Risk and User Preferences in Winner Determination*

Güleser Demir
Department of Computer Science and Engineering, University of Minnesota and
Department of Electrical and Electronics Engineering, University of Dokuz Eylul, Turkey

Maria Gini
Department of Computer Science and Engineering, University of Minnesota
gdemir, gini@cs.umn.edu, demir@eee.deu.edu.tr

Abstract

We discuss 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. We are interested in bids for tasks that have time and precedence constraints. We develope an equivalent unit approach to the group of tasks to analyze the system and use Expected Utility Theory as the basic mechanism for decision-making. Our theoretical and experimental analysis shows that Expected Utility is especially 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

Parallel to the fact that auctions are becoming the basic negotiation tools for e–commerce systems, solving the winner determination problem in auctions has become an important problem. Generally, many winner determination algorithms approach the problem as a cost minimization problem (or, equivalently, profit maximization when the auctioneer is the buyer) [Sandholm, 2002, Collins *et al.*, 2002a]. However, we know that cost is not the only factor playing a rôle in making decisions. Usually, companies tend to work with prequalified suppliers whose quality, delivery performance, and flexibility influence the long-term relationships among them.

There are further issues concerning how one deals with risk and uncertainty. Suppose you have planned a job in which several tasks should be completed with time precedence relations. At the winner determination stage you have to select the winning bids, but you don't know whether all suppliers will finish their tasks on time. If any of them fails to finish, then your whole plan will probably fail. However, you will still have to pay the costs of all the tasks which precede 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 the final product. Although we may use a penalty for late or unfinished tasks to decrease our loss, this is not a complete solution to deal with

*Work supported in part by the National Science Foundation, awards NSF/IIS-0084202 and NSF/EIA-9986042

uncertainty. The main issue is that there is always some probability that suppliers may not finish on time, and you have to consider that uncertainty in your winner determination process.

In addition to uncertainty, there is another important issue concerning how risk–averse or risk–seeking you are. While you prefer cheap but more risky bids, someone can prefer more expensive but less risky ones. So, winner determination should also consider the risk tolerance.

In our work, what complicates winner determination further is the fact that tasks have precedence constraints and that all tasks have to be completed. Winner determination for combinatorial tasks scales exponentially with the number of tasks, and at best polynomially with the number of bids [Sandholm, 2002]. However, temporal constraints make the winner-determination problem to scale exponentially in the number of bids as well [Collins, 2002].

In this work, we focus on the MAGNET (Multi-AGent NEgotiation Testbed) [Collins *et al.*, 2002b] system, but the results are applicable to other systems as well. MAGNET supports agents in negotiation of contracts for tasks that have temporal and precedence constraints. In MAGNET there are two agent roles: the *customer* and the *supplier*. 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.

To model this concept we use Expected Utility Theory (EUT) [Biswas, 1997], which describes human economic decision-making. We take into account not only costs, time, and precedence constraints, but also 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. In Section 3 we show how to transform a task network into an equivalent one and how to compute its EU. Then, in Section 4 we propose a formulation of the winner determination problem as a nonlinear mixed integer programming. In Section 5 we show experimental results. Section 6 summarizes the related work. Finally, we conclude the work in Section 7.

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 [Goicoechea *et al.*, 1982], just to name a few.

What EU does is to accomplish 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.

Suppose your duty is to complete a set of tasks. It is the responsibility of your customer agent to use an auction process to obtain a set of commitments for resources, that can be composed into a feasible schedule with maximum expected utility.

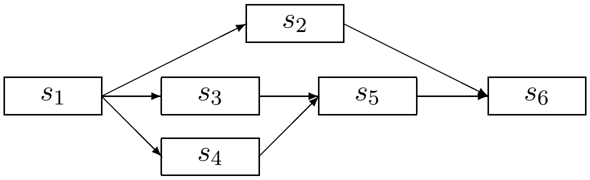

Figure 1: Task network

We represent the set of tasks as a *task network*, which consists of a set of nodes and the temporal constraints among them. A sample task network is shown in Figure 1

To compute EU, 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. At the time of completion of each event, the customer must pay some cost c_i to the supplier. After completion of the last task, the customer gains the benefit of plan completion V. Once a task starts, the customer is liable for its full cost at completion, regardless of whether, in the meantime, the plan 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 their probability and costs be p_1, p_2, c_1, and c_2 respectively. There are 3 different situations that can occur after the first task starts:

