this approach to a setting with simultaneous one-shot (SPSB) auctions, producing self-confirming prediction versions of a range of heuristic strategies previously proposed in the literature. An in-depth simulation study in service of empirical game-theoretic analysis finds strong support for the BidEvaluator family of strategies in this setting.

1 Introduction

One of the most attractive features of automated trading is the ability to monitor and participate in many markets simultaneously. Compared to human traders, a software agent can take in data from multiple sources at high throughput, and in principle process a massive quantity of information relevant to trading decisions in short time spans. Dealing with multiple markets also poses one of the greatest strategic challenges for automated trading. When markets interact, a strategy for bidding in one market must consider the implications of and ramifications for outcomes in others.

Markets are interdependent when an agent’s preference for the outcome of one market depends on the outcomes in other markets. For instance, when an agent’s value for one good is increased by obtaining another, the goods are complementary. Dealing with multiple markets under complementary preferences presents an agent with the classic exposure problem: to obtain a valuable bundle, the agent must risk getting stuck with a strict subset of the goods, which it may not have wanted at the prevailing prices. For example, in the Trading Agent Competition (TAC) Travel game [Wellman et al., 2007], agents face an exposure problem for hotels—they must obtain a room each night for the client, otherwise the whole trip is infeasible.

Experience from TAC and many other domains has demonstrated the importance of price prediction for bidding in interdependent markets [Wellman et al., 2004]. Given probabilistic projections of prices across markets, the agents can manage exposure risk, choosing bids that optimally trade off the profits and losses of the possible bundles of goods obtained.

Greenwald and Boyan [2004] framed the problem of bidding across interdependent markets given probabilistic price predictions. Follow-on work [Greenwald et al., 2009, Wellman et al., 2007] formalized this bidding problem in decision-theoretic terms, and established properties of optimal bidding strategies given the assumption that bids do not affect other-agent behaviors. Further experimental comparison was performed by Greenwald et al. [2010]. These works introduced a taxonomy of heuristic bidding strategies, which we employ in the current study as described in §3.

The current study extends or departs from this previous line of investigation in three ways:

1. The market mechanism is simultaneous one-shot auctions, specifically second-price sealed bid auctions. This contrasts with the abstract “pseudo-auction” model employed in theoretical analysis, and forms of ascending auction employed in experimental studies.

2. Our analysis is based on an empirical game model derived through simulation of strategy profiles.

3. The bidding strategies employ market predictions based on derivation of self-confirming prices (defined in §4).

The idea of bidding based on self-confirming prices was first explored in the context of simultaneous ascending auctions (SAAs) [Cramton, 2005]. For the SAA environment, we found bidding using such predictions to be highly effective for dealing with the exposure problem [Wellman et al., 2008]. Given this success, the approach is a natural candidate for that role in simultaneous one-shot auctions. To our knowledge, no other general price-prediction methods have been proposed for simultaneous one-shot auctions, other than learning from historical observations.

2 Environment

Our analysis environment is defined by a market mechanism, and a distribution of valuations over goods ascribed to trading agents. The strategies defined here are applicable for any valuation distribution, but our experimental results are based on the particular distribution specified here.
2.1 Market Mechanism: Simultaneous SPSB

The second-price sealed-bid (SPSB) auction is one of the classic basic auction mechanisms analyzed in the seminal work of Vickrey [1961]. That the bids are sealed means that bidders have no information about the bids of other participants until the outcome is revealed. That it is second-price means that the winning bidder pays the second-highest bid rather than its own (highest) bid.

We consider a situation where multiple goods are auctioned through simultaneous SPSB auctions. The mechanism is one-shot, that is, the agents submit bids to each auction in parallel, upon which the auctions compute and report their results.

Our study employs the familiar independent private values model [Krishna, 2010], where each agent’s valuation for outcomes is drawn independently from a common-knowledge probability distribution. For a single SPSB auction with independent private values, it is a dominant strategy for an agent to bid its true valuation. This fails to hold for multiple simultaneous SPSB auctions, unless the agents’ valuations across goods are also independent. When agents’ valuations for different goods interdepend (e.g., through complementarity), bidding truthfully in simultaneous individual-good auctions is not even an option, as the value for an individual good is not well-defined.

2.2 Valuations: Market-based Scheduling

Let \( \mathcal{X} \) be the set of goods available in all markets. The valuation function \( v_i: 2^\mathcal{X} \rightarrow \mathbb{R} \) specifies agent \( i \)'s value for the subsets of goods it may obtain in the markets.

