Exploring Bidding Strategies
for Market-Based Scheduling *

Daniel M. Reeves, Michael P. Wellman, Jeffrey K. MacKie-Mason,
Anna Osepayshvili

University of Michigan
Ann Arbor, MI 48109 USA
`{dreeves,wellman,jmm,annaose}@umich.edu`

Abstract

A market-based scheduling mechanism allocates resources indexed by time to alternative uses based on the bids of participating agents. Agents are typically interested in multiple time slots of the schedulable resource, with value determined by the earliest deadline by which they can complete their corresponding tasks. Despite the strong complementarities among slots induced by such preferences, it is often infeasible to deploy a mechanism that coordinates allocation across all time slots. We explore the case of separate, simultaneous markets for individual time slots, and the strategic problem it poses for bidding agents. Investigation of the straightforward bidding policy and its variants indicates that the efficacy of particular strategies depends critically on preferences and strategies of other agents, and that the strategy space is far too complex to yield to general game-theoretic analysis. For particular environments, however, it is often possible to derive constrained equilibria through evolutionary search methods.

Key words: market-based scheduling, evolutionary game theory, evolutionary search, replicator dynamics, simultaneous ascending auctions, Nash equilibrium, bidding strategies

---

* This is a revised and extended version of a paper appearing in the *ACM Conference on Electronic Commerce* (EC'03), June 2003.

1   Introduction

1.1   Strategies for Complex Market Games

Consider a set of agents who have values over a set of available resources. A unit of a particular resource can be employed only by one agent, and though multiple units may be available, the demand for at least some resources at zero prices is greater than the finite supply available. Deciding how to assign the available resources to agents is an *allocation problem*. This characterization of the problem encompasses many complex planning and coordinating activities.

Solution methods for allocation problems frequently are developed under the assumption of centrally available information, or distributed information with cooperative behavior. When a centralized approach is possible, it will in general yield results superior to any purely decentralized method. Nevertheless, centralized methods are not directly applicable when agents have distinct interests and privately held information about the requirements for and values of possible uses. We cannot rely on the agents to reliably communicate such information to the center, as the center's use of the information to determine an allocation will typically create incentives for the agents to misrepresent their situations in order to obtain more advantageous results.

A *resource allocation mechanism* defines a structured communication process that determines which agents get which resources based on messages exchanged. The field of *mechanism design* considers how to organize such mechanisms taking into account the information and incentives facing the participating agents. Typically, carefully designed mechanisms induce agents to reveal essential information via monetary transfers tied to the messages and resources allocated.

Markets comprise a special class of resource allocation mechanisms in which agents exchange resources for money, at *prices* determined through communication of offers, or *bids*. When the price-determination process is mediated, and follows explicit rules mapping bids into allocations, the mechanism constitutes an *auction*. A set of agents interacting through auctions defines a *market game*, with payoffs representing value of market outcomes, which in turn are a function of joint bidding strategy. Market games are typically characterized by incomplete information, complex dynamics, and large sets of possible actions (bids).

Market games corresponding to even moderately complex scenarios are notoriously difficult to solve. That is, except for the simplest market mechanisms (e.g., a one-shot auction for a single item, or a mechanism specially designed to have dominant strategies), deriving a Bayes-Nash equilibrium is not analytically tractable.

Strategic complexity presents a particularly difficult problem when resources may

2

be complements for some agents. Complementarity manifests when an agent's value for a particular resource is greater if it also obtains one or more other resources. For example, an airline passenger may wish to obtain two connecting segments to complete a trip. The airline, meanwhile, needs to obtain reservations for both a takeoff slot and a landing slot for each flight segment. When an agent must bid for one resource with uncertainty about the market resolution of complements, its decision presents risky tradeoffs.

Given the commonality of market games involving complementary resources, it is perhaps surprising that very little is known about optimal strategies for such games. For example, except under very restrictive assumptions, we do not know the optimal bidding strategy in multiple item simultaneous ascending auctions (discussed below). The explorations reported here represent our effort to address this large gap in strategic understanding, in the context of the simple scheduling domain.

1.2 Market-Based Scheduling

Our investigations apply to the general problem of strategy exploration in complex allocation mechanisms. For concreteness, however, we study an important instance of this class: *scheduling* problems. A problem can be described as *scheduling* when the resources are distinguished (at least in part) by the time periods in which they are available, so that a schedule is an allocation of these resources over time. Scheduling arises at the core of problems in manufacturing, telecommunications, logistics, and many other common contexts where reserving resources is called for. It has of course been extremely well studied, by researchers in operations research, computer science, and artificial intelligence [4,33]. Most scheduling methods are centralized, in effect requiring that relevant information be globally available.

A configuration of markets that allocates resources over time defines a *market-based scheduling* mechanism. Computational markets of many kinds have been proposed for a variety of scheduling domains, including time-shared computer systems [28], airport time-slot allocation [26], and railroad track allocation [3].

We focus on the strategic problem faced by an agent participating in a market-based scheduling mechanism. We address scheduling in particular because it represents an intrinsically important class of problems, and the temporal pattern of available resources is often significant in problems beyond scheduling. Moreover, restricting attention to scheduling (indeed, a particularly simple version) helps to focus our investigation and provides some structure constraining the problem.

When there are strong interdependencies in agent preferences for distinct goods, mechanism designers often recommend that the scope of the allocation mechanisms be extended to consider the resources together. One increasingly popular class of mechanisms taking this approach is the *combinatorial auction*, where agents submit offers for bundles of goods, and the auction calculates an allocation based on some global optimization criterion [7].

However, combinatorial auctions are often not practical, for example because of the difficulty of coordinating the allocations of the various resources, which may have overlapping yet distinct groups of potentially interested agents. Regardless of the reason, it is a simple fact that real-world markets quite typically operate separately and concurrently despite significant interactions in preferences or costs. Thus, the problem of dealing with simultaneous markets for related goods is a ubiquitous one, and though it might seem that the problem can be alleviated in many cases by widening the scope of the market, this is not a universally applicable solution.

Note that in some cases allocating multiple resources through simultaneous auctions is a deliberate design choice, for example (until quite recently) in the series of FCC spectrum auctions [15]. In some it is a result of separate initiation of commonly-operated auctions (e.g., all of the auctions running concurrently—despite potential interactions—on eBay), and in many others a result of completely separate initiation and operation of markets.

Although most of the literature on auction theory deals with mechanisms mediating a single resource [13], some recent work has addressed the issues of simultaneous auctions for multiple goods. The main lesson seems to be that simultaneous ascending auctions tend to work well when there exists a competitive price equilibrium. A sufficient condition for competitive equilibrium is that goods are substitutes [11,2]. This is not altogether surprising given that substitutability is the standard condition for stability of the tatonnement protocol [14]. It is also well-established that simultaneous ascending auctions can fail miserably whenever there are any complements [18,9].

Simultaneous auctions are quite common even when the gross substitutes condition does not hold.[1] Thus, an understanding of how agents *should* behave when faced with separate markets for complements would constitute useful knowledge for strategists as well as mechanism designers.

---

[1] For example, complements are patently important in the FCC spectrum allocation problem, yet separate markets were chosen over a combinatorial auction despite its high stakes and the authority of a single entity to set the mechanism.

2 Scheduling Problem Definition

In the simple scheduling problem we consider [32], there are M units (called *time slots*) of a single schedulable resource, indexed $1,\ldots,M$. Each of N agents has a single job that can be accomplished using the resource. Agent j's job requires λ_j time slots to complete, and by accomplishing this job it obtains some value depending on the time it completes. Specifically, if j acquires λ_j time slots by deadline t, it accrues value $v_j(t)$. Deadline values are nonincreasing: $t < t'$ implies $v_j(t) \geq v_j(t')$.

If $\lambda_j = 1$ for all j, we call the scheduling problem *single unit*. Problems violating this constraint are *multiple unit*. If each agent j has a single deadline (there exists a T_j such that $v_j(t) = v_j(t') > 0$ for all $t, t' \leq T_j$, and $v_j(t'') = 0$ for all $T_j < t'' \leq M$), we call the problem *fixed deadline*. If $v_j(t) > v_j(t') > 0$ for some j, t, t' (i.e., j accrues greater value for finishing the job sooner), then we call the problem *variable deadline*.

