

Agents' Coalitions Based on a Dynamic Programming Approach

Baltazar Frankovič, Tung-T. Dang, Ivana Budinská

Institute of Informatics, Slovak Academy of Sciences

Dúbravská c. 9, 84237 Bratislava, Slovakia

E-mails: utrrfran@savba.sk, utrrbudi@savba.sk

Abstract: Cooperation among agents is a very actual problem, which is a focus of lots papers published recently. The main purpose of cooperation is to improve each agent's execution or whole execution of agents' society and it depends strongly on concrete kinds of agents and a situation in which the agents are situated. Moreover, agents' intelligence and their relationship also influence lots to cooperation process, mainly, when the agents' society is huge or composed from a variety different agents. In this paper we focus to a problem how to create optimal coalitions among the given agents, where a coalition is a one of possible forms for cooperation and in which the common goal has highest priority for all the agents creating this coalition. Further, we will introduce methods, which are able to guarantee sub-optimal solutions by reducing a search space and estimate the domain, where the optimal solution possibly could be. In the end we will discuss about a problem of creating coalition with more parameters.

Keywords: Agents, Cooperation, Coalition, Optimization and Multi-Agent Systems

1 Introduction

In a lot of recent papers an agent has been introduced. The agent may be software, hardware, robot or another, but one of the relevant differences between the agent approach and the well-known traditional ones is an agent's ability to migrate from one place to another, to exchange its experiences with other ones in purpose to improve knowledge, skill, etc. The agent does not have to work only alone, in that case, the final result depends only on agent's capabilities and given tasks, but moreover it can cooperate with another ones if it is able. Cooperation may be executed in a variety forms, depends on which kinds of the agent are and which capabilities each agent can possess. If the agents are radically different, then, their cooperation may be very complicate. From that reason it is necessary to restrict and clearly define exact areas where the agents can cooperate, main objects of cooperation, methods and a common protocol for agents' communication. These

main features, however, are very important for formulation of agents' cooperation, but unfortunately, in general we cannot specify them exactly in advance when we do not know which kinds of the agent will be implemented to and which functions each agent will have. Another essential property inside of agents' cooperation is behavior of each agent. Agent's behavior may be simply predictable – it means we can describe, approximate or explain it by exact mathematical formulations, or completely stochastic – belong to this category such agents as for example: a dealer in stock exchange, a manager of a factory, etc. Cooperation between various kinds of agents is also complicated problem, if is considered that the agents' characteristics are very pragmatic. Every kind of the agent may have different capabilities of reasoning, inferring and evaluating, moreover, via combination with variety of knowledge or experiences' basis, then, their behavior perhaps becomes unimaginable and the final result of their cooperation may grow to immeasurable dimensions [12], [13]. Another property of agent's behavior, which possible influences a lot to cooperation process with another ones, is an agent's capability to improve its knowledge's basis during cooperation. They are known several kinds of such agents, for example: an adaptive agent, which is able to analyze its environment (including behavior of another agents working in the same environment) and act (or react) so as to achieve the best results for itself. Next kind of such agent is a learning agent, which, in a difference of the previous kind, sequentially learns from behaviors of another agents and improves its knowledge's basis to enable to meet its goals. A quality of a learning process can be reflected via actions that the agent takes in cooperation process and their results the agent receives for them. A learning process may last as long as the received results satisfy or come near agent's goals; even though, the process of learning is strongly dependent on duration of agents' wiliness to join cooperation with another ones and intelligence of each agent [4], [7], [8]. The creation of coalition and coalition structure of agents is one of possible technical way for formulation agents' cooperation. A coalition of agents essentially may be considered in the framework of multi-agent-system (MAS) as a group of such agents, which are willing to cooperate with all another members in this group and their common cooperative activities aim to reach the optimum of the given criteria, but on other hand, methods for cooperation are collected by these members via negotiation among themselves and they depend on concrete situations. The optimum, which the members in coalition try to reach, does not have to be always a global optimal solution, but in most cases only is optimal from any point of view (criterion) or a Nash optimal solution (Nash optimal solution is defined in [2], [9]). From this reason the coalition formation includes the activities as follows:

- Generation of coalition structure: to create or distribute the agents to individual sets for special problem solution.
- Solving the optimization problem in each coalition and a coalition structure, too.
- Distribution of the obtained requirements to each coalition.