The valuation distribution we employ in this study is based on a model of market-based scheduling [Reeves et al., 2005]. In this model, goods represent time slots of availability for some resource: for example, a machine, a meeting room, a vehicle, or a skilled laborer. Agents have tasks, which require this resource for some duration of time to complete.

Specifically, let there be \( n \) agents, and \( \mathcal{X} = \{ x_1, \ldots, x_m \} \) the set of \( m \) time slots available to be scheduled. Agent \( i \)'s task requires \( \lambda_i \) time slots to accomplish, and the agent values a set of time slots according to when they enable completion of the task. If an agent \( i \) acquires \( \lambda_i \) time slots by the deadline \( t \), it obtains value \( v_i^T \). The value with respect to the deadline is a nonincreasing function: for all \( i \), if \( t < t' \) then \( v_i^T \geq v_i^{t'} \). If it fails to obtain a sufficient set of goods, the agent accrues value \( v_i^\infty = 0 \). Let \( X \subseteq \mathcal{X} \) denote a set of slots. The expression \( \{ \{ x_j \in X \mid j \leq t \} \} \) represents the number of these that are for time \( t \) or earlier. We can thus write

\[
T(X, \lambda) = \min \{ t \text{ s.t. } \{ \{ x_j \in X \mid j \leq t \} \geq \lambda \} \cup \{ \infty \} \}
\]

to denote the earliest time by which \( X \) contains at least \( \lambda \) slots. The overall valuation function for agent \( i \) is then

\[
v_i(X) = v_i^{T(X, \lambda_i)}.
\]

In our experiments, we take \( n = 8 \) and \( m = 5 \). For each agent, a task length \( \lambda_i \) is drawn uniformly over the integers \( \{ 1, \ldots, 5 \} \). The deadline values are drawn uniformly over \( \{ 1, \ldots, 50 \} \), then pruned to impose monotonicity [Reeves et al., 2005]. The valuations induced by this scheduling scenario exhibit strong complementarity among goods. When \( \lambda > 1 \), the agent gets no value at all for goods in a bundle of fewer than \( \lambda \). On the other hand, there is some degree of substitutability across goods when there may be multiple ways of acquiring a bundle of the required size.

3 Bidding with Price Predictions

To deal effectively with interdependent markets, an agent’s bid in each auction must reflect its beliefs about the outcomes of others. Following prior work, we assume agents have some source of predictions about the prices at which it might obtain goods in the respective auctions. The bidding strategy is then a function of the available price prediction. In this section, we present the bidding strategies considered in this study, along with relevant concepts necessary to understand how the strategies operate. Our methods for deriving price predictions are presented in the following section.

3.1 Marginal Values and Optimal Bundles

Interdependence dictates that the value of individual goods be assessed relative to the bundle of goods that an agent obtains. This can be expressed by the notion of marginal value.

**Definition 1 (Marginal Value)** Agent \( i \)'s marginal value, \( \mu_i(x, X) \), for good \( x \) with respect to a fixed bundle of other goods \( X \) is given by:

\[
\mu_i(x, X) = v_i(X \cup \{ x \}) - v_i(X).
\]

Given a fixed vector of prices, \( p = \langle p_1, \ldots, p_m \rangle \), let \( \sigma_i(X, p) \) denote agent \( i \)'s surplus from obtaining the set of goods \( X \) at those prices:

\[
\sigma_i(X, p) = v_i(X) - \sum_{j \mid x_j \in X} p_j.
\]

**Definition 2 (Acquisition [Boyan and Greenwald, 2001])**

Given prices \( p \), the acquisition problem selects optimal goods to purchase:

\[
X^* = \text{ACQ}_i(p) \equiv \arg \max_{X \subseteq \mathcal{X}} \sigma_i(X, p),
\]

where \( \sigma \) is defined by (1).

Faced with known prices, an optimal strategy would be to compute a solution \( X^* = \text{ACQ}_i(p) \) and buy the goods \( X^* \). Its result is the optimal surplus at these prices:

\[
\sigma^*_i(p) \equiv \sigma_i(\text{ACQ}_i(p), p).
\]

To assess goods with respect to given prices, we extend the concept of marginal value. Let \( \hat{p}[p_j \leftarrow q] \) be a version of the price vector with the \( j \)th element revised as indicated:

\[
\hat{p}[p_j \leftarrow q] = \langle p_1, \ldots, p_{j-1}, q, p_{j+1}, \ldots, p_m \rangle.
\]

**Definition 3 (Marginal Value with Buying Opportunities)** Agent \( i \)'s marginal value \( \mu_i(x_j, p) \) for good \( x_j \) with respect to prices \( p \) is given by:

\[
\mu_i(x_j, p) = \sigma^*_i(\hat{p}[p_j \leftarrow 0]) - \sigma^*_i(\hat{p}[p_j \leftarrow \infty]).
\]

