Evaluating Risk: Flexibility and Feasibility in Multi-Agent Contracting

John Collins, Maksim Tsvetovat, Rashmi Sundareswara,
Joshua van Tonder, Maria Gini
Department of CSE, University of Minnesota

Bamshad Mobasher
School of CTI, DePaul University

Abstract

In an automated contracting environment, where a "customer" agent must negotiate with other self-interested "supplier" agents in order to execute its plans, there is a tradeoff between giving the suppliers sufficient flexibility to incorporate the requirements of the customer's call-for-bids into their own resource schedules, and ensuring the customer that any bids received can be composed into a feasible plan. In this paper, we introduce a bid evaluation process that incorporates cost, task coverage, temporal feasibility, and risk estimation. Using this evaluation process, we provide an empirical study of the tradeoffs between flexibility, plan feasibility, and cost in the context of our MAGNET multi-agent contracting market infrastructure. Our experimental results demonstrate that the advantage of increasing supplier flexibility is dependent on the number of available suppliers. In other words, if the number of suppliers is small, the risk of plan infeasibility outweighs the advantage of added flexibility. On the other hand, if the number of suppliers is large, the more flexible plan specifications result in lower-risk plans.

1 Introduction

Current Internet-based procurement systems are limited to non production related procurement, such as office supplies or computer equipment [9]. In the existing systems, the ability to conduct complex business-to-business transactions, such as automated contracting, is not yet fully realized. Existing software agents mostly help in searching for product and price information (see, for instance, [5]), but there are very few agents capable of automated negotiations [2, 10]. With most organizations spending a large percentage of their budget to purchase goods or services, savings in purchasing can be significant. The ability to automate the negotiation of complex contracts among multiple suppliers is especially important for the coordination of supply-chain management with production scheduling [20]. Buyer-supplier relationships depend on factors such as quality, delivery performance, flexibility as opposed to just cost [11], and these must be taken into account in automated negotiation.

We have designed and implemented a generalized multi-agent market infrastructure, called MAGNET [4], that provides explicit support for complex agent interactions, such as in automated contracting, as well as other types of negotiation protocols. MAGNET uses an independent market infrastructure which adds value and practicality to automated contracting protocols. By independently verifying the identities of participants, by tracking the state of the negotiations and any commitments that result, and by enforcing the rules of the protocol, fraud and misrepresentation are curtailed, and unproductive counterspeculation is minimized.

The goal of MAGNET is to support *Plan Execution by Contracting*, an activity in which a contractor agent, in order to fulfill its goals, must contract with other self-interested supplier agents for all or part of the necessary tasks. These agents are self-interested and exhibit limited rationality. Agents providing resources or services will attempt to gain the greatest possible benefit, and agents requesting resources or services will attempt to pay the lowest price. The MAGNET system incorporates a simple three step, leveled commitment protocol, with a contractor agent issuing a call-for-bids, suppliers replying with bids, and the contractor accepting the bids it chooses. We avoid the need for open-ended negotiation by means of bid break-downs and time-based decommitment penalties [3]. A detailed description of the MAGNET architecture and its underlying negotiation protocol for automated contracting were presented in [4] and [13].

Once the contractor agent receives the bids from supplier agents, it must evaluate the bids based on cost and time constraints, and select the optimal set of bids (or parts thereof) which can satisfy its goals. The resulting *task assignment* forms the basis of an initial schedule for the execution of the tasks.

In this paper, we introduce a bid evaluation process for automated contracting that incorporates cost, task coverage, temporal feasibility, and risk estimation. Using this evaluation process, we provide an empirical study of the tradeoffs between flexibility, plan feasibility, and cost in the context of the MAGNET market infrastructure. Our experimental evaluation will shed light on the behavior of agents in a multi-agent contracting environment, and in particular, on task allocation strategies for the contractor agent. The bid evaluation process is of general interest and can be applied in the context of other contracting protocols.

This paper is organized as follows. Section 2 covers the background and related work. Section 3 describes the interactions of agents with the MAGNET infrastructure and the bid evaluation process. In Section 4, we describe our experimental study of the bid mechanism, and present our empirical results on tradeoffs between flexibility in plan specification and cost/risk factors in plan execution.

2 Related Work

Markets play an essential role in the economy, by facilitating the exchange of information, goods, and services [1], and there is growing evidence that software agents will play an increasing role as mediators in electronic markets [10, 21]. Mediators typically provide services such as searching for a product or supplier, negotiating the terms of a deal, providing payment services, and ensuring delivery of goods.

