# Creating agent coalitions by using approximated values

**Dang T.-Tung, Frankovic Baltazar and Budinska Ivana**

Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9, 84507, Bratislava, Slovakia, {utrrtung, utrrfran}@savba.sk

*Abstract: Task execution in multi-agents systems might require cooperation among agents. An agent coalition is a special kind of cooperation, whose purpose is to improve the collective performance. Since creating coalitions requires complicated negotiations among participating agents, this paper suggests other approach, which tries to avoid this dilemma by using approximated values to find an optimal coalition configuration (CC). This method is realizable even the number of agents is large and allows predicting a range where a value of the optimal solution could be.*

## 1 Introduction

In manufacturing or other application field cooperation among autonomous agents might be necessary in order to fulfil the designed aim. Let us take an example where a number of agents (factories), each of them owns a number of equivalent machines and has to satisfy any order. Of course, as an autonomous agent (factory), the assigned task could be satisfied without interventions or helps of other ones. But the use of machines might not be as effective as each agent can expect. Some agents might have too busy plan, on the other hand, some other ones might have idle machines. Joining a coalition in such a case can improve all agents' performance. If agents share machines and sets of tasks that each of them has to do, then it is possible to find a better manner to fulfil the assigned goals (faster, cheaper, more profits, etc.). This paper tends to find a method how to choose agents to distribute to such coalitions, in order to improve the collective performance.

This paper is a continuation of two previous ones [1], [2] which dealt with the same topic too. Two methods *merging agents* and *linear regression* were proposed for finding a sub-optimal coalition configuration in these works. These methods require knowing all rewards that each agent can get when it joins any coalition before starting a process of solving. Of course, that requirement is difficult being satisfied, when agents have complicated payoff functions and time available to finish all processes of negotiation is limited. The novelty presented here is that. Instead of realistic values agents exploit approximated ones to find optimal coalition configuration (CC). Clearly, a solution achieved by this method might not be the actually optimal one, but we will try to estimate the difference between them.

## 2 Formulation of the problem solving

Firstly, Let us introduce some remarks: {**Agent_Set**} ={$A_1$, $A_2$, …, $A_n$}denotes a set of *n* agents, and **I** denotes a set of their index, **I**={1,..,n}. A remark $i\in\mathbf{I}$ means agent $A_i$ from set **Agent_Set**; $K\subseteq\mathbf{I}$ denotes a subset created by the agents from set **Agent_Set** with index belonging to set K. The next assumption is:

*Assumption 1*: each agent takes part in one and only one coalition.
Further notations are: $\forall i\in\mathbf{I}$, $K\subseteq\mathbf{I}$,

- The *agent's reward* $f_i|_{i=1,..,n}$ is a function mapping from a set of all possible plans that the agent can apply to $\Re_+$. It is used to assess the agent's performance.
- $q_i^* \geq 0$ denotes a reward for agent $A_i$ if it works alone,
- $q_i^K \geq 0$ denotes a reward for agent $A_i$ if it joins coalition K,
- $f_i = q_i^*$ if agent $A_i$ works alone, or $q_i^K$ if agent $A_i$ joins coalition K.
- $F_K$ denotes the total reward for agent's coalition $K\subseteq\mathbf{I}$ and it is defined as follows:

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

In order to show how complicated calculation of values $q_i^*$ and $q_i^K$ is we should take a simple example, which was used in [3] for illustration.

*Example 1*: Given two producers (agents) **I**=[1,2]; each of them owns two different resources that are used to perform certain kinds of products. Operations could be executed in the equivalent resources of an arbitrary agent and with the same quality.

