Coalition formation among autonomous agents: Strategies and complexity (preliminary report) *

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. Autonomous agents are designed to reach goals that were pre-defined by their operators. An important way to execute tasks and to maximize payoff is to share resources and to cooperate on task execution by creating coalitions of agents. Such coalitions will take place if, and only if, each member of a coalition gains more if he joins the coalition than he could gain before. There are several ways to create such coalitions and to divide the joint payoff among the members. Variance in these methods is due to different environments, different settings in a specific environment, and different approaches to a specific environment with specific settings. In this paper we focus on the cooperative (super-additive) environment, and suggest two different algorithms for coalition formation and payoff distribution in this environment. We also deal with the complexity of both computation and communication of each algorithm, and we try to give designers some basic tools for developing agents for this environment.

* This material is based upon work supported in part by the NSF under Grant No. IRI-9123460.

1 Introduction

Cooperation among autonomous agents may be mutually beneficial even if the agents are selfish and try to maximize their own expected payoffs [9, 20, 24]. Mutual benefit may arise from resource sharing and task redistribution. Coalition formation is an important method for cooperation in multi-agent environment. Agent membership in a coalition may increase the agent's ability to satisfy its goals and maximize its own personal payoff. There are two major questions concerning coalition formation: 1. which procedure should a group of autonomous agents use to coordinate their actions and cooperate; namely, how should they form a coalition? 2. among all possible coalitions, what coalition will form, and what reasons and processes will lead the agents to form that particular coalition?

Work in game theory such as [14, 17, 10, 12] 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. That is, they concentrate mainly on the second problem above. These results do not take into consideration the constraints of a multi-agent environment, such as communication costs and limited computation time.

In this paper we adjust the game theory concepts to autonomous agents, and present different coalition-formation procedures. The resulting coalitions of each of these procedures may be different even in the same setting. We developed general criteria for choosing between the coalition-formation procedures. Giving a specific setting and using these criteria will enable the agents to decide which coalition-formation procedure to prefer. We will concentrate on widely cooperative environments [4, 7] such as the postmen problem of [23]. The coalition-formation procedures that will be presented are either computation-oriented or negotiation-oriented. The appropriate procedure can be chosen according to the constraints of the environment and will lead to beneficial coalition formation.

We begin by describing the environment with which we will deal and give the basic definitions of coalitions in section 2.1. In section 3 we concentrate on super-additive environments. The negotiation protocol is described in section 3.1, together with its complexity analysis. The Shapley value protocol and its complexity are described in section 3.2. Finally, we give the criteria for choosing between the two protocols.

2 Environment description

This 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. Resources may include: materials, energy, information, communications and others. Autonomous agents can act and reach goals by themselves, but may also join together to reach some or all of their goals. In such a case we say that the agents form a coalition.

In order to deal with different types of agents' environments, we give some general notations and definitions for concepts (i.e., resources, tasks, methods of reaching goals, success in reaching them, etc). Resources will be treated by numbers which will denote the quantities of each. Success in fulfilling tasks will be defined as the production or the payoff of an agent. The concept of payoff will be preferred specifically when reaching tasks is exchangeable with some kind of payment (e.g., money). The ways of reaching goals can be formalized by functions from the resources to the production or to the payoff. Each agent will have such a function of its own that tells what its way of using the resources to reach goals is.

An example of such an environment is the postmen domain [23]. While Zlotkin and Rosenschein consider only bi-agent environments (e.g., two postmen), we provide cooperation procedures for multi-agent environments (e.g., several agents, possibly more than two). In addition, Zlotkin and Rosenschein do not consider both the resource allocation problem and task distribution in the same setting e.g., we provide the postmen with procedures for division of their overall transportation capabilities and the benefits from fulfilling the letter distribution.

In order to make the creation of mutually beneficial coalitions possible, we make the following two assumptions:

Assumption 1 Communication
We assume that various communication methods exist, so that the agents can negotiate and make agreements [22]. We also assume that communications require time and effort on the part of the agent.

Assumption 2 Goods transferability & side-payments
We assume that resources and products can be transferred between agents in the environment, and that there is a monetary system that can be used for side-payments.

Multi-agents may reach agreements and form coalitions even if the second assumption is not valid (e.g., [10, 23, 1, 2]). However, the ability of goods transferability or side-payments may help the agents form more mutually-beneficial coalitions.

Our postmen example satisfies the assumptions above; they can communicate, and letters, transportation and money (their resources and production) can be transferred.

2.1 Definitions

Let N be a group of n autonomous agents, $N = \{A_1, A_2, \ldots, A_n\}$. Let L be a set of m resources, $L = \{l_1, l_2, \ldots, l_m\}$. The resources in the environment are limited. We consider only the resources that are distributed among the agents. This is formalized by the following definition:

Definition 1. Resource domain
We say that Q is the possible resource domain if Q is a set of the possible vectors of quantities of resources, $Q = \{\langle q^1, q^2, \ldots, q^m\rangle\}$, where q^j is a quantity of the resource l_j.

