I.J. Information Engineering and Electronic Business, **2026, 1, 42-55**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2026.01.03

Evaluation of Coalition Fault Tolerance

Viktor Mashkov*
University J. E. Purkyne, Department of Informatics, Usti nad Labem, 40096, Czech Republic
Email: viktor.maskov@ujep.cz
ORCID iD: https://orcid.org/0000-0001-9817-3388
*Corresponding Author

Received: 18 August, 2025; Revised: 06 October, 2025; Accepted: 23 December, 2025; Published: 08 February, 2026

Abstract: The paper deals with alliances and coalitions that can be formed by entities. In the paper, we consider unselfish (not self-interested) entities that do their best to achieve their common goal(s) without expecting any compensations or payoffs. The number of alliance members is assumed to be limited and fixed. To solve specific tasks, alliance members form coalitions. Generally, many coalitions can be formed by alliance members, and the problem arises to select the best of them. It can be done on the basis of some criteria, one of which could be coalition fault tolerance. Despite the great volume of conducted researches, only a few metrics have been proposed that can be used to quantify the coalition fault tolerance. The paper proposes new metrics for measuring coalition fault tolerance. A simple example explains how to compute coalition fault tolerance by using the proposed metrics. Bi-objective optimization problem, in which one of the objectives is coalition fault tolerance, was solved in the paper. Compromise programming was used to solve the optimization problem.

Index Terms: Entities, Alliance, Coalition, Coalition Fault Tolerance, Multi-objective Optimization

1. Introduction

In the past years, diverse groups of entities have been researched. In general, an entity as a member of a group can act as an agent in multi-agent systems, as a robot in multi-robot teams, or as a government organization or military unit. Entities can cooperate and in this way can form alliances. In [1], the term “alliance” is understood as a group of agents that have agreed to share some of their private information and cooperate eventually. In the paper, we consider an alliance in a broader sense according to which the entities agree to contribute all (or part of) their capabilities to the alliance. Capabilities can be regarded as resources and services. Resources may include: materials, energy, information, communications and others [2]. In general, an alliance may have different properties, such as lifespan (long-time or short-time), profit from cooperation (selfish or unselfish when entity doesn’t have intention to gain any benefits/payoff from its participation in the alliance/coalition), possibility to split the capabilities (splittable or non-splittable), autonomy (ability of an entity to make decisions and act independently), expandability (expandable or non-expandable), etc. In this paper, we consider the alliance as a long-term cooperation among the entities. We assume that in the fault-free situation the members of the alliance remain the same. Entity population of the alliance doesn’t change during coalition formation and mission execution (i.e., the number of the alliance members is fixed). We also assume that entities are unselfish. When it is necessary to perform the necessary mission the entities of the alliance form the coalition(s).

Many formal description techniques, methods and tools were applied to solve the tasks of alliance and coalition formation. Here we enumerate only some of them: LOTOS [3], agent UML [4], Erlang/OTP platform [5], etc. Several aspects should be taken into consideration for solving the tasks of alliance and coalition formation. These aspects include: centralized or decentralized manner of coalition formation, mutual communication between alliance members during coalition formation, evaluation of the formed coalitions by using special criteria and choosing the best of them, etc. Coalition can be perceived as a short-term cooperation among the entities. In our research, the best coalition is the coalition that is able to carry out a task by using the minimum amount of capabilities or by engaging the minimum number of entities and has maximum value of coalition fault tolerance. Excess of coalition capabilities is costly and may be time consuming (e.g., it may require some coordination efforts). The number of entities in the coalition should be minimal because a greater number of entities can cause overheads due to communication, coordination and internal organization costs for the formation, operation and maintenance of the coalition [2].