Many market-based architectures have been proposed for multiple agents (see, for instance, [7, 15, 22, 25, 24]). KQML is emerging as the preferred communication language for agents [8], but other languages have been proposed for electronic commerce, such as CommerceNet's eCo/CBL [23].

Of the essential functions of a market identified by [1], (i) matching buyers and sellers, (ii) facilitating transactions, and (iii) providing institutional infrastucture, existing software agents mostly satisfy the need to search for product and price information (see, for instance, [5]), but there is a growing need for agents capable of more sophisticated automated negotiations [2, 10]. The MAGNET architecture [4] improves on existing open architecture for electronic commerce [23, 14] by adding support for complex negotiations during the contracting phase, providing some protection against fraud and misrepresentation, and curtailing unproductive value-based [16] or time-based [3] counterspeculation by participating agents.

Automated contracting protocols generally assume direct agent-to-agent negotiation. For example, Smith [19] pioneered research in communication among cooperating distributed agents with the Contract Net protocol. The Contract Net has been extended by Sandholm and Lesser [17] to self-interested agents. In these systems, agents communicate and negotiate directly with each other. In MAGNET, on the other hand, agents interact with each other through an independent market infrastructure.

Mechanisms to reduce counterspeculation, such as the Clarke tax mechanism [6] or the Vickrey auction have been proposed for automated negotiation of self-interested agents. In Sandholm's TRACONET system [18] both the bidding and contract execution mechanisms are complicated by the need to operate in an environment where agents cannot trust each other. He does not assume or take advantage of an independent market infrastructure.

To the extent that we require the existence of an external market mechanism as an intermediary, the MAGNET framework is similar to that of Wellman's market-oriented programming used in AuctionBot [26] and The University of Michigan Digital Library [25]. AuctionBot, for instance, supports a variety of auction types each imposing a set of market rules on the agent interactions. Hence, the auctions, themselves, become the intermediaries. In contrast, the MAGNET framework provides explicit market mechanisms which can not only specify and enforce auction parameters, but also support more complex interactions such as automated contracting. Furthermore, these market mechanisms also enforce general market rules and "social laws", such as government regulations, by which all participants must abide.

Rosenschein and Zlotkin [16] showed how the behavior of the agents can be influenced by the set of rules that the system designers choose for the agents' environment. In their study the agents are homogeneous and there are no side payments. In other words, the goal is to share the work, not to pay for work. Sandholm's agents [17, 18] redistribute work among themselves by a contracting mechanism. Unlike Rosenschein and Zlotkin, Sandholm considers agreements involving explicit payments, but he also assumes that the agents are homogeneous – they have equivalent capabilities, and any agent can handle any task. In MAGNET, on the other hand, agents are heterogeneous, and decide on their own what tasks to handle by responding to a call for bids that requires specific tasks or products within a specified time window.

3 Agent Interactions in Automated Contracting

The interactions involved in the basic bidding and execution cycle among the contractor, supplier, and market are illustrated in Figure 1. The customer agent starts with an initial plan which forms the basis for its call-for-bids. It then initiates a *session* in the market to encapsulate the bidding and execution activities related to meeting its goal. Thus, from the point of view of a customer agent (also known as the contractor agent), interactions fall into three phases: planning, bidding, and execution.

In the planning phase, the customer selects one or more markets, each of which specializes in certain types of product or service categories. The customer uses an *ontology*, provided by the market, to develop a partial plan, and then estimates tentative values for plan components based on the value of its current goal and the "criticality" of each component. The call-for-bids C contains a subset of the tasks in the plan P with their precedence relations. Let $C.t_{es}(s)$ denote the early start time for task s as specified in a call-for-bids C. Other contexts in which times and other factors can be specified include bids $(b.t_{es}(s))$ and the computed critical path $(A.t_{es}(s))$. The notation $s' \prec s$ denotes the constraint $t_f(s') \leq t_s(s)$, meaning that task s' must be complete before task s can start. A necessary condition for this is $(t_{es}(s') + d(s')) \leq (t_{lf}(s) - d(s))$, where $d(s)$ refers to the duration of task s.

