In *Proceedings of the Eighteenth International Conference on Distributed Computing Systems (ICDCS-98),* Amsterdam, The Netherlands, May 1998

Some Economics of Market-Based Distributed Scheduling

William E. Walsh Michael P. Wellman Peter R. Wurman Jeffrey K. MacKie-Mason

University of Michigan, Ann Arbor, MI 48109-2110 USA
{ wew, wellman, pwurman, jmm }@umich.edu

Abstract

Market mechanisms solve distributed scheduling problems by allocating the scheduled resources according to market prices. We model distributed scheduling as a discrete resource allocation problem, and demonstrate the applicability of economic analysis to this framework. Drawing on results from the literature, we discuss the existence of equilibrium prices for some general classes of scheduling problems, and the quality of equilibrium solutions. We then present two auction protocols for implementing solutions, and analyze their computational and economic properties.

1 Introduction

Solving scheduling problems with and for distributed computing systems presents particular challenges attributable to the distributed nature of the computation. Consider, for instance, the problem of scheduling network access to programs representing various users on the Internet. In such an environment, system modules (user programs) represent independent entities (users) with conflicting and competing scheduling requirements, and may possess localized information relevant to their tasks. To recognize this independence, we treat the modules as *agents*, ascribing each of them autonomy to decide how to deploy resources under their control in service of their interests.

Within this model, a distributed scheduling method can be analyzed according to how well it exhibits the following properties:

- Self-interested agents can make effective decisions with local (private) information—without knowing the private information and strategies of other agents.
- The method requires minimal communication overhead.
- Solutions do not waste resources. If there is some way to make some agent(s) better off without harming others, it should be done. A solution that cannot be improved in this way is called *Pareto optimal*.

In some settings, it might be appropriate to adopt some stronger optimality criteria, based on a judgment about social value of the various agents.

Straightforward distributed scheduling policies—such as first-come first-served, shortest-job-first, priority-first, and combinations thereof—do not generally possess these properties. For example, queue-position schemes are insensitive to relative value based on the *substance* of the task being performed. On the other hand, priority-based schemes beg the question of how to set priorities so that desirable results follow. If self-interested agents are free to set their own priorities, then without some incentive to the contrary, they will specify maximum priority for whatever they are interested in.

Citing such limitations, several have proposed that distributed resource allocation problems be solved via market mechanisms [4], an approach we have called *market-oriented programming* (MOP) [24]. In MOP, we define agent activities in terms of resources required and produced, reducing an agent's decision problem to evaluating the tradeoffs of acquiring different resources. These tradeoffs are represented in terms of market prices, which define a common scale of value across the various resources. The problem for designers of computational markets is to specify the configuration of resources traded (formally designated *goods* in the market), and the mechanism by which agent interactions determine prices.

Assuming that a scheduling problem must be decentralized, markets can provide several advantages:

- Markets are naturally distributed. Agents make their own decisions about how to bid based on the prices and their own relative valuations of the goods.
- Communication is limited to the exchange of *bids* and *prices* between agents and the market mech-

- anism. In particular settings, it can be shown that price systems minimize the dimensionality of messages required to determine Pareto optimal allocations [8].

- Since agents must back their representations with exchange offers, some mechanisms can elicit the information necessary to achieve Pareto and system optima (or come within some tolerance of optimal) in some well-characterized situations.

Of course, all of these benefits do not automatically accrue as a result of setting up a market-like environment. Prior work applying market-inspired mechanisms to scheduling [2, 18, 21, 22] and other distributed computing problems [4] has produced promising empirical results. Understanding the scope of these methods, and developing a general design methodology for computational markets, however, requires an analytical characterization of their properties. In our own MOP work, we have adopted the framework of general equilibrium theory [10], and have found that our computational markets behave predictably when conditions of the theory are met [14, 24]. We have also applied the approach to discrete optimization problems—where the conditions guaranteeing desirable outcomes are not satisfied—and have found (not surprisingly) that the methods sometimes work, and other times break down [25].

Since scheduling problems very often involve discrete (indivisible) resource units, we have undertaken to analyze directly the behavior of computational market mechanisms for such problems. We start by defining a general class of discrete allocation problems, and characterizing some distinctions particularly meaningful in the scheduling domain. We show how some recent results in economic theory apply to the scheduling problem, and report our own extensions and analysis.

In the next section, we motivate the work with a concrete example of a simple factory scheduling problem. In Section 3, we provide a formal economic model of a general version of the problem, and in Section 4 we relate some equilibrium and optimality properties associated with the problem. In Section 5, we briefly describe a general framework for auction protocols, and describe and analyze two protocols in Sections 6 and 7. Finally, we consider future work in Section 8.

2 A Factory Scheduling Economy

Consider a factory with an unscheduled day shift. There are eight one-hour time slots, labeled 9:00 to 16:00 according to their respective end times. Slots can be allocated for the production of customer orders. The factory has a *reserve price* for each time slot, representing the minimum price that the factory is willing to accept in exchange for that time slot.