An entity can fail during the coalition formation or while the task of the coalition is performed. After entity failure it is possible to recover the coalition by replacing the failed entity with the fault-free entity from the same alliance. In our research, we consider several types of coalition member failure such as complete, substantial, partial and minor failure. These types of failure correspond to the deterioration (reducing) of the entity's capabilities (e.g., deterioration of functionality or performance). Coalition can tolerate the failures of its members, and therefore can tolerate the loss of some amount of its capabilities when it has enough redundancy/excess of capabilities. Since excess of coalition capabilities is costly, it is necessary to have a minimum amount of redundant capabilities. On the other hand, lack of redundant capabilities may lead to the situations when substitution of the failed coalition member is the only one possible option (i.e., the other options that consist in reducing coalition capabilities cannot help). For some missions, (e.g., rescue operations) time is of the essence, and therefore the coalition's recovery time is limited. In view of this, there may not be enough time to substitute a failed entity.

In order to evaluate the coalition fault tolerance, it is necessary to take into account some assumptions made and restrictions imposed on the coalitions. Concept of fault tolerance and metrics for its measuring has been considered, mostly, in relation to complex and distributed systems [6, 7]. As concerns the coalitions formed by entities, the problem of choosing the appropriate metrics for measuring coalition fault tolerance or suggesting the ones has been little studied and deserves attention.

Coalition fault tolerance is just one of the objectives that should be provided. Taking into account the other objectives, a multi-objective optimization can be solved. Results of such optimization will allow determining the best coalition among the possible coalitions.

This paper is organized as follows. Section 2 provides related work. Section 3 describes the metrics for measuring coalition fault tolerance. Section 4 presents the results of computing fault tolerance of the simple coalitions and the results of the bi-objective optimization applied on the possible coalitions. Section 5 discusses the conclusions and future work.

2. Related Work

The concept of an alliance was proposed in [1]. The authors considered alliance members as different social entities (e.g., people, organizations, institutions, army troops, etc.). Entities agree to share their private information and form coalitions. In [8], the concept of restricted alliance was suggested. In restricted alliance, not all its members are willing to cooperate with all other alliance members. In our research, we assume that entities are unselfish. It means that an entity will deliver all of its resources to the coalition regardless of the coalition structure.

A coalition can be formed by autonomous agents [9, 10, 11, 12]. In this case, the agents are selfish (self-interested) and try to maximize their own expected payoffs while trying to maximize the group (joint) payoff.

In [13], the authors defined a cooperative game as a set of agents in which each subset of this set is called a coalition. In the given case, self-interested software agents can negotiate rationally in stable coalitions.

Modeling and simulation of coalition formation is presented in [14]. Paper [15] considered heterogeneous agents (represented by the countries) that form coalitions. Coalitions can interact with other coalitions and can change their composition. Unlike this approach, in our research, we strictly distinguish between two concepts: alliance and coalition. A coalition member can be substituted (i.e., coalition is changed) when it fails and there is an alliance member capable of making the substitution.

Coalition can be formed by robots [16, 17, 18]. Multi-robot coalitions are considered as multi-agent systems. In this case, the problem of coalition formation is considered in relation to multiple robot – multiple task problem [19]. The formation of a coalition also leads to the allocation of tasks among the autonomous robots. It is worth noting, that coalition formation for multi-robot tasks is still a significant problem. As for non-autonomous robots, to the best of our knowledge, the groups of non-autonomous robots do not attract much attention of researchers.

A coalition member may fail to complete a task. The recovery of the coalition consists mainly in substitution of the failed coalition members by available ones [20]. In [17], the problem of replacing a failed robot with another robot with similar capabilities was considered. The authors also introduced a fault tolerance coefficient (FTC) which can be incorporated into the algorithm that determines the best coalition. FTC takes into account resource distribution and coalition size.

In [21], to improve the system fault tolerance, an approach based on the use of replication of individual agents was proposed. Agent and its replica act as one entity. In this way, the authors try to reduce the complexity of the system.

A coalition member can leave the coalition for various reasons. In the context of multi-agent systems, a coalition member can depart from the coalition due to its assigned payoff. A coalition is called stable if no agent wants to leave the coalition [22, 23]. In our research, a coalition member is not self-interested, and therefore there is only one reason why a coalition member leaves the coalition, namely its failure.

