Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 02-034

Asking the Right Question: Risk and Expectation in Multi-Agent Contracting

Alexander Babanov, John Collins, and Maria Gini

November 05, 2002

Asking the Right Question: Risk and Expectation in Multi-Agent Contracting

Alexander Babanov, John Collins, and Maria Gini
Department of Computer Science and Engineering
University of Minnesota

Keywords: Automated auctions, multi-agent contracting, expected utility, risk estimation, optimization

Abstract

This paper investigates methods of reducing risk in market-based auctions of tasks with complex time constraints and interdependencies. The research addresses problems in a contracting setting in which a buyer has a set of tasks to be performed. Because of the complex dependencies among the tasks, a task not completed on time might have devastating effect on other tasks. Therefore, the problem is to sequence tasks and allocate time windows to maximize the expected utility of the agent. Because there is a probability of loss as well as a probability of gain, the decision process must deal with the risk posture of the person or organization on whose behalf the decision maker is acting.

1 Introduction

E-commerce technology has the potential to benefit society by reducing the cost of buying and selling and by opening new market opportunities. The main domain of application we envision is the management of agile and dynamic supply-chains, an area in which the potential payoff is high, given the projected size of the business-to-business and make-to-order e-commerce markets.

More production processes are being outsourced to outside contractors, making supply chains longer and more convoluted. This increased complexity is compounded by increasing competitive pressure, and accelerated production schedules which demand tight integration of all processes.

Finding potential suppliers is only a step in the process of producing goods. Time dependencies among operations make scheduling a major factor. A late delivery of a part might produce a cascade of devastating effects. Unfortunately, current auction-based systems do not have any notion of time. Handling auctions for tasks with time constraints is beyond the capabilities of current e-commerce systems.

We present the results of a study of how an autonomous agent can maximize its profits while predicting and managing its financial risk exposure when requesting bids for tasks with complex time constraints. We show how this can be done by specifying appropriate time windows for tasks when soliciting bids, and by using received bids effectively in building a final work schedule.

This study is a part of the MAGNET (Multi-AGent NEgotiation Testbed) research project. MAGNET agents participate in first-price, sealed-bid combinatorial auctions over collections of tasks with precedence relations and time constraints. MAGNET promises to increase the efficiency of current markets by shifting much of the burden of market exploration, auction handling, and preliminary decision analysis from human decision makers to a network of heterogeneous agents.

We distinguish between two agent *roles*, the *Customer* and the *Supplier* (see Figure 1). A customer has a set of tasks to be performed, with complex dependencies among the tasks, and solicits resources from suppliers by presenting a *Request for Quotes* (RFQs) through an agent-mediated market. Supplier agents may offer to provide the requested resources or services, for specified prices, over specified time periods. Once the customer receives bids, it evaluates them and selects the optimal set of bids. This paper deals with decision problems in the Bid Manager component of the Customer Agent.

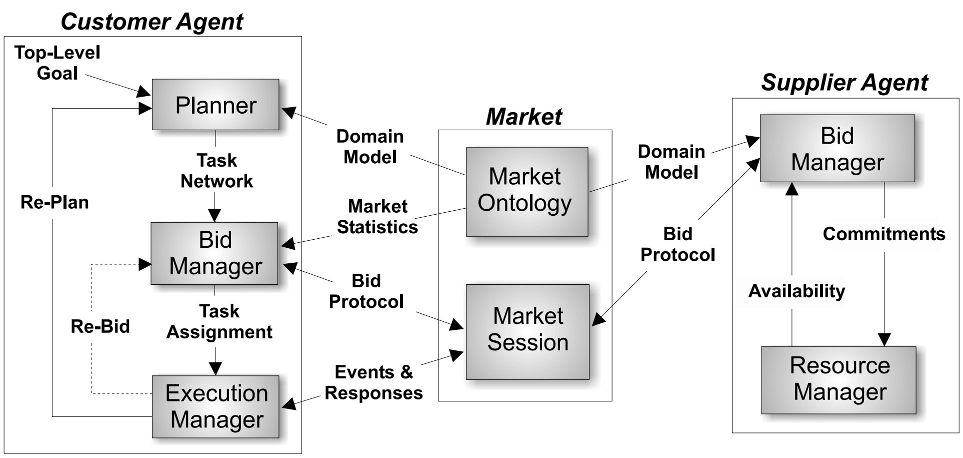


Figure 1: The MAGNET architecture

This is a schematic outline of the main interactions among agents:

- A customer issues an RFQ which specifies tasks, their precedence relations, and a timeline for the bidding process. For each task, a time window is specified giving the earliest time the task can start and the latest time the task can end.
- Suppliers submit bids. A bid includes a set of tasks, a price, a portion of the price to be paid as a non-refundable deposit, and estimated duration and time window data that reflect supplier resource availability and constrain the customer's scheduling process.
- The customer decides which bids to accept. Each task needs to be mapped to exactly one bid (i.e. no free disposal [22]), and the constraints of all awarded bids must be satisfied in the final work schedule.
- When the customer awards a bid, it pays a deposit and specifies the work schedule.
- When the supplier completes a task, the customer pays the remainder of the price.
- If the supplier fails to complete a task, the price is forfeit and the deposit must be returned to the customer. A penalty may also be levied for non-performance, or a leveled-commitment protocol [35] may be used.

1.1 A Motivating Example

As an example, imagine that we need to construct a garage. Figure 2 shows the tasks needed to complete the construction. The tasks are represented in a *task network*, where links indicate precedence constraints. The first decision we are faced with is how to sequence the tasks in the RFQ and how much time to allocate to each of them. For instance, we could reduce the number of parallel tasks, allocate more time to tasks with higher variability in duration or tasks for which there is a shortage of laborers, or allow more slack time.

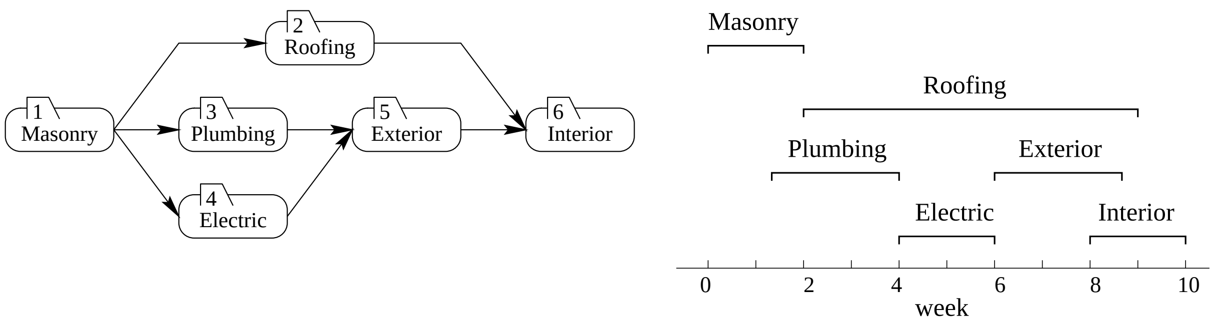


Figure 2: A task network example and the corresponding RFQ.

A sample RFQ is shown in Figure 2. Note that the time windows in the RFQ do not need to obey the precedence constraints; the only requirement is that the accepted bids obey them. We assume that the supplier is more likely to bid, and submit a lower-cost bid, if it is given a greater flexibility in scheduling its resources.