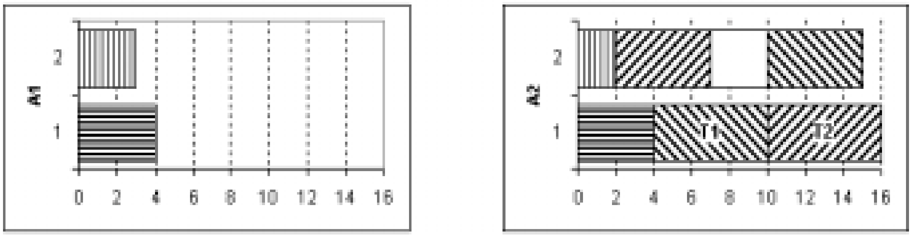


**Fig. 1**: Initial plans agent $A_1$ and $A_2$

In Figure1 there are initial plans that each agent has to perform. Slots with same pattern are operations of one product and they must be executed one after other as in these plans. Another assumptions are: Cost(*transport*) between $A_1$ and $A_2$ is 1$ and *Time_tran*(Δ) between them is 1 unit time where Δ is any operation. The agent $A_1$ agrees with payment in interval [2,6] $ and the agent $A_2$ is agreeable to pay from interval [0,5] $. For both the agents the following condition is valid: if each product is terminated before a time 15 units, agent gets 15$, if after this time gets 5$ (deadline for all products is 15 time units).

Figure 1 shows that, $A_1$ has two different products and it is able to terminate before deadline (the total payoff is 30$); $q_1^*=30$. Agent $A_2$ has four products but one product is terminated after deadline (at time 16) therefore it gets only (3*15+5=) 50$; $q_2^*=50$; and wants to ask agent $A_1$ to help. If both agents exchange whole plans, then they may agree with such plans as shown in the following figure.

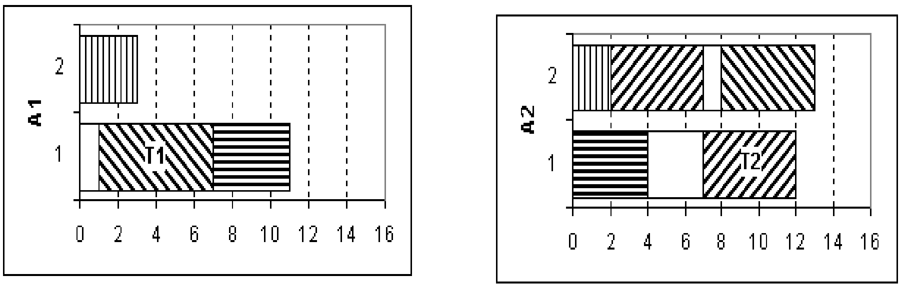


**Fig. 2**: A new plan for agents

In that case, for a coalition K={1,2}, the agent's expected utility is: $q_1^{\{1,2\}}=32$, $q_2^{\{1,2\}}=56$ and the coalition outcome is $F_{\{1,2\}}=88$.

The plans shown in Figure 2 achieve the best reward for collective performance of a coalition K={1,2}, not for each individual agent. It is easy to recognize that there are various ways to create new plans with different rewards. Moreover, if payoff functions are more complicated, a negotiation process might take much more time than the process of searching for an optimal CC. That fact motivates us to approximate rewards that agents can get to avoid complicated negotiations among them.

# 3 Evaluation of coalitions by approximated value

Let $q^k_i|_{k=1,\dots,n}$ be the approximated reward that agent $A_i$ can receive when joins an arbitrary coalition with ($k$-1) other agents. If all values $q^K_i|_{K\subseteq \mathbf{I}}$ are known, then this variable could be calculated as follows:

$$\forall k \in [1,n], \text{ then, } q_i^k = \frac{1}{m_k} \sum_{\forall K \subseteq I, where |K|=k} q_i^K \tag{2}$$

where $m_k$ is a number of all coalitions consisted of $k$ members including agent $A_i$: $m_k = \binom{n-1}{k-1}$ . This variable expresses the average reward that agent $A_i$ can obtain in $k$-member coalitions. For general case, Equation (2) could be rewritten as: $q_i^k = E(q_i^K|_{|K|=k})$. Estimates of $E(q_i^K)$ could be made even if some values $q_i^K$ are unknown. Instead of trying all coalitions of the same size, each agent can take a small subset of them and calculates how much it can get in such coalitions. After that, an agent can make an estimate of $E(q_i^K)$ by using an appropriate combination of already identified values $q_i^K$ . However that is not the main objective of this paper. We assume that exist a method to get these approximated values, which are then applicable to find an optimal CC.