Each agent A_i has its own resources, therefore it has a vector $q_i \in Q$, $q_i = \langle q_i^1, q_i^2, \ldots, q_i^m \rangle$, which denotes the quantities of resources that A_i has. Due to the limitation of the quantities of the resources, we define $q_D \in Q$ to be the vector of the quantities of the resources of the domain. It is obvious that every element k of q_D,must satisfy $q_D^k = \sum_{j=1}^{n} q_j^k$, where q_j^k is agent A_j's quantity of resource k. The vector q_D denotes the total amounts of resources available to all agents in the environment, together.

We are interested in q_D particularly in cases of cooperation between agents, where q_D may sometimes be re-distributed among them. The agents use the resources they have to execute their missions. Depending on the quantities of resources an agent has and on its way of using them to fulfill tasks, each agent can reach all, part or none of its goals. For example, the postmen resources are the letters they have to distribute and their transportation abilities.

Production is a quantitative way of measuring the agent's success in fulfilling tasks. As such, we shall call production any kind of success in fulfilling tasks or reaching goals. Therefore, we give the following definition:

Definition 2. Production set
We say that P is a production set if P is a set of values which are the possible production values of a group of agents.

For example, if the agents are postmen, then the production set may be the set of amounts of letters that they can distribute, or their potential income.

Each agent in the environment has its own method of using the resources to reach goals. Therefore, we say that each agent A_i has a function f_P^i that formulates its way of using resources to reach goals; this function is defined as:

Definition 3. Production Function
We say that $f_P^i : Q \rightarrow P$ is a production function, if f_P^i is a function which gives a value within the production set that measures the production of agent A_i with an arbitrary resources vector.

It may be inconvenient to use the production function for agents' decision making. Each designer of automated agents has to provide its agent with a decision mechanism based on some given set of preferences [21]. Therefore, we suggest that each autonomous agent will be provided with a numerical payoff function that gives a transformation from the production set values and the resources to the reals. Each agent A_i has such a function, U^i, that exchanges its resources and production into monetary units. This monetary system can be used for evaluation of production and for side-payments.

Definition 4. Payoff Function
We say that $U^i : (P, Q) \rightarrow \mathcal{R}$ is a payoff function if U^i gives a measure, in monetary units, of the outcome that an agent has for some arbitrary resources and production.

A postman's payoff function is the difference between his transportation costs and the payment that he receives for distributing letters. The payments that postmen receive are transferable among them.

Individual self-motivated agents may be cooperative; they may cooperate by sharing resources, redistributing tasks and passing side-payments. Self-motivated agents will cooperate only if, as a result of this cooperation, they increase their payoff.

A coalition can be defined as a group of agents that decided to cooperate and also decided how the total benefit should be disbursed among its members. Formally, we define:

Definition 5. Coalition
Given a group of agents N, a resource domain Q, and a production set P, we define a coalition as a quadrate $C = \langle N_C, Q_C, \overline{q}, U_C \rangle$. In this quadrate, $N_C \subseteq N$; $Q_C = \langle q^1, q^2, \ldots, q^m \rangle$, $Q_C \in Q$, is a coalitional resource vector, where $q^j = \sum_{A_i \in N_C} q_i^j$ is the quantity of resource l_j that the coalition has. $\overline{q}$ is the set of vectors of resource quantities after the redistribution of Q_C among the members of N_C ($\overline{q}$ satisfies $q^j = \sum_{A_i \in N_C} \overline{q}_i^j$). $U_C = \langle u^1, u^2, \ldots, u^{|C|} \rangle$, $u^i \in \mathcal{R}$, is the coalitional payoff vector, where u^i is the payoff of agent A_i after the redistribution of the payoffs.

We say that $\mathcal{C}$ is a set of possible coalitions if $\mathcal{C}$ is the group of all possible coalitions over N. In order to provide the agents with a method for coalition evaluation, we give each coalition a value (as can be found in game theory [12]), and a function for calculating this value.

Definition 6. Coalition Value & Coalitional Function
Let $C = \langle N_C, Q_C, \overline{q}, U_C \rangle$. We say that V is the value of C if the members of N_C can together reach a joint 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 vector of resources after their redistribution in the coalition.

Let q_i be agent A_i's original vector of resource quantities; we assume that $\forall A_i$, $\overline{q}_i \in \overline{q}$, $u^i \geq U^i(q_i)$. That is, an agent will join a coalition only if the payoff it will receive in the coalition is greater than, or at least equal to, what it can obtain by staying outside the coalition. Hence, it is easy to conclude that $V \geq \sum_{A_i \in C} U^i(q_i)$. Moreover, we assume that the resources are redistributed within $\overline{q}$ in a way that maximizes the value of the coalition. Therefore, 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 structure of the coalition's members' payoff functions. For example, for linear payoff functions the simplex method can be used.