1.2 Experiences and Open Issues

We have shown [3] that the time constraints specified in the RFQ can affect the customer's outcome in two major ways:

1. by affecting the number, price, and time windows of bids. We assume that bids will reflect supplier resource commitments, and therefore larger time windows will result in more bids and better utilization of resources, in turn leading to lower prices [5]. However, an RFQ with overlapping time windows makes the process of winner determination more complex [4]. Another less obvious problem is that every extra bid over the minimum needed to cover all tasks adds one more rejected bid. Ultimately, a large percentage of rejections will reduce the customer agent's credibility, which, after repeated interactions in the market, will result in fewer bids and/or higher costs.
2. by affecting the financial exposure of the customer agent [3]. We assume non-refundable deposits are paid to secure awarded bids, and payments for each task are made as the tasks are completed. The payoff for the customer occurs only at the completion of all the tasks. Once a task is completed in the time period specified, the customer is liable for its full cost, regardless of whether in the meantime other tasks have failed. If a task is not completed by the supplier, the customer is not liable for its cost, but this failure can ruin other parts of the plan. Slack in the schedule increases the probability that tasks will be completed or that there will be enough time to recover if any fail. However, slack

extends the completion time and so reduces the payoff. In made-to-order products the speed is the key; the value of the final payoff may drop off very steeply with time.

Because there is a probability of loss as well as a probability of gain, we must deal with the risk posture of the person or organization on whose behalf the agent is acting. We need a principled method for generating RFQs that models and makes effective use of this risk posture, using both experience and available market information.

2 Expected Utility Approach to Generating Optimal RFQ

In this section we describe a new approach to the construction of optimal RFQs that employs the Expected Utility Theory to reduce the likelihood of receiving unattractive bids, while maximizing the number of bids that are likely to be awarded. We pay special attention to the relation between the size of RFQ time windows and the number of expected bids by investigating the balance between the quantity and the quality of expected bids.

2.1 Terminology

A *task network* (see Figure 2) is a tuple $\langle N, \prec \rangle$ of a set N of individual tasks and strict partial ordering on them. We also use N to denote the number of tasks where appropriate.

A task network is characterized by a *start time* t^{s} and a *finish time* t^{f}, which delimit the interval of time when tasks can be scheduled. The placement of task n in the schedule is characterized by *task n start time* t_n^{s} and *task n finish time* t_n^{f}, subject to the following constraints:

$$t^{\mathrm{s}} \leq t_m^{\mathrm{f}} \leq t_n^{\mathrm{s}}, \quad \forall m \in P_1(n) \quad \text{and} \quad t_n^{\mathrm{f}} \leq t_m^{\mathrm{s}} \leq t^{\mathrm{f}} \quad \forall m \in S_1(n)$$

where $P_1(n)$ is the a set of *immediate predecessors* of n, $P_1(n) = \{m \in N \mid m \prec n, \nexists m' \in N, m \prec m' \prec n\}$. $S_1(n)$ is defined similarly to be the set of *immediate successors* of task n.

The *probability of task n completion* by time t, conditional on the ultimate successful completion of task n, is distributed according to the cumulative distribution function (CDF) $\Phi_n = \Phi_n(t_n^{\mathrm{s}}; t), \Phi_n(\cdot; \infty) = 1$. Observe that Φ_n is defined to be explicitly dependent on the start time t_n^{s}. To see the rationale, consider the probability of successful mail delivery in x days for packages that were mailed on different days of a week.

There is an associated unconditional *probability of success* $p_n \in [0, 1]$ characterizing the percentage of tasks that are successfully completed given infinite time (see Figure 3).

Task n bears an associated *cost*[1]. We assume the total cost of task n has two parts: a deposit, which is paid when the bid is accepted, and a cost c_n which is due some time after successful completion of n. In this analysis we will not compare plans with different deposits, so we assume without loss of generality the deposit to be 0.

There is a single *final reward* V scheduled at the plan finish time t^{f} and paid conditional on all tasks in N being successfully completed by that time.

[1]Hereafter we use words "cost" and "reward" to denote some monetary value, while referring the same value as "payoff" or "payment" whenever it is scheduled at some time t.

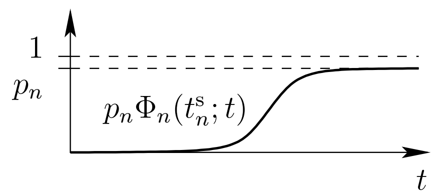


Figure 3: Unconditional distribution for successful completion probability.

There is an associated *rate of return* ${q_n}^2$ that is used to calculate the *discounted present value* (PV) for payoff c_n due at time t as

$$\mathrm{PV}\left(c_n;t\right) := c_n\left(1+q_n\right)^{-t}.$$

We associate the return q with the final reward V.

2.2 Expected Utility and Certainty Equivalent

We represent the customer agent's preferences over payoffs by the von Neumann-Morgenstern utility function u [19]. We further assume that the absolute risk-aversion coefficient $r := -u''/u'$ of u is constant for any value of its argument, hence u can be represented as follows:

$$u\left(x\right) = -\exp\left\{-rx\right\} \text{ for } r \neq 0 \quad \text{and} \quad u\left(x\right) = x \text{ for } r = 0$$

A *gamble* is a set of payoff-probability pairs $G = \left\{\left(x_i,p_i\right)_i\right\}$ s.t. $p_i > 0, \forall i$ and $\sum_i p_i = 1$. The expectation of the utility function over a gamble G is the *expected utility* (EU):

$$Eu\left[G\right] := \sum_{(x_i,p_i)\in G} p_i u\left(x_i\right)$$

The *certainty equivalent* (CE) of a gamble G is defined as the single payoff value whose utility matches the expected utility of the entire gamble G, i.e. $u\left(\mathrm{CE}\left[G\right]\right) := Eu\left[G\right]$. Hence under our assumptions

$$\mathrm{CE}(G) = \frac{-1}{r}\log\sum_{(x_i,p_i)\in G} p_i\exp\left\{-rx_i\right\} \text{ for } r\neq 0 \quad \text{and} \quad \mathrm{CE}(G) = \sum_{(x_i,p_i)\in G} p_i x_i \text{ for } r = 0$$

Naturally, the agent will not be willing to accept gambles with negative certainty equivalent, and the higher values of the certainty equivalent will correspond to more attractive gambles.

To illustrate the concept, Figure 4 shows how the certainty equivalent depends on the risk-aversity r of an agent. In this figure we consider a gamble that brings the agent either 100 or nothing with equal probabilities. Agents with positive r's are risk-averse; those with negative r's are risk-loving. Agents with risk-aversity close to zero, i.e. almost risk-neutral, have a CE equal to its weighted mean 50.

[2]The reason for having multiple q_n's is that individual tasks can be financed from different sources, thus affecting task scheduling.

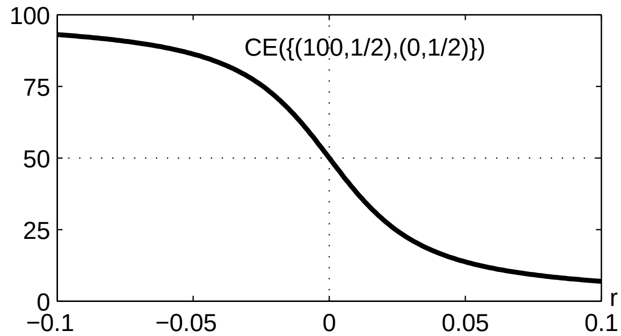


Figure 4: Certainty equivalent of a simple gamble as a function of the risk-aversity.

2.3 Cumulative Probabilities

To compute the certainty equivalent of a gamble we need to determine a schedule for the tasks and compute the payoff probability pairs.

