A Kernel-Oriented Model for Coalition-Formation in General Environments: Implementation and Results*

Onn Shehory **Sarit Kraus**
Department of Mathematics and Computer Science
Bar Ilan University Ramat Gan, 52900 Israel
{shechory, sarit}@bimacs.cs.biu.ac.il
Tel: +972-3-5318863
Fax: +972-3-5353325

Abstract

In this paper we present a model for coalition formation and payoff distribution in general environments. We focus on a reduced complexity kernel-oriented coalition formation model, and provide a detailed algorithm for the activity of the single rational agent. The model is partitioned into a social level and a strategic level, to distinguish between regulations that must be agreed upon and are forced by agent-designers, and strategies by which each agent acts at will. In addition, we present an implementation of the model and simulation results. From these we conclude that implementing the model for coalition formation among agents increases the benefits of the agents with reasonable time consumption. It also shows that more coalition formations yield more benefits to the agents.

Introduction

An important method for cooperation in multi-agent environments is coalition formation. Membership in a coalition may increase the agent's ability to satisfy its goals and maximize its own personal payoff. Game theory literature such as (Rapoport 1970) describes which coalitions will form in N-person games under different settings and how the players will distribute the benefits of the cooperation among themselves. These results do not take into consideration the constraints of a multi-agent environment, such as communication costs and limited computation time, and do not present algorithms for coalition formation. Our research presents a multi-agent approach to the coalition formation problem, and provides a coalition-formation procedure. The paper deals with autonomous agents, each of which has tasks it must fulfill and access to resources that it can use to fulfill these tasks. Agents can satisfy goals by themselves, but may also join together to satisfy their goals. In such a case we say that the agents form a coalition.

In this paper we present a modification of the Kernel concept from game theory (Davis & Maschler 1965). The modified Kernel serves as a basis for a polynomial-complexity mechanism for coalition formation. The mechanism is partitioned into two levels - the social level and the strategic level. The coordination-regulation protocols constitute the social level[1]. Different designers of agents must agree upon the regulation protocols of the social level in advance. The strategic level consists of strategies for the individual agent to act in the environment for maximization of its own expected payoff, given the social level, and can be decided upon by individual agents during the coalition formation process.

Related work in DAI

Research in DAI is divided into two basic classes: *Distributed Problem Solving (DPS)* and *Multi-Agent Systems (MA)* (Bond & Gasser 1988; Durfee & Rosenschein 1994). Our research is closer to MA since it deals with interactions among *self-motivated*, *rational* and *autonomous* agents. However, any interaction among agents requires some regulations and structure. The minimal requirement for interactions in multi-agent systems is a common language or a common background (Gasser 1993). In coalition-formation, the need for regulations increases further(Shapley & Shubik 1973).

Shoham, Tennenholtz and Moses (Moses & Tennenholtz 1993; Shoham & Tennenholtz 1992) show that pre-compiled highly-structured "social laws" are able to coordinate agent activity. Agents are assumed to follow the social laws since they were designed to do so and not because they benefit individually from following these laws. In our research, we do not explicitly use social laws in order to coordinate agent activity. We provide the agents with a social level of cooperation. The designers of agents should agree in advance which regulations the agents in a given environment will use. These regulations are incorporated into all of the agents, but each agent chooses its strategy for the

*Kraus is also affiliated with the Institute for Advanced Computer Studies, University of Maryland. This material is based upon work supported in part by the NSF under grant No. IRI-9423967 and the Israeli Ministry of Science, grant No. 6288. We thank S. Aloni and M. Goren for their major contribution to the implementation of the model.

[1] We use the concepts of the social level and the notion of regulations interchangeably.

interaction individually and joins a coalition only if it increases its personal payoff.

Our mechanisms are different from these that are applied in DPS environments. However, as in the Contract Net framework (Davis & Smith 1983; Malone *et al.* 1988) or the Functionally Accurate, Cooperative, (FA/C) paradigm (Durfee 1988; Decker & Lesser 1993) that considers DPS environments, we also enables cooperation of groups of agents, though not necessarily of all of them.

