Winner Determination for Combinatorial Auctions for Tasks with Time and Precedence Constraints

Güleser K. Demir
Dept of Electrical and Electronics Engineering
The Dokuz Eylul University
kalayci@eee.deu.edu.tr
Maria Gini
Dept of Computer Science and Engineering
University of Minnesota
gini@cs.umn.edu

Abstract

We present a solution to the winner determination problem which takes into account not only costs but also risk aversion of the agent that accepts the bids, and which works for auctioning tasks that have time and precedence constraints. We use Expected Utility Theory as the basic mechanism for decision-making. Our theoretical and experimental analysis shows that Expected Utility is useful for choosing between cheap-but-risky and costly-but-safe bids. Moreover, we show how bids with similar costs and similar probabilities of being successfully completed but different time windows can be efficiently selected or rejected.

Keywords: Winner determination, risk, expected utility, automated auctions, multi-agent contracting.

1 Introduction

Combinatorial auctions in which bidders submit bids on combinations of various items rather than on an individual item have recently received much attention in the research community (see, [20], [12]), in parallel with an increase in several application areas, such as auctions for railroad segments [6], for airspace [2], for allocation of radio spectrum for wireless communications [16], and for supply chain management [24].

Many business operations require combinations of goods and services, where some sort of constraint-satisfaction or combinatorial optimization problem needs to be solved in order to assemble a "deal." We envision a new generation of decision support systems that will help potential partners negotiate mutually beneficial deals. We assume that the primary negotiation paradigm among these systems will be combinatorial auctions.

The Multi-AGent NEgotiation Testbed (MAGNET) [11] represents a first step in the process of developing this vision into reality. MAGNET has unique features that allow self-interested agents to negotiate over complex coordinated tasks, with precedence and time constraints, in an auction-based market environment. Agents in MAGNET can have two roles, *customers* and *suppliers*. A customer is an agent who needs resources beyond its direct control to accomplish its goal. A supplier is an agent who can provide resources and services upon request, for specified prices, over specified time periods. MAGNET agents participate in first-price, sealed-bid, reverse combinatorial auctions. With our collaborators in the MAGNET project we have studied different issues, such as generating schedules for *Requests for Quotes* (RFQ) [1], evaluating bids to determine the winners of the auction [9, 7], and increasing the security of the whole multi-agent system [14].

In this paper we focus on winner determination. i.e the process of evaluating bids to determine the winners of an auction. We solve the winner determination problem by taking into account not only costs, time, and precedence constraints, but also uncertainty and risk posture of the agent. The work we present here uses as base for the cost formulation the method we proposed in [8], where we showed how to compute the marginal Expected Utility of completing successfully all the tasks within the duration specified in the winning bids. The formulation we used there was limited to sequential tasks. We extend it here to parallel tasks and to complex task networks. Even though our work is presented in the context of MAGNET, the results are applicable to any combinatorial auction system.

Most winner determination algorithms approach the problem as a cost minimization problem (or, equivalently, profit maximization when the auctioneer is the buyer) [22, 9]. However, we know that cost is not the only factor which plays a role in making decisions. Companies prefer to work with pre-qualified suppliers whose quality, delivery performance, and flexibility influence the long-term relationships between them. Risk and uncertainty and the risk–aversion of the seller also play a crucial role in decision making.

Winner determination for combinatorial tasks scales exponentially with the number of tasks, and at best polynomially with the number of bids [22]. More precisely, combinatorial auction winner determination is known to be $\mathcal{NP}$-complete and inapproximable [22]. Precedence and time constraints make it to scale exponentially in the number of bids as well [7].

To motivate our work, let's start with an example. Suppose you need to find resources for a job which has to be completed in a few days. You start by decomposing the job into several tasks, and you assign to each task a time window within which it has to be completed. Some tasks have also precedence constraints between them, that place constraints on the corresponding time windows. Once you have completed the decomposition into tasks and decided on the time windows, you generate a Request for Quotes (RFQ), where you specify the tasks and the start and end times for each individual task. Bids will have to include one or more tasks, a cost, an estimated duration for completing the task, and a time window within which each task can be started. When you select the winning bids, you need to estimate how likely is that each supplier will finish its tasks on time as specified in the bid. If any of them fails to finish, then your whole job will probably fail. However, you will still have to pay the costs of all the tasks which started before the one that failed.

Uncertainty depends not only on supplier reliability but also on the choice of the schedule. A schedule with slack between tasks and with wider time windows is less risky than a tight schedule, but the extra time taken can reduce the value of accomplishing the job. Although you may decrease your loss by adding a penalty for late or unfinished tasks, this might not solve completely the problems caused by uncertainty. The main issue is that there is always some probability that a supplier will not finish on time, and you have to consider that uncertainty in the winner determination process. Another important factor is how risk–averse or risk–seeking you are. While you prefer cheap but more risky bids, someone else may prefer more expensive but less risky ones. So, the winner determination process should also consider the risk tolerance of the decision maker.

In summary, if an autonomous agent is to produce decisions that are acceptable to a human decision-maker, the agent must have the ability to handle decision-making in an environment of uncertainty and must be able to deal with the risk posture of the person or organization on whose behalf the agent is acting.

To model this concept we use Expected Utility Theory [5], which describes human economic decision-making. We take into account not only costs, time, and precedence constraints, but also uncertainty and the risk posture of the agent. To accomplish this we formulate the winner determination problem as a non-linear, mixed integer programming problem with the objective of maximizing the Expected Utility (EU) of the customer agent.

In Section 2 we show how to compute EU when there are time and precedence constraints among tasks. We show how to transform a task network into an equivalent one and how to compute its EU in Section 3. We propose a formulation of the winner determination problem as a nonlinear mixed integer programming in Section 4, and we present preliminary experimental results in Section 5. Section 7 summarizes the related work. Finally, we conclude the work in Section 8.

2 Computing Expected Utility for Tasks with Time and Precedence Constraints

Utility theory has proven useful in a wide collection of applications, such as consumer demand, corporate management, portfolio analysis, land reclamation, and city-airport development [13], just to name a few.

The utility of an agent facing uncertainty is computed as the weighted average of the utility of each possible state, where the weights are the agent's estimates of the probability of each state. Therefore, EU accomplishes the decision-making process by using probabilities and an utility curve, $U(W)$, which relates a given gain level W to the utility U. Accordingly, for n given choices of outcomes, the Expected Utility is

$$E(U) = \sum_{i=1}^{n} U(W_i)\ \mathcal{P}_i \tag{1}$$

where $\mathcal{P}_i$ is the probability that the outcome W_i will occur. W_i is the resulting gain for the decision–maker if the i–th outcome is realized. We look at the risk posture of the customer agent using a von Neumann-Morgenstern utility function U, which assumes a constant risk-aversion coefficient r, which equals $-U''/U'$ [23]. We assume the utility function to have the form

$$U(W_i) = -e^{-rW_i}\ , \tag{2}$$

but other choices would work as well [1].