We assume that the payoff c_n for task n is scheduled at t_n^{f}, so its present value $\tilde{c}_n$[3] is

$$\tilde{c}_n := c_n \left(1 + q_n\right)^{-t_n^{\mathrm{f}}}$$

We define the conditional probability of task n success as

$$\tilde{p}_n := p_n \Phi_n \left(t_n^{\mathrm{s}}; t_n^{\mathrm{f}}\right).$$

We also define the *precursors* of task n as a set of tasks that finish before task n starts in a schedule, i.e.

$$\tilde{P}\left(n\right) := \left\{m \in N | t_m^{\mathrm{f}} \leq t_n^{\mathrm{s}}\right\}.$$

The unconditional probability that task n will be completed successfully is

$$\tilde{p}_n^{\mathrm{c}} = \tilde{p}_n \times \prod_{m \in \tilde{P}(n)} \tilde{p}_m.$$

That is, the probability of successful completion of every precursor and of task n itself are considered independent events. The reason this is calculated in such form is because, if any task in $\tilde{P}\left(n\right)$ fails to be completed, there is no need to execute task n.

The probability of receiving the final reward V is therefore

$$\tilde{p} = \prod_{n \in N} \tilde{p}_n.$$

[3]Hereafter we "wiggle" variables that depend on the current task schedule, while omitting all corresponding indices for the sake of simplicity.

2.4 Example and Discussion

To illustrate the definitions above, let's return to the task network in Figure 2 and consider the sample task schedules shown in Figure 5. In this figure the x-axis is time, the y-axis shows both the task numbers and the cumulative distribution of the unconditional probability of completion (compare to Figure 3). Circle markers show start times t_n^{s}. Crosses indicate both finish times t_n^{f} and success probabilities $\tilde{p}_n$ (numbers next to each point). Square markers denote that the corresponding task cannot span past this point due to precedence constraints. The thick part of each CDF shows the time allocated to each task.

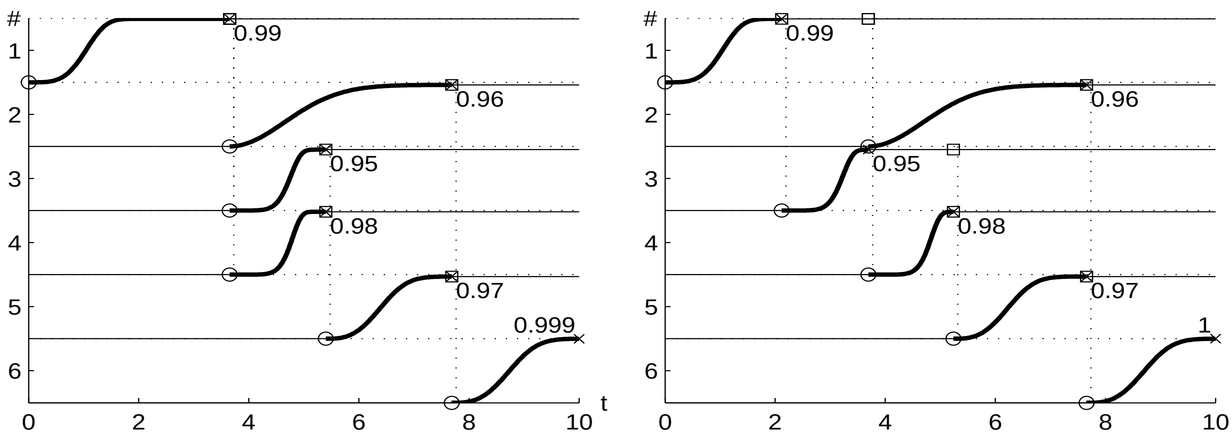


Figure 5: CE maximizing time allocations for the plan in Figure 2 for $r = -0.01$ (left) and $r = 0.02$ (right).

The customer agent needs a way of collecting the market information necessary to build and use the model. The probability of success is relatively easy to observe in the market. This is the reason for introducing the cumulative probability of success Φ_n and probability of success p_n, instead of the average project life span or probability of failure. Indeed, it is rational for the supplier to report a successful completion immediately in order to maximize the present value of a payment. Also it is rational not to report a failure until the last possible moment due to a possibility of earning the payment by rescheduling, outsourcing or somehow else fixing the problem.

2.5 Gamble Calculation Algorithm and Maximization

Given a schedule, like the one shown in Figure 5, we need to compute the payoff probability and then maximize the CE for the gamble. Writing an explicit description of the expected utility as a function of gambles is overly complicated and relies on the order of task completions. Instead we propose a simple recursive algorithm that creates these gambles. We then maximize the CE over the space of gambles. The proposed algorithm does not depend on the structure of the task network, but on the number of tasks scheduled in parallel.

Algorithm: $G \leftarrow$ **calcGamble**(T, D)
    Requires: T "*tasks to process*", D "*processed tasks*"

Returns: G *"subtree gamble"*

$M \leftarrow \{m \in T | \tilde{P}(m) \subset D\}$
if $M \neq \varnothing$ *"it's a branch"*
    $n \leftarrow$ ***first***$\{M\}$ *"according to some ordering"*
    $T \leftarrow T \setminus \{n\}$
    $G \leftarrow \varnothing$
    $E \leftarrow$ **calcGamble**(T, D) *"follow* $\ldots \rightarrow \bar{n}$ *path"*
    forall $(x, p) \in E$
        $G \leftarrow G \cup \{(x, p \times (1 - \tilde{p}_n)\})$
    endfor
    $I \leftarrow$ **calcGamble**$(T, D \cup \{n\})$ *"follow* $\ldots \rightarrow n$ *path"*
    forall $(x, p) \in I$
        $G \leftarrow G \cup \{(x + \tilde{c}_n, p \times \tilde{p}_n)\}$
    endfor
    return G *"subtree is processed"*
else *"it's a leaf"*
    if $N = D$ *"all tasks are done"*
        return $\{(V, 1)\}$
    else *"some task failed"*
        return $\{(0, 1)\}$
    endif
endif

In the first call, the algorithm receives a "todo" task list $T = N$ and a "done" task list $D = \varnothing$. All the subsequent calls are recursive. To illustrate the idea behind this algorithm, we refer to the payoff-probability tree in Figure 6. This tree was built for the time allocations in Figure 5 (right) and reflects the precursor relations for this case.

Looking at the time allocation, we note that with probability $1 - \tilde{p}_1$ task 1 fails, the customer agent does not pay or receive anything and stops the execution (path $\bar{1}$ in the tree). With probability $\tilde{p}_1^c = \tilde{p}_1$ the agent proceeds with task 3 (path 1 in the tree). In turn, task 3 either fails with probability $\tilde{p}_1 \times (1 - \tilde{p}_3)$, in which case the agent ends up stopping the plan and paying a total of c_1 (path $1 \rightarrow \bar{3}$), or it is completed with the corresponding probability $\tilde{p}_3^c = \tilde{p}_1 \times \tilde{p}_3$. In the case where both 1 and 3 are completed, the agent starts both 2 and 4 in parallel and becomes liable for paying c_2 and c_4 respectively even if the other task fails (paths $1 \rightarrow 3 \rightarrow 2 \rightarrow \bar{4}$ and $1 \rightarrow 3 \rightarrow \bar{2} \rightarrow 4$). If both 2 and 4 fail, the resulting path in the tree is $1 \rightarrow 3 \rightarrow \bar{2} \rightarrow \bar{4}$ and the corresponding payoff-probability pair is framed in the figure.

The algorithm's complexity is $O\left(2^{K-1} \times N\right)$, where K is the maximum number of tasks that are scheduled to be executed in parallel. Reducing the complexity of **calcGamble** is critical, since it is in the inner loop of the CE maximization process. In commercial projects the ratio K/N is likely to be low, since not many of these exhibit a high degree of parallelism. Our preliminary experiments allow us to conclude that the K/N ratio is lower for risk-averse agents (presumably, businessmen) than for risk-lovers (gamblers). These two considerations may reduce the need for a faster algorithm, though additional work to improve the algorithm is planned.

