Anytime Coordination for Progressive Planning Agents

Abdel-Illah Mouaddib

CRIL-IUT de Lens-Université d'Artois
Rue de l'université, S.P. 16
62307 Lens Cedex, France
mouaddib@cril.univ-artois.fr

From: AAAI-99 Proceedings. Copyright © 1999, AAAI (www.aaai.org). All rights reserved.

Abstract

We address in this paper the problem of coordinating resource-bounded agents under time constraints in a dynamic environment. The agent society we deal with consists of coordinated agents each of which has a goal to achieve before a deadline and new agents can asynchronously appear to achieve time-constrained goals and to coordinate their plans with the already coordinated agent society. Agents use progressive planning that adapt the detail of their local plans according to local deadlines and available resources. The plan consists of a hierarchy of partial plans where each partial plan satisfies a part of the goal. In such environments, constructing a complete plan and then coordinating it with other agents doesn't guarantee that the planning and coordination operations will finish before the given deadline. What we propose is an anytime coordination that allows an agent to return a coordinated plan at any time by using series of partial planning followed by a coordination until the complete plan is constructed and coordinated or the deadline is met. This progressive plan merging operation is assessed in a resource allocation problem.

Introduction

Negotiation (Kraus, Wilkenfeld, & Zlotkin 1995) between distributed planning agents is imposed to handle potential interactions and solve conflicts. Cooperation is necessary when no agent has sufficient resources and information to solve a problem. When agents have a particular view on a problem independently of the other agents, inconsistency between local plans could arise.

Application domains concerned with such problems include the airport ground service scheduling (Neiman, Hildum, & Lesser 1994), resource allocation (Schwartz & Kraus 1997), cooperating robots (Mouaddib 1997), distributed scheduling (Sen & Durfee 1996; Mouaddib 1998) and others. Several coordination techniques (Durfee & Lesser 1987; Decker & Lesser 1992) have been developped for such applications where it is assumed that agents are not dynamic and with unlimited rationality. It is also assumed that the coordination overhead is negligible. The system with which we work consists of a set of agents performing in a common environment with limited resources and where the coordination cost can be time consuming.

We conduct basic research in a coordination technique for a resource-bounded agent society which supports limited rationality of agents and time-consuming planning and coordination operations. This technique is a progressive planning agent-based system where each agent determines progressively its local plan (Mouaddib 1997) and reacts dynamically to local plans of other agents. This planning technique adapts dynamically the detail of the plan to the available resources. Each local plan is represented with a hierarchy of partial local plans from the mandatory one to the optional ones. The coordination is based on series of partial planning cycles followed by negotiation cycles. After each partial planning cycle a partial plan is available for a coordination with the plans of the other agents. This strategy, through an iterative coordination, allows to minimize the backtracking and to reduce the costs of planning and coordination at unnecessarily detailed levels (Mouaddib 1997). This flexible coordination can be interrupted at any time and the plan merging operation of the new agents stops by accepting only the coordinated partial local plans.

The rest of the paper presents this approach, an analysis of its performance, first experimental results and its characteristics.

Cooperative Distributed Providers

We consider the application of distributed databases (DDB) which are located in different geographical areas and time-constrained transactions conveying queries on information stored on DDB sent to distributed providers that access to DDB to construct their response. Transactions use approximate query processing where transactions are logically splitted into mandatory subtransactions that convey queries to get the mandatory information and optional subtransactions to get optional informations. Each subtransaction gets its required information from *one* database. Providers should allocate the required database in order to coordinate their access and to avoid conflicts. Providers negotiate their database access. Clients sending requests have time-bounded connection in the net to formulate their requests and receive the required information. This scenario is inspired from an electronic commercial application (Sandholm & Lesser 1995) that can be seen as a time-constrained resource allocation problem.

Providers try to maximize the overall performance of the system by providing a high quality solution. The quality improvement, achieved through the progressive satisfaction of subtransactions, is along the degree of the relevance and the level of detail. The satisfaction of each subtransaction is assigned by a monetary charge that the client is willing to pay for the information. When providers cannot construct their optimal response under time pressure and resources constraints, they negotiate to borrow resources from the least important providers (low monetary gain). The providers are cooperative because they try to maximize the overall performance by maximizing the global rewarded monetary gain.