We represent the set of tasks as a *task network*, which consists of a set of nodes, each corresponding to a task, connected by arcs, each corresponding to a temporal constraint between tasks. A sample task network is shown in Figure 1.

[1] In this work we deal exclusively with risk-averse agents. Since the agent who runs the auction is always a buyer, we always consider utility functions with respect to cost instead of gain. We remind the reader that we assume cost to be a positive number, while gain is the negative of the cost. This function can be generalized to cover both risk-averse and risk-loving agents: $U(W_i) = -\text{sign}(r)e^{-rW_i}$ where r is positive (negative) for risk-averse (risk-loving) agents.

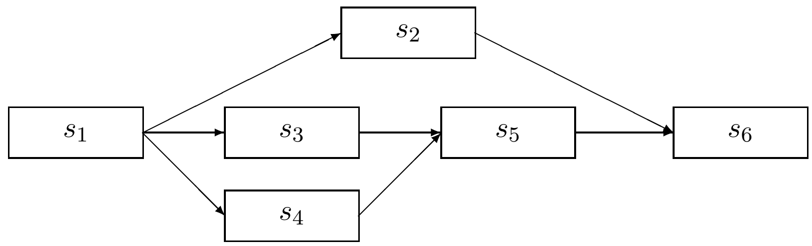

Figure 1: Task network.

To compute EU, following [1], we treat the problem as a set of ordered task completion events. Each event i has probability p_i of being completed successfully in the time allocated. We assume the probability of completion of a task is independent of other tasks because of different suppliers and different time windows. We assume that at the time of completion of each task, the customer must pay some cost c_i to the supplier. After completion of the last task, the customer gains the benefit of job completion V. Once a task starts, the customer is liable for its full cost at completion, regardless of whether, in the meantime, the job as a whole has been abandoned due to the failure of some task. If any task fails to complete we ignore any residual value of the work completed.

Suppose there are only two tasks which have to be completed sequentially. Let the probability of being completed in the time window allocated to each of them be p_1 and p_2, respectively, and the costs be c_1 and c_2. There are 3 different situations that can occur after the first task starts:

- Task_1 remains uncompleted with probability $P_1 = 1 - p_1$, you pay nothing, your gain is $W_1 = 0$, the job is abandoned (Task_2 doesn't need to start).
- Task_1 is completed, but Task_2 is not, with overall probability $P_2 = p_1.(1 - p_2)$. You pay just the cost of Task_1, c_1, your gain is $W_2 = -c_1$, the job is abandoned.
- Both Task_1 and Task_2 are completed with overall probability $P_3 = p_1 \times p_2$. You pay the cost of both tasks, $c_1 + c_2$, and gain the benefit of completing the job. Your gain is $W_3 = V - c_1 - c_2$.

To calculate EU as in Eq. (1) we use the set of probability and gain pairs, (P_i,W_i), $i = 1, 2, 3$. Clearly, the construction and the number of these pairs depend on the structure of the task network and on the schedule of the tasks. The algorithm to compute them is reported in [1].

3 Constructing an Equivalent Task Network to Compute EU

For analyzing the task network, it is advantageous to reduce sequential and parallel task groups into equivalent simpler units. We expect that such a reduction would allow for a clearer interpretation of the system, such as to show the dependence of the EU on changes in price, or the probability of success of a task, or different task networks. The topology of different task networks is illustrated in Figure 2 for sequential, parallel and parallel–sequential cases.

3.1 Sequential Tasks

We start with the case where there are n tasks to be completed, one after the other, in a sequential manner, as shown in Figure 2.

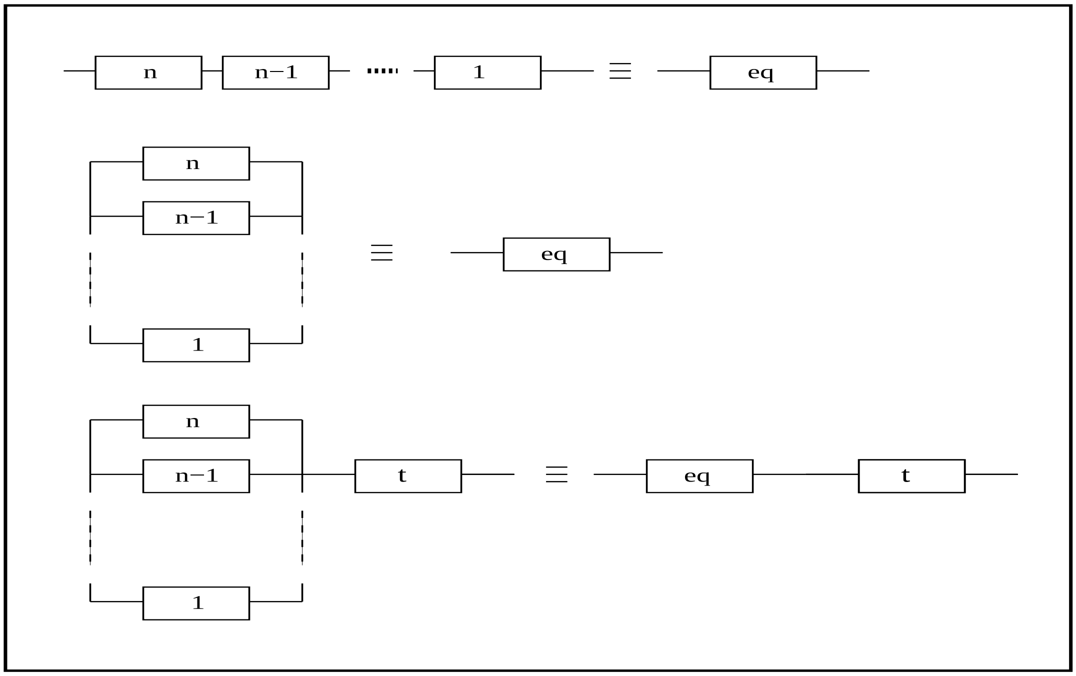

Figure 2: Typical task networks for sequential (top), parallel (middle) and parallel–sequential (bottom) cases.