Zlotkin and Rosenschein (Zlotkin & Rosenschein 1994) and Ketchpel (Ketchpel 1994) consider coalition formation in supper-additive environments. Our work discusses general multi-agent environments, and provides the designers of agents with explicit coalition-formation mechanisms for these environments.

Sandholm and Lesser (Sandholm & Lesser 1995) presented a coalition formation model for bounded-rational agents and present a general classification of coalition games. In their model, the value of a coalition depends on the computation time, however, all configurations and possible coalitions are considered when computing a stable solution. We consider problems where the computational time of coalition values is polynomial. Therefore, we concentrate on the polynomial coalition-configuration formation and utility distribution and provide polynomial concepts of stability.

Environment Description

We assume the following: (i) The designers of the agents agree upon the regulations in advance and incorporate them into their agents, and these regulations are enforceable by deviation revealability and penalties. (ii) Various communication methods exist, so that the agents can negotiate and make agreements (Werner 1988). However, communications require time and effort. (iii) Resources can be transferred among agents. (iv) There is a monetary system that can be used for side-payments. The agents use this monetary system in order to evaluate resources and productions that result from the use of the resources. The money is transferable among the agents and can be redistributed in a case of coalition formation. The monetary system is part of the regulation of the environment. We present it as a possible regulation since it increases the agents' benefits from cooperation, although agents may reach agreements and form coalitions even if this assumption is not valid, e.g. (Kraus & Wilkenfeld 1991).

We consider a group N of n autonomous agents, $N = \{a_1, a_2, \ldots, a_n\}$. The agents are provided with, or have access to, resources. Each agent a_i has its own resource vector $q_i = \langle q_i^1, q_i^2, \ldots, q_i^l \rangle$, which denotes the quantities of resources that it has. The agents use the resources they have to execute their tasks. a_i's outcome from task-execution is expressed by a payoff function from the resources domain Q to the reals. Such a function $U^i : Q \rightarrow \mathcal{R}$ exchanges resources into monetary units. Each agent tries to maximize its payoff.

Individual self-motivated agents can cooperate by forming coalitions. A coalition is defined as a group of agents that have decided to cooperate and have also decided how the total benefit should be distributed among them. Formally:

Definition 1 Coalition *Given a group of agents N and a resource domain Q, a coalition is a quadrate $C = \langle N_C, q_C, \overline{Q}, U_C \rangle$, where $N_C \subseteq N$; $q_C = \langle q^1, q^2, \ldots, q^l \rangle$ is the coalition's resource vector, where $q^j = \sum_{a_i \in N_C} q_i^j$ is the quantity of resource j that the coalition has. $\overline{Q}$ is the set of resource vectors after the redistribution of q_C among the members of N_C ($\overline{Q}$ satisfies $q^j = \sum_{a_i \in N_C} \overline{Q}_i^j$). $U_C = \langle u^1, u^2, \ldots, u^{|C|} \rangle$, is the coalitional payoff vector, where $u^i \in \mathcal{R}$ is the payoff of agent a_i after the redistribution of the payoffs. V is the value of C if the members of N_C can jointly reach a payoff V. That is, $V = \sum_{a_i \in N_C} U^i(\overline{Q}_i)$, where U^i is the payoff function of agent a_i and $\overline{Q}_i$ is its resource-vector after redistribution in C.*

The specific distribution of the resources among the members of the coalition strongly affects the results of the payoff functions of the agents and thus affects the coalitional value. It is in equilibrium to reach a resource distribution that will maximize the coalitional value. Therefore, the resources are redistributed within $\overline{Q}$ in a way that maximizes the value of the coalition. Thus, the coalition value V of a specific group of agents N_C is unique. The complexity of computing the redistribution of the resources and calculating the coalitional value depends on the type of payoff function of the coalition-members. For example, if the payoff functions are linear functions of resources, then polynomial calculation methods can be applied.

