Utility-Based Multiagent Coalition Formation with Incomplete Information and Time Constraints*

Leen-Kiat Soh
Computer Science and Engineering Department
University of Nebraska
Lincoln, Nebraska
lksoh@cse.unl.edu

Costas Tsatsoulis
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS
tsatsoul@ittc.ku.edu

Abstract - *In this paper we propose a coalition formation model for a cooperative multiagent system in which an agent forms sub-optimal coalitions in view of incomplete information about its noisy, dynamic, and uncertain world, and its need to respond to events within time constraints. Our model has two stages: (1) when an agent detects an event in the world, it first compiles a list of coalition candidates that it thinks would be useful (coalition initialization), and (2) then negotiates with the candidates (coalition finalization). A negotiation is an exchange of information and knowledge for constraint satisfaction until both parties agree on a deal or one opts out. Each successful negotiation adds a new member to the agent's final coalition. This paper talks about the steps we have designed to enhance the finalization stage.*

Keywords: Multagent systems, negotiation, coalition formation.

1 Introduction

In this paper we describe a coalition formation model for agents with incomplete information and time constraints within a dynamic and uncertain world. A coalition is a group of agents that collaborate to perform a coordinated set of tasks that may be a response to an event that has occurred in the environment. A dynamic coalition is one that is formed as a response to an event and dissolved when the event no longer exists or when the response is completed. A coalition is necessary when an agent cannot respond to an event all by itself due to lack of information, knowledge, or functional capabilities. Ideally, the agent would prefer to form an optimal coalition to maximize the yield of the system as a whole. However, such optimal rationalization requires the agent to have complete information about its world and its neighboring agents, and also about the uncertainty associated with all factors related to the multiagent infrastructure. When that information is not readily available and the collection of that information is too costly, an agent cannot afford such optimality. In the following, we elaborate on some of the problem characteristics.

(1) Our model applies to an environment where each agent has incomplete information about its world. Incomplete information may be due to polling and updating costs, constrained resources, and decentralized information base.

(2) An optimal rationalization for coalition formation may not be possible due to (a) noise and uncertainty in the environment, and (b) time constraints. For example, the communication channels among the agents may be congested or faulty, messages may be noisy or lost, perceived events may be qualified inaccurately, and so on.

(3) We assume all agents are peers—there is no hierarchy among the agents. Each agent is able to sense its environment, revise its own perceptions, and form its own coalitions. This allows the agents to be reactive to environmental changes, without having the directives passed from a higher-up agent while encouraging diversity in information stored at each agent.

(4) We propose using negotiations to refine a coalition. We see negotiation as an exchange of necessary information pertinent to individual constraints, perceptions, and commitments. This exchange of information is performed only when the coalition-initiating agent approaches potential coalition partners to request for help. Thus, the initial coalition can be less than optimal and be computed hastily as the negotiation will refine or finalize the selection.

(5) Our model expects coalition members to refuse to join in a coalition, especially in a resource-constrained environment and also plans for failed communication due to congestion, noise, or message loss. Thus, the initial coalition may not survive after negotiations as the working coalition is finalized.

Briefly, our proposed model works as follows. When an event is detected in a multiagent system, one of the agents initiates the coalition formation process in hope of organizing a group of cooperative agents to perform tasks

* **0-7803-7952-7/03/$17.00 © 2003 IEEE.**

in response to the event. This initiating agent (also known as the "computing agent" [3])) shoulders the responsibility of designing the best coalition given the situated information to increase the chance of forming a working and useful coalition at the end of the process. The model consists of two stages. First, during the *coalition initialization*, the initiating agent extracts a ranked list of useful agents. Then, the initiating agent approaches the potential coalition partners and requests for negotiations during a *coalition finalization* step. Our negotiation is based on a case-based reflective argumentative model [7]. This paper focuses on the different strategies during the initialization stage and the steps we have designed to enhance the finalization stage.

Before further discussions, here we outline some assumptions about our agents. In general, our model assumes that agents have the following characteristics: autonomous, rational, communicative, honest, and cooperative. There are in general three reasons why agents cooperate [6]. First, an agent cannot perform a specific task by itself. Second, an agent can perform a specific task, but other agents are more efficient in performing the task. Third, an agent can perform a specific task, but working on it collaboratively will increase the benefits from the task (or reduce the costs). We also assume that each agent exists in a neighborhood where it knows some basic properties of its neighbors (such as functional capabilities) and can communicate to them directly. Each agent has a neighborhood and can communicate directly with all its neighbors, and each neighbor can communicate with the agent directly as well. It is from this neighborhood that an agent forms a coalition. Readers are referred to [10] for a detailed discussion on various agent characteristics.