There exists a lot of possible ways to search for and create an optimal coalition – mainly via negotiation, or sequentially search all possible variants - and several of potential methods that we shall use for finding the solution of such formulated problem are as for example: an application of automat theory, game theory, genetic algorithms or another. The application field of this approach may be a variety and very wide, it is applicable in resolving planning problems in manufacturing [9], [10]. Another applicable field is in marketing for enterprises, where to maximize own profit each manager (considered as one agent) can try to cooperate with another ones [6], [11]. Coalition in this case may have another effect to eliminate execution of agents not belonging to the same coalition. Finding a solution for the above-introduced problem has been the subject of many publications, which tried to obtain a feasible way to resolve a general problem of creating coalition [1], [2], [3]. As a final note, a cooperation problem in general has many open questions to research; one of them is how to create the optimal structure so as the agents cooperate together with the maximal final effects. The focus of this paper is to find an optimal coalition structure of agents, where each agent and coalition has an own value expressed by its criterion function. The optimal coalition structure must guarantee that the solution is within permitted bounds for each agent and moreover the performance of whole system (involving all agents' performance) will be optimal. In this paper are presented some algorithms (computation methods), which are able to reduce a searching process in comparison with traditional approaches until the final solution will be found.

2 General Formulations for Agents' Coalition

Initially, let us introduce some general notations that we will use in the rest of this paper. Afterwards we will show some characteristics related with agents' coalitions and some basic approach, which are able to find the global optimal solution.

2.1 General Notifications

Let $\{A_1, \dots, A_n\}$ be a set of n agents, and I denote a set of index, $I = \{1, \dots, n\}$. In some parts of this paper possible occurs a remark $i \in I$, it means the agent A_i from this set. For simplicities we shall use a note I and a mark $K \subseteq I$ denotes a subset created by the agents, whose index belongs to a set K . Let denote $\mathcal{R}_+^s$ as an s -dimensional set of nonnegative real values, which is used to assess an agent's execution. Each option in this set represents one parameter of agent's execution, e.g. cost, payoff, effectiveness, etc.

Definition 1: An *agent's value* is a function mapping from a set of all possible results that the agent can achieve (independently or via cooperation) to $\mathfrak{R}_+^s$.

Further notations are: for $\forall i \in I, K \subseteq I$,

- q_i^* denotes a value if the agent A_i works independently,
- q_i^K denotes a value if the agent A_i joins coalition K ,
- f_i denotes an expected value of the agent A_i .

$f_i = q_i^*$ if the agent A_i works independently, or

$= q_i^K$ if the agent A_i joins the coalition K .

- F_K denotes a value of a set of agents $K \subseteq I$ and is defined as:

$$F_K = \sum_{i \in K} c_i * f_i.$$

where $\forall i \in K$, c_i is a parameter, which is used to express the priority of the agent A_i in this society (since some agents might be more important than other ones depending on concrete applications, therefore these parameters might be various) and $\sum_{i=1}^k c_i = \text{const}$. Generally, all the agents have the same priority;

therefore, the function F_K can be defined as:

$$F_K = \sum_{i \in K} f_i \tag{1}$$

In this case $\forall i \in [1, n]$, $c_i = 1$.

In practice, each agent can know what it will get after joining any coalition until the discussion with other ones (about what every member within this coalition has to do regarding their possibilities, aims and willingness, and consequently which results each member can achieve) within this coalition finishes. As a result, in order to know all the values q_i^K the agents need another phase to negotiate with other ones about each possible coalition. It is clear that this process requires very exhausted work, and that is not just a focus of this paper, therefore to simplify let assume that these values are defined in advance or taken from historic database. Certainly, the more these values are defined, the larger is a probability to achieve the optimal solution. If the agents' society is too large, and that results an agent

cannot explore every possible coalition, in such case all the values $q_i^K \mid_{i \in K}$, where the coalition K has not been explored, are defined to be zero. In the next part we will show some important definitions and characteristics of the agents' coalition.

2.2 General Definitions and Properties of Agents' Coalitions

Initially, let consider a following important property of the optimal coalition: for each set $K \subseteq I$ is valid that the set K is an optimal coalition if every experiment trying to separate the set K to a set of smaller coalitions will decrease a value F_K (defined by an equation (1)). This property is derived from the assumption, in which the agents are always willing to remain in the large coalition, if they know that breaking this one to smaller sets will bring the whole party worse results.