Coalition-formation usually requires disbursement of payoffs among the agents. We define a payoff vector $U = \langle u^1, u^2, \ldots, u^n \rangle$ in which its elements are the payoffs to the agents[2]. In each stage of the coalition formation process, the agents are in a coalitional configuration. That is, the agents are arranged in a set of coalitions $\mathcal{C}= \{C_i\}$, that satisfies the conditions $\cup_i C_i = N$, $\forall C_i, C_j$, $C_i \neq C_j$, $C_i \cap C_j = \phi$. A pair of a payoff vector and a coalitional configuration are denoted by $PC(U,\mathcal{C})$, or just PC (Payment Configuration). Since we assume individual rationality, we consider only individually rational payment configurations (IRPC's in game theory, e.g.,(Rapoport 1970)).

A (rational) coalitional configuration space (CCS) is the set of all possible coalitional configurations such that the value of each coalition within a configuration is greater or equal to the sum of the payoffs of the coalition-members[3]. The size of the CCS is $O(n^n)$. A

[2] Note that U is a payoff vector of all of the agents while U_C is a payoff vector of the members of a specific coalition.

[3] It is very common to normalize the agents' payoffs to zero. In such a case, the requirement on the sum becomes simpler – the coalitional values must be non-negative.

payment configuration space (PCS) is a set of possible individually-rational PCs. That is, a PCS consists of pairs (U, C) where U is an individually-rational payoff vector and C is a coalitional configuration in CCS. Since for each coalitional configuration there can usually be an infinite number of payoff vectors, the number of PCs is infinite, and the PCS of all rational PC's is an infinite space.

We would like the resulting payoff vector of our coalition-formation model to be stable and Pareto-optimal. A payoff vector is Pareto-optimal if there is no other payoff vector that dominates it, i.e., there is no other payoff vector that is better for some of the agents and not worse for the others (Luce & Raiffa 1957). It seems in the best interest of individually rational agents to seek Pareto-optimal payoff vectors. However, a specific Pareto-optimal payoff vector is not necessarily the best for all of the agents. There can be a group of Pareto-optimal payoff vectors where different agents prefer different payoff vectors. This may lead to difficulty when agents negotiate cooperation and coalition formation. Pareto-optimality is not sufficient for evaluating a possible coalition for a specific agent, hence we present the concept of stability.

The issue of stability was studied in the game theory literature in the context of n-person games (Rapoport 1970; Luce & Raiffa 1957). These notions are useful for our purposes, when coalitions are formed during the coalition formation procedure. The members of such coalitions can apply these techniques to the distribution of the coalitional value. Game theorists have given several solutions for n-person games, with several related stability notions. In this paper we concentrate solely on the Kernel solution concept. However, we shall discuss the other solution concepts in an extended version of this paper.

The Kernel K

The kernel (Davis & Maschler 1965) is a PCS in which the coalitional configurations are stable in the sense that there is equilibrium between pairs of individual agents which are in the same coalition. Two agents a_1, a_2 in a coalition C, in a given PC, are in equilibrium if they cannot outweigh one another from C, their common coalition. a_1 can outweigh a_2 if a_1 is stronger than a_2, where strength refers to the potential of agent a_1 to successfully claim a part of the payoff of a_2 in PC.

During the coalition formation, agents can use the kernel concept to object to the payoff distribution that is attached to their coalitional configuration. This objection will be done by agents threatening to outweigh one another from their common coalition. Given a $PC(U, C)$, agents can make objections based on the excess concept. We recall the relevant definitions.

Definition 2 Excess *The excess (Davis & Maschler 1965) of a coalition C with respect to the coalitional configuration PC is defined by $e(C) = V(C) - \sum_{a_i \in C} u^i$, where u^i is the payoff of agent a_i in PC. C is not necessarily a coalition in PC.*

Given a specific PC, the number of the excesses with respect to the specific coalitional configuration is 2^n. Any change in the payoff vector U, either when the coalitional configuration changes or when it remains unchanged, may cause a change in the set of excesses. Such a change will require recalculation of all of the excesses. Agents use the excesses as a measure of their relative strengths. Since a higher excess correlates with more strength, rational agents must search for their highest excess, i.e., the surplus.