For each task s in the call-for-bids C, the following variables must be specified:

- a time window, consisting of an earliest start time $C.t_{es}(s)$ and a latest finish time $C.t_{lf}(s)$,

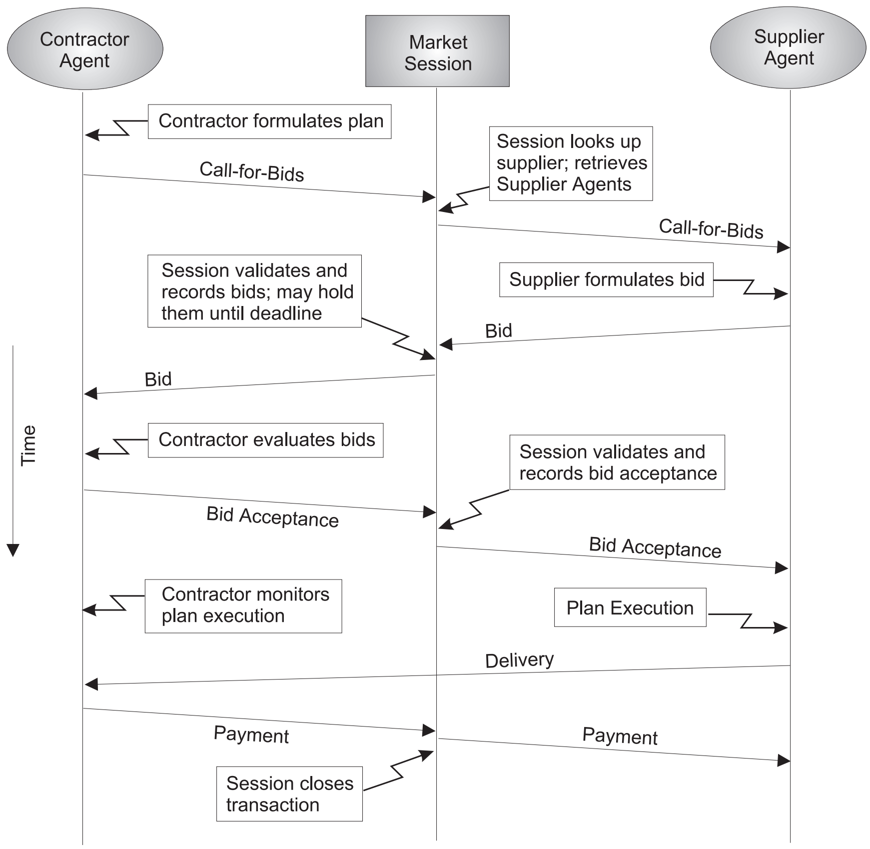

Figure 1: A Typical Session-Mediated Negotiation

- a set of precedence relationships $C.Pred(s) = \{s' \in C \mid s' \prec s\}$, the set of other tasks $s' \in C$ whose completion must precede the start of s, and

- a decommitment penalty $dcom(s)$. This is the penalty the supplier has to pay to the contractor agent if the supplier decommits. (See [3] for further explanation of the decommitment penalties.)

The agent may have general knowledge of normal durations of tasks, or it may not. In order to minimize bid prices, vendors need flexibility in making resource commitments. Two possible strategies are (a) to use time windows that start at time t_0 and end at time t_{goal}, where t_0 is the start time for the entire plan and t_{goal} is the deadline for achieving the top level goal, and (b) to schedule tasks ahead of time using approximate durations, computing early start and late finish times using the Critical Path algorithm [12]. The minimum duration of the entire plan is called the *makespan* of the plan. The difference between t_{goal} and the latest early finish time is called the *total slack* of the plan. If t_{goal} is set equal to $t_0 + makespan$, then the total slack is 0, and all tasks for which $t_{ef} = t_{lf}$ are called *critical* tasks. Paths in the graph through critical tasks are called *critical paths*.

In the bidding phase, the customer agent must evaluate and assemble the bids into a minimum-cost feasible schedule in order to determine which bids to accept. Let B be the set of returned bids, each bid $b \in B$ includes a price $b.price(S')$ for some subset S' of the elements of the call-for-bids, prices $b.price(s)$ for individual elements of the bid subset $s \in S'$, and timing information for the bid elements. When $b.price(S') \leq \sum_{s \in S'} b.price(s)$ then a *discount* is associated with the bid b. Timing information for a bid b and a task s includes early start $b.t_{es}(s)$, late finish $b.t_{lf}(s)$, and duration $b.d(s)$ for each task in the bid. The semantics of a bid is that a supplier is willing to perform the task s for the bid price $b.price(s)$ starting at any time $t_s(s)$ such that $b.t_{es}(s) \leq t_s(s) \leq (b.t_{lf}(s) - b.d(s))$, and finishing at time $(t_s(s) + b.d(s))$.