Definition 2: A set $K \subseteq I$ is an optimal coalition if

$$\sum_{i \in K} f_i \geq \sum_{K_j} \sum_{i \in K_j} f_i, \quad (2)$$

Where is valid $K = \bigcup_j K_j$.

Every agent, in fact, is an optimal coalition itself, because it is impossible to divide it to smaller subsets. From definition 1 the following theorem is deduced.

Theorem 1: Let be any structure of a set I for which the value F_I achieves maximum, then, such structure must be only composed from the optimal coalitions.

Proof: Let a set I be divided to m subsets like coalitions.

$$I = \bigcup_{j=1}^m K_j, \text{ and } \forall i \neq j \in [1, m] \text{ is valid } K_i \cap K_j = \{\emptyset\}.$$

If any subset $K_{i \in [1, m]}$ is not an optimal coalition, then, from the definition 1 it results: here must exist such a structure for organization a set K_i , $K_i = \bigcup_r K_r^i$, in which a value F_{K_i} is bigger than an actual value. Afterwards a structure composed

as $I = \{ \bigcup_{j=1, j \neq i}^m K_j \} \cup K_r^i$ will have a bigger value than the actual one.

Confrontation. □

Further, we will introduce some notations that are useful for later applications.

Definition 3: Let S_i denote a set of coalitions, which can bring the agent A_i at least the same or better results than when it works independently.

$$S_i = \{K \subseteq I \mid q_i^K \geq q_i^*\} \quad (3)$$

In order to understand it let's take a simple example:

Example 1: Let be three agents $\{A_1, A_2, A_3\}$ and the values that each agent might get by joining coalitions or working alone be given in the Table 1.

On a basis of the values shown in this table and according to the definition 3 the agent A_1 will prefer to choose coalitions (except itself): $\{A_1, A_2\}$ and $\{A_1, A_2, A_3\}$, since in these ones it could gain more than when it works alone ($f_1=6$ and 8 in comparison with $f_1=3$ when it works alone), then the set S_1 will be $\{\{1\}, \{1,2\}, \{1,2,3\}\}$. Similarly it might be possible to calculate that $S_2 = \{\{2\}, \{1,2,3\}\}$ and $S_3 = \{\{3\}, \{2,3\}, \{1,2,3\}\}$.

Table 1
Expected values - example with three agents

	f_1	f_2	f_3
$\{A_1\}, \{A_2\}, \{A_3\}$	3	8	9
$\{A_1, A_2\}, \{A_3\}$	6	7	9
$\{A_1, A_3\}, \{A_2\}$	2	8	5
$\{A_1\}, \{A_2, A_3\}$	3	7	12
$\{A_1, A_2, A_3\}$	8	12	9

Definition 4: Let S_K denote a set of coalitions, which can bring at least the same or better results for all agents belonging to a set K than when they work independently.

$$S_K = \{K_0 \subseteq I \mid \forall i \in K, q_i^{K_0} \geq q_i^*\} \quad (4)$$

Example 2: In the previous example, if a set $K = \{1,2\}$, then, $S_{\{1,2\}} = \{1,2,3\}$, etc.

From these definitions the following results are derived:

Lemma 1:

1. If $K_1 \subset K_2$, then, $S_{K_1} \supset S_{K_2}$.
2. $S_{K_1 \cup K_2} \subset S_{K_1} \cap S_{K_2}$.
3. $S_{K_1} \cap S_{K_2} \subset S_{K_1 \cap K_2}$.

$$K \in \bigcap_{i \in K} S_i \quad (5)$$

Proof: (see in [1])

Consequences 1:

1. $S_{K_1 \cup K_2} \subset S_{K_1 \cap K_2}$
2. $S_{K_1 \cup \dots \cup K_m} \subset S_{K_1} \cap \dots \cap S_{K_m}$,

where $K_1, \dots, K_m \subseteq I$.

Proof:

1. The consequence 1.1 is followed directly from the lemmas 1.2 and 1.3.
2. The consequence 1.2 could be proved by successively applying the lemma 1.2.

$$S_{K_1 \cup \dots \cup K_m} = S_{(K_1 \cup \dots \cup K_{m-1}) \cup K_m} \subset S_{(K_1 \cup \dots \cup K_{m-1})} \cap S_{K_m} \dots \subset S_{K_1} \cap \dots \cap S_{K_m}.$$

Certainly, we want to attempt to implement all possible features of the real agent's behavior into our model, however, a mathematical model cannot copy exactly and completely agent's behavior, but one property has to be considered and it is also very often happened in practice. It is that, every agent is willing to join coalition with another ones if and only if this coalition brings it at least the same or better results than when it works independently. Furthermore, a coalition might exist if all the agents creating this one are willing to join.