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

$$\mathbf{I} = \bigcup_{i=1}^{m} K_i \text{ and } \forall i \neq j \in [1,m]\text{: } K_i \cap K_j = \{\emptyset\}. \ |K_i| = k_i \text{ and } n = \sum_{i=1}^{m} k_i \tag{3}$$

Then, by combination with Equation (1) we can get:

$$E(F_I) = E(\sum_{K_i} \sum_{j \in K_i} q_j^{K_i}) = \sum_{K_i} \sum_{j \in K_i} E(q_j^{K_i}) = \sum_{K_i} \sum_{j \in K_i} q_j^{k_i} \tag{4}$$

Since variables $q_j^K|_{\forall j,\ K}$ have the same distribution. Therefore, a variance of $F_I$ could be calculated as follows:

$$Var(F_I) = Var(\sum_{K_i} \sum_{j \in K_i} q_j^{K_i}) = \sum_{K_i} \sum_{j \in K_i} Var(q_j^{K_i}) \tag{5}$$

Combination of both the average value and the maximal variance allows predicting the range of a value of the optimal solution (e.g. using approximation method with the maximal credibility). Now, the main goal is to find such a configuration of set **I** as shown in Equation (3), which maximizes value E($F_I$) defined by Equation (4). The final solution is a CC, which has the highest *expected* reward among all.

# 4 Searching for the optimal coalition configuration with approximated values

In this section we present a generic method for resolving the introduced task above. Let consider an arbitrary configuration consisted of *m* coalitions with $k_1,\ldots,k_m$ members respectively, $n=\sum_{i=1}^{m} k_i$ . Theoretically, a number of such configurations are:

$$\Delta = \frac{n!}{k_1!*\ldots*k_m!}, \text{ if } k_1 \neq k_2 \neq\ldots\neq k_m, \text{ or} \quad (6)$$

$$\Delta' = \frac{n!}{(r_1 k_1^{\alpha})!\times..\times(r_\alpha k_\alpha^{\alpha})!}, \text{ if } \sum_{i=1}^{\alpha} r_i = m \text{ and} \quad (7)$$

$$k_1 = .. = k_{r_1} (= k_1^{\alpha}) \neq k_{r_1+1} = .. = k_{r_1+r_2} (= k_2^{\alpha}) \neq \ldots = k_m (= k_\alpha^{\alpha}). \quad (8)$$

Proof of Equations (6-8) could be achieved by using standard combinational methods. That means agents necessarily examine $\Delta$ (or $\Delta'$) different configurations with the same structure (a number of coalitions and sums of members in each coalition). But a lot of configurations can be omitted by using an appropriate method to reduce and as a result the practical number of cases that agents need to examine is not as high as the theoretical ones. To reduce useless configurations, I introduce the following definition.

*Definition 1*: Let each agent use the approximated values defined by Equation (2) and let be $n=k_1+ k_2 +..+ k_m$, where $k_i$ are integer. Then, a configuration $\mathbf{I}=\cup K_i|_{i=1,..,m}$, where each coalition $K_i|_{i=1,...,m}$ 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 value E($F_I$).

The final solution that maximizes E($F_I$), of course is one of these *stable* CC. Therefore, the focus now is to find a *stable* CC, when a sum of coalitions and numbers of members included in them are known.

Given a number of coalitions *m* and numbers of members in each coalition $k_1,\ldots,k_m$ ($n=k_1+ k_2+ \ldots+k_m$). Let consider the first case when $k_1,\ldots,k_m$ are different. Without loss of generality assume that $k_1 > k_2 >\ldots> k_m$ and $K_1, \ldots, K_m$ are *m* disjoint coalitions consist of $k_1,\ldots,k_m$ members respectively, which create a *stable* configuration.