During the execution phase, the interactions can be much more complex than indicated here, since either party may decommit from a contract (and pay a penalty).

4 Experiment

The main goal of the experiment is to observe the relationships among numbers of suppliers, the temporal flexibility offered to suppliers, and the price and risk factors experienced by a customer agent.

4.1 Hypothesis

The experiment is based on the following assumptions:

- Different suppliers will require varying amounts of time to perform a given task.

- Suppliers are motivated to maximize the profitable utilization of the resources under their control.
- Suppliers may or may not have flexibility in resource loading over time. For a supplier with a fixed set of resources, bids can only be submitted for time intervals where resources are uncommitted.

We expect that larger numbers of bidders will result in lower prices, a higher probability of being able to compose a feasible plan, and lower risk factors. We also expect that giving bidders larger time windows in which to bid will result in larger numbers of bids and lower costs, but may also result in greater difficulty finding feasible compositions of bids. If the number of bidders is large enough, larger time windows should allow the customer to compose lower-risk plans by finding bid combinations that contain more slack time for tasks that are likely to involve higher risks, such as cases where only one bidder is available for a given task.

4.2 Experimental Setup

The experimental setup includes 3 main components: a MAGNET Server as described in [4], a Customer Agent that generates plans, requests bids, and evaluates bids, and a Supplier Agent that generates and submits bids.

4.2.1 Customer Agent: Generate Plan

The customer agent generates a plan with a combination of parallel and serial precedence relationships, consisting of a set S of 7 tasks. Each task has an expected duration $d_e(s)$ of 1 hour. In the Call For Bids, each task $s \in S$ is specified by an early start time $C.t_{es}(s)$, a late finish time $C.t_{lf}(s)$, and the set of tasks $pred(s)$ that must precede it. In order to allow for variability in actual task durations, and to allow suppliers some degree of flexibility in their resource scheduling, the specified start and finish times are relaxed from their expected values by varying degrees. These times are computed as follows:

- *start time*: The target start time t_0 is set to an arbitrary point in time in the near future.

- *deadline*: Using expected task durations $d_e(s)$, the earliest possible completion time for the entire set of tasks is found by analyzing the dependency graph. The target completion time t_d is then set to allow an overall slack of 10%.

- *time windows*: Early start and late finish time specifications are computed for each task $s \in S$. For each root task which has no predecessors, $C.t_{es}(s) = t_0$. For each leaf task which has no successors, $C.t_{lf}(s) = t_d$. For all other tasks,

$$C.t_{es}(s) = \max_{s' \in pred(s)} (C.t_{es}(s') + d_a(s'))$$

  and

$$C.t_{lf}(s) = \min_{s' \in succ(s)} (C.t_{lf}(s') - d_a(s')),$$

  where d_a is an adjusted duration used to introduce flexibility into the specification and allow for variation in actual task durations. The ratio $\frac{d_e(s)}{d_a(s)}$ is one of the experimental variables, as described below.

4.2.2 Supplier Agent: Generate Bids

The supplier agent is a test agent that masquerades as an entire community of suppliers. Each time a new Call For Bids is announced by the Market, the supplier attempts to generate some number of Bids. In the first series of experiments, the number is 10 to represent a relatively small community of suppliers, and in the second series the number of simulated suppliers is 30.

As described in [13], each Bid represents an offer to execute some subset of the tasks specified in the Call For Bids. A price for the whole set is specified, and for each task in the set, a price, early start time, late finish time, and maximum duration are specified. It is a requirement of the protocol that the time parameters in a Bid are within the time windows specified in the Call For Bids. Bids are generated randomly as follows:

- *select initial task*: A task s is selected at random from the Call For Bids.

- *set duration*: The duration $d(s)$ for each task is a normally distributed random variable with a mean of 1 hour and standard deviation of 6 minutes.

- *determine feasibility*: If the duration is longer than the time window

$$[C.t_{es}(s)\,,\,C.t_{lf}(s)]$$

specified in the Call For Bids, the task is dropped.

- *expand the bid*: For tasks that are not dropped, each predecessor and successor link is followed, with a probability of 50%, to attempt to add a task to the bid. This has the effect of producing "realistic" bids composed of contiguous sets of tasks. For each of those bids, the duration is set and feasibility checked as noted above. The result of this expansion is the set of tasks $b.S$ to be included in the bid.