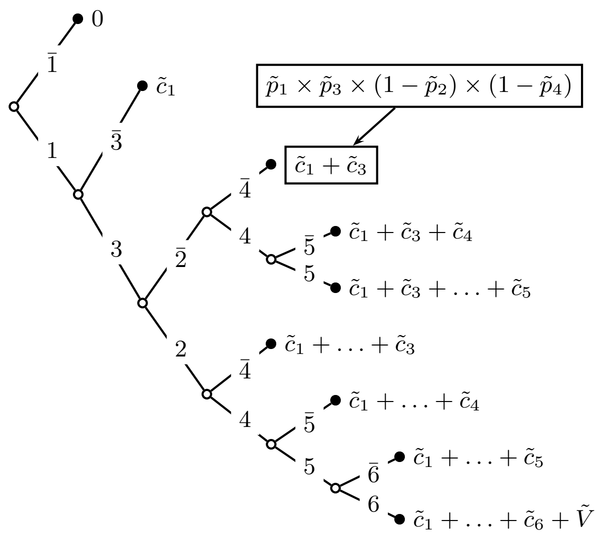


Figure 6: Payoff-probability tree for the time allocations in Figure 5 (right).

2.6 Preliminary Experimental Results

We have conducted a preliminary set of experiments on CE maximization. Some of the results are summarized in Figure 7. In this figure, the y-axis shows 11 different risk-aversity r settings, the bottom x-axis is time t in the plan, and the top x-axis shows maximum CE value for each r setting. The rounded horizontal bars in each of 11 sections denote time allocations for each of six tasks with task 1 being on top. Sections $r = -0.01$ and $r = 0.02$ correspond to Figure 5 (left) and Figure 5 (right) respectively. Finally, the vertical bars show the maximum CE values.

Let's examine the relative placement of time allocations as a function of r. For this purpose we highlighted task 3 (black bars) and task 4 (white bars). Here task 3 has higher variance of CDF and lower probability of success than task 4 (0.032 and 0.95 vs. 0.026 and 0.98). Task 3 is also more expensive (-15 vs. -7). There are four different cases in the experimental data:

1. Risk-loving agents tend to schedule tasks in parallel and late in time in order to maximize the present value of the expected difference between reward and payoffs to suppliers. This confirms the intuition from Figure 4 – risk-lovers lean toward receiving high, risky payoffs rather than low certain payoffs.
2. Risk neutral and minimally risk-averse agents place risky task 3 first to make sure that the failure doesn't happen too far into the project. Note, that they still keep task 2 in parallel, so in case 2 fails, they are liable for paying the supplier of task 4 on success. One can consider those agents as somewhat optimistic.
3. Moderately risk-averse agents try to dodge the situation above by scheduling task 3 after task 2 is finished. These agents are willing to accept the plan, but their expectations are quite pessimistic.
4. Highly risk-averse agents shrink task 1 interval to zero, thus "cheating" to avoid covering any costs. One may interpret this as a way of signaling a refusal to accept the plan.

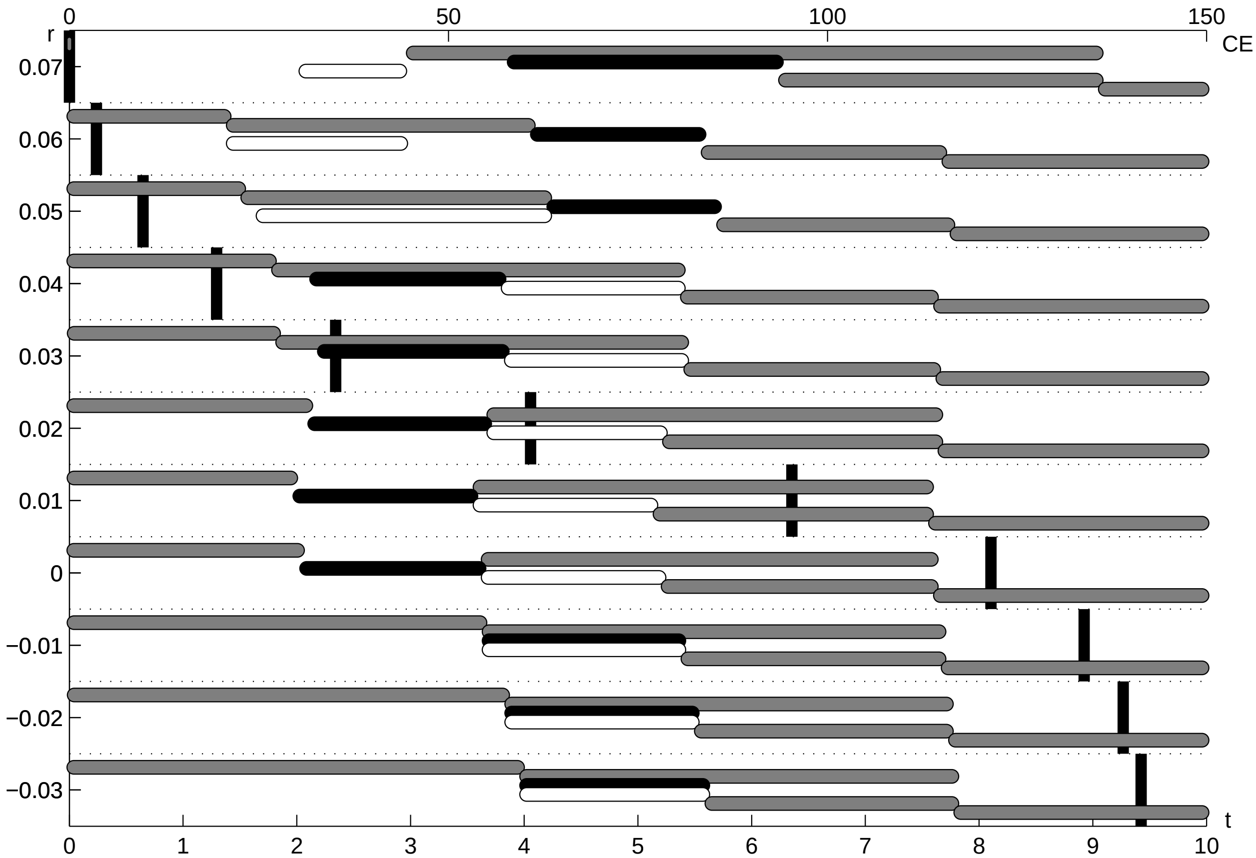


Figure 7: CE maximizing schedules and CE values for the plan in Figure 2 and $r \in [-0.03, 0.07]$.

2.7 RFQ generation

In the previous section we have shown a way of generating a CE maximizing schedule of task execution, which we hereafter refer as the *ideal schedule*. The ideal schedule insures the highest possible quality of the bids that satisfy it, where by quality we assume some function of the expectations over the cost, the probability of successful completion, and the profitability of the incoming bids in their feasible combinations with other bids. At the same time it cannot serve as the optimal RFQ, since it is unlikely that bids will be available to cover precisely the same intervals as mandated by the CE maximizing schedule.

In order to construct a viable RFQ based on the CE maximizing schedule, the customer agent should lower its expectations of the bid quality to some level by widening the RFQ time windows around the ideal ones, thus increasing[4] the expected number of the incoming bids. In this section we discuss criteria that allow us to rationalize the selection among all such RFQs.

[4]At least to some extent, — there is a fair chance that the number of the incoming bids will cease to increase whenever RFQ time windows become too large to inspire confidence on the part of suppliers.

