Contract Types for Satisficing Task Allocation: II Experimental Results

Martin R. Andersson and Tuomas W. Sandholm[1]
{mra, sandholm}@cs.wustl.edu
Department of Computer Science,
Washington University
One Brookings Drive
St. Louis, MO 63130-4899

Abstract

We provide experimental results for a task allocation problem where self-interested, individually rational agents (re)contract tasks among themselves. Traditional contract types allow only one task to be transferred between agents at a time (original contracts). In this paper the original and four other contract types are studied: cluster-, swap-, multiagent, and OCSM-contracts. The OCSM-contracts will reach the global optimum even if the agents are individually rational, but in large-scale problems the number of steps required can be prohibitively large, albeit finite. In such cases it is more important to find the best solution reachable in a bounded amount of time. To construct algorithms that achieve that we study different contract types evaluate their performance.

This paper discusses the quality of local optima reached by the different contract types, and how quickly they are reached. It is shown how environmental characteristics such as the number of agents and the number of tasks affect these results. This analysis is used as a basis for making prescriptions about which contract types agents should use in different environments. Out of original-, cluster-, swap-, and multiagent-contracts, either original-contracts (if the ratio agents to tasks is great) or cluster-contracts (if the same ratio is small) reach a local optimum with a higher social welfare than the others.

Introduction

The importance of automated negotiation systems is increasing as a consequence of the development of technology as well as increased application pull, e.g, vehicle routing systems (Sandholm 1993) and electronic commerce (Sandholm & Ygge 1997). A central part of such systems is the ability for the agents to reallocate their tasks. Tasks can interact positively or negatively with each other, so they are preferably handled by the same agent or by different agents, respectively. The agents also have different resources that lead to different costs for handling the various tasks and it is possible that not every agent is capable of handling types of tasks.[2]

The contract type most commonly used in multiagent contracting systems only allows for one task to move from one agent to another at a time (Sen 1993)(Smith 1980). We will refer to this type of contract as *original* (O-contracts). By allowing more than one task to be transferred in the same contract, more efficient contracting can be achieved. Hence, four new types of contracts have been recently introduced (Sandholm 1996)(Sandholm 1997a)(Sandholm 1997b)(Sandholm 1998): *cluster-* (C), *swap-* (S), and *multiagent-contracts*, (M) as well as a combination of all three of the above with the original contract (*OCSM-contracts*). The C-contracts[3] transfer at least two tasks from one agent to another, while the S-contracts let two agents swap tasks with each other (one task is transferred from each agent to the other agent). In the M-contracts[4] at least three tasks are being transferred between at least three agents. Each agent gives away only one task, but can receive more than one task. All the characteristics in the contracts above have been combined to form the OCSM-contract. That is, any number of tasks can be transferred between any number of agents in one single contract. Any time a contract is performed (tasks are transferred) agents which take on tasks can incur side payments to cover their extra expenses. OCSM-contracts guarantee that a global op-

[1]Supported by NSF CAREER award IRI-9703122 and NSF grant IRI-9610122.

[2]The dependencies between tasks in human negotiations are discussed by Raiffa in (Raiffa 1982). The concepts of linkage and log-rolling are also presented, which are similar to swapping tasks and clustering of tasks.

[3]In the auction of airwave bandwidth, the Federal Communication Commission used a simultaneous ascending auction to provide for the bidders the possibility to cluster the frequencies for which they were bidding, without explicit cluster contracts (McAfee & McMillan 1996).

[4]Sathi and Fox (1989) (Sathi & Fox 1989) studied a simpler version of multiagent-contracts where bids were grouped into cascades.

timum is reached in a finite number of steps (contracts), when used among myopically individually rational agents, irrespective of the order in which contracts are proposed and made (Sandholm 1993)(Sandholm 1996)(Sandholm 1997a). Although this is a powerful result for small problem instances, in large-scale problems the number of steps needed to reach the global optimum may be impractically large. In these problems it is more important to obtain the best achievable solution in a bounded amount of time.[5]

To examine the difference in contracting behavior and reachable optima for the different contract types, we constructed a *multiagent version* of the optimization Traveling Salesman Problem (TSP). The TSP was used as an example domain because it is NP-complete, and the space of task allocations contains many local optima when using hill-climbing-based contracting algorithms (Sandholm 1997a)(Sandholm 1993). Another advantage of the TSP is its structural simplicity, which provides for repeatability and presentability without unnecessary contextual overhead.