Cooperative Progressive Planning Agents

Progressive planning agents

Progressive planning (Mouaddib 1997) consists in presenting each local plan as a hierarchy of processing levels (satisfaction of a subtransaction) $P_\alpha = \{P_\alpha^1, P_\alpha^2, \ldots, P_\alpha^{n_\alpha}\}$. The agent α starts its processing by constructing the partial local plan P_α^1 achieving the mandatory part of the goal while the other partial local plans achieve the optional parts of the goal to refine the quality resolution. We assume that the goal is logically splitted into several parts from the mandatory one to the most optional. The agent ignores the optional parts of the goal when the resources (time, servers, etc ...) are not enough large to construct a complete plan. This processing allows first an agent to adapt the detail of its local plan to its deadline and to the available resources. Agents are cooperative and try to maximize the overall performance of the system. An agent can accept to borrow resources used by an optional partial plan to another agent requiring the same resource for a more important partial plan.

We improve the expressiveness of the plan by introducing the resources used by partial local plans in order to anticipate conflicts. We assume in our environment that each partial local plan uses only one resource. The extension to several resources consists in assuming lower granularity of partial local plans. Then the local plan of an agent α can be seen as follows: $P_\alpha = \{(P_\alpha^1, r_i, I_\alpha); (P_\alpha^2, r_j, J_\alpha); \ldots; (P_\alpha^d, r_k, K_\alpha)\}$. The plan P_α is constructed through series of partial planning cycles followed by negotiation cycles. We name in the following the Partial Local Plan (PLP) an element of the plan represented by the tuple *(actions, resource, interval)*. This tuple means that the PLP (the execution of the action *actions*) will use the resource *resource* during the interval *interval*. Mandatory PLP is the tuple $(P_\alpha^1, r_i, I_\alpha)$ while optional PLPs are the tuples $(P_\alpha^k, r_i, J_\alpha)$ where $k > 1$.

The agent constructs, first, the mandatory PLP and then coordinates it with the local plans of the other agents. Afterwards, it proceeds in the same way with the optional PLPs one by one.

Stating the coordination problem

Our scenario consists of a group $A = \{\alpha, \beta, \ldots, \gamma\}$ of n agents, an environment represented with its set of resources $R = \{r_1, \ldots, r_k\}$ and a set of goals planning { G_1, G_2, ..., G_n }. The coordination problem in this context is summarized with the following points: (1) Each agent has a goal to achieve before a deadline. (2) Each goal can be solved at different levels of detail. (3) Each agent uses a subset of available resources. And (4) A local plan consists of a hierarchy of PLPs where each one improves the quality of the solution of its predecessor. Furthermore, agents have to coordinate their local plans to avoid conflicts on resources. The coordination of the agent society consists of two constraints:

- Each resource r_j cannot be used by more than one agent at the same time.

$$\forall (P_\alpha^i, r_j, I_\alpha) \; and \; (P_\beta^j, r_j, I_\beta) : I_\alpha \; do \, not \; overlap \; I_\beta \quad (1)$$

- Each agent α should respect its deadline.

$$\forall r_j \in \mathcal{R},\ \alpha \in A : \max_{(P_\alpha^i, r_j, I_\alpha) \in P_\alpha} (end(I_\alpha)) \leq deadline(\alpha) \quad (2)$$

The strategy adopted to maintain these constraints respected, consists of an incremental merging PLP operation. The merging agent proceeds in series of partial planning cycles followed by negotiation cycles to construct and to coordinate its local plan with the other local plans of the agent society. This strategy stops as soon as the deadline of the merging agent is reached or a negotiation cycle fails (a merging current PLP operation fails). This strategy has real-time properties of anytime algorithms since it can be interrupted at any time and returns a solution consisting of the sequence of PLPs coordinated in previous cycles.

To apply to a cooperative distributed providers application, each database is a resource and a transaction is a time-bounded goal of a provider where its local plan consists of a set of tuples *(subtransaction, database, interval)* representing the subtransaction to process, the database to allocate and the interval of time during which the database will be used.

Anytime Coordination

Agent interactions: Contract-Net Protocol

The communication is based on a message-passing mechanism. A new agent, once the goal received, derives, first, the mandatory PLP. The agent broadcats, then, this plan to the other agents for coordination. These agents respond in turn only when a conflict occurs with their local plans by sending disagreement specifying the conflicts. To these disagreements, the new agent sends a resolution to the agents for avoiding conflicts. When these conflicts are solved and the PLP of the new agent is maintained, a new PLP is constructed for an optional part of the goal and a new coordination round is performed. This iterative processing is repeated until a PLP of the new agent is rejected or a deadline is met or all PLPs of the new agent are accepted.