Coalition fault tolerance is one of the important characteristics of the coalition. In general, there are also other important characteristics which we try either maximize or minimize. In many cases, maximizing one characteristic leads to decreasing another characteristic or vice versa. In the paper [24], multi-objective optimization was considered. In the given case, the objectives are of conflict to each other. Applications of multi-objective optimization can be found in multi-agent systems and multi-robot systems. In [25], multi-robot coalition formation problem was modeled as a multi-objective optimization problem with conflicting objectives. The authors considered evolutionary multi-objective optimization problem and proposed an effective algorithm for the multi-robot coalition formation problem. In our research, we considered only two objective functions that coalitions have to balance, and solved the bi-objective problem using one of the existing methods.

3. Metrics for Measuring Coalition Fault Tolerance

It is assumed that alliance *A* consists of *N* entities (members of alliance), i.e., $A=\{a_1,..,a_N\}$. To fulfill the necessary mission, the members of the alliance form coalition *COAL*. In general, each coalition member a_i , $a_i \in COAL$, has *m*-dimensional vector of real non-negative capabilities.

$$C_{a_i} = < C_1^{a_i}, .., C_m^{a_i} > \tag{1}$$

For simplicity reason, hereafter in the paper, we consider only one dimension of capability vector. It is also assumed that capabilities of coalition members can be mapped onto integer numbers. In the given case, the total coalition capabilities can be expressed as the sum of the capabilities that coalition members contribute to the coalition.

$$C_{COAL} = \sum_{i=1}^{k} C_{a_i} \tag{2}$$

Where *k* is the number of coalition members.

We assume that to perform a task, the coalition must have capabilities no less than the required capabilities C_{needed}. Thus, when

$$C_{COAL} \geq C_{needed} \tag{3}$$

a task can be performed.

The following simple example illustrates the capabilities of a coalition.

3.1. Example 1

Coalition consisting of three autonomous mobile robots (R_1, R_2 and R_3) has to perform a mission. For example, dismantle the rubble after a building collapse. Robots have different sizes and powers. Robot R_1 is capable of moving a load weighing a maximum of 100 kg. Robot R_2 is capable of doing the same work for a load weighing a maximum of 80 kg. Robot R_3 is capable of doing this work for a load weighing a maximum of 60 kg. In the given case, the robot capabilities can be interpreted as pushing force and can be measured by the weight of the load that the robot can move.

Coalition capabilities equal the sum of robots capabilities, i.e.,

$$C_{COAL} = \sum_{i=1}^{3} C_{R_i} = 240kg \tag{4}$$

One of the tasks of the mission is removing a larger obstacle out of the way to clear the passage. This obstacle weighs 200 kg. Thus, the required capabilities C_{needed} for performing this task are equal to 200 kg. Since the condition (3) is satisfied, the task can be performed. It should be noted that no coalition consisting of two robots can perform this task. Excess of robots' capabilities can be used to compensate the loss of capability of one of the robots. This loss of capability can be caused by failure of one of the robot's elements.

3.2. The failure function

The failure of a coalition member is manifested by the loss of all or part of its capabilities. To describe the dependence of the failure of a coalition member on the loss of capabilities of this coalition member, the failure function $f_F(L)$ is introduced. We assume that the failure function is a linearly increasing function. The failure function $f_F(L)$ takes real numbers from the interval [0 ÷100] as input and produces real numbers lying in the interval [0 ÷1] as output.

We consider four types of failure: complete, substantial, partial and minor.

Complete loss of capabilities (i.e., L = 100%) corresponds to complete failure (see Fig. 1).

Loss of capabilities that satisfies inequality

$$L_1 \leq L < 100 \tag{5}$$

corresponds to substantial failure.

Loss of capabilities that satisfies inequality

$$L_2 \leq L < L_1 \tag{6}$$

corresponds to partial failure.