The multiagent TSP is defined as follows: several salesmen will visit several cities in a world that consists of a unit square (square with sides of length one). Each city must be visited by exactly one salesman, and each salesman must return to his starting location after visiting the cities assigned to him. A salesman can visit the cities assigned to him in any order.

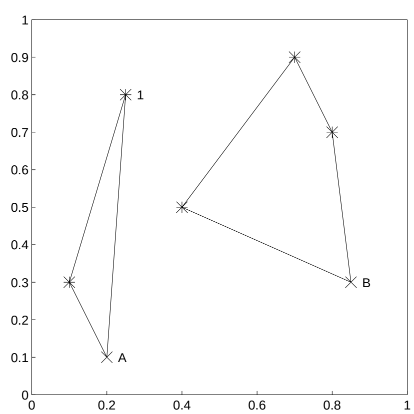

Figure 1: *An example problem instance of a multiagent TSP consisting of five cities (*) and two salesmen (X). If salesman A contracts out city 1 to salesman B, the social welfare will increase due to less travel, i.e., lower costs.*

The objective of the salesmen is to minimize their own total cost, *i.e.*, distance traveled. The outcome of the system is the social welfare, *i.e.*, the negative of the sum of all the agents' costs. Initially the location of the cities and starting points of the salesmen are chosen randomly as is each salesman's initial assignment of cities to visit. After this initial assignment the salesmen can exchange cities with each other. The salesmen are *myopically individually rational*, which means that they agree to a contract if and only if the contract increases the agent's immediate payoff. An agent's payoff consists of the side payments received from other agents (for handling their tasks) minus the travel cost, c_i, of handling tasks. Therefore the sum of all salesmen's total distance traveled decreases monotonically, i.e, the social welfare increases monotonically. In the following discussion the general multiagent task allocation terms, "agent" and "task", are used interchangeably with the TSP specific terms "salesman" and "city".

The Contracting System

In principle our implementation of a contracting system can solve the multiagent optimization TSP for any number of agents and tasks. In the simulations, we studied problem instances with up to eight agents and eight tasks. These restrictions were set in order to limit the computation time. For all combinations of numbers of agents between two and eight, and numbers of tasks between two and eight, 1000 randomly generated TSP instances were solved.[6] Each problem instance was solved in five ways: using a myopically individually rational (*i.e.* hill-climbing) contracting method for the O-, C-, S-, and M-contracts, respectively, and an exhaustive enumeration of task allocations in order to find the globally optimal allocation (this corresponds to the outcome reached via myopically individually rational contracting using OCSM-contracts).

The cost c_{qr} of traveling between locations q and r (either city or location of salesman) equals the Euclidean distance between the locations. The total cost each salesman incurs, c_i, when visiting his cities, is the sum of the costs along his tour:

$$c_i = \sum_{q,r \in \text{The tour of salesman } i} c_{qr}$$

The objective of agent i is to maximize payments received from others (for handling their tasks) minus the agent's own total cost c_i. The social welfare is given by the negative of the sum of all agents' total costs, c_i. The side payments do not affect the social

[5] A more detailed discussion of these new contract types, and the exact definitions of the contracts used in this contracting approach can be found in (Sandholm 1997a).

[6] For the combination consisting of 8 agents and 8 tasks, 500 instances were used.

welfare as they are merely redistributing wealth among the agents.

In the experiments, each problem instance was tackled in two phases: first all possible TSPs were solved,[7] and then experiments with different solution approaches to the task allocation problem were conducted. This way the agents did not have to recalculate the TSPs every time a different contracting algorithm was tried on the same problem instance. In order to provide all the data necessary for determining the individually rational contracts used when contracting, the TSPs of all the possible combinations of cities together with any one salesmen were solved.[8]

When searching for a good task allocation with the O-, C-, S-, and M-contracts, all possible combinations of agents and tasks that are applicable for that contract type are tried repeatedly. If no changes in the task allocation have occurred during a period when all contracts (of a given type; O, C, S, or M) have been attempted for all combinations of agents and tasks applicable, the local optimum achievable with that contract type has been reached.