Assumption 1: the agent A_i joins coalition $K \Leftrightarrow (q_i^K \geq q_i^*)$.

Assumption 2: a set $K \subseteq I$ is an *acceptable coalition* $\Leftrightarrow \forall i \in K : (q_i^K \geq q_i^*)$

For simplicities we will assume that these properties will be valid in the whole rest of this paper. With these assumptions it is possible to deduce a following theorem.

Theorem 2: A set $K \subseteq I$ is an *acceptable coalition* (does not have to be optimal) if and only if:

Proof:

1. If K is an *acceptable coalition*, it means $\forall i \in K$ it is valid: $q_i^K \geq q_i^*$. Afterwards, $K \in S_i$ for $\forall i \in K$. A result from this is $K \in \{ \bigcap_{i \in K} S_i \}$.
2. If $K \in \bigcap_{i \in K} S_i$, it means that the coalition K can bring at least the same or better results for all the agents belonging to this coalition $\Rightarrow$ a set K can be an *acceptable coalition* (does not have to be optimal). $\square$

Consequence 2: If $\bigcap_{\alpha_i \in I, i=1, \dots, m} S_{\alpha_i} = \{\emptyset\}$, then these agents $A_{\alpha_1}, \dots, A_{\alpha_m}$ cannot be in the same acceptable coalition. $\square$

The proof is derived directly from the theorem 2.

Example 3: In the previous example, since $S_1 \cap S_2 \cap S_3 = \{1,2,3\}$, therefore, the set $\{1,2,3\}$ is a unique possible coalition that all the agents are willing to join.

The main task in this paper that we want to resolve is: to find such a structure of a set I to maximize a function F_I defined by (1). Another way to say: we want to find such a structure $I = \bigcup_{i=1}^m K_i$, where $\forall i, j \in [1, m] \ K_i \cap K_j = \{\emptyset\}$ and

$$F_I = \sum_{i=1}^n f_i = \max. \text{ To resolve completely this task and to find a maximal value of}$$

a function F_I it is necessary to search the whole space of all possible structures of a set I . The complex searching of all possible structures of a set I can be executed in a variety ways, but a complexity of those approaches is the same for every method, because every variant have to be examined and for arbitrary n the problem is known to be NP-hard, thus, it is necessary to turn to heuristic methods to search, which are computationally efficient but which might guarantee only sub-optimal solutions from any point of view, for example: *tabu search, branch and bound, genetic algorithms*, etc. Generally, a quality of those methods strongly depends on how long the process has repeated and a quality of an initial choice. On the other hand, each agent presents one independent object with own goals and intelligence, and consequently the agent's execution (or choice) will aim to the best results for the agent; therefore, naturally each agent might poll such coalitions, in which its benefit is the best from its point of view.

Furthermore, next property of such formulated problem that is easy to verify is that the mentioned problem might be resolved by a *recursive approach*, sequentially for cases with 1 to n agents, where each new agent will try to join any formulated coalition, afterwards a result of the criterion function F_I will be compared with the best one so far. This approach can guarantee the complex searching of the whole space and the final solution certainly will be the globally optimal one. But, as we have mentioned above, agents' choices might influence the searching process a lot, because on a basis of these choices lots of coalitions might not be necessary to be examined and searching space could be reduced too much. In the following section we shall introduce an approach based on a principle supporting complex searching and concurrently regarding agents' choices.

3 A General Dynamic Programming Scheme

For a set $K \subseteq I$ it is assumed: a set K can be decomposed to an amount of independent nonempty subsets. $K = \bigcup_{j=1, \dots, m} K_j$, where, $\forall i, j \in [1, m]$,

$K_i \cap K_j = \{\emptyset\}$. Certainly $m \leq |K|$. Let us define a function Q_K as follows:

$$Q_K = \arg \max_{\{K_j\}} \left\{ \sum_{K_j} \sum_{i \in K_j} f_i \right\} \quad (6)$$

Or by other words, Q_K is a maximal sum of all agents' expected values, which are creating the set K , among all possible structures of the set K .

Furthermore, by comparison an equation (1) and (6) it is possible to derive that

$Q_K = \max \{F_K\}$ for $\forall K \subseteq I$. Therefore, the main task mentioned in the previous section might be transformed to the task to find a value Q_I and such a structure, which achieves this value.