- *fill in bid data*: The following steps are performed on each selected task $s \in b.S$:

  - *determine bid start and finish times*: The expected early start time of the task $t_{es}(s)$ is selected as a uniformly distributed variable between the early start time $C.t_{es}(s)$ for s specified in the Call For Bids, and a late start time derived by subtracting $d(s)$ from the late finish time $C.t_{lf}(s)$ specified in the Call For Bids. The resulting bid early start time $b.t_{es}(s)$ is inserted in the Bid. A bid late finish time $b.t_{lf}(s)$ is similarly determined by picking a random value in the interval

  $$[b.t_{es}(s) + d(s)\,,\,C.t_{lf}(s)]$$

  - *determine price*: Price $b.p(s)$ for a task s is set as

  $$b.p(s) = \rho \times C.\mathit{flex}(s) \times b.\mathit{commit}(s)$$

  where ρ is a normally-distributed value with a mean of 1000 and a standard deviation of 50, *flex* is a flexibility discount, and *commit* is a resource commitment premium. The value of *flex* varies linearly from 1.0 to 0.8 as the ratio of $\frac{C.t_{lf}(s) - C.t_{es}(s)}{d(s)}$ increases from 1.0 to 2.0, and then remains at 0.8. Similarly, the value of *commit* varies linearly from 1.0 to 1.2 as the ratio $\frac{b.t_{lf}(s) - b.t_{es}(s)}{d(s)}$ increases from 1.0 to 2.0, and then remains at 1.2.

- *set decommit penalty*: The decommitment penalty that the customer must pay for backing out of a bid is chosen as a random value between 0 and the price of the bid. In the full MAGNET protocol, an arbitrary piecewise-linear function is permitted as a decommitment penalty specification, but for the present experiment we use a value that does not vary with time.

- *set overall discount*: If the supplier is bidding on multiple contiguous tasks, it may offer a discount in exchange for acceptance of the entire combination. The overall price for the bid is

$$b.p = \sum_{s \in b.S} (b.p(s) \times b.discount)$$

  where $b.discount$ is a random value between 1.0 and 0.8.

4.2.3 Customer Agent: Evaluate Bids

Once the bidding deadline is past, the customer evaluates the set of bids in an attempt to find a combination that minimizes a combination of cost and risk, provides coverage of all tasks, and allows for a feasible schedule. For each bid, the customer has the option of selecting the entire bid and paying the overall discounted price $b.p$, or selecting a subset of the task bids from $b.S$.

The evaluator uses a stochastic search, executing the following steps:

- *choose a node*: A previous partial or complete solution is chosen as a base for the next expansion. A node is a partial or complete mapping of tasks to bids. The root node is the mapping of all tasks to no bids. The nodes are kept in an ordered queue, and the choice is made by choosing an exponentially-distributed value whose mean is 10% of the range of evaluations in the queue. The range is limited; nodes with evaluations that fall off the end are dropped, and when a new best node is found, the tail is trimmed.

- *choose an expansion*: Expansions are made by adding a bid or individual task-bid to a node. The probability of adding a task-bid rather than a bid is equal to the current coverage factor of the node being expanded, where the coverage factor is the proportion of tasks that are mapped to bids. This is to encourage use of discounted whole bids when coverage is low, and individual task-bids when coverage is high and the addition of a bid is likely to displace other bids from the mapping because of task overlaps. Bids that have already been used to expand a given node are disallowed, as are bids that are in the *tabu list*, which contains the last 10 expansions in the parentage of the node. In addition, if there are any "singleton" bids, or tasks for which only one bid was submitted, bids for those tasks may not be displaced by expansions.

- *evaluate the resulting node*: If a valid node was produced by the expansion, it is tested for coverage, feasibility, cost, and risk, as follows:
  - *coverage*: A penalty is added for each task that is not mapped to a bid.
  - *feasibility*: The Critical Path algorithm is run across the task-bid mapping, with unmapped nodes assumed to have a duration of 0. A penalty proportional to the total negative slack is added.
  - *cost*: The total cost of all mapped bids is added.
  - *risk*: In general, Risk factors include decommitment cost, recovery cost, loss of value as the end date is delayed, cost of plan failure, and other factors. For the purposes of this experiment, we have chosen to focus on the expected cost of decommitment penalties. These are penalties that must be paid by the customer to suppliers if the customer backs out of a commitment. We assume that the customer will only do this with regard to a particular task if execution of a prior task experiences an unrecoverable failure.

