Computing and Informatics, Vol. 22, 2003, 53–83

OPTIMAL CREATION OF AGENT COALITIONS FOR MANUFACTURING AND CONTROL

T.-Tung Dang, Baltazár Frankovič, Ivana Budinská

Institute of Informatics, Slovak Academy of Sciences
Dbravsk cesta 9, 845 07 Bratislava, Slovakia
e-mail: {utrrtung, utrrfran, utrrbudi}@savba.sk

Manuscript received 9 January 2002; revised 28 March 2003
Communicated by Hong Zhu

Abstract. Cooperation among agents has been the object of many recently published papers. Cooperation might be formulated and work in many forms, between different kinds of agents and situations in which they are situated. In addition, it is also influenced a lot by the agent's intelligence, mutual relationships and the willingness to cooperate with other ones. The main focus of this paper is to solve the problem of how to create optimal coalitions of the given agents with the purpose to improve the collective performance. The coalition is a possible form of cooperation in which the common goal has the highest priority for all members included in it. Further, we introduce methods for finding the sub-optimal solutions, which are able to approximate the range of the optimal solutions. Finally we discuss the problem of creating coalition with more parameters.

Keywords: Agents, cooperation, coalition, optimization and multi-agent systems

1 INTRODUCTION

Introduction of agents has already been presented in many recent papers. An agent can work alone; in this case the final results depend only on its capabilities and on the given tasks, but it can also cooperate with others. Cooperation depends mainly on types of agents and their capabilities. If there are too many differences, the cooperation may be very complicated. Agents with different capabilities of reasoning, evaluating and decision-making, in combination with different kinds of knowledge or experience, can create unpredictable situations [12, 13]. Thus, it is necessary

to restrict the area and the objectives of cooperation, to classify methods and languages for communication, etc. The above-mentioned features, however, are very important for formulating agent cooperation; in general, they cannot be specified precisely in advance if agent functionalities are not known. Agent's behavior is the next important point in cooperation. The behavior could be deterministic -meaning it can be described, approximated or explained by mathematical formulations, or stochastic — such as dealers in the stock exchange, managers of companies, etc. Agents could also improve their knowledge and behavior during cooperation. For example, adaptive agents are able to react to changing situations in the environment, learning agents sequentially learn from the behavior of others and improve their knowledge, etc. The quality of the adaptation or learning process is assessed by the chosen actions within the cooperation process and according to the received results. Both the adaptation and learning processes may last as long as the received results satisfy the agent's goals. They depend also on each agent's intelligence and willingness to cooperate [4, 7, 8].

Coalition is one form of cooperation. Agent coalition is defined as a group of agents, which are willing to cooperate with each other in order to achieve the desired goals. The methods for realization are defined by all the members through negotiation. The goal that the coalition tends to achieve might not be the globally optimal solution for all of them, but in many cases only to find the solutions that are optimal for certain criteria or the Nash optimal solutions (the Nash optimal solution is defined in [2, 9]). Formulation of coalition includes the following activities:

- Creating groups of agents for specific problems.
- Distributing the defined requirements to each coalition.
- Solving the optimization problems in each coalition or among coalitions.

There exist many possible ways to create and to search for optimal coalitions, e.g. via negotiation, or using heuristic search algorithms. There are also several well-known methods for making decisions while formulating coalitions such as game theory, Markovian decision process, Fuzzy sets, etc. More details will be presented in the next part.

Agent coalition is applicable in many application fields, essentially in manufacturing for optimizing production; e.g. a flexible manufacturing system (FMS) consists of different types of production lines and each of them also has a number of production devices (PD-hardware agents) controlled by software agents (SW agents). Joining of coalitions can effectively use the capacity of PDs and increase the total profit of production [9, 10]. The next domain where coalitions are often applied is planning of enterprises, business, etc. [6, 11]. The managers (considered as agents) try to cooperate with others to improve the profit. In this case, joining a coalition can bring additional effect and there is a possibility to eliminate the performance of the other competitor agents out of the coalition.

1.1 Our Contribution

The main issue that this paper addresses is to find the optimal configuration for agent coalitions. The optimal configuration will fulfil each agent's requirements and maximizes the performance of the whole system. The first contribution is the introduction of the coalition properties, which are useful in creating coalitions. The next novelty presented in this paper is to solve the above-mentioned problem by approximation. Using approximate values can accelerate the solving process. However, the achieved solutions might not be the best ones; we can specify the range of the optimal solution on the basis of approximated results. The next contribution is a proposition of four algorithms used for automatic negotiation to create coalitions and to find the sub-optimal coalition structure. The fourth contribution is a proposition of methods for solving the multi-parameter coalition problem, which is, to our knowledge, quite new in this field.

The paper is organized as follows: Section 2 discusses the related work. Section 3 introduces the basic formulations related to agent's coalitions and their properties. Section 4 looks at the problem of searching for globally optimal and sub-optimal solutions by approximation. Section 5 presents four methods for creating coalitions and finding the sub-optimal coalition configuration via sequential regression making. Section 6 introduces the multi-criterion coalition problem. Section 7 outlines some open problems for future work.

2 RELATED WORK

The first task that this paper aims to solve is to find the optimal coalition configuration, which maximizes the collective performance. The above problem has been the subject of many papers, whose common goal was to create a group of interested agents for solving specific problems. One of the main tools for solving is the game theory. Game theory provides methods of calculation to define the best coalitions in various types of problems. In particular, its application to multi-agent systems was studied in [3]. The limits of its use are related to two underlying assumptions: (1) the agents are generally considered as perfectly rational and (2) game theory focuses generally on the value of the optimal solution and not on the most efficient method to reach that solution. The next limit of many recent papers is that they have focused mainly on superadditive coalitions [e.g. 3, 15]. Superadditivity means two coalitions or agents can achieve better results by merging into one. However, many applications might be classified as superadditive; this paper does not concentrate on such a special case, but deals with the general coalitions where agents can gain more or less if joining coalitions.

In [3], the optimal coalition configuration could be found by using a hierarchical search, but the difference between the achieved solution and the optimal one could be specified if coalitions are superadditive. For general case it is impossible to assess the achieved solution. In [15] a mechanism for coalition formation that uses cryptography techniques was proposed. An agent sets each coalition by a weight,

which expresses a probability that it joins this coalition. The solving process is realized in a random order that starts with the first agent, and successively it adds one agent at a time to each coalition. However, this mechanism can be applied to only small-sized multi-agent systems because of its combinatorial complexity due to the calculation of all possible coalitions. [1] proposed a method for coalition formation based on merging agents, but the error might be too large.

The first difference from the mentioned work is to reduce the high complexity of the problem solving. This paper tries to solve it with the approximated data and uses the achieved results to predict the range of the really optimal solutions.

The next issue that this paper addresses is to create the optimal coalition configuration through negotiation. The research of many recent papers has focused on negotiation between self-interested agents [2, 11, 16, 17]. The sequential process of coalition formation is similar to the sequential process of trading where agents try to reach a compromise by exchanging proposals and the final agreement usually is Nash equilibrium [17]. [1] uses fuzzy sets for evaluating and choosing coalitions. In [11], negotiation process was described by game theory as min-max problem between two players. [2] focuses on achieving a compromise between agents within a negotiation process, but the agent's private goals have higher priority than the collective performance. Similarly, [16] tries to find the Pareto optimum through negotiation, where each agent can select the coalition into which it prefers to join.

The second difference from the above-mentioned papers is that this paper supposes that each agent is able to know what other ones can gain, when they join any coalitions (e.g. by sharing their utility functions). In addition, agent's choice is deterministic (whether to join or not) and there is the obligation for each agent to concede within negotiation.

The last novelty presented in this paper that all the above-mentioned papers have not dealt with is the multi-parameter's coalition. In the following sections we will discuss the new methods for solving the above-mentioned problems.

3 GENERAL DEFINITIONS AND FORMULATIONS OF THE COALITION

First, let us introduce some general notations that will be used in the rest of this paper. Then we will introduce and prove several important properties related to agent coalitions and the general method for finding the globally optimal solution.

3.1 General Notifications

Let $A = (A_1, \ldots, A_n)$ denote a set of n agents, and I denote a set of their index, $I = 1, \ldots, n$. A remark $i \in I$ means agent A_i from set A; $K \subseteq I$ denotes a subset created by the agents from set A with index belonging to set K. Further notations are: $\forall i \in I, K \subseteq I$.

- The agent's expected utility $f_i|_{i=1,\ldots,n}$ is a function mapping from a set of all possible plans that the agent can apply to $\mathcal{R}_+$. It is used to assess the agent's performance.
- q_i^* denotes a resulting utility for agent A_i if it works alone
- q_i^K denotes a resulting utility for agent A_i if it joins coalition K
- $f_i = q_i^*$ if agent A_i works alone, or
  $f_i = q_i^K$ if agent A_i joins coalition K
- F_K denotes a resulting utility for agent coalition $K \subseteq I$ and it is defined as follows:
$$F_K = \sum_{i \in K} c_i \cdot f_i,$$
where $\forall i \in K$, c_i is a weight setting for agent A_i, and $\sum_{i=1}^{k} c_i = const.$ Generally, all agents have the same priority; therefore function F_K can be rewritten as follows:
$$F_K = \sum_{i \in K} f_i. \tag{1}$$

In this case, $\forall i \in [1, n]$, $c_i = 1$. In the next part several definitions and important properties of agent coalitions will be introduced.

3.2 The Properties of Agent Coalitions

Let us consider the optimal coalition as follows: Set $K \subseteq I$ is an optimal coalition if every experiment trying to separate it into a number of smaller coalitions will decrease the value F_K (defined in Equation 1).

Definition 1. A set $K \subseteq I$ is an optimal coalition if

$$\sum_{i \in K} f_i \geq \sum_{K_j} \sum_{i \in K_j} f_i, \text{ where } K = \bigcup_j K_j. \tag{2}$$

Remark (a special case): we consider that each agent itself is an optimal coalition, because it cannot be divided into smaller subsets.

From Definition 1, the following theorem is derived:

Theorem 1. The configuration that maximizes value F_I must consist of only optimal coalitions.