Assume that there are three sequential tasks. The corresponding EU is given by

$$\mathrm{EU} \quad = -\left(1 - p_3 + p_3\,(1-p_2)\,e^{rc_3} + \; p_3p_2\,(1-p_1)\,e^{r(c_3+c_2)} + p_3p_2p_1e^{-r(V-c_3-c_2-c_1)}\right) \tag{3}$$

which equals the more suggestive form

$$\mathrm{EU} \quad = -\left(1 - p_3 + p_3e^{rc_3}\left(1 - p_2 + \; p_2e^{rc_2}\left(1 - p_1 + p_1e^{-r(V-c_1)}\right)\right)\right)\,. \tag{4}$$

This implies that for n sequential tasks the EU has the form of a *difference* equation:

$$S(k) = 1 - p_k + p_ke^{rc_k}\;S\,(k-1) \tag{5}$$

where $k = 1, 2, \cdots, n$, $S(0) = 1$ (if the nth task is the final task of the network $S(0) = e^{-rV}$), and $\mathrm{EU}(n) = -S(n)$. As a simple application of Eq. (5) consider n tasks with common probability p and cost c, which gives

$$\begin{aligned}\mathrm{EU}(n) \quad = \quad & -\frac{e^{c\,r}\left(1 - p + \left(-e^{-c\,n\,r} + e^{-r\,(c\,n+V)}\right)p^n\right)}{e^{c\,r} - p} - \\ & \frac{\left(e^{-c\,n\,r} - e^{-r\,(c+c\,n+V)}\right)p^{1+n}}{e^{c\,r} - p}\,.\end{aligned} \tag{6}$$

In general the probability p_j and cost c_j are different for different tasks, so Eq. (5) becomes a variable-coefficient difference equation. Then its solution can be found as:

$$\mathrm{EU}(n) = -\left(\Pi_{k=1}^{n}p_ke^{rc_k}\;\sum_{l=1}^{n}\frac{1-p_l}{\Pi_{m=1}^{l}p_me^{rc_m}}\right) \tag{7}$$

The fact that n sequential tasks are equivalent to a single unit with probability p_{eq} and cost c_{eq} can be seen from the following relations:

$$p_{eq} = p_n \quad \textit{and} \quad c_{eq} = c_n - \frac{1}{r}\ \log\{-\mathrm{EU}(n-1)\} \tag{8}$$

In maximizing the EU, in addition to ordering the tasks, one can follow simple rules which become evident from the difference equation (5). First of all, one recalls that $p_j < 1$ and $e^{rc_j} > 0$ so that the quantity $F_j = 1 - p_j + p_j e^{rc_j}$ is always positive, hence the EU is always negative. Examining Eq. (5) one observes that to maximize EU, if there is more than one bid for task j within the same time window, it is sufficient to choose the pair (p_j, c_j) among all bids which minimizes F_j no matter what the value of $S(j-1)$ is. The proof of this statement is as follows. Suppose that there are two bids having probability-cost pairs (p_j^i, c_j^i), $i = 1, 2$, for task j, and

$$\begin{aligned} 1 - p_j^1 + p_j^1 e^{rc_j^1} &< 1 - p_j^2 + p_j^2 e^{rc_j^2} \\ F_1 &< F_2 \ . \end{aligned} \tag{9}$$

Then, if we can show that there exists no A (take it to be $S(j-1)$) which satisfies the inequality

$$1 - p_j^1 + p_j^1 e^{rc_j^1} A > 1 - p_j^2 + p_j^2 e^{rc_j^2} A \tag{10}$$

our assertion will be confirmed. Let us start with Eq. (10), and proceed step by step:

$$\begin{aligned} 1 - p_j^1 + p_j^1 e^{rc_j^1} - p_j^1 e^{rc_j^1} + p_j^1 e^{rc_j^1} A &> 1 - p_j^2 + p_j^2 e^{rc_j^2} - p_j^2 e^{rc_j^2} + p_j^2 e^{rc_j^2} A \\ F_1 + p_j^1 e^{rc_j^1}(A-1) &> F_2 + p_j^2 e^{rc_j^2}(A-1) \\ (p_j^1 e^{rc_j^1} - p_j^2 e^{rc_j^2})(A-1) &> F_2 - F_1 \\ (F_1 - F_2 + (1-p_j^2) - (1-p_j^1))(A-1) &> F_2 - F_1 \\ (p_j^1 - p_j^2)(A-1) &> -(F_1 - F_2)A \\ p_j^1 A - p_j^2 A - p_j^1 + p_j^2 &> -A(1 - p_j^1 + p_j^1 e^{rc_j^1} - (1 - p_j^2 + p_j^2 e^{rc_j^2})) \\ -p_j^1 + p_j^2 &> A(-p_j^1 e^{rc_j^1} + p_j^2 e^{rc_j^2}) \\ A(p_j^1 e^{rc_j^1} - p_j^2 e^{rc_j^2}) &> p_j^1 - p_j^2 \end{aligned} \tag{11}$$

This inequality is in manifest contradiction with our starting assumption that $F_1 < F_2$, and thus

$$1 - p_j^1 + p_j^1 e^{rc_j^1} \quad < \quad 1 - p_j^2 + p_j^2 e^{rc_j^2} \tag{12}$$

hence

$$1 - p_j^1 + p_j^1 e^{rc_j^1} A < 1 - p_j^2 + p_j^2 e^{rc_j^2} A \tag{13}$$

for all $A < 1$ (In our task network $S(0) = e^{-rV}$ where V is greater than the sum of all costs, so that A remains less than 1).

In addition, among all bids for task j we choose the one having the lowest cost for identical probabilities, and we choose the one having highest probability for identical costs. All these observations (natural human decisions) are supported by the form of F_j.

If there are no time constraints on the order of the tasks, forming a sequential task network can be done as follows. For two distinct tasks with identical probabilities, the one having the lowest cost must come first. On the other hand, when the costs are identical, the one with lowest probability must be done first. In the other cases, one requires $-p_j + p_j(1-p_l)e^{rc_j} < -p_l + p_l(1-p_j)e^{rc_l}$ for task j to precede task l.

3.2 Parallel Tasks

As suggested by Figure 2, the EU of two parallel tasks takes the form

$$\mathrm{EU} = (1-p_1)p_2e^{rc_2} + p_1(1-p_2)e^{rc_1} + (1-p_1)(1-p_2) + p_1p_2e^{r(c_1+c_2)} \tag{14}$$

A direct evaluation of Eq. (14) reveals that the EU of n parallel tasks can be obtained by solving the difference equation:

$$P(k) = (1 - p_k + p_ke^{rc_k})\ P(k-1) \tag{15}$$

where $k = 1, 2, \cdots, n$, $P(0) = 1$, and $\mathrm{EU}(n) = -P(n)$. A solution of Eq. (15) reduces a group of parallel tasks into a single unit with probability p_{eq} and cost c_{eq}:

$$\begin{aligned} p_{eq} &= 1 - \Pi_{i=1}^{n}(1-p_i) \\ c_{eq} &= -\frac{1}{r}\ \log\{-\frac{\mathrm{EU}(n) + 1 - p_{eq}}{p_{eq}}\} \end{aligned} \tag{16}$$

It is clear that to maximize EU one has to choose the (p_j, c_j) among all bids given for task j which minimizes $F_j = 1 - p_j + p_je^{rc_j}$. In general, the observations made for the sequential case, that among all bids for task j we choose the one with the lowest cost for identical probabilities, and we choose the one with the highest probability for identical costs, are also valid here.