Because each agent has only two alternatives: to work alone or to join any coalition. From this reason we shall present a general dynamic programming method for resolving problem Q_I . We consider n stages, where in each stage one agent is added and after n -th stage the whole problem will be resolved. For $j=1, \dots, n-1$ we denote $I_j = \{1, \dots, j\}$, where every set I_j means we will search an optimal solution (an optimal structure) for j agents $\{A_1, \dots, A_j\}$, which guarantees a maximal value $F_{I_j} (= Q_{I_j})$. Since each set I_j can be obtained by adding the agent A_j to the set I_{j-1} , we might have the following recursive equation:

$$Q_{I_j} = \arg \max_K \{(Q_{I_{j-1}} + q_j^*), (Q_{I_{j-1} \setminus K} + Q_{K \cup j})\} \quad (7)$$

Where $K \subseteq I_{j-1}$ and $j=\{1, \dots, n-1\}$, $I_0 = \{\emptyset\}$.

Further, for each $j=\{1, \dots, n-1\}$, if we denote

$$h_j = \arg \max_{K \subseteq I_{j-1}} \{Q_{I_{j-1} \setminus K} + Q_{K \cup j}\} \quad (8)$$

The equation (7) could be rewritten as:

$$Q_{I_j} = \max \{(Q_{I_{j-1}} + q_j^*), h_j\} \quad (9)$$

On the basis of these recursive equations we can, in principle, find an optimal solution for Q_I by the following backward scheme:

- Sub-problem in stage n : Search a value h_n in (8) by examining all possible coalitions that the agent A_n is willing to join with using theorem 2 and consequence 2. After that, replace them to (9) to obtain a value Q_I .
- Sub-problem in stage $n-1$: Analogously as above: it is necessary to examine those coalitions that the agent A_{n-1} is willing to join and do not involve the agent A_n , etc.
- Sub-problem in stage 1: $Q_{I_1} = q_1^*$.

Now we get:

Theorem 3: By solving sub-problem in n stages we will obtain the optimal structure for the given set I and a maximal value of F_I .

Proof: (outline) Since by adding an agent A_n to a set I_{n-1} we obtain the set I , and from these equations (7), (8) it is easy to verify that calculating of a value Q_I can be made from the known values in the previous stages from the first to $(n-1)$, otherwise, these values have been calculated for optimal cases, in which a structure of each subset was optimal. Consequently, a value Q_I from an equation (9) has to be optimal one.

From an equation (8) it is possible to contend that the main complication in the general method presented above is that the process of calculating a function $h_{j \in [1, \dots, n-1]}$ is huge and complicated, essentially with large values of n . Because, theoretically, the new added agent is able to join every coalition (if there are $j_{j=1, \dots, n-1}$ agents in the previous set, then, the new $j+1$ agent could join one of 2^j possible coalitions), therefore dimensions of this problem will grow with an exponential speed when n is increased.

To improve the previous approach to computable dimensions, the first step that we need to do is to reduce an amount of coalitions that the new added agent can join. To do it, it is necessary to use agents' intelligence and their attributes. Because each agent can own different mechanisms to poll coalitions that it is willing to join. The second step is: applying theorem 2 and consequent 2 to remove coalitions that the new added agent cannot join and leave only the *acceptable coalitions*, which are accepted by all the agents creating these ones. Afterward, an equation (8) could be resolvable. Certainly, the results of the first step will hardly

depend on each agent's choices even though, these choices might be executed by different ways and they also might be dependent on another agents' decisions too. In the next section we will discuss this problem.

4 Creating the Optimal Structure by Sequential Conceding

As we have mentioned above, reducing an amount of potential coalitions that the new added agent could join is a necessary thing, which enables to overcome huge complexities of an equation (8). We assume that, these expected values ($q_{i \in I}^{K \subseteq I}$), which each agent is able to achieve by joining coalition with another ones, are random and known for all the agents creating this coalition. Moreover, any continue function does not exist, which is able to approximate exactly these values (because if it exists, this problem could be solved by different ways as for example: searching global maximum or minimum of any multi-parameters' function). Furthermore, for simplicities we will omit a negotiation part to achieve these values $q_{i \in I}^{K \subseteq I}$ and assume that they are known.