Proof. Let set I be divided to m disjoint coalitions:

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

If any subset $K_{i \in [1,m]}$ is not an optimal coalition, then from Definition 1 a manner for reorganization of set K_i, $K_i = \bigcup_r K_r^i$ must exist, in which the new value

of F_{K_i} is larger than the current one. Then the configuration composed as $I = \{\bigcup_{j=1,j\neq i}^{m} K_j\} \bigcup_r K_r^i$ will achieve a better resulting utility than the current one. Confrontation. □

Further, let us introduce some notations that will be used in the following sections.

Definition 2. Let S_i denote a set of coalitions, which can bring agent A_i a better or equal resulting utility in comparison with the one that the agent can gain when it works alone.

$$S_i = \{K \subseteq I \mid q_i^K \geq q_i^*\} \tag{3}$$

Definition 3. Let S_K denote a set of coalitions that bring all the agents belonging to set K a better or equal resulting utility in comparison with the one that the agents can gain when each of them works alone.

$$S_K = \{Ko \subseteq I \mid \forall i \in K, q_i^{Ko} \geq q_i^*\}. \tag{4}$$

From these definitions the following results are derived.

Lemma 1.

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

Proof.

1. Let K_0 be a coalition, which brings all members included in set K_2 a better or equal resulting utility in comparison with the one they can gain when each of them works alone ($K_0 \supseteq K_2$). Clearly, this one brings each agent within coalition K_1 the same effects too. Thus, if $K_0 \in S_{K_2} \Rightarrow K_0 \in S_{K_1}$, as a result $S_{K_1} \supseteq S_{K_2}$.
2. Let $K_0 \in S_{K_1 \cup K_2}$; then K_0 is a coalition that brings all the agents belonging to sets K_1 and K_2 a better or equal resulting utility than when they work alone. Thus, $K_0 \in S_{K_1}$ and $K_0 \in S_{K_2}$. As a consequence $K_0 \in S_{K_1} \cap S_{K_2}$. The lemma is proved.
3. If $K_0 \in S_{K_1} \cap S_{K_2}$, then K_0 brings all the agents belonging to sets K_1 and K_2 a better or equal resulting utility than when they work alone; consequently it brings all the agents belonging to set $K_1 \cap K_2$ the same effects too. Then, $K_0 \in S_{K_1 \cap K_2}$ and $S_{K_1} \cap S_{K_2} \subseteq S_{K_1 \cap K_2}$. The lemma is proved.

□

Consequence 1.

1. $S_{K_1 \cup K_2} \subseteq S_{K_1 \cap K_2}$.

2. $S_{K_1 \cup \ldots \cup K_m} \subseteq S_{K_1} \cap \ldots \cap S_{K_m}$, where $K_1, \ldots, K_m \subseteq I$.
3. $S_{K_1} \cap \ldots \cap S_{K_m} \subseteq S_{K_1 \cap \ldots \cap K_m}$, where $K_1, \ldots, K_m \subseteq I$.

Proofs are similar to those presented above.

Consequence 1 is just Lemma 1 extended for a general case. It is used to restrict the area of exploration of $S_{K_1 \cup \ldots \cup K_m}$ when single sets $S_{K_1}, \ldots, S_{K_m}$ are known.

Of course, implementation of all features of the real world behavior into the mathematical model is quite difficult. To simplify we suppose that each agent is willing to join a coalition if and only if it can gain a better or at least equal resulting utility than when it works alone.

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

We consider that this assumption is valid in the rest of this paper.

Definition 4. A coalition is *acceptable* if each agent in it can gain a better or at least equal resulting utility than when the agent works alone.

With such an assumption the following theorem can be derived.

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

$$K \in \left\{ \bigcap_{i \in K} S_i \right\}. \tag{5}$$

Proof.

1. If K is an *acceptable* coalition, then $\forall i \in K : q_i^K \geq q_i^*$. Then, $K \in S_i, \forall i \in K$. As a result $K \in \{\bigcap_{i \in K} S_i\}$.
2. If $K \in \bigcap_{i \in K} S_i$, then coalition K brings all agents belonging to this coalition a better or at least equal resulting utility than when they work alone $\Rightarrow$ set K is an *acceptable* coalition (does not have to be optimal).

□

Consequence 2. If $\bigcap_{\alpha_i \in I, i \in 1,\ldots,m} S_{\alpha_i} = \{\emptyset\}$, then these agents $A_{\alpha_1}, \ldots, A_{\alpha_m}$ cannot be in one *acceptable* coalition.

Consequence 2 is used for checking a capability of creating acceptable coalitions between specific agents.

The main task that this paper tends to solve is to find such a configuration of set I that maximizes function F_I defined by (1); in other words, to find a manner dividing set I to subsets $I = \bigcup_{i=1}^{m} K_i$, where $\forall i, j \in [1, m]$ $K_i \cap K_j = \{\emptyset\}$ so $F_I = \sum_{i=1}^{n} f_i$ is maximal. To resolve the stated task and to find the maximal value of F_I it is necessary to explore all possible configurations of set I. Let M_n denote a set of all possible configurations of set I, where $|I| = n$; then it is possible to verify that:

$$|M_n| = \sum_{i=1}^{n} M(n, i),$$

where $M(n,i)$ is a number of variants allowing to decompose set I to i independent subsets. For this value the following recursive equation is valid:

$$M(n,i) = M(n-1,i-1) + i \cdot M(n-1,i). \tag{6}$$

The first part in the right side of (6) is the number of configurations when $\{A_n\}$ is one subset. The second part is the number of configurations when A_n joins one of i subsets. In [2] was proved that $|M_n| \cong O(n^n)$.

In the next section a general method that allows achieving the optimal solution is presented.

3.3 A Method for Finding the Globally Optimal Solution

The method presented in this section is based on the $A*$ search algorithm, which guarantees the globally optimal solution. Due to Theorem 1, searching will focus on only optimal coalitions.

3.3.1 An Algorithm for Finding the Globally Optimal Solution (Sketch)

Phase 1

a. Search for sets $\{S_i\}_{i=1,\ldots,n}$.
b. Search for optimal coalitions according to Definition 1. Let K^{op} denote a set of such coalitions.

Phase 2 Use $A*$ search algorithm to find the optimal configuration(s) that maximizes F_I. $I = \bigcup_{j=1}^{m} K_j$, where $\forall j \in [1,m]$, K_i is an optimal coalition $K_i \in K^{op}$ and $\forall i \neq j \in [1,m]$ $K_i \cap K_j = \{\emptyset\}$.

At first step each agent specifies its set $S_i|_{i=1,\ldots,n}$ (coalitions that it is willing to join). Then they search for *optimal* coalitions, since the optimal solution consists of only such coalitions. To reduce useless solutions, agents can use Consequence 2 for checking the agent capability of being in the same *acceptable* coalition. Phase 2 uses $A*$ search algorithm to explore all possible configurations that consist of optimal coalitions from set K^{op}; therefore the achieved solution will be the globally optimal one.

The main difficulty is to solve Step 1.b. Since checking the optimality of coalitions has recursive character, Step 1.b can be executed according to a hierarchical scheme. In the first level only coalitions of two members are examined (there are $\binom{n}{2}$ such coalitions), in each following level the sum of members in coalitions increases by one. In level i, the number of all possible coalitions consisting of $i+1$ members is $\binom{n}{i+1}$. Let K be coalition of $i+1$ members in level i, $K = \{\alpha_1, \ldots, \alpha_{i+1}\} \subseteq I$. Set K is an optimal coalition if and only if the following condition is satisfied (the proof is shown in [1]):

$$\sum_{j \in K} q_j^K \geq Q_{K_1} + Q_{K_2}, \forall K_1, K_2 \text{ where } K_1 \cup K_2 = K \text{ and } K_1 \cap K_2 = \{\emptyset\}, \tag{7}$$

where function Q_K is defined as follows:

$$\forall K \subseteq I, Q_K = \arg \max_{K_1,\ldots,K_m} \left\{ \sum_{r=1}^{m} \sum_{j \in K_r} q_j^{K_r} \right\}, \tag{8}$$

where $K = \bigcup_{r=1}^{m} K_r$ and $\forall i \neq j \in [1,m]\ K_i \cap K_j = \{\emptyset\}$.

It is clear that $|K_1|$ and $|K_2| < |K| = i+1$; then these values Q_{K_1} and Q_{K_2} could be taken from calculation in the previous levels. On the other hand, for $i+1$ members, there are 2^i methods to decompose them into such form shown in Equation (7). Thus, it is necessary to verify 2^i-times to ensure whether this coalition is *optimal* one or not. Finally, the total number of operations necessary to find all *optimal* coalition will be: $M_{max} = \sum_{i=1}^{n-1} \binom{n}{i+1} \cdot 2^i$.

Values of M_n and M_{max} are manageable for a small value n (see in Table 1, values of M_n for $n \leq 10$). For arbitrary value n this problem is known as a NP-hard problem; therefore it is necessary to turn to heuristic searching methods (e.g. iterative improvement: *genetic algorithm* or iterative search: *branch and bound*, *simulated annealing*, etc.), which are computationally efficient but which might guarantee only sub-optimal solutions. Values q_i^K are fully independent and random; moreover, when some values q_i^K change it is necessary to repeat the search process completely. For that reason it is appropriate to approximate these values and to find sub-optimal solutions with predictable errors. In the next section, a method based on the approximated principle, which has also manageable complexity, is presented.

4 CREATION OF COALITION CONFIGURATIONS BY APPROXIMATION

The method presented in this section is based on the assumption that an average value of each variable $q_i^K|_{i \in [1,n], K \subseteq I}$ can be calculated, although it might be altered at any time. Instead of using the realistic values, agents will use the average values to create coalitions.