2 Related Work

A definition from the rational coalition theory outlined in [1] states that a coalition game with transferable utility in normal characteristic form. The value of a coalition is the total utility that the members of the coalition can achieve by coordinating and acting together. However, in our problem domain, the agents do not have the information they need to compute accurately the value of a coalition but we have proposed general heuristics for estimating such values. In our model, the value of a coalition is not independent of nonmembers' actions as in some studies in characterstic function games [9].

Sandholm and Lesser [3] introduce a bounded rationality in which agents are guided by performance profiles and computation costs in their coalition formation process. The authors' bounded rationality model requires each agent to pay for computational resources that it uses for deliberation. Our model is also affected by the current status of an agent, such as the availability of negotiation threads, and the environment, such as the availability of the communication channel.

Zlotkin and Rosenschein [11] describe a coalition driven by task-oriented utilities. In a task-oriented domain (TOD), a coalition can coordinate by redistributing their tasks among themselves. In our model, we address the problem from the viewpoint of an agent. That is, an agent can be of two or more coalitions simultaneously as each agent is autonomous and capable of reacting to separate events.

Shehory *et al.* [6] relax some of the restrictive assumptions of theoretical coalition formation algorithms for a real-world system. Their model assumes that the agents are group-rational and the agent population does not change during the coalition formation. There are several differences between this model and ours. First, the authors' model assumes that all agents know about all of the tasks and the other agents. In our model, an initiating agent knows only the agents in its neighborhood and knows about partially the updated status of a selective subset of the neighbors after negotiation. Second, the details of intra-coalitional activity are not necessary for agents outside of the coalition in the authors' model. On the contrary, in our model, an agent can and does belong to multiple coalitions concurrently.

Tohmé and Sandholm [9] studies coalition formation among self-interested agents that cannot make sidepayments—reward each other with payments for agreement to join some coalition, making the evaluation of a coalition solely on its utility, and proposes a model that guarantees convergence or stability in its coalition solution. However, in our model, there are external factors such as the dynamic events and noisy communication channels that may thwart the successful completion of a negotiation, rendering a negotiation outcome unpredictable.

Sen and Dutta [4] propose an order-based genetic algorithm (OBGA) as a stochastic search process to identify the optimal coalition structure. A significant difference between the authors' work and our model is the scope of coalition formation. The authors' algorithm is for searching for an optimal coalition structure, which consists of all the agents in the environment grouped into one or more coalitions.

Finally, note that our coalition formation activities differ from that presented in [2] which defines coalition formation as three interacting activities:

(1) Coalition structure generation where agents within each coalition coordinate their activities but do not coordinate between coalitions. This means partitioning the set of agents into exhaustive and *disjoint* coalitions

and the partition is called a coalition structure. In our model, an agent forms a coalition from its neighborhood. Some neighbors may be part of the coalition, may be part of other coalitions, or may be simply idle. Coalitions in our model may also overlap.

(2) Solving the combinatorial optimization problem of each coalition whose objective is to maximize the utility of the coalition. This means pooling the tasks and resources of the agents in the coalition, and solving their joint problem. In our model, the initiating agent shoulders this approximation using imperfect information to increase the chance for a successful coalition.

(3) Dividing the value of the generated solution among agents. There is no such explicit value distribution in our model. Our agents are altruistic and directed to help if possible to achieve global goals, and thus do not require additional motivation such as rewards or values. However, our agents do want to manage their own resources efficiently and this motivates negotiations and the task allocation among agents.

3 Methodology

Our methodology has two stages: (1) when an agent detects an event in the world, it first compiles a list of coalition candidates that it thinks would be useful (coalition initialization), and (2) then negotiates with the candidates (coalition finalization). A negotiation is an exchange of information and knowledge for constraint satisfaction until both parties agree on a deal or one opts out. Each successful negotiation adds a new member to the agent's final coalition.

