Search Strategies for Bid Selection in Multi-Agent Contracting

John Collins[1], Rashmi Sundareswara[1], Maksim Tsvetovat[1], Maria Gini[1], and Bamshad Mobasher[2]

[1] Department of Computer Science and Engineering, University of Minnesota
[2] School of Computer Science, Telecommunications, and Information Systems, DePaul University

Abstract. Bid evaluation in a multi-agent automated contracting environment presents a challenging search problem. We introduce a multi-criterion, anytime bid evaluation strategy that incorporates cost, task coverage, temporal feasibility, and risk estimation into a simulated annealing framework. We report on an experimental evaluation using a set of increasingly informed search heuristics within simulated annealing. The results show that excess focus on improvement leads to faster improvement early on, at the cost of a lower likelihood of finding a solution that satisfies all the constraints. The most successful approach was an interleaving of random and focused selectors.

1 Introduction

The University of Minnesota's MAGNET (Multi-Agent Negotiation Testbed) system is an innovative agent-based approach to complex contracting and supply-chain management problems. The MAGNET system [5] comprises a set of agents who negotiate with each other through a market infrastructure using a finite, leveled-commitment protocol. It is designed to support the execution of complex plans among a population of independent, autonomous, heterogeneous, self-interested agents. We call this activity *Plan Execution By Contracting*.

Plan Execution by Contracting is designed to extend the applicability of agent negotiation to new domains, where schedules for production and delivery affect the cost and feasibility of services and products, and where monitoring the performance of task execution is an essential part of the process. This is especially important for domains such as logistics, dynamic planning and scheduling, and coordination of supply-chain management with production scheduling, where what is important is flexibility, ease of use, quality, performance, as opposed to just cost [10].

In general, a MAGNET agent has four basic functions: planning, negotiation, execution monitoring, and resource management. Within the scope of a negotiation, we distinguish between two agent *roles*, the *Customer* and the *Supplier*. A Customer is an agent who has a plan to satisfy some goal, and needs resources outside its direct control in order to carry out that plan. The plan may have a *value* that varies over time. In response to a *call-for-bids*, some Supplier agents may offer to provide the requested resources or services, for specified prices, over specified time periods.

Once the Customer agent receives bids, it must evaluate them based on cost, task coverage, 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. The Customer's goals are time-sensitive, the negotiation process requires Customer and Suppliers to agree on the times for execution of tasks, and time factors can affect the cost of execution. Because of the pervasive influence of time, the Customer must allocate time to the bidding and deliberation processes before knowing how many bids will be submitted and therefore how complex the evaluation process will be [17]. In order to be able to award bids and start execution on schedule, an anytime search [2] must be used, and its characteristics must be known ahead of time.

We have developed a highly tunable anytime search, based on a Simulated Annealing [15] framework, a set of evaluators, and a set of expansion selectors that are mapped to evaluation characteristics. The search must solve both bid-allocation and temporal feasibility constraints, as well as minimize cost and risk. Given that the time allocated to search will seldom be sufficient to explore a significant fraction of the space, we must find an appropriate tradeoff between systematic optimization and random exploratory behavior. We describe here the first of a set of experiments that will allow us to allocate time to the various agent activities, and to maximize the performance of the search.

This paper is organized as follows. We describe our study on bid selection in Section 2, and the experimental setup in Section 3. In Section 4 we present our empirical results on the performance of the various bid-selection strategies. Section 5 covers the background and related work. Section 6 describes our conclusions and suggestions for future work related to this problem.

2 Agent Interactions and Time Allocation

The negotiation portion of the MAGNET protocol is a finite 3-step process that begins when a Customer agent issues a call-for-bids. The call-for-bids specifies a set of tasks that must be performed, along with time and precedence constraints. Suppliers may reply with bids, and the Customer accepts the bids it chooses with bid-accept messages.