Here, \( \sigma^*_i(\hat{p}[p_j \leftarrow 0]) \) represents agent \( i \)'s optimal surplus (solving its acquisition problem) at the given prices, assuming it receives good \( x_j \) for free. Similarly, \( \sigma^*_i(\hat{p}[p_j \leftarrow \infty]) \) represents the optimal surplus if \( x_j \) were unavailable. The
difference is precisely the marginal value of good $x_i$ with respect to its buying opportunities for other goods. Note that Definition 3 can be viewed as a generalization of Definition 1, under the interpretation that goods in $X$ have zero price, and all other goods have infinite price.

3.2 A Non-Predictive Baseline Strategy

To calibrate the performance of bidding strategies based on price prediction, we introduce a baseline strategy that employs no price information. Without price guidance, there is little basis to choose among alternative bundles in a one-shot auction. We therefore simply assume the agent selects its most valued set of goods, which is the solution to its acquisition problem at zero prices. Let $X^* = \text{ACQ}_0(0)$.

The strategy $\text{BaselineBidding}$ bids for goods in $X^*$, with a total bid budget of $v(X^*)$. To determine the bid price for each individual good, it starts with the value for each good taken individually, and evenly divides the remaining budget among the goods. More precisely, let the excess value be given by

$$e = v(X^*) - \sum_{x \in X^*} v\{x\}.$$ 

For complementary preferences, $e$ is guaranteed to be non-negative. $\text{BaselineBidding}$ then bids for good $j$ as follows:

$$\text{BaselineBidding}_j = \begin{cases} v\{x_j\} + \frac{e}{|X^*|} & \text{if } x_j \in X^*, \\ 0 & \text{otherwise.} \end{cases}$$

For the market-based scheduling valuation, $X^*$ comprises the first $\lambda_i$ goods. Bundles with additional goods also maximize value, but we take the minimal maximizing set. Individual goods are valued at zero, unless $\lambda_i = 1$, in which case $v_i(x_j) = v_i^j$. $\text{BaselineBidding}$ agents with a single-unit task duration thus bid true value for the first slot, and otherwise they bid for the first $\lambda_i$ goods at the price $v_i^\lambda_i/\lambda_i$.

3.3 Bidding with Point Price Predictions

We consider first a set of strategies that employ predictions in the form of a vector of point prices, $\mathbf{p} = (p_1, \ldots, p_m)$. All strategies considered in this section and the next are described with further motivation and detail by Wellman et al. [2007, Chapter 5].

Like $\text{BaselineBidding}$, strategies in the $\text{TargetBidder}$ family restrict their bidding to goods in $X^* = \text{ACQ}_0(\mathbf{p})$. They differ in the amounts they bid. The $\text{TargetPrice}$ strategy bids the predicted price for each good in its optimal bundle.

$$\text{TargetPrice}_j = \begin{cases} p_j & \text{if } x_j \in X^*, \\ 0 & \text{otherwise.} \end{cases}$$

$\text{TargetMV}$ bids at marginal value rather than predicted price.

$$\text{TargetMV}_j = \begin{cases} \mu(x_j, \mathbf{p}) & \text{if } x_j \in X^*, \\ 0 & \text{otherwise.} \end{cases}$$

$\text{TargetMV}^*$ also bids at marginal value, except that it calculates $\mu$ under the assumption that goods outside the target bundle are unavailable.

$$\text{TargetMV}^*_j = \begin{cases} \mu(x_j, \mathbf{p}, p_{\ell} = \infty | x_{\ell} \not\in X^*) & \text{if } x_j \in X^*, \\ 0 & \text{otherwise.} \end{cases}$$

The virtue of $\text{TargetBidder}$ strategies is that they will not obtain goods outside the desired bundle. A weakness is their fragility—if the agent fails to get the full bundle, there is no recourse to others outside the set. The $\text{StraightMV}$ strategy hedges against incorrect predictions by bidding marginal value for all goods.

$$\text{StraightMV}_j = \mu(x_j, \mathbf{p}).$$

3.4 Bidding with Price Distributions

A point price estimate fails to convey the uncertainty inherent in predicting the prices at which goods will be available. Probability distributions over prices provide a more general representation, expressing degrees of belief over the possible prices that might obtain. We therefore expand our strategy set to incorporate bidding methods that take as input $F$, a probability distribution over prices.