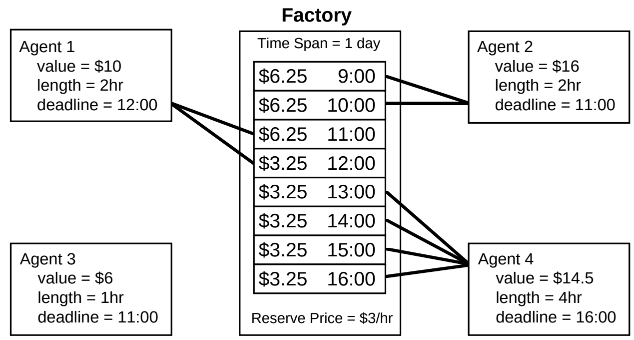

Figure 1: A factory scheduling economy. Lines connecting the agents to time slots represent one feasible allocation.

Assume each customer agent has one job it wants completed. An agent's job is defined by its duration (length), its deadline, and the value (expressed in dollars) the agent places on the job. An agent is willing to spend up to this value to complete its job. To do so, the agent must acquire a number of slots no less than the length (not necessarily contiguous), no later than the deadline. The agent gets no value if its job cannot be completed before its deadline. The value of a solution is the sum of values of the agents holding the goods, which is the sum of the reserve price for each time slot that was not sold, plus the value associated with each customer agent that meets its job deadline.

Example 1 *The agents are shown in Figure 1.*[1] *Since the sum of lengths exceeds available factory time, it is not possible for all of the agents to produce their orders. The allocation depicted in Figure 1 represents a system optimum.*

Given an assignment of prices to goods, we can define an agent's optimal choice as a set of slots that complete the job at the minimum cost, or the empty set if the the job could not not be completed for less than its value. The reader can verify that at the prices shown in Figure 1, each agent makes a locally optimal choice in the globally optimal allocation.

3 Formal Model of the Scheduling Economy

We define a general discrete resource allocation problem in terms of the following elements:

[1] An interactive online demonstration of the ascending auction (Section 6) applied to this example can be found at `http://auction.eecs.umich.edu/demos/factory.html`.

- G, a set of n discrete goods,
- A, a set of m agents, and $\perp$ representing the seller or null agent,
- prices $p = \langle p_1, \ldots, p_n \rangle$.

We assume that agents have quasilinear utility functions, meaning that their valuations can be measured in terms of a common numeraire, which for convenience can be taken to be "money". Therefore, we can directly compare the utility of different agents, and meaningfully treat the sum as a measure of global value. Agent j gets utility $v_j(X) + M_j$ for holding the set of goods X, $X \subseteq G$, and M_j units of money.

Let $H_j(p)$ denote the maximum surplus value achievable by agent j at prices p. That is,

$$H_j(p) \equiv \max_{X \subseteq G} \left[v_j(X) - \sum_{i \in X} p_i \right].$$

Note that, for some prices, an agent may maximize its surplus with the empty set.

A *solution* is a mapping $f : G \rightarrow A \cup \{\perp\}$, indicating which agent, if any, gets each good. Let $F_j \equiv \{i | f(i) = j\}$ denote the set of goods allocated to agent j and $F_\perp \equiv \{i | f(i) = \perp\}$ denote the set of unallocated goods in f.

The seller of good i has utility equal to its *reserve value* q_i if the good is unallocated, or the money it receives for the good if it is allocated. Intuitively, the reserve value denotes the value to the owner, or the "system", of not allocating the good to any agent. Different time slots could potentially have different reserve values; for instance, a factory may have a higher reserve price for evening hours to cover overtime expenses.

The *global* or *system value of a solution*, $v(f)$, is the sum of the agent values achieved and the reserve value of goods not used by agents,[2]

$$v(f) \equiv \sum_{i \in F_\perp} q_i + \sum_{j=1}^{m} v_j(F_j).$$

We measure the system value of a solution *ex post*, that is, conditional on knowing all agents' valuations.[3] A solution is *optimal* if no other solution has higher value.

[2] Because all agents have utility that is linear in money, the total value obtained from money is constant and hence can be ignored.

[3] It is sometimes useful to measure the value of a solution *ex ante*, that is, with respect to the expectation of agent valuations.

In Sections 6 and 7 we present auction protocols for this very general resource allocation problem. However, the theoretical results and examples we present focus on particular subclasses of scheduling problems where each agent has one job to complete. For these problems, we associate each agent j with a job length λ_j and $K_j \geq 1$ deadlines $d_j^1, \ldots, d_j^{K_j}$. The value $v_j(X)$ of a set of goods X is determined by the earliest deadline d_j^k such that X includes at least λ_j time slots no later than d_j^k. For convenience we represent the time slots as integers, starting from one.