3 Ascending Auctions

3.1 Mechanism

In the *simultaneous ascending auction* for scheduling, a separate auction is run for each slot. Each auction can have multiple rounds of bidding. At any given time, the *bid price* on slot m is β_m, defined to be the highest bid b_j^m made thus far (or zero if there have been no bids). The *ask price* for slot m is defined to be $\alpha_m = \beta_m + \varepsilon$. To be admissible, a bid must satisfy $b_j^m \geq \alpha_m$. If an auction receives multiple admissible bids in a given round, it admits the highest (breaking ties arbitrarily). An auction is *quiescent* when a round passes with no new admissible bids.

The auctions proceed concurrently. When all of them are simultaneously quiescent, the auctions all close and allocate their respective slots per the last admitted bids. Because no slot is committed until they all are, an agent's bidding strategy on one slot cannot be contingent on the outcome for another slot.

3.2 Straightforward Bidding

In order to analyze the overall market *protocol*, we evaluate how the simultaneous auction mechanism performs when agents pursue particular strategies. We begin

with a baseline strategy called *straightforward bidding* (SB).[2] A straightforward bidder takes a vector of *perceived prices* for the slots as given, and bids those prices for the bundle of slots that would maximize the agent's surplus if it were to win all of its bids at those prices.

If agent j is assigned a set of slots X, it accrues value $v_j(X)$ based on the best deadline it can achieve: $v_j(X) = v_j(X(\lambda_j))$, where $X(t)$ is the tth time slot in X. Given that it obtains X at prices $\vec{p}$, the agent's *surplus* is its value less the amount paid, $\sigma(X, \vec{p}) = v_j(X) - \sum_{m \in X} p_m$. When agent j is winning the set of slots X_{-1} in the previous bidding round, we define the current perceived prices to be $\hat{p}_m = \beta_m$ for $m \in X_{-1}$, and $\hat{p}_m = \alpha_m$ otherwise. Then, under SB, agent j bids $b_j^m = \hat{p}_m$ for $m \in X^*$ such that $X^* = \arg\max_X \sigma(X, \vec{\hat{p}})$.

3.3 Baseline Strategy Performance

The straightforward bidding strategy is quite simple, involving no anticipation of other agents' strategies. For the single-unit problem, such anticipation is unnecessary, as the agent would not wish to change its bid even after observing what the other agents did [2]. This is called the *no regret* property [10], and means that from the agent's perspective, no bidding policy would have been a better response to the other agents' bids.

A single-unit, fixed-deadline problem in which all agents have the same deadline (M without loss of generality), is equivalent to a problem in which all buyers have an inelastic demand for a single unit of a homogeneous commodity. For this problem, Peters and Severinov [22] showed that straightforward bidding is a perfect Bayesian equilibrium.

Up to a discretization error, the allocations from simultaneous ascending bid auctions are efficient when agents follow straightforward bidding. We have shown elsewhere [32] that the final price vector will differ from the minimum unique equilibrium price by at most $\epsilon\kappa$, where $\kappa \equiv \min(M, N)$. The value of the allocation, defined to be the sum of the bidder surpluses, will differ from the optimal by at most $\kappa\epsilon(1+\kappa)$. A similar bound was established by Bertsekas [1] in a more general setting.

Unfortunately, the very nice properties for straightforward bidding in single-unit problems do not carry over to multiple-unit problems. Indeed, the resulting price vector can differ from the minimum equilibrium price vector, and the allocation value can differ from the optimal, by arbitrarily large amounts [32].

---

[2] We adopt the terminology introduced by Milgrom [18]. The same strategy concept is also referred to as "myopic best response", or "myopically optimal", or even "myoptimal" [12].

Name	Job Length	Deadline	Value
Agent 1	1	2	5
Agent 2	2	2	8

Table 1
A simple problem illustrating the pitfalls of SB (Example 1).

Example 1 *There are two agents, with values as shown in Table 1. The bid increment is* $\varepsilon = 1$. [3]

One admissible straightforward bidding path leads to a state in which agent 2 is winning both slots at prices $(4,3)$. *Then, in the next round, agent 1 would bid* $b_1^2 = 4$. *The auction would end at this point, with agent 1 receiving slot 2 and agent 2 receiving slot 1, both at a price of* 4.

In this example, SB leads to a result with value 5, whereas the optimal allocation would produce a value of 8. Adding slots and agents would enable constructing slightly more complex examples, magnifying the suboptimality to an arbitrary degree.

4 Alternative Bidding Strategies

We have seen that straightforward bidding fails to guarantee high quality allocations except in highly restricted problems. It is also quite easy to show that straightforward bidding is not an equilibrium strategy in general.

Consider again Example 1. With SB agents, the mechanism reached quiescence at prices $(4,3)$. However, it is not rational for agent 2 to stop at this point. If, for example, it offered $b_2^2 = 5$, the auction would end (assuming agent 1 is a straightforward bidder), and agent 2 would be better off, with a surplus of -1 rather than -4.

It is clear that SB is not a reasonable candidate for a general strategy in the simultaneous ascending auction for simple scheduling.

4.1 Equilibrium Strategies in the Scheduling Game

Finding equilibrium strategies for even the relatively simple scheduling problem we have defined is a daunting task. First, the strategy space is enormous. An agent's preferences are described by a job length plus a potentially different payoff value

---

[3] We adopt this unit bid increment and specify integer values for achieving jobs in all subsequent examples as well.

for each of M deadlines. Therefore, the space of joint preferences is $(M+1) \times N$ dimensional. The state information that agents have consists of the price-quote history. When the bid increment ε is small relative to the range of agent valuations, the number of bidding rounds can be quite large. Each round will generate an M-vector of current bid prices $\vec{\beta}$. The strategy space is all functions mapping the Cartesian product of the space of preferences and the space of all price-quote histories into a vector of next-round bids. Finding an optimal strategy by enumeration will be computationally infeasible for any nontrivial problem instance in this class.

We might hope that thoughtful reflection on the structure of the scheduling problem might lead us to a workable set of plausible strategies, within which the search for an optimal strategy would be manageable. However, our first example above rules out SB as a candidate, and even a small amount of further exploration reveals that best-response behavior in simple examples is highly sensitive to the realizations of agent preferences, and to the strategies of other agents.

A desirable approach to search for optimal strategies is to analytically derive them from the optimality criterion and certain axiomatic restrictions on rationality. Unfortunately, optimal strategies are rarely found for problems of even moderate complexity, such as ours. At least one difficulty in interesting schedule problems that we believe makes analytic derivation of optimal strategies extremely difficult is the presence of complementarities in preferences: that is, the value to an agent for a given slot typically depends on whether or not the agent acquires one or more other slots. This presents the agent with the so-called *exposure problem*, where in order to obtain the combination it prefers it must expose itself to the risk it will get caught paying for a far less desirable (in our scheduling problem, completely worthless) subset. Managing such exposure presents a complex risk assessment and decision problem, and we have as yet seen no evidence that a general solution is forthcoming.

Rather than restrict our analyses of scheduling mechanisms to their performance when agents implement ad hoc bidding strategies we know are arbitrarily far from optimal, we propose a method for directed search to find improved strategies. The idea is not complicated. In our work to date, we select a set of candidate strategies, and then evaluate their performance against each other through a statistical simulation based on an evolutionary game. Strategies are assigned population frequencies, and samples of agents compete against each other. Strategies that perform relatively well are rewarded with higher population frequencies. Thus, through what amounts to a structured Monte Carlo simulation, poor strategies are weeded out.

We describe our evolutionary game tool in greater detail in Section 5 below. First, we illustrate our specification of the strategy space with a concrete example.

4.2   Variant Strategy: "Sunk Awareness"

We showed in Example 1 that in some problems agents following a straightforward bidding strategy may stop bidding prematurely. A bit of reflection indicates why straightforward bidding is failing in this situation. In a given round, agents bid on the set of slots that, at the current ask prices, will maximize their surplus. If no configuration of slots would yield positive net surplus, the agent chooses not to bid, because the alternative is to earn negative surplus. However, this behavior ignores outstanding commitments: the agent may already be winning on one or more slots. If the agent drops out of the bidding, and others do not bid away the slots the agent already is winning, then its alternative surplus is not zero, but negative the sum of bid prices for the slots the agent wins. Thus, this failure of straightforward bidding is due to ignoring the true opportunity cost of not bidding.