Our contracting protocol allows agents to bid on combinations of items, does not assume superadditivity of bid prices, and assumes exclusive OR bids (when bids on combinations are submitted by the same agent, only one of them can be accepted). Bids have multiple attributes and include a time window. To determine which bids (or parts of bids) to accept, the Customer agent considers coverage, feasibility, cost, and 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. The Customer agent's objective is to maximize utility, which requires minimizing cost and risk of not accomplishing the task.

The timeline in Figure 1 shows an abstract view of the progress of a single negotiation. At the beginning of the process, the Customer agent must allocate time to deliberation for its own planning, for supplier bid preparation, and for its own bid evaluation. In general, it is expected that bid prices will be lower if suppliers have more time to prepare bids, and more time and schedule flexibility in the execution phase. On the other hand, the Customer's ability to find a good set of bids is dependent on the time allocated to bid evaluation, as we shall see.

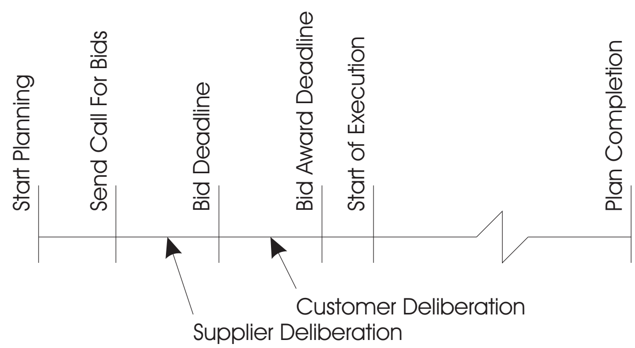

Fig. 1. Agent Interaction Timeline

3 Experimental Setup

The experimental setup includes three main components: a MAGNET Server as described in [5], a Customer agent that generates plans, requests bids, and evaluates bids, and a Supplier agent that generates and submits bids. The bid evaluation process is instrumented to measure the rate of improvement for various search strategies. Random variable seeds are controlled to ensure that different search strategies are presented with exactly the same problems.

3.1 Customer Agent: Generate Plan and Issue Call for Bids

The Customer agent generates a plan, which consists of a set of tasks with their precedence relations, and issues a call-for-bids. For each task in the call-for-bids the following must be specified:

- a time window, consisting of an earliest start time $t_{es}(s)$ and a latest finish time $t_{lf}(s)$,
- a set of precedence relationships for the task.

We assume the agent has general knowledge of normal durations of tasks. One of the roles of the MAGNET market infrastructure is to gather this information. In order to minimize bid prices while minimizing the overall duration of the plan, the Customer agent schedules tasks ahead of time using expected durations, computing early start and late finish times using the Critical Path (CPM) algorithm [11]. The minimum duration of the entire plan is called the *makespan* of the plan. The difference between t_{goal} and the earliest finish time $t_0 + \mathit{makespan}$ computed by CPM is called the *total slack* of the plan. The trade-off between minimizing plan duration and attracting usable bids from suppliers affects how slack should be set.

For these experiments, plans are randomly-generated task networks of 50 tasks, with a fixed average connectivity (sum of predecessors and successors) that is high enough to make schedule feasibility a significant factor in the evaluation. We arbitrarily set total slack to 10% of the expected makespan.

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. For our experimental conditions, we relax the time windows by 40%, an amount that has been shown to give a good balance between supplier flexibility and the need of the Customer agent to be able to compose feasible plans [4]. This is done by setting t_{goal} as discussed above, reducing the task durations to 60% of their expected values, and re-running the CPM algorithm on the task graph. Figure 2 shows in Gantt chart form how such a plan would be specified by a Customer. The tasks have expected durations of 1.0, and the plan has an overall slack of 10%, giving a target finish time of 3.3. The call-for-bids is specified with task durations of 0.6. The arrows show precedence relationships, and the extension bars show per-task slack. Task 2 is labeled to show the early start and finish and late finish times.