From Definition 1 it infers that, if both the agents $i \in K_1$ and $j \in K_2$ are exchanged, value $E(F_I)$ will not increase. That means:

$$\forall i \in K_1, j \in K_2: \; q_i^{k_1} + q_j^{k_2} \geq q_j^{k_1} + q_i^{k_2} \\ \Leftrightarrow q_i^{k_1} - q_i^{k_2} \geq q_j^{k_1} - q_j^{k_2} \tag{9}$$

Similarly, it is possible to get:

$$\forall i \in K_3, j \in K_4: \; q_i^{k_3} + q_j^{k_4} \geq q_j^{k_3} + q_i^{k_4} \\ \Leftrightarrow q_i^{k_3} - q_i^{k_4} \geq q_j^{k_3} - q_j^{k_4} \tag{10}$$

Etc. Equation (9) could be explained by the words as follows. If all the agents are sorted according to value ($q_i^{k_1} - q_i^{k_2}$),$i \in \mathbf{I}$, 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 3).

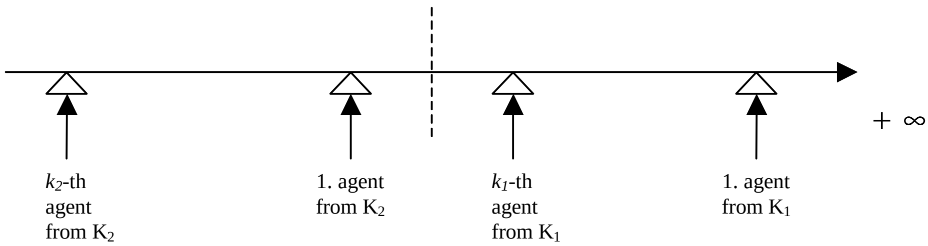


**Fig. 3**: An agent's order according to ( $q_i^{k_1} - q_i^{k_2}$ ).

It might be possible to happen a situation as shown in Figure 4, where $r_1$ agents ($r_2$ from $K_1$ and ($r_1$-$r_2$) from $K_2$, respectively) have the same value $q_i^{k_1} - q_i^{k_2}$ ($r_1 > r_2 > 0$). In this case, a number of *stable* configurations could be more by exchanging these $r_1$ agents from one coalition to the second, but value $E(F_I)$ remains unchanged. Therefore to get value $E(F_I)$ it suffices to examine one of these configurations. The similar conclusion could be made for agents from $K_3$, …, $K_m$.

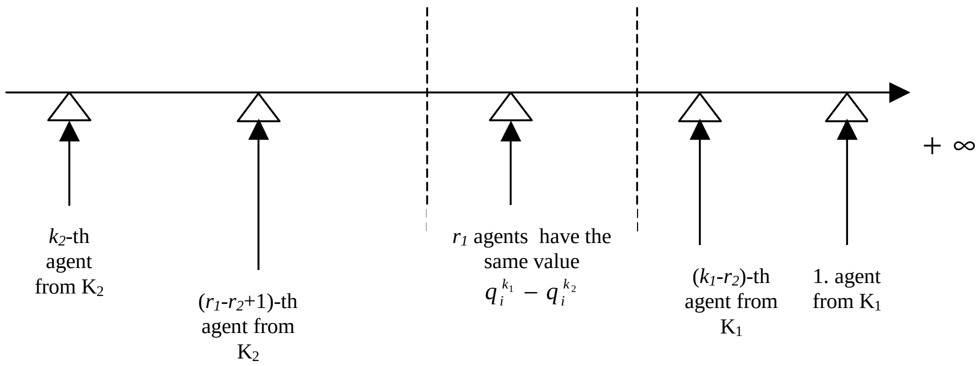


**Fig. 4**: A special case when $r_2$ agents from $K_1$ and $(r_1 - r_2)$ agents from $K_2$ have the same value ( $q_i^{k_1} - q_i^{k_2}$ ).

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

***G**reedy **A**lgorithm for finding a stable CC – **S**pecial case (GAS):*

Input: m, $k_1,\ldots,k_m$, ($k_1 > k_2 > \ldots > k_m$).

Output: a stable coalition configuration that maximizes E($F_I$).