Now, let us take the production function in definition 3 and expand its scope: We say that f_P^C :C$\rightarrow P$ is a coalitional production function if f_P^C attaches a value $p \in P$ to all coalitions in $\mathcal{C}$. We say that $U^C : (P, Q) \rightarrow R$ is a coalitional payoff function if U^C transforms the coalitional production into coalitional payoff.

We now formally state more assumptions that consider a multi-agent environment in which the agents are self-motivated and try to maximize their own payoff function.

Assumption 3 Coalition joining (personal rationality)
We assume that each agent in the environment has personal rationality, i.e., it joins a coalition only if it can benefit at least as much within the coalition as it could benefit by itself. An agent benefits if it fulfills some or all of its tasks, or gets a payoff that compensates it for the loss of resources or non-fulfillment of some of its tasks[2].

Assumption 4 Personal payoff maximization
We assume that each agent in the environment tries to maximize its personal payoff; among all the possibilities that it has, an agent will choose the one that will give it the maximum expected payoff [3].

Our postmen example satisfies the two assumptions above: they are personally rational, and they try to maximize their personal payoff.

Now we shall define a new concept – the coalitional rationality. The environment is coalitionally rational if each coalition in it will add a new member only when the value of the coalition that will be formed by this addition is greater than (or at least equal to) the value of the original coalition. Not all environments are coalitionally rational. Situations come to mind whereby an agent is added to an existing coalition only because of the additional benefit for powerful agents and for the newly-added agent. Thus, the assumption of personal rationality is fulfilled, while the value of the new coalition is lower than the value of the original coalition. One can say that such a situation is not likely to happen, and can justify this statement by claiming that if some agents benefit more and the value of the new coalition is less than the value of the original one, then there must be at least one agent that will benefit less in the new coalition. That agent should never let such a coalition form, as the personal rationality assumption dictates. We reject this claim and say that such situations may happen if, after the change in the coalition, the agents that benefit less still prefer to stay in the new coalition, because they believe that this is the best option, given the new situation.

3 Super-additive environment

The concept of coalitional rationality can be broadened if we project it from the relation between a coalition and a single agent to the relation between two coalitions. By this expansion we define the super-additive environment:

Definition 7. Super-additive environment
A super-additive environment is a set of possible coalitions $\mathcal{C}$ that satisfies the following rule: for each pair of coalitions C_1,C_2 in the set $\mathcal{C}$, $C_1 \cap C_2 = \phi$, if C_1,C_2 join together to form a new coalition, then the new coalition will have a new value $V_C^{new} \geq V_C^1 + V_C^2$, where V_C^1,V_C^2,V_C^{new} are the values of the coalitions.

[2] This assumption is usually called "personal rationality" in the game theory literature [8, 17, 12].

[3] Note that assumption 3 can be derived from assumption 4. We present here both assumptions only for clarification purposes.

Not all environments are super-additive, but if an environment is super-additive, then, under assumptions A1 - A4, after a sufficient time period a grand coalition will form. A grand coalition is a coalition that includes all of the agents. We come to the conclusion that a grand coalition will form because in any other situation there will be at least two coalitions that are not a grand coalition. If the environment is super-additive then the coalitional rationality and the personal payoff maximization will make them join together and form a joint coalition, that, according to the super-additivity concept, will have a value $V^{new} \geq V^1 + V^2$ [18]. We can let the agents negotiate coalition formation, but all agents know that in a super-additive environment a grand coalition will form. Hence, the only problem that still remains is how the payoff should be distributed among its members. It seems that the best way to save time and effort for all agents will be to agree, without any argument, to form a grand coalition, and to solve the problem of payoff distribution either by negotiation or by calculation.

Definition 8. Common extra payoff
In a super-additive environment, if two coalitions C_i, C_j with corresponding coalitional values V_i, V_j can form a joint coalition and obtain together $V_{ij} = V_i + V_j + D_{ij}$, then D_{ij} is the common extra payoff (the indices i, j may be dropped).

We present two algorithms for solving the problem stated above: one that requires negotiations and another that requires only computations. We discuss the advantages of each later.