Expected Value Methods

The expected value method [Birge and Louveaux, 1997] approximates a stochastic optimization method by collapsing probability distributions into point estimates through expectation. Let $\mathbf{p}_F = (p_1, \ldots, p_m)$, where $p_j = \mathbb{E}_F[p_j]$ is the expectation of $p_j$ under distribution prediction $F$.

Any bidding strategy defined for point price predictions can be adapted to take price distributions through the expected value method, simply by using $\mathbf{p}_F$ for the point price prediction. We thus define $\text{StraightMU}$, $\text{TargetMU}$, and $\text{TargetMU}^*$ as distribution-based methods that employ corresponding point prediction strategies in this way. “MU” stands for “marginal utility”, but its usage here is simply to distinguish the name from its corresponding point strategy ending with “MV”. Our implementation calculates $\mathbf{p}_F$ by sampling from $F$. Since accuracy depends on the number $k$ of samples taken, we indicate a version of the strategy by appending “k” to the name. For example, $\text{StraightMU8}$ calculates $\mathbf{p}_F$ as the mean of eight samples from $F$, and employs $\text{StraightMV}$ with that average price vector as input.

AverageMU

Whereas $\text{StraightMU}$ bids the marginal value of the expected price, the strategy $\text{AverageMU}$ bids the expected marginal value:

$$\text{AverageMU}_j = \mathbb{E}_F[\mu(x_j, \mathbf{p})].$$

Our implementation samples from the price distribution, calculating marginal values from each sample, and averaging the results.

BidEvaluator

The final approach we consider is to generate candidate bid vectors, and evaluate them according to the given price distribution. The $\text{BidEvaluator}$ strategy uses other bidding strategies to propose candidates, and estimates their performance by sampling from $F$ and averaging over the resulting surpluses. For instance, $\text{BidEvaluator}(\text{SMU8})$ generates candidates using $\text{StraightMU8}$. Since each invocation of $\text{StraightMU8}$ employs a new draw of eight samples from $F$ to estimate $\mathbf{p}_F$, we generally obtain different bids. In our study, we define $\text{BidEvaluator}(\text{SMU8})$ to generate four candidate bids using $\text{StraightMU8}$, which takes 32 samples from
We employ another 32 samples to evaluate these candidates, for a total of 64 required by this strategy. The result of BidEvaluator(SMU8) is the candidate bid vector which performed best on average according to the 32 test samples.

4 Self-Confirming Price Predictions

Now that we have a suite of strategies that employ price predictions, we turn next to the question of how to generate such predictions. We propose here to employ self-confirming price predictions, originally introduced and evaluated in the context of simultaneous ascending auctions [Wellman et al., 2008]. Our definitions and derivation methods follow this prior work.

In the following definitions, let \( \Gamma \) be an instance of a simultaneous auction game, and \( PP \) denote a bidding strategy that employs price predictions: either points or distributions.

**Definition 4 (Self-Confirming Price Prediction)** The prediction \( \pi \) is a self-confirming prediction for strategy \( PP \) in \( \Gamma \) iff \( \pi \) is equal to the expectation (over the type distribution) of the final prices when all agents play \( PP(\pi) \). Similarly, \( F \) is a self-confirming price distribution for \( PP \) in \( \Gamma \) iff \( F \) is equal to the expectation (over the type distribution) of the final prices when all agents play \( PP(F) \).

We also consider a relaxed version of self-confirming distributions, where the final prices and predictions need coincide only on marginal—rather than joint—probabilities.

4.1 Procedure for Self-Confirming Point Price Prediction Approximation

We employ a simple iterative procedure to derive an approximate self-confirming point prediction for strategy \( PP \) in \( \Gamma \).

Our procedure is defined by the following set of parameters:

1. An initial price prediction \( \pi^0 \). We take \( \pi^0 = (1, \ldots, 1) \).
2. A number \( G \) of game instances per iteration. We set \( G \) to one million in our study.
3. A threshold \( \delta \). When the distance between results of successive iterations is less than \( \delta \), we halt the procedure and return the latest result. We set this at 0.001 times the maximum value, hence \( \delta = 0.05 \).
4. A maximum number \( L \) of iterations. We take \( L = 100 \) in our experiments.
5. A decaying sequence, \( \kappa_t \), which controls the updating of predictions in each iteration. We employ \( \kappa_t = \frac{L-t+1}{L} \).

At iteration \( t \), we run \( G \) instances of game \( \Gamma \), with all agents playing strategy \( PP(\pi^{t-1}) \). We tally the resulting prices resulting from each instance, and denote the average price vector by \( \bar{p}^t \). We then update the price prediction by

\[
\pi^t \leftarrow \pi^{t-1} + \kappa_t (\bar{p}^t - \pi^{t-1}).
\]

When the maximum price difference between two successive iterations is below threshold,

\[
\max_j |\pi^t_j - \pi^{t-1}_j| \leq \delta,
\]

the procedure halts and returns \( \pi^t \).