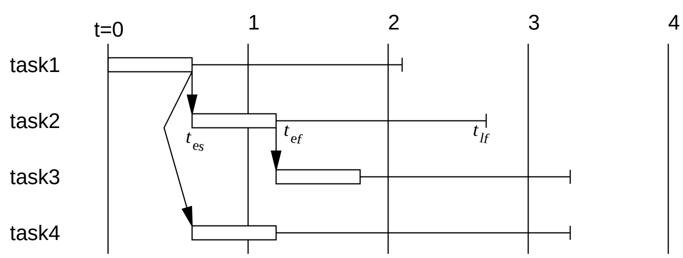

Fig. 2. Plan as specified by Customer

3.2 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. For these experiments, 100 bids were generated for each run.

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, along with prices for each task in the set. The difference between the overall bid price and the sum of the individual task prices is referred to as the *discount* or the *premium* of the combination. In addition, the early start time, late finish time, and maximum duration are specified for each task. 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. Task durations are randomly distributed around the expected value used by the Customer, and the time windows are randomly set to be smaller than the time windows specified in the call-for-bids and larger than the computed durations. Because of this, arbitrary combinations of bids frequently generate infeasible schedules.

Bids are generated for random sets of contiguous tasks within the call-for-bids. Full details on the bid generation process are given in [4]. Previous work [20] has shown that the size of bids can have a significant impact on the difficulty of the search problem; intuitively, larger bids are harder to compose together because there is a higher likelihood of overlap.

3.3 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 provides coverage of all tasks, allows for a feasible schedule, and minimizes a combination of cost and risk. Since bids are exclusive OR (when multiple bids are submitted by the same agent, only one of them can be accepted) for each bid the Customer has the option of selecting which parts of the bid (if any) to accept. Bids for combinations might include a *discount* or a *premium* for accepting the whole bid.

Timing information for a *bid* and a *task* includes early start $bidt_{es}(task)$, late finish $bid.t_{lf}(task)$, and duration $bid.duration(task)$ for each task in the bid. The semantics of a bid is that a supplier is willing to perform the task *task* for the bid price $bid.price(task)$ starting at any time $t_s(task)$ such that $bid.t_{es}(task) \leq t_s(s) \leq bid.t_{lf}(task) - bid.duration(task)$, and finishing at time $t_s(s) + bid.duration(task)$.

The evaluator uses a generalized simulated annealing search as described below. Nodes are kept in an ordered queue of fixed maximum length (the *beam width*), sorted by the node's value. We use an exponential distribution for selecting the next node, as required by simulated annealing [15]. The algorithm is outlined in Figure 3.

The following bid-selection methods, used in Step 2.2 of the algorithm in Figure 3, have been implemented and tested. Note that the feasibility and cost improvement methods have significant complexity costs associated with them.

- *Random Bid, Random Bid Component*: Choose a bid or a bid component at random, and attempt to add it to the node. The ratio of bids to bid components is adjustable. This method is fast ($O(1)$) and promotes general exploration of the search space.
- *Coverage Improvement*: Choose a bid or bid component that covers a task that is not mapped in the node. The probability of choosing a bid component

Algorithm: Simulated Annealing

1. **Initialize Search**:
   1.1 **Pre-process bids**: For each task generate a list of the bids that include the task and its average bid price.
   1.2 **Coverage test**: If any task has no bids, exit (coverage cannot be achieved).
   1.3 **Single bid test**: If any task has a single bid, then the bid must be part of any solution. The bid might contain just that task or more tasks. Create node(s) that map the bid, compute their value V, and add them to the queue.
   1.4 **Initialize queue**: If there were no singletons then create a node mapping all tasks to no bids, and add it to the queue.
   1.5 **Set initial annealing temperature**.
2. **while not** timeout **and** improving **do**:
   2.1 **Select a node N for expansion**:
   Select a random number $R = rT(V_{max} - V_{min})$, where
      - r is an exponentially distributed random number with a mean of 1.0,
      - T is the current annealing temperature, and
      - V_i is the value of node i in the queue.