3.1 Negotiation in a super-additive environment

The initial coalitional state consists of n, single agent, coalitions. The coalitions then begin negotiating [11] and, step by step, form coalitions. Coalitions will continue negotiations via agents who are representatives of the coalitions of which they are members. These will form bigger coalitions, until a grand coalition forms. At the beginning of each step of this coalition formation process, each coalition will find what the common extra payoff from forming a new coalition with each of the other current coalitions is. Next, each coalition will sort its list of extra payoffs. The first coalition on each sorted list (i.e., the one that can bring maximum extra payoff) is "wanted" by the coalition that made this list. Two coalitions that want one another can start a bargaining process. In a super-additive environment, at least one such pair must exist. If there is more than one pair, then all will start a bargaining process. A bargaining process entails coalitions offering one another partition of their common D. A coalition that received such an offer may accept it, and then a new joint coalition can form. A coalition may also reject the offer, and either ask for a better one, or make its own offer. To establish order in the bargaining process, we define a strength relation between coalitions, and use this relation to determine which coalitions will be first to make offers.

Strong coalition We shall define a strong coalition by saying that coalition C_i is stronger than coalition C_j, if C_i has an extra payoff D_{ik} with some other coalition C_k, $k \neq j$, which is bigger than D_{ij}. It can be shown that two members of a pair of coalitions that are "wanted" by one another (such as that mentioned above), are both stronger than all other coalitions[4]; therefore we shall call them "powerful" [3]. We denote these two coalitions as C_p^1, C_p^2.

The bargaining process is begun by the powerful coalitions. Within the pair of powerful coalitions, the coalition with the higher computational capabilities will be the first to calculate an offer and send it to its partner. We shall designate this first coalition as C_1^p and its associate as C_2^p. The first offers made to one another will be exactly half of D, their joint extra payoff[5]. Then, each powerful coalition will contact all coalitions that are weaker than it, inform them of the amount of the first offer it received and find among them the one coalition that is willing to give it the largest share of their common D. Any strong coalition, having received an offer (or offers) with δ higher than that presented by powerful coalition, will use the offer to challenge this powerful coalition. That is, the strong coalition will ask the powerful coalition for exactly as much of the common D as it could get from other coalitions.

At this point, the powerful coalition C_p^1 must contact its parallel coalition C_p^2, inform it about: 1. the amount of the highest offer it received, and 2. the coalition C_w^i that offered it. If C_p^2 was also challenged, and its highest offer was given by a coalition C_w^j, $i \neq j$, then C_p^1 and C_p^2 will reject one anothers' offers and each will join with its challenger. If both C_p^1 and C_p^2 are challenged by C_w^i, then C_p^1 has priority to join C_w^i and so C_p^2 should check if it has another challenging coalition. If so, then again, as above, both will reject one another. If C_p^2 was not challenged by the offers made, it should find the highest offer among these. If the difference between D_{12}^p and this highest offer is greater than C_p^1's highest offer, C_p^2 will accept the new offer and join C_p^1. Otherwise it will reject C_p^1 and C_p^1 will join C_w^i.

Weak coalition Each weak coalition, C_w, that was approached by another coalition, C_s, will calculate what it can offer by considering two issues: a. whether or not it is able to offer at least as much as C_s could get without cooperation with C_w; b. if the answer to (a) is positive, how much more it can offer. Consequently, C_w will follow this procedure: among all coalitions, excluding C_s, the coalition C_w will find those that are weaker or equal to it (the stronger ones will approach C_w). If there are none, then C_w will offer C_s as much as C_s claims it can get, plus ε (ε is a positive infinitesimally small number), up to a maximum of all of their common D_{ws}. This proposition is valid, unless D_{ws} is smaller than the offer that C_s already has – then C_w will have to offer D_{ws}, which will

[4] Note that it can be that C_i is stronger than C_j and C_j is stronger than C_i.

[5] This partition is suggested only as a fair starting point of the negotiation. During the negotiation process the partition of D will change according to the relative strength of the coalitions.

probably be rejected. If weaker coalitions exist, C_w will contact them, and each of these coalitions will have to go through the same procedure, recursively, to calculate their answer to the calling coalition. After C_w was answered by all of the contacted coalitions, it finds the one that offers it the maximum amount of payoff. This amount is the minimal part of D that C_w should ask for, from its common D with C_s, and the maximum is the amount that C_s claimed it could get, plus ε. To make it more clear, the following should be the typical answer of a requested coalition to its caller: "You said that you can get x, and therefore I am willing to give you $x + \varepsilon$, but if you get a better offer, I can give you a maximum of $x + \delta$, and I will still be satisfied".[6]

Payoff disbursement When two coalitions C_1 and C_2 join together to form a new coalition, they get extra payoff D_{12}, which is divided between them into D_1, D_2, such that $D_1 + D_2 = D_{12}$. Now, we must present a method for distributing D_1 and D_2 among the members of C_1, C_2, respectively. Under the assumptions of coalitional rationality and super-additivity, we suggest that once a coalition forms, it will not split up. Therefore, in order to make these assumptions acceptable to the members of a coalition, we say that within an existing coalition, each member keeps its previous strength – the strength that was expressed by the ratio between the total D and the member's part of it when it joined the coalition. Therefore, any new added payoff should be divided among the coalition members according to their strengths during the process of coalition formation. Of course, this does not prevent them from keeping for themselves any amount of payoff which they had before-hand.