1. $j = 1$.
2. Choose ($k_{2j-1} + k_{2j}$) arbitrary agents from the unselected ones.
3. Classify the selected agents according to value ( $q_i^{k_{2j-1}} - q_i^{k_{2j}}$ ), start with the largest one.
4. Distribute $k_{2j-1}$ first members in this queue to coalition $K_{2j-1}$, the remained ones to coalition $K_{2j}$.
5. Indicate the selected agents ($\mathbf{I}=\mathbf{I}\backslash(K_{2j-1}\cup K_{2j})$), j=j+1; and return to step 2 until $j = [\frac{m}{2}]$.
6. If $j = [\frac{m}{2}]$, calculate value E($F_I$) of the obtained configuration and to compare 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 necessary to examine is reduced to many times. We will discuss more in the next section.

By similar way we can propose the method to find a *stable* CC, when coalitions have arbitrary sizes, even more coalitions can have the same size. Due to the short framework of this paper we will show only an algorithm without detail explanation.

***G**reedy **A**lgorithm for finding a stable CC – **G**eneral case (GAG)*:

Input: *m*, $k_1,\ldots,k_m$, ( $k_1 = \ldots = k_{r_1} \neq k_{r_1+1} = \ldots = k_{r_1+r_2} \neq \ldots = k_{r_1+\ldots+r_\alpha}$ ). Denote:

$$k_{r_1} = k_1^{\alpha}, .., k_{r_1+..+r_\beta} = k_\beta^{\alpha}, .., k_{r_1+..+r_\alpha} = k_\alpha^{\alpha} \text{, and}$$
$$K_1^{\alpha} = \bigcup_{i=1,..,r_1} K_i, \ldots, K_\alpha^{\alpha} = \bigcup_{i=r_1+..+r_{\alpha-1}+1,...,r_1+..+r_\alpha} K_i. \tag{11}$$

And let us assume that $r_1 k_1^{\alpha} \geq r_2 k_2^{\alpha} \geq \ldots \geq r_\alpha k_\alpha^{\alpha}$.

Output: stable coalition configurations that maximize E($F_I$).

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

On the basis of two proposed algorithms for finding a *stable* CC, we suggest the following method to find a CC that maximizes a value E($F_I$). This algorithm consists of three following phases:

*An **A**lgorithm for finding an optimal CC with **A**pproximated **V**alues - (AAV):*

Input: $n$, $q_i^k |_{i=1,..,n}^{k=1,..,n}$ (the average values that agents can get).

Output: The coalition configurations that maximize E($F_I$).

- Phase 1: Decompose $n$ to smaller integer numbers: $n=k_1 + k_2 + ..+k_m$.
- Phase 2: For each variant, using GAS or GAG to find a *stable* CC.
- Phase 3: Choosing one of the achieved configurations that maximizes E($F_I$) for the optimal solution.

To ensure that the AAV is a realizable method, in the next section we will show some simulation results with real values.

# 5 Simulation results and discussion

To verify how the AAV works we have made a number of experiments with real values. In these experiments the total number of agents was selected from 8 to 12 and values of an expected utility that each agent can get in any coalition were generated randomly from interval [0,100]. On the basis of values of an expected utility each agent can calculate the average utility that it can get when joins any coalition of size from 1 to $n$. The AAV is applied to find a configuration that has a maximal value $E(F_I)$. Simultaneously, with the same data we try to find a maximal value of $F_I$. The purpose of these experiments is to show out how many configurations the AAV has to examine to achieve a value $\max(E(F_I))$ in comparison with the total number of all configurations that agents need to examine, in order to get a maximal value of $F_I$. Of course, it is a worth to show the total utility $F_I$ of the CC corresponding to a value $\max(E(F_I))$, in comparison with value $\max(F_I)$. A program is made in C++ and for one computer. Results of simulation are shown in following tables.