    The total expected decommitment cost is

    $$c_d(total) = \sum_{s \in S} (P_{pud}(s) \times c_d(s))$$

    where $P_{pud}(s)$ is the probability that a task $s' \in pred^*(s)$ in the transitive closure of the predecessors of s will experience an unrecoverable decommitment. We calculate $P_{pud}(s)$ as

    $$P_{pud}(s) = 1 - \prod_{s' \in pred^*(s)} (1 - (P_{sd}(s') \times (1 - P_{rec}(s'))))$$

where $P_{sd}(s')$ is the probability that the supplier chosen for task s' will decommit, and $P_{rec}(s')$ is the probability that recovery can be achieved after decommitment:

$$P_{rec}(s) = (1 - \frac{1}{1 + \frac{A.t_{lf}(s) - A.t_{es}(s)}{b.d(s)}})(1 - \frac{1}{bidCount(s)})$$

The justification for this formula is that the probability of recovery increases with increasing slack, and with a larger number of available suppliers as evidenced by the number of bids received for the task.

- *stopping criterion*: The search ends when a number of expansions equal to 20 times the number of bids have been attempted without finding an improvement. The best feasible and covered solution, if any, is returned.

4.3 Experimental Results

We explored two dimensions of the bid-evaluation problem with the setup described above. The first dimension is simply the number of bidders. One set was run with 10 bidders to simulate a rather small set of possible suppliers. We ignore the possible effects of collusion and collaboration among vendors in a restricted market. The other set included 30 bidders.

For each of the two sets, we performed 3 runs, each of 40 iterations of composing the Call For Bids, receiving Bids, and evaluating them. The runs varied the time windows specified in the Call For Bids, using $\frac{d_e(s)}{d_a(s)}$ set to 0.9, 0.6, and 0.3. In the charts below, these numbers correspond to "overlap" percentages of 10%, 40%, and 70%.

As we can see from Figure 2, the probability of finding a feasible, covered plan is reduced dramatically with small numbers of bidders. When time windows are short, the infeasibility test in the bid preparation process results in smaller numbers of bids, with the bids received being composed of fewer tasks. However when the time windows are increased, the effect of larger number of (larger) bids is more than offset by fewer combinations that form temporally feasible plans.

Figure 3 shows the relation between expected decommitment cost and time-window size. Given that the deadline for the overall plan is constant

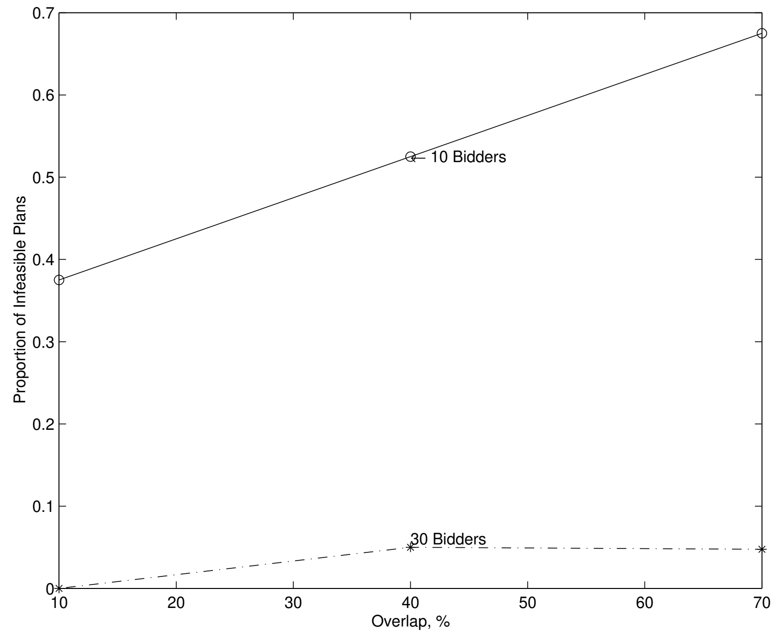

Figure 2: Infeasibility vs. Slack

across these runs, and given that the average task duration is also constant, it is interesting that the risk evaluation generally decreases as the time windows are increased. The anomaly in the 10-bidder case appears to be related to small sample size; only 6 feasible plans were found in the 70% overlap case.

With the larger number of bidders there is a clear relationship between the size of the time windows specified in the Call For Bids and the expected decommitment cost. This is because the bidders are generally composing bids with larger time windows, and even though many of them cannot be composed while preserving precedence relations, the evaluator is able to find combinations that maximize slack early in the plan, where it has the greatest impact on expected decommitment cost.