Specifying and detecting conflicts

The merging agent receives several disagreements where each of which describes a conflict, the agent centralizes all these disagreements in order to optimally solve them.

Definition 1 *We say that the agent α is in conflict with an agent β when the merging PLP P^i_α of α shares the same resource at the same time with at least one PLP P^j_β of the agent β. Formally:*

$$\exists\, (P^i_\alpha,\ r_c, I_\alpha),\ (P^j_\beta,\ r_c, I_\beta) :\ I_\alpha\ overlaps\ I_\beta \qquad (3)$$

From this definition, the merging agent, that we name in the following α, classifies the collected conflicts in four categories (Mouaddib 1997). According to the category, an appropriate rule of resolution is adopted.

- **Hard critical conflict:** The mandatory PLP P^1_α of α is in conflict with the mandatory PLP P^1_β of an agent β.
- **Hard non-critical conflict:** The mandatory PLP P^1_α of α is in conflict with an optional PLP P^i_β ($i > 1$) of an agent β.
- **Soft critical conflict:** An optional PLP P^i_α ($i > 1$) of α is in conflict with the mandatory PLP P^1_β of an agent β.
- **Soft non-critical conflict:** An optional PLP P^i_α of α is in conflict with an optional PLP P^j_β of β (i and j $>$ 1).

When the merging agent α receives disagreements from other agents, it defines for each PLP P^i_α a set of PLPs $S_{P^i_\alpha}$ with which it is in conflict. The agent α classifies $S_{P^i_\alpha}$ in four categories according to type of the conflicts:

- $S_{P^i_\alpha}$ is a **hard critical set** when P^i_α is a mandatory PLP and a hard critical conflict occurs with one element of the set $S_{P^i_\alpha}$.
- $S_{P^i_\alpha}$ is a **hard non-critical set** when P^i_α is a mandatory PLP and only hard non-critical conflicts occur with the elements of the set $S_{P^i_\alpha}$.
- $S_{P^i_\alpha}$ is a **soft critical set** when P^i_α is an optional local plan and soft critical conflicts occur with one element of the set $S_{P^i_\alpha}$.
- $S_{P^i_\alpha}$ is a **soft non-critical set** when P^i_α is an optional PLP and only soft non-critical conflicts occur with the elements of the set $S_{P^i_\alpha}$.

The detection and type of conflicts are up to the agents sending the disagreement while the agent α, receiving disagreements, is in charge of progressively resolving them. Decentralizing the detection and type of conflict can participate in reducing the time-consuming process of negotiation while centralizing their resolution can be a costly and time-consuming process. The organisation of conflicts into categories allows to define the appropriate strategy to solve them. These strategies validate progressively PLPs of merging agents as soon as the category of conflicts that they provoke are resolved.

Utility of plans

The negotiation, for resolving conflicts, uses plan's utility measure to solve these conflicts. When the utility for a plan is greater than the utility of another one, the resolution of the conflict prefers the first plan over the second. PLPs are assigned with the utility which takes two attributes into account: the *reward R* and *penalty P* values.

Definition 2 *Each PLP has a reward function $R(P^i_\alpha,\ r_j,\ duration(I_\alpha) = end(I_\alpha) - begin(I_\alpha))$ representing the value rewarded by the agent when the PLP has been executed during an amount of time by using a resource. This value can represent an amount of money to pay for delivered information.*

Definition 3 *The penalty function P evaluates the cost paid to use the resource during an amount of time*

$$P(r_j, I_\alpha) = Cost_{r_j}(duration(I_\alpha)) \qquad (4)$$

Definition 4 *A* **time-dependent utility** *function, $U^{r,I}_{P^i_\alpha}$, measures the utility of executing the PLP P^i_α if it uses the resource r during the interval of time I.*

$$U^{r,I}_{P^i_\alpha} = R(P^i_\alpha, r, duration(I)) - P(r, I) \qquad (5)$$

Global strategy of resolution

The set $S_{P^i_\alpha}$ contains the PLPs in conflicts with the merging PLP P^i_α. The strategy of resolution consists of two functions: first, *Avoiding conflict* which consists in delaying, when it is possible, merging PLP of the agent α to avoid conflict and second, *non-avoided conflicts resolution* which consists in using utility of PLPs to decide whether the merging PLP should be maintained or not. We describe in the following both functions.