**Figure 1:** Effect of the decay parameter for iterative update of TargetMV* predictions.

We evaluated the convergence of this procedure with respect to three different \( PP \) strategies: StraightMV, TargetMV, and TargetMV*. For StraightMV, we reach a fixed point within 20 iterations, even with the setting \( \kappa_t = 1 \). For TargetMV and TargetMV*, the price predictions tend to oscillate, hence we need a decaying sequence to settle the process. For these cases, we do not expect the results to approximate self-confirmation as well. Nevertheless, the iterative procedure produces a well-defined prediction, which in many cases is a reasonable basis for bidding.

Figure 1 shows the series of pointwise price distances between consecutive iterations in deriving point price predictions for TargetMV* with and without a decay parameter. The absence of a decay parameter is expressed simply by setting \( \kappa_t = 1 \). At this setting, the iterative update of price predictions for TargetMV* fails to converge. Since TargetMV* bids for goods in its target bundle under the assumption that no others are available, these bids tend to be at the extremes of its value range. With all agents playing this strategy, this leads to high prices, which causes changes in targets in the next iteration (by all agents), causing these same goods to have very low prices then. TargetMV produces a similar pattern, though attenuated by its consideration of alternative buying opportunities. With regular oscillation, the effect of the decay sequence is to return the average of the swings that would otherwise be observed.

4.2 Procedure for Self-Confirming Price Distribution Approximation

A similar iterative procedure is used to derive self-confirming price distributions. The cumulative price distribution at iteration \( t \) is denoted \( F^t \). We employ a discrete approximation, rounding all observed prices to the closest integer. We can thus represent the distribution using a probability mass function, \( f(q) \), corresponding to the probability \( \Pr(p = q) \). We adopt the simplifying (and incorrect) assumption that prices are probabilistically independent across goods, which allows us to maintain this as an array of marginal distributions, \( (f_1(q_1), \ldots, f_m(q_m)) \).

To measure the difference between distributions at successive iterations, we adopt the Kolmogorov-Smirnov (KS) statistic,

\[
KS(F, F') = \max_x |F(x) - F'(x)|.
\]

Since we maintain our prediction in terms of marginal distributions, our comparison takes the maximum of the KS statis-
tic separately for each good: $KS_{\text{marg}} = \max_j KS(F_j, F'_j)$.

Our initial prediction $F^0$ considers all integer prices in the feasible range equally likely. As for the point-prediction case, at iteration $t$, we run $G$ instances of game $\Gamma$. Here all agents play the distribution-prediction strategy $PP(F^t)$. We tally the resulting prices resulting from each instance, and denote the sample distribution for good $j$ by $f^t_j$. We then update the distribution prediction, for all $j$ and price possibilities $q$, by

$$f^t_j(q) \leftarrow f^{t-1}_j(q) + \kappa_t(f^t_j(q) - f^{t-1}_j(q)).$$

When $KS_{\text{marg}}(F^t, F')$ is below threshold (here $\delta = 0.01$), we halt and return $F^t$.

A comparison of the self-confirming price predictions derived for StraightMU (distributions) and StraightMV (points) is depicted in Figure 2.

![Price distribution from StraightMU](image)

![Point price prediction from StraightMV](image)

Figure 2: Self-confirming price predictions derived for StraightMU and StraightMV for a particular game.

### 4.3 Accuracy of Self-Confirming Price Predictions

The decay parameter ensures the iterative procedure for deriving price predictions will terminate, but potentially masks violations of self-confirmation in the result. To test the accuracy of self-confirming price predictions for this study, we ran an extra iteration of one million game instances after a supposed self-confirming price prediction is derived by the procedure. The distance of derived price predictions from the resulting prices (without any attenuation) are reported in Table 1. As shown in the table, all are self-confirming to a fine level of approximation.

<table>
<thead>
<tr>
<th>Distribution Strategy</th>
<th>KS Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>StraightMV64</td>
<td>0.00450</td>
</tr>
<tr>
<td>TargetMV64</td>
<td>0.00879</td>
</tr>
<tr>
<td>TargetMV*64</td>
<td>0.00788</td>
</tr>
<tr>
<td>AverageMU64</td>
<td>0.00516</td>
</tr>
</tbody>
</table>

Table 1: Accuracy of self-confirming point and distribution price predictions.