We refer to this property of straightforward bidding as "sunk unawareness". Agents are bidding as if the incremental cost for slots they are currently winning is the full price. However, since they are already committed to purchasing these slots (if another agent does not raise the bid price), the cost is sunk, and the incremental cost is zero.

Given this clear failure of straightforward bidding, we have parameterized a family of strategies to permit agents to account to a greater or lesser extent for the true incremental cost for slots they are currently winning. We call this strategy "sunk aware". A sunk-aware agent bids as if the incremental cost for slots currently winning is somewhere on the interval of zero and the current bid price. In Section 6, we report on our experiments to discover the optimal setting of the sunk awareness parameter.

We now formalize the sunk-aware strategy family. Define agent j's *perceived price* for slot m to be $k\beta_m$ if the agent j is currently winning slot m, and $\beta_m + \varepsilon$ otherwise. The sunk awareness parameter is $k \in [0,1]$. If $k = 1$ the strategy is identical to straightforward bidding. At $k = 0$ the agent is fully sunk aware, bidding as if it would retain the slots currently winning with certainty. Intermediate values are consistent with bidding as if the agent puts an intermediate probability on the likelihood of retaining the slots it is currently winning.

5   Searching for Restricted Equilibria

We explore bidding strategies for our scheduling market by searching within a small set of candidate strategies for a Nash equilibrium (with respect to *ex ante* expected payoffs), then extending the search to new strategy sets. To do this we first convert our game from extensive to strategic form, by using Monte Carlo simulation to

generate an expected payoff matrix for every combination of the strategies playing against each other. Given the expected payoff matrix, we find Nash equilibria with one of three methods: replicator dynamics (an evolutionary tournament), GAMBIT (a computational game solver), or amoeba (a function minimization search algorithm). In the remainder of this section we describe each of these tasks.

5.1   Generating Payoff Matrices

We have implemented a simulator for the mechanism described in Section 2. We estimate the payoff matrix for a *restricted game*, in which agents are permitted to play only strategies drawn from a restricted set.

An instance of the restricted game is defined by

- the auction rules (e.g., simultaneous ascending bid; minimum bid increment ε),
- a number of slots M,
- a number of participating agents N,
- the distribution of agent preferences, and
- a finite set S of strategies permitted to agents.

Each strategy is a function that maps agent preferences plus available auction information to a set of bids. For our simulations, we construct agents implementing selected strategies. We calculate the expected payoffs with respect to specified distributions from which agent preferences are drawn. See Section 6 for specifics on the preference distributions.

As discussed in Section 4.1, the space of strategies is enormous. In the present work, we explore a very restricted region of this space. First, we consider only *reflex* agents: they consider only information from the current auction round (prices and which slots the agent is winning), not from previous rounds. We further consider only a specific parameterized family of strategy functions defined on this restricted space of preferences and information. Our family of strategy functions is based on SB, generalized in six dimensions: sunk awareness, price monotonicity belief, price time bias, reluctance, price aggressiveness, and slot aggressiveness. [4] For the explorations reported in this paper, we vary only the sunk awareness parameter (k).

With the game parameters and agent strategies specified, we compute the expected *payoff matrix*. An element in the matrix is an N-vector of expected payoffs associated with a particular *strategy profile*. A strategy profile for an N-player game is a list of the specific strategies that each player follows. For example, in a five-player game in which two players play with sunk awareness $k = 0.5$ and the others play

---

[4]   The terms are suggestive of the intent of the corresponding parameters. The precise specification of these parameters is omitted for space.

$k = 1$, the profile is $\{0.5, 0.5, 1, 1, 1\}$. There is a distinct element in the payoff matrix for each of the $\binom{N+|S|-1}{N}$ possible strategy profile combinations, where $|S|$ is the number of permissible strategies.

To estimate an entry of the expected payoff matrix, our Monte Carlo simulator repeatedly draws preferences and assigns them to agents, simulates the auction protocol for the given strategy profile to quiescence, and averages the resulting surpluses. Note that this means that the values used in the payoff matrix are only estimates. In Section 7, we consider the problem of determining how sensitive the derived equilibria are on the sample sizes used to estimate the payoff matrices.

5.2 Evolutionary Search for Equilibria

In his original exposition of the concept, Nash [19] suggested an evolutionary interpretation of the Nash equilibrium. We use the *replicator dynamics* formalism, introduced by Taylor and Jonker [29] and Schuster and Sigmund [27]. If the probabilities in a mixed strategy are cast as proportions of a large population of agents playing the corresponding pure strategies, then an agent population that has reached a fixed point with respect to the replicator dynamics will be a candidate symmetric mixed strategy Nash equilibrium. Weibull [31] shows that for two-player, two-strategy games, fixed points of a broad class of replicator processes are Nash equilibria if neither strategy is extinct. For N-player games, the set of fixed points that are locally asymptotically stable (all states sufficiently close converge to the same state) are a subset of the set of Nash equilibria (NE) [8]. A fixed point (i.e., a stationary state) is a population in which every pure strategy with more than zero representatives in the population does as well in expectation against $N-1$ strategies drawn randomly from the population as N strategies drawn randomly from the population do against each other. In other words, every pure strategy is doing as well as every other, given the population proportions.

This definition suggests an iterative (evolutionary) algorithm for finding an equilibrium in NE. We choose an initial population proportion for each pure strategy in the permissible set, and then update them in successive generations so that strategies that perform well increase in the population at the expense of low-performing strategies. The proportion $p_g(s)$ of the population playing strategy s in generation g is given by

$$p_g(s) \propto p_{g-1}(s) \cdot (EP_s - W),$$

where EP_s is the expected payoff for pure strategy s against $N-1$ players all playing mixed strategies according to the population proportions, and W is a lower bound on payoffs (e.g., the minimum value in the payoff matrix) which serves as a dampening factor. To calculate the expected payoff EP_s from the payoff matrix, we average the payoffs for s in the profiles in which it appears, weighted by the probabilities of those profiles. The probability of a particular profile $(n_1, ... n_{|S|})$, where

n_s is the number of players playing strategy s, is

$$\frac{N!}{n_1! \cdot ... \cdot n_{|S|}!} \cdot p(1)^{n_1} \cdots p(|S|)^{n_{|S|}}.$$

If the population update process reaches a fixed point, then the final population proportions are a candidate mixed strategy equilibrium. We verify directly that the candidate is indeed a static Nash equilibrium by checking that the evolved strategy is a best response to itself.[5] In all the experiments reported here, this process indeed reaches a fixed point and these fixed points always correspond to Nash equilibria. However, we have found examples for which the replicator dynamics do not converge and the population proportions cycle.

5.3 Solving Payoff Matrices with Gambit

GAMBIT [17] is a tool for solving finite games. It takes the full matrix representation of a strategic form game and proceeds by iteratively eliminating strongly dominated strategies and then applying the simplicial subdivision algorithm [16] to enumerate all Nash equilibria.

The problem with using GAMBIT in its current implementation is that it cannot take advantage of symmetry in a payoff matrix. This means that the matrix will consist of $|S|^N$ cells.[6] The other approaches described in this section take advantage of a sparse representation of the payoff matrix that exploits the symmetry and requires only $\binom{N+|S|-1}{N}$ cells. For example, with $N = |S| = 5$, symmetry reduces the number of distinct profiles to 126, versus 3125 for the full matrix representation.

---

[5] Friedman [8] has shown in a fairly general setting that a dynamically stable equilibrium of the evolutionary game is also a Nash equilibrium of the static game. However, we cannot strictly rely on this result. First, our numerical results at best identify an *approximate* fixed point, with a finite error tolerance (typically 10^{-10}). Second, a fixed point is only a necessary condition for a Nash equilibrium; sufficiency requires verifying that the fixed point is Lyapunov stable. There is also a regularity condition for Friedman's results—that the dynamics are continuous, not discrete in time—that we do not satisfy (though Friedman conjectures that the generalization holds).

[6] This computational burden is not trivial. For example, in many experiments we have run on five-player/five-strategy games, GAMBIT took hours or sometimes days to find all the Nash equilibria when it could find them at all.

5.4   Searching for Equilibria with Amoeba

One of the many characterizations of a (symmetric) Nash equilibrium is as a global minimum of the following function from mixed strategies to the reals:

$$f(p) = \sum_{s \in S} \max[0, u(s, p) - u(p, p)]^2,$$

where $u(x, p)$ is the payoff from playing strategy x against everyone else playing strategy p. The function f is bounded below by zero and in fact for any $p \in$NE, $f(p)$ is zero. This is because $f(p)$ is positive iff any pure strategy is a strictly better response than p itself.