Complexity of the negotiation algorithm The efficiency of this algorithm should be judged from two main perspectives: computations and communications. The first step of the negotiation process will be the transfer of all of the relevant information, i.e., payoff functions and resources vectors, between the agents (unless it is known in advance). This requires $(n-1)^2$ communication operations. The negotiation process will proceed from the initial state of n single agents to the final state of a grand coalition, and will take up to $n-1$ steps, since at each step the number of coalitions will decrease by at least 1. At each step, each determines its extra common payoff with all other coalitions; this requires, $n-1$ maximization operations, which are rather complicated. Upon deriving this information, each coalition must sort it ($o(n \lg_2 n)$), and then contact the first coalition on the sorted list to find out if they are a powerful pair. Next, all strong coalitions contact all weaker coalitions, and all weaker coalitions, after having finished their calculations, re-establish contact with the corresponding strong coalitions in order to answer them. Altogether, there are $2(n-1)^2$ communication operations, and each such operation is connected with $o(1)$ calculations. Afterwards, the powerful coalitions contact one another once again,

[6] This assumes honesty or complete information on both the coalitional values and the messages that are transmitted. Otherwise, it will require long and exhausting negotiations to reach an agreement.

and search for the highest offer they received. At last, coalitions join together to form new ones. Since, in the end, all coalitions join together, there can be at least $\lceil \lg_2(n-1) \rceil$ and at most $n-1$ such events. Any two coalitions that join together and form a new coalition, must perform the following actions:
a. choose a representative from among its members. This action will take $o(n)$ communication operations and the same order of computations.
b. distribute the extra payoff among all members. With a given simple algorithm to do so, the complexity of this action is $o(n)$.

Now that we have determined the complexity of all the parts of the algorithm, we can construct the complexity of the general algorithm. There are at most $n-1$ negotiation steps; at each step there are $o(n^2)$ communication operations, and $o(n^2 \lg_2 n)$ computations. Therefore, the complexity of the general algorithm is, in the worst case, $o(n^3)$ communication operations, and $o(n^3 \lg_2 n)$ computations.

Example We shall use here an example to illustrate the negotiation algorithm. There are three agents in a super-additive environment. Each agent receives no payoff by itself, so that the coalitional values of a single-member coalition is zero. The values of the other coalitions are $V_{12} = 10$, $V_{13} = 7$, $V_{23} = 2$, $V_{123} = 15$.
Agents A_1 and A_2 are the pair of powerful coalitions, since their common extra utility D_{12} is greater than their extra utility with A_3 (This is easy to conclude since in our specific example $\forall i, j,\ D_{ij} = V_{ij}$). Suppose that agent A_1 starts the negotiation by offering A_2 5, half of their common D. A_2 will offer A_1 the same amount. At this point A_1 will approach A_3, inform it about the current offer and ask for a better offer. Agent A_3 will offer A_1 $5+\varepsilon$, and will announce that it can offer up to 7, if necessary. Meanwhile, Agent A_2 will also approach A_3. The offer of A_3 to A_2 will be 2. With the offer A_1 has received from A_3 it will approach A_2 and ask for more than 5, since it can receive more than 5 from A_3. Agent A_2 will compete with A_3's offer by offering A_1 $7+\varepsilon$. At this point, none of the agents can make an offer that will change the payoffs and will be accepted. Therefore A_1 and A_2 will form a coalition; A_1 will receive a payoff of $7+\varepsilon$ and A_2 will receive a payoff of $3-\varepsilon$. Thus, one step of the negotiation is accomplished. If agent A_3 would not have indicated that he can offer up to 7, a possible scenario is that A_2 will offer A_1 $5+\delta$ for some $\delta > \varepsilon$, and A_3 will respond by increasing its offer. This incremental process can proceed until eventually A_2 offers A_1 $7+\varepsilon$. As we explained above, the fact that A_3 indicated that it can offer up to 7 shortened the bargaining process.

The next step of the algorithm will cause the formation of the grand coalition. Since there are only two coalitions at this stage, and their common extra payoff is 5, the only possible offer is to divide D by 2, so that each coalition will receive 2.5. This offer will be accepted, and the coalition of A_1 and A_2 will divide its new additional payoff according to the previous relative strengths as was shown in the last step. The resulting payoff vector will be: $U_1 = 7 + 0.7 \times 2.5 + \varepsilon \approx 8.75$, $U_2 = 3 + 0.3 \times 2.5 - \varepsilon \approx 3.75$, $U_3 = 2.5$. Note that the final payoffs of the agents reflect their relative strength.