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 ($K_j = 1$), we call the problem *fixed deadline*. If $K_j > 1$ for some j (i.e., j accrues greater value for finishing the job sooner), then we call the problem *variable deadline*.

4 Equilibrium and Optimality in the Scheduling Economy

Definition 1 *A solution f is in* equilibrium *at prices p iff*

1. *For all agents j, $v_j(F_j) - \sum_{i \in F_j} p_i = H_j(p)$.*
2. *For all $i \in F_\perp$, $p_i = q_i$.*
3. *For all $i \notin F_\perp$, $p_i \geq q_i$.*

Intuitively, this definition states that, in equilibrium, each agent gets an allocation that maximizes its utility given the current prices. Equilibria sometime exist, and are generally not unique. Consider Example 1. The solution shown, with only agent 3 receiving no goods, is in equilibrium at the set of prices suggested, with slots 9:00, 10:00, and 11:00 each having a price of \$6.25, and all other slots having a price of \$3.25. The same solution is also in equilibrium with respective prices of \$6.50 and \$3.35, and many other combinations. The equilibrium solution has value \$40.50, which is optimal. Indeed it had to be, as demonstrated by the following result.

Theorem 1 *For the general discrete resource allocation problem, if there exists a p such that f is in equilibrium at p, then f is an optimal solution.*

Proof. Bikhchandani and Mamer provide a proof for an exchange economy without reserve prices [3]. The extension to account for reserve values is straightforward [23]. □

This result confirms the usual consequence of competitive equilibrium: that no further gains from trade

Name	Job Length	Deadline	Value
Agent 1	2	2	$3
Agent 2	1	2	$2

Table 1: A problem with no equilibrium. Adapted from a demonstration [12] that price equilibria may not exist in the FCC market for radio spectrum.

are possible and so the result is Pareto optimal. Since we assume that agent values are expressible in price units, Pareto optimality corresponds to global optimality.

Example 2 *There are two agents as described in Table 1, and the reserve price of each good is zero.*

The optimal solution, $f(1) = f(2) = 1$, *is not in equilibrium at any prices, and indeed no equilibrium exists in this case. If* p *were in equilibrium, then* $p_1 \geq \$2$ *and* $p_2 \geq \$2$, *otherwise agent 2 would demand one of the goods. But if these inequalities hold then agent 1 would not demand the two time slots it requires.*

In this example, the nonexistence of equilibrium prices is due to *complementarities* in agent preferences. Agent 1 considers the two time slots complementary in that it values one iff it has the other. Complementarities cannot arise in the single-unit scheduling problem.

Lemma 2 *The single-unit scheduling problem always has a unique minimum equilibrium price vector.*

Proof. An exchange economy characterized by quasi-linear utilities for single goods always has a unique minimum equilibrium price vector [17]. The single-unit scheduling problem is a subclass of this type of economy. □

Theorem 3 *Any optimal solution to the single-unit scheduling problem (fixed or variable deadline) is supported by a price equilibrium.*

Proof. By Lemma 2, the single-unit scheduling problem always has at least one price equilibrium p. By Theorem 1, p supports an optimal solution. Since p supports an optimal solution, it can be shown that all optimal solutions must be supported by p [3, 7]. □

Together, Theorems 1 and 3 establish that a solution to the single-unit scheduling problem is optimal iff it is supported by a price equilibrium. Example 2 demonstrates that relaxing the single-unit restriction immediately leads to the possibility that an equilibrium will not exist.

In addition to the single-unit restriction of Theorem 3, we can identify a few other conditions that guarantee the existence of equilibrium. If all agents have additive preferences over goods then an equilibrium exists.[4] Additive preferences is a sufficient condition for *gross substitutability* — if the price for one good goes up, demand does not go down for any other good — which in turn guarantees the existence of equilibrium [9]. Bikhchandani and Mamer [3] present some other technical conditions for existence of equilibrium, which do not seem to be immediately expressible in scheduling terms.

5 Auction Protocols

We use the term *protocol* to refer to a *mechanism*, along with agent *bidding policies*. The mechanisms we consider are generically called *auctions*. McAfee and McMillan provide the following definition [11]:

> An auction is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants.

This definition includes the well known English open-outcry and first-price sealed bid auctions—commonly used to sell art and to award procurement contracts, respectively—as well as a broad range of other mechanisms, including fixed pricing, Dutch auction, Vickrey auction, commodities markets, and the ascending and Generalized Vickrey auctions described in Sections 6 and 7, respectively.

In order to place greater structure on the space of mechanisms, and also to provide a common interface to agents, we define a somewhat restricted, but still very general auction protocol.

1. Agents send bids to the mechanism to indicate their willingness to exchange goods.

2. The auction may post *price quotes* to provide summarized information about the system-wide value of goods.

   Steps 1 and 2 may be iterated, particularly if price quotes are posted.

3. The auction determines an allocation and notifies the agents as to who purchases what from whom at what price.

[4] Note that preferences are not additive in the multiple-unit scheduling problem. However, equilibrium would exist if agents had additive preferences for completing multiple single-unit jobs.