A complete description of the contracting system can be found in (Andersson & Sandholm 1997).

Results

To be able to compare the different contract types, the ratio bound (ratio of the welfare of the obtained local optimum for a given contract type to the welfare of the global optimum) was used. The mean ratio bounds (over the 1000 problem instances) were calculated for all possible combinations of numbers of agents and numbers of tasks. The differences of the ratio bounds between the contract types were also calculated, from which the significance of the results could be statistically analyzed.

Evaluation Criteria

The description below concerns a fixed number of agents and a fixed number of tasks. For all numbers of agents and tasks, mean ratio bounds and the mean of the difference in ratio bounds were calculated. Let x_j^l denote the social welfare (sum of all the salesmen's total traveling distances) of the jth problem instance, $j \in 1, \ldots, n$ (n=1000), after task reallocation has been performed until a local optimum has been reached using contract type $l \in$ {O,C,S,M,G}, where G indicates the global optimum (or equivalently OCSM-contracts). The social welfare of the local optimum obtained using l-contracts, x_j^l, over the social welfare of the global optimum, x_j^G gives the ratio bound $r_j^l = \frac{x_j^l}{x_j^G}$ for the jth problem instance using l-contracts. The difference in ratio bounds between two different contract types applied to the same instance j, is given by:

$$r_j^{kl} = r_j^k - r_j^l$$

The mean difference, $\overline{r}^{kl}$ between the contract types is:

$$\overline{r}^{kl} = \frac{1}{n} \sum_{j=1}^{n} r_j^{kl}$$

The mean ratio bound of contract type $l \in$ {O,C,S,M} is:

$$\overline{r}^{l} = \frac{1}{n} \sum_{j=1}^{n} r_j^{l}$$

Comparison of Social Welfare

This section presents a comparison of the means of the ratio bounds, $\overline{r}^l$, and the mean differences of ratio bounds, $\overline{r}^{kl}$, for the different contract types used to solve the problem. Compared to the other contract types, the mean ratio bound for the O-contracts, $\overline{r}^O$, does not vary as much in the number of agents or tasks. The ratio bound increases slightly with both the numbers of agents (Figure 2) and the number of tasks (Figure 3). The ratio bound for O-contracts varies between 1.1 and 1.2, which means that the social welfare using O-contracts is 10% - 20% off the global optimum.

As the number of tasks increases, the mean ratio bound, $\overline{r}^C$, for C-contracts decreases (Figure 2), *i.e.*, using C-contracts leads to local optima closer to the global optimum when the number of tasks is large. While the decrease is monotonic, it is greatest for small numbers of tasks. The ratio bound increases as the number of agents increases (Figure 2). This is especially noticeable in the cases with few tasks (2-5). For greater numbers of tasks, the increase in the ratio bound is smaller (Figure 2, bottom left). The mean ratio bound, $\overline{r}^S$, for S-contracts also decreases with the number of tasks, and increases with the number of agents. However, for S-contracts the increase in the ratio bound is considerable even for large numbers of tasks. As expected, M-contracts perform better both when the number of agents increases (Figure 2) and

[7] The IDA* search algorithm (Korf 1985) was used to solve the optimization TSPs. To ensure that the optimal solution was reached an admissible $\hat{h}$-function was used. It was constructed by under estimating the cost function of the remaining nodes by the minimum spanning tree (Cormen, Leiserson, & Rivest 1990) of those nodes (that is, of nodes not yet on that path of the search tree, the last city of that path of the search tree, and the finish (=start) location of the salesman).

[8] Salesman 1 visits city 1, salesman 1 visits city 2, ..., salesman 1 visits cities 1 and 2, ..., salesman 2 visits city 1, ..., salesman 8 visits all eight cities.