Definition 3 Surplus and Outweigh *The maximum surplus S_{ab} of agent a over agent b with respect to a PC is defined by $S_{ab} = MAX_{C|a \in C, b \notin C} e(C)$, where $e(C)$ are the excesses of all the coalitions that include a and exclude b, and the coalitions C are not in PC, the current coalitional configuration. Agent a outweighs agent b if $S_{ab} > S_{ba}$ and $u^b > V(b)$, where $V(b)$ is the value of b as a single agent.*

If two agents cannot outweigh one another, we say that they are in equilibrium. We say that a, b are in equilibrium if one of the following conditions is satisfied: (i) $S_{ab} = S_{ba}$; (ii) $S_{ab} > S_{ba}$ and $u^b = V(b)$; (iii) $S_{ab} < S_{ba}$ and $u^a = V(a)$. Using the concept of equilibrium, the kernel and its stability are:

Definition 4 Kernel and K-Stability
A PC is K-stable if $\forall a, b$ agents in the same coalition $C \in PC$, the agents a, b are in equilibrium. A PC is in the kernel iff it is K-stable.

The kernel stability concept provides a stable payoff distribution for *any* coalitional configuration in the rational CCS (Davis & Maschler 1965; Aumann, Peleg, & Rabinowitz 1965)[4]. Using this distribution, the agents can compare different coalitional configurations. However, checking the stability does not direct the agents to a specific coalitional configuration. The coalition formation model that we develop will perform this direction. The kernel leads symmetric agents to receive equal payoffs. Such symmetry is not always guaranteed in other solution concepts (e.g., the bargaining set) (Rapoport 1970). Another property of the kernel – it is a comparatively small subset of the PCS of all rational PC's. In addition, the mathematical formalism of the kernel allows one to divide its calculation into small processes, thus simplifying it. Some exponentially-complex computing schemes for the kernel solution were provided, e.g., by (Aumann, Peleg, & Rabinowitz 1965). Stearns (Stearns 1968) presented a transfer scheme that, given a coalitional configuration and a payoff vector, converges to an element of the kernel. Due to its advantages, the kernel was chosen as

[4]The kernel does not provide coalitional configurations, it only determines how the payoffs will be distributed given a coalitional configuration. Therefore, it cannot be used as a coalition formation algorithm.

the theoretical background for our coalition formation model.

Coalition Negotiation Algorithm CNA

The CNA is a coalition formation algorithm based on negotiation. It consists of steps in which coalitions transmit, accept and reject proposals for creating new coalitions. Initially, all agents are in single-membered coalitions. The CNA proceeds through a sequence of steps as described below. In each step, at least one coalition will make an attempt to improve the payoffs of its members by making a coalition formation proposal to another coalition. The CNA may continue either until all of the proposals of all of the agents are rejected or until a K-stable and Pareto-optimal PC has been reached, thus reaching a steady state. The CNA may also terminate when the allocated time-period ends.

When the agents use the CNA, the coalitional configurations that are formed when the agents reach the steady state are stable according to a new stability concept that we define below, the polynomial-K-stability. However, since the CNA is an anytime algorithm (Dean & Boddy 1988), even if it is terminated after a limited number of steps *before* reaching a steady state, it will still provide the agents with a polynomial-K-stable PC.