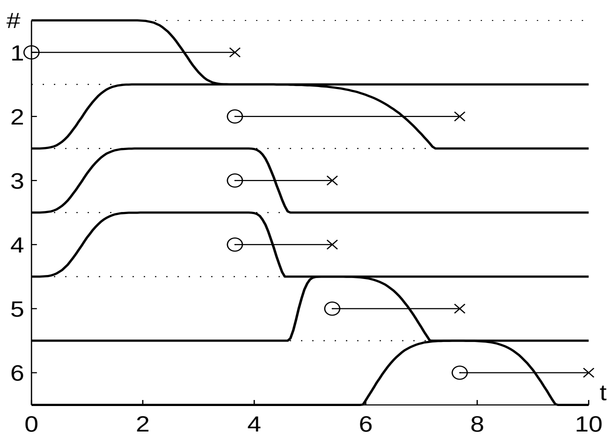
We approach the optimal RFQ generation based on the ideal schedule as follows:

1. Measure the sensitivity of the expected bid quality to the deviations from the CE maximizing schedule.
2. Derive the relation between the quality of incoming bids and the size of RFQ time windows.
3. Decide on the choice of the optimal quality-quantity combination.

Note, the term "optimality" as we apply it to the choice of the customer agent can be expressed in other words as "individually rational and comprehensible." That is, we search for the solution concept that generates viable RFQs *and* is legible enough for a human user of the system.

2.7.1 CE sensitivity to schedule changes

We propose measuring the sensitivity of CE by investigating how CE values change with variations of a single task n start time t_n^{s} in the ideal schedule. For the sake of brevity the resulting dependency of CE values is denoted by $\mathrm{CE}\left(t_n^{\mathrm{s}}\right)$. Figure 8 shows $\mathrm{CE}\left(t_n^{\mathrm{s}}\right), n = 1 \ldots 6$ for our 6-task sample problem for $r = -0.01$ and $r = 0.02$ respectively. In the figure y-axis of each horizontal stripe n represents the percentage of the maximum CE value, x-axis represents t_n^{s} and the horizontal lines with circle and cross ends show the corresponding ideal schedules.



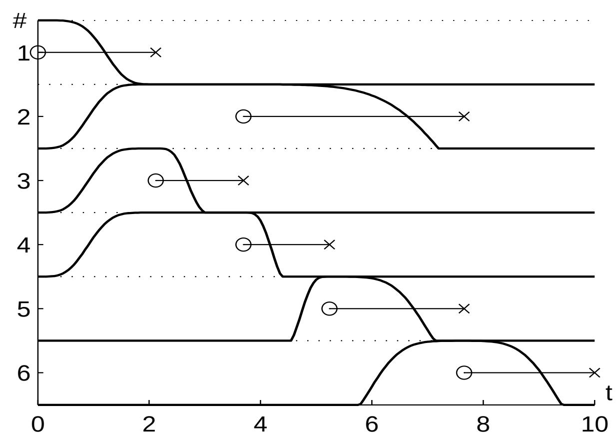


Figure 8: $\mathrm{CE}\left(t_n^{\mathrm{s}}\right)$ graphs for the corresponding ideal schedules in Figure 5.

The tasks 1, 3 and 5 in the right graph are relatively restrictive to the start times of the bids that can be bundled with the ideal bids without considerably impairing the resulting bundle's value. However, the fact that the task 2 in the right graph is more flexible does not guarantee that it will attract a higher number of bids, since the latter depends both on the size of the corresponding time window and on the market properties of the task: resource availability, number of prospective bidders, seasonal changes, etc.

We assert that for the purpose of creating an optimal RFQ it is admissible to choose time windows based on the sensitivity of CE to deviations of a single time restriction from the ideal schedule. The rationale is that the relations between tasks are already encapsulated in the calculations of CE, so the change of one restriction will approximate the rescheduling of several related tasks in the neighborhood of the ideal schedule.

2.7.2 Quality vs. quantity

Observe, that the time window for the task n, $\{t_n^{\mathrm{s}} | \mathrm{CE}\,(t_n^{\mathrm{s}}) \geq x\}$, grows as the lowest expected CE value, x, decreases. The relation between these two variables for the tasks 3 and 4 of the test problem is shown in Figure 9. The corresponding relation between the lowest expected CE value and the expected number of bids as a function of the window size is shown in Figure 10. In the last graph we assumed, for the sake of example, that the supply of the task 3 is higher than of the task 4, hence the difference in relative positions of task 3 and task 4 graphs in the two figures.

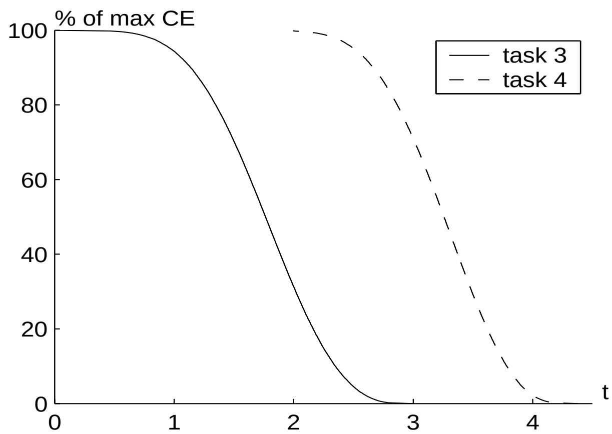


Figure 9: Relationship between the RFQ window size (shown in units of time on x-axis) and the lowest admissible percentage of the maximum CE value.

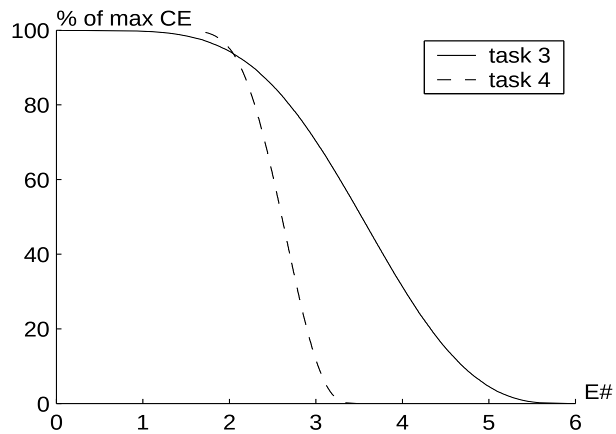


Figure 10: Relationship between the expected number of bids (shown on x-axis) and the lowest admissible percentage of the maximum CE value.

The type of graph in Figure 10 reflects the relation between the quality and the quantity of bids we were searching for. Indeed, the only independent variable in this graph is t_n^{s}. The quantity of bids depends on the size and positions of RFQ time windows that, in turn, depend on the decision about the lowest admissible CE value. The quality of bids is a function of the RFQ choice and the properties of the plan. Finally, it is expected that the customer agent will prefer a point on the graph to any point below and to the left of it, hence the optimal choice should lie on the graph.

2.7.3 Optimal quality-quantity choice and RFQ

We illustrate the decision process of the customer agent in Figure 11, where the customer agent's preferences over quality-quantity combinations are represented by a family of indifference curves and the graph of underlying quality-quantity relationship derived in the text above[5]. Each indifference curve shows quality-quantity

[5]The precise derivation of the solution requires introduction of several concepts from economic theory that span beyond the scope of this paper.

pairs that are equivalent from the agent's point of view. The optimal choice belongs to the intersection of the quality-quantity graph and the highest indifference curve (shown as solid line in the graph).

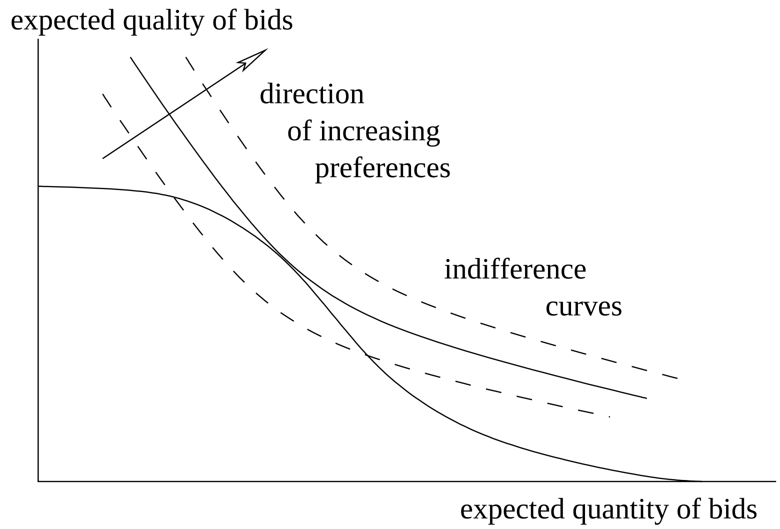


Figure 11: Quality-quantity graph with three indifference curves.