Figure 4 shows the expected relationship between cost and number of bidders. The increasing cost with larger time windows is due to the same effect that reduces risk: suppliers are specifying larger time windows in the bids, and are applying the resource-commitment premium. In a more realistic situation, bidders could submit both a higher-cost, more flexible and lower-risk bid, along with a lower-cost, less flexible and higher-risk bid, allowing

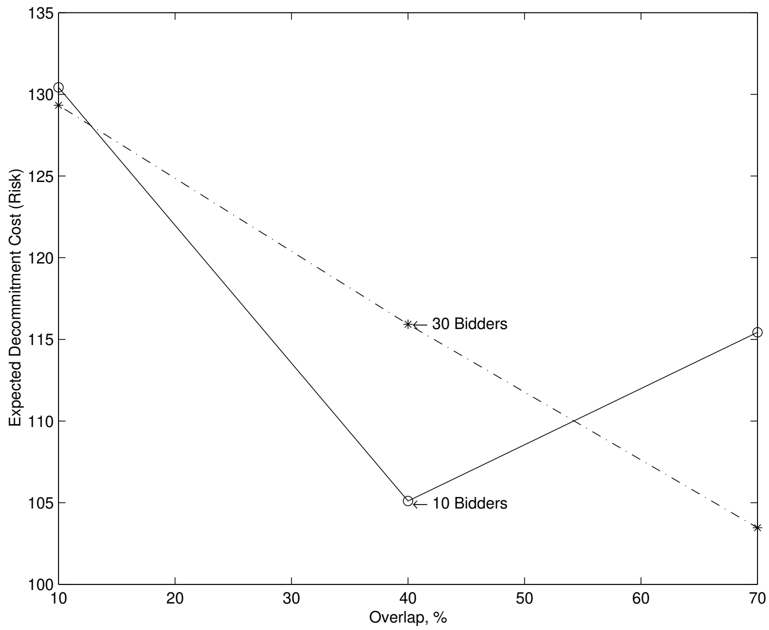

Figure 3: Risk vs. Slack

the customer to make the decision.

5 Conclusions and Future Work

We have presented a bid evaluation process for automated contracting that incorporates cost, task coverage, temporal feasibility, and risk estimation, and we have provided, using this evaluation process, an empirical study of the tradeoffs between flexibility, plan feasibility, and cost in the context of the MAGNET market infrastructure. We show that a customer agent can make tradeoffs between price, feasibility, and risk by adjusting the temporal specifications of the tasks to be performed. We also show that the nature of these tradeoffs varies with the number of potential suppliers in the market.

This work raises several interesting questions for future research. A game-theoretic analysis of the protocol could be done to determine optimal strategies for its use by agents. In particular, how should the time windows and decommitment penalties be used, and how should proposed decommitment

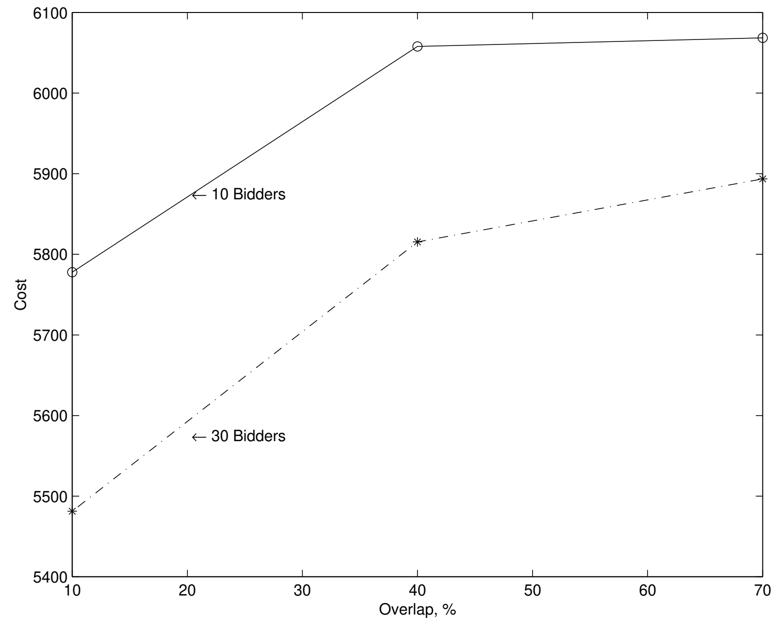

Figure 4: Cost vs. Slack

functions be evaluated when computing marginal costs of plan alternatives. The risk evaluation process also needs to be expanded to include recovery cost estimates and time-varying decommitment penalties.