And finally, loss of capabilities that satisfies inequality

$$0 < L < L_2 \tag{7}$$

corresponds to minor failure.

Table 1 presents the types of failure of a coalition member. In Table 1, values V_1 and V_2 can be set based on the available information about the entity and the tasks that must be performed. Hence, different types of coalition member failure can be expressed by using the failure function $f_F(L)$.

We assume that the conditional probabilities of different types of failure can be determined on the basis of information collected about entity's failure. These conditional probabilities of different types of failure are denoted as follows. P_C = *P(complete/failure)*, P_S = *P(substantial/failure)*, P_P = *P(partial/failure)* and P_M = *P(minor/failure).*

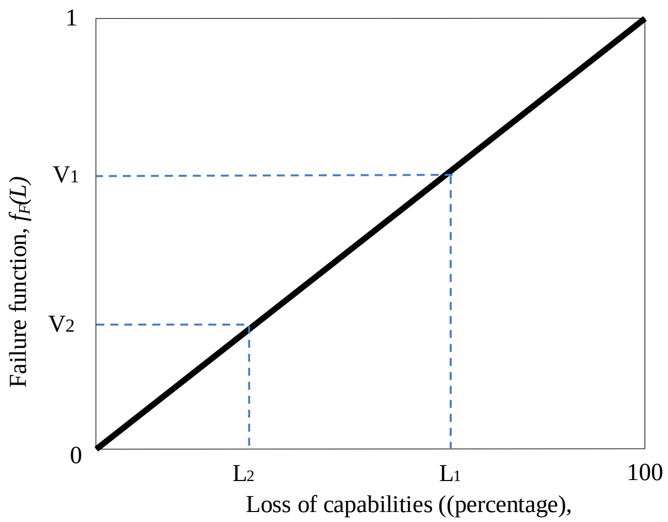


Fig. 1. The failure function.

Different types of failure can be described with the help of Example 1. Failure of robot's locomotion system is a substantial or complete failure. Failure of robot's sensor can be considered a partial failure. Failure of one of the robot's actuators, which can be caused by mechanical wear, dirt clogging or poor functioning of a communication system, is a minor failure.

Table 1. The types of coalition member failure

Type of failure	**Value of the failure function**
Complete	$f_F(L) = 1$
Substantial	$V_1 \leq f_F(L) < 1$
Partial	$V_2 \leq f_F(L) < V_1$
Minor	$0 < f_F(L) < V_2$

3.3. Assumptions

In this paper, some assumptions related to failures of coalition members and coalitions that can be formed were made. It is assumed that multiple failures of the coalition members have a very low probability (i.e., the value of the probability of occurrence of multiple failures is very small or negligible). It is also assumed that a failed coalition member should be substituted by no more than one fault-free alliance member if substitution is possible. A complete failure of any coalition member leads to coalition failure if the coalition members cannot be substituted. It can occur when the formed coalition doesn't have redundant coalition members.

After the failure of a coalition member, three situations are possible.

3.4. The First Situation

In the given case, the failed coalition members cannot be substituted for some reasons such as lack of time, very long distance between coalition members and alliance members, etc. Coalition fault tolerance can be provided by exploiting the redundant capabilities of coalition members. It is assumed that coalition member capabilities (resources) are splittable. It means that a failed coalition member can contribute to the coalition only part of his capabilities. In fault-free situation, a coalition member can split his capabilities and contribute to the coalition only necessary amount of capabilities. As soon as one of the coalition members fails, it can increase the amount of capabilities that it contributes to the coalition.