3.3 Parallel–Serial Tasks

The configuration illustrated in the bottom of Figure 2 differs from the other two because it includes both sequential and parallel tasks.

A straightforward analysis reduces such a combination to a pure sequential combination, where the parallel tasks are reduced to an equivalent unit, via the following rules:

$$\begin{aligned} p_{eq} &= 1 - \Pi_{i=1}^{n}(1 - p_i + p_ie^{rc_i}) + \Pi_{i=1}^{n}p_ie^{rc_i} \\ c_{eq} &= -\frac{1}{r}\ \log\{-\frac{\Pi_{i=1}^{n}p_ie^{rc_i}}{p_{eq}}\}\ . \end{aligned} \tag{17}$$

3.4 Reduction of Hybrid Networks

In this section we will use the rules derived above to find the equivalent for a general hybrid network. Given an auction where there are multiple bids for a given task, each bid with its own time window and cost preferences, then the network is established when the time ordering of the tasks is fixed. In what follows we assume that such an ordering of tasks has already been made so that there are various branches with parallel or sequential structures. The reduction methods developed previously are useful in reducing a complex network structure into simple units with known expressions for equivalent costs and probabilities. In discussing the reduction of a hybrid structure we will refer to Figure 3. The reduction process starts at the top of the figure where a typical network structure is given. As shown there one starts combining the sequential tasks in the branches encircled by dashed lines. Using the rules derived in Section 3.1 one passes to the next step shown in the second row of the figure which now contains *eq1* and *eq2* with respective equivalent probabilities and costs. Now, using the rules derived in Section 3.3 one can combine the three parallel tasks in the second row of the figure which are encircled. This gives the equivalent unit *eq3* shown in the third row of the figure. At this stage one should use the rules presented in Section 3.1 to combine the tree sequential tasks into a single unit *eq*, as shown in the last row of the figure. Clearly, *eq* is expressed in terms of various equivalent probabilities and costs as described above.

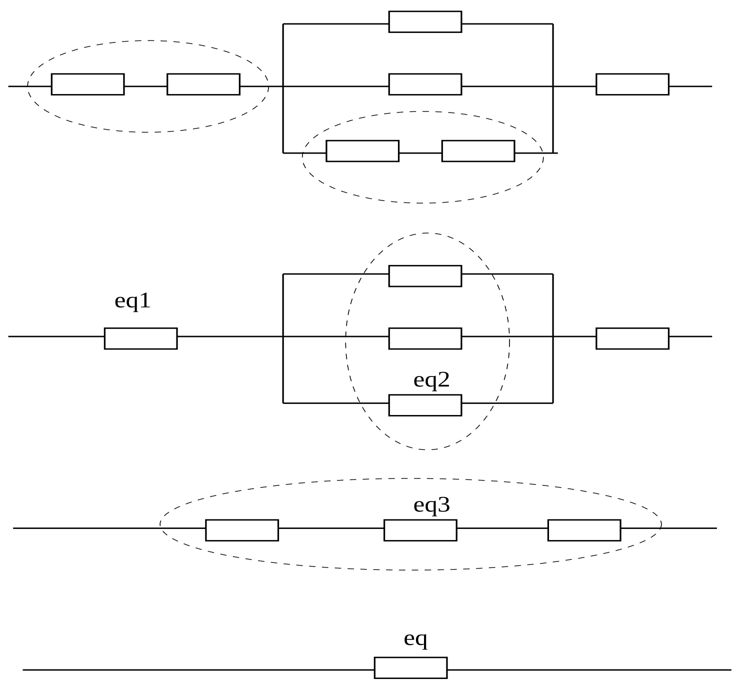

Figure 3: A sample task network for illustrating the reduction to a single equivalent unit.

4 Winner Determination Formulation

As noted before, we are interested in choosing bids that maximize the customer agent Expected Utility and that satisfy the temporal and precedence constraints of the tasks. Customer agents do a first-price, sealed-bid reverse auction. A bid includes a set of tasks and a cost, along with timing data, duration, and the earliest and latest times the task(s) may be started. We assume that information on the probabilities of completion of the tasks by the supplier/bidder can be obtained from market aggregate data.

In previous work, we have developed various algorithms for winner determination in MAGNET, including one based on Integer Programming [10], IDA* using bidtree ordering [9], and Simulated

Annealing. Those algorithms use cost as the main criterion for choice.

To formulate the winner determination problem, we start by introducing some notations and concepts. A task network consists of a set $\mathcal{S}$ of tasks with elements $s_j, j = 1 \cdots m$. Each task s_j has a precedence set $\mathcal{P}_j = \{s_{j'} | s_{j'} \prec s_j\}$, the set of tasks $s_{j'}$ that must be completed before s_j is started. At the conclusion of the bidding process, we have a set $\mathcal{B}$ of bids with elements $b_i, i = 1 \cdots n$. Each bid b_i specifies a set of tasks $\mathcal{S}_i$ and a cost c_i. For each task s^i_j included in $\mathcal{S}_i$, a bid b_i may specify an early start time e^i_j, a late start time f^i_j, and a duration d^i_j. Additionally, we need to know the probability p^i_j that task s^i_j will be completed in the proposed time duration by the bidder of bid b_i.

To formulate the winner determination problem it is necessary to introduce the main decision variable x_i which is associated with each bid b_i:

$$x_i = \begin{cases} 1 & \text{if } b_i \text{ is accepted (winner)} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Although in a combinatorial auction each bid has a cost for the entire set of tasks, we assign the entire cost to the finishing task of a bid and give zero cost to all the preceding ones. Namely, we assign costs so that $c^i_j = 0$ if the task j is not the finishing one in the set $\mathcal{S}_i$, and $c^i_j = c_i$ if the task j is the last one. Doing so, we assume that there is no payment until all the tasks of a given combinatorial bid are completed.

To use EU for winner determination, it is useful to regard each bid as a task of probability $(p^i_j)^{x_i}$ and cost $x_i c^i_j$, where their time parameters determine the structure. Clearly, once the x_i's are determined, the rejected bids will be represented by the set $\{1, 0\}$ in the bid structure (unit probability and vanishing cost, no effect on EU), and the ones accepted will have the set $\{p^i_j, c^i_j\}$, where we recall that x_i is either zero or one.