- Task$_1$ remains uncompleted with probability $P = 1 - p_1$, you pay nothing, your gain is $W = 0$, the plan is abandoned (Task$_2$ doesn't need to start).
- Task$_1$ is completed, but Task$_2$ is not, with overall probability $P = p_1.(1 - p_2)$. You pay just the cost of Task$_1$, c_1, your gain is $W = c_1$, the plan is abandoned.[1].
- Both Task$_1$ and Task$_2$ are completed with overall probability $P = p_1.p_2$. You pay the cost of both tasks, $c_1 + c_2$, your gain is $W = V + c_1 + c_2$, the plan is completed, you gain the benefit of completion of the plan.

To calculate EU as in (1) we use the set of probability and gain pairs, (P_i, W_i), $i = 1, 2, 3$ Clearly, the construction of these pairs depends on the structure of the task network and on the schedule of the tasks. The algorithm to compute them is reported in [Babanov *et al.*, 2002].

We collect the preferences of the customer agent in a von Neumann-Morgenstern utility function U, assuming constant risk-aversion coefficient r, which equals $-U''/U'$ [von Neumann and Morgenstern, 1947]. We take the utility function to have the form $U(W_i) = -e^{-rW_i}$.

3 Constructing an Equivalent Task Network to Compute EU

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

[1] Since all costs are defined in terms of gain all of them are negative

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. The EU for three sequential tasks is given by

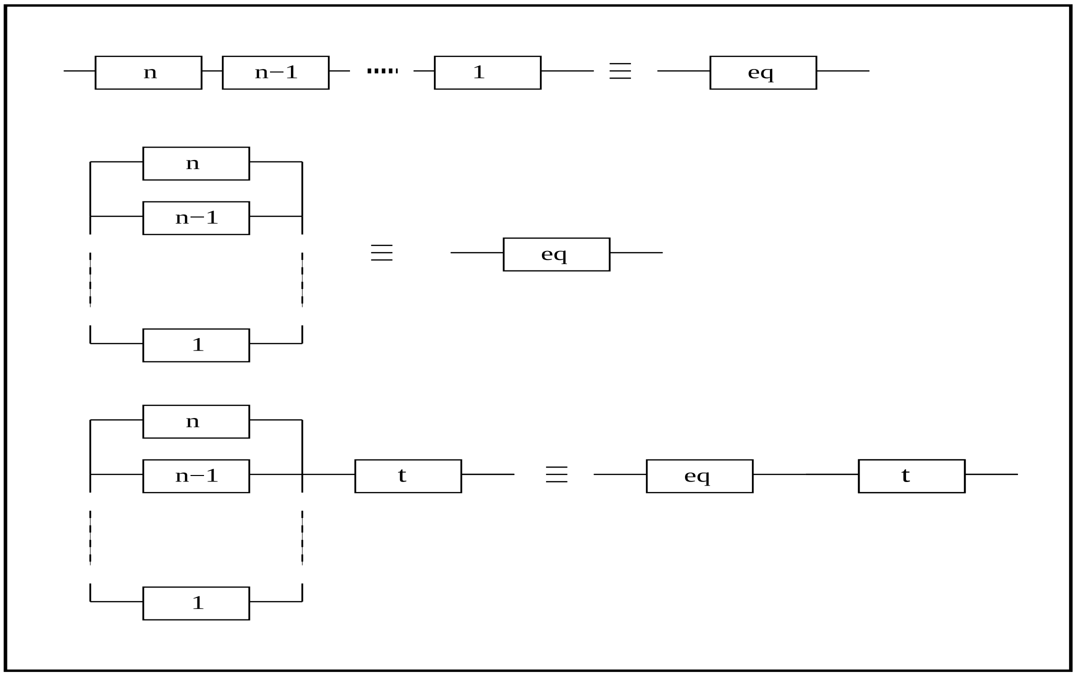

Figure 2: A typical task networks for sequential (top), parallel (middle) and parallel–sequential (bottom) cases.