The above sequence may be performed once or iterated any number of times.

Auctions can be differentiated by parameterized values including, but not limited to: matching algorithm, price determination algorithm, event timing, bid restrictions, and intermediate price revelation [26]. We have implemented the Michigan Internet AuctionBot [1, 26], an online parameterized auction server that operates as a research and development platform for creating and experimenting with auction-based economies. The AuctionBot provides interfaces for human and software agents to easily create and bid in auctions. Currently the AuctionBot supports several classical auction types [26], including the mechanism for the ascending auction protocol described in Section 6.

In order to predict auction outcomes, we must consider the agents' presumed bidding policies, which in turn are based on our model of their beliefs and preferences. With some auctions we are able to determine analytically that a particular bidding policy is part of a Bayesian-Nash equilibrium or even the dominant strategy. With other auctions we rely on experimentation and rules of thumb based on economic principles to determine reasonable bidding policies.

Our analysis is from the standard noncooperative perspective, which assumes that agents do not directly coordinate their bidding. *Collusion* has been an issue in the FCC spectrum auctions; anti-collusion measures are considered by Milgrom [13].

The auction mechanisms we discuss are distributed in the sense that each agent calculates its own bid strategy, based on local information. The ascending auction can be further distributed in that allocations for each good can be computed separately.

6 Ascending Auction

We define the ascending auction protocol for the general discrete resource allocation problem. Separate auctions determine prices for each of the goods. Agents submit successively higher bids to the auctions, and auctions immediately report price quotes to all interested agents upon receiving a bid. When the bidding stops, each auction allocates its respective good to the highest bidder at the price the agent bid, or the good is retained by the seller if there are no bids.

6.1 Bidding Rules

At any point in time, the *bid price* in the auction for good i, denoted β_i, is the highest bid in the auction thus far. If auction i has received no bids, β_i is undefined. Auction i's *ask price*, denoted α_i, is $\beta_i + \epsilon$, for some fixed ϵ, if β_i is defined. Otherwise, the ask price is q_i.

The ascending auction rejects any bid less than its ask price. Agents are not allowed to withdraw bids. An agent may replace its bid with another, but the new bid must be at least the current ask price. These rules guarantee that prices do not decrease and that the bidding process terminates.

6.2 Agent Bidding Policies

When an agent j enters the market, it bids the ask prices for the set of goods, X, that maximizes its surplus H_j, based on the current ask prices (breaking ties arbitrarily). As other agents continue to bid, agent j may lose some of its bids. When this occurs, j bids the ask price on the set of goods that maximizes its surplus, assuming that it must pay for the goods it is currently winning. For the single-unit scheduling problem, whenever an agent is not already winning a bid, it simply bids the ask price for the single good that maximizes its surplus at the ask prices. If no good would provide it with a positive surplus, then the agent "drops out" of the auction.

This 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. This is called the *no regret* property [3], and means that from the agent's perspective, no bidding policy would have been a better response to the other agents' bids. The no-regret property does *not* hold, however, for the ascending auction in the multi-unit scheduling problem, regardless of the bidding strategy [3]. In general, an agent might perform better, for example, through accurate prediction of the other agents' behavior. In the absence of a basis for prediction, however, the simple strategy proposed may indeed be reasonable.

6.3 Analysis of the Ascending Auction

Let p_i denote the price paid for i. Under the ascending auction rules, when the auction closes, $p_i = \beta_i$ if defined, otherwise $p_i = q_i$.

It is possible that the ascending auction can determine prices that differ from an equilibrium of a multiple-unit scheduling economy by arbitrarily large amounts.

Example 3 *The bid increment is $\epsilon = \$1$ and the reserve prices are zero. The agents are described in Table 2.*

Although there are many equilibrium price sets (one of which is $p_1 = \$8$, $p_2 = \$8$, and $p_3 = \$1$), the ascending auction may not find an equilibrium. Agent 2 could bid up good 3 until $\alpha_3 > \$2$ while it and agent 1 both bid up the prices on 1 and 2. The reader can

Name	Job Length	Deadline	Value
Agent 1	2	2	\$20
Agent 2	2	3	\$8
Agent 3	1	3	\$2

Table 2: A multiple-unit problem (Example 3).

Name	Job Length	Deadline	Value
Agent 1	1	1	\$3
Agent 2	2	2	\$11

Table 3: A multiple-unit problem (Example 4).

verify that any equilibrium must have agent 3 winning good 3 at a price no greater than \$2.

In the multiple-unit scheduling problem, the ascending auction can produce allocations that are arbitrarily far from optimal.

Example 4 *There are two agents as shown in Table 3. Reserve prices are* $q_1 = \$1$ *and* $q_2 = \$9$*, and the bid increment is* $\epsilon = \$1$.