After the optimal choice of the quality-quantity combinations for all tasks in the plan is revealed, we proceed with constructing the optimal RFQ time windows. The choice of early start time t_n^{es} and late start time t_n^{ls} are determined by the value of the reciprocal of the $\text{CE}\,(t_n^{\text{s}})$ at the minimum admissible CE choice for the task n. The late finish time t_n^{lf} is chosen to be at the ideal time window length distance from t_n^{ls}. Figure 12 shows two sample RFQs for the garage building example. In the figure gray bars show start time intervals, $[t_n^{\text{es}}, t_n^{\text{ls}}]$, the ends of white bars correspond to late finish times, t_n^{lf} and the horizontal lines with circle and cross ends show two corresponding ideal schedules.

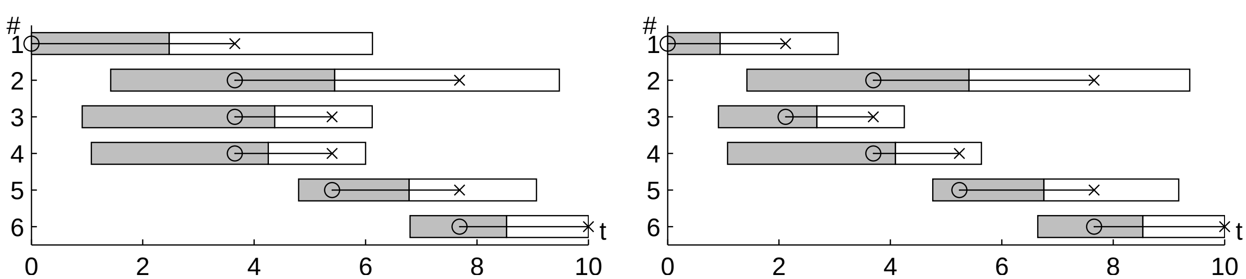


Figure 12: Sample RFQs for the corresponding ideal schedules in Figure 5 and the following vector of the maximum CE percentages: $(80\%, 95\%, 50\%, 70\%, 50\%, 90\%)$.

Our choice of the RFQ may not be optimal in the quantitative sense, however it is individually rational for the customer agent, it is also fast to compute, and arguably easy to grasp for a human user of the system. It should be emphasized here that the choice of the RFQ is based on the uncertain market information, hence the quantitatively optimal solution is itself a compromise. In the second part of the next section we address the issue of testing the efficiency of the described RFQ generation algorithm.

3 Open Issues and Further Research

In this section we outline two major issues that arise when we employ the expected utility approach to generate the optimal RFQ. The first issue concerns the CE maximization in the domain with temporal and precedence constraints. The second issue is the assessment of the EU approach and, ultimately, the MAGNET system itself in the absence of the real-world data for the domain of interest.

3.1 Multiple Local Maxima

One of the most important issues related to the CE maximization is the presence of multiple local maxima of CE even in cases where task networks are fairly simple. We argue that this property is partially due to the relative positioning of the tasks off the critical path. Any two tasks that are not ordered by the precedence constraints can be scheduled in three ways: parallel and two sequential. Scheduling tasks in parallel increases the probability of successful completion, while sequential scheduling minimizes overall payments, in case one of the tasks fails. In cases where extra slack allows for sequential scheduling, it turns out that parallel and sequential positionings of two independent individual tasks lead to similar resulting CE values.

To illustrate the issue, we constructed a sample task network with two parallel tasks. Task 1 has a higher variance of completion time probability and lower probability of success than task 2, everything else is the same. The resulting graph of CE is shown in Figure 13. There are 3 local maxima in this figure: one in the left side that corresponds to the task two being scheduled first in sequential order, another on the right side corresponding to the task one being first, and yet another one in the furthermost corner of the graph representing both tasks being scheduled at time 0 and executed in parallel. The number of local maxima grows considerably with the number of the tasks that are not restricted by the precedence relationship.

3.1.1 Domain and Algorithm Properties

The following list shows the properties of the domain that influence the search algorithm design:

- local maxima are due to different scheduling order of tasks off the critical path;
- groups of local maxima have similar CE values;
- an RFQ based on the global maximum can be overly restrictive.

The properties of the domain frame the properties of the search algorithm that we design to fit this domain. Namely, the search algorithm must be able to test different orderings of tasks, it should know how to explore groups of similar local maxima and, whenever possible, it should provide alternative schedules with CE values close to the global maximum.

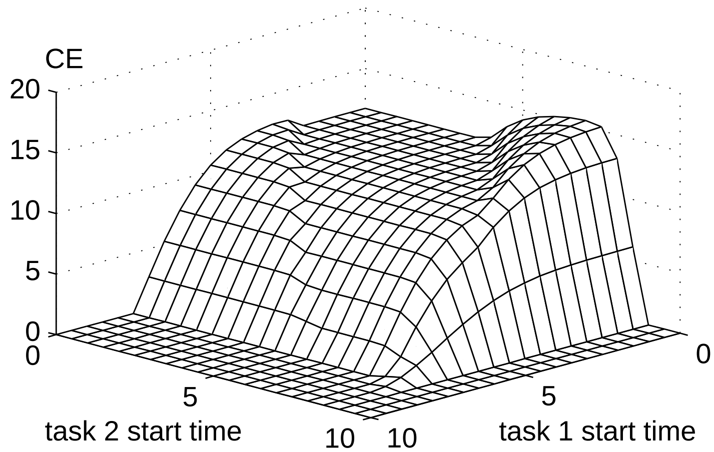


Figure 13: Local maxima for two parallel tasks.

We propose a search algorithm based on the ideas of the Simulated Annealing [27] and Genetic Algorithms [7]. The algorithm will combine the stochastic temperature-driven nature of the Simulated Annealing with the simultaneous search space exploration of the Genetic Algorithms. In this section we describe the proposed algorithm in more details and explain the rationale of its design.

3.1.2 Search Algorithm

The proposed search algorithm explores several alternative schedules in parallel. The initial set of alternatives can be generated in many ways: random generation, hill-climbing from random schedule, CPM, etc. The execution of the algorithm proceeds in steps by randomly applying one of the following five transformation rules to each alternative schedule. Figure 14 illustrates the algorithm for the case of three pairwise independent tasks.

Distortion is performed with the highest probability. Distortion alters start and finish times of one or several tasks as well as adjusts time windows of all related tasks to maintain precedence constraints ($1 \rightarrow 5 \rightarrow 9$ in Figure 14). Distortion mimics the basic step of the SA algorithm.

Shuffling is performed with lower probability than the distortion, yet with the higher probability than the rest of the transformations. Shuffling changes relative scheduling of two or more tasks wherever it is permitted by the precedence constraints. Shuffling can switch ordering of tasks ($6 \rightarrow 10$), change sequential ordering to parallel or reschedule parallel tasks to be executed sequentially ($4 \rightarrow 8$). The major role of shuffling is to explore local maxima that have similar CE values due to different scheduling of tasks off the critical path.

Explosion adds a copy of the subject schedule to the list of alternatives ($2 \rightarrow 6, 7$). Explosion compliments shuffling by allowing for simultaneous exploration of the groups of similar schedules. We may choose to decrease the rate of explosions with the annealing temperature to focus on improving the current set of solution after the search space was explored to extent.