The Polynomial Approach

We modify some concepts to adjust them to the polynomial-K-stability algorithm. Polynomial excesses are excesses that are calculated with respect to a polynomial subset of all 2^n possible coalitions. The designers of agents must agree upon regulations that will direct their agents to a well defined polynomial set of coalitions. We suggest that the designers agree upon two integral constants K_1, K_2. These constants $K_1 \leq K_2$ should not depend on n. Nevertheless, K_1, K_2 small w.r.t. $n/2$ should be preferred. In the regulation of the coalition formation model, the agents shall be allowed to consider excess calculations only for coalitions of sizes in the ranges $[K_1, K_2]$. Choosing K's contradictory to the agreed upon K_1, K_2 by a specific agent shall be avoided, mainly because according to the regulations objections based on different K's are not acceptable.

Definition 5 *A polynomial maximum surplus SP is a maximum surplus that is computed from a set of polynomial excesses. A coalition C in a coalitional configuration is polynomially-K-stable if for each pair of agents $a, b \in C$, either one of the agents has a null normalized payoff in C, or $|SP_{ab} - SP_{ba}| \leq \varepsilon$, i.e., the agents are in equilibrium with respect to ε, where SP are the polynomial surpluses, and ε is a small predefined constant.*

Given a specific coalitional configuration with an arbitrary payoff distribution vector, it is possible to compute a polynomial-K-stable PC by using a truncated modification of the convergent transfer scheme in (Stearns 1968). We implement the Stearns-scheme by using the n-correction of Wu (Wu 1977)[5] to initialize the process. The iterative part of the scheme is modified so that it will terminate whenever a payoff vector that is close, according to the predefined small ε, to an element of the polynomial kernel has been reached.

A CNA Scheme

The CNA scheme is aimed at advancing the agents from one coalitional configuration to the other, to achieve more cooperation and increase the agents' payoffs. Given a specific configuration, the agents attempt to find a correspondent *stable* payoff vector. The social level of the CNA will be constructed as follows:

Regulation 1 Negotiation scheme

1. *Initially, all entities are single agents.*
2. **First stage:** *members of a coalition may receive proposals only as part of the coalition (thus, coalitions can only expand in this stage).*
3. *Each coalition will coordinate its actions either via a representative or by voting (or both) e.g., (Peleg 1984).*
4. *Each coalition C_p iteratively performs the following:*
   - *Decide which other coalitions it is interested in forming a joint coalition with.*
   - *Design proposals to be offered to others:*
     - *A proposal of C_p to C_r is the details of the joint coalition C_{new} and coalitional configuration PC_{new}: must be polynomially-K-stable, calculated with respect to K_1,K_2.*
     - *C_p will design C_{new}, $N_{new} = N_p \cup N_r$, with payoff vectors U_{new} that increase C_{new}'s payoffs.*
     - *In addition to C_{new}, C_p will design the coalitional configuration in which C_{new} is included. Other coalitions stay unchanged in PC_{new}.*

     transmit one proposal to one target coalition at a time; wait for response.
   - *When an offer is accepted, the offering coalition and the accepting coalition form a new coalition according to the details of the proposal*[6]*. The new PC determines the payoffs of the agents.*
   - *Other active proposals will be canceled.*
5. *If C_p has no more proposals to make, it shall announce it.*
6. *If a steady state is reached, where all coalitions announce that they have no proposals, proceed to second stage of the CNA. Otherwise, start a new iteration.*
7. *If the agents run out of computation time before a steady state has been reached, the algorithm terminates and the last PC holds.*

[5] While Wu uses this correction as a means for iterating in a transfer scheme for the core, we use it as a single correction in the beginning of our iterative algorithm. This n-correction is not necessary in the Stearns scheme.

[6] The payoff vector U of the new PC is valid from now on and is used as the basis for future negotiation.

8. **Second stage:** *the coalitions will follow the same sequence of steps as in the first stage of the CNA. However, proposals that involve destruction are allowed.*
9. *When a new PC changes the payoffs, dissatisfied agents may leave their coalitions. These coalitions will destruct.*
10. *The second stage will end either when a steady state*[7] *is reached or when the computation time ends.*

The regulation above is enforceable since deviation from it is revealable. For example, proposals addressed to coalition-members will be detected immediately when accepted due to the coalition formation. The limitation that destruction of coalitions be avoided in the first stage will radically shorten the coalition formation process by avoiding most of the intra-coalitional computation and communication in this stage. The number of iterations for reaching a steady state in the first stage of the CNA is $O(n)$. If the agents have enough time and computational resources they will continue through stage 2. The number of steps until a steady state is reached in the second stage may be $O(n^n)$ due to the size of the PCS.