$$\mathrm{EU} = -\left(1 - p_3 + p_3\,(1-p_2)\,e^{-rc_3} + p_3p_2\,(1-p_1)\,e^{-r(c_3+c_2)} + p_3p_2p_1e^{-r(V+c_3+c_2+c_1)}\right) \quad (2)$$

which equals the more suggestive form

$$\mathrm{EU} = -\left(1 - p_3 + p_3e^{-rc_3}\left(1 - p_2 + p_2e^{-rc_2}\left(1 - p_1 + p_1e^{-r(V+c_1)}\right)\right)\right)\ . \quad (3)$$

This implies that for n sequential tasks the EU follows form the *difference* equation

$$S(k) = 1 - p_k + p_ke^{-rc_k}\ S\,(k-1) \quad (4)$$

where $k = 1, 2, \cdots, n$, $S(0) = 1$ (if the n–th task is the final task of the network $S(0) = e^{-rV}$), and $\mathrm{EU}(n) = -S(n)$. As a simple application of (4) consider n task with common probability p and cost c, which gives

$$\mathrm{EU}(n) = -\left(\frac{e^{c\,r}\left(1 - p + \left(-e^{-c\,n\,r} + e^{-r\,(c\,n+V)}\right)p^n + \left(e^{-c\,n\,r} - e^{-r\,(c+c\,n+V)}\right)p^{1+n}\right)}{e^{c\,r} - p}\right)\ . \quad (5)$$

In general the probability p_i and cost c_i are different for different tasks, so (4) becomes a variable-coefficient difference equation. Then the general solution can be found as

$$\mathrm{EU}(n) = -\left(\Pi_{k=1}^{n}p_ke^{-rc_k}\ \sum_{l=1}^{n}\frac{1-p_l}{\Pi_{m=1}^{l}p_me^{-rc_m}}\right) \quad (6)$$

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

$$p_{eq} = p_n \quad and \quad c_{eq} = c_n - \frac{1}{r}\ \log\{-\mathrm{EU}(n-1)\} \tag{7}$$

In maximizing the EU, as well as in ordering the tasks, one can follow simple rules which are evident from the difference equation (4). First of all, one recalls that $p_i < 1$ and $e^{-rc_i} > 0$ so that the quantity $f_i = 1 - p_i + p_i e^{-rc_i}$ is always positive, hence the EU is always negative. Then, it is straightforward to see that to maximize EU, if there are more than one bid for the i–th task within the same time slacks, one then has to choose (p_i, c_i) among all bids which minimizes f_i. Among all bids for task i we choose the one with the lowest cost for identical probabilities, and we choose the one having highest probability for identical costs. All these observations are supported by the form of f_i.

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 $-p_i + p_i(1-p_j)e^{-rc_i} < -p_j + p_j(1-p_i)e^{-rc_j}$ for task i to precede task j.

3.2 Parallel Tasks

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

$$\mathrm{EU} = (1-p_1)p_2 e^{-rc_2} + p_1(1-p_2)e^{-rc_1} + (1-p_1)(1-p_2) + p_1 p_2 e^{-r(c_1+c_2)} \tag{8}$$

A direct evaluation of (8) reveals that the EU of n parallel tasks can be obtained by solving the difference equation:

$$P(k) = \left(1 - p_k + p_k e^{-rc_k}\right)\ P\left(k-1\right) \tag{9}$$

where $k = 1, 2, \cdots, n$, $P(0) = 1$, and $\mathrm{EU}(n) = -P(n)$. A solution of (9) reduces a groups of parallel tasks into a single unit with probability p_{eq} and cost c_{eq}:

$$p_{eq} = 1 - \Pi_{i=1}^{n}\left(1 - p_i\right) \quad and \quad c_{eq} = -\frac{1}{r}\ \log\{-\frac{\mathrm{EU}(n) + 1 - p_{eq}}{p_{eq}}\} \tag{10}$$

Then, to maximize the EU, one has to choose (p_i, c_i) among all bids for the i–th task which minimizes $f_i = 1 - p_i + p_i e^{-rc_i}$. In general, the observations made for the sequential case are also valid here.