Although there are time precedence relations between some tasks there can be different tasks orderings which satisfy the precedence constraints. For example, consider the network which has the following precedence relations: $s_1 \prec (s_2, s_3, s_4)$, $(s_2, s_5) \prec s_6$, and $(s_3, s_4) \prec s_5$. According to its start time, s_4 can be sequential to s_3 or parallel to s_3. Figure 1 shows the task network when s_4 is parallel to s_3 under the above precedence constraints. Each of these different tasks orderings will yield a different network structure and a different value for EU. In the winner determination process, to track these different orderings and compute the related EU we use a continuous variable l^i_j, which represents the start time of a task in each bid. We have to construct the bid structure and compute the associated EU as l^i_j varies between the early and late start times, and the variable x_i decides if the bid under consideration is rejected or accepted.

In solving the winner determination problem one must take into account the following constraints on the fundamental variables l^i_j and x_i:

- Bid Selection:

$$x_i \quad \in \quad \{0, 1\} \;\; \forall i = 1 \cdots n \tag{19}$$

  which means that a bid is either accepted or rejected.

- Start Time Limits:

$$e^i_j \leq l^i_j \leq f^i_j \;\; \forall j = 1 \cdots m, \; \forall i = 1 \cdots n, \tag{20}$$

  which ensures that the start time of a task is bounded by its early and late start times.

- Coverage:

$$\sum_{i|s_j \in \mathcal{S}_i} x_i = 1 \;\forall j = 1 \cdots m \tag{21}$$

which means that each task must be included exactly once.

- Feasibility

$$\begin{aligned} x_i l^i_j &\geq x_{i'}(l^{i'}_{j'} + d^{i'}_{j'}) - M(1 - x_i) \\ \forall j &= 1 \cdots m, \forall i | s_j \in \mathcal{S}_i, \forall i' | s_{j'} \in (\mathcal{S}_{i'} \cap \mathcal{P}_j) \end{aligned} \tag{22}$$

which guarantees both locally and globally that each task starts after the completion of its immediate predecessors. The feasibility constraint in Eq. (22) is a nonlinear function that can be rewritten in linear form as

$$\begin{aligned} l^i_j &\geq l^{i'}_{j'} + d^{i'}_{j'} - M(2 - x_i - x_{i'}) \\ \forall j &= 1 \cdots m, \forall i | s_j \in \mathcal{S}_i, \forall i' | s_{j'} \in (\mathcal{S}_{i'} \cap \mathcal{P}_j) \;, \end{aligned} \tag{23}$$

We prefer to use this linear form in the winner determination process. Here, M is a "large" number, and the last term $M(2 - x_i - x_{i'})$ is used to make the constraint satisfied in the case where $x_i = 0$ or $x_{i'} = 0$.

The number of constraints generated by these formulas is highly variable, and depends on the length of the longest path in the task network and on the detailed composition of the bids. Details on how to derive similar constraints for winner determination of combinatorial bids using Integer Programming are given in [10].

5 Case Study

In this Section, we show the results of a numerical study of winner determination employing the EU theory we have presented.

We simulate a bidding cycle by introducing different sets of bids, each for different experiments. Figure 4 shows various bids. Each line corresponds to a specific bid, and includes the corresponding task, early start times, duration, latest finish time, cost, and probability of success in the given time window. Different data sets are indicated by different patterns for the corresponding bids.

The task network we use in these experiments is the one shown earlier in Figure 1. The network has the following precedence relations: $s_1 \prec (s_2, s_3, s_4)$, $(s_2, s_5) \prec s_6$, and $(s_3, s_4) \prec s_5$.

5.1 Experiment 1

In this experiment we use only bids $b_1 \cdots b_1 1$. Each bid includes just one task and in the winning–bid structure all tasks are scheduled sequentially.

We arrive at the following results from this experiment:

- Bid b_1 is rejected in all cases, since it requires 30 units of time but the bids for task s_2 (b_3, b_7, and b_8) must start no later than at time 15; therefore, the choice of b_1 violates the precedence relations. As a result, bid b_2 wins task s_1.