An acceptance of a proposal implies an acceptance of the corresponding payoff vector by all agents. This may change the payoffs of agents who are not involved in the negotiation, because a proposal consists of a payoff vector to all of the agents. The aim of the change in the payoffs is to preserve the polynomial-K-stability. Since some of the agents may be dissatisfied with their new payoff, they can make new proposals according to which they receive a greater payoff.

CNA Strategies

According to part 4 of regulation 1, proposals for the generation of new coalitions should be designed by the current coalitions. We denote a coalition that designs and transmits a proposal by C_p and a coalition that receives a proposal by C_r. Other coalitions will be denoted by C_a. We suggest that C_p, that designs a proposal for C_r, use the following strategy:

Strategy 1 Proposal design *Coalition C_p shall calculate the coalitional value of the joint coalition V_{p+r}. If $V_p + V_r \geq V_{p+r}$ then C_p shall stop designing a proposal for C_r. Otherwise, C_p shall calculate the coalitional values of all other coalitions of all sizes in the range $[K_1, K_2]$. C_p shall calculate the payoff vector U_{new} of the new PC wherein coalitions C_p and C_r join to form C_{new}, and all the other coalitions do not vary. These calculations will be done by using the truncated transfer scheme, starting from the initial payoff vector U. Coalition C_p will compare U_{new} to U. If the payoffs to all of the members of C_r and C_p in PC_{new} are not smaller than their payoffs in the current PC and are also better than in all of the proposals that C_p has in the received-proposals queue then coalition C_p will send the resultant PC_{new} as a proposal to coalition C_r. Otherwise, C_p shall stop the process of designing a proposal for C_r.*

[7] Note that the steady state in the second stage is different from the steady state in the first stage since there are different restrictions on the proposals in these two stages.

This strategy for proposal design shall be used by agents that are interested in reaching beneficial coalition formation and act under the regulations, which forces polynomial-K-stability of proposals. This is because the calculation of the new coalitional value V_{p+r} and the comparison to the sum of original coalitional values V_p and V_r is done to avoid worthless proposals in advance. In cases where V_{p+r} enables beneficial coalition joining, coalition C_p shall seek all coalitional values (of coalitions of sizes in the range $[K_1, K_2]$) in order to use these values for PC calculations.

Strategies are not enforced and agents can act without using our strategies. We propose them in order to increase the payoff to the individual agent and because they satisfy the weak equilibrium requirement. A set of strategies is in weak equilibrium if none of the entities that act according to these strategies can guarantee, by deviating from its strategy, an increase in its benefits. An entity may be able to calculate all possible proposals and all of their consequences. However, due to time and communication uncertainties, it cannot predict the exact results of the negotiation, and therefore it cannot guarantee an increase in its payoff.

Complexity of the CNA

The complexity of calculation of the polynomial set of coalitions, the coalitional values and the coalitional configurations is of the same order of the number of the coalitions which is given by

$$n_{coalitions} = \sum_{i=K_1}^{K_2} \binom{n}{i} = \sum_{i=K_1}^{K_2} \frac{n!}{i!(n-i)!}$$

which is a sum of polynoms of order $O(n^i)$.

Computation of values and configurations
The CNA requires the computation of $n_{coalitions}$ coalitional values. It also requires the design of coalitional configurations, and the number of these depends on the time constraints. In a case where only the first stage of the CNA is performed, the number of coalitional configurations which are treated is $O(n)$. If the CNA proceeds through the second stage, the number of coalitional configurations increases. In each iteration of the CNA, when one coalitional configuration is treated, a polynomial-K-stable PC shall be calculated.