   Choose node N as the first node in the queue for which $V_N < R + V_1$ where V_1 is the value of the first node in the queue.
   2.2 **Select a bid B**:
   Discard all bids that appear on the tabu list of N.
   Discard all bids that have already been used to expand N.
   Choose a bid according to the current bid selection policy.
   2.3 **Expand node N with bid B, producing node N'**:
   For each task mapped by bid B that was already mapped to bid B', remove bid B'.
   If B' was a singleton bid, abandon the expansion.
   Add B to the expansions-tried list of node N.
   Copy the tabu list of node N into node N' and add bid B in front.
   Truncate the tabu list in node N' to the tabu size limit.
   2.4 **Evaluate node N'**:
   $V_N = Cost_N + Risk_N + Feas_N + Cov_N$. where
      - $Cost_N$ is the sum of bid prices (uncovered tasks are assigned an average price),
      - $Risk_N$ is the expected cost of recovering from plan failure, times a weighting factor.
      - $Feas_N$ is the weighted sum of schedule overlaps.
      - Cov_N is the number of tasks that are not mapped to a bid, times a weighting factor.

   2.5 **Update best-node statistics**.
   2.6 **Adjust the annealing temperature T**.
3. **return** the best node found.

Fig. 3. The Simulated Annealing algorithm used to evaluate bids.

is equal to the coverage factor of the node. If a bid component is chosen, the coverage factor will increase; if a bid is chosen, the probability of improving coverage is $P_{ci} > 0.5$. This method is also $O(1)$ if the set of unmapped tasks and the set of bids per task is stored.

- *Feasibility Improvement*: The mapping is scanned to find tasks that have negative slack $bid.t_{es} + duration_a > bid.t_{lf}$, are constrained by their bids rather than by predecessors or successors, and could be moved in a direction that would relieve the negative slack. They are sorted by their potential to reduce infeasibility, and saved. The untried bid or bid component with the highest potential to reduce infeasibility is chosen. Note that when a bid is chosen, there is no guarantee that it will not introduce other infeasibilities. The complexity of this method is $O(xy)$, where x is the number of tasks in the plan and y is the number of tasks in the mapping that meet the above improvement criteria, incurred once the first time a node is subjected to feasibility improvement, and $O(z)$, where z is the number of bids that could potentially be mapped to a task, each time a feasibility improvement is attempted on a node.
- *Cost Improvement*: Choose the (untried) bid or bid component that is responsible for the maximum positive deviation from the average price, and replace it with a lower-priced bid that covers at least the task with the highest positive cost deviation. The first time this method is applied to a node, it has a complexity of $O(xy + z)$, where x is the number of bids mapped to a node, y is the number of components (tasks) in a bid, and z is the number of potential bids per task. Subsequent expansions of the same node by this method incur a complexity of only $O(z)$.

These selectors can be composed together and used to generate focused improvement for a given node. Four selectors were used in our experiments, as follows:

- *Random*: The random selector described above.
- *FeasCov*: If the node is infeasible, use the feasibility improvement selector; otherwise if it is not fully covered, use the coverage improvement selector; otherwise use the random selector.
- *CostFeasCov*: If the cost of the covered portion of the node is above average, attempt to reduce its cost; otherwise use the *FeasCov* selector.
- *Combined*: Run the *Random* selector as long as it produces improvement, then switch to *Feasibility Improvement* until that fails to produce improvement, then switch back to *Random*, then to *Coverage Improvement*, then back to *Random*, then to *CostFeasCov*, and finally back to *Random*.

4 Experimental Results

In this study, we are attempting to learn whether the simulated-annealing search technique can be effectively applied to the bid-evaluation problem. The results show that the *Combined* selector outperforms all the others and can be used effectively on moderate-sized problems, and that the more focused methods alone are ineffective, even if used with high annealing temperatures. It seems that excess focus on improvement leads to faster improvement early on, at the cost of a lower likelihood of finding a solution that satisfies all constraints.

In order to probe a range of problem complexity factors, we ran the *Random*, *Cov*, *FeasCov*, and *Combined* selectors against two different problem types of the same size but different levels of complexity. Both of them contain 50 tasks and 100 bidders, and all are generated with the same random number sequences. In the *small-bid* problem, the average bid size (number of tasks included in a discounted bid) is 5, and in the *large-bid* problem, the average bid size is 15. Earlier work [20] has shown that this difference has a significant impact on the search difficulty due to the greater probability of overlap among bids.