If agent 2 places its bids first, it will bid \$1 for 1 and \$9 for 2. Agent 1 will then bid \$2 for 1. The bidding will stop with good 1 allocated to agent 1 and good 2 allocated to agent 2. This solution has a value of \$3 yet the optimal solution, with 2 unallocated, has a value of \$12. It is easy to see—by increasing q_2 *and* v_2 *by the same amount—that the ascending auction can produce a solution that is arbitrarily far from optimal.*

If we restrict each agent's requirement to a single time slice, then by Theorem 3 an equilibrium exists. However, the ascending auction protocol is not guaranteed to reach an equilibrium even with this restriction. Consider the following economy.

Example 5 *The bid increment is* $\epsilon = \$1$*. The reserve prices are* $q_1 = \$4$*,* $q_2 = \$3$*, and* $q_3 = \$3$*. The agents are described in Table 4.*

It is possible that agent 2 may bid first, for 2. Then $\alpha_2 = \$4$*. Agent 1 will then bid \$4 for either 1 or 2. If it bids for 1 then the bidding will stop and agent 1 will win 1 for \$4 and agent 2 will win 2 for \$3. But since* $p_2 = \$3 < p_1$*, agent 1 would maximize its surplus by demanding 2 at the final prices. However, the bidding rules prohibit any readjustment towards an equilibrium. The auction does not allow agent 1 to withdraw its bid for 1, and hence the final allocation violates condition 1 of the definition of equilibrium.*

Name	Job Length	Deadline	Value
Agent 1	1	2	\$6
Agent 2	1	3	\$7

Table 4: A single-unit problem (Example 5).

It is not hard to see that the potential failure to reach equilibrium can be demonstrated for any positive value of ϵ, no matter how small. Nevertheless, unlike the multiple-unit problem, we can bound the distance from the equilibrium price vector by $\kappa\epsilon$, where $\kappa = \min(n, m)$.

Theorem 4 *For the variable-deadline, single-unit scheduling problem, the final price of any good determined by ascending auction protocol will differ from the unique minimum equilibrium prices by at most* $\kappa\epsilon$.

Proof. Demange et al. prove this result for the ascending auction protocol in an exchange economy where buyers want no more than a single item from a set of available goods [5]. In the single-unit scheduling problem, no agent wishes to obtain more than a single item, hence the result holds for this problem. □

Consider again Example 5. The solution shown has a value of \$16. If agent 1 had received good 2 and agent 2 had received good 3 then the value of the solution would be \$17, which is optimal. However, the solution can be suboptimal by only a bounded amount.

Theorem 5 *The ascending auction protocol with a given* ϵ *produces a solution to the variable-deadline, single-unit scheduling problem that is suboptimal by at most* $\kappa\epsilon(1 + \kappa)$.

Proof. Let f be the allocation reached by the ascending auction and f^* an optimal allocation. p_i is the price found for i in the ascending auction, and p_i^* the unique minimum equilibrium price for i (recall that Lemma 2 and Theorem 3 established that a unique minimum price vector exists and supports f^*). Let $e_i = p_i^* - p_i$. From Theorem 4 we know that $\mid e_i \mid \leq \kappa\epsilon$.

Let F and F^* be the set of goods allocated in f and f^*, respectively. To get the error, we can subtract the value of the final allocation from the optimal allocation.

$$v(f^*) - v(f)$$

$$= \left(\sum_{i \in F_\perp^*} q_i + \sum_{j=1}^{m} v_j(F_j^*) \right) - \left(\sum_{i \in F_\perp} q_i + \sum_{j=1}^{m} v_j(F_j) \right)$$

$$= \sum_{i \in F_\perp^* \setminus F_\perp} q_i - \sum_{i \in F_\perp \setminus F_\perp^*} q_i + \sum_{j=1}^{m} v_j(F_j^*) - \sum_{j=1}^{m} v_j(F_j). \quad (1)$$

In the single-unit problem, an agent bids for the good that maximizes its surplus. In the solution allocation, this surplus must be at least the surplus it would get from any other good at the ask price, otherwise the agent would have bid for that good instead. Therefore, when the ascending auction stops,

$$\sum_{j=1}^{m} v_j(F_j) - \sum_{i \in F} p_i \geq \sum_{j=1}^{m} v_j(F_j^*) - \sum_{i \in F^*} \alpha_i \geq \sum_{j=1}^{m} v_j(F_j^*) - \sum_{i \in F^*} (p_i + \epsilon).$$

Rearranging, and using the facts that $F \setminus F^* = F_\perp^* \setminus F_\perp$ and $F^* \setminus F = F_\perp \setminus F_\perp^*$, we have

$$\sum_{j=1}^{m} v_j(F_j^*) - \sum_{j=1}^{m} v_j(F_j) \leq \sum_{i \in F^*} (p_i + \epsilon) - \sum_{i \in F} p_i$$

$$= \sum_{i \in F^* \setminus F} p_i - \sum_{i \in F \setminus F^*} p_i + \sum_{i \in F^*} \epsilon$$

$$= \sum_{i \in F_\perp \setminus F_\perp^*} p_i - \sum_{i \in F_\perp^* \setminus F_\perp} p_i + \sum_{i \in F^*} \epsilon. \quad (2)$$