For our experiments, we use an adaptation of the amoeba algorithm [24] developed by Walsh et al [30]. Amoeba is an implementation of the Nelder-Mead algorithm [20] for highly nonlinear function minimization.

5.5   Replicator Dynamics and Biased Sampling

Based on the observation that estimating the cells of the payoff matrix is far more compute-intensive than finding equilibria, we have explored another use for our evolutionary game simulator: as a biased-sample approach to generating the expected payoff matrix. For this method, we do not calculate the payoff matrix first, then search for equilibria; rather, we tackle the two simultaneously. We start with an initial set of population proportions for each pure strategy. Then, as before, we repeatedly sample from the preference distribution, iterating our auction mechanism to quiescence. However, now strategies are randomly drawn to participate according to their population proportions. Then, after a relatively small number of samples—long before we have confidence that they are precise estimates of the expected payoffs—we apply the replicator dynamics using the realized average payoffs. Then, given the new population proportions, we iterate, calculating a sequence of new generations, except that for each generation we retain the accumulated estimate of average payoffs from previous generations, and calculate a correctly-weighted average of the old and new sample information. The iteration of generations continues until the population proportions are stationary with respect to the replicator dynamics.

The above procedure straightforwardly accumulates a statistically precise estimate of the expected payoff matrix. However, the sampling is biased: strategies that are more successful (and thus more highly represented in the population) are sampled more often. We conjecture that this approach would be more efficient than uniform sampling, especially for problems with a large number of permissible strategies.

13

Since many of them will likely not be present in a Nash equilibrium—that is, their population fractions will converge to zero—extensive sampling to lower the standard error of the estimated payoff would be wasteful. However, we have found it easier to study the dynamics of our experiments when the full payoff matrix is calculated to high precision before the replicator dynamics are applied, and for the relatively small problems we simulate here the computational cost has been manageable.

6   Experiments in Sunk-Awareness

We have begun our search in strategy space with a systematic exploration of bidding strategies that vary on the sunk-awareness parameter (k). We consider parameter settings in multiples of $1/20$ from 0 to 1. For simplicity of reference, we designate strategies by an integer from 0 to 20, such that strategy i refers to the baseline agent with $k = i/20$. For all of these experiments, we used the following game parameters:

- number of slots $M = 5$,
- number of agents $N = 5$ (except in Section 6.4),
- bid increment $\varepsilon = 1$.

We varied the distributions from which preferences are drawn. The primary structural distinction we have explored is with respect to the distributions of job lengths, λ. In the *uniform* model, job length is $\sim U[1,M]$, that is, $\Pr(\lambda = l) = 1/M$, $l \in \{1,\ldots,M\}$. In the *constant* model, λ_j is fixed at a particular value for all j (in all of the trials we report, $\lambda_j = 2$). Finally, in the *exponential* model, we draw job length from an exponential distribution. Specifically,

$$\Pr(\lambda = l) = \begin{cases} 2^{-l} & l = 1,\ldots,M-1 \\ 2^{-M+1} & l = M. \end{cases}$$

In all three of these models, the deadline values for each slot are initialized as integers $\sim U[1,50]$, but then modified to ensure monotonicity:

$$v(t) \leftarrow v(\min\{i \geq t \mid v(i) \leq v(t-1)\}) \ or \ 0, \ \ t = \lambda+1,\ldots,M.$$

In words, iterate through the deadline values and whenever one violates monotonicity (i.e., exceeds its predecessor) set it to the earliest later deadline value that restores monotonicity (i.e., is less than or equal to its predecessor).

Finally, in Section 6.4, we explore games with varying number of agents (players) within the exponential preference model.
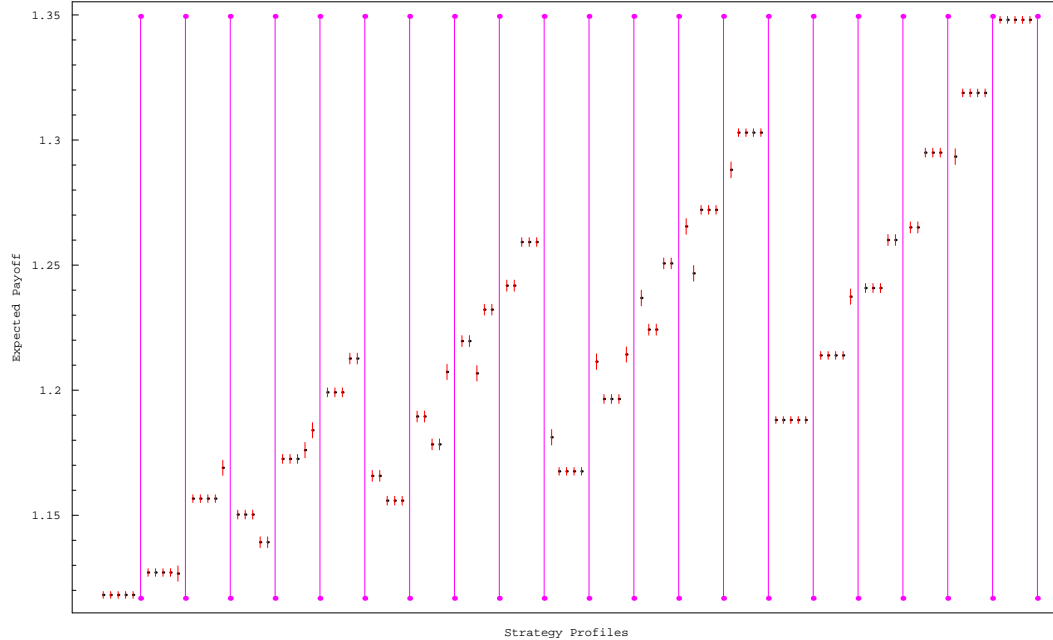
14

Fig. 1. Payoff matrix for strategies 18,19,20 with uniform preferences. Each column corresponds to a strategy profile: {18,18,18,18,18} through {20,20,20,20,20} in lexicographic order. The jth dot within a column represents the mean payoff for the jth strategy in the profile. This payoff matrix is based on over 45 million games simulated for each of the 21 profiles, requiring weeks of cpu time. The error bars denote 95% confidence intervals.

6.1 Uniform Job Length

In Figure 1 we offer a representation of the payoff matrix for the restricted game with strategies 18, 19, 20, and uniformly distributed job length. To illustrate, the first column represents the payoffs for the strategy profile {18,18,18,18,18}. Each strategy in this profile receives the same payoff (since each agent is playing the same strategy and the game is symmetric) of about 1.12. The second column presents the payoffs for {18,18,18,18,19}. When playing against one $k = 19/20$ agent, the other four $k = 18/20$ agents now do better than in the all-18 profile, and very slightly better than the sole $k = 19/20$ agent does in this profile. When one agent deviates from 18 to 20 (third column), it does noticeably better and so do the agents playing 18.

In Figure 2 we show the result of running the payoff matrix through the replicator dynamics. The population evolves to all playing 20 ($k = 1$). This is in fact a Nash equilibrium as can be seen by noting that the all-20 profile in the payoff matrix (Figure 1) scores higher than any unilateral deviation. In this restricted game, 20 is a dominant strategy (this can be verified, albeit tediously, by inspecting the payoff matrix), and hence the only Nash equilibrium.