4.1 Evaluation of Coalitions by Approximated Values

The basic idea of approximation is due to the fact that all the agents decide independently whether to join coalition or not (according to value q_i^K); they are not influenced by other agent decisions. Practically, the agents can be considered as n independent unknown objects represented by variables q_i^K, $i \in [1,n]$, $K \subseteq I$. Moreover, q_i^K are stochastic variables that could be changed dynamically or randomly. Therefore, it is more effective to work with approximated values than with the realistic ones. Let $q_i^K|_{k=1,\ldots,n}$ be the approximated resulting utility that agent A_i can receive by joining arbitrary coalition with $(k-1)$ other agents, and it is defined as follows:

$$\forall k \in [1,n], \text{ then, } q_i^k = \frac{1}{m_k} \sum_{\forall K \subseteq I, \text{ where } |K|=k} q_i^K, \tag{9}$$

where m_k is the number of all coalitions consisting of k members including agent A_i: $m_k = \binom{n-1}{k-1}$. This variable expresses the average utility that agent A_i can obtain into k-member coalitions.

Let a set I be decomposed to m disjoint coalitions as follows:

$$I = \bigcup_{i=1}^{m} K_i, \text{ and } \forall i \neq j \in [1, m]\ K_i \cap K_j = \{\emptyset\}, |K_i| = k_i \text{ and } \sum_{i=1}^{m} k_i. \tag{10}$$

Then, from (1) it is possible to rewrite:

$$F_I = \sum_{K_i} \sum_{j \in K_i} q_j^{K_i}. \tag{11}$$

Because of the independence between the agents, from Equation (11) the following property for the average value of F_I can be derived:

$$E(F_I) = E\left(\sum_{K_i} \sum_{j \in K_i} q_j^{K_i}\right) = \sum_{K_i} \sum_{j \in K_i} E\left(q_j^{K_i}\right) = \sum_{K_i} \sum_{j \in K_i} q_j^{k_i}. \tag{12}$$

It is also assumed that an agent does not prefer one coalition to other ones; therefore, the variables q_j^K have the same distribution. The variance of F_I could be calculated as follows:

$$Var(F_I) = Var\left(\sum_{K_i} \sum_{j \in K_i} q_j^{K_i}\right) = \sum_{K_i} \sum_{j \in K_i} Var\left(q_j^{K_i}\right). \tag{13}$$

Combination of both the average value and the maximal variance allows predicting the range of the optimal solution (e.g. using approximation method with the maximal credibility).

Now the main goal is now to find such configuration of set I as shown in (10), which maximizes the value $E(F_I)$ defined by Equation (12). Because each agent has only n values q_i^k, $k \in [1, n]$, the search space will be reduced by many times compared to when using realistic values.

4.2 Searching for the Optimal Coalition Configuration with Approximated Values

In this section, a generic method resolving the task mentioned above is presented. Let us consider an arbitrary configuration consisting of m coalitions with $k_1, \ldots, k_m$ members, namely, $n = \sum_{i=1}^{m} k_i$, $k_i, m < n$. Theoretically, the number of such configurations is

$$\Delta = \frac{n!}{k_1! \cdot \ldots \cdot k_m!}, \text{ if } k_1 \neq \ldots \neq k_m, \text{ or} \tag{14}$$

$$\Delta' = \frac{n!}{(r_1 k_1^{\alpha})! \cdot \ldots \cdot (r_\alpha k_\alpha^{\alpha})!}. \tag{15}$$

$$\text{If } \sum_{i=1}^{\alpha} r_i = m \text{ and } k_1 = \ldots = k_{r_1} (= k_1^{\alpha}) \neq k_{r_1+1} = \ldots$$
$$= k_{r_1+r_2} (= k_2^{\alpha}) \neq \ldots = k_m (= k_{\alpha}^{\alpha}). \tag{16}$$

Thus, it is necessary to examine Δ (or Δ') different configurations with the same structure (a number of coalitions and sums of members in each coalition). But in practice, a lot of configurations can be omitted and therefore the realistic number of cases needed to be examined is not as high as it seems. To reduce useless configurations, we introduce the following definition.

Definition 5. Let each agent use the expected values defined by Equation (9) and let $n = \sum_{i=1}^{m} k_i$, where k_i, $m \leq n$ are integers. Then, the configuration $I = \bigcup_{i=1}^{m} K_i$, where each coalition $K_i|_{i=1,\ldots,m}$ consists of k_i members, is *stable* if and only if each attempt to exchange two agents in two different coalitions will decrease or achieve the same $E(F_I)$.

Of course, the final solution that maximizes $E(F_I)$ belongs to these *stable* coalition configurations. Therefore, the focus now is to find the *stable* coalition configuration, when a sum of coalitions and numbers of members included in them are known. Methods for finding *stable* coalition configurations will be presented in the next section.

4.2.1 Greedy Algorithm for Finding Stable Coalition Configurations: Special Case (GAS)

A number of coalitions (m) and numbers of members in each coalition ($k_1, \ldots, k_m$) ($n = \sum_{i=1}^{m} k_i, k_i|_{i=1,\ldots,m}, m \leq n$) are given. Let us consider the first case when k_1, ..., k_m are different. Without loss of generality let us assume that $k_1 > k_2 > \ldots > k_m$ and K_1, ..., K_m are m disjoint coalitions consisting of k_1, ..., k_m members, respectively, which create a *stable* configuration. From Definition 5 we infer that, if both the agents $i \in K_1$ and $j \in K_2$ are exchanged, the value $E(F_I)$ will not increase. Thus:

$$\forall i \in K_1, j \in K_2 : q_i^{k_1} + q_j^{k_2} \geq q_j^{k_1} + q_i^{k_2} \iff q_i^{k_1} - q_i^{k_2} \geq q_j^{k_1} - q_j^{k_2}. \tag{17}$$

Similarly, we can have

$$\forall i \in K_3, j \in K_4 : q_i^{k_3} + q_j^{k_4} \geq q_j^{k_3} + q_i^{k_4} \iff q_i^{k_3} - q_i^{k_4} \geq q_j^{k_3} - q_j^{k_4} \tag{18}$$

etc.

Equation (17) could be explained in words as follows: If all the agents are sorted according to value ($q_i^{k_1} - q_i^{k_2}$), $I \in I$, from the largest to the smallest one, then the agents in coalition K_2 cannot be before the agents belonging to K_1 (an example is shown in Figure 1).

A situation as shown in Figure 2 may happen, where α agents (β from K_1 and $(\alpha - \beta)$ from K_2, respectively) have the same value $q_i^{k_1} - q_i^{k_2}$ ($\alpha > \beta > 0$). In this case,

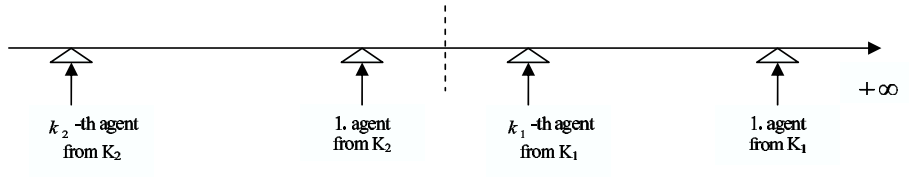


Fig. 1. The order of agents according to $\left(q_i^{k_1} - q_i^{k_2}\right)$

the number of *stable* configurations could be higher by exchanging these α agents from one coalition to another, but value $E(F_I)$ remains unchanged. Therefore, to get value $E(F_I)$ it suffices to examine one of these configurations. A similar conclusion could be made for agents from $K_3, \ldots, K_m$.

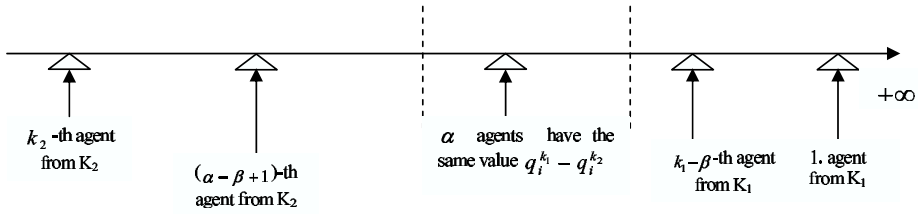


Fig. 2. A special case when β agents from K_1 and $\alpha - \beta$ agents from K_2 have the same value $\left(q_i^{k_1} - q_i^{k_2}\right)$

On the basis of the above explanation we derive the following algorithm for finding stable coalition configurations when coalitions have different dimensions.

Greedy Algorithm for finding stable coalition configurations – Special case (GAS):
Input: m, $k_1, \ldots, k_m$, $k_1 > k_2 > \ldots > k_m$.
Output: stable coalition configurations that maximize $E(F_I)$.

1. $j = 1$.
2. Choose $(k_{2\times j-1} + k_{2\times j})$ arbitrary agents from the unselected ones.
3. Classify the selected agents according to value $(q_i^{k_{2\times j-1}} - q_i^{k_{2\times j}})$, starting with the largest.
4. Distribute $k_{2\times j-1}$ first members in this queue to coalition $K_{2\times j-1}$, the remaining ones to coalition $K_{2\times j}$.
5. Indicate the selected agents $(I = I \backslash (K_{2\times j-1} \cup K_{2\times j}))$, $j = j + 1$ and return to step 2 until $j = \left[\frac{m}{2}\right]$.
6. If $j = \left[\frac{m}{2}\right]$, calculate a value $E(F_I)$ of the obtained configuration and comparing it with the best current one. Return to step 1.

GAS shows a definite way to create coalitions to achieve *stable* configurations (in Step 4). As a result the overall configurations to be examined is reduced many times. The complexity of GAS is expressed by the following theorem.

Theorem 3. Given $\{m,\ k_1,\ \ldots,\ k_m$ are integers, and $k_1 > k_2 > \ldots > k_m$, $\sum_{i=1}^{m} k_i = n\}$. Let be the actual number of configurations examined by GAS. Δ defined from (14) is the theoretical number of all possible configurations. Then

$$\frac{\Delta}{\Delta_{GAS}} \geq \frac{(k_1+k_2)!}{k_1!\,k_2!} \times \frac{(k_3+k_4)!}{k_3!\,k_4!} \times \ldots \times \frac{(k_{2\times[\frac{m}{2}]-1}+k_{2\times[\frac{m}{2}]})!}{k_{2\times[\frac{m}{2}]-1}!\,k_{2\times[\frac{m}{2}]}!}. \tag{19}$$

Proof. (see Appendix 1) □

Theorem 3 points out how many times faster GAS achieves a solution in comparison with the traditional search. Such acceleration allows resolving the abovementioned task during an acceptable time. The next part will deal with the general case when many coalitions have the same dimension.