3.3 Parallel–Serial Tasks

The configuration illustrated in the bottom of Figure 2, differs from the other two in terms of the equivalent probabilities and costs. 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:

$$\begin{aligned} p_{eq} &= 1 - \Pi_{i=1}^{n}\left(1 - p_i + p_i e^{-rc_i}\right) + \Pi_{i=1}^{n} p_i e^{-rc_i} \\ c_{eq} &= -\frac{1}{r}\ \log\{-\frac{\Pi_{i=1}^{n} p_i e^{-rc_i}}{p_{eq}}\}\,. \end{aligned} \tag{11}$$

3.4 Reduction of Hybrid Networks

In this section we will use the aforementioned rules to find the equivalent unit for a general hybrid network of tasks. In discussing the reduction of a hybrid structure we will refer to Figure 3 which shows which tasks are combined at each step via encircling them. First, one combines the sequential tasks, then the parallel ones in the middle branch, and finally the resulting sequential ones, following the rules described in last three subsections. In this picture the place where V is to be inserted is clearly the single-unit *eq* network at the very end of the reduction process.

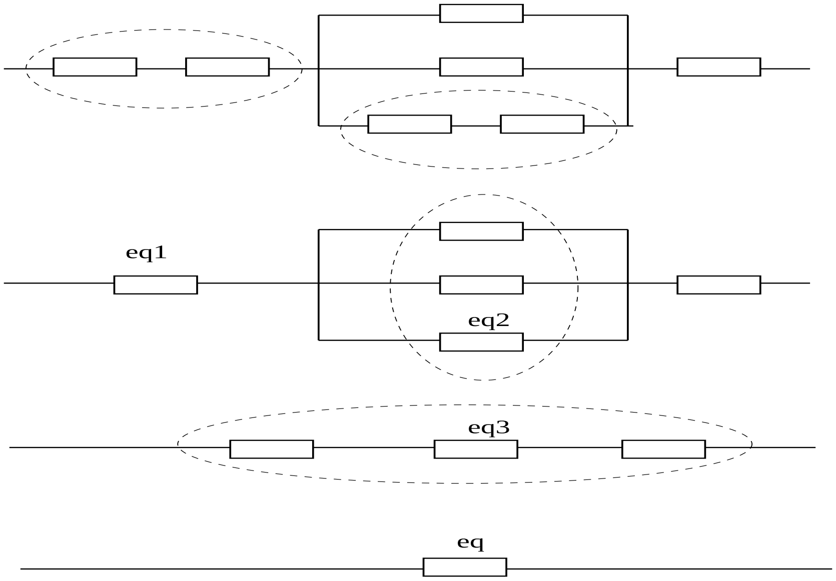

Figure 3: A sample task network for the illustration of the reduction to a single equivalent unit.

4 Winner Determination Formulation

As noted before, customer agents do a first-price, sealed-bid reverse auction. A bid includes a set of tasks and a price, along with timing data, including duration, and the earliest and latest times the task(s) may be started. Information on the probabilities the completion of the tasks can be obtained from market aggregate data.

In previous work, we have developed various algorithms for winner determination in MAGNET, including the one based on Integer Programming [Collins and Gini, 2001], IDA* using bidtree ordering [Collins *et al.*, 2002a], and Simulated Annealing. Those algorithms use cost as the main criterion for choice, and allow for combinatorial bids. Systematic studies to characterize the performance of those winner determination algorithms in terms of speed, scalability, and predictability have been reported in [Collins, 2002]. Speed and scalability are specially important, because combinatorial auction winner determination is known to be $\mathcal{NP}$-complete and inapproximable [Sandholm, 2002].

We are interested in choosing bids that maximize the customer agent Expected Utility and that satisfy the temporal and precedence constraints. 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, a bid b_i that includes the task 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.

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:

$$x_i = \begin{cases} 1 & \text{if } b_i \text{ is accepted (winner)} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Although in a combinatorial auction each bid has a total cost for the entire set of tasks, we assign the total cost to the finishing task of a bid and give zero cost to all the preceding ones. Namely, we assign costs such that $c^i_j = 0$ if the task j is not the finishing one, and $c^i_j = c_i$ if the task j is the last one (We assume that there is no payment until all tasks of a given bid finish).

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.

Each of the 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 in between the early and late start limits, 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:

$$x_i \quad \in \quad \{0, 1\} \ \forall i = 1 \cdots n \tag{13}$$

which means that a bid is either accepted or rejected.

- Start Time Limits:

$$e^i_j \leq l^i_j \leq f^i_j \ \forall j = 1 \cdots m, \ \forall i = 1 \cdots n, \tag{14}$$

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

- Coverage:

$$\sum_{i | s_j \in \mathcal{S}_i} x_i \quad = \quad 1 \ \forall j = 1 \cdots m \tag{15}$$

which means that each task must be included exactly once.