We have confirmed this result with both GAMBIT and our implementation of amoeba.
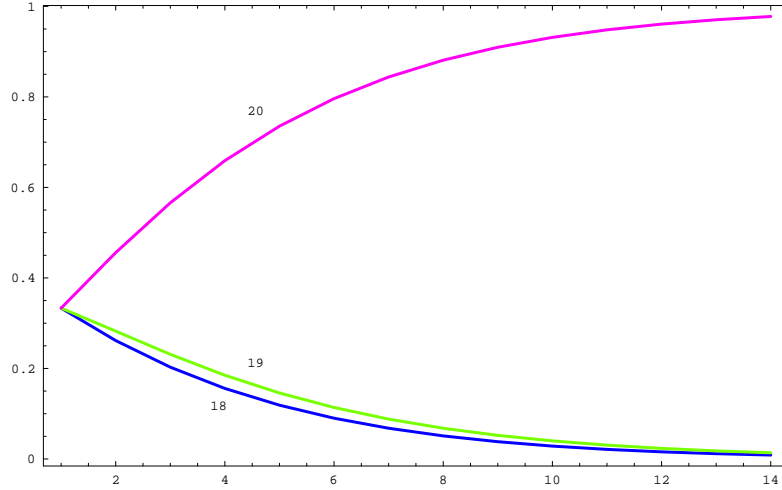
Fig. 2. First 14 generations of replicator dynamics for strategies $20k \in \{18, 19, 20\}$ in a game with uniform preferences. Strategy 20 quickly takes over the population, hence the evolved equilibrium is (0,0,1) meaning everyone plays 20 (i.e., $k = 1$).
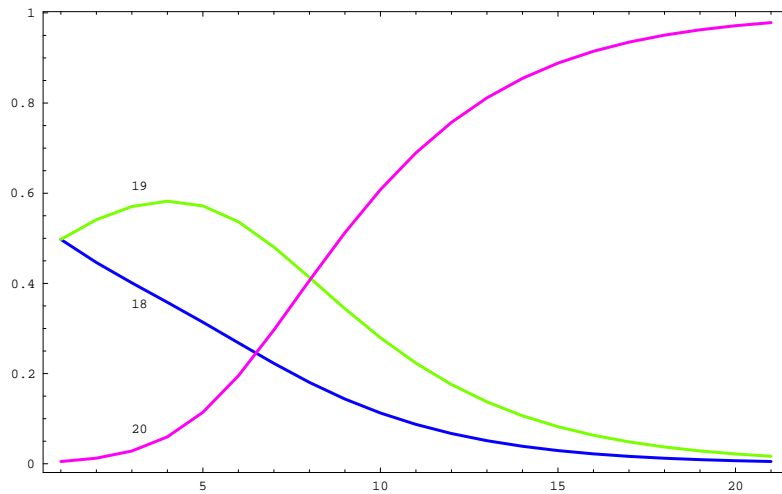

Fig. 3. Replicator dynamics for the same game as shown in Figure 2, but with 100 times fewer 20s in the initial population as 18s or 19s.

We also find that the replicator dynamics converges to the unique Nash equilibrium from various initial population proportions (for example, see Figure 3). Note that the strategy $k = 1$ corresponds to straightforward bidding (no sunk awareness).

6.2 Constant Job Length

It is not the case that $k = 1$ is always a dominant strategy. This follows immediately from the discussion above (Section 4) for the unrestricted game. Even when we restrict strategies to straightforward bidding extended only by the sunk awareness parameter (k), we can find environments in which $k = 1$ does not dominate.

In our experiments with constant job length, we fix $\lambda_j = 2$ for all j, and we consider a slightly larger set of strategies. In our first run, we consider strategies 16, 17, 18, 19, and 20. We present the evolutionary dynamics for our estimate of the expected payoff matrix in Figure 4. The payoff matrix required 22 million game simulations for each of the 126 strategy profiles. When run through our replicator dynamics, the system evolves to $\{0.745, 0.255, 0, 0, 0\}$ which constitutes a mixed-strategy Nash equilibrium. Convergence to this equilibrium is robust to a variety of different initial population proportions. Note that in this environment, the baseline sunk-unaware strategy is not even supported. Instead, the most sunk-aware (i.e., lowest k) strategies have greatest weight in the mixed strategy. GAMBIT reveals that strategies 19 and 20 are dominated.

We have not successfully verified that this is a unique equilibrium; GAMBIT (which attempts to find all equilibria) was not able to find a symmetric equilibrium for this game after days of cpu time (it did find one asymmetric equilibrium). Amoeba takes about 15 minutes to find this equilibrium.
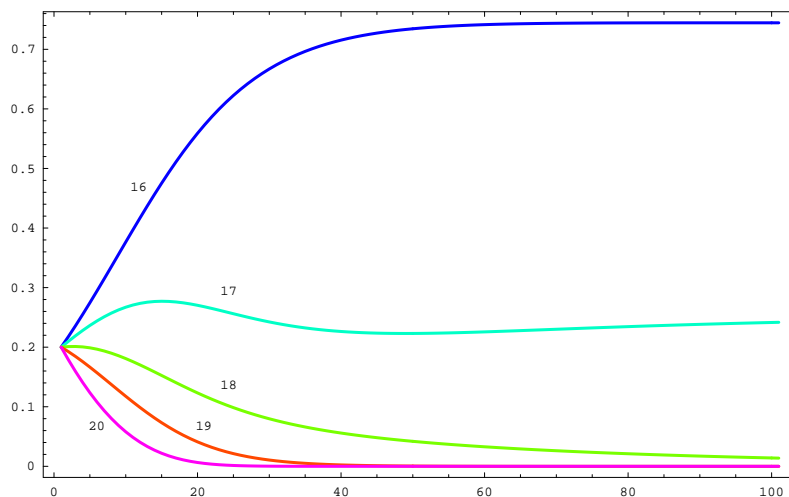

Fig. 4. Replicator dynamics for strategies $20k \in \{16, ..., 20\}$ in a game with uniform preferences but a fixed job length of two. The evolved Nash equilibrium is (0.745, 0.255, 0, 0, 0).

Given that 16 was the most heavily represented strategy when the game is restricted to the 16–20 range, it is natural to investigate whether lower values might perform better still. We tested the above game with a broader but coarser grid of strategies: 0, 8, 12, 16, and 20. We show the evolutionary dynamics in Figure 5 based on a payoff matrix estimated by eight million simulations per profile. The evolved Nash equilibrium is for everyone to play 16. According to GAMBIT, strategies 0 and 8 are dominated, and everyone playing 16 is the only Nash equilibrium.

This game stressed two of our solution methods: it took GAMBIT about a day of runtime to reach its conclusion. The amoeba algorithm did not find any Nash equilibria at all (though it identified a mixed strategy close to pure strategy 16 as nearly in equilibrium).
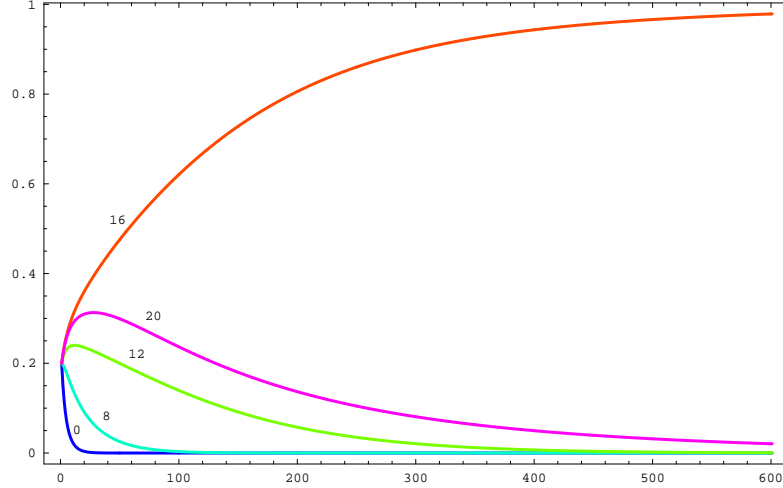
17

Fig. 5. Replicator dynamics for strategies $20k \in \{0, 8, 12, 16, 20\}$ in a game with uniform preferences but fixed job length of two. The evolved equilibrium is (1,0,0,0,0)—i.e., everyone play 16.

6.3 Exponential Job Length

Our final variation on the agent preference model applies exponential preferences. We present the evolutionary dynamics for the strategy set 16–20 in Figure 6, which is based on a payoff matrix estimated from 22 million samples per profile. The system evolves to $\{0, 1, 0, 0, 0\}$ (i.e., everyone play 17), which is a Nash equilibrium. This equilibrium is robust to initial population distribution using our evolutionary method. Amoeba does not find this equilibrium, but again identifies a nearby mixed strategy as close. GAMBIT determined that no strategy was dominated in this game, and was unable to determine whether the equilibrium is unique. From this configuration, the exponential model yields an observed equilibrium distribution for k intermediate between the uniform and constant models.