*Remark*: In these tables, Max $E(F_I)$ is a maximal value of $E(F_I)$, $F_I$ is a value of the actual CC corresponding to a value $\max(E(F_I))$. Column "% of Max(Fi)" expresses a ratio between a value $F_I$ of the achieved solution and $\max(F_I)$ in percentages. In columns "nc $E(F_I)$" and "nc $F_I$" there are a number of configurations that a program has to examine in order to get values $\max(E(F_I))$ and $\max(F_I)$, respectively. Numbers in the last column express a ratio between nc $F_I$ and nc $E(F_I)$.

**Table 1**: Simulation results with 8 agents

| 8 Agents | Max $E(F_I)$ | $F_I$ | % of $\text{Max}(F_I)$ | $\text{Max}(F_I)$ | nc $E(F_I)$ | nc $F_I$ | *Ratio* $\frac{\text{nc } F_I}{\text{nc } E(F_I)}$ |
|---|---|---|---|---|---|---|---|
| 1 | 489 | 489 | 72,7 | 673 | 97 | 3063 | 31,6 |
| 2 | 432 | 483 | 73,0 | 662 | 97 | 3685 | 38,0 |
| 3 | 408 | 336 | 55,3 | 608 | 97 | 2891 | 29,8 |
| 4 | 470 | 432 | 66,0 | 655 | 97 | 2867 | 29,6 |
| *average* | | | *66,7%* | | | | *32,2x* |

**Table 2**: Simulation results with 10 agents

| 10 agents | Max $E(F_I)$ | $F_I$ | % of $\text{Max}(F_I)$ | $\text{Max}(F_I)$ | nc $E(F_I)$ | nc $F_I$ | *Ratio* $\frac{\text{nc } F_I}{\text{nc } E(F_I)}$ |
|---|---|---|---|---|---|---|---|
| 1 | 505 | 545 | 66,1 | 824 | 796 | 36302 | 45,6 |
| 2 | 580 | 580 | 67,3 | 862 | 796 | 82774 | 104,0 |
| 3 | 548 | 591 | 70,5 | 838 | 796 | 67498 | 84,8 |
| 4 | 655 | 623 | 71,1 | 876 | 796 | 31330 | 39,4 |
| *average* | | | *68,8%* | | | | *68,4x* |

**Table 3**: Simulation results with 12 agents

| 12 agents | Max $E(F_I)$ | $F_I$ | % of $\text{Max}(F_I)$ | $\text{Max}(F_I)$ | nc $E(F_I)$ | nc $F_I$ | *Ratio* $\frac{\text{nc } F_I}{\text{nc } E(F_I)}$ |
|---|---|---|---|---|---|---|---|
| 1 | 634 | 497 | 52,5 | 946 | 5974 | 835362 | 139,8 |
| 2 | 603 | 564 | 55,8 | 1010 | 5974 | 739956 | 123,8 |
| 3 | 613 | 694 | 67,1 | 1034 | 5974 | 682665 | 114,3 |
| 4 | 616 | 549 | 55,6 | 988 | 5974 | 1326012 | 222,0 |
| *average* | | | *57,7%* | | | | *150,0x* |

On the basis of these results we can conclude that the AAV really needs to examine only an omissible number of configurations in order to achieve a maximal value of E($F_I$), in comparison with searching for a maximal value of $F_I$. With 8 agents, searching for a value max($F_I$) requires examining averagely 32 times more configurations than searching for max(E($F_I$)) needs. This coefficient might increase up to 200 times when the number of agents is 12. In most of cases, the actual CC corresponding to a value max(E($F_I$)) achieve averagely 60% of max($F_I$). From the theoretical point of view, 60% of a value of the optimal solution might not be too high bound, but regard to the practical use where agents work only with approximated values and they do not have to negotiate one with other to get all information, such a solution could be considered as acceptable. Essentially in such situations when agents do not have enough time to finish negotiation.

Since the aim is to find a configuration with a maximal value E($F_I$), we can say that the achieved coalition configuration has the highest expected reward among all possible ones, although the real value $F_I$ that this CC has might not be the highest in a given instance.