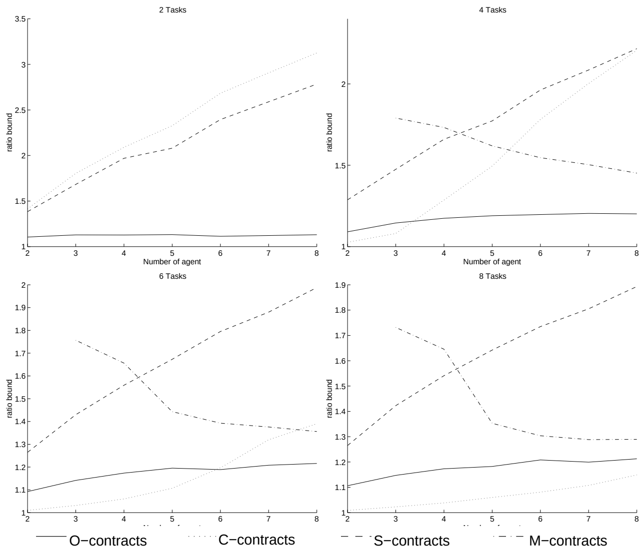

Figure 2: *The ratio bounds as a function of the number of agents: (top left) 2 tasks, (top right) 4 tasks, (bottom left) 6 tasks, and (bottom right) 8 tasks. The graphs for M-contracts do not include any values for two agents or two tasks, since at least three agents and three tasks are needed to perform an M-contract.*

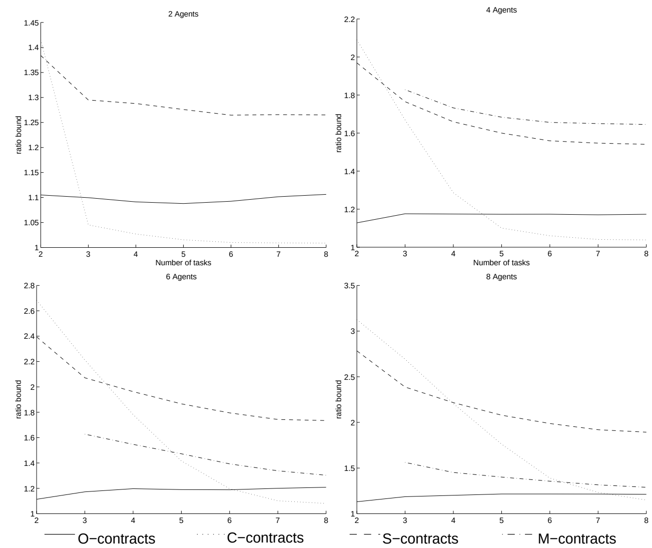

Figure 3: *The ratio bounds as a function of the number of tasks: (top left) 2 agents, (top right)4 agents, (bottom left) 6 agents, and (bottom right) 8 agents. The graphs for M-contracts do not include any values for two agents or two tasks, since at least three agents and three tasks are needed to perform an M-contract.*

when the number of tasks increases (Figure 3). In other words the mean ratio bound, $\overline{r}^M$, decreases with the number of tasks and agents. This is obvious in the bottom right graph in Figure 3. Extrapolating from these results one can expect M-contracts to reach a lower ratio bound than any of the other contract types for much greater numbers of agents and tasks than eight.

In Figure 4 it can be seen that O-contracts always perform better than S- and M-contracts. C-contracts provide a lower ratio bound than O-contracts when the number of tasks is greater than the number of agents. In fact, for those numbers of agents and tasks, C-contracts are the best contract type also when compared to S- and M- contracts. So, the top left graph in Figure 4 summarizes which of the O-, C-, S-, or M-contracts is the best one to use for different combinations of numbers of agents and tasks. The reason why the S-contracts are outperformed by all other contract types is that number of tasks allocated to each agent can not change when using S-contracts.

Computational Aspects

The number of contracts that has to be tried before one can be certain that a local optimum has been reached, varies noticeably between the different contract types, Figures 5 and 6. Both the numbers of contracts needed to be tried and those performed in order to reach a local optimum, increase with the number of tasks in the system. From Figure 6 we can conclude the number of contracts performed and tried by all the contract types increases sub-exponentially.