6.4 Varying Number of Players

The experiments reported above all employ a five-player game configuration. We have performed further trials varying the number of agents ($N = 2, 8, 10$), maintaining the other game parameters as in our standard setup, with exponentially distributed job length (Section 6.3). The objective of this variation was to exercise the methodology on a range of settings of game shapes, and to identify any systematic relation between N and the equilibria we find.
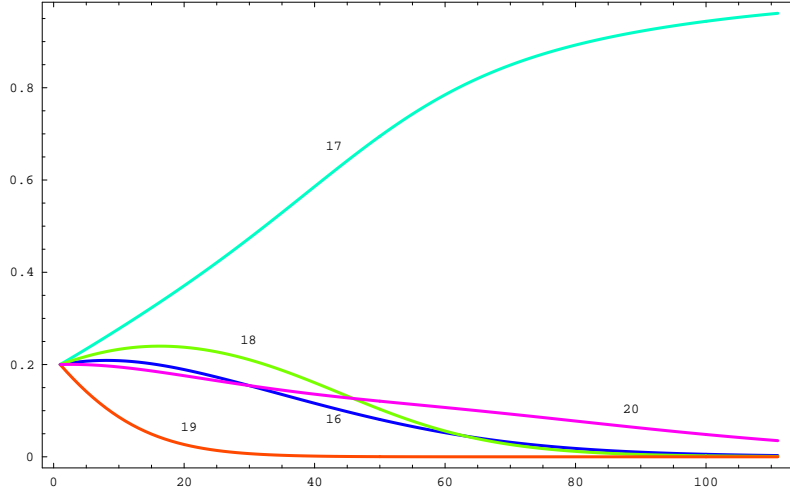
18

Fig. 6. Replicator dynamics for strategies $20k \in \{16, ..., 20\}$ in a game with exponential preferences. The evolved equilibrium is $(0, 1, 0, 0, 0)$—i.e., everyone play 17.

6.4.1 Two Agents

With only two players, we can consider a larger set of candidate strategies. For this experiment we investigated 14 strategies, defined by the set $20k \in \{0, 3, 6, 8, 10, 11, \ldots, 17, 18, 20\}$. This yields 105 profiles, for each of which we simulated 1.2 million games to construct the payoff matrix, depicted in Figure 7. Our replicator dynamics (shown in Figure 8) finds the Nash equilibrium in which all agents play 15. GAM-BIT identifies this as one of three equilibria (all symmetric) for this payoff matrix. All playing 14 is also Nash, as is the mixed strategy of playing 14 with probability 0.514 and 15 otherwise. Only strategies 14, 15, and 16 survive iterated elimination of dominated strategies. We see in Figure 8 that these are indeed the three most tenacious strategies under our replication process.

6.4.2 Eight and Ten Agents

With more than a handful of agents, it is not generally feasible to create a payoff matrix with more than a handful of strategies. Our experiments with eight and ten-player games employ a pool of four strategies: $20k \in \{10, 14, 17, 20\}$. This yields 165 profiles for the eight-player case, for which we simulated 1.5 million games per profile. For the ten-player case there are 286 profiles. We simulated 3.9 million games per profile, which took many cpu-weeks.

The conclusion for both eight and ten players is the same: $k = 1$ is a dominant strategy. In both cases, the replicator dynamics show strategy 20 overwhelming the population within 40 generations. For the eight-player case, GAMBIT confirms that 20 is dominant (and therefore also the unique Nash equilibrium). But for ten players, the raw payoff matrix (i.e., the normal form without exploiting symmetry) contains 10 million payoff values. GAMBIT is not able to use the more compact

Fig. 7. Payoff matrix for two-player game with exponential preferences and with strategies $20k \in \{0, 3, 6, 8, 10, 11, ..., 17, 18, 20\}$.
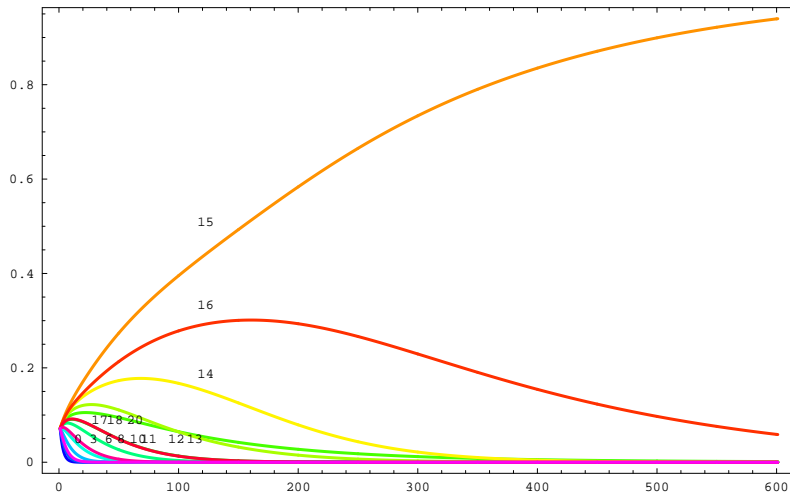

Fig. 8. Replicator dynamics for two-player game with exponential preferences with strategies $20k \in \{0, 3, 6, 8, 10, 11, ..., 17, 18, 20\}$.

symmetric representation, and in our installation, crashes trying to load this game into memory.

6.5 Discussion

In our experiments with exponential preferences, the equilibrium k value was monotone in the number of agents, N. This can be explained by observing that increasing

20

N can ameliorate the exposure problem. Consider the situation when the prices pass the threshold at which an agent stops bidding. The presence of more competing bidders increases the likelihood that the stopped agent will be let "off the hook" for its current winnings by being outbid. Therefore, it is less compelling for an agent to treat its current winnings as a sunk cost. In other words, *k* should be closer to 1 the more agents there are, which is what we find here.

7   Sensitivity Analysis: Robustness to Sampling Noise

As discussed in Section 5.1, the payoff matrices used in deriving the above results are estimates based on sampling. An important question is whether the equilibria we find are robust or would they change with further sampling? By the Central Limit Theorem, the mean of a sample from any distribution approximates a normal distribution, given enough samples. [7] Therefore, we have a probability distribution for each of the expected payoffs in the payoff matrix. By sampling from these distributions independently, we can generate a new, variant payoff matrix. If many such variant payoff matrices yield the same equilibrium results then we can conclude that we are insensitive to sampling noise. We expect this to be a conservative measure of sensitivity since it treats the errors in the expected payoffs as independent. So to the extent that the errors are correlated, we expect this measure will overestimate our sensitivity to sampling noise. We have in fact observed informally that our sensitivity analysis tends to report wider variances in equilibria than there turns out to be after gathering additional samples.

Several of our results reported above are impervious to sampling noise. This was determined by performing our equilibrium analysis on several thousand variants and finding that the equilibrium never changed. This was the case for the uniform preferences game reported in Section 6.1 and for the games with eight and ten players reported in Section 6.4.2. For our other results we find varying amounts of sensitivity. Figures 9, 10, 11, and 12 illustrate this by showing cumulative distribution functions for the equilibrium proportions of each of the strategies. The dotted vertical lines show the mean proportion for the corresponding strategy over all the variant payoff matrices sampled.

For example, we see in Figure 9 that for the fixed job length game reported in Section 6.2, the mean proportion of strategy 16 is identical to the proportion found for the maximum likelihood payoff matrix (using the actual sample means) and it varies according to a near-perfect normal distribution. We also see that strategy 19,

---

[7]   In fact, the sample mean follows a t-distribution with $n-1$ degrees of freedom where n is the sample size. With our sample sizes in the millions, the t-distribution is indistinguishable from the normal distribution. In other words, we can safely use the Central Limit Theorem's normal approximation.
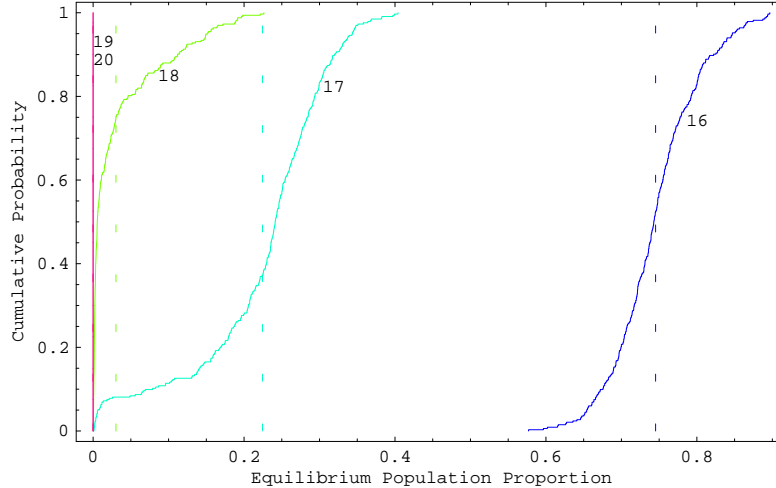
Fig. 9. Sensitivity analysis for payoff matrix for five-player game with uniform preferences but a fixed job length of two and strategies $20k \in \{16, \ldots, 20\}$. Compare to the replicator dynamics for the maximum likelihood payoff matrix for this game in Figure 4.

which died out for the maximum likelihood payoff matrix has a 10% chance of actually holding on to 5% of the population in equilibrium. Again, note that since this measure is conservative, the true equilibrium results are actually more likely to match those reported in Section 6 for the max-likelihood payoff matrices.