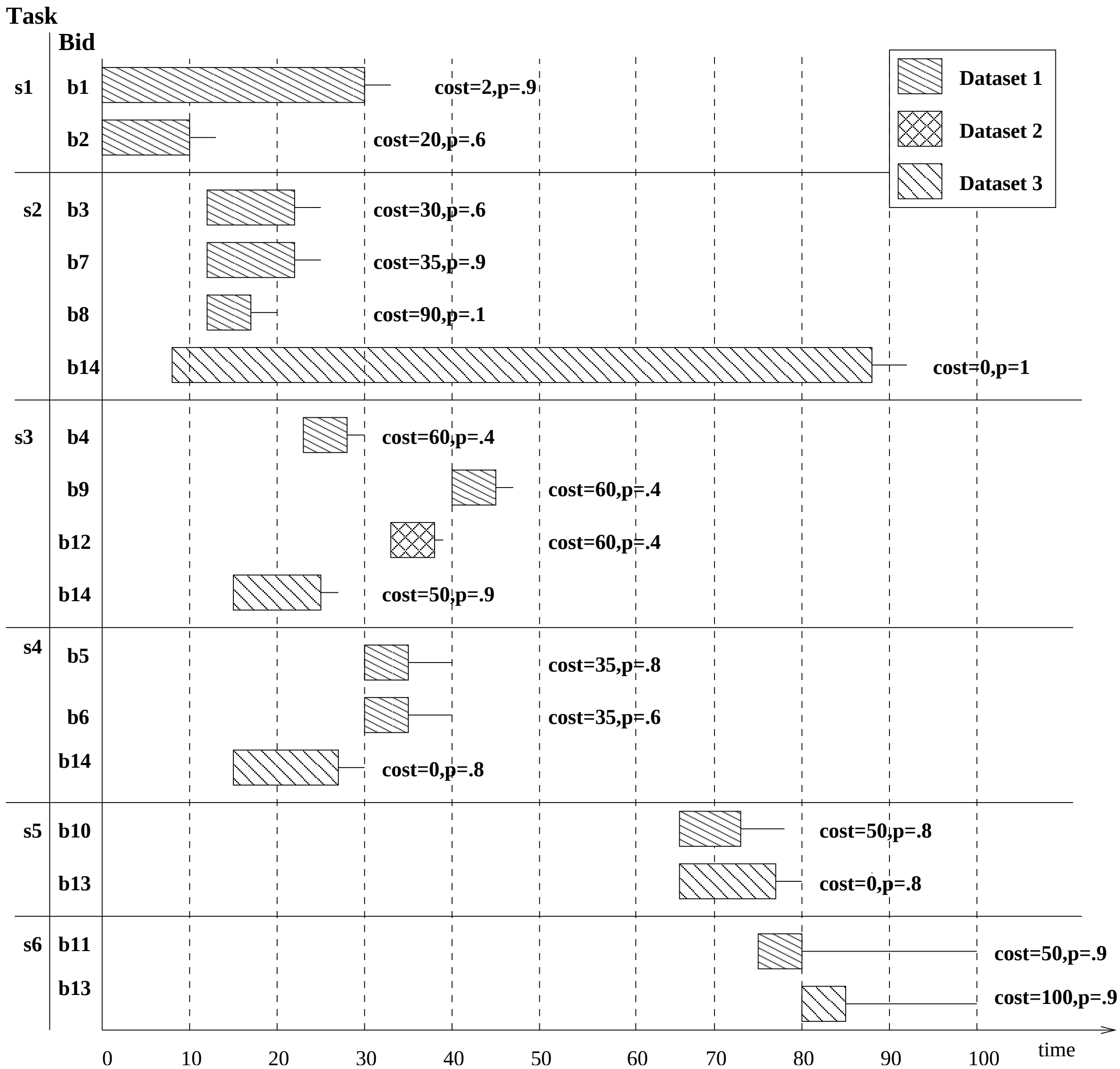

Figure 4: A bidding cycle. The figure shows early start time, duration, latest finish time, cost, and probability values for bids b_i for tasks s_j.

- Task s_4 is covered by bids b_5 and b_6 with identical costs but different probabilities. As expected, bid b_5 wins, thanks to its higher chance of completing the task.
- Task s_2 is covered in b_3, b_7 and b_8. Obviously, bid b_8 loses due to its high cost and rather small probability of success. Deciding on which bid, b_3 or b_7, to accept is more complicated and its solution requires knowledge of the costumer's risk preferences. Essentially, the customer must have a preference for *costly and less risky* or *cheap and more risky* situations. Here the risk is parametrized by the *risk factor* r present in the utility function in Eq. (2). Therefore, such a situation illustrates one of the main uses of EU. Depending on how risk–averse the

customer is, either b_3 or b_7 wins.

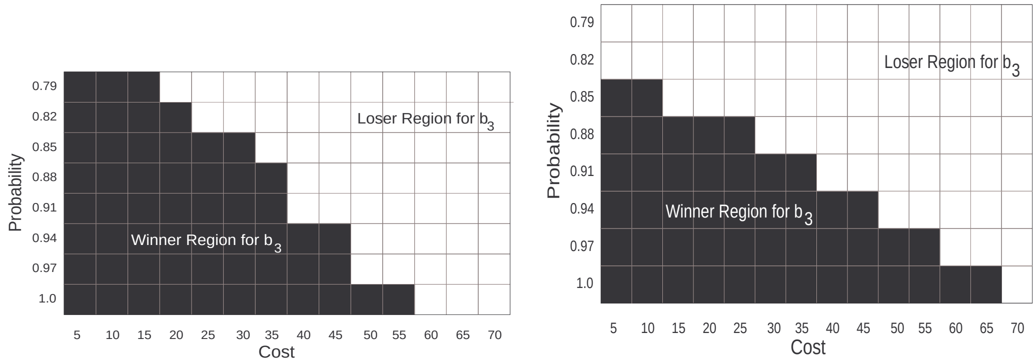

Figure 5: The winner/loser regions for b_3 with two different risk-aversion coefficient; (left) r=0.001 and (right) r=0.0001.

To illustrate the situation, it is useful to perform another experiment and examine the effects of different risk factors on the winning bids. Depicted in Figure 5 are the winner/loser regions in the cost–probability plane for bid b_3 with respect to b_7 for two different risk factors: $r = 0.001$ and $r = 0.0001$. Here, $r = 0.001$ is for a more risk–seeking customer compared to the case with $r = 0.0001$. As the figure shows, for $r = 0.001$ less likely to succeed and cheaper alternatives are preferred, whereas for $r = 0.0001$ the customer is more risk–averse and prefers more costly but more likely to succeed bids.

- Bids b_4 and b_9 both cover task s_3, where b_4 completes the task before s_4 starts, whereas b_9 starts task s_3 after s_4 is completed. Both of these orderings satisfy the feasibility constraints, and therefore, the winning bid can be decided only after maximizing the EU. In Figure 6, we show the winner/loser regions in the cost–probability plane for b_9 relative to b_4 for $r = 0.001$

  The importance of the order for bid evaluation can be explained as follows. Suppose that the bids for s_3 and s_4 have equal probabilities but different costs. The natural choice is to do the task with lower cost first to minimize the total cost in case the job is abandoned. On the other hand, if the costs for s_3 and s_4 are equal but the probabilities are different then the one with lower probability must be chosen first, again for minimizing the total cost in case the job is abandoned. EU supports and realizes both of these observations.

In light of the observations above, the winning bid vector is as follows : $X = \{0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1\}$.

5.2 Experiment 2

So far we have illustrated mainly sequential structures, where winning bids always start one after the other. Now we extend the scenario by adding b_12 to $b_1 \cdots b_11$ from Experiment 1.

We illustrate the winner/loser regions for b_{12} in Figure 7. A comparison with Figure 6 in which the winner/loser regions for b_9 are depicted for the same task, shows that the customer, when b_{12} is included, prefers to use b_{12} because of its lower cost.

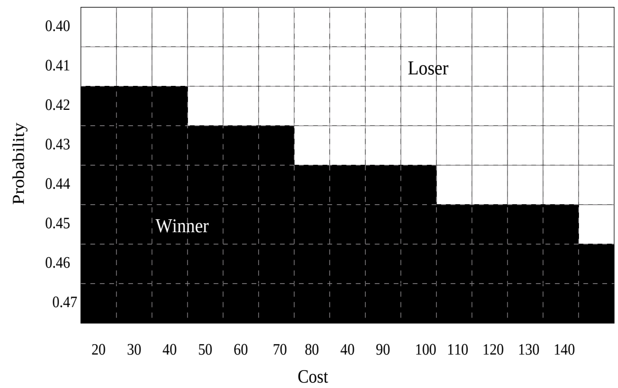

Figure 6: The winner/loser regions for b_9 with risk coefficient r=0.001 for Dataset 1.

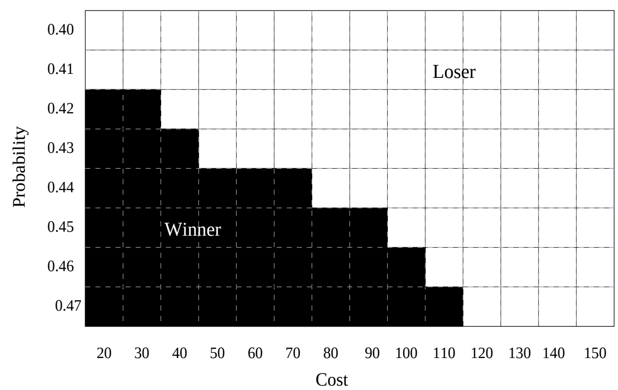

Figure 7: The winner/loser regions for b_{12} for Dataset 1 and Dataset 2.