Computation of polynomial-K-stable PC's
The CNA will employ the transfer scheme for calculating polynomial-K-stable PC's. The total complexity of one iteration of the transfer scheme is $O(n \times n_{coalitions})$. The number of iterations that should be performed to reach convergence depends on the predefined allowed error ε. The resulting payoff vector of the transfer

scheme will converge to an element of the polynomial-kernel (with a relative error not greater than ε) within $n \log_2(e_{r_0}/\varepsilon)$ iterations (Stearns 1968), where e_{r_0} is the relative error of the initial PC.

The transfer scheme will be performed for $O(n)$ coalitional configurations in the first stage of the CNA and up to $O(n^n)$ in the second stage. Therefore, the complexity of the CNA is of at least $O(n^2 n_{coalitions})$ and up to $O(n^n n_{coalitions})$ computations. If the computations are distributed among the agents, this order of complexity of computations is divided by n. There is an additional communicational complexity, which is of order $O(n^2 n_{coalitions})$.

Implementation

figure 1

The performance of the CNA was tested with respect to different constants (K's,ε) and different environmental settings. Running the simulation has provided several results as presented below. Initially, we have shown that the simulated CNA reaches a stable PC within a reasonable time (for the first stage of the CNA). In addition, it has been found (for the settings that we have examined) that the CNA continuously improves the agents' payoffs, whether it is normally terminated or halted artificially. The main results of the simulation for 5 through 13 agents, without limitation on K_1 and K_2 and with $\varepsilon = 1$ (i.e., less than 1% of the average coalitional value) are as follows:

1. The number of agents that participate in coalitions is an increasing monotonic function of the average of potential coalitional values. The most appropriate analytical curve-fit to the results is a logarithmic function, as in figure 1. We can also conclude that the increment in the coalitional values is not a sufficient condition for increasing the number of coalitions' members. This may arise from unresolvable conflicts that are present in non-super-additive environments.
2. In cases where cooperation is beneficial, we observe (figure 2) that the utility is growing as a function of the number of coalition formations. This means that not only it is beneficial to form coalitions, formations of more coalitions increase the average benefits of the agents.

figure 2

3. As expected, the time necessary for coalition formation without bounding the K's is an exponential function of the number of agents that comprise the agent-system (see figure 3). However, as can be observed from the graph, this exponent is not too steep. For example, in a system of 13 agents, where each agent is implemented on an Intel® 486 processor, the computation time per agent is only a few minutes.
4. We also observed that the use of the algorithm by agents does not violate, in average, their individual rationality.

According to these intermediate results, it shows that the CNA is a good coalition formation model for MA systems in general environments. We shall report more results in future work.

figure 3

Conclusion

The CNA is useful for instances where the number of agents may be large (e.g., tens of agents), computations are costly and time is limited. This is because the model leads to coalition formation within a polynomial time and a polynomial amount of calculations. We introduce the *polynomial* K-stability. The original K-stability refers to a PC where agents cannot make justified objections, using (exponential) surplus calculations. The new polynomial K-stability entails polynomial objection calculations.

The CNA leads to distribution of both calculations and communications. In addition, it is an anytime algorithm: if halted after any negotiation step, it provides the agents with a set of formed polynomial-K-stable coalitions. A deficiency of the CNA is that in polynomial time it cannot guarantee that a Pareto-optimal PC will be reached. However, for calculating a Pareto-optimal PC all of the coalitional configurations in the CCS shall be approached, and therefore there cannot be any polynomial method to find Pareto-optimality. An important advantage of our algorithm is that the average expected payoff of the agents is an increasing function of the time and effort spent by the agents performing the CNA steps. Therefore, if cooperation is beneficial for the agents, using the CNA will always improve their payoffs. The last property, the anytime and distribution properties and other advantages of the CNA, have been proved via simulations.