The first step mentioned in the Section 3: to reduce an amount of coalitions that the new added agent could join, might be made in the following principal:

Because for each agent, a coalition, in which its expected value is maximal of all, will have the highest priority in its choice. But, on other hand, with such choice of each agent, they will with a very high probability never reach an optimal structure. Therefore, each agent will have to concede, however, a new question: which agent has to concede and how much is a further complicated problem, because the agent could prefer one coalition before another from its criteria or on the basis of its deliberations it might consider its choices as the best for its execution. In this paper we shall propose three methods, in which the agents are assumed to have the same aim to achieve the global optimal executions of the whole agents' society. Before that, we have to define some important specifications:

Definition 5: for each agent $i \in I$, let $v_i^0 = \max_{K \subseteq I} \{q_i^K\}$ be a maximal expected value that the agent A_i can receive and for $\forall 0 < \alpha \leq v_i^0$ let $\Omega_\alpha^i = \{K \subseteq I \mid q_i^K \geq \alpha\}$ be a set of coalitions in which the expected value for the agent A_i is more or equal than α . After these preparations we can now formulate our algorithms:

First, each agent states its aim equal a maximal value that it can achieve in all coalitions and collects only these coalitions in which its expected value is more or

equal this value. After that, the agents try together to find a solution among these sets, if they cannot find any solution, all the agents sequentially have to concede by decreasing its aim. In an algorithm 1, all the agents sequentially decrease their aim but with the same ratio (ratio between a new aim and present one), in an algorithm 2, a difference between a new aim and the present one is the same for every agent. In an algorithm 3, we consider that in each turn only one agent decreases its aim and it always collects such coalitions, in which its expected value is maximal between remained ones.

An algorithm 1: (based on an equal ratio of conceding)

Step 1:

- a) $\forall i \in I$ Choose coefficient $c=1$ and $\alpha_i = c * v_i^0$.
- b) $\forall i \in I$ Search a set $\Omega_{\alpha_i}^i$.

Step 2:

- a) Using a dynamic programming scheme presented in section 3 to search a sub-optimal solution among sets $\Omega_{\alpha_i}^i, i \in I$.
- b) If a solution is not found, decrease a coefficient c and repeat a step 1.
- c) If any solution is found, this process is stopped.

To effectively resolve a step 1.b, at first each agent sorts out all possible coalitions accordingly a value $q_i^K, K \subseteq I$ (since each agent could join 2^{n-1} possible coalitions, then, this procedure has a complexity $\cong O(2^{n-1} * (n-1) * \log 2)$). Afterwards, in every time when the coefficient c is decreased, each agent can choose immediately from its sorted coalitions feasible ones, which satisfy a condition for creating a set $\Omega_{\alpha}^i, \forall i \in I$. To solve a step 2.a we could apply the approach based on a dynamic programming scheme presented in a Section 3.

Similarly, we shall propose further algorithm based on equal regressions of each agent in a searching process.

An algorithm 2: (based on an equal regression of conceding)

Step 1:

- a) $\forall i \in I$ Choose coefficient $c=0$ and $\alpha_i = v_i^0 - c$.
- b) $\forall i \in I$ Search a set $\Omega_{\alpha_i}^i$.

Step 2:

- a) Search a sub-optimal solution among sets $\Omega_{\alpha_i}^i, i \in I$.

- b) If a solution is not found, increase a coefficient c and repeat a step 1.
- c) If any solution is found, this process is stopped.

Differences between these both algorithms are not very significant; the results obtained by applying them depend on values q_i^K , for example: when values q_i^K are significantly different (a value $\text{var}(q_i^K)_{K \subseteq I}$ is high) for every $i \in I$, then, the first algorithm could achieve better results than the second one, but in another case when these values are nearly similar, the results could be contrary. To improve these algorithms, we shall present the third one, which is modified from the previous methods and differs from them in such point that in each turn only one agent tries to decrease its aim and how much this agent decreases, it depends on its decision. Certainly, if each agent chooses a different way to decide how much it wants to concede, then, this process might be very interesting and immediately occur lots of problems necessary to resolve. For short framework of this paper we should omit these related problems as agent's decision, negotiation or further problems as selecting agent to concede, etc.

In the next method, each agent sequentially will modify its aim in turn and we assume that each will choose immediately the best coalitions from a set of the remained ones. For the next algorithm we define some notations:

Let $\Psi_i = \{\varphi_i^j\}_{j=1}^{\lambda_i}$ be a sorted set of values q_i^K for the agent A_i (from top to down). λ_i is a sum of possible coalitions that the agent A_i could join - $\max(\lambda_i) = 2^{n-1}$. Certainly it is valid: $\forall i \in I, \varphi_i^1 = v_i^0$. Instead of coefficient c we will use only α_i for simplicity.