- Feasibility

$$\begin{aligned} x_i l^i_j &\geq x_{i'}(l^{i'}_{j'} + d^{i'}_{j'}) - M(1 - x_i) \\ \forall j &= \cdots m, \forall i | s_j \in \mathcal{S}_i, \forall i' | s_{j'} \in (\mathcal{S}_{i'} \cap \mathcal{P}_j) \end{aligned} \tag{16}$$

which guarantees both locally and globally that each task starts after the completion of its immediate predecessors. Here, M is a “large” number, and the last term $M(1 - x_i)$ is used to make the constraint satisfied in the case where $x_i = 0$.

The number of constraints generated by these formulas is highly variable, depending 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 [Collins and Gini, 2001].

5 Experimental Results

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. Table 1 shows these set of bids. Each row of the table corresponds to a specific bid, and includes the corresponding task, early and late start times, duration, cost, and probability of success.

The task network we are using 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 $(s_3, s_4) \prec s_5$.

Experiment 1: In this experiment we use only the bids of Set-1. Each bid includes just one task and in the winning–bid structure all tasks are sequential.

Set	Name	Task	Early Start Time	Late Start Time	Duration	Cost	Probability
1	b_1	s_1	0	3	30	2	0.9
	b_2	s_1	1	3	10	20	0.6
	b_3	s_2	12	15	10	30	0.6
	b_4	s_3	23	25	5	60	0.4
	b_5	s_4	30	35	5	35	0.8
	b_6	s_4	30	35	5	35	0.6
	b_7	s_2	12	15	10	35	0.9
	b_8	s_2	12	15	5	90	0.1
	b_9	s_3	40	45	5	60	0.4
	b_{10}	s_5	65	70	15	50	0.8
	b_{11}	s_6	75	95	5	50	0.9
2	b_{12}	s_3	32	33	6	60	0.4
3	b_{13}	s_5	65	68	12		0.8
		s_6	80	95	5	100	0.9
	b_{14}	s_4	15	18	12		0.8
		s_2	8	12	80		1
		s_3	15	17	10	50	0.9

Table 1: A bidding cycle. The corresponding time, cost, and probability values are shown in the respective columns.

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, bidder b_2 wins task s_1.
- 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 prefer between the *costly and less risky* and *cheap and more risky* situations. Here the

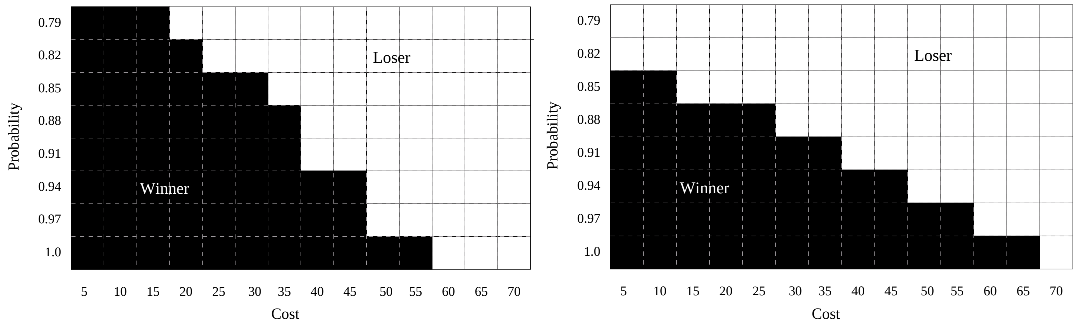

Figure 4: The winner/loser regions for b_3 with two different risk-aversion coefficient; (a) r=0.001 and (b) r=0.0001.

risk is parametrized by the *risk factor* r present in the utility function. 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.

To illustrate the situation, it is useful to perform another experiment and examining the effects of different risk factors on the winning bids. Depicted in Figure 4 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 and cheaper alternatives are preferred, whereas for $r = 0.0001$ the customer is more risk–averse and prefers more costly but more likely 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 5, 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 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 act is that the task with lower cost must be done first to minimize the total cost in case the plan 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 plan is abandoned. EU supports and realizes both of these observations.