4.2.2 Greedy Algorithm for Finding Stable Coalition Configurations General Case (GAG)

In general case, many coalitions might have the same size. Let us take the same labels as in the previous part and consider m disjoint coalitions $\{K_1, \ldots, K_m\}$ with corresponding $k_1, \ldots, k_m$ members, which create a *stable* coalition configuration, where $k_1 = \ldots = k_{r_1} \neq k_{r_1+1} = \ldots = k_{r_1+r_2} \neq \ldots = k_{r_1+\ldots+r_\alpha}$, $\sum_{i=1}^{\alpha} r_i = m$. Let denote:

$$k_{r_1} = k_1^\alpha, \ldots, k_{r_1+\ldots+r_\beta} = k_\beta^\alpha, \ldots, k_{r_1+\ldots+r_\alpha} = k_\alpha^\alpha, \text{ and} \tag{20}$$

$$K_1^\alpha = \bigcup_{i=1,\ldots,r_1} K_i, \ldots, K_\alpha^\alpha = \bigcup_{i=r_1+\ldots+r_{\alpha-1}+1,\ldots,r_1+\ldots+r_\alpha} K_i.$$

Similarly to the previous part, it is possible to derive:

$$\forall i \in K_1^\alpha, j \in K_2^\alpha : q_i^{k_1^\alpha} + q_j^{k_2^\alpha} \geq q_j^{k_1^\alpha} + q_i^{k_2^\alpha} \Longleftrightarrow q_i^{k_1^\alpha} - q_i^{k_2^\alpha} \geq q_j^{k_1^\alpha} - q_j^{k_2^\alpha}, \tag{21}$$

etc.

Furthermore, the agents within $K_i^\alpha|_{i=1,\ldots,\alpha}$ can move from one coalition to another, because the agent's approximated utility does not change, so value $E(F_I)$ remains unchanged. These deductions lead to the following algorithm for finding stable coalition configurations in a general case.

Without loss of generality, let assume that $r_1 \times k_1^\alpha \geq r_2 \times k_2^\alpha \geq \ldots \geq r_\alpha \times k_\alpha^\alpha$.

Greedy Algorithm for finding stable coalition configurations – General case (GAG):
Input: m, $k_1, \ldots, k_m$, $(k_1 = \ldots = k_{r_1} \neq k_{r_1+1} = \ldots = k_{r_1+r_2} \neq \ldots = k_{r_1+\ldots+r_\alpha})$.
Output: stable coalition configurations that maximize $E(F_I)$.

1. $j = 1$.
2. Choose $(r_{2\times j-1} \times k^{\alpha}_{2\times j-1} + r_{2\times j} \times k^{\alpha}_{2\times j})$ arbitrary agents from the unselected ones.
3. Classify the selected agents according to value $(q_i^{k^{\alpha}_{2\times j-1}} - q_i^{k^{\alpha}_{2\times j}})$, starting with the largest one.
4. Distribute $r_{2\times j-1} \times k^{\alpha}_{2\times j-1}$ first members in this queue to $r_{2\times j-1}$ coalitions included in $K^{\alpha}_{2\times j-1}$, the remaining ones to $r_{2\times j}$ coalitions those create $K^{\alpha}_{2\times j}$.
5. Indicate the selected agents $(I = I\backslash(K^{\alpha}_{2\times -1} \cup K^{\alpha}_{2\times j}))$, $j = j + 1$ and return to step 2 until $j = [\frac{\alpha}{2}]$.
6. If $j = [\frac{\alpha}{2}]$, calculate value $E(F_I)$ of the obtained configuration and compare it with the best current one. Return to step 1.

The complexity of GAG is expressed by the following theorem.

Theorem 4. Given a general case $(n = \sum_{i=1}^{m} k_i,\ k_1 = \ldots = k_{r_1}(= k^{\alpha}_1) \neq k_{r_1+1} = \ldots = k_{r_1+r_2}(= k^{\alpha}_2) \neq \ldots = k_{r_1+\ldots+r_{\alpha}}(= k^{\alpha}_{\alpha}),\ \sum_{i=1}^{\alpha} r_i = m)$. Let Δ_{GAG} be the actual number of configurations examined by GAG. Δ' defined from (15) is the theoretical number of all possible configurations. Then

$$\frac{\Delta'}{\Delta_{GAG}} \geq \frac{(r_1 \times k^{\alpha}_1 + r_2 \times k^{\alpha}_2)!}{(r_1 \times k^{\alpha}_1)!\,(r_2 \times k^{\alpha}_2)!} \times \ldots \times \frac{(r_{2\times[\frac{\alpha}{2}]-1} \times k^{\alpha}_{2\times[\frac{\alpha}{2}]-1} + r_{2\times[\frac{\alpha}{2}]} \times k^{\alpha}_{2\times[\frac{\alpha}{2}]})!}{(r_{2\times[\frac{\alpha}{2}]-1} \times k^{\alpha}_{2\times[\frac{\alpha}{2}]-1})!\,(r_{2\times[\frac{\alpha}{2}]} \times k^{\alpha}_{2\times[\frac{\alpha}{2}]})!}. \quad (22)$$

Proof. (see Appendix 2) □

Both presented theorems show out the computational advantage of GAS and GAG, which are used to find the *stable* coalition configurations when a number of coalitions and their corresponding sizes are known. In the next section the method for finding the optimal configurations that maximize $E(F_I)$ will be introduced.

4.2.3 An Algorithm for Finding Optimal Coalition Configurations with Approximated Values (AAV)

This algorithm has the following three phases:
An Algorithm for finding optimal coalition configurations with Approximated Values (AAV):
Input: n, $q_i^k|^{k=1,\ldots,n}_{i=1,\ldots,n}$.
Output: The coalition configurations that maximize $E(F_I)$.

Phase 1 Decompose n to smaller integer numbers: $n = \sum_{i=1}^{m} k_i$.

Phase 2 For each variant, use GAS or GAG, search for *stable* coalition configurations.

Phase 3 Choose one of the achieved configurations that maximizes $E(F_I)$ for the optimal solution.

It is clear what the goal of each phase is. However, to ensure that AAV is realizable, the next section will discuss the complexity of this algorithm.

4.3 Complexity and Prediction of Errors

When the number of coalitions and their sizes are known, the total number of configurations necessary to examine can be calculated using Theorems 3 and 4. The complexity of AAV depends also on the number of methods for decomposition of n in Phase 1. Let Sum_n be the number of methods for decomposition n. Value Sum_n can be calculated by the following recursive equations (proof will be given in Appendices 3 and 4):

Let s_i^n be a number of variants for decomposition of n, where each element is larger than or equal to i; then:

$$Sum_n = s_i^n = 1 + s_1^{n-1} + s_2^{n-2} + \ldots + s_{[\frac{n}{2}]}^{n-[\frac{n}{2}]} = 1 + \sum_{i=1}^{[\frac{n}{2}]} s_i^{n-i}, \tag{23}$$

where

$$\forall i > \left[\frac{n}{2}\right] : s_i^n = 1 \text{ and } \forall n > 3 : s_{[\frac{n}{2}]}^n = 2; \text{ and } s_1^2 = 2. \tag{24}$$

$$\forall i < \left[\frac{n}{2}\right] : s_i^n = 1 + \sum_{j=i}^{[\frac{n}{2}]} s_j^{n-j}.$$

An example with 4 agents ($n = 4$):

$$Sum_4 = 1 + s_1^3 + s_2^2 = 2 + 1 + s_1^2 = 5.$$

In this case, overall configurations necessary to be examined to find $\max(E(F_I))$ are 5 (GAS and GAG are used to search in each variant of decomposition).

Table 1 shows the comparison between the number of configurations necessary to be examined in searching for $\max(F_I)$ and $\max(E(F_I))$ by applying Equations (6) and $(19 + 22 + 23 + 24)$ respectively.

Checking whether AAV has a polynomial complexity or not is very difficult. However, the results in Table 1 show that using approximated values requests examining many times fewer instances in comparison with the traditional search for $\max(F_I)$. This fact confirms again the computational advantage of using approximated values in case when the number of agents is large and values $q_i^K | i \in I, K \subseteq I$ change frequently.

The difference between the solutions obtained by using approximated values and the really optimal one can be calculated from value $\max Var(F_I)$ and by using Appendix 5. According to Appendix 5 we can guess the probability that the absolute difference $|\max F_I - \max E(F_I)| \leq \delta \times \sqrt{\max Var(F_I)}$ is larger than or equal to $(1 - \frac{1}{\delta^2})$, where δ is a positive real number used to access the prediction.

Comparison of both Equations (12) and (13) shows that the problem of a search for $\max Var(F_I)$ is similar to the search for $\max(E(F_I))$. Therefore, $\max Var(F_I)$ could be obtained by using AAV too.

The number of agents — n	Sum_n	The number of configurations in searching for $\max(E(F_I))$	M_n — The number of configurations in searching for $\max(F_I)$
4	5	5	15
5	7	7	52
6	11	16	203
7	15	40	877
8	22	97	4140
9	30	272	21147
10	42	688	115975

Table 1. Comparison of complexity in searching for $\max E(F_I)$ and $\max F_I$

Despite the computational efficiency AAV does not consider agents personal requirement during solving. In the next section, we introduce other methods, which allow the agents to express their requirements within the search for solutions.

5 LINEAR REGRESSION ALGORITHMS TO CREATE COALITION CONFIGURATIONS

This section introduces methods for finding sub-optimal configurations based on automatic negotiation and the linear regression principle. In these methods, each agent can select the preferable coalitions to join through negotiation. There are many papers dealing with agent negotiation based on the game theory or other ones and in which agents compete with each other with the purpose to improve own profits. This paper differs from the traditional negotiation approaches in that agents always aim at improving the collective performance. However, due to the practical realization agents might have to be satisfied with sub-optimal solutions, which are achieved by reducing a part of coalitions that are not so interesting for the agents. We propose that agents calculate coalitions that they want to join in order to avoid the complication of classification of reduced coalitions. Only the selected ones are used for constructing the final configurations. We discuss these problems in this section.