In cases where the value of the goal is a decreasing function of time, or where schedule slack is low and probability of missing the goal deadline is high, there may be value in offering to suppliers not only a penalty for decommitment or failure to perform, but also a premium for early completion of subtasks. We will study how to calculate the value of such a premium, and develop methods for incorporating this into the negotiation protocol.

References

[1] Yannis Bakos. The emerging role of electronic marketplaces on the Internet. *Comm. of the ACM*, 41(8):33–42, August 1998.

[2] Carrie Beam and Arie Segev. Automated negotiations: A survey of the state of the art. Technical Report CITM Working Paper 96-WP-1022,

Walter A. Haas School of Business, 1997.

[3] John Collins, Scott Jamison, Maria Gini, and Bamshad Mobasher. Temporal strategies in a multi-agent contracting protocol. In *AAAI-97 Workshop on AI in Electronic Commerce*, July 1997.

[4] John Collins, Ben Youngdahl, Scott Jamison, Bamshad Mobasher, and Maria Gini. A market architecture for multi-agent contracting. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 285–292, May 1998.

[5] B. Doorenbos, O. Etzioni, and D. Weld. A scalable comparison-shopping agent for the world-wide web. In *Proceedings of the First International Conference on Autonomous Agents*, pages 39–48, February 1997.

[6] E. Ephrati and J. Rosenschein. Deriving consensus in multiagent systems. *Artificial Intelligence*, 87:21–74, 1996.

[7] J. Eriksson and N. Finne. Marketspace: an open agent-based market infrastructure. Master's thesis, Uppsala University, Computing Science Department, 1997.

[8] Tim Finin, Yannis Labrou, and James Mayfield. Kqml as an agent communication language. In Jeff Bradshaw, editor, *Software Agents*, pages 291–316. MIT Press, Cambridge, MA, 1997.

[9] Judith Gebauer and Arie Segev. Assessing internet-based procurement to support the virtual enterprise. 2(3), September 1998.

[10] Robert H. Guttman, Alexandros G. Moukas, and Pattie Maes. Agent-mediated electronic commerce: a survey. *Knowledge Engineering Review*, June 1998.

[11] S. Helper. How much has really changed between us manufacturers and their suppliers. *Sloan Management Review*, 32(4):15–28, 1991.

[12] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, 1990.

[13] Scott Jamison. A negotiation protocol and market agent model for complex transactions in electronic commerce. Technical report, University of Minnesota, Department of Computer Science and Engineering, Minneapolis, MN, 1997.

[14] S. McConnell, M. Merz, L. Maesano, and M. Witthaut. An open architecture for electronic commerce. Technical report, Object Management Group, Cambridge, MA, 1997.

[15] J. A. Rodriguez, F. Noriega, C. Sierra, and J. Padget. FM96.5 - a Java-based electronic auction house. In *Second Int'l Conf on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, London, April 1997.

[16] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter*. MIT Press, Cambridge, MA, 1994.

[17] Tuomas Sandholm and Victor Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *1st International Conf. on Multiagent Systems*, pages 328–335, San Francisco, 1995.

[18] Tuomas W. Sandholm. *Negotiation Among Self-Interested Computationally Limited Agents*. PhD thesis, University of Massachusetts, 1996.

[19] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Trans. on Computers*, 29(12):1104–1113, December 1980.

[20] J. M. Swaminathan, S.F. Smith, and N.M. Sadeh. Modeling the dynamics of supply chains: A multi-agent approach. *Decision Sciences*, 1997.

[21] K. Sycara, K. Decker, and M. Williamson. Middle-agents for the Internet. In *Proceedings of the 15th Joint Conference on Artificial Intelligence*, 1997.

[22] Katia Sycara and Anandeep S. Pannu. The RETSINA multiagent system: towards integrating planning, execution, and information gathering. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 350–351, 1998.

[23] J. M. Tennenbaum, T. S. Chowdhry, and K. Hughes. eCo System: CommerceNet's architectural framework for internet commerce. Technical report, Object Management Group, Cambridge, MA, 1997.

[24] M. Tsvetovatyy, M. Gini, B. Mobasher, and Z. Wieckowski. MAGMA: An agent-based virtual market for electronic commerce. *Journal of Applied Artificial Intelligence*, (6), 1997.

[25] M.P. Wellman, W.P. Birmingham, and E.H. Durfee. The digital library as a community of information agents. *IEEE Expert*, 11(3):10–11, June 1996.

[26] P.R. Wurman, M.P. Wellman, and W.E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second Int'l Conf. on Autonomous Agents*, May 1998.