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

Experiment 2: So far we have illustrated mainly sequential structures, where winning bids always start one after the other. Now we extend the scenario in Table 1 by adding Set-2 to Set-1. This means we have a new bid b_{12}, whose time interval coincides with one of the existing bids, but which covers a different task.

We illustrate the winner/loser regions for b_{12} in Figure 6. A comparison with Figure 5, show that the customer, when b_{12} is included, prefers to use b_{12} because of its lower costs.

Experiment 3: In this experiment, all three sets including 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

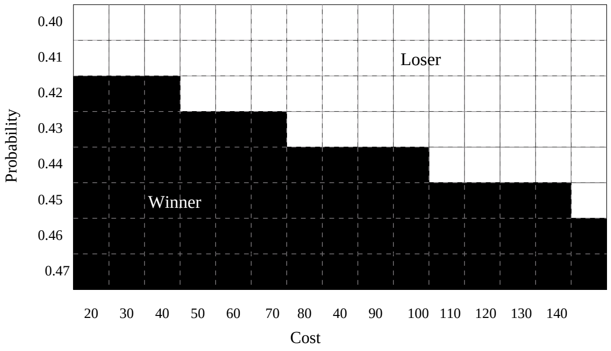

Figure 5: The winner/loser regions for b_9 with risk coefficient r=0.001

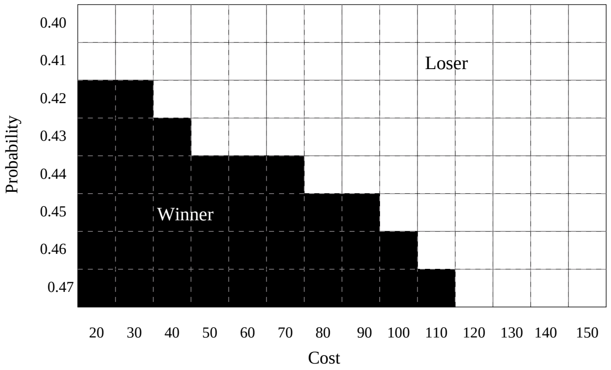

Figure 6: The winner/loser regions for b_{12}.

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 7 the EU values for each cases are shown for different costs of b_{13}. As seen from figure, for costs higher than 140 the EU of bids including b_{13} is lower than the EU of 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 7 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

$$EU = -K\left(1 - p^i_j + p^i_j e^{-rc^i_j}\left(1 - p^{i'}_{j'} + p^{i'}_{j'} e^{-r(V+c^{i'}_{j'})}\right)\right) \tag{17}$$

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. It is useful to compare the EUs of the two cases:

$$\begin{aligned} \text{LHS} &= -K\left(1 - p^{10}_5 + p^{10}_5 e^{-rc^{10}_5}\left(1 - p^{11}_6 + p^{11}_6 e^{-r(V+c^{11}_6)}\right)\right) \\ \text{RHS} &= -K\left(1 - p^{13}_5 + p^{13}_5\left(1 - p^{13}_6 + p^{13}_6 e^{-r(V+c^{13}_{5,6})}\right)\right) \end{aligned} \tag{18}$$

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 (18) for different cost and probability values.

As an example, let's take the parameters from Table 1 and vary the cost of b_{13}. For individual bids one finds $-0.3918K$. On the other hand, for combinatorial bids one gets $-0.3877K$, $-0.3910K$,

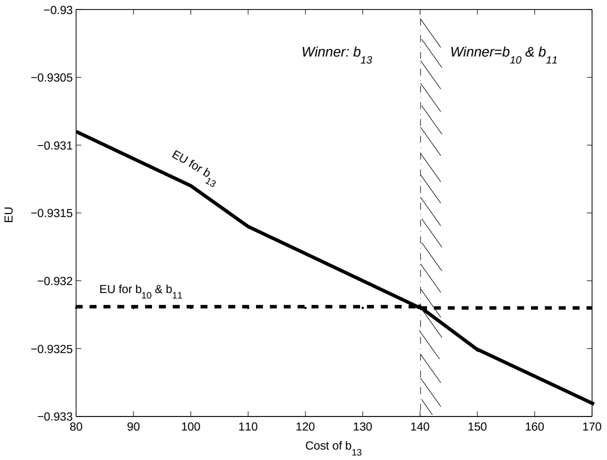

Figure 7: The EU of bid structure for combinatorial (b_{13}) vs. individual bids (b_{10} and b_{11}).