The failure of coalition member $a_i, i \in \{1, .., k\}$ can be expressed by the value of its failure function $V^{(i)}$ (i.e., $f_F(L) = V^{(i)}$, where $(0 < V^{(i)} \leq 1)$. Value $V^{(i)}$ corresponds to the loss of coalition member capabilities which is equal to 100 $V^{(i)}$

(expressed as a percentage). Taking into account that loss of 100% means the loss of all capabilities of coalition member a_i, the loss of 100 $V^{(i)}$% means the loss of capabilities equals to $V^{(i)}C_{ai}$. Where C_{ai} is the initial capabilities of coalition member a_i . It means that the failed coalition member now can contribute to the coalition the capabilities equal to C_{ai} (1 - $V^{(i)}$). If the inequality

$$(C_{COAL} - V^{i}C_{a_i}) \geq C_{needed} \tag{8}$$

is satisfied, the coalition is still able to perform the task. In the given case, we can say that the coalition tolerates the failure of its member. It is worth noting, that inequality (8) may not be satisfied for some values of $V^{(i)}$. In view of this, the task arises to determine the maximum value of $V^{(i)}$ for which inequality (8) is still met (see Fig. 2).

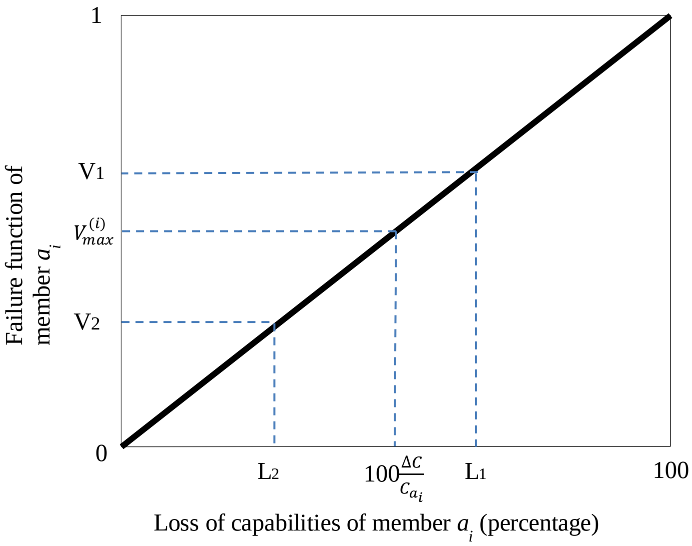


Fig. 2. The failure function of coalition member a_i.

A coalition can tolerate the maximum loss of its capabilities, which is caused by the coalition member failure, equals to ΔC, where $\Delta C = C_{COAL} - C_{needed}$.

Thus, the coalition member failure, which causes the loss of coalition capabilities by the value no more than ΔC, can be tolerated. In the given case, value $V_{max}^{(i)}$ is equal to

$$V_{max}^{(i)} = \frac{\Delta C}{C_{a_i}} \tag{9}$$

Taken into account the conditional probabilities of different types of coalition member failure, it is possible to determine the probability of the event that coalition tolerates the failure of its member a_i, i.e., $P_{CFT}^{(a_i)}$.

$$P_{CFT}^{(a_i)} = \begin{cases} 0, if\ V_{max}^{(i)} = 0 \\ P_M \frac{V_{max}^{(i)}}{V_2}, if\ V_{max}^{(i)} \leq V_2 \\ P_M + P_P \frac{V_{max}^{(i)} - V_2}{V_1 - V_2}, if\ V_2 < V_{max}^{(i)} \leq V_1 \\ P_M + P_P + P_S \frac{V_{max}^{(i)} - V_1}{1 - V_1}, if\ V_1 < V_{max}^{(i)} < 1 \\ 1, if\ V_{max}^{(i)} = 1 \end{cases} \tag{10}$$

As it follows from (10), the greater value $V_{max}^{(i)}$ the greater value $P_{CFT}^{(a_i)}$ (i.e., the greater coalition tolerance to the failure of coalition member a_i). It is possible to consider the coalition tolerance to all other coalition members and determine the following values

$$P_{CFT}^{(a_j)},\ j = 1, .., K,\ j \neq i \tag{11}$$

The expected value $E[P_{CFT}]$ is proposed as a metrics for measuring coalition fault tolerance.

$$E[P_{CFT}] = \sum_{i=1}^{K} p_i\, P_{CFT}^{a_i} \tag{12}$$

Where p_i – probability of failure of coalition member a_i , $i \in \{1,..,K\}$.