5.1 Agent Choices and their Influences to the Coalition Configurations

Due to the fact that sets of coalitions that agents are willing to join $(S_i|_{i=1,\ldots,n})$ might be very large, it is necessary to reduce them to manageable sizes. Each agent can reduce coalitions that are not interesting for it, but in order to guarantee as high quality of the final solutions as possible it should collaborate with other agents before deciding to reduce. Naturally, each agent tends to join coalitions which bring the best resulting utility, but when agent A_i chooses coalition $K \subseteq I$, it has also to consider the decision of $(|K| - 1)$ remaining agents included in this coalition. If one of them refuses joining (e.g. this coalition does not fulfil its requirement),

the coalition will not be successful. In order to achieve a compromise each agent must concede something of its requirement. The familiar work with this paper is Literature [2] is related to this work; there concession mechanisms were used in negotiation process to achieve the Nash equilibrium. In this paper we do not deal with the Nash equilibrium and assume that each agent decides definitely to join or to refuse coalitions — the probability that it joins any coalition is only 1 or 0, respectively.

Let us take a simple example for illustration.

	A_1	A_2	A_3
$\{A_1, A_2, A_3\}$	1	2	**5**
$\{A_1, A_2\}$	2	3	
$\{A_1, A_3\}$	**3**		3
$\{A_2, A_3\}$		**6**	2
$\{A_1\}, \{A_2\}, \{A_3\}$	1	1	3

Table 2. An example with 3 agents and their expected resulting utility

Example 1. Let $I = \{1, 2, 3\}$ be a set of three agents. Values of the resulting utility that each agent can get when it joins any coalition or works alone are shown in Tab. 2.

First, naturally, agents A_1, A_2, A_3 choose coalitions $\{A_1, A_3\}$, $\{A_2, A_3\}$ and $\{A_1, A_2, A_3\}$, respectively. It is easy to see that such a choice cannot be successful, because the agents do not agree with each other.

This simple example demonstrates the necessity of making regressions in order to achieve a compromise. Let us assume that each agent is willing to concede, but the question is: which agent must concede and how much? In Example 1, if agents A_1 and A_3 make a regression to agent A_2 and choose such coalitions that do not have a conflict with the A_2's choice, then possibly they can reach the optimal solution (configuration $(A_1, \{A_2, A_3\})$ is the optimal solution since $F_I = 9$ is maximal). Simply, the A_2's choice depends strongly on the regression of agents A_1 and A_3.

The main task now is to find the optimal coalition configuration that consists only of coalitions supported by all the members included in them and concurrently maximizes F_I. In the next part we present four methods for solving the formulated task based on the linear regression principle.

5.2 Searching for the Sub-Optimal Coalition Configurations On the Basis of the Linear Regression Principle

The methods presented here can be explained as follows: Each agent selects coalitions it prefers to join according to the resulting utility it can gain. The solutions will consist only of these selected coalitions. Because all the agents have the same

priority, all of them have to concede until any solution is achieved. We propose four basic strategies for making regression, which could also combine one with others and create more strategies applicable for various kinds of situations. The first two strategies are: in each turn (1) the agent requirements are decreased by *the same rate* or (2) by *the same value*. These strategies support parallel calculation and simultaneously allow the agents to calculate their favorite coalitions at once. The remaining two strategies are based on the sequential negotiation scheme: in each turn (3) the agent (or agents) with the highest requirement has to concede; or (4) sequentially each agent (or several ones) makes regression. Before presenting these algorithms, let us consider the following definition:

Definition 6. $\forall i \in I$, let $v_i^0 = \max_{K \subseteq I}\{q_i^K\}$ be the maximal resulting utility that agent A_i can gain and $\forall 0 < \beta \leq v_i^0$ let $\Omega_\beta^i = \{K \subseteq I \mid q_i^K \geq \beta\}$ be a set of coalitions in which agent A_i can gain more or at least β.

The main task is re-formulated as follows: $\forall i \in I$, let β_i be the minimal requirement of A_i. The goal is to search for such coalition configuration $I = \bigcup_{i=1}^m K_i$ which satisfies the following conditions:

- $\forall i \in [1, m]$ and $\forall j \in K_i : K_i \in \Omega_{\beta_j}^j$,
- F_I is maximal.

The first condition guarantees that all the agents will support the final solution. The second condition ensures that the achieved solution is the best one among all candidates satisfying the first condition.

5.2.1 Automatic Negotiation Algorithms Based on Parallel Calculation

First, we propose two algorithms to automatically create coalitions based on parallel calculation.

Both proposed SRC/SR algorithms support parallel processing, so the search process can be accelerated by agents calculating sets $\Omega_{\beta_i}^i$ simultaneously. Besides, before starting Phase 2, agents can use Theorem 2 and Lemma 1 (in Section 3) to remove useless coalitions, which do not satisfy at least one agent's requirement. To effectively resolve Step 1.b, each agent sorts out coalitions according to value $q_i^K|_{i \in I}^{K \subseteq I}$. Since each agent can join maximally 2^{n-1} coalitions, the classification process will have a complexity $\cong O(2^{n-1} \times (n-1) \times \log 2)$ by using the *quick sort* algorithm. At any time when coefficient λ changes, each agent can immediately choose coalitions that satisfy the new requirements.

Example 2. Let us continue with Example 1. From values in Table 2 we can have: $v_1^0 = 3$, $v_2^0 = 6$, $v_3^0 = 5$. Let us assume agents use SRC algorithm with $\lambda = 0.5$; then the following sets are obtained: $\Omega_{1.5}^1 = \{\{A_1, A_2\}, \{A_1, A_3\}\}$, $\Omega_3^2 = \{\{A_1, A_2\}, \{A_2, A_3\}\}$, $\Omega_{2.5}^3 = \{\{A_1, A_2, A_3\}, \{A_1, A_3\}, \{A_3\}\}$. Before starting Phase 2, coalition $\{A_2, A_3\}$ could be omitted from set Ω_3^2, because it does not appear in set $\Omega_{2.5}^3$; i.e., agent A_3 does not support this coalition $\{A_1, A_2, A_3\}$ (similarly to coalition from set $\Omega_{2.5}^3$, which could also be omitted). The remaining sets

the Same Rate of Conceding — SRC	*the Same Regression — SR*
Phase 1	**Phase 1**
a. Choose coefficient $\lambda = 1$. b. $\forall i \in I$ set $\beta_i = \lambda v_i^0$ and find set $\Omega^i_{\beta_i}$.	a. Choose coefficient $\lambda = 0$. b. $\forall i \in I$ set $\beta_i = v_i^0 - \lambda$ and find set $\Omega^i_{\beta_i}$.
Phase 2	**Phase 2**
a. Search for solutions from sets $\Omega^i_{\beta_i}$, $i \in I$. If any solution is found, then store it for comparison. b. Decrease coefficient λ and return to step 1.b, if time does not expire. Stop otherwise.	a. Search for solutions from sets $\Omega^i_{\beta_i}$, $i \in I$. If any solution is found, then store it for comparison. b. Increase coefficient λ and return to step 1.b, if time does not expire. Stop otherwise.

Fig. 3. An algorithm for creating optimal coalition configurations based on the Same Rate of Conceding (SRC) and the Same Regression (SR)

for solving Phase 2 are: $\Omega^1_{1.5} = \{\{A_1, A_2\}, \{A_1, A_3\}\}$, $\Omega^2_3 = \{\{A_1, A_2\}\}$, $\Omega^3_{2.5} = \{\{A_1, A_3\}, \{A_3\}\}$. After examining all possible configurations consisting of the remaining coalitions the final solution is $(\{A_1, A_2\}, \{A_3\})$. This is the best configuration in the current situation supported by all the agents. Value F_I of the solution is 8 (however the maximal value of F_I is 9).

Both above algorithms have an important property. When coefficient λ decreases or increases (SRC or SR algorithm, respectively), all coalitions in the previous cycle remain as candidates for solutions in the new cycle too. In other words, the more agents concede the more coalitions satisfy their requirement, but not historical coalition is reduced because of decreasing the requirement. That leads to the following lemmas.

Lemma 2 (for SRC algorithm)**.** Let $F_I|_\lambda$ be the total resulting utility of the solution of the SRC algorithm when each agent has to concede λ-times from the initial requirements. Then $\forall \lambda_1 > \lambda_2 : F_I|_{\lambda_1} \leq F_I|_{\lambda_2}$.

Proof. [outline] $\forall \lambda_1 > \lambda_2$, it is valid that $\forall i \in I, \beta_i|_{\lambda_1} \geq \beta_i|_{\lambda_2}$. $\forall K \in \Omega^i_{\beta_i}|_{\lambda_1}$ follows; then K has to be in set $\Omega^i_{\beta_i}|_{\lambda_2}$ too. Consequently, $\forall i \in I : \Omega^i_{\beta_i}|_{\lambda_1} \subseteq \Omega^i_{\beta_i}|_{\lambda_2}$, i.e. each potential coalition configuration when $\lambda = \lambda_1$ must also be a potential solution in case $\lambda = \lambda_2 \Rightarrow F_I|_{\lambda_1} \leq F_I|_{\lambda_2}$. □

The following lemma can be inferred similarly:

Lemma 3 (for SR algorithm)**.** Let $F_I|_\lambda$ be the total resulting utility of the solution of the SR algorithm when each agent has to reduce λ units of the initial requirements. Then $\forall \lambda_1 > \lambda_2 : F_I|_{\lambda_1} \geq F_I|\lambda_2$.

Proof. Similar as above. □

The quality of achieved solutions of the presented algorithms compared to the optimal one is expressed by the following theorems.

Theorem 5 (for SRC algorithm)**.** Let $\lambda_0 < 1$ be a value of coefficient λ when the SRC algorithm achieves a solution for the *first* time. Let F_I^{sup} be the total resulting utility of the solution of the algorithm and $F_I^{max} = \max\{F_I\}$ be the maximal resulting utility of all possible configurations. Then,

$$\lambda_0 F_I^{max} \leq F_I^{sup} \leq F_I^{max}.$$

Proof. $F_I^{sup} \leq F_I^{max}$ follows from definition of F_I^{max}. Let us denote

$$v_{max} = \sum_{i=1}^{n} v_i^0, \tag{25}$$

then, $F_I^{max} \leq v_{max}$, because $\forall i \in I, q_i^K \leq v_i^0$. On the other hand, each coalition that an agent inserts into its set $\Omega^i_{\beta_i}|_{i\in I}$ always satisfies the condition $\forall K \in \Omega^i_{\beta_i} | q_i^K \geq \beta_i = \lambda_0 v_i^0$. Then, $F_I^{sup} \geq \lambda_0 v_{max} \geq \lambda_0 F_I^{max}$. The theorem is proved. □

Similarly we can prove that the following theorem is valid too.