## 5 Empirical Game-Theoretic Analysis of Bidding Strategies

We evaluate the performance of the bidding strategies with self-confirming price predictions through a process of empirical game-theoretic analysis (EGTA). In this EGTA approach, we simulate profiles of an enumerated strategy set, and use the observed payoffs to estimate a normal-form game. The induced game model supports conclusions about the strategic properties of subject strategies and profiles, through application of standard solution concepts.

### 5.1 Hierarchical Game Reduction

The space of strategy profiles grows exponentially in the number of players and strategies considered. Specifically, for a symmetric normal-form game with $n$ players and $S$ strategies, there are $(n+S-1)$ distinct strategy profiles. The primary scenario we analyze includes eight bidding agents. To enable us to scale up the strategy space, we employ hierarchical game reduction [Wellman et al., 2005] to recast the scenario as a four-player game. In this framing, each player controls two bidding agents, imposing its selected strategy on each. The reduction entails some loss in fidelity, but we have found in many EGTA studies that it provides a good approximation at exponentially reduced computational expense.

### 5.2 Regret

We evaluate the stability of strategy profiles by measuring regret, the maximal gain a player could achieve by deviating from the profile. Formally, let $\Gamma = \{n,S,u(\cdot)\}$ be a symmetric normal-form game with $n$ players, strategy space $S$ for each player (since the game is symmetric), and payoff function $u : S \times S^{n-1} \rightarrow \mathbb{R}$. The expression $u(s_i, s_{-i})$ represents the payoff to playing strategy $s_i$ in a profile where the other players play strategies $s_{-i} \in S^{n-1}$.

**Definition 5 (regret)** The regret $\epsilon(s)$ of a strategy profile $s = (s_1, \ldots, s_n)$ is given by

$$\epsilon(s) = \max_i \max_{s'_i \in S} (u(s'_i, s_{-i}) - u(s_i, s_{-i})).$$
A Nash equilibrium profile has zero regret, and more generally regret provides a measure of approximation to equilibrium. Using this regret definition, profile $s$ is an $\epsilon$-Nash equilibrium.

Regret is a property of profiles. Evaluation of a particular strategy is inherently relative to a context of strategies played by other agents. Jordan et al. [2007] proposed ranking strategies according to their performance when other agents are playing an equilibrium.

**Definition 6 (NE regret [Jordan, 2010])** Let $s^{NE}$ be a symmetric (possibly mixed) Nash equilibrium of game $\Gamma$. The NE regret of a strategy $s_i \in S$ is $u(s_i^{NE}, s^{NE}_{-i}) - u(s_i, s^{NE}_{-i})$.

NE regret represents the loss of an agent deviating to a specified strategy from a Nash equilibrium of a game. The rationale for this measure comes from the judgment that all else equal, Nash equilibria provide a most compelling strategic context for evaluating a given strategy.

### 5.3 Simulations

Our game is defined by the scheduling valuation distribution described in §2.2, with $n = 8$ agents and $m = 5$ time slots. We simulate a game instance for a specified strategy profile by drawing from the valuation distribution, applying the strategies to generate bids, and calculating the outcome with simultaneous one-shot SPSE auctions. Since each instance is so computationally simple, we sample one million instances per profile for estimating expected payoffs in all experiments reported here.

### 5.4 Effect of Price Samples

As a preliminary study, we investigated the effect of the number of price samples on the performance of strategies based on distribution predictions. Payoffs for an 8-player empirical game over two strategies, StraightMU8 or StraightMU64, are displayed in Figure 3. Recall that these strategies differ only in the number of price samples that are averaged to produce the price vector employed in marginal value calculations. It is apparent from the graph that StraightMU64 dominates StraightMU8, as we might expect.

Additional price sampling, however, produces diminishing returns. We estimated the 4-player empirical reduced game for five versions of StraightMU, differing on how much they sample prices. Table 2 presents some relevant statistics. Although StraightMU64 and StraightMU256 are dominated, the other strategies are all present in some equilibrium. The performance gain after around 1000 samples becomes difficult to measure.

### 5.5 Strategy Space

To balance performance and computational cost, we choose a sample size of 64 for the price distribution strategies in our primary empirical game analysis. To ensure fairness, our BidEvaluator strategy also uses 64 samples in total, as explained in §3.4.

We analyze an empirical game that consists of eleven strategies: one BaselineBidding, three point price predictors—StraightMV, TargetMV, and TargetMV*; and seven distribution predictors—StraightMU64, TargetMU*64, AverageMU64, BidEvaluator(TMU8), BidEvaluator(TMU*8), and BidEvaluator(SMU8). Seven of these employ self-confirming price predictions derived by the iterative process of §4 with the strategy itself playing the role of PP. The exceptions are BaselineBidding, which employs no price prediction, and the three BidEvaluator strategies, which use price distributions that are self-confirming with respect to other specified bidding strategies. We follow the main analysis by investigating a few smaller empirical games with additional strategies suggested by our initial findings.