Figure 4 shows the improvement curves for the four bid selectors on the *small-bid* problem, and Figure 5 shows improvement curves for the same selectors on the *large-bid* problem. Error bars show $st.dev/\sqrt{2}0$. The *Combined* selector clearly gives the best overall performance, both in terms of solution quality and in terms of consistency.

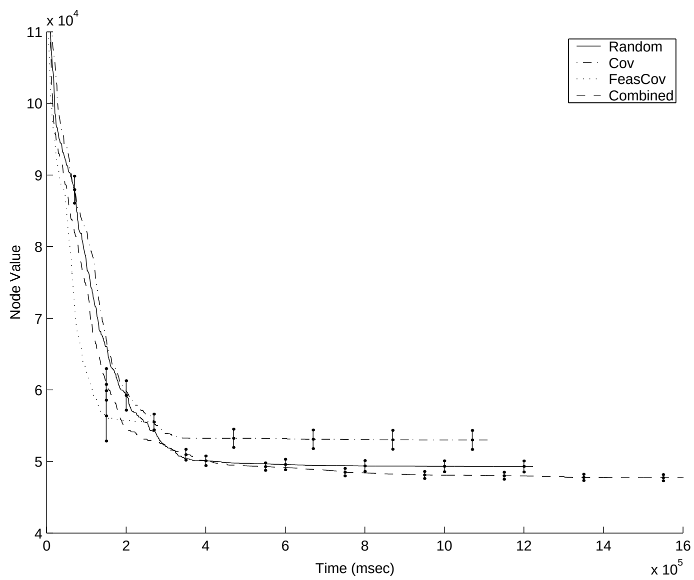

Fig. 4. Improvement curves for the *small-bid* problem. Averages are shown for 20 runs.

The following table shows the number of acceptable assignments found for the *small-bid* and *large-bid* problems. The table shows how effective the four selectors were at finding solutions that satisfied all constraints. The actual number of such solutions is not known. Again, we see the advantage of the *Combined* selector,

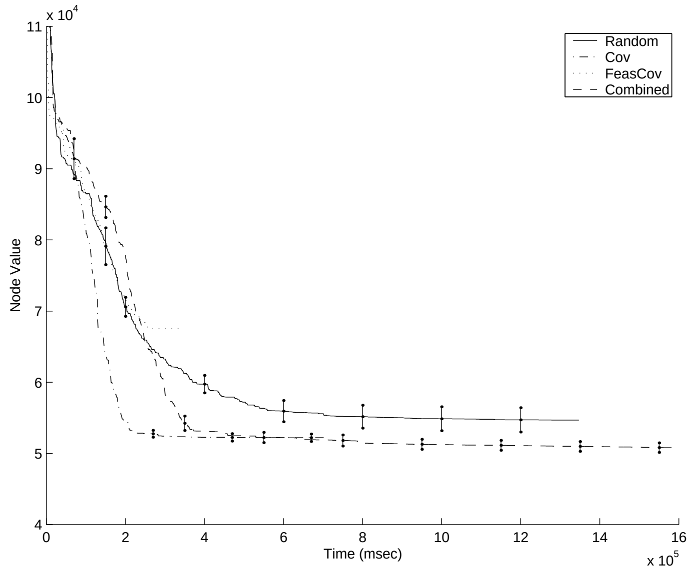

Fig. 5. Improvement curves for the *large-bid* problem. Averages are shown for 20 runs.

which uses random selection to generate sets of candidates, and then switches to more focused selectors to clean up.

Selector	small-bid problem	large-bid problem
Random	2	2
Cov	3	0
FeasCov	2	0
Combined	6	1

In Figure 6, we explore the effect of raising the annealing temperature on the performance of the selectors. The experiments described earlier are all run with an initial annealing temperature of 0.3. Raising the annealing temperature does not have any significant effect on performance, focused selectors do not perform any better at higher temperatures.