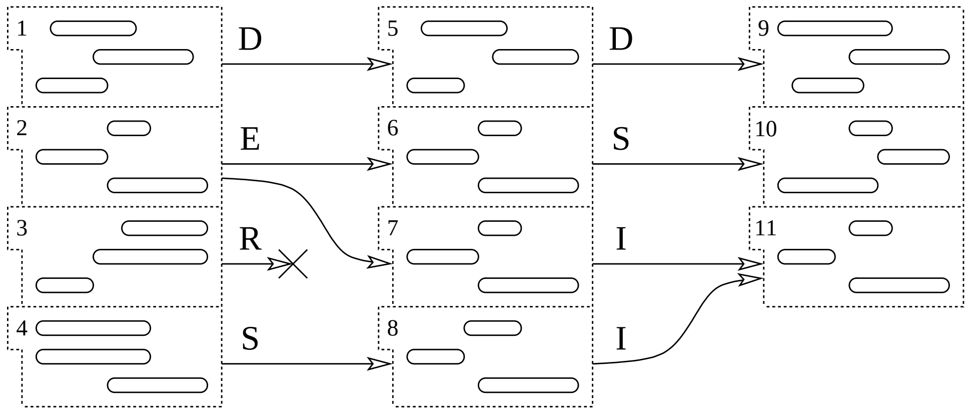


Figure 14: Two steps of the search algorithm execution.

Implosion merges two similar[6] schedules in one. Implosion helps reducing computational expenses from crowding several alternative schedules around one maximum ($7, 8 \to 11$). The rate of implosions will change in the opposite direction to the rate of explosions.

Removal eliminates alternatives that do not score well relative to others ($3 \to \varnothing$). This transformation takes care of the schedules that are stuck in local maxima with low CE values. The rate of removals grows as the annealing temperature decreases.

Each of the first four transformation is tested against SA temperature rule whenever it leads to a decrease in the CE value. In case it is discarded, other transformations are chosen at random and applied until one of them increases CE or passes the temperature rule.

The probabilities of transformations as well as details of the proposed search algorithm's properties are subject to further research. It is reasonable to believe though that the comprehensive study of the RFQ generation mechanism is only possible in the dynamic market environment. In the next section we discuss the approach to the large-scale testing of the MAGNET system that will provide us with the necessary data.

3.2 Evolutionary Framework for Large-scale Testing

After finalizing the EU-based customer agent, as well as developing its supplier counterpart, we plan to devote efforts to testing them against various criteria. In particular, we are interested in testing how well individual agents interact in a populated market. This will help us understand the nuances of the application of EU to RFQ generation. The major goals of this part of the study would be:

- provide the statistical data necessary for the evaluation of the theoretical assumptions and derivations;
- facilitate the understanding of the nuances of the EU-based RFQ generation and to drive improvements to the theory and implementation;

[6] Similarity is a function the distance between two schedules as between two points in the $2N$-dimensional time space.

- study the relative performance of agents in the simulated market, developing an understanding of the properties of automated and mixed-initiative combinatorial auction-based trading societies.

The most compelling approach would be to gather a rich set of statistical data from a commerce domain. That has not proven to be feasible, for two reasons. First, few industrial organizations are sufficiently open to expose the type of data you would need to do that, and we would need data from multiple organizations in a single market. Second, data is gathered to serve a purpose, and our experience tells us that when you attempt to apply existing data to a new purpose, it frequently turns out to be full of inconsistencies and methodological problems.

In lieu of using real industry data, we will design our large-scale test suite atop an abstract domain with controllable statistics, and an evolutionary approach to economic simulation. The structure of the simulation will be defined like this:

- The society will initially consist of one customer agent and many heterogeneous supplier agents. The choice of this setup is due to the assumption that there is little competition on the customer side, so customers can be replaced by one representative agent who issues RFQs with high frequency. Each supplier will initially be provided with a "factory" that produces one type of good or service and maintains the schedule of production.
- The customer agent will issue RFQs for one or several tasks according to a (stationary or not) Poisson process. Multiple RFQs will be open concurrently, so that suppliers must frequently evaluate several RFQs at once. Upon receiving bids, the customer agent will find the winning bundle of bids and award bids. After that, it will monitor the execution of the plan and make appropriate payments to suppliers.
- The performance of supplier agents will be evaluated on the basis of profit averaged over a substantial period of time. The market will run in an evolutionary fashion, i.e. by removing suppliers with negative profits over periods of time, and introducing new suppliers with strategies from the pool of all available strategies whenever the average profit in the market exceeds some positive value.
- The information on successful bids and completed tasks will be collected, processed, and provided to the customer agent to be used in the RFQ generation and winner determination procedures.

The rationale behind our choice of an evolutionary framework is that it provides the necessary information without requiring any complex theory on agent motivation, optimization criteria, or strategic interaction. Unsuccessful species of supplier agents will be washed away from the market, creating places for the more fit. At the same time, the market will provide customer agents with dynamic information on supplier availability, market prices, and cumulative success probabilities.

Evolutionary frameworks have been used extensively in Economics [21, 29, 38]. The framework will allow us to tune the market by tweaking the frequency of issuing RFQs and will allow for the dynamic introduction of new supplier strategies, without imposing any assumptions on the nature of strategies. We will later extend the framework to support trade games to be played with human subjects. This will be a tool specially useful for teaching, as a tool to explore strategic behaviors and to study the emergence of cooperation [1, 2].

4 Related Work

Expected Utility Theory [26] is a mature field of Economics that has attracted many supportive as well as critical studies, both theoretical [17, 18] and empirical [14, 36]. We believe that expected utility will play an increasing role in automated auctions, since it provides a practical way of describing risk estimations and temporal preferences.

Our long term objective is to automate the scheduling/execution cycle of an autonomous agent that needs the services of other agents to accomplish its tasks. Pollack's DIPART system [25] and SharedPlans [11] assume multiple agents that operate independently but all work towards the achievement of a global goal. Our agents are trying to achieve their own goals and to maximize their profits; there is no global goal.

Combinatorial auctions are becoming an important mechanism not just for agent-mediated electronic commerce [12, 41, 32] but also for allocation of tasks to cooperative agents (see, for instance, [13, 6]).

In [13] combinatorial auctions are used for the initial commitment decision problem, which is the problem an agent has to solve when deciding whether to join a proposed collaboration. Their agents have precedence and hard temporal constraints. However, to reduce search effort, they use domain-specific *roles*, a shorthand notation for collections of tasks. In their formulation, each task type can be associated with only a single role. MAGNET agents are self-interested, and there are no limits to the types of tasks they can decide to do. In [9] scheduling decisions are made not by the agents, but instead by a central authority. The central authority has insight to the states and schedules of participating agents, and agents rely on the authority for supporting their decisions. Nisan's bidding language [23] allows bidders to express certain types of constraints, but in MAGNET both the bidder and the bid-taker (the customer) need to communicate constraints.

Despite the abundance of work in auctions [20], limited attention has been devoted to auctions over tasks with complex time constraints and interdependencies. In [24], 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. Bids are expressed by specifying a price to enter a line and a time window. The bidding language, which is similar to what we use in MAGNET, avoids use of discrete time slots. Time slots are used in [40], where a protocol for decentralized scheduling is proposed. The study is limited to scheduling a single resource. MAGNET agents deal with multiple resources.

Most work in supply-chain management is limited to hierarchical modeling of the decision making process, which is inadequate for distributed supply-chains, where each organization is self-interested, not cooperative. Walsh et al [39] 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. MAGNET agents have also to ensure the scheduling feasibility of the bids they accept, and must evaluate risk as well. Agents in MASCOT [31] coordinate scheduling with the user, but there is no explicit notion of payments or contracts, and the criteria for accepting/rejecting a bid are not explicitly stated. Their major objective is to show policies that optimize schedules locally [15]. Our objective is to optimize the customer's utility.