$-0.3932K$ for $c^{13}_{5,6} = 100, 130$ and 150, respectively. This simple analysis is sufficient to determine the winning bids. Indeed, the EU for the combinatorial case falls below that of the individual case at $c^{13}_{5,6} = 140$, which is in complete agreement with Figure 7. In this way, which is nothing but a restatement of the equivalent network construction, one can eliminate various bids in a simple way.

The optimization algorithm used in these experiments is a branch and bound algorithm for nonlinear mixed integer programming. The algorithm has certain limitations which deserve an explanation. It tends to fix the start time, instead of varying it between the early and late start times, if the start time happens to be in the feasible region. In principle, one might find new maxima for EU as the start time changes from the early to the late start time.

6 Related Work

Despite the abundance of work in auctions [Milgrom, 1989], limited attention has been devoted to auctions over tasks with complex time constraints and uncertainty. Execution uncertainty is studied in [Porter *et al.*, 2002] where the design of mechanisms is extended from traditional game-theoretic approaches to take into account not only cost but also probability of failure.

In [Parkes and Ungar, 2001], 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 [Wellman *et al.*, 2001], where a protocol for decentralized scheduling is proposed. The study is limited to scheduling a single resource, MAGNET agents deal with multiple resources. Walsh et al [Walsh *et al.*, 2000] 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 [Sadeh *et al.*, 1999] coordinate scheduling with the user. The problem a MAGNET agent faces is not job-shop scheduling; we are In MAGNET we are not scheduling resources the agent has, we want to produce a schedule that other agents will execute. The results reported in [Watson *et al.*, 2002] 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 [Bichler, 2001]. Unfortunately, this approach does not extend easily to time and precedence constraints.

In [Collins *et al.*, 2001] we showed how to compute the marginal expected utility of completing successfully all the tasks within the duration promised. The formulation was limited to sequential tasks, but it is the basis for the cost formulation we use in this work.

In other previous work [Babanov *et al.*, 2002] 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.

7 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 EUT 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 show 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.

References

[Babanov *et al.*, 2002] Alexander Babanov, John Collins, and Maria Gini. Risk and expectations in a-priori time allocation in multi-agent contracting. In *Proc. of the First Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, volume 1, pages 53–60, Bologna, Italy, July 2002.

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

[Biswas, 1997] Tapan Biswas. *Decision-Making Under Uncertainty.* St. Martin's Press, Inc., 1997.

[Collins and Gini, 2001] John Collins and Maria Gini. An integer programming formulation of the bid evaluation problem for coordinated tasks. In Brenda Dietrich and Rakesh 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.

[Collins *et al.*, 2001] John Collins, Corey Bilot, Maria Gini, and Bamshad Mobasher. Decision processes in agent-based automated contracting. *IEEE Internet Computing*, pages 61–72, March 2001.

[Collins *et al.*, 2002a] John Collins, Güleser Demir, and Maria Gini. Bidtree ordering in IDA* combinatorial auction winner-determination with side constraints. In J. Padget, Onn Shehory, David Parkes, Norman Sadeh, and William Walsh, editors, *Agent Mediated Electronic Commerce IV*, volume LNAI2531, pages 17–33. Springer-Verlag, 2002.

[Collins *et al.*, 2002b] John Collins, Wolfgang Ketter, and Maria 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.

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

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

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

[Parkes and Ungar, 2001] David C. Parkes and Lyle 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.

[Porter *et al.*, 2002] Ryan Porter, Amir Ronen, Yoav Shoham, and Moshe Tennenholtz. Mechanism design with execution uncertainty. In *UAI-02*, 2002.

[Sadeh *et al.*, 1999] Norman M. Sadeh, David W. Hildum, Dag Kjenstad, and Allen 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.

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

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

[Walsh *et al.*, 2000] William E. Walsh, Michael Wellman, and Fredrik Ygge. Combinatorial auctions for supply chain formation. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, October 2000.

[Watson *et al.*, 2002] Jean Paul Watson, J. Christopher Beck, Adele Howe, and L. Darrell Whitley. Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence*, 2002.

[Wellman *et al.*, 2001] 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.