Goods that were unallocated in f must have prices equal to their reserve prices,

$$\sum_{i \in F_\perp \setminus F_\perp^*} p_i - \sum_{i \in F_\perp \setminus F_\perp^*} q_i = 0. \quad (3)$$

Goods that were unallocated in f^* must have minimum equilibrium prices equal to their reserve prices,

$$\sum_{i \in F_\perp^* \setminus F_\perp} q_i = \sum_{i \in F_\perp^* \setminus F_\perp} p_i^* = \sum_{i \in F_\perp^* \setminus F_\perp} (p_i + e_i).$$

Rearranging, we have

$$\sum_{i \in F_\perp^* \setminus F_\perp} q_i - \sum_{i \in F_\perp^* \setminus F_\perp} p_i = \sum_{i \in F_\perp^* \setminus F_\perp} e_i. \quad (4)$$

Substituting (2), (3), and (4) into (1) gives

$$v(f^*) - v(f) \leq \sum_{i \in F_\perp^* \setminus F_\perp} q_i - \sum_{i \in F_\perp \setminus F_\perp^*} q_i + \sum_{i \in F_\perp \setminus F_\perp^*} p_i - \sum_{i \in F_\perp^* \setminus F_\perp} p_i + \sum_{i \in F^*} \epsilon$$

$$= \sum_{i \in F_\perp^* \setminus F_\perp} e_i + \sum_{i \in F^*} \epsilon.$$

The total error is maximized when $e_i = \kappa\epsilon$ for all $i \in F_\perp^* \setminus F_\perp$. Since there can be at most κ goods in $F_\perp^* \setminus F_\perp$ and F^*, this gives an upper bound on the total error: $v(f^*) - v(f) = \kappa\epsilon(1 + \kappa)$.
□

The computation of the clearing and price quotes is trivial in the ascending auction. Communications costs dominate the run time, which can therefore be measured in terms of the bids required. Because bids increase by a fixed increment, the number of iterations is inversely proportional to ϵ. Hence, in choosing the value for ϵ, we trade off solution value for communication efficiency.

We have shown that the simple bidding policy is reasonable for individual agents, and produces allocations with desirable system properties in the single-unit problem. The results do not provide strong support for this simple policy in the multiple-unit problem. Other strategies, such as jump bidding—where an agent bids in large increments for sets of goods to signal its willingness to aggressively pursue that set—may provide potential advantages to individuals or the system. However, it is an open question as to whether there exists a policy for the ascending auction (or any complete protocol) that always finds (within some tolerance) an equilibrium when it exists.

7 Generalized Vickrey Auction

The ascending auction performs well for single-unit allocation problems. At the end of Section 3 we note that the single-unit restriction is only one sufficient condition for existence of a price equilibrium. However, even when equilibria exist for a multiple-unit problem, the ascending auction may not find one, as shown by Example 4. Further, as Example 2 demonstrates, many scheduling problems have optimal, value-maximizing allocations that are not supportable by any price equilibrium.

The Generalized Vickrey Auction (GVA) [19] can implement optimal allocations for a broad class of scheduling problems with multiple goods, multiple units, requirements contingencies, and externalities (i.e., values for one agent that depend on the allocations obtained by other agents). The GVA does not use a price system. Rather, it computes payments for agents' allocations that only sometimes, but not always, translate into meaningful prices for individual goods. Thus the GVA can obtain optimality in problems for which a price equilibrium does not exist.

7.1 Bidding Rules for the GVA

The GVA is a direct revelation mechanism. Agents report utility functions in their bids, rather than price-quantity points. Recall that v_j is agent j's actual utility function. Each agent announces $\hat{v}_j$, its alleged utility function. The circumflexes are used to indicate that the agent is not constrained to be truthful, that is, it may be that $\hat{v}_j \neq v_j$. The auction knows the reserve values, q_i. After receiving the bids, the GVA returns an allocation, and a vector of positive or negative payments to be made to the agents.

7.2 Allocation Rules for the GVA

Recall that a solution is a mapping f, and the value of a solution is given by $v(f)$. The auction mechanism:

1. Computes a solution,

$$f^* = \arg\max_f \sum_{i \in F_\perp} q_i + \sum_{j=1}^{m} \hat{v}_j(F_j). \tag{5}$$

2. Computes payments to agents,

$$V_j \equiv W_{-j}(f^*) - P_j(\hat{v}_{-j}),$$

where

$$\begin{aligned} W_{-j}(f^*) &= \sum_{i \in F_\perp^*} q_i + \sum_{s \neq j} \hat{v}_s(F_s^*), \\ P_j(\hat{v}_{-j}) &= \max_f \sum_{i \in F_\perp} q_i + \sum_{s \neq j} \hat{v}_s(F_s). \end{aligned} \tag{6}$$

The W_{-j} component represents the total reported value for agents other than j at the solution f^*. The residual payment P_j could be any function of other agents' reported valuations. However, we restrict attention here to the function described in (6).