O-contracts perform on average the largest number of contracts for reaching the optimum, followed by C-contracts, S-contracts, and M-contracts. The fact that O-contracts only move one task in each contract is likely to contribute to this result. It is an interesting result since O-contracts are desirable for the reason that they require the least number of contracts to be tried to verify that a local optimum has been reached. On the other hand, O- and C-contracts need to try a larger number of contracts before reaching a local optimum, than S- and M-contracts. In the case of six agents and six tasks, O-contracts and C- contracts still need less than 100 contracts, to reach a local optimum, except in a small number of cases. With the exclusion of some exceptional cases where several thousand contracts are needed, M-contracts find local optima after a small numbers of contracts have been tried. This may be affected by the specific order of trying M-contracts. Note that the discussion above concerns the number of contracts needed to reach a local optimum, not the number of contracts needed to verify that the system has reached a local optimum. M-contracts need by far the greatest number of contracts to verify that the solution is a local optimum. Then, in order, C- contracts, S-contracts, and O-contracts follow. The CPU-time used for negotiation is proportional to the number of contracts tried, but the constant of proportionality

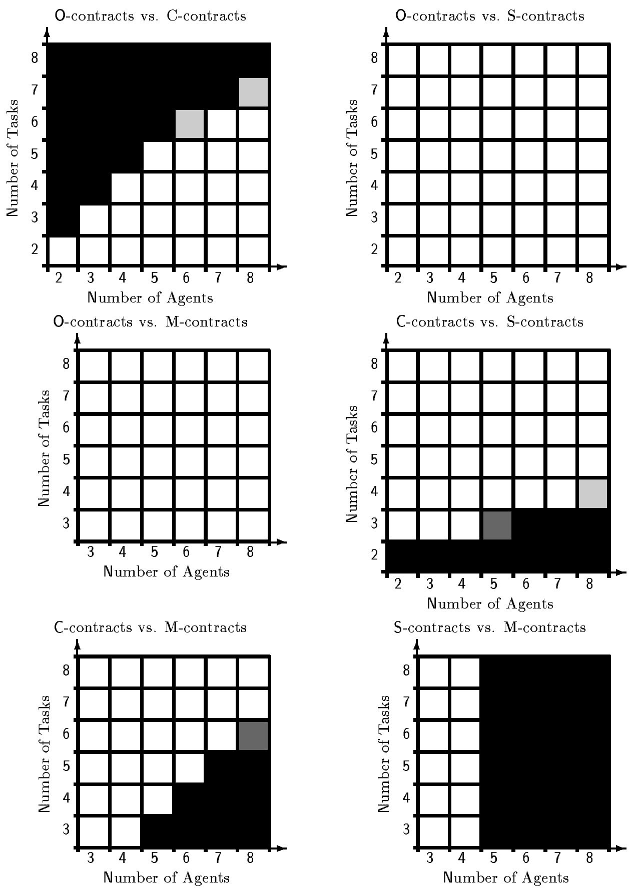

Figure 4: *Pairwise comparison of the differences in ratio bounds* $\overline{r}^{kl}$ *for O-, C-, S-, and M-contracts. Above each graph the two contract types under comparison are stated. The darker the color in a square is, the better the latter contract type. Black and white indicate areas where the difference in performance is most significant. The black and white areas represent results that are significant at the 0.05 confidence level of the mean difference ratio bounds in a paired t-test* (Cohen 1995). *The gray areas represent results that are not significant at the 0.05 level, yet one of the contracts is better. In the dark gray areas the latter contract is better while in the lighter gray areas the former is better. The graphs for M-contracts do not include any values for two agents or two tasks, since at least three agents and three tasks are needed to perform an M-contract.*

