

# Exploring Decision Processes in Multi-Agent Automated Contracting

John Collins  
Dept of Computer Science and Engineering  
University of Minnesota  
jcollins@cs.umn.edu

Maria Gini  
Dept of Computer Science and Engineering  
University of Minnesota  
gini@cs.umn.edu

## ABSTRACT

We are interested in multi-agent contracting, in which customers must solicit the resources and capabilities of other, self-interested agents in order to accomplish their goals. Goals may involve the execution of multi-step plans, in which different steps are contracted out to different suppliers. We have developed a testbed that can generate sets of plans with known statistical attributes, formulate and submit requests for quotations, generate bids with well-defined statistics, and evaluate those bids according to a number of criteria.

## 1. INTRODUCTION

Over the past decade, manufacturing logistics has become significantly more complex. Many processes are outsourced to outside contractors, making supply chains longer and more convoluted. The increased complexity is often compounded by accelerated production schedules which demand tight integration of all processes. Thus, the field is ripe for the introduction of systems that automate logistics planning among multiple entities such as manufacturers, part suppliers, shippers, and specialized subcontractors.

We are interested in understanding how a community of heterogeneous, self-interested agents, can make commitments and carry out plans that require multiple tasks and coordination among multiple agents. We assume that some agents (*customer agents*) have goals that they themselves cannot satisfy, either because they lack the abilities, or the resources, to carry out at least some of the operations. There are also other agents (*supplier agents*) who have resources to offer, and who are willing to make those resources available to other agents in a way that maximizes their value to the agents that control them.

To help automate logistics planning, we propose a combination of an auction-based negotiation process along with a generalized market architecture. We have implemented a prototype testbed, called MAGNET (Multi AGent NEgotiation Testbed).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AGENTS'01 May 28-June 1, 2001, Montréal, Quebec, Canada.  
Copyright 2001 ACM 1-58113-326-X/01/0005 ...\$5.00.

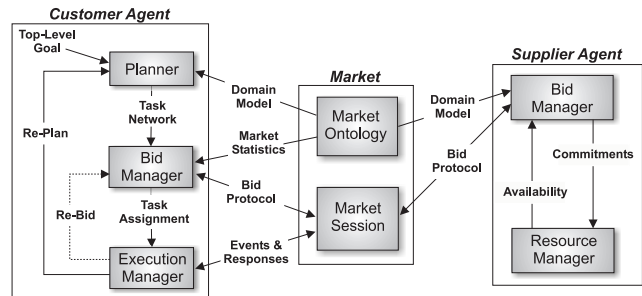


Figure 1: The MAGNET architecture

## 2. AGENTS AND THEIR ENVIRONMENT

Agents may fulfill one or both of two roles with respect to the MAGNET architecture, as shown in Figure 1. Customer agents pursue their goals by formulating and presenting Requests for Quotations (RFQs) to Supplier agents through a market infrastructure. The RFQ specifies a task network that includes task descriptions, a precedence network, and possibly other time constraints. Customer agents attempt to satisfy their goals for the least net cost, where cost factors can include not only bid prices, but also goal completion time and risk factors. More precisely, these agents are attempting to maximize the utility function of some user, as discussed in [1]. Supplier agents attempt to maximize the value of their resources by submitting bids in response to those RFQs, specifying what tasks they are willing to undertake, when they are available to perform those tasks, and at what price.

## 3. A CUSTOMER AGENT

We now focus on the structure and responsibilities of a Customer agent in the MAGNET environment. As indicated in Figure 1, the basic operations are planning, bidding, and plan execution. We have implemented a simple Planner that generates random plans with well-defined statistics, and we have a Bid Manager with a fairly rich implementation of tools for composing RFQs and selecting bids. The Execution Manager is not yet implemented.

### 3.1 Planner

The Planner's task is to produce executable plans, which are represented as task networks with temporal constraints. The planner in the testbed generates tasks by selecting randomly from a library of task types, and then creates random precedence relations among them. It can also accept pre-

defined plans. We expect that in many domains, plans will be chosen from a library or defined by a human user rather than being generated by a general-purpose planner.

The definitions of tasks in the Domain Model are shared among the agents. This model includes not only the task definitions, but statistics (collected by the market) about each task type. These statistics include expected duration and variability, expected price and variability, and resource availability data.