However, if all values $q^K{}_i$ are known, there are also other methods to find a coalition configuration, whose value $F_I$ even can reach closer to a value max($F_I$) than the proposed method, e.g., linear regression [2]. But the AAV has several significant advantages over other heuristic methods, mainly from the practical point of view, namely:

- *More practical*: the AAV does not have to work with realistic values $q_i^K$, but their average value, which could be predicted much more simpler than $q_i^K$. Most of other heuristic search methods, which tend to find a coalition configuration with max($F_I$), require knowing all values $q_i^K |_{\forall i \in \mathbf{I} \text{ and } K \subseteq \mathbf{I}}$. That means agents need to examine all $2^n$ possible coalitions and their corresponding parameters. In Example 1 introduced above there are only two agents with two different machines and simple payoff functions, which have only several values. But it might be easy to admit that a number of combinations of plans that each agent can have when they join a common coalition are efficiently large. When an order or a place of execution of any operation changes, each agent will get a new plan with different outcomes. In a general instance, when the number of agents is large, calculating the maximal outcome of a coalition ($F_K|_{\forall K \subseteq \mathbf{I}}$) and a reward that each member of this coalition gets ($q_i^K |_{\forall i \in K}$) is much more complicated and it takes much more time than searching for an optimal coalition configuration with known values $q_i^K$. The AAV, in contrary, does not try to achieve a coalition configuration with the highest value $F_I$, but it tries to find a coalition configuration with the highest expected value E($F_I$). A lot of practical applications require such an aim - to find a coalition configuration with a maximal expected reward -

$E(F_I)$, because of the reason identifying all values $q_i^K |_{\forall i,K}$ in a short time is impossible.

- *Independency of variables*: all variables $q_i^k |_{i=1,..,n}^{k=1,..,n}$ are independent. Therefore, when something of any agent $A_j$ changes, e.g., a payoff function, any resource is not available for other ones, etc. only *n* variables associated with this agent need to be recalculated. It is not valid for a case when agents work with real values $q_i^K$ . All coalitions that include agent $A_j$ need to be recalculated to get new values $q_i^K$ . Since a number of such coalitions that include agent $A_j$ are $2^{n-1}$, it is easy to recognize that recalculation will take enough long time – there is a half of all possible coalitions that agents can join.
- *Realizable complexity*: simulation results confirm that the number of configurations that the AAV has to examine is manageable, even if a value *n* is large.

There are some main reasons that motivate to use approximated values to find an optimal CC, if it is impossible to identify all values $q_i^K$ in an acceptable time.

# 6 Conclusions

We have presented a new method to resolve the problem of creating and finding an optimal CC. Although this problem solving is known as a NP-problem, the proposed method has a manageable complexity; even the number of agents is large. Simulation results confirm a quality of the algorithm in various situations. We have also discussed several important reasons that motivate to use this algorithm in practice. However there is point that could be considered as a limitation of this paper that is how to approximate precisely rewards of agents ($q^k_i$) without negotiation. In the future we are going to deal with this problem in order to get closer to the real-world situations.

**References**

[1] Dang, T.-Tung, Frankovic, B. And Budinska, I.: Agents Coalition in Coordination process. In Proceeding of The 15-th IFAC Congress b'02, Barcelona, July, 2002

[2] Dang, T.-Tung, Frankovic, B. and Budinska, I.: Create agent Coalition Based on a Dynamic Programming Approach. In Proceeding of The 15th European Conference on

Artificial Intelligence ECAI 02, Workshop “Agent Technologoes and Logistics”,Lyon, France, 16-24,July 2002

[3] Frankovic, B. and 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, pp. 06-11, august 2001. By Kovacs, George; Mim2001 ISBN 0080439624 December 1, 2001. Elsevier Science Ltd.

[4] Shehory, O. and Kraus, S.: Feasible Formation of Stable Coalitions among Autonomous Agents in Non-super-additive Environments. Computational Intelligence, 15(3):218--251, 1999.

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

[6] Caillou P., Aknine S. and Pinson S.: Multi-agent models for searching Pareto optimal solutions to the problem of forming and dynamic restructuring of coalitions. In Proceedings of the 15.th European Conference on Artificial Intelligence, Lyon, France, 13-17, 2002.

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