7.3 Bidding Policy for the GVA

An auction is *incentive compatible* if truthful revelation of utility functions is the dominant bidding policy.

Theorem 6 *If $v_j(F_j^*) + V_j \geq 0$ and if $P_j(\hat{v}_{-j})$ is independent of agent j's reported preferences, then the GVA is incentive compatible.*

The intuition behind the proof [19] generalizes that of Vickrey's original result [20]. An agent receives $v_j(F_j^*) + V_j = v_j(F_j^*) + W_{-j} - P_j$, from the value of its allocation and the payment from the auction. The auction mechanism chooses the solution f^* to maximize $\hat{v}_j(F_j^*) + W_{-j}(f^*)$. Therefore, if the agent bids truthfully ($\hat{v}_j = v_j$), it receives the auction mechanism's maximand less a constant (recall P_j is unaffected by agent j's bid). Clearly the agent can do no better than to get the auction to maximize its true value. Thus, for a rational agent, truthful bidding dominates all other strategies.

7.4 Optimality Analysis of the GVA

If all agents behave rationally, then since the GVA is incentive compatible, they bid truthfully. The GVA computes the optimal allocation based on the bids, and since all bids are truthful, the allocation is system optimal.

The GVA solves problems with multiple units, and problems without a price equilibrium. Example 2 above has both features:

Example 6 (From Example 2) *If the agents truthfully report their value functions, the auction mechanism finds the optimal solution f^*: $f^*(1) = f^*(2) = 1$. It then calculates $W_{-1} = 0$ and $W_{-2} = 3$. Agent 1 receives total value $3 + [0 - P_1]$, and agent 2 receives $0 + [3 - P_2]$. No untruthful bid can increase these payoffs, so the agents should bid truthfully. The condition that $v_j(F_j) + V_j \geq 0$ requires that $P_j \leq 3$ for $j \in \{1, 2\}$; otherwise, rational agents would choose not to participate in the auction. Thus, $P_1 = 2$ (agent 1 pays \$2), $P_2 = 3$ (agent 2 pays \$0), and the mechanism has a net revenue of \$2.*

7.5 Limitations on the GVA

A mechanism is *individually rational* if no agent can be worse off from participating in the auction than if it had declined to participate. A mechanism is *budget balanced* if the net payment over all agents is nonnegative. Generally, these, along with optimality, are the properties we desire when agents play their equilibrium strategies in a mechanism. In our scheduling problem we can obtain all three using P_j from (6). The payment function $V_j(P_j)$ transfers to agent j the net value increment to all other agents that results from j's participation in the auction. Agent j's only effect on others is that it may get time slices that others desire, so its participation always makes other agents weakly worse off. Thus, V_j is nonpositive for all j, and the auction mechanism runs a surplus.

Theorem 7 *If the GVA uses the payment function $W_{-j} - P_j$ then the individual rationality constraint is satisfied and the net monetary payments to the auction mechanism are nonnegative.*

However, the problem statement assumes that the auction mechanism knows the reserve values q_i. If instead the q_i are the private information of seller agents, then no mechanism can obtain more than two out of the three desired properties. Myerson and Satterthwaite [15] proved this impossibility theorem for bilateral exchange problems, some of which are scheduling problems with seller agents.

Example 7 (Bilateral exchange) *Suppose there is one buyer, who has a single-unit job with deadline 1 and value v. Let the seller be an agent, with reserve value q_1. Suppose $v > q_1$. The GVA would induce truthful reporting of v and q_1, give the good to the buyer, require the buyer to pay q_1, and pay v to the seller. Although the mechanism is individually rational and would produce the optimal allocation, the auction would run a deficit of $v - q_1$.*

We can always use the GVA to obtain individual rationality and optimality, but with an auction deficit, by setting, for example, $P_j = 0$. Alternatively, the GVA can obtain optimality and budget balance by setting a sufficiently high P_j, which, however, makes it irrational for some agents to participate.

7.6 GVA Computation

As a baseline for computational efficiency, we note that Neapolitan and Naimipour [16] show that a simple centralized greedy algorithm solves the single-unit, fixed-deadline scheduling problem optimally, in time $\Theta(m \lg m)$. The GVA mechanism must solve multiple optimization problems to process the bids, one to determine the optimal allocation, and one for each agent j with its bid removed to determine P_j. For a single-unit, fixed-deadline problem we can use the centralized algorithm for each optimization, with a total runtime of $\Theta(m^2 \lg m)$. Thus, inducing preference revelation (and thereby obtaining full optimality) raises the auction cost by a factor of m; this is the computational cost of distributing the problem via the straightforward implementation of the GVA.