3.1 Coalition Initialization

We will briefly discuss this initialization stage here. Readers are referred to [8] for details. The first stage of the utility-based multiagent coalition formation model is the determination of the set of the initial coalition candidates, denoted as $\Lambda_{ini}(a_i, e_j)$ for agent a_i and event e_j. We denote a candidate as α_k. In our model, the initiating agent a_i first generates the initial coalition candidates, $\Lambda_{ini}(a_i, e_j)$, to deal with an event e_j. $\Lambda_{ini}(a_i, e_j)$ represents the neighbors that *the agent thinks can be of help* to respond to e_j. To find out whether these candidates are *willing* to help, the initiating agent needs to negotiate. Due to resource constraints, our design first ranks the candidates on their potential utility values to the coalition so that the initiating agent can negotiate with the agents with the highest utility values first. For a candidate $\alpha_k \in \Lambda_{ini}(a_i, e_j)$, we base its potential utility, PU_{α_k, a_i}, on three sets of attributes: (1) the past relationship between the initiating agent and the candidate, $rel_{past,a_i}(\alpha_k, t)$, where t is the point in time when the set of attribute-value pairs in the relationship is collected, (2) the current relationship between the initiating agent and the candidate, $rel_{now,a_i}(\alpha_k, t)$, and (3) the ability of the candidate in handling the event, $ability_{a_i}(\alpha_k, e_j, t)$. All these sub-utility measures map into $\Re : 0 \ldots 1$ and each is asymmetric such that $rel_{past,a_i}(\alpha_k, t) \neq rel_{past,\alpha_k}(a_i, t)$, where t denotes time.

Finally, the potential utility, PU_{α_k, a_i}, of a candidate α_k is a weighted sum of $rel_{past,a_i}(\alpha_k, t)$, $rel_{now,a_i}(\alpha_k, t)$, and $ability_{a_i}(\alpha_k, e_j, t)$:

$$PU_{\alpha_k, a_i} = W_{\Lambda_{ini}(a_i, e_j)} \bullet \left[rel_{past,a_i}(\alpha_k, t) \quad rel_{now,a_i}(\alpha_k, t) \quad ability_{a_i}(\alpha_k, e_j, t) \right] \tag{1}$$

where $W_{\Lambda_{ini}(a_i, e_j)} = \left[w_{past,a_i,e_j} \quad w_{now,a_i,e_j} \quad w_{ability,a_i,e_j} \right]^T$ and $w_{past,a_i,e_j} + w_{now,a_i,e_j} + w_{ability,a_i,e_j} = 1$.

3.2 Coalition Finalization

In our model, we use a real-time case-based logical negotiation protocol to dictate the rules of encounter or the negotiation strategies between two agents. Interested readers are referred to [7] for a detailed presentation of the logical protocol. In this paper, we focus on the steps that we have taken to enhance the finalization process : (1) awareness, and (2) relaxation and termination.

Since our negotiation protocol is multi-step, it facilitates interactions between negotiation threads. An initiating agent can invoke a host of concurrent negotiations, bounded by $\left|\Lambda_{can\_approach}(a_i, e_j)\right|$. While the negotiation threads are actively engaged in their respective negotiations, the initiating agent continues to monitor its world, examine its tasks, communicate with other agents, and watch the status of its negotiation threads. Since each of these threads knows how to negotiate on its own, all it needs from time to time is for the parent agent to update the agent's current beliefs and intentions that might interrupt the negotiation or change the negotiation issues. The parent agent thus is able to infuse a high-level of awareness in the negotiation threads, relax the negotiation issues (less demanding or more conceding, for example), and abort negotiations with diminishing returns. In the following subsections, our discussions focus on the indirect inter-thread activities that the agent provides for its negotiation threads.

Awareness

Since the environment is dynamic, an ongoing negotiation may become useless. For example, if the negotiation is part of a response to an event e_j and e_j becomes false, then the negotiation has to be terminated. Since a negotiation thread handles the negotiation semi-autonomously, it must be aware of such a situation, and the parent agent has to provide such awareness. This coalition awareness has several benefits. First, it allows an agent to free up its negotiation threads, communication channels, and communication bandwidth for other negotiation tasks. Second, it allows an agent to immediately abandon failing coalition, re-assess its environments, and start another coalition formation. Third, by terminating useless negotiations, an agent is able to base its reasoning on updated, more correct status profile.