Discussion of the negotiation algorithm The outcome that a coalition gains as a result of the negotiation algorithm reflects its relative contribution to possible coalitions and its computational capabilities. Coalitions which contribute more relative to others will be preferred by the other coalitions and will receive a larger share of the common extra payoff.

As written above, when a pair of "powerful" coalitions is created, the coalition with the higher computational capabilities will be the first to calculate an offer and send it to its partner. The situation of being the first to make an offer in the negotiation process may give coalitions an advantage when joining with computationally weaker coalitions. This means that the algorithm may be advantageous to agents that have better computational capabilities.

There may be occasions when a coalition A obtains its highest common extra payoff with more than one other coalition. If A's highest common extra payoff with two different coalitions B_1 and B_2 is identical, then A should choose one of them to be its partner in a "powerful" pair. Being chosen by A may give the chosen coalition an advantage over the other. The ability to choose from among B_1 and B_2 the one that will be included in a pair provides A with some advantage in the negotiation. For example, A can demand to be designated as C_p^1, i.e., to be the first to make an offer when negotiating the formation of a joint coalition. One possible criteria that A may use for choosing among B_1, B_2 is their computational power. The algorithm requires that after a coalition has chosen its partner in a pair, it cannot depart. This requirement leads to stability of the negotiation process.

Being chosen as a "powerful" coalition according to the algorithm does not necessarily give an advantage to the chosen coalition. This is because a powerful coalition C_p has to approach a weaker coalition C_w. The proposal that C_w returns to C_p may be the minimum necessary for cooperation with C_p, although C_p can decide to avoid cooperation with C_w and cooperate with another weak coalition or with its associate, the other powerful coalition. The main factor that plays a role in the coalition's eventual outcome is its relative contribution to possible coalitions. Whether being "powerful" is advantageous for the powerful coalition depends on the details of the specific situation, and cannot be predicted without complex computation.

Although powerful coalitions do not necessarily have an advantage over other coalitions, in the negotiation algorithm we do prefer that these coalitions, which have a larger common extra payoff, will begin the negotiation process. This is because coalitions with larger common extra payoffs will have more flexibility when negotiating coalition formation. Therefore, it is more likely that such coalitions will form joint coalitions faster and with less computations than other coalitions. Thus, the overall costs will decrease, an advantage that the designers of agents should seek.

3.2 Shapley value

A well-known solution to the problem stated above was suggested by Shapley[7] [18]. Given a group of agents in a super-additive environment, and assuming that a grand coalition had formed, Shapley suggests a function φ that attaches to each agent A_i in the grand coalition a unique value v_i, which is its share of the joint payoff. φ is a general payoff function, which is defined for all of the agents. Shapley suggested that φ should satisfy the following axioms:

Axiom 3.1 Symmetry
For every pair of agents $A_i, A_j \in N$, for all the coalitions such that $A_i, A_j \notin N_C$, if $V(N_C \cup \{A_i\}) = V(N_C \cup \{A_j\})$ then $v_i = v_j$, and we say that A_i, A_j are exchangeable.

Axiom 3.2 Null agent
For every agent $A_i \in N$, for all the coalitions such that $A_i \notin N_C$, if $V(N_C \cup \{A_i\}) = V(N_C)$ then $v_i = 0$, and we say that A_i is null.

Axiom 3.3 Efficiency
For all the agents $A_i \in N$, $V(N) = \sum_i v_i$.

Axiom 3.4 Additivity
For every pair of coalitions $C_i, C_j \in \mathcal{C}$, such that $N_C^i = N_C^j = N_C$, if Q_C^i, Q_C^j and $\overline{q}^i, \overline{q}^j$ give these coalitions the values V^i, V^j, then the value of the coalition $C_{ij} = \langle N_C, Q^i + Q^j, \overline{q}^{ij}, U_C^{ij} \rangle$ is $V^{ij} = V^i + V^j$.

Shapley proves [18] that the four axioms above are enough for determining φ uniquely, and also gives an explicit formula for calculating φ.

We shall denote by $\mathcal{R}$ a permutation of the members of N. It is obvious that there are $n!$ different permutations. We shall denote by $C_i^{\mathcal{R}}$ the group of agents that were included in $\mathcal{R}$ before agent A_i. Thus, the expression $V(C_i^{\mathcal{R}} \cup \{A_i\}) - V(C_i^{\mathcal{R}})$ is the marginal payoff that agent A_i brings to the coalition $C_i^{\mathcal{R}}$. If we sum agent A_i's marginal payoff over all possible $\mathcal{R}$ and calculate the mean by division by $n!$, then we obtain Shapley's formula:

$$\varphi(A_i) = \frac{1}{n!} \sum_{\mathcal{R}} \left[V(C_i^{\mathcal{R}} \cup \{A_i\}) - V(C_i^{\mathcal{R}}) \right]$$