In MAGNET agents interact with each other through a market. The market infrastructure provides a common vocabulary, collects statistical information that helps agents estimate costs, schedules, and risks, and acts as a trusted intermediary during the negotiation process. The market acts also as a matchmaker [37], allowing us to ignore the issue of how agents will find each other.

The determination of winners of combinatorial auctions [30] is hard. Dynamic programming [30] works well for small sets of bids, but does not scale and imposes significant restrictions on the bids. Algorithms such as CABOB [34], Bidtree [33] and CASS [8] reduce the search complexity. Reeves et al [28] use auction mechanisms to "fill in the blanks" in prototype declarative contracts that are specified in a language based on Courteous Logic Programming [10]. These auctions support bidding on many attributes other than price, but the problem of combining combinatorial bids with side constraints is not addressed.

Leyton-Brown et al [16] suggest a way of constructing a universal test suite for winner determination algorithms in combinatorial auctions. Their work does not include cases with precedence and time constraints and, thus, is not directly applicable to the MAGNET framework. It nevertheless provides well-understood test cases for comparing the performance of algorithms.

5 Conclusions

Auction mechanisms are an effective approach to negotiation among groups of self-interested economic agents. We are particularly interested in situations where agents need to negotiate over multiple factors, including not only price, but task combinations and temporal factors as well.

We have shown how an agent can use information about the risk posture of its principal, along with market statistics, to formulate Requests for Quotes that optimize the tradeoff between risk and value, and increase the quality of the bids received. This requires deciding how to sequence tasks and how much time to allocate to each of them. Bids closest to the specified time windows are the most preferred risk-payoff combinations.

The work described here is a part of a larger effort at the University of Minnesota that aims to learn how autonomous or semi-autonomous agents can be used in complex commerce-oriented domains.

Acknowledgments

Partial support for this research is gratefully acknowledged from the National Science Foundation under award NSF/IIS-0084202.

References

[1] R. M. Axelrod. *The evolution of cooperation.* Basic Books, 1984.

[2] Robert Axelrod. *The complexity of cooperation.* Princeton University Press, 1997.

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

[4] John Collins, Maria Gini, and Bamshad Mobasher. Multi-agent negotiation using combinatorial auctions with precedence constraints. Technical Report 02-009, University of Minnesota, Department of Computer Science and Engineering, Minneapolis, Minnesota, February 2002.

[5] John Collins, Maksim Tsvetovat, Rashmi Sundareswara, Joshua Van Tonder, Maria Gini, and Bamshad Mobasher. Evaluating risk: Flexibility and feasibility in multi-agent contracting. In *Proc. of the Third Int'l Conf. on Autonomous Agents*, May 1999.

[6] M. B. Dias and A. Stentz. A free market architecture for distributed control of a multirobot system. In *Sixth Int'l Conf. on Intelligent Autonomous Systems*, pages 115–122, Venice, Italy, July 2000.

[7] Stephanie Forrest. Genetic algorithms: Principles of natural selection applied to computation. *Science*, 261:872–878, 1993.

[8] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. of the 16th Joint Conf. on Artificial Intelligence*, 1999.

[9] Alyssa Glass and Barbara J. Grosz. Socially conscious decision-making. In *Proc. of the Fourth Int'l Conf. on Autonomous Agents*, pages 217–224, June 2000.

[10] B. N. Grosof, Y. Labrou, and H. Y. Chan. A declarative approach to business rules in contracts: Courteous logic programs in XML. In *Proc. of ACM Conf on Electronic Commerce (EC'99)*, pages 68–77. ACM, 1999.

[11] Barbara J. Grosz, Luke Hunsberger, and Sarit Kraus. Planning and acting together. *AI Magazine*, 20(4):23–34, 1999.

[12] Robert H. Guttman, Alexandros G. Moukas, and Pattie Maes. Agent-mediated electronic commerce: a survey. *Knowledge Engineering Review*, 13(2):143–152, June 1998.

[13] Luke Hunsberger and Barbara J. Grosz. A combinatorial auction for collaborative planning. In *Proc. of 4th Int'l Conf on Multi-Agent Systems*, pages 151–158, Boston, MA, 2000. IEEE Computer Society Press.

[14] Bruno Jullien and Bernard Salanié. Estimating preferences under risk: The case of racetrack bettors. *The Journal of Political Economy*, 108(3):503–530, June 2000.

[15] Dag Kjenstad. *Coordinated Supply Chain Scheduling*. PhD thesis, Dept of Production and Quality Engineering, Norvegian University of Science and Technology, Trondheim, Norway, 1998.

[16] Kevin Leyton-Brown, Mark Pearson, and Yoav 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.

[17] Mark J. Machina. Choice under uncertainty: Problems solved and unsolved. *The Journal of Economic Perspectives*, 1(1):121–154, 1987.

[18] Mark J. Machina. Dynamic consistency and non-expected utility models of choice und er uncertainty. *The Journal of Economic Literature*, 27(4):1622–1668, December 1989.

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

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

[21] Richard R. Nelson. Recent evolutionary theorizing about economic change. *Journal of Economic Literature*, 33(1):48–90, March 1995.

[22] Noam Nisan. Bidding and allocation in combinatorial auctions. In *1999 NWU Microeconomics Workshop*, 1999.

[23] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, pages 1–12, Minneapolis, Minnesota, October 2000. ACM SIGecom, ACM Press.

[24] 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.

[25] Martha E. Pollack. Planning in dynamic environments: The DIPART system. In A. Tate, editor, *Advanced Planning Technology*. AAAI Press, 1996.

[26] John W. Pratt. Risk aversion in the small and in the large. *Econometrica*, 32:122–136, 1964.

[27] Colin R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, New York, NY, 1993.

[28] Daniel M. Reeves, Michael P. Wellman, and Benjamin N. Grosof. Automated negotiation from declarative contract descriptions. In *Proc. of the Fifth Int'l Conf. on Autonomous Agents*, pages 51–58, Montreal, Quebec, May 2001. ACM Press.

[29] David Rode. Market efficiency, decision processes, and evolutionary games. Department of Social and Decision Sciences, Carnegie Mellon University, March 1997.

[30] Michael H. Rothkopf, Alexander Pekeč, and Ronald M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[31] 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, May 1999.

[32] Tuomas Sandholm. An algorithm for winner determination in combinatorial auctions. In *Proc. of the 16th Joint Conf. on Artificial Intelligence*, pages 524–547, 1999.

[33] Tuomas Sandholm. Approaches to winner determination in combinatorial auctions. *Decision Support Systems*, 28(1-2):165–176, 2000.

[34] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. of the 17th Joint Conf. on Artificial Intelligence*, Seattle, WA, USA, August 2001.

[35] Tuomas W. Sandholm. *Negotiation Among Self-Interested Computationally Limited Agents*. PhD thesis, Department of Computer Science, University of Massachusetts at Amherst, 1996.

[36] V. Kerry Smith and William H. Desvousges. An empirical analysis of the economic value of risk changes. *The Journal of Political Economy*, 95(1):89–114, February 1987.

[37] Katia Sycara, Keith Decker, and Mike Williamson. Middle-agents for the Internet. In *Proc. of the 15th Joint Conf. on Artificial Intelligence*, pages 578–583, 1997.

[38] Leigh Tesfatsion. Agent-based computational economics: Growing economies from the bottom up. ISU Economics Working Paper No. 1, Department of Economics, Iowa State University, December 2001.

[39] 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.

[40] 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.

[41] 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 Int'l Conf. on Autonomous Agents*, pages 301–308, May 1998.