A negotiation thread conducts its negotiation following a logical real-time protocol that spells out what it should do in each negotiation step, and a negotiation strategy that dictates how the thread should negotiate—how much time it has, how conceding it should be, what kind of arguments it has, which arguments it should send first, and so on [7]. It also needs to know the context of the negotiation—the request, the amount of resource to give up, the counterpart agent, and so on. When activated, a negotiation thread downloads the negotiation context and strategy from the parent agent. Then, if the thread does not hear from the parent agent, it knows how to negotiate on its own and report back to the parent agent only when the negotiation is completed.

The parent agent, on the other hand, carries out its normal tasks such as monitoring the world, actuating its sensors, and so on. It holds a shared data object with each negotiation thread. When the event changes or the current status of the coalition changes, the agent evaluates the current status of each negotiation thread and makes a decision as to whether to relax, to terminate, or refine the negotiation. This decision together with its pertinent information is stored at that data object. We call this shared data object the *awareness link*, or AL_{a_i,β_z} for the β_z negotiation thread of agent a_i, where AL_{a_i,β_z} has X negotiation threads, $\{\beta_1, \beta_2, \cdots, \beta_X\}$. Both the agent and the negotiation thread can store and access data on this awareness link. With this design, the parent agent shoulders the task of feeding its negotiation threads additional instructions. There are several reasons why we adopt this awareness link design in our model. First, the environment is dynamic and real-time critical. It does not make sense for each negotiation thread to setup its own sensors, monitors, and even decision makers to determine the current status of the agent and changes its negotiation behavior accordingly, since it would have to sieve through unrelated information and data and that would be time consuming. Second, it is natural for the agent to disperse the information to all its negotiation threads. A negotiation thread does not know the status of a coalition (e.g., whether the coalition is failing or succeeding); only the agent knows that. With that knowledge, an agent can decide whether to scale back on some of its negotiations, or make other modifications. This way, the chain of command is direct and less confusing, and certainly less computationally intensive. Third, with each negotiation thread having its own dedicated awareness link, the information or data passed through the parent agent and that particular thread does not interfere with the other threads. This way, each negotiation thread can concentrate on the instructions specifically directed to it from the parent agent.

In our model, the parent agent checks the negotiation status of its negotiation threads within a framework of tasks. It checks its messages, its sensors for events, tasks, and the negotiations, and then repeats. This lifecycle varies in its duration, as the environment is dynamic and uncertain. For the negotiation thread, we propose a gradated scheme based on the percentage of time elapsed. For example, a thread checks and updates its status (1) less frequently in the beginning, (2) more frequently towards the end, (3) less frequently when it is progressing according to plan, and (4) more frequently when it is failing. This is because we assume that the event status is still relatively constant in the beginning of the negotiation and only changes after a certain time period has passed. We also assume that when a negotiation is progressing well and succeeding, that negotiation thread should carry on and complete the negotiation unless some significant event occurs and calls it off. In this manner, the agent does not lose the utility of such a negotiation and learns to be efficient. We also assume that when a negotiation is not doing well, after reporting it to the parent agent, the negotiation thread can expect further instructions from the parent agent, hence the increase in its access of the awareness link.

Relaxation and Termination

Each agent is responsible for the coordination among its negotiation threads as the negotiation threads do not talk to each other directly. The agent monitors the status of the negotiations and makes decisions. Two of the decisions it can make are relaxation and termination. From the initiating agent standpoint, this relaxation results in a smaller demand; from the responding agent standpoint, this relaxation results in a more yielding stance. Since this paper's focus is in coalition formation, we will discuss relaxation and termination from the viewpoint of an initiating agent.

Suppose in a 1-to-1 task allocation problem, we have $\left|\Lambda_{approached}\left(a_i, e_j\right)\right| > \left|F_\tau\right|$. That is, the number of candidates that the agent a_i approaches is greater than the number of tasks required to respond to the event e_j. At time *t*, a_i activates all its negotiation threads, each with a partial-assignment $\rho = \left\langle \alpha_\rho, f_\rho \right\rangle$. At time $t + \Delta$, some changes have been detected such that F_τ is now F'_τ. If $F'_\tau \not\subset F_\tau$, then the agent needs to (1) immediately terminate all negotiation threads with $\rho = \left\langle \alpha_\rho, f_\rho \right\rangle$ where $f_\rho \notin F'_\tau$, and (2) label this change as a new event and proceed from there accordingly. If $F'_\tau \subset F_\tau$, then the agent needs to relax the negotiations. In a many-to-1 task allocation problem, suppose that the negotiation thread in question has $\rho = \left\langle \alpha_\rho, \left\{ f_{\rho_1}, f_{\rho_2} \right\} \right\rangle$. Then, the agent can compute $benefit_{a_i}\left(\alpha_\rho, f_{\rho_1}, t\right)$ and $benefit_{a_i}\left(\alpha_\rho, f_{\rho_2}, t\right)$ (assuming disjoint tasks). It then can decide to drop f_{ρ_1} or f_{ρ_2} from its original demand or keep both. The algorithm becomes:

Algorithm Many-to-1 Relaxation and Termination: (1) for all ongoing negotiations with $\rho = \left\langle \alpha_\rho, f_\rho \right\rangle$ where $f_{\rho_s} \in f_\rho$ and $f_{\rho_s} \notin F'_\tau$ do: (1.1) compute $benefit_{a_i}\left(\alpha_\rho, f_{\rho_s}, t\right)$ for all $f_{\rho_s} \in f_\rho$ and $f_{\rho_s} \notin F'_\tau$, (1.2) check the current status of the negotiation and denote this as $progress_{\beta_x}\left(\alpha_\rho, f_\rho, t\right)$ for negotiation thread β_x, (1.3) compute the expected utility of continuing with this negotiation for all $f_{\rho_s} \in f_\rho$ and $f_{\rho_s} \notin F'_\tau$:

$$EU_{\beta_x}\left(\alpha_\rho, f_{\rho_s}, t\right) = \frac{PU_{\alpha_\rho, f_{\rho_s}, a_i} + progress_{\beta_x}\left(\alpha_\rho, f_\rho, t\right) \cdot benefit_{a_i}\left(\alpha_\rho, f_{\rho_s}, t\right)}{2} \tag{2}$$

and (1.4) if $EU_{\beta_x}\left(\alpha_\rho, f_{\rho_s}, t\right)$ is greater than what the agent can afford to spend in resources, then the agent retains f_{ρ_s} in the negotiation; otherwise, it drops f_{ρ_s} from the negotiation. The expected utility is basically the potential utility of the coalition member to the original event response plus the utility of continuing with the negotiation. The latter utility says that if the negotiation is progressing well and the eventual outcome will benefit the environment, then the agent should not discard the ongoing effort.

The relaxation and termination behavior is both rational and altruistic. An agent should not be conducting negotiations that use its resources when those negotiations have become useless. Neither should an agent *impose* or *transfer* that cost to its responding agent by insisting on useless negotiations. Without this relaxation and termination capability, the initiating agent would have to be more careful in its initialization since it has less room for errors. That would mean for the initiating agent to collect more information in its rationalization which would in turn decrease the autonomy and robustness of the multiagent system. So, the coupling of the initialization and relaxation/termination is very important in our utility-based, dynamic coalition formation model.

Finally, at the end of all negotiations, we have the final coalition $\Lambda_{final}\left(a_i, e_j\right)$ and $\Lambda_{final}\left(a_i, e_j\right) \subseteq \Lambda_{approached}\left(a_i, e_j\right) \subseteq \Lambda_{ini}\left(a_i, e_j\right)$.

4 Implementation and Results

The driving application for our system is multisensor target tracking, a distributed resource allocation and a constraint satisfaction problem [7]. The objective is to track as many targets as possible and as accurately as possible using a network of sensors. Each sensor has a set of consumable resources, such as beam-seconds (the amount of time a sensor is active), battery power, and communication channels, which each sensor desires to utilize efficiently. Each sensor is at a fixed physical location and, as a target passes through its coverage area, it has to collaborate with neighboring sensors to triangulate their measurements to obtain an accurate estimate of the position and velocity of the target.

Here we report on some preliminary experiment results for the behavior analysis of our multiagent system, specifically for multisensor target tracking. We consider here an exemplary run that we used to adjust our system parameters. In this run, the total number of attempts to form a coalition was 150. The total number of coalitions successfully formed (after coalition finalization) was 30, or 20%. The total number of coalitions confirmed by all coalition members was 26, or 86.7% of all successfully formed coalitions. Finally, the total number of coalitions executed on time was 18, or 61.5% out of all successfully confirmed coalitions.