5.3 Experiment 3

In this experiment, all the bids from Figure 4, including the individual and combinatorial bids, are used to determine the winning bids. We compare bids b_{10} for task s_5 and b_{11} for task s_6 versus a combinatorial bid, b_{13}, for both s_5 and s_6. In both cases, the tasks s_5 and s_6 are in sequential order. In Figure 8 the EU values for each cases are shown for different costs of b_{13}. As seen from the figure, for costs higher than 140 the EU of the bids including b_{13} is lower than the EU of the bids including b_{10} and b_{11}. Therefore above the cost 140 the bids b_{10} and b_{11} are the winning ones. It is useful to look at Figure 8 from a different perspective. Assume s_i and $s_{i'}$ are sequential tasks connected to a network. The EU of the network is computed as

$$EU = -K\left(1 - p^i_j + p^i_j e^{rc^i_j}\left(1 - p^{i'}_{j'} + p^{i'}_{j'} e^{-r(V-c^{i'}_{j'})}\right)\right) \tag{24}$$

where we took into account the fact that s_i and $s_{i'}$ are the last two tasks. Here K represents the contribution of the rest of the network. We denote the probability of completion of task j given by bid i as p^i_j and the corresponding cost as c^i_j. It is useful to compare the EUs of the two cases:

$$\text{LHS} \quad = \quad -K\left(1 - p^{10}_5 + p^{10}_5 e^{rc^{10}_5}\left(1 - p^{11}_6 + p^{11}_6 e^{-r(V-c^{11}_6)}\right)\right)$$

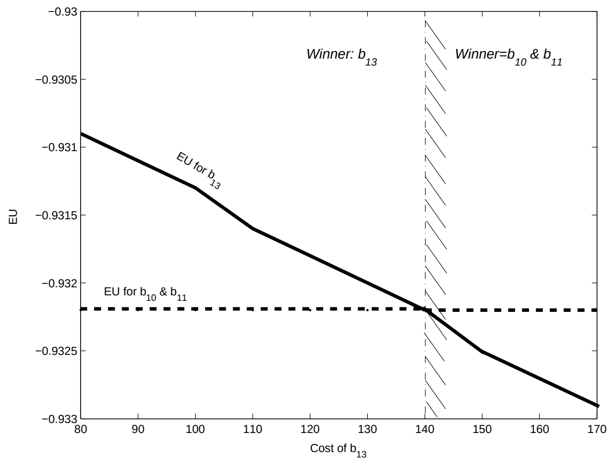

Figure 8: The EU of the combinatorial bid b_{13} vs. the individual bids b_{10} and b_{11}.

$$\text{RHS} \;=\; -K\left(1 - p_5^{13} + p_5^{13}\left(1 - p_6^{13} + p_6^{13} e^{-r(V - c_{5,6}^{13})}\right)\right) \tag{25}$$

where LHS refers to individual bids, and RHS to the combinatorial bid for s_5 and s_6. Whichever one, LHS or RHS, is larger that bid(s) is the winner(s). Irrespectively of the value of K, one can determine the winner by checking Eq. (25) for different cost and probability values.

As an example, let's take the parameters from Figure 4 and vary the cost of b_{13}. For individual bids one finds $-0.3918K$. On the other hand, for combinatorial bids one gets $-0.3877K$, $-0.3910K$, $-0.3932K$ for $c_{5,6}^{13} = 100, 130$ and 150, respectively. Choosing the bid(s) whose corresponding cost(s) gives maximum EU one can easily determine that b_{13} is the winner if its cost is 100 or 130. If the cost of b_{13} is 150 then the winner bids are b_{10} and b_{11}. This simple analysis is sufficient to determine the winning bids without considering the remaining part of task network. Indeed, the EU for the combinatorial case falls below that of the individual case at $c_{5,6}^{13} = 140$, which is in complete agreement with Figure 8. In this way, which is nothing but a restatement of the equivalent network construction, one can eliminate various bids in a simple way.

6 Experimental Results on Scalability

To illustrate how our approach scales we have made several experiments. Although there are common combinatorial auction benchmarks distributions, such as CATS (Common Auction Test Suite) [15], we have not used instances generated by these distributions in our experiments because those distributions do not cover some essential parameters, such as time-window specifications of tasks, precedence relations between tasks, and risk-aversion of customer.

Our experimental setup to generate problem sets, as in [7], consists of 3 main units:

- **Task network generator** in which the desired number of tasks is generated, and random precedence relations among them are created;

- **Bid generator** in which each bid is generated by selecting a task at random from the task network, by assigning randomly time window values using predefined time-window parameters

Number of tasks	Number of bids	Time	Std in time
5	10	6.3040	3.5423
5	13	36.0318	11.6602
5	16	176.5704	110.5704
5	19	324.3663	178.7854
5	21	2667.2	1686.3

Table 1: Experiments with varying the number of bids. Time is in milliseconds.

Number of tasks	Number of bids	Size of bids	Time	Std in time
3	14	2.1	874.652	588.726
5	14	1.6	979.2	541.41
6	14	1.8	1248.1	623.3

Table 2: Experiments with varying number of tasks. Time is in milliseconds.

(such as expected duration, variability of the late start time, etc.), and by determining a uniformly distributed cost between [0,1] for the overall bid;

- **Bid evaluator** in which the winner-determination search is performed by considering the risk-aversion coefficient of the customer and giving random probabilities of completion to the selected tasks by the given bidders.

Each block requires a stream of random numbers. We generate these streams using pre-specified seeds, so we can use the same task networks with different bid sets. For the following experiments we generated several problem sets with randomly-generated task networks and randomly-generated bids. We kept constant all of the parameters of the problem generator except the task count and bid count. All experiments were conducted on a 1.6GHz Pentium M processor PC with 496MB RAM.

In the first experiment we computed the performance of the algorithm with varying number of bids while keeping the number of tasks constant. Table 1 shows the result of this experiment. The "time" column gives the average values in milliseconds and "Std in time" column gives the standard deviation in time for several randomly generated problem sets for the given number of tasks and of bids. As the number of bids increases, the resulting choices to be considered for determining EU increase exponentially; this requires too much CPU time to determine the EU in each search-step of the nonlinear mixed integer solver. It is evident from the table that the mean time scales exponentially with the number of bids.

The second experiment shows the scalability of the search process as the size of the task network varies. Table 2 shows problem characteristics for this experiment. In this experiment we tried to keep the size of the bids, i.e. the mean number of tasks specified in each bid, constant. In the table, the bid size is not constant exactly due to the nature of the bid generation process.