Theorem 6 (for SR algorithm)**.** Let λ_0 be a value of coefficient λ when the SR algorithm achieves a solution for the *first* time. Let F_I^{sup} be the total resulting utility of the solution of the algorithm and $F_I^{max} = \max\{F_I\}$ be the maximal resulting utility of all possible configurations. Then,

$$F_I^{max} - n\lambda_0 \leq F_I^{sup} \leq F_I^{max}.$$

Proof. [outline] Similarly to the previous proof, $F_I^{sup} \geq v_{max} - n\lambda_0 \geq F_I^{max} - n\lambda_0$, where v_{max} is defined by Equation (25). □

Lemmas 2 and 3 show that if agents continue searching after the first solution was found, the later solutions are always better than (or at least as good as) the current one. We can infer Theorems 5 and 6 are also valid for the newly achieved solutions. Value v_{max} could be identified easily; therefore the main goal is to identify λ_0 in order to predict the range of the optimal solution.

The difference between both algorithms is not too large. If values q_i^K are not too different, the SRC algorithm is preferable; but, if values q_i^K are distributed over a wide range, then the SR algorithm can be more suitable for use. A more complicated situation occurs when agents have different distribution of values q_i^K, e.g. in Example 1, agent A_1 might prefer to use the SRC algorithm, but for agent A_2 the SR algorithm is better. To overcome this problem we propose two other algorithms, in which agents do not concede at once, but successively one or only several ones at a time. The second modification is that agents do not have to concede

the same value, but in each turn agent A_1 can choose coefficient λ such that at least one unselected coalition satisfies its new (reduced) requirement. The aim of the first modification is to avoid useless investigation when agents are satisfied with the achieved solutions. The second modification prevents unpractical regressions, which do not bring improvements. The algorithms will be presented in the next section.

5.2.2 Automatic Negotiation Algorithms Based on Sequential Calculation

Two algorithms presented in this section are based on the assumption that only one or a limited number of agents make regression in each cycle and the agents do not have to concede equally. In both algorithms, instead of coefficient λ we use only β_i.

Sequentially Making Regression — SMR	*Agents with the Highest Requirements make Regression — HRR*
Phase 1 a. $\forall i \in I$ choose coefficient $\beta_i = v_i^0$. b. $\forall i \in I$ find set $\Omega^i_{\beta_i}$.	**Phase 1** a. $\forall i \in I$ choose coefficient $\beta_i = v_i^0$. b. $\forall i \in I$ find set $\Omega^i_{\beta_i}$.
Phase 2 $i = 1$. a. Search for solutions from sets $\Omega^j_{\beta_j}\vert_{j\in 1,\ldots,n}$. If any solution is found, then store it for comparison. b. Decrease coefficient β_i to: $\beta_i \leftarrow \max_{\forall K \subseteq I}\{q_i^K \mid q_i^K < \beta_i\}$. c. Update set $\Omega^i_{\beta_i}$ (call step 1.b), then $i = i + 1$ if $i < n$, otherwise $i = 1$ and return to step 2.a. d. If time expires, then stop. Otherwise return to Step 2.a.	**Phase 2** a. Search for solutions from sets $\Omega^j_{\beta_j}\vert_{j\in 1,\ldots,n}$. If any solution is found, then store it for comparison. b. Choose $i \vert \beta_i = \max_{\forall j=1,\ldots,n} \beta_j$. c. Decrease coefficient β_i to: $\beta_i \leftarrow \max_{\forall K \subseteq I}\{q_i^K \mid q_i^K < \beta_i\}$. d. Update set $\Omega^i_{\beta_i}$ (call step 1.b) and return to step 2.a. e. If time expires, then stop. Otherwise return to Step 2.a.

Fig. 4. An algorithm for creating optimal coalition configurations based on the sequential regression

Similarly to both previous algorithms, to accelerate the process solving before starting Phase 2 each agent can sort out coalitions that it is willing join according to value q_i^K. Both SMR and HRR are rather sequential search algorithms, which perform the search in only certain directions (where agents reduce their requirements) at once. Prediction of the range of the optimal solutions could be calculated similarly to the SR algorithm, but agents have to report how much they have already conceded. The complexity of the presented algorithms, however, cannot be precisely specified, and is practically low, because the set of coalitions that agents

are willing to join (always the best coalitions are selected at first) is not as huge as the theoretical prediction.

Notation: Com = complete search of all variants.

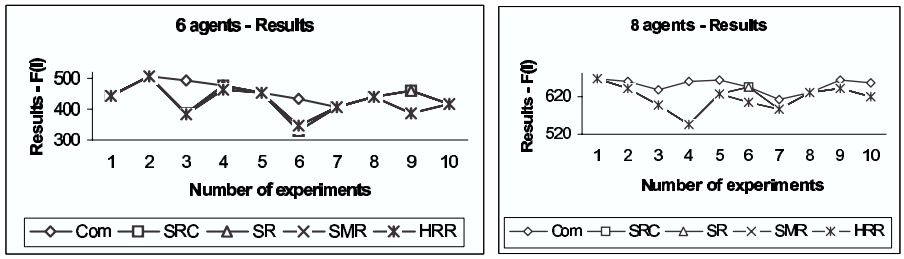


Fig. 5. Graph 1: The simulation results with 6 and 8 agents

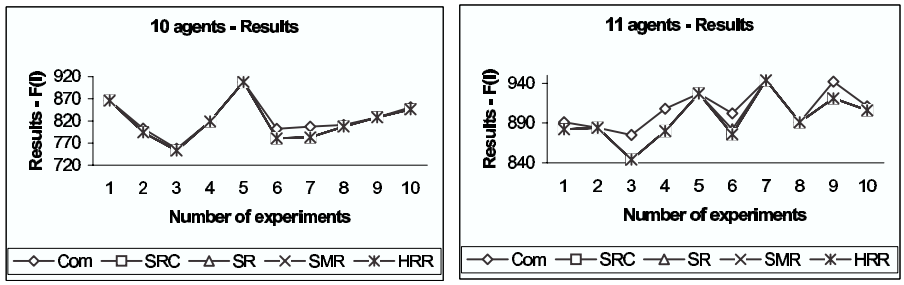


Fig. 6. Graph 2: The simulation results with 10 and 11 agents

Com	SRC	SR	SMR	HRR
6599	26	15	39	39
6328	40	22	47	49
5902	69	65	94	109
5809	126	61	152	152
5751	186	83	216	216
6600	24	15	12	12
5591	67	28	55	55
7052	57	35	69	75
6406	6	6	20	22
7198	173	82	197	197

Table 3. Number of cycles — 8 agents

In Graphs 1, 2 and Tables 3, 4 we present the comparison of all the proposed algorithms and a complete search in order to ensure the optimality of the obtained solutions.

Com	SRC	SR	SMR	HRR
134399	43	33	72	67
170593	13	16	17	17
159661	110	52	186	186
119208	99	89	175	170
140611	38	26	70	70
137295	187	97	327	334
160283	109	65	211	211
161738	91	45	14	14
164538	43	57	65	72
176702	87	60	129	129

Table 4. Number of cycles — 10 agents

In these experiments values q_i^K were generated randomly from interval $[0, 100]$. Values in Tables 3 and 4 show how many steps the program had repeated until the final result was found. These values may also be used for assessment of time consumption of each method. In graphs of results, the results achieved by complete searching always have the maximal values. The obtained results lead to the conclusion that the complete search approach is applicable for a small number of agents (up to 8), because the complexity is realizable. In case when the number of agents is larger (9 and more), the presented methods have a negligibly low complexity in comparison with the complete search approach, and they acquire results very close to the optimal ones. Better results could be obtained by modifying Step 2.b in the SMR algorithm. Instead of adding one coalition, an agent can attach more coalitions to increase the chance to obtain the optimal solutions. Experiment results also show that the number of examination steps to ensure that the obtained solutions are the optimal ones increases exponentially (from about 150 000 in case with 10 agents up to 1 million in case with 11 agents). Therefore, from the practical point of view it is preferable to use the presented algorithms.

The presented algorithms can be combined with other heuristic search algorithms to solve more complicated negotiation scenarios. In addition, the search can be distributed to all agents for parallel processing to accelerate the negotiation process. We will deal with these objects in future work.

6 MULTI-PARAMETER COALITION PROBLEMS

In practice the subject of joining coalitions often consists of various parameters, which could also be mutually influenced. Complexity of the problem solving grows exponentially with the number of parameters. This motivates us to search for methods to reduce the solution space. One of these methods presented in [1, 21] is using fuzzy sets for representation. The choice of each agent is represented by a fuzzy set, and negotiation in order to create optimal coalitions is calculated by using fuzzy operators. In this paper the multi-parameter coalition problem is represented by

using linear vectors. In the next section the basic characteristics of multi-parameter coalition problem will be presented.

6.1 Data Representation in the Multi-Parameter Coalition Domain (MPCD)

Let s be a number of parameters and let $\overline{q_i^K} = \{x_{i,1}^K, \ldots, k_{i,s}^K\}$ be a vector of the expected resulting utility that agent A_i receives by joining coalition $K \subseteq I$. Similarly, let $\overline{q_i^*} = \{x_{i,1}^*, \ldots, x_{i,s}^*\}$ be a vector of the expected resulting utility that agent A_i receives when it working alone. An operation $||\overline{q_i^K}|| = \sqrt{\sum_{j=1}^{s}(x_{i,j}^K)^2}$ denotes a norm of vector $\overline{q_i^K}$. Similarly, let I be decomposed to a number of disjoint coalitions: $I = \bigcup_i K_i$, then the criterion function is defined as follows:

$$\overline{F_I} = \sum_{K_i} \sum_{j \in K_i} \overline{q_j^{K_i}} = \left\{x_1^F, \ldots, x_s^F\right\}, \tag{26}$$

where $\forall r \in [1, s] : x_r^F = \sum_{K_i} \sum_{j \in K_i} x_{j,r}^{K_i}$.