First, the percentage of successfully formed coalitions was only 20.0%. Out of the 120 failed attempts, 86 (71.7%) of them were caused by one of the coalition members outright refusing to negotiate, 17 (14.2%) were caused by the communication channels being jammed, and 17 (14.2%) were caused by busy negotiation threads. When an initiating agent initiates a negotiation request to a candidate and that candidate immediately refuses to entertain the negotiation, it can be due to (1) the responding agent does not have idle negotiation threads, or (2) the responding agent cannot project the requested task into its job queue. Thus, we expect this failure rate to decrease once we increase the number of negotiation threads allocated per agent. When an agent fails to send a message to another agent, or fails to receive an expected message, we label this as a communication "channel-jammed" problem. When an initiating agent fails to approach at least two candidates, it immediately aborts the other negotiation process that it has invoked for the same coalition. This causes the coalition to fail.

Second, the probability of a successfully formed coalition getting confirmed completely was 86.7%. For each coalition successfully formed, three confirmations were required. Out of 30 coalitions, 4 coalitions were confirmed only by two of the members. The causes were (1) the acknowledgment message sent out by the initiating agent was never received by the responding agent expecting a confirmation, and (2) the agreed task had been removed from the job queue before the confirmation arrived. The first cause happened since communication channels could be jammed. The second cause happened because of a contention for a slot in the job queue by two tasks. For example, suppose agent A receives a request from agent B to track a target starting at 8:00 a.m. Agent A responds to the request and starts a negotiation. Then later on, agent A receives a request from agent C to track a target starting also at 8:00 a.m., but using a different sensing sector (each sensor has three). Agent A checks its job queue and sees that it is free at that time and thus agrees to negotiate. Note that a task is inserted into the job queue only after the agent agrees to perform it. Now, suppose that both negotiations are successful. The negotiation between A and B ends first and then that between A and C. When the first negotiation ends, agent A adds the task requested by B to the job queue. Immediately after, when the second negotiation also ends successfully, agent A adds the second task, requested by C to the job queue, and this causes the second task to replace the first task. This is a problem with over-commitment.

Our preliminary results were promising as the agents are able to form coalitions using the outlined model and methodology. However, the results showed that there are timing and task scheduling issues that are currently being addressed.

5 Conclusions

In this paper, we have introduced several steps that we have designed to enhance a utility-based multiagent coalition formation model. The model uses awareness links and flexible relaxation and termination schemes to deal with the dynamism in a coalition formation process due to incomplete information and time constraints. The preliminary results were insightful and promising.

Acknowledgement

The authors would like to thank Xin Li for her programming and running the experiments. The work described in this paper was partially sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL).

References

[1] J. P. Kahan and A Rapoport, *Theories of Coalition Formation*, Lawrence Erlbaum, 1984.

[2] T. W. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *AI*, vol. 111, no. 1-2, pp. 209-238, 1999.

[3] T. W. Sandholm and V. R. Lesser, "Coalition formation amongst bounded rational agents," in *Proc. IJCAI 1995*, Montreal, Canada, pp. 662-669, 1995.

[4] S. Sen and P. S. Dutta, "Searching for optimal coalition structures," in *Proc. ICMAS'2000*, July 7-12, Boston, MA, pp. 286-292, 2000.

[5] O. Shehory and S. Kraus, "Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents," in Proc. ICMAS'96, Kyoto, Japan, pp. 330-337, 1996.

[6] O. M. Shehory, K. Sycara, and S. Jha, "Multi-agent coordination through coalition formation," in *Proc. ATAL'97*, Providence, RI, 1997.

[7] L.-K. Soh, and C. Tsatsoulis, "Reflective negotiating agents for real-time multisensor target tracking," in *Proc. IJCAI'01*, August 4-10, Seattle, WA 2001.

[8] L.-K. Soh, and C. Tsatsoulis, "Allocation Algorithms in Dynamic Negotiation-Based Coalition Formation," in *Proc. IJCAAMAS Workshop on Teamwork and Coalition Formation'02*, July 15-16, Bologna, Italy 2002.

[9] F. Tohmé and T. Sandholm, "Coalition formation processes with belief revision among bounded-rational self-interested agents," *J. Logic & Computation*, vol. 9, no. 6, pp. 793-815, 1999.

[10] M. Wooldridge and N. Jennings, "Intelligent agents: theory and practice," *Knowledge Engr. Rev.*, vol 10, no. 2, pp. 114-152, 1995.

[11] G. Zlotkin and J. S. Rosenschein, "Coalition, cryptography and stability: mechanisms for coalition formation in task oriented domains," in *Proc. AAAI*, July, Seattle, WA, pp. 432-437, 1994.