### 5.6 Primary Game Results

We evaluated the payoffs for each of the 1001 distinct strategy profiles of our symmetric 4-player game with eleven strategy choices. The strategy BidEvaluator(SMU8) emerged as the clear choice—in fact it is a dominant strategy in this game. Table 3 lists all of the strategies, and for each presents the regret of the profile where all agents follow that strategy. Since BidEvaluator(SMU8) is dominant, the regret in all cases represents the benefit of deviating to that strategy. The strategies are ranked by NE regret (Definition 6), which in this case is the regret of the profile where one player employs the corresponding strategy and the rest play BidEvaluator(SMU8).

One perhaps surprising observation is that a point prediction strategy, TargetMV*, is runner-up with respect to the NE-regret measure. Prior literature had found it effective
Table 3: Regret and efficiency of bidding strategies in the original game, and the follow-up game with 256-sample strategies.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Regret of symmetric profile</th>
<th>NE Regret</th>
<th>Eff. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BidEvaluator(SMU8)</td>
<td>0</td>
<td>0</td>
<td>77.5</td>
</tr>
<tr>
<td>TargetMV*</td>
<td>0.393</td>
<td>0.351</td>
<td>78.2</td>
</tr>
<tr>
<td>TargetMU*64</td>
<td>0.590</td>
<td>0.438</td>
<td>76.0</td>
</tr>
<tr>
<td>AverageMU64</td>
<td>0.783</td>
<td>0.450</td>
<td>72.2</td>
</tr>
<tr>
<td>BidEvaluator(TMU*8)</td>
<td>1.099</td>
<td>0.665</td>
<td>74.0</td>
</tr>
<tr>
<td>StraightMV</td>
<td>0.648</td>
<td>0.684</td>
<td>73.8</td>
</tr>
<tr>
<td>StraightMU64</td>
<td>0.720</td>
<td>0.726</td>
<td>72.9</td>
</tr>
<tr>
<td>TargetMV</td>
<td>1.023</td>
<td>0.990</td>
<td>67.1</td>
</tr>
<tr>
<td>BidEvaluator(TMU8)</td>
<td>1.179</td>
<td>1.055</td>
<td>77.5</td>
</tr>
<tr>
<td>TargetMU64</td>
<td>1.161</td>
<td>1.059</td>
<td>65.5</td>
</tr>
<tr>
<td>BaselineBidding</td>
<td>2.164</td>
<td>2.890</td>
<td>60.5</td>
</tr>
<tr>
<td>TargetMU*256</td>
<td>0.452</td>
<td>0.350</td>
<td>77.2</td>
</tr>
<tr>
<td>AverageMU256</td>
<td>0.751</td>
<td>0.431</td>
<td>72.9</td>
</tr>
<tr>
<td>StraightMU256</td>
<td>0.653</td>
<td>0.677</td>
<td>73.6</td>
</tr>
</tbody>
</table>

Table 4: Regret of hybrid bidding strategies.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Regret of symmetric profile</th>
<th>NE Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>BidEvaluator(SMU8)</td>
<td>0.143</td>
<td>0</td>
</tr>
<tr>
<td>BidEvaluatorMix8</td>
<td>0.167</td>
<td>0</td>
</tr>
<tr>
<td>BidEvaluator(TMU*8)_SMU8</td>
<td>0.757</td>
<td>0.153</td>
</tr>
<tr>
<td>TargetMV* _SMV</td>
<td>0.321</td>
<td>0.219</td>
</tr>
<tr>
<td>TargetMU*64 _SMU64</td>
<td>0.425</td>
<td>0.267</td>
</tr>
<tr>
<td>BidEvaluator(SMU8)_TMU*8</td>
<td>0.998</td>
<td>0.669</td>
</tr>
<tr>
<td>StraightMV _TMV*</td>
<td>1.214</td>
<td>0.825</td>
</tr>
<tr>
<td>StraightMU64 _TMU*64</td>
<td>1.342</td>
<td>0.891</td>
</tr>
</tbody>
</table>

5.7 Hybrid Strategies

From the analysis thus far we cannot separate the effects on performance due to accurate price prediction from that of exploiting a given prediction well in bidding. With the exception of BidEvaluator, all strategies employ predictions that are self-confirming with respect to themselves. It is also possible to construct hybrid bidders, which employ different strategies for purposes of deriving predictions and for purposes of bidding given predictions.