Avoiding conflicts This function delays the merging PLPs in conflicts rather than delaying the PLPs of the other agents that could put back in cause solved conflicts between the other agents and then other negotiation cycles should be perfomed. This solution can cause a combinatorial complexity of the strategy. The conflict avoidance function consists in making the interval I_α after the intervals I_β with which it overlaps. This means that the interval I_α should be delayed with the amount of time $end(I_\beta) - begin(I_\alpha)$. When delaying the merging PLP, the agent α verifies whether the deadline is reached or not. In order to coordinate the merging PLP P^i_α with the PLPs of P^j_β the set $S_{P^i_\alpha}$, the agent α sorts this set according to

$end(I_\beta)$. Afterwards, it assesses the avoidance of conflicts between the merging PLP and the PLPs of these agents β one by one. To this end, the agent α delays progressively its merging PLP by the amount of time $end(I_\beta) - begin(I_\alpha)$ until this amount violates its deadline. By delaying the merging PLP, some conflicts in $S_{P_\alpha^i}$ disappear, but others could appear. The merging agent α sends, then, its modified PLP to the other agents that are not in negotiation to make them informed on the modification of the PLP and then to assess new conflicts. The new detected conflicts are added in the set $S_{P_\alpha^i}$. The agent α don't try to avoid them because they are later than the non-avoided ones in the set $S_{P_\alpha^i}$ and then it is not possible to avoid them. Afterwards, the set $S_{P_\alpha^i}$ contains only the conflicts that cannot be avoided and then the *conflict resolution* function is called.

Resolving non-avoided conflicts The agent α resolves conflicts according to their type. When the merging PLP is rejected, the negotiation cycle fails and the agent α stops its strategy of resolution. When a PLP P_β^j is rejected by a merging agent α, the agent β must discard all its PLPs P_β^k $(k \geq j)$ due to the characteristics of the progressive planning. That's why, the resolving conflict function perfers a merging PLP P_α^i over another PLP P_β^j when its utility is higher than the cumulated utility of all P_β^k $(k \geq j)$. In the following, we describe the resolving conflicts rules.

• For a hard critical set: When ${U_{P_\alpha^1}}^{r,I}$ is greater than $\sum_{P_\beta^j \in S_{P_\alpha^1}} \sum_{k \geq j} U_{P_\beta^k}^{r',J}$ the agent α maintains its mandatory PLP and sends back all PLPs P_β^j of $S_{P_\alpha^1}$ to their respective agents β.

When $\sum_{P_\beta^j \in S_{P_\alpha^1}} \sum_{k \geq j} U_{P_\beta^k}^{r',J}$ is greater than $U_{P_\alpha^1}^{r,I}$, the agent α accepts all the PLPs of $S_{P_\alpha^1}$ and it sends a message to the corresponding agents β. The negotiation cycle of the merging agent fails and its local plan is rejected. Consequently, the corresponding goal can not be achieved. This situation means in our application that the provider cannot deliver information, even the mandatory information.

• For a hard non-critical set: The agent α maintains its mandatory PLP P_α^1 and rejects all optional PLPs in the set $S_{P_\alpha^1}$. In this resolution, we prefer maintaining a mandatory PLP rather than several optional PLPs.

• For a soft critical set: The agent α discards its optional PLP. The negotiation cycle fails and then the incremental construction of the merging plan stops. The merging agent returns the PLPs coordinated up to this step.

• For a soft non-critical set: When the utility $U_{P_\alpha^i}^{r,I}$ is greater than $\sum_{P_\beta^j \in S_{P_\alpha^i}} \sum_{k \geq j} U_{P_\beta^k}^{r',J}$, the agent α rejects all optional PLPs of the set $S_{P_\alpha^i}$ and maintains its merging optional partial local plan. Otherwise, the agent α discards its optional PLP and negotiation cycle fails. The agent α stops its processing and returns the PLPs coordinated up to this step.

Society of new merging agents

Several new agents can proceed to a progressive plan merging opertaion. In this approach we described how a new agent progressively constructs and coordinates its plan. The same processing is repeated for each new agent one by one according to their deadline. Indeed, agents in society use two message queues where the first queue is used to store messages concerning the declaration of merging and the second queue used for messages exchanged during the coordination with the current new merging agent. As soon as the current new merging agent is coordinated, agents in society, including the new coordinated, proceed to the coordination of the next new merging agent (with the most urgent deadline). A new coalition is then created.