Figure 12 shows a clear need for more samples before we can give much credence to the equilibria reported in Section 6.4.1. However, we have run a smaller two-player experiment with nine strategies ($20k \in \{0, 3, 6, 8, 10, 12, 14, 17, 20\}$) where 14 was dominant and found this result to be perfectly robust to sampling noise. Therefore, the qualitative conclusions about this game are not seriously suspect. Nonetheless, we ran an additional three million simulations (per profile) and found some slight changes: strategy 16 no longer survives iterated elimination of dominated strategies and the mixed strategy equilibrium is skewed more towards strategy 14. There was no change in the pure strategy equilibria. The sensitivity analysis with the additional games shows greater robustness to sampling error with the additional samples.

8   Related Work: Evolutionary Search for Trading Strategies

Several prior studies have employed evolutionary techniques for the derivation of trading strategies. Price [25] demonstrates the use of genetic algorithms (GAs) for a variety of standard industrial organization games (e.g., Bertrand and Cournot duopoly). In Price's approach, the GA serves as an optimization method, employed to derive a best response. For instance, his GA model for the duopoly games comprises populations of strategies for each producer, each updated separately according to GA rules. The search is co-evolutionary in the sense that fitness statistics are derived by joint sampling from the pair of populations.
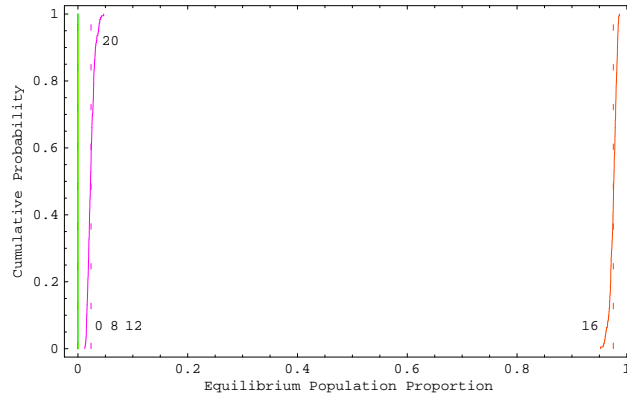
22

Fig. 10. Sensitivity analysis for five-player game with uniform preferences but a fixed job length of two and strategies $20k \in \{0, 8, 12, 16, 20\}$. Compare to the replicator dynamics for the maximum likelihood payoff matrix for this game in Figure 5.
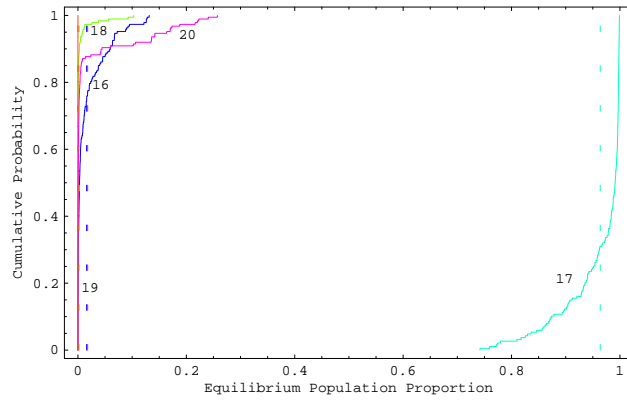

Fig. 11. Sensitivity analysis for payoff matrix for five-player game with exponential preferences and strategies $20k \in \{16, \ldots, 20\}$. Compare to the replicator dynamics for the max-likelihood payoff matrix for this game in Figure 6.
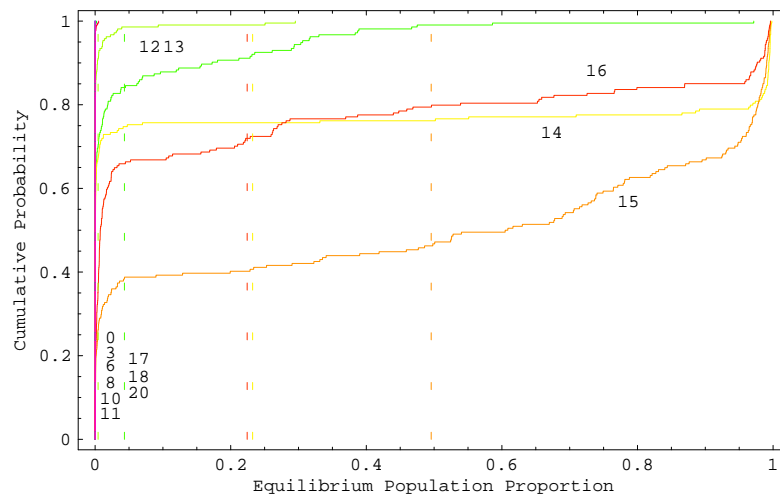

Fig. 12. Sensitivity analysis for two-player game with exponential preferences and with strategies $20k \in \{0, 3, 6, 8, 10, 11, \ldots, 17, 18, 20\}$. Strategy 15 is the only one with most of its mass significantly above zero. In Figure 8, 15 is the only strategy to survive.

Cliff [5] applied GAs to evolve improved versions of his "ZIP" trading strategy for continuous double auctions. Improvement in his study is defined in terms of convergence to competitive equilibrium prices, as opposed to surplus for particular agents. The evolutionary search, therefore, is for a high-performing homogeneous trading policy, rather than a strategic equilibrium. In more recent work, Cliff [6] expanded the search space to include a market-mechanism dimension, thus evolving a trading strategy in conjunction with an auction rule. As above, the GA's fitness measure is in terms of aggregate market performance, rather than individual profit.

Using a co-evolutionary approach similar to that of Price discussed above, Phelps et al [23] employ genetic programming to derive strategies for an electricity trading game studied by Nicolaisen et al [21]. They then extend the model to evolve auction rules in tandem with the trading strategies. Unlike Cliff, Phelps et al. evaluate fitness of the mechanism based on aggregate performance, while evolving trading strategies based on individual performance.

9   Best Response to SB

The foregoing account of our experiments suggests that our approach does not escape the curse of dimensionality: we cannot derive an unrestricted characterization of equilibrium behavior in the full strategy space. However, we have shown that focused simulations can reveal restricted equilibria in selected environments, and we continue to work on extending this method. An approach toward more general strategy recommendations would be to relax the restriction on one agent's strategy selection, while maintaining a focusing constraint on the others'. This would not yield an equilibrium result, of course, but it might establish a best response strategy to *some* environment the agent might face. Since prior researchers have given this strategy substantial attention, one natural candidate assumption about the other agents is that they are straightforward bidders. Thus, we pose the question: What is the best response to SB? And in particular, can the best response be usefully characterized as an SB variant? Unfortunately, it does not appear that this is the case, nor that the best response has any simple form, as indicated by the following examples. These examples demonstrate that, at least when an agent has strong priors about the preferences and strategies of other agents, its optimal behavior can be very far from "straightforward".

Example 2 *Let $N = M = 2$, with "our" agent requiring both slots for its job, which has value 250. Suppose the other agent (by assumption, a straightforward bidder) has job length 1, and one of the following deadline/value profiles, known to our agent:*

- *(100,98) (i.e., value of 100 for slot 1 or 98 for slot 2), or*
- *(200,170).*

Note that our agent will be profitable in the first case, but not the second.