In order to compare two different configurations we introduce the following definition.

Definition 7. Given two coalition configurations Y and Z. We say Y is better than or as good as Z if and only if: $||\overline{F_I}|_Y|| \geq ||\overline{F_I}|_Z||$.

Now the main task now is re-formulated as follows: to find the optimal coalition configuration that maximizes $||\overline{F_I}||$.

To simplify, we consider that Assumption 1 is also valid here. Agent A_i is willing to join coalitions K if it can gain more or at least as when working alone: $||\overline{q_i^K}|| \geq ||\overline{q_i^*}||$. Definition 4 in Section 3.2 about *acceptable* coalitions is also valid in the multi-parameter coalition domain. Sets $S_i|_{i=1,\ldots,n}$ are defined as follows: $S_i = \{K \subseteq I : ||\overline{q_i^K}|| \geq ||\overline{q_i^*}||\}$. With newly formulated sets $S_i|_{i=1,\ldots,n}$ we can prove that the following lemmas hold too.

Lemma 4. Lemma 1 is also valid in the multi-parameter coalition domain (MPCD).

Lemma 5. Theorem 2 is also valid in the MPCD.

Proofs of Lemmas 4 and 5 are similar to Section 3.2.

The problem of finding the optimal multi-parameter coalition configurations using linear vectors for representation is similar to the task introduced in Section 3 for one-parameter case, because the number of all possible coalitions remain unchanged. Using properties of the vector norm leads to the following interesting conclusion.

Lemma 6. $\max(||\overline{F_I}||) \leq \max(\sum_{K_i} \sum_{j \in K_i} ||\overline{q_j^{K_i}}||)$, where is defined by (26).

Proof. [outline] Using properties of the vector norm. □

In practice the criteria might have a different importance degree. In this case a vector of the resulting utility could be defined as: $\overline{q_i^K} = \{\omega_1^i x_{i,1}^K, \ldots, \omega_s^i x_{i,s}^K\}$, where ω_j^i is the weight by which agent A_i assesses its j-th parameter and $\forall i \in [1,n], \sum_{j=1}^{s} \omega_j^i = 1$. Then, the vector's norm is defined as:

$$||\overline{q_i^K}|| = \sqrt{\sum_{i=1}^{s} \left(\omega_j^i x_{i,j}^K\right)^2}. \tag{27}$$

$\overline{q_i^*}|_{i=1,\ldots,n}$ and $\overline{F_I}$ can be defined similarly. It can be proved that Lemma 6 is also valid for this case. In the next section we present methods for finding the sub-optimal coalition configuration in the MPCD.

6.2 Methods for Finding Optimal Coalition Configurations in the MPCD

The methods presented here are extended from those which have been proposed for solving the same problem in the one-parameter coalition domain. First, we shall discuss how the approximate method could be extended in the MPCD. Then, we shall present methods for automatic negotiation for finding the sub-optimal coalition configurations in the MPCD.

6.2.1 Searching for the Sub-Optimal Coalition Configurations with Approximated Values in MPCD

The method introduced in Section 4 can be extended for the use in the MPCD; to simplify, we present only the basic changes here. Let us define $\overline{q_i^k}$ as the approximate utility when agent A_i joins an arbitrary coalition with $(k-1)$ other agents.

The AAV algorithm is applicable without changes in the MPCD, but instead of one-parameter variables it will work with linear vectors. Since Equations (17) and (18) are valid only for one-parameter coalitions, phase 2 of AAV has to use the $A*$ search algorithm in order to ensure that the achieved solutions have the maximal value $||E(\overline{F_I})||$. Thus, the complexity of the algorithm is quite high; therefore we turn rather to the second group of the algorithms presented in Section 5, which have realizable dimensions in MPCD too.

6.2.2 Automatic Negotiation Algorithms for Creating Sub-Optimal Coalition Configurations in MPCD

All methods presented in Section 5 are applicable in MPCD, but with the following relevant modifications. Definition 6 is extended for the multi-parameter case as follows:

Definition 8 (for SRC and SR algorithms)**.** $\forall i \in I$, let $\overline{v_i^0} = \{v_i^1, \ldots, v_i^s\}$ be an s-dimension vector consisting of maximal values of individual parameters that agent A_i can gain in all possible coalitions: $\forall j \in [1,s] : v_i^j = \max_{\forall K \subseteq I}(\overline{q_i^K})_j$.

$\forall\overline{\beta} \in \mathcal{R}_+^s$ let $\Omega_\beta^i = \{K \subseteq I \mid \forall j \in [1,s] : (\overline{q_i^K})_j \geq (\overline{\beta})_j\}$ be a set of coalitions in which agent A_i can gain more or at least $\overline{\beta}$ where $(\overline{\beta})_j$, $(\overline{q_i^K})_j$ express element j of vector $\overline{\beta}$ and $\overline{q_i^K}$ respectively.

The SRC algorithm is applicable without any change. In the SR algorithm, we have to use $\overline{\lambda}$ as an s-dimension vector ($\overline{\lambda} = \{\lambda\}_{s-\text{times}}$). With regard to the similarity between Sections 5 and 6, the following lemmas will be presented without proofs.

Lemma 7 (for SR extended algorithm)**.** Lemma 3 is also valid in MPCD.

Theorem 7. Let $\overline{\lambda_0} = \{\lambda_0\}_{s-\text{times}}$ be the value of vector $\overline{\lambda}$ when the SR *extended* algorithm achieves solution for the *first* time. Let $\overline{F_I^{sup}}$ and $\overline{F_I^{max}}$ be vectors of the total resulting utilities of the solution of the SR *extended* algorithm and the optimal solution, respectively $||\overline{F_I^{max}}|| = \max\{||F_I||\}$. Then $||\overline{F_{sup}}|| \geq ||\overline{F_{max}} - n\overline{\lambda_0}||$.

Proof. Similar to Section 5. □

In order to extend SMR and HRR algorithms in MPCD we need the following definition.

Definition 9 (for SMR and HRR algorithms)**.** $\forall i \in I$, let $v_i^0 = \max_{K\subseteq I}\{||\overline{q_i^K}||\}$ be the maximal resulting utility that agent A_i can gain and let $\forall 0 < \beta < v_i^0$ $\Omega_\beta^i = \{K \subseteq I : ||\overline{q_i^K}|| \geq \beta\}$ be a set of coalitions in which agent A_i can gain more or at least β.

By using Definition 9, both the SMR and HRR algorithms could be applied in MPCD without changes.

We have tested all proposed methods in two-parameter coalition domain. The achieved results have confirmed the computational advantages of these methods in MPCD. All the extended methods have manageable complexities and concurrently reach very good final solutions (very close to the optimal ones).

There are other methods, which could be implemented in the negotiation process, e.g. the agents can concede sequentially according to each parameter, not all at once; or the SRC and SR algorithms can be extended in MPCD with using Definition 9, etc. We will discuss this problem in later work.

7 CONCLUSION AND FUTURE WORK

This paper has dealt with the problem how to create the optimal coalition configurations that maximize the collective performance. We have also shown and proved the main properties and conditions related to creation of coalitions. Further, we have proposed a method for finding sub-optimal solutions based on the approximate principle. Our method is appropriate for cases when the resulting utilities of agents change randomly and dynamically. The method has very low complexity in

the one-parameter coalition domain, and allows predicting the range of the globally optimal solutions. In Section 5 we have introduced four methods for finding sub-optimal coalition configurations that are based on the linear regression principle. The presented algorithms have been tested in the one-parameter coalition domain, and have achieved very good results. The achieved experiment results confirm the computational advantages of our methods. Finally, in Section 6 we have discussed the problem of creating coalitions when the resulting utilities consist of many parameters. All the methods proposed in Section 5 are extended for use in MPCD. However, the approximate method in MPCD has not achieved the expected results; we will improve it in our future work.

Because of large extent of the original problem solving, all methods which we have introduced here concentrate on searching for sub-optimal solutions. However, the achieved results of these algorithms allow us to calculate the range of the optimal solutions. In Section 6 we have mentioned some problems that can influence the accuracy of the final solutions in MPCD. One of them is inequality among single parameters (through weights $w_i|_{i=1,\ldots,s}$). When these values change, the problem solving becomes very complicated; therefore adaptive algorithms could be helpful for solving such cases. The next problem is creating optimal coalition configurations among self-interested agents. We will deal with these problems in our future work.

REFERENCES

[1] Dang, T.-Tung—Frankovič, B.—Budinská, I.: Agents Coalition in Coordination Process. XV IFAC World Congress b'02, Barcelona 2002.

[2] Wooldridge, M.—Bussmann, S.—Klosterberg, M.: Production Sequencing as Negotiation. Proceedings of the First International Conference on the Practical Application of Agents and Multi-Agent Technology (PAAM–97), pp. 709–726, 1996.

[3] Sandholm, T.—Larson, K.—Andersson, M.—Shehory, O.—Tohmé, F.: Coalition Structure Generation with Worst Case Guarantees. Artificial Intelligence 111, pp. 209–238, 1999.

[4] Stone, P.—Littman, M. L.—Singh, S.—Kearns, M.: An Adaptive Autonomous Bidding Agent. Journal of Artificial Intelligence Research 15, pp. 189–206, 2001.

[5] Cohen, W.—Schapire, R.—Singer, Y.: Learning to Order Things. Journal of Artificial Intelligence Research 10, pp. 243–270, 1999.

[6] Cantamessa, M.—Villa, A.: Negotiation Models and Production Planning for Virtual Enterprises. Proceeding of The IFAC Workshop on Manufacturing, Modeling, Management and Control — MIM 2001, Prague, Czech Republic, 1–5, 2001.

[7] Liebrman, H.—Nardi, B. A.—Wright, D. J.: Training Agents to Recognize Text by Example. Autonomous Agents and Multi-Agent Systems, 4, pp. 79–92, 2001.