We suggest a method for using Shapley's formula:
Assuming honesty, we can randomly choose an agent A_r that will be responsible for calculating the Shapley values (A_r may be paid for its efforts). A_r may also be elected by some voting procedure [15, 16]. To be able to calculate the Shapley values, A_r must first contact all other agents and ask for all of the relevant information. Therefore it contacts $n-1$ agents, which respond by transmitting their payoff functions and their resource vectors. After A_r has received all of the

[7] For more discussion on Shapley see [13, 6, 5].

information, it computes the Shapley values; a process which requires the computation of all 2^n possible coalitions of the agents. For each possible coalition, A_r must calculate the corresponding payoff. This calculation is a maximization of a function of several variables, which is rather complicated. When the computation is complete, A_r once again contacts all of the agents and informs them of the results. That is, A_r directs all agents how to divide the resources, tasks and extra payoff among them.

Complexity of the Shapley algorithm The Shapley algorithm requires $o(n)$ communication operations to be performed in the first stage. The number of computations for the calculation of Shapley values is dictated by 2^n, and hence the computational complexity is $o(2^n)$. After calculating these values, A_r must contact all of the agents; this requires $o(n)$ communication operations and the same order of computations. Altogether, the complexity of the Shapley algorithm is $o(n)$ communication operations, and $o(2^n)$ computations. A new formula for the Shapley value that offers a reduction in computation time was presented by [5], but it remains of order $o(2^n)$.

Example Recall of the last example of the three agents in a super-additive environment. Using Shapley's formula to calculate the payoffs of the agents in the grand coalition yields the following results: $\varphi(A_1) = 7.167$, $\varphi(A_2) = 4.667$, $\varphi(A_3) = 3.167$. These results are different from the results of the negotiation algorithm example, but the difference is small with respect to the payoffs. We can indicate that in this specific case, the negotiation algorithm strengthens the stronger and weakens the weaker, while the Shapley formula tends to impose "fairness". However, in other examples the negotiation algorithm may strengthen the weaker and weaken the stronger.

3.3 Discussion

Previously we presented two algorithms for payoff distribution among the members of a grand coalition in a super-additive environment. If communication is expensive compared to computation and all designers agree that a "fair" payoff distribution function should satisfy axioms 1 – 4, then it is worthwhile for all designers to agree, in advance, to use the Shapley algorithm. We derive the first condition (communication is expensive compared to computation) by comparing the complexities of the two algorithms. It is obvious that the Shapley algorithm is better only when computation is cheap in comparison with communication.

From the above, it follows that designers have some information about when to choose the Shapley algorithm. We must now compare it to the negotiation algorithm. Both the Shapley algorithm and the negotiation algorithm start with transmission of information; at this stage, the negotiation algorithm requires $o(n^2)$ communication operations, whereas the Shapley algorithm requires only $o(n)$ communication operations. Next, both algorithms require calculations of common payoffs of coalitions; each of these calculations is a maximization over many variables, which is quite a complex calculation. In the Shapley algorithm, the common payoff calculation must be done for all 2^n possible coalitions, and all of these calculations are performed by one single agent in sequence. In the negotiation algorithm, common payoff calculations should be done only for coalitions that form during the negotiation process; that is, only $o(n^3)$ such calculations. Moreover, these calculations are distributed among the agents. This distribution is "natural" in the sense that it is an outcome of the algorithm characteristics, i.e., each agent performs only those calculations that are required for its own actions during the process. In contrast, the Shapley algorithm does not enable distribution of the calculations, mainly because of the ultimate need for agreement upon the redistribution of resources and payoff; if calculations are distributed, agreement requires negotiation.

An important advantage of the negotiation process is the ability to suspend it before it ends (i.e., anytime algorithm). If this algorithm stops before completion, then there is already a coalitional configuration which is for all agents at least equal to, and probably better than, their initial condition. It is obvious that such termination of the Shapley algorithm will bring the agents to the starting point and no coalition will form.

The negotiation algorithm can be performed without real negotiations, if we let one agent simulate the negotiation process and make all of the calculations. In this case the number of calculations will not change, but they will no longer be distributed. The number of communication operations will decrease to $o(n)$, exactly as in the Shapley algorithm.

The Shapley solution takes all possible negotiation scenarios into consideration while the negotiation algorithm follows only one scenario. Hence, we expect that the resulting payoff distributions of the two algorithms will rarely be equal (we have explicitly shown it for $n = 3$). It is hard to predict which algorithm will grant more to a given agent. Therefore, we cannot give an agent a method that will enable it, for any given setting, to determine which of the two methods it should prefer. Hence, we suggest to the designers of agents to take into consideration the costs of both communication and computation, and the significance of computation distribution.