If our agent in Example 2 bids straightforwardly, it will eventually reach a point where the prices reach or exceed the first of the possible profiles of the other agent's values. If prices go past these values, then the agent must either take a significant loss on one of the slots and stop bidding, or proceed to pay 370 to outbid the high profile. (If the other agent has the high profile, then the prices at the threshold point will be (100,70), so taking the loss of 70 is preferred to going all out and losing 120.)

A variant on the straightforward bidding strategy, SB', would distinguish the two profiles based on observed prices long before reaching the threshold of the low profile. If the second agent has the low profile, it will never bid the price of slot 1 more than two above the price of slot 2, since the gain in surplus from winning slot 1 is only two. If it has the high profile, however, it will bid the price for slot 1 as much as 30 higher than the price for slot 2. Therefore, if the other agent is actively bidding and the difference in prices between the two slots exceeds 2, the other agent must have the higher profile. By recognizing this and stopping intelligently when the price of the first slot reaches 4 or greater, our agent playing SB' would lose only 1 (what it bid for the second slot).

Example 3 *Consider a setup identical to that of Example 2, except that there is a third agent with job length 1 and job value of 20 for either deadline.*

SB' behaves qualitatively the same in this example, except that it would lose 21 rather than 1 in case the second agent has the high profile. The reason is that the third agent masks the behavior of the second, by alternately bidding up the prices for each slot (since their value is the same to this agent). The second agent's profile-dependent behavior is reflected in prices only when the third agent drops out. By then, SB' has offered 21 for slot 2.

A "smarter than straightforward bidding" strategy (SSB') would avoid this loss by actively bidding in a way to distinguish the cases earlier. Specifically, suppose SSB' offers 23 for slot 1 right away. The response of the straightforward second agent will immediately reveal whether it has a high or low profile. If high, it will outbid our agent, which can then drop out without having lost anything. If low, it can proceed straightforwardly to win a profitable job.

Example 4 *Consider a setup identical to that of Example 2, except that the (sole) other agent has possible profiles:*

- *(150,20), or*
- *(200,75).*

For this example, even SSB' cannot distinguish the two profiles until the difference in slot prices is at least 125, and moreover in the high profile case it has offered that

much for slot 1 when the other agent reveals itself by bidding for slot 2. It will then be stuck with that magnitude of loss.

We require a "smarter than smarter than straightforward" bidder, $SSSB'$, to realize that what it should do at this point is outbid the agent on slot 2 (a bid of 4 suffices), causing it to switch back to slot 1. At this point our agent stops bidding, and is stuck with the loss of -4 on slot 2, but this amount is quite small compared to what it would have lost if it had stopped with slot 1. Notice that this $SSSB'$ bidder behaves exactly as a sunk aware ($k = 0$) SB bidder (see Section 3.2) until the distinguishing information is revealed about its opponent's preferences, and then has the intelligence to drop out (long before an SB bidder would stop).

Examples 2–4 clearly illustrate the point that even relatively simple (two-slot) scenarios with one or two straightforward bidders can call for rather sophisticated bidding strategies in response. In fact, it is easy to construct scenarios where small distinctions about time slots in which our agent has no interest (i.e., past its latest deadline) resolve critical uncertainty with respect to the slots we do care about. Thus, we tend to be skeptical that any simple strategy form will capture general situations where information revelation is pivotal.

One might object that the foregoing examples, despite their simplicity, are unrepresentative of realistic environments precisely because of the way that price patterns reveal sharp distinctions among valuation levels. For example, the distributions of deadlines and their values assumed in the experiments of Section 6 do not have this property. One potentially fruitful avenue for future work would be to characterize problem classes based on "diffuse" distributions where information revelation is not a driving factor. It may well be that variants of SB are reasonable responses to SB in some natural scheduling environments.

10   Conclusion

The foregoing study illustrates the difficulty of drawing conclusions about strategy choices in even a relatively simple simultaneous ascending auction game. Straightforward bidding is not even approximately optimal, nor are straightforward variants of same, even in response to other straightforward bidders. The space of strategies is too large for analytic methods to be directly fruitful, and exhaustive simulation studies are out of the question. Analysis and simulation *can* produce results for restricted versions of the game, however restrictions on the strategy choices available are inherently somewhat arbitrary, and results remain sensitive to particular distributions of agent preferences.

Some may interpret this pessimistic assessment as further argument that a simultaneous auction design in the presence of complementarities is untenable, and pro-

pose that combinatorial or other mechanisms of broader scope be imposed in their stead. We agree that more coordinated market designs have advantages, and advocate use of combinatorial auctions or even direct mechanisms where possible. Nevertheless, we observe that separation of related markets is a prevalent situation today, and expect that it will always be so, since coordinating the allocation of all significantly related resources in the world through a single mechanism is simply infeasible. Thus, despite its difficulty, strategic analysis of games such as that studied here is a necessary task for those interested in market-based resource allocation.

Although we are still far from achieving a comprehensive understanding of the market scheduling game, we are hopeful that the techniques developed here will prove useful tools. Specifically, through parameterization of a strategy space, sampling-based simulation, and evolutionary search, it is possible to explore systematically the strategic issues salient to the simultaneous auction setup, as well as a variety of similar market games.

11 Acknowledgments

References

[1] Dimitri P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1:7–66, 1992.

[2] Sushil Bikhchandani and John W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74:385–413, 1997.

[3] Paul J. Brewer and Charles R. Plott. A binary conflict ascending price (BICAP) mechanism for the decentralized allocation of right to use railroad tracks. *International Journal of Industrial Organization*, 14:857–886, 1996.

[4] Donald E. Brown and William T. Scherer. *Intelligent Scheduling Systems*. Kluwer, 1995.

[5] Dave Cliff. Evolving parameter sets for adaptive trading agents in continuous double-auction markets. In *Agents-98 Workshop on Artificial Societies and Computational Markets*, pages 38–47, Minneapolis, MN, May 1998.

[6] Dave Cliff. Explorations in evolutionary design of online auction market mechanisms. *Electronic Commerce Research and Applications*, 2:162–175, 2003.

[7] Sven de Vries and Rakesh Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15, 2003.

[8] Daniel Friedman. Evolutionary games in economics. *Econometrica*, 59:637–666, 1991.

[9] Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:95–124, 1999.

[10] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.

[11] Alexander S. Kelso and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50:1483–1504, 1982.

[12] Jeffrey O. Kephart, James E. Hanson, and Jakka Sairamesh. Price and niche wars in a free-market economy of software agents. *Artificial Life*, 4:1–23, 1998.

[13] Paul D. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13:227–286, 1999.

[14] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, New York, 1995.

[15] R. Preston McAfee and John McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10(1):159–175, 1996.

[16] Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In *Handbook of Computational Economics*, volume 1. Elsevier, 1996.

[17] Richard D. McKelvey, Andrew McLennan, and Theodore Turocy. Gambit game theory analysis software and tools, 1992. `http://econweb.tamu.edu/gambit`.

[18] Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108:245–272, 2000.

[19] John Nash. *Non-cooperative games*. PhD thesis, Princeton University, Department of Mathematics, 1950.

[20] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[21] James Nicolaisen, Valentin Petrov, and Leigh Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5:504–523, 2001.

[22] Michael Peters and Sergei Severinov. Internet auctions with many traders. Technical report, University of Toronto, 2001.

[23] Steve Phelps, Simon Parsons, Peter McBurney, and Elizabeth Sklar. Co-evolution of auction mechanisms and trading strategies: Towards a novel approach to microeconomic design. In *GECCO-02 Workshop on Evolutionary Computation in Multi-Agent Systems*, pages 65–72, 2002.

[24] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[25] Tony Curzon Price. Using co-evolutionary programming to simulate strategic behaviour in markets. *Journal of Evolutionary Economics*, 7:219–254, 1997.

[26] S. J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.

[27] P. Schuster and K. Sigmund. Replicator dynamics. *Journal of Theoretical Biology*, 100:533–538, 1983.

[28] I. E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11:449–451, 1968.

[29] P. Taylor and L. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156, 1978.

[30] William E. Walsh, Rajarshi Das, Gerald Tesauro, and Jeffrey O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, Edmonton, 2002.

[31] Jörgen W. Weibull. *Evolutionary Game Theory*. MIT Press, 1995.

[32] Michael P. Wellman, William E. Walsh, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35:271–303, 2001.

[33] K. P. White. Advances in the theory and practice of production scheduling. In C. T. Leondes, editor, *Advances in Control and Dynamic Systems*, pages 115–157. Academic Press, 1990.