[8] Gmytrasiewicz, P. J.—Durfee, E. H.: Rational Communication in Multi-Agent Environments. Autonomous Agents and Multi-Agent Systems, 4, pp. 233–272, 2001.

[9] Sandholm, T.—Sikka, S.—Norden, S.: Algorithms for Optimizing Leveled Commitment Contracts. In International Joint Conference on Artificial Intelligence (IJCAI), pp. 535–540, Stockholm, Sweden, 1999.

[10] Odubiyi, J. B.—Kocur, D. J.—Weinstein, S. M.: A Scalable Agent-Based Information Retrieval Engine. Proceedings of the First Annual Conference on Autonomous Agents. California USA, ACM Press/ACM SIGART, pp. 292–299, 1997.

[11] Wellman, M. P.: A MarketOriented Programming Environment and its Application to Distributed Multicommodity Flow Problems. 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. Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, pp. 355–359, 1997.

[13] Frankovič, B.—Dang, T.-Tung: Cooperating Agents for Planning and Scheduling. Proceeding of The IFAC Workshop on Manufacturing, Modeling, Management and Control — MIM 2001, Prague, Czech Republic, pp. 6–11, 2001.

[14] Derman, C.: Finite State Markovian Decision Processes. Mathematics in Science and Engineering, Vol. 67, Academic Press, London, 1970.

[15] Zlotkin, G.—Rosenschein, J.: Coalition, Cryptography and Stability: Mechanisms for Coalition Formation in Task Oriented Domains. Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pp. 432–437, 1994.

[16] Caillou, P.—Aknine, S.—Pinson, S.: Multi-Agent Models for Searching Pareto Optimal Solutions to the Problem of Forming and Dynamic Restructuring of Coalitions. Proceedings of the 15th European Conference on Artificial Intelligence, Lyon, France, pp. 13–17, 2002.

[17] Kraus, S.—Wilkenfeld, J.—Zlotkin, G.: Multiagent Negotiation Under Time Constraints. Artificial Intelligence, Vol. 75, 1995, No. 2, pp. 297–345.

[18] Tar, J. K.—Rudas, J. F.—BITO, J. F.: Group Theoretical Approach in Using Canonical Transformation and Symplectic Geometry in the Control of Approximately Modeled Mechanical Systems Interacting with Unmodelled Environment. Robotica, Vol. 15, 1997, pp. 163–179.

[19] Tran, D. V.—Hluchý, L.—Nguyen, T. G.: Parallel Program Model for Distributed Systems. Proceeding of EURO PVM/MPI 2000, Lecture Notes in Computer Science, Vol. 1908, 2000, pp. 250–257.

[20] Tran, D. V.—Hluchý, L.—Nguyen, T. G.: Parallel Program Model and Environment. Parallel Computing PARCO '99, pp. 697–704, 1999.

[21] Rudas, I. J.: Evolutionary operators; new parametric type operator families. Int. Journal of Fuzzy Systems, Vol. 23, 1999, No. 2, pp. 147–166.

APPENDIX 1 — PROOF OF THEOREM 3

Due to the fact that for each selected subset of agents there is a definite manner, which assigns where each agent has to be (Step 4). As a result the maximum number

of cases necessary to be examined is the same as the number of possibilities to select these subsets. Then:

$$\Delta_{GAS} \leq \frac{n!}{(k_1+k_2)!\,(k_3+k_4)!\ldots(k_{2[\frac{m}{2}]-1}+k_{2[\frac{m}{2}]})!} \tag{28}$$

if m is an even number. If m is an odd number,

$$\Delta_{GAS} \leq \frac{n!}{(k_1+k_2)!\,(k_3+k_4)!\ldots(k_{2[\frac{m}{2}]-1}+k_{2[\frac{m}{2}]})!\,k_m!}. \tag{29}$$

By comparing Equation (14) with (28, 29) the theorem will be proved.

APPENDIX 2

Similarly as in the previous proof, let $K_1 = \{a_1^1, \ldots, a_{k_1}^1\}$, $\ldots$, $K_m = \{a_1^m, \ldots, a_{k_m}^m\}$ be subsets of I, they create a stable coalition configuration for given decomposition (as shown in (16)). Let $K_1^\alpha = \bigcup_{i=1,\ldots,r_1} K_i$, $\ldots$, and $K_\alpha^\alpha = \bigcup_{i=r_1+\ldots+r_{\alpha-1}+1,\ldots,r_1+\ldots+r_\alpha} K_i$. Clearly, $|K_1^\alpha| = r_1 \cdot k_1^\alpha$, $\ldots$, $|K_\alpha^\alpha| = r_\alpha \cdot k_\alpha^\alpha$ and the members, which are in one of sets $\{K_i^\alpha\}|_{i=1\,\text{to}\,\alpha}$ can be exchanged arbitrarily. However, two members belonging to two different sets of $\{K_i^\alpha\}|_{i=1\,\text{to}\,\alpha}$ cannot be exchanged. This case is similar to the previous one; therefore by applying the similar way to prove, we can show that this theorem is valid too.

APPENDIX 3 — PROOF OF EQUATION (23)

- Sum_n denotes the number of variants of decomposition n to individual subsets
- s_i^n denotes the number of variants of decomposition n to individual subsets, where each subset has at least i members: $n = \sum_{j=1}^m k_j$ where $m \leq n$, $\forall j : k_j$ is a integer and $k_j \geq i$.

Clearly, $Sum_n = s_1^n$. The variants of decomposition n to individual subsets can be divided to the following categories:

a. The first category involves such variants, where $k_1 = 1$ and $\forall j > 1$, $k_j \geq 1$. It is easy to verify that the number of such variants is s_1^{n-1}.

b. The second category involves all variants, where $k_1 = 2$ and $\forall j > 1$, $k_j \geq 2$. The number of such variants is s_1^{n-2}, etc.

c. This process may continue up to $[\frac{n}{2}]$-th category, where $k_1 = [\frac{n}{2}]$ and $\forall j > 1$, $k_j \geq [\frac{n}{2}]$. The number of such variants is $s_{[\frac{n}{2}]}^{n-[\frac{n}{2}]}$.

d. For higher categories, where $\forall j$, $k_j > [\frac{n}{2}]$, only one variant that satisfies this condition exists: there is a case where $m = 1$ and $k_1 = n$.

Another problem that we have to prove that there is no such variant, which would be in two different categories simultaneously. For example: A variant in the first category ($k_1 = 1$) cannot be in the second one, because in the second category $\forall j,\ k_j \geq 2$. We can show in a similar way that a variant in the second category cannot be among variants in the third category, etc. Thus, the proposed categories are disjunctive. Then:

$$Sum_n = s_1^n \geq 1 + s_1^{n-1} + s_2^{n-2} + \ldots + s_{[\frac{n}{2}]}^{n-[\frac{n}{2}]}. \qquad (*)$$

In addition, we will show that each variant of decomposition n to individual subsets has to be in one of the proposed categories.

If $m > 1$, then $1 \leq \min_j\{k_j\} \leq [\frac{n}{2}]$. Let $\delta = \min_j\{k_j\}$, then $1 \leq \delta \leq [\frac{n}{2}]$ and from this point it follows: a variant $n = \sum_{j=1}^{m} k_j$ must belong to the δ-th category as assumed above. If $m = 1$, then $k_1 = n$ and only one such variant exists. Therefore:

$$Sum_n = s_1^n \leq 1 + s_1^{n-1} + s_2^{n-2} + \ldots + s_{[\frac{n}{2}]}^{n-[\frac{n}{2}]}. \qquad (**)$$

By combination of (*) and (**) Equation (23) will be achieved.

APPENDIX 4 — PROOF OF EQUATION (24)

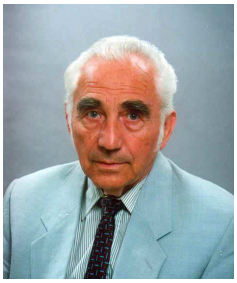
For $i < [\frac{n}{2}]$ we have to calculate value s_i^n, which is the number of variants of decomposition n to individual subsets, where each subset has minimal i members.

Similarly to the proof in the previous appendix we can separate variants in set s_i^n to single categories, beginning with $k_1 = i$ up to $[\frac{n}{2}]$. The number of variants in each category is exactly equal to s_j^{n-j}, where $j = i$ up to $[\frac{n}{2}]$. For another case where each member is larger than $[\frac{n}{2}]$, only one variant satisfying this condition exists; namely, $m = 1$ and $k_1 = n$. Therefore, $s_i^n = 1 + \sum_{j=1}^{[\frac{n}{2}]} s_j^{n-j}$ and Equation (24) is valid.

APPENDIX 5 — CHEBYSEV INEQUALITY

Let X be a random variable with a variance $var(X)$; then $\forall\delta > 0 : P(\{|X - E(X)| \leq \delta\}) \geq 1 - \frac{var(X)}{\delta^2}$.

T.-Tung Dang was born in 1973. He received the Ing. (MSc.) degree from the Slovak Technical University, Faculty of Electrical Engineering and Informatics, Bratislava in 1997. Currently he is working towards his PhD degree. His research interests cover MAS, planning and scheduling, reasoning and knowledge management.



Baltazár Frankovič, a member of a scientific council of the Technical University and its Faculty of Informatics; a member of IFAC/IFIP TC (technical committee), scientific society ECCAI, Slovak Society of Cybernetics and Informatics, J.von Neumann Society Budapest; a member of the Central European Academy of Sciences and Arts; .a member of the Editorial Board of several journals, including Computing and Informatics. His current research interests are in the field of modeling and simulation of Flexible Manufacturing Systems (FMS) (where he is the head of the project), by means of utilizing the adaptive and learning control algorithms, applied knowledge representation and applied artificial intelligence. The research in this area is aimed at creation of systems for the technology design verification. His publications include 5 books, 124 scientific papers from the area of optimal control, adaptive and learning systems and utilization of artificial intelligence in discrete events dynamic systems.

Ivana Budinská graduated from Slovak Technical University, Faculty of Electrical Engineering in 1987. She is with the Institute of Informatics, Slovak Academy of Sciences, Department of discrete processes modelling and control. Her research interests include DEDS, control theory, MAS.