Analysis

The termination of the iterative processing of this approach is guaranteed because the number of rounds is bounded by the maximum number k_{max} of levels per agent. While the termination of each round is guaranteed by the number of agents in the society n. The complexity of the approach is then $O(k_{max} * n)$. In large societies (n is very large) the complexity is $O(n)$ while in small societies ($n \approx k_{max}$, $k_{max} = 5$), the complexity is equivalent to $O(n^2)$. Consequently, this approach is more suitable for the large societies. We also analyze in critical situations (where only some PLPs are accepted) and in non-critical situations (where the complete merging local plan is accepted) the cumulated cost of planning C_p and coordination C_c of our approach and a *basic approach* similar to PGP approach.

• **Our approach:** The cost of *one* round of the anytime coordination is the cost of partial plan construction C_p cycle and a coordination cycle C_c. Thus the cost of *one* round is as follows:

$$Cost_{a\ round} = C_p(P_\alpha^i) + \max_{\beta \in A}(\sum_j C_c(P_\beta^j, P_\alpha^i)) \qquad (6)$$

In the most critical situation the cost of the anytime coordination is then the cost of the first round while in the other situations is the cumulated cost of *one* round: $\sum_i Cost_i$, where i is the round i of the iterative processing of the anytime coordination.

• **Basic approach:** The cost of this approach is the cost of the complete plan construction and its coordination. Thus the cost in critical and non-critical situations is the same.

$$Cost = \sum_i C_p(P_\alpha^i) + \max_{\beta \in A}(\sum_j \sum_i C_c(P_\beta^j, P_\alpha^i)) \qquad (7)$$

The time consumed by our approach in non-critical situations is greater than the time consumed by the basic approach because it is trivial to prove the following:

$$\sum \max(\sum cost) > \max \sum \sum cost \qquad (8)$$

In critical situations our approach is more suitable because the cost of the basic approach is the same than in non-critical situations while our approach is only the cumulated costs of performed rounds that can be in critical situations no more than $k/2$ where k is the number of partial plans of the merging agent. Finally, the approach is a greedy technique based on a local optimisation approach to a distributed allocation resource problem that is hard to optimize globally.

Experimental evaluation: First Results

This section illustrates the operation of the resulting approach and examines two fundamental questions. The first goal is to compare the performance of our approach to a basic approach similar to PGP approach (Durfee & Lesser 1987). This basic approach is based on a planning cycle to construct a complete plan followed by a negotiation cycle to coordinate the complete plan. It ignores time constraints and the overhead of construction and coordination. The second goal of the experimental evaluation is to assess the benefit of our approach in time-constrained situations.

Experiment design

A progressive planning agents language is specified allowing the system to simulate the progressive plans for the application. We have collected experimental data on the performance of our approach and the basic approach. The quality of the results for each problem instance is assessed according to two parameters: the *time consumed* in critical and non-critical situations by each approach to construct and to coordinate the plan according to the number of agents in the society, and the *cumulated utility* of coordinated partial plans according to the deadline before which the merging agent must construct and coordinate its plan. Problem instances are developed with 10 agents. This second experimental evaluation will show the benefit of our approach in time-constrained situations.

Empirical results on Consumed time

Figures 1 and 2 show that our approach is more suitable than the basic approach for critical situations while in non-critical situations it is the contrast. The theoretical explanation to that is given in equations 6, 7 and 8.

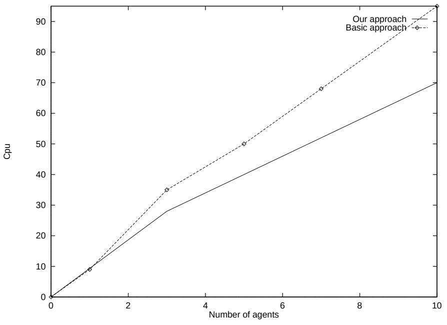

Figure 1: Time consumption in critical situations

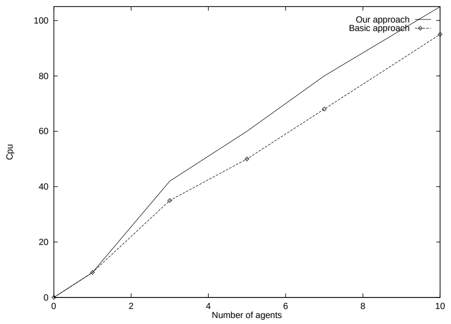

Figure 2: Time consumption in non-critical situations

Empirical results on utility

In this experiments, we measure tha value as a function of available time (deadline). This value consists of the cumulated utilities of coordinated PLPs: $\sum_i(U_i)$. Figure 3 shows the difference between the values of our approach and the basic approach over the deadlines. The figure confirms the fact that our approach leads to a substantial quality gain over the basic approach. The main reason to that is the basic approach can consume time for constructing partial plans that it cannot be merged in a sosciety while our approach as soon as a partial plan is constructed its coordination is immediately processed. The basic approach cannot trade-off between planning time and coordination time as it is done in our approach.