To explore this idea, we considered a representative selection of strategies that mix-and-match price prediction and bidding strategy flavors. Our next empirical game starts with BidEvaluator(SMU8), and adds four strategies that mix across StraightMV/StraightMU and TargetMV*/TargetMU*, and two BidEvaluator that employ different methods for generating predictions and candidate bids. To identify a hybrid strategy we adopt the naming convention BiddingStrategy_PredictionStrategy. For instance, StraightMU64_TMU*64 bids with StraightMU using the price prediction derived from TargetMU*64. The strategy BidEvaluator(TMU*8)_SMU8 generates bid candidates using TargetMU*8, but evaluates them using the self-confirming prediction of StraightMU8. We also introduced BidEvaluatorMix, which is identical to BidEvaluator(SMU8) except that two of its four candidate bids are generated using TargetMU*8.

Table 4 ranks the eight strategies of this empirical game by NE regret. The NE employed for this ranking is a symmetric mixture of BidEvaluator(SMU8) at probability 0.507, and BidEvaluatorMix8 at 0.493. There is also a pure strategy Nash equilibrium (PSNE) with two players choosing each of these strategies (efficiency: 78.4%). This suggests some benefit to diversifying the source of bid candidates, particularly when other agents use a single source.

5.8 Effect of Extending Available Samples

We investigated the potential advantage of allowing more computation (sampling) in the BidEvaluator(SMU) and BidEvaluatorMix strategies. There are three ways to use additional samples: marginal utility calculation, generating
additional bid candidates, or testing the candidates. Table 5 ranks by NE regret the combinations we evaluated in an empirical game. BidEvaluator(SMU64) with four candidate bids and 256 samples for testing and BidEvaluatorMix64 with the equivalent number of samples, where each strategy is chosen by two players, turn out to be the unique PSNE in this game. We observed a positive effect on performance with increased samples for marginal utility and testing, though the actual payoff benefit is relatively small. In fact, the performance of a strategy degrades with 1024 samples, demonstrating the limitation of extending available samples.

<table>
<thead>
<tr>
<th>Strategy Base</th>
<th>C</th>
<th>T</th>
<th>Total samples</th>
<th>Regret of symm. profile</th>
<th>NE Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEMix64</td>
<td>4</td>
<td>256</td>
<td>512</td>
<td>0.126</td>
<td>0</td>
</tr>
<tr>
<td>BE(SMU64)</td>
<td>4</td>
<td>256</td>
<td>512</td>
<td>0.246</td>
<td>0</td>
</tr>
<tr>
<td>BE(SMU64)</td>
<td>8</td>
<td>512</td>
<td>1024</td>
<td>0.254</td>
<td>0.017</td>
</tr>
<tr>
<td>BEMix64</td>
<td>8</td>
<td>512</td>
<td>1024</td>
<td>0.146</td>
<td>0.025</td>
</tr>
<tr>
<td>BE(SMU8)</td>
<td>4</td>
<td>32</td>
<td>64</td>
<td>0.323</td>
<td>0.112</td>
</tr>
<tr>
<td>BEMix8</td>
<td>4</td>
<td>32</td>
<td>64</td>
<td>0.291</td>
<td>0.157</td>
</tr>
</tbody>
</table>

Table 5: BidEvaluator(SMU) and BidEvaluatorMix variants with differing allotments of price samples (C = Candidate bids, T = Test samples).

6 Conclusion

We proposed the use of self-confirming price predictions for bidding in simultaneous SPSB auctions with complementary preferences. Simulations conducted for an 8-agent environment with scheduling valuations demonstrated the viability of this approach, and its applicability across a range of heuristic bidding strategies. An empirical game-theoretic analysis finds strong support for the BidEvaluator family of strategies, particularly in conjunction with price predictions that are self-confirming with respect to StraightMU bidders.

Further work will attempt to establish the generality of these conclusions, by exploring a wider range of market configurations: numbers of agents and goods, and valuation distributions. We also intend to explore the effects on BidEvaluator performance of extending computation to allow more price samples or bid candidates, and the possibility of employing price predictions that are self-confirming for BidEvaluator itself. Finally, we are quite interested in exploiting our ability to derive equilibria for these markets to assess the performance of alternative mechanisms. In particular, we should be able to compare the allocative efficiency of simultaneous one-shot versus ascending auctions, as well as simultaneous versus sequential market designs.

Acknowledgments

This research was supported in part by Grant CCF-0905139 from the U.S. National Science Foundation. We are grateful for helpful comments from Amy Greenwald, Eric Sodomka, and members of the Strategic Reasoning Group at the University of Michigan AI Laboratory.

References