After preparations we can present the following algorithm:

An algorithm 3: (modified from the algorithm 1 and 2)

Step 1:

- a) $\forall i \in I$ Choose coefficient $\alpha_i = \varphi_i^1 = v_i^0$.
- b) $\forall i \in I$ Search a set $\Omega_{\alpha_i}^i$.

Step 2:

$i = 1$.

- a) Search a sub-optimal solution among sets $\Omega_{\alpha_j}^j$, for all $j \in I$.
- b) If a solution is not found,

1. decrease a coefficient α_i to: $\alpha_i = \max\{\phi_i^k \mid \phi_i^k < \alpha_i\}$,
 2. update a set $\Omega_{\alpha_i}^i$,
 3. repeat a step 2.a,
 4. if $i < n$, then $i = i + 1$, else $i = 1$.
- c) If any solution is found, this process is stopped.

Comparison between these algorithms is implemented in an illustrate example with 11 agents and shown in following graphs and a number of steps of the searching process to achieve the final solution is depicted in a Table 2.

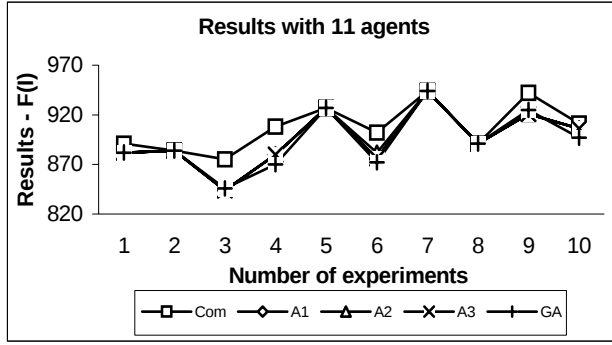


Figure 1

Results of searching $\max(F_I)$ with 11 agents

Notation: in figure 1 and table 2: *com* = complex searching, *GA* = genetic algorithm, *A1* = algorithm 1, *A2* = algorithm 2, *A3* = algorithm 3.

In these experiments we have bounded that each value q_i^K lies in an interval $[0,100]$ and is randomly generated. From the obtained results it is possible to conclude that the algorithm presented above have achieved average very good results (average 95 percent of the optimal results, in some cases they even achieved the optimal results) and for a short time (see Table 2). The difference between these algorithms is not so significant, in some examples one algorithm could achieve better results than another ones, but both algorithms 1 and 2 are more appropriate for parallel working, since after each round each agent could immediately calculate its value α_i and does not have to wait until its turn. On the other hand, for an arbitrary set of values $\{q_i^K\}$, the algorithm 3 generally produced less complicated sets $\Omega_{\alpha_i}^i \mid_{i=1,\dots,n}$ than both first ones, since after each turn only a small number of possible coalitions is added, not massively as in the

algorithms 1 and 2. Moreover, the algorithm 3 could be extended and modified for the case when each agent should choose a different way to calculate its value α_i . In this graph, the results of complex searching always have the maximal values. Another problem that may appear in this graph is that in several points is only one pattern; it means more results have the same value in this point.

In Table 2 is shown how many steps the searching program has repeated until the final result is found. These values may be used also to appoint to a time-complexity of each method to achieve the final results. For a genetic algorithm we fixed that the program will finish after certain number of steps.

Table 2
A case with 11 agents - complexity for searching $\max(F_I)$

Order	Complex searching	GA	A1	A2	A3
1	859868	8500	134	173	238
2	900848	8500	112	113	189
3	1025352	8500	359	190	522
4	839476	8500	76	47	135
5	864455	8500	144	125	228
6	904389	8000	86	77	154
7	926803	8000	57	36	89
8	865845	8000	12	24	20
9	886832	8000	164	103	287
10	1112554	8000	152	168	260

The experiment has been repeated with more agents and from the obtained results we can see that: with a small number of agents (up to 8 agents) the complex searching approach can be applied, because a complexity is acceptable and all another reducing algorithms give sufficiently large differences. For cases with more agents (9 agents and more), all reducing methods have an omisable low complexity in comparison with using the complex searching approach or a genetic algorithm; nevertheless the results of these algorithms achieve just about the best results obtained by searching complex space. Better results we can reach by modifying the step 2.b.1 in the algorithm 3, instead of adding only coalitions, which have a value φ_i^k or q_i^K respectively immediately after a coefficient α_i , we can attach more ones at once. By this way a number of satisfied solutions will increase and a chance to reach the best solution will be bigger. Otherwise, for a case with 11 agents, a number of steps that the searching program repeated are about 1 million and this number will continuously increase with an exponential speed, therefore, from a practical point of view it is not possible to apply the complex searching method but more effectively is to use one of some presented reducing mechanisms, which give also average optimal results.