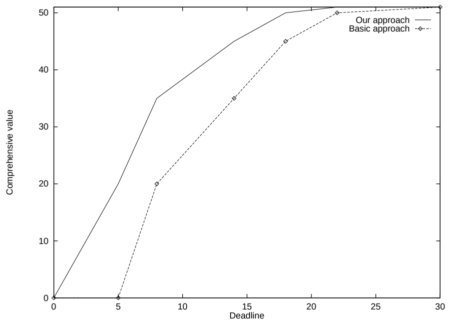

Figure 3: Comprehensive value for different deadlines

Discussion

The approach, we present, addresses a problem much more complicated than these addressed in previous works (Botti, Crespo, & Ripoll 1994; Schwartz & Kraus 1997). Our approach addresses a problem of sharing several resources while in (Botti, Crespo, & Ripoll 1994), it is proposed a mechanism of one resource (the blackboard). Unlike the work presented in (Schwartz & Kraus 1997) consisting of distributed servers of information negotiating to allocate each resource (dataset) to one server, our agents negotiate to allocate several resources. Furthermore, our approach is more suitable to dynamic and time-constrained situations through its anytime coordination. Unlike approaches described in (Sandholm 1993; Sen & Durfee 1996) which are task-based negotiation approaches, our approach can be seen as a resource-based negotiation.

PGPP approach (Mouaddib 1997) is an extension of PGP to constrained time environments. This approach progressively constructs a complete local plan and it coordinates it incrementally PLP by PLP while our new approach it constructs a PLP and it immediately coordinates it. Our new approach by stopping the incremental coordination of PLP when the merging PLP operation fails, avoids to plan at unnecessarily level of details and to predict judiciously whether more actions maximize the expected gain or not (Vidal & Durfee 1995). Furthermore, with this processing this model can take the overhead of the negotiation into consideration.

Unlike other investigators in resources allocation domain that have proposed distributed techniques (Neiman, Hildum, & Lesser 1994; Schwartz & Kraus 1997) which can perform backtracking to coordinate activites, our approach minimize backtracking and then reduces the complexity of the algorithm. Further work will concern the application of this approach in auctions and multi-robots domains.

Acknowledgments

Support for this work was provided by the GanymedeII Project of Plan Etat/Nord-Pas-De-Calais

References

Botti, V.; Crespo, A.; and Ripoll, I. 1994. Multiple access and coherence management in a real-time temporal blackboard. In *ECAI*, 312–316.

Decker, K., and Lesser, V. 1992. Generalizing the partial global planning algorithm. *Journal on Intelligent Cooperative Information Systems* 1(2):319–346.

Durfee, E., and Lesser, V. 1987. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on systems, Man, and Cybernetics* 21(5).

Kraus, S.; Wilkenfeld, J.; and Zlotkin, G. 1995. Multiagent negotiation under time constraints. *Artificial Intelligence* 75:297–345.

Mouaddib, A.-I. 1997. Progressive negotiation for time-constrained autonomous agents. In *First International Conference on Autonomous Agents*, 8–16.

Mouaddib, A.-I. 1998. Multistage negotiation for distributed scheduling of resource-bounded agents. In *AAAI Spring Symposium on Satisficing Models*, 54–59.

Neiman, D.; Hildum, D.; and Lesser, V. R. 1994. Exploiting meta-level information in a distributed scheduling system. In *AAAI-94*.

Sandholm, T., and Lesser, V. 1995. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *First ICMAS*, 328–335.

Sandholm, T. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *AAAI*, 256–262.

Schwartz, R., and Kraus, S. 1997. Negotiation on data allocation in multi-agents environments. In *AAAI*, 29–35.

Sen, S., and Durfee, E. 1996. A contracting model for flexible distributed scheduling. *Annals of Operating Research* 65:195–222.

Vidal, J., and Durfee, E. 1995. Recursive agent modeling using limited rationality. In *ICMAS*, 126–133.