If we remove the single-unit restriction, then any centralized algorithm that can solve the scheduling problem optimally can solve the Integer Knapsack problem. Hence the multiple-unit scheduling problem is NP-Complete.[5] By the preceding argument, distributing the multiple-unit problem via the GVA contributes a factor of m to the computation.

8 Discussion

We have presented two auction mechanisms that can compute optimal or near-optimal solutions to the single-unit distributed scheduling problem in a computationally efficient manner. The multiple-unit problem is significantly more difficult and entails a sharper tradeoff among solution quality, computational efficiency, and the degree to which the mechanism is distributed. The computation performed by the ascending auction is trivial, and can be distributed by goods. However, we cannot guarantee the quality of solutions produced by this mechanism for the multiple-unit problem. The GVA, by solving multiple combinatorial problems, finds the optimal solution for this problem, even when a price equilibrium does not exist.

We view this work as a first important step in developing a broad framework for using markets to solve distributed scheduling problems. In order to move forward we must identify broader classes of scheduling problems and develop associated mechanisms such that we can effectively predict and analyze the behavior of the economy. We do not expect to find a single mechanism that reaches equilibrium in all situations where such equilibria exist. However we wish to develop a suite of mechanisms that collectively cover a broad range of problems. That is, we want to be able to choose a mechanism for a given problem and know that it will reach or come close to equilibrium when it exists, or else perform acceptably in some other respect when equilibrium does not exist. In addition to the auctions described in this paper, we are also exploring more complex mechanisms with multiple stages, and activity rules [6, 12, 13].

We are exploring the theoretical aspects of market mechanisms to support our experimental work in more complex, real-time network scheduling domains. These domains require more elaborate models, including multiple-stage scheduling which is necessary when, for instance, data must pass through several different network nodes. We are in the process of joining our top-down economic approach with a bottom-up analysis of network scheduling requirements.

Acknowledgments

Thanks to Fredrik Ygge, David Pennock, Terence Kelly, and the anonymous reviewers for constructive comments. This work was supported by DARPA grant F30602-87-1-0228 from the Information Survivability program.

References

[1] The Michigan Internet AuctionBot web site: `http://auction.eecs.umich.edu`.

[2] Albert D. Baker. Metaphor or reality: A case study where agents bid with actual costs to schedule a factory. In Clearwater [4].

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

[5] The problem can be viewed as pseudo-polynomial because dynamic programming solves it in time polynomial in the sum of all agent values.

[4] Scott Clearwater, editor. *Market-Based Control: A Paradigm for Distributed Resource Allocation.* World Scientific, 1995.

[5] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[6] D. Grether. *The Allocation of Scarce Resources. Experimental Economics and the Problem of Allocating Airport Slots.* Westview Press, 1989.

[7] Faruk Gul and Ennio Stacchetti. Walrasian equilibrium without complementarities. Technical report, Princeton University and University of Michigan, 1997.

[8] J. S. Jordan. The competitive allocation process is informationally efficient uniquely. *Journal of Economic Theory*, 28:1–18, 1982.

[9] A. S. Kelso and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50:1483–1504, 1982.

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

[11] R. P. McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

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

[13] Paul Milgrom. Auction theory in practice: The simultaneous ascending auction. Technical report, Department of Economics, Stanford University, 1997.

[14] Tracy Mullen and Michael P. Wellman. A simple computational market for network information services. In *First International Conference on Multiagent Systems*, pages 283–289, San Francisco, CA, 1995.

[15] Roger B. Myerson and Mark A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29:265–281, 1983.

[16] R. E. Neapolitan and K. Naimipour. *Foundations of Algorithms.* D. C. Heath and Company, Lexington, MA, 1996.

[17] L. S. Shapley and M. Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1(2):111–130, 1972.

[18] Michael Stonebraker, Robert Devine, Marcel Kornacker, Witold Litwin, Avi Pfeffer, Adam Sah, and Carl Staelin. An economic paradigm for query processing and data migration in Mariposa. In *Third International Conference on Parallel and Distributed Information Systems*, Las Vegas, NV, 1994.

[19] Hal R. Varian and Jeffrey K. MacKie-Mason. Generalized Vickrey auctions. Technical report, Dept. of Economics, Univ. of Michigan, June 1994.

[20] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[21] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and Scott Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18:103–117, 1992.

[22] Carl A. Waldspurger and William E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Proceedings of the First Symposium on Operating System Design and Implementation (OSDI)*, pages 1–11, 1994.

[23] William E. Walsh, Michael P. Wellman, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Some economics of market-based distributed scheduling: Extended version. `http://www-personal.engin.umich.edu/~wew/Papers/mb-scheduling-extended.ps.Z`, 1998.

[24] Michael P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[25] Michael P. Wellman. A computational market model for distributed configuration design. *AI EDAM*, 9:125–133, 1995.

[26] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second International Conference on Autonomous Agents*, 1998.