### 3.2 Bid Manager

The Bid Manager is responsible for ensuring that resources are assigned to each of the tasks of a plan, that the assignments taken together form a feasible schedule, and that the cost of executing the plan is minimized. This cost must also be less than the value of the goal at the time the goal is reached.

The Bid Manager must construct and issue a RFQ, evaluate bids, and accept bids in order to carry out its responsibilities. Initially, its primary responsibility is to decide how much time to allocate to negotiation and how much to plan execution. It also has to decide which markets to use, when to consult catalog and timetable databases, and which alternatives to pursue and in what order. For example, it may decide to solicit bids on a high-value but risky approach, and if that fails to fall back on a lower-value but safer alternative. It could also decide to defer taking bids on later tasks until earlier tasks were underway or even completed.

Before bids can be solicited in a market, an RFQ must be composed. The RFQ is a structure that contains tasks and precedence relations, along with a set of scheduling constraints. One role of the Bid Manager is to determine those scheduling constraints. Its goal is to produce an RFQ that will solicit the most advantageous set of bids possible, by finding a balance between giving maximum flexibility to suppliers, ensuring that the resulting bids will combine feasibly, and ensuring that the job will be completed by the deadline.

After bids have been received, the Bid Manager has to evaluate them in an attempt to find an optimal or near-optimal mapping of bids to tasks, respecting temporal constraints, and then award them. The evaluation must be completed within the period of time allocated to negotiation.

Several approaches to bid evaluation for combinatorial auctions have been proposed [5, 3, 6], but they cannot be easily extended to deal with temporal constraints.

We have implemented two evaluators. One is based on Integer Programming, and the other is a highly modular Simulated Annealing (SA) search engine [2]. The Integer Programming (IP) solver operates in two phases. The first phase generates basic bid-compatibility constraints, and then walks all paths of length 2 or greater in the precedence network, across all compatible bid combinations, to discover feasibility constraints. These are then packaged up and sent off to an external IP solver. The core of the simulated-annealing engine is similar to the one described in [4]. Starting with a plan and a set of bids, we generate and evaluate bid mappings until one of several stopping conditions holds. These include failure to find improvement for a configurable number of iterations, expiration of the deliberation time limit, and lack of mappings that have any untried expansions.

Our experiments show a significant variation in the solu-

tion time for the IP method. We are exploring how to combine the IP optimizer with the simulated annealing engine to produce any-time behavior. This is important because MAGNET agents have a strictly limited amount of time to make bid-award decisions.

### 4. SUPPLIER AGENTS

Since our primary interest has been in the workings of the Customer agent, our Supplier agents are currently fairly simple-minded entities. They receive RFQs and respond by submitting bids. They do not maintain resource schedules, and they have no persistent identity.

We have implemented different generators for bids. Examples include one that always bids on certain task types if they are present in the RFQ, one that generates a random set of bids, and one that attempts to generate a set that covers all tasks in the RFQ.

Each bid contains one or more individual tasks. For each task the bid specifies its expected duration, early-start and late-finish time. Duration and cost are selected from random distributions specified in the task-type description. Early-start and late-finish times are also randomly generated from the resource-availability data in the task-type description.

### 5. CONCLUSIONS

The MAGNET testbed is a prototype implementation of a Customer agent, along with simulated Supplier agents. It is highly configurable and extensible, and has been used for several statistical studies aimed at understanding the decision processes of a Customer agent.

### Acknowledgments

Partial support for this research provided by NSF under award NSF/IIS-0084202.

### 6. REFERENCES

- [1] J. Collins, C. Bilot, M. Gini, and B. Mobasher. Mixed-initiative decision support in agent-based automated contracting. In *Proc. of the Fourth Int'l Conf. on Autonomous Agents*, pages 247–254, June 2000.
- [2] J. Collins, R. Sundareswara, M. Gini, and B. Mobasher. Bid selection strategies for multi-agent contracting in the presence of scheduling constraints. In A. Moukas, C. Sierra, and F. Ygge, editors, *Agent Mediated Electronic Commerce II*, volume LNAI1788. Springer-Verlag, 2000.
- [3] Y. Fujishjima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. of the 16th Joint Conf. on Artificial Intelligence*, 1999.
- [4] C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, New York, NY, 1993.
- [5] T. Sandholm. An algorithm for winner determination in combinatorial auctions. In *Proc. of the 16th Joint Conf. on Artificial Intelligence*, pages 524–547, 1999.
- [6] W. E. Walsh, M. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, October 2000.