Conclusions

In this paper we have introduced one of especially actual problems that is how to create an optimal structure of agents' coalitions. Characteristics and conditions related creation coalitions have been pointed and proved. Further, we have introduced some methods, which are addressed to find sub-optimal solutions. Our methods have been formulated on a basis of modifying dynamic programming scheme for searching the global optimal solution and on other hand we have considered also to use agents' intelligence and their properties of independent deliberation to reduce useless coalitions, which are omitted in a searching process. These algorithms that have been presented here, then, were implemented for concrete examples and compared with another heuristic algorithm as genetic algorithm or the worst case when all variants are examined. From some declared problems in a Section 4 we see several features, that we should continue to resolve in future works, that are: a problem to create optimal structure for agents' coalitions with more parameters, and more kinds of agents may be connected in a cooperation process. Another problem is implementation of agents' negotiation abilities to improve cooperation process and approximate closer real situations. The last possible focus in the future is to find an optimal manner for gathering input information, because this problem has very important effect to the solving process and the achieved results respectively.

Acknowledgement

The work presented in the paper was supported by following projects:

- APVV LPP-0231-06 ; VEGA 2/7101/27

References

- [1] Dang, T-Tung, Frankovic, B., Budinska, I.: Agents Coalition in Coordination Process. In *XV IFAC World Congress b'02*, Barcelona, 2002
- [2] Wooldridge, M., Bussmann, S., Klosterberg, M.: Production Sequencing as Negotiation. In *Proceedings of the First International Conference on the Practical Application of Agents and Multi-Agent Technology (PAAM '97)*, 709-726, 1996
- [3] Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohmé, F.: Coalition Structure Generation with Worst Case Guarantees. In *Artificial Intelligence* 111, 209-238, 1999
- [4] Stone, P., Littman, M. L., Singh, S., Kearns, M.: An Adaptive Autonomous Bidding Agent. In *Journal of Artificial Intelligence Research* 15, 189-206, 2001
- [5] Cohen, W., Schapire, R., Singer, Y.: Learning to Order Things. In *Journal of Artificial Intelligence Research* 10, 243-270, 1999

- [6] Cantamessa, M., Villa, A.: Negotiation Models and Production Planning for Virtual Enterprises. In *Proceeding of The IFAC Workshop on Manufacturing, Modeling, Management and Control – MIM 2001, Prague, Czech Republic*, 1-5, 2001
- [7] Lieberman, H., Nardi, B. A., Wright, D. J.: Training Agents to Recognize Text by Example. In *Autonomous Agents and Multi-Agent Systems*, 4, 79-92, 2001
- [8] Gmytrasiewicz, P. J., Durfee, E. H.: Rational Communication in Multi-Agent Environments. In *Autonomous Agents and Multi-Agent Systems*, 4, 233–272, 2001
- [9] Sandholm, T., Sikka, S., Norden, S.: Algorithms for Optimizing Leveled Commitment Contracts. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 535-540, Stockholm, Sweden, 1999
- [10] Odubiyi, J. B., Kocur, D. J., Weinstein, S. M.: A Scalable Agent-based Information Retrieval Engine. In *Proceedings of the First Annual Conference on Autonomous Agents*. California USA, ACM Press / ACM SIGART, 292-299, 1997
- [11] Wellman, M. P.: A Market-oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems. In *Journal of Artificial Intelligence Research* 1, 1-23, 1993
- [12] Maturana, F., Shen, W., Hong, M., Norrie, D. H.: Multi-Agent Architectures for Concurrent Design and Manufacturing. In *Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing*, Banff, Canada, 355-359, 1997
- [13] Frankovič, B., Dang, T. Tung: Cooperating Agents for Planning and Scheduling. In *Proceeding of The IFAC Workshop on Manufacturing, Modeling, Management and Control – MIM 2001, Prague, Czech Republic*, 06-11, 2001