The model we present is not restricted to the superadditive environment, for which there are already several coalition formation algorithms in DAI (Shehory & Kraus 1993; Zlotkin & Rosenschein 1994; Klusch & Shehory 1996). However, the generality of the model does not make it inapplicable. As opposed to the majority of solution concepts presented in game theory, we present a detailed method for how the individual agent should act in order to form coalitions that increase its personal payoff.

References

Aumann, R. J.; Peleg, B.; and Rabinowitz, P. 1965. A method for computing the kernel of n-person games. *Mathematics of Computation* 19:531–551.

Bond, A. H., and Gasser, L. 1988. An analysis of problems and research in DAI. In Bond, A. H., and Gasser, L., eds., *Readings in Distributed AI*. California: Morgan Kaufmann Publishers, Inc. 3–35.

Davis, M., and Maschler, M. 1965. The kernel of a cooperative game. *Naval research Logistics Quarterly* 12:223–259.

Davis, R., and Smith, R. 1983. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20:63–109.

Dean, T., and Boddy, M. 1988. An analysis of time-dependent planning. In *Proceedings, AAAI88*, 49–54.

Decker, K., and Lesser, V. 1993. A one-shot dynamic coordination algorithm for distributed sensor networks. In *Proc. of AAAI93*.

Durfee, E. H., and Rosenschein, J. S. 1994. Distributed problem solving and multi-agent systems: Comparisons and examples. In *Proc. of 13th International Distributed AI Workshop*, 94–104.

Durfee, E. H. 1988. *Coordination of Distributed Problem Solvers*. Boston: Kluwer Academic Publishers.

Gasser, L. 1993. Social knowledge and social action. In *IJCAI93*, 751–757.

Ketchpel, S. P. 1994. Forming coalitions in the face of uncertain rewards. In *Proc. of AAAI94*, 414–419.

Klusch, M., and Shehory, O. 1996. Coalition formation among rational information agents. In de Velde, W. V., and Perram, J. W., eds., *LNAI No. 1038, Agents Breaking Away*. Springer-Verlag. 204–217.

Kraus, S., and Wilkenfeld, J. 1991. Negotiations over time in a multi agent environment: Preliminary report. In *Proc. of IJCAI-91*, 56–61.

Luce, R. D., and Raiffa, H. 1957. *Games and Decisions*. John Wiley and Sons, Inc.

Malone, T. W.; Fikes, R. E.; Grant, K. R.; and Howard, M. T. 1988. Enterprise: A marketlike task schedule for distributed computing environments. In Huberman, B. A., ed., *The Ecology of Computation*. Amsterdam: North Holland. 177–205.

Moses, Y., and Tennenholtz, M. 1993. Off-line reasoning for on-line efficiency. In *IJCAI93*, 490–495.

Peleg, B. 1984. *Game Theoretic Analysis of Voting in Commities*. Cambridge University Press.

Rapoport, A. 1970. *N-Person Game Theory*. University of Michigan.

Sandholm, T. W., and Lesser, V. R. 1995. Coalition formation among bounded rational agents. In *Proc. of IJCAI-95*, 662–669.

Shapley, L. S., and Shubik, M. 1973. *Game Theory in economics*. California: Rand Corporation.

Shehory, O., and Kraus, S. 1993. Coalition formation among autonomous agents: Strategies and complexity. In Castelfranchi, C., and M J. P., eds., *LNAI No. 957, From Reaction to Cognition*.

Shoham, Y., and Tennenholtz, M. 1992. On the synthesis of useful social laws for artificial agent societies. In *Proc. of AAAI-92*, 276–281.

Stearns, R. E. 1968. Convergent transfer schemes for n-person games. *Transactions of the American Mathematical Society* 134:449–459.

Werner, E. 1988. Toward a theory of communication and cooperation for multiagent planning. In *Proc. of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, 129–143.

Wu, L. S. 1977. A dynamic theory for the class of games with nonempty cores. *Siam Journal of Applied Mathematics* 32:328–338.

Zlotkin, G., and Rosenschein, J. S. 1994. Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains. In *Proc. of AAAI94*, 432–437.