Finally, we compared the performance of the Simulated Annealing search with a systematic search on a series of 20 smaller problems with 10 tasks and 10 bidders. This was the largest problem on which we could run a fully systematic search in less than 10 hours. Six of these problems had solutions. As the following table shows, the *Combined* selector performs respectably, finding solutions to all but one of the problems in a few seconds. The "solutions/problem" column shows the number of distinct solutions found for the problems that were solved, and the "best value" column compares the average values of the best solutions found for each method.

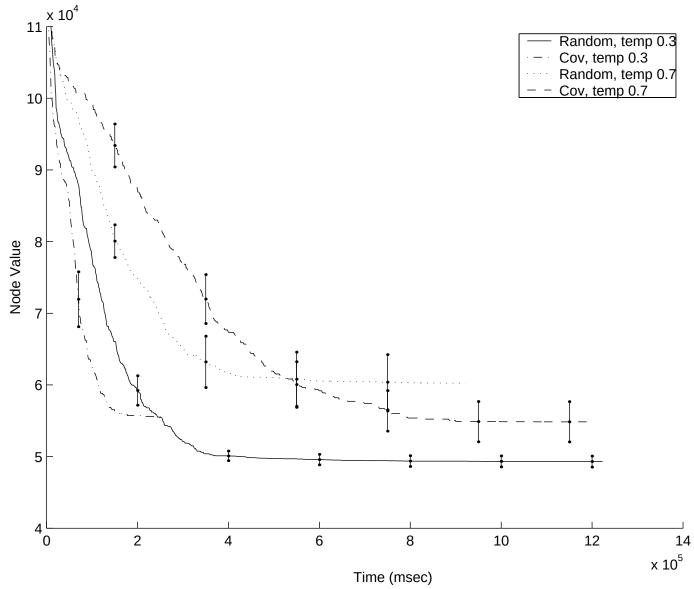

Fig. 6. Improvement curves for two different annealing temperatures.

Selector	problems solved	solutions/problem	best value found
Random	3	10.3	9392
Combined	5	32.2	9244
Systematic	6	172.5	8994

5 Related Work

Markets play an essential role in the economy, and there is a growing need for agents capable of sophisticated automated negotiations [1,8]. Automated contracting protocols either assume direct agent-to-agent negotiation [19,6] or an auction mechanism [21]. Sandholm [19] 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. Our agents are heterogeneous, and decide what tasks to handle by responding to a call for bids that requires specific tasks or products within a specified time window.

The determination of winners of combinatorial auctions [13] is hard. Dynamic programming [16] works well for relatively small sets of bids, but does not scale well and imposes significant restrictions on the bids. Sandholm [18] relaxes some of the restrictions, but assumes superadditivity and no exclusive-OR bids. For more general settings, optimality of solution is traded off for computational cost by using polynomial time approximation algorithms.

Since the introduction of iterative sampling [12], a strategy that randomly explores different paths in a search tree, there have been numerous attempts to improve search performance by using randomization. Randomization has been shown to be useful in reducing the unpredictability in the running time of complete search algorithms [7]. A variety of methods that combine randomization with heuristics have been proposed, such as Least Discrepancy Search [9], heuristic-biased stochastic sampling [3], and stochastic procedures for generating feasible schedules [14], just to name a few. The algorithm we presented is based on simulated annealing, and as such combines the advantages of heuristically guided search with some random search.

6 Conclusions and Future Work

Bid evaluation in the MAGNET automated contracting environment is a difficult optimizing search problem that must be performed within a predetermined amount of time. Ignoring the use of individual bid components, in our experiment there are approximately 10^{14} bid combinations that could be tested. We have chosen the Simulated Annealing approach to drive broad exploration of the search space. For this to be effective, it is necessary to set a number of parameters, including the beam width, annealing temperature and rate, and penalty factors, to avoid placing large potential barriers around the solutions. Eventually, we hope to be able to use problem metrics to set these parameters on the fly.