The procedures presented above were developed for super-additive environments where forming bigger coalitions is always beneficial. However, in environments where there are unresolvable conflicts between the agents, forming bigger coalitions may not be beneficial or may even be harmful to some agents. Such environments are non-super-additive environments.

The Shapley value algorithm is designed particularly for super-additive environments and cannot be adjusted to non-super-additive environments. However, the negotiation algorithm can be adjusted, by making small changes, to environments which are not super-additive environments. Note that in several such situations the grand coalition will not form. This is beyond the scope of this paper. Non-super-additive environments are discussed in [19].

4 Conclusion

This paper discusses multi-agent environments where agents are designed to reach goals that were pre-defined by their operators. An important way to execute tasks and to maximize payoff is to share resources and cooperate on task execution by creating coalitions of agents. The paper discusses the advantages of two methods of coalition-formation and payoff distribution in super-additive environments and suggests occasions when each is most suitable. Both algorithms ultimately lead to grand coalition formation and payoff disbursements among coalition members. The algorithms are: 1. the Shapley value algorithm, which employs one representative, is computation-oriented and is best used in instances where communication is expensive. If calculations are halted in the middle, the process will not lead to any coalition formation, and; 2. the negotiation algorithm, which leads to distribution of both calculations and communications, and is primarily communication-oriented. If halted in the middle this algorithm still provides the agents with a set of formed coalitions.

References

1. R. J. Aumann. The core of a cooperative game without side-payments. *Transactions of the American Mathematical Society*, 98:539–552, 1961.
2. R. J. Aumann and B. Peleg. Von Neumann-Morgenstern solutions to cooperative games without side-payments. *Bulletin of the American Mathematical Society*, 66:173–179, 1960.
3. C. Castelfranchi. Social power. In Y. Demazeau and J. P. Muller, editors, *Decentralized A.I.*, pages 49–62. Elsevier Science Publishers, 1990.
4. R. Conte, M. Miceli, and C. Castelfranchi. Limits and levels of cooperation: Disentangling various types of prosocial interaction. In Y. Demazeau and J. P. Muller, editors, *Decentralized A.I. - 2*, pages 147–157. Elsevier Science Publishers, 1991.
5. G. Gambarelli. A new approach for evaluating the Shapley value. *Optimization*, 21(3):445–452, 1990.
6. S. Guiasu and M. Malitza. *Coalition and Connection in Games*. Pergamon Press, 1980.
7. J. C. Harsanyi. A simplified bargaining model for n-person cooperative game. *International Economic Review*, 4:194–220, 1963.
8. J. C. Harsanyi. *Rational Behavior and Bargaining Equilibrium in Games and Social Situations*. Cambridge University Press, 1977.
9. S. Kraus. Agents contracting tasks in non-collaborative environments. In *Proc. of AAAI93*, pages 243–248, France, 1993.
10. S. Kraus and J. Wilkenfeld. Negotiations over time in a multi agent environment: Preliminary report. In *Proc. of IJCAI-91*, pages 56–61, Australia, 1991.
11. T. Kreifelts and F. Von Martial. A negotiation framework for autonomous agents. In *Proc. of the Second European Workshop on Modeling Autonomous Agents in a Multi Agent World*, pages 169–182, France, 1990.
12. R. D. Luce and H. Raiffa. *Games and Decisions*. John Wiley and Sons, Inc, 1957.
13. M. Maschler and G. Owen. The consistent Shapley value for games without side payments. In R. Selten, editor, *Rational Interaction*, pages 5–12. Springer-Verlag, 1992.

14. J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, N.J.
15. B. Peleg. Consistent voting systems. *Econometrica*, 46:153–161, 1978.
16. B. Peleg. *Game Theoretic Analysis of Voting in Committees.* Cambridge University Press, Cambridge, 1984.
17. A. Rapoport. *N-Person Game Theory.* University of Michigan, 1970.
18. L. S. Shapley. A value for n-person game. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games.* Princeton University Press, 1953.
19. O. Shehory and S. Kraus. Feasible formation of stable coalitions in general environments. Technical Report, Institute for Advanced Computer Studies, University of Maryland, 1994.
20. K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.
21. M. Wellman and J. Doyle. Modular utility representation for decision-theoretic planning. In *Proc. of AI planning Systems*, pages 236—242, Maryland, 1992.
22. Eric Werner. Toward a theory of communication and cooperation for multiagent planning. In *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 129–143, Pacific Grove, California, March 1988.
23. G. Zlotkin and J. Rosenschein. Negotiation and task sharing among autonomous agents in cooperative domain. In *Proc. of the nth International Joint Conference on Artificial Intelligence*, pages 912–917, Detroit, MI, 1989.
24. G. Zlotkin and J. Rosenschein. Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, 21(6):1317–1324, December 1991.

This article was processed using the LaTeX macro package with LLNCS style