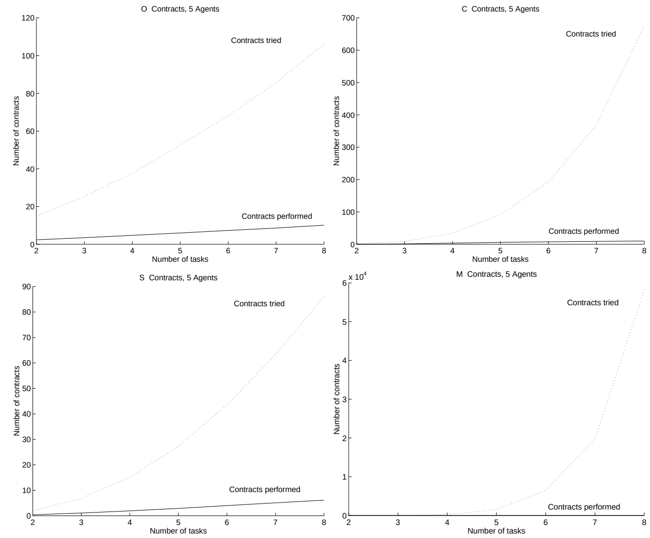

Figure 5: *For each contract type the number of contracts tried before the system reached a local optimum is represented by the dotted line (includes those contracts needed to verify that it is a local optimum). The number of contracts performed before reaching the local optimum is represented by the solid line. Note that the scales on the value axes are different.*

varies greatly between the contract types: it is much greater for M- contracts than O-, C-, and S-contracts. The reason is that M-contracts are more complicated than the other contract types, and that many contracts need to be checked before it can be verified that a local optimum has been reached.

Dynamics of Contracting

The typical final task allocations are very different between the contract types. C-contracts tend to concentrate the tasks to one or to a few agents, while O-contracts tend to spread the contracts to all the agents. Because of the sequencing of the M-contracts, the tasks tend to be allocated too often to agent number 1 [9]. The number of tasks per agent cannot change at all when S-contracts are used, which contributes to their poor performance. As is desired from an anytime contracting perspective, the contracts performed earlier often improved the social welfare more than the contracts performed later.

Conclusions

The current most widely used contract type allows for only one task at a time to be moved from one agent to another (O-contracts). Recently new contract types,

[9] One could avoid such anomalies by randomly picking contracts to try. However, a systematic scheme is necessary to verify that a local optimum has been reached.

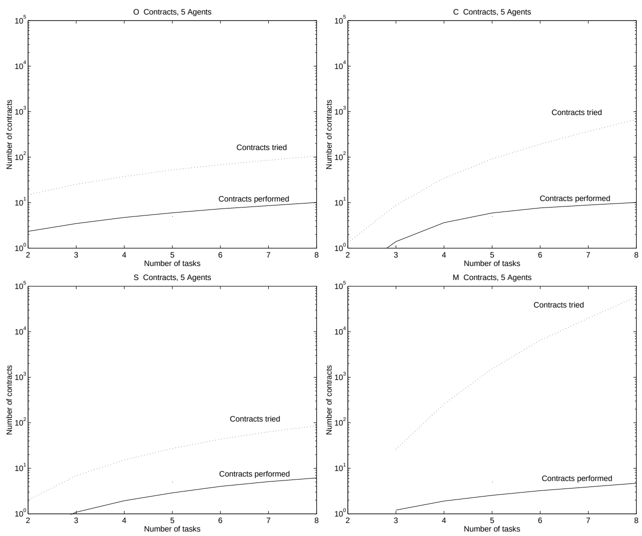

Figure 6: *For each contract type the number of contracts tried before the system reached a local optimum is represented by the dotted line (includes those contracts needed to verify that it is a local optimum). The number of contracts performed before reaching the local optimum is represented by the solid line. The y-axis uses a logarithmic scale.*

cluster (C), swap (S), multiagent (M), and OCSM-contracts (all the other contract types combined) have been introduced. They are all based on the idea of moving several tasks in a single contract, which reduces the number of local optima in the search space of task allocations for hill-climbing-based contracting algorithms.

OCSM-contracts guarantee that a global optimum is reached in a finite number of steps when used in any hill-climbing algorithm. Although this is a powerful result for small problem instances, in large-scale problems the number of steps needed to reach the global optimum may be impractically large. In these problems it is more important to obtain the best achievable solution in a bounded amount of time, than it is to reach the global optimum. To be able to construct algorithms which obtain the best achievable solution in a bounded amount of time, we compared the five contract types (all of which guarantee the optimal task allocation is reached) in an example problem (multiagent version of the TSP).