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 focus and ultimate success on this large, multicriterion search problem. Time constraints dictate that only a tiny fraction of the search space can be explored, and local minima abound. The result was that, in this environment, the purely random approach was more successful at finding solutions that satisfied all constraints than were the more informed search techniques, and an interleaved combination of random and focused selectors produced the best results.

This work raises several interesting questions for future research. Work is required to develop a clear understanding of how the various tuning parameters should be adjusted in accordance with problem parameters. An ideal tuning would reduce the incidence and size of local minima, and minimize potential barriers around usable solutions. Variations on the interleaved random and focused searches need to be explored. At a higher level, the scheduling of planning time, bid preparation time, bid evaluation time, and execution time needs to be driven by knowledge of the relative contributions of each of those components to overall solution quality.

References

1. Carrie Beam and Arie Segev. Automated negotiations: A survey of the state of the art. CITM Working Paper 96-WP-1022, Haas School of Business, Univesity of California, Berkeley, 1997.

2. Mark Boddy and Thomas Dean. Solving time-dependent planning problems. In *IJCAI89*, 1989.
3. J. Bresina. Heuristic-biased stochastic sampling. In *Proc. of the Thirteenth Nat'l Conf. on Artificial Intelligence*, 1996.
4. John Collins, Maksim Tsvetovat, Rashmi Sundareswara, Joshua Van Tonder, Maria Gini, and Bamshad Mobasher. Evaluating risk: Flexibility and feasibility in multi-agent contracting. Technical Report 99-001, University of Minnesota, Dept of CSE, Minneapolis, Minnesota, February 1999.
5. John Collins, Ben Youngdahl, Scott Jamison, Bamshad Mobasher, and Maria Gini. A market architecture for multi-agent contracting. In *Proc. of the Second Int'l Conf. on Autonomous Agents*, pages 285–292, May 1998.
6. Peyman Faratin, Carles Sierra, and Nick R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1997.
7. Carla Gomes, Bart Selman, and Henry Kautz. Boosting combinatorial search through randomization. In *Proc. of the Fifteen Nat'l Conf. on Artificial Intelligence*, pages 431–437, 1998.
8. Robert H. Guttman, Alexandros G. Moukas, and Pattie Maes. Agent-mediated electronic commerce: a survey. *Knowledge Engineering Review*, June 1998.
9. William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *Proc. of the 14th Joint Conf. on Artificial Intelligence*, pages 607–613, 1995.
10. S. Helper. How much has really changed between us manufacturers and their suppliers. *Sloan Management Review*, 32(4):15–28, 1991.
11. Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research.* McGraw-Hill, 1990.
12. Pat Langley. Systematic and nonsystematic search strategies. In *Proc. Int'l Conf. on AI Planning Systems*, pages 145–152, College Park, Md, 1992.
13. R. McAfee and P. J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.
14. Angelo Oddi and Stephen F. Smith. Stochastic procedures for generating feasible schedules. In *Proc. of the Fourteen Nat'l Conf. on Artificial Intelligence*, pages 308–314, 1997.
15. Colin R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* John Wiley & Sons, New York, NY, 1993.
16. Michael H. Rothkopf, Alexander Pekeč, and Ronald M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
17. Stuart J. Russell and Shlomo Zilberstein. Composing real-time systems. In *Proc. of the Ninth Nat'l Conf. on Artificial Intelligence*, volume 1, pages 212–217, Sydney, Australia, August 1991.
18. Tuomas Sandholm. Approaches to winner determination in combinatorial auctions. *Decision Support Systems*, 1999.
19. Tuomas Sandholm and Victor Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *1st Int'l Conf. on Multiagent Systems*, pages 328–335, San Francisco, 1995.
20. Erik Steinmetz, John Collins, Maria Gini, and Bamshad Mobasher. An efficient algorithm for multiple-component bid selection in automated contracting. In *Agent Mediated Electronic Trading*, volume LNAI1571, pages 105–125. Springer-Verlag, 1998.
21. P. R. Wurman, W. E. Walsh, and M. P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24(1):17–27, 1998.