7 Related Work

Despite the abundance of work in auctions [17, 12], limited attention has been devoted to auctions over tasks with complex time constraints and uncertainty. Execution uncertainty is studied in [19] where the design of mechanisms is extended from traditional game-theoretic approaches to take into account not only cost but also probability of failure. A software framework with APIs for winner determination algorithms is introduced in [4]. The framework extends auction mechanisms to single/multi-unit, single/multiple attributes, single/multiple items, but does not include time and other constraints.

In [18], a method is proposed to auction a shared track line for train scheduling. The problem is formulated with mixed integer programming, with many domain-specific optimizations. Time slots are used in [26], where a protocol for decentralized scheduling is proposed. The study is limited to scheduling a single resource, while MAGNET agents deal with multiple resources. Walsh et al [24] propose a protocol for combinatorial auctions for supply chain formation, using a game-theoretical perspective. They allow complex task networks, but do not include time constraints like we do in MAGNET.

Agents in MASCOT [21] coordinate scheduling with the user. Their major objective is to show policies that optimize schedules locally. Our objective is to optimize the customer's utility. In MAGNET we are not scheduling resources the agent has, we are producing a schedule that other agents will execute. The results reported in [25] on the problem difficulty in job-shop scheduling share many similarities to the problems we encounter when solving the winner determination problem in MAGNET.

A commonly used way for handling multiple attributes is to convert qualitative attributes into price-equivalents, or to define scoring functions for each attribute and combine the scores by using a utility function [3]. Unfortunately, this approach does not extend easily to time and precedence constraints.

In previous work [1] we proposed an approach based on Expected Utility to compute agent's preferences over different schedules for the tasks in an RFQ, and proposed the algorithm for computing the payoff-probability outcomes of different schedules that we are using here. The major difference is that there we were interested in generating schedules of tasks to maximize expected utility before any bid is submitted. Here we are maximizing the agent's expected utility once the bids have been submitted, in the winner determination process.

8 Conclusions

MAGNET is designed to support negotiation among multiple, heterogeneous, self-interested agents over the distributed execution of complex tasks. If an agent is to act on behalf of a human decision-maker in such an environment it must be able to evaluate risk factors in ways that the person will find reasonable. We believe Expected Utility Theory offers a good framework for doing this.

In this work we have studied the winner determination problem using EU so that not only cost but also time and risk postures are taken into account. EU proves useful for making decisions when one must choose between *cheap-but-risky* and *costly-but-safe*. Moreover, EU is a powerful tool for choosing between bids with similar costs and probabilities but different time spans in feasible regions. The theoretical bases discussed in Section 4 as well as our examples confirm the usefulness of EU in determining the winning bids.

The maximization of EU is solved as a nonlinear mixed integer programming problem. While the continuous variation of the individual start times for each task determines the structure, the acceptance/rejection of a bid is decided via an integer variable.

Additionally, we have shown how to find an equivalent task network to compute EU. The calculation of equivalent units proves useful in reducing a complex network structure into simple units with known expressions for equivalent costs and probabilities.

Acknowledgment

Work supported in part by the National Science Foundation under grant IIS-0414466.

References

[1] A. Babanov, J. Collins, and M. Gini. Asking the right question: Risk and expectation in multi-agent contracting. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 17(4):173–186, September 2003.

[2] M. Ball, G. Donohue, and K. Hoffman. Auctions for a safe, efficient, and equitable allocation of airspace system resources. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

[3] M. Bichler. *The Future of eMarkets: Multi-Dimensional Market Mechanisms*. Cambridge University Press, 2001.

[4] M. Bichler, J. Kalagnanam, H. S. Lee, and J. Lee. Winner determination algorithms for electronic auctions: A framework design. In K. Bauknecht, A. M. Tjoa, and G. Quirchmayr, editors, *E-Commerce and Web Technologies*, LNCS 2455. Springer-Verlag, 2002.

[5] T. Biswas. *Decision-Making Under Uncertainty*. St. Martin's Press, Inc., 1997.

[6] P. J. Brewer. Decentralized computation procurement and computational robustness in a small market. *Economic Theory*, 13:41–92, 1999.

[7] J. Collins. *Solving Combinatorial Auctions with Temporal Constraints in Economic Agents*. PhD thesis, University of Minnesota, Minneapolis, Minnesota, June 2002.

[8] J. Collins, C. Bilot, M. Gini, and B. Mobasher. Decision processes in agent-based automated contracting. *IEEE Internet Computing*, 5(2):61–72, March 2001.

[9] J. Collins, G. Demir, and M. Gini. Bidtree ordering in IDA* combinatorial auction winner-determination with side constraints. In J. Padget, O. Shehory, D. Parkes, N. Sadeh, and W. Walsh, editors, *Agent Mediated Electronic Commerce IV*, volume LNAI2531, pages 17–33. Springer-Verlag, 2002.

[10] J. Collins and M. Gini. An integer programming formulation of the bid evaluation problem for coordinated tasks. In B. Dietrich and R. V. Vohra, editors, *Mathematics of the Internet: E-Auction and Markets*, volume 127 of *IMA Volumes in Mathematics and its Applications*, pages 59–74. Springer-Verlag, New York, 2001.

[11] J. Collins, W. Ketter, and M. Gini. A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. *Int'l Journal of Electronic Commerce*, 7(1):35–57, 2002.

[12] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[13] A. Goicoechea, D. Hansen, and L. Duckstein. *Multiobjective Decision Analysis with Engineering and Business Applications*. John Wiley & Sons, New York, 1982.

[14] A. Jaiswal, Y. Kim, and M. Gini. Security model for a multi-agent marketplace. In *Proc. of 5th Int'l Conf. on Electronic Commerce*, pages 119–124, Pittsburgh, PA, October 2003.

[15] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, pages 66–76, Minneapolis, MN, October 2000.

[16] J. McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 1994.

[17] P. Milgrom. Auction and bidding: a primer. *Journal of Economic Perspectives*, 3(3):3–22, 1989.

[18] D. C. Parkes and L. H. Ungar. An auction-based method for decentralized train scheduling. In *Proc. of the Fifth Int'l Conf. on Autonomous Agents*, pages 43–50, Montreal, Quebec, May 2001. ACM Press.

[19] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *UAI-02*, 2002.

[20] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[21] N. M. Sadeh, D. W. Hildum, D. Kjenstad, and A. Tseng. MASCOT: an agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In *Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain, at Agents '99*, pages 133–138, 1999.

[22] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, 2002.

[23] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavor.* Princeton Univ. Press, Princeton, 1947.

[24] W. E. Walsh, M. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, pages 260–269, October 2000.

[25] J. P. Watson, J. C. Beck, A. Howe, and L. D. Whitley. Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence*, pages 189–217, 2002.

[26] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35:271–303, 2001.