The results regarding the social welfare of the local optima of the different contract types provide guidelines to system builders regarding what contract types to use in different environments when computation is limited. We also presented timing results which can be used in the choice of contract type if there is not enough time to even reach a local optimum. In addition, our results help in the choice of contract type when certain properties of the final outcome are desired, *e.g.*, that tasks are distributed among multiple agents or that tasks are concentrated to just a small number of agents. For six agents and six tasks, a local optimum was reached within the first 100 contracts tried, with the exclusion of some exceptional cases. This is important since several hundred - sometimes several thousand - contracts often were tried before it could be verified that a local optimum had been reached. M-contracts reached a local optimum faster (when measured in the number of contracts tried or in the number of contracts performed before the optimum was reached) than the other contract types. However, M-contracts require more CPU-time per contract than O-, C-, or S- contracts. They also require a significantly larger number of contracts to be tried in order to verify that a local optimum has been reached. For these relatively small problem instances the mean of the social welfare of the O- and C- contracts proved to be closer to the global optimum (OCSM-contracts) than that of the S- and M- contracts. C-contracts performed best when the number of tasks was greater than the number of agents; otherwise O-contracts were best. Extrapolating to problems containing more agents and tasks, M-contracts obtain local optima closer to the global optimum. Despite the fact that O-contracts and C-contracts have similar values of social welfare of the local optima reached, the typical task allocations are very different: O-contracts tend to spread the tasks among all agents while C-contracts tend to concentrate the tasks to only one agent or a small number of agents.

The sequencing of contracts within a particular contract type influences the results. Analyzing this effect further is part of our future research. Also, to improve the social welfare, more than one contract type could be used during contracting. Further research is required to determine the best way to sequence the different contract types in order to obtain satisfactory social welfare with bounded computation. There are several possible approaches: change the contract type for every single contract, apply many possible contracts (maybe all) of one contract type before changing the type, or find a local optimum using one contract type before changing to another contract type. There is also the question of which of the contract types should be interleaved with each other. Yet another interesting area for future work is combining the different contract types, thus forming atomic contracts having characteristics of more than one of the O-, C-, S-, and M-contracts, but not all of them (unlike OCSM-contracts). These composite contract types would not guarantee that individually rational agents will reach the global optimal task allocation, but they would lead to a local optimum faster then OCSM-contracts, and to higher average social welfares than O-, C-, S-, or M-contracts individually.

References

Andersson, M., and Sandholm, T. W. 1997. Contract types for optimal task allocation: II experimental results. Technical Report WUCS-97-36, Washington University, Department of Computer Science.

Cohen, P. R. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press.

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms*. MIT Press.

Korf, R. E. 1985. Depth-first iterative-deepening: an optimal admissible tree search. *Artificial Intelligence* 27(1):97–109.

McAfee, R. P., and McMillan, J. 1996. Analyzing the airwaves auction. *Journal of Economic Perspectives* 10(1):159–175.

Raiffa, H. 1982. *The Art and Science of Negotiation*. Cambridge, Mass.: Harvard Univ. Press.

Sandholm, T. W., and Ygge, F. 1997. On the gains and losses of speculation in equilibrium markets. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 632–638.

Sandholm, T. W. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence*, 256–262.

Sandholm, T. W. 1996. *Negotiation among Self-Interested Computationally Limited Agents*. Ph.D. Dissertation, University of Massachusetts, Amherst.

Sandholm, T. W. 1997a. Contract types for optimal task allocation: I theoretical results. Technical Report WUCS-97-35, Washington University, Department of Computer Science.

Sandholm, T. W. 1997b. Necessary and sufficient contract types for optimal task allocation. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 87. Poster session abstracts.

Sandholm, T. W. 1998. Contract types for satisficing task allocation: I theoretical results. In *AAAI Spring Symposium Series: Satisficing Models*.

Sathi, A., and Fox, M. 1989. Constraint-directed negotiation of resource reallocations. In Huhns, M. N., and Gasser, L., eds., *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*. Pitman. chapter 8, 163–193.

Sen, S. 1993. *Tradeoffs in Contract-Based Distributed Scheduling*. Ph.D. Dissertation, Univ. of Michigan.

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