In the case when probabilities of failure of the coalition members are the same, the average value of coalition tolerance to failure of its members can be used, i.e.,

$$P_{CFT} = \frac{1}{K}\sum_{i=1}^{K} P_{CFT}^{(a_i)} \tag{13}$$

Calculation of values $P_{CFT}^{(a_j)}$ can be explained as follows. Let's consider one member of coalition, particularly a_1, which has capability C_{a_1}. Let coalition has excess of capabilities equal to ΔC. Coalition can tolerate only such failures of elements of member a_1 that cause the loss of his capability which corresponds to value $V_{max}^{(1)}$ of its failure function (see Fig. 2). This value can lie either in interval $[0, V_2]$ or in interval $[V_2, V_1]$ or in interval $[V_1, 1]$. These intervals correspond to the minor, partial and substantial failures of member a_1 respectively. If value $V_{max}^{(1)}$ is equal to V_2 then coalition can tolerate minor failures of its member a_1. In this case, $P_{CFT}^{(a_j)} = P_M$. If value $V_{max}^{(1)}$ is greater than V_2 but less than V_1 then coalition can tolerate all minor failures of member a_1 and some partial failures that cause the loss of its capability which corresponds to value V_{a_1} that lies in interval $[V_2, V_{max}^{(1)}]$. In the given case, probability that coalition tolerates the failures of elements of member a_1 can be determined as

$$P_{CFT}^{(a_1)} = P_M + P_P\, P\left(V_{a_1} \in [V_2, V_{max}^{(1)}]/partial\right) \tag{14}$$

Where P_M is the conditional probability that occurred failure is a minor failure;

P_P is the conditional probability that occurred failure is a partial failure;

$P\left(V_{a_1} \in [V_2, V_{max}^{(1)}]/partial\right)$ is the conditional probability of the event that occurred partial failure causes the loss of capability which corresponds to value V_{a_1} that lies in interval $[V_2, V_{max}^{(1)}]$.

Thus, coalition tolerates all minor failures of its member a_1 and only those partial failures that cause the loss of capability which corresponds to value V_{a_1} that lies in interval $[V_2, V_{max}^{(1)}]$.

The conditional probability $P\left(V_{a_1} \in [V_2, V_{max}^{(1)}]/partial\right)$ is determined as follows.

$$P\left(V_{a_1} \in [V_2, V_{max}^{(1)}]/partial\right) = \frac{V_{max}^{(1)} - V_2}{V_1 - V_2} \tag{15}$$

The wider the interval $[V_2, V_{max}^{(1)}]$ the greater is this probability. When this interval is equal to $(V_1 - V_2)$ all partial failures of coalition member a_1 can be tolerated as well as all its minor failures. In this case, coalition tolerance to failures of elements of member a_1 is equal to

$$P_{CFT}^{(a_1)} = P_M + P_P \tag{16}$$

Similar considerations can be applied to the situation when value $V_{max}^{(1)}$ lies in interval $[V_1, 1]$.

3.5. The Second Situation

In the given case, coalition fault tolerance is provided in two ways: by using redundant capabilities of coalition members and by substituting the failed coalition member.

Coalition member a_i, $i = 1,..,K$, can be substituted by alliance member $a_j \in \{1,..,N\}$ when either capabilities of alliance member a_j is greater or equal to capabilities of the failed coalition member a_i, i.e., $C_{a_j} \geq C_{a_i}$ or

$$C_{a_j} < C_{a_i}\ and\ \left(C_{a_i} - C_{a_j}\right) < \Delta C \tag{17}$$

Coalition tolerance in relation to failure of the substitutable coalition members is equal to

$$P_{CFT}^{(a_i)} = 1, for\ \ a_i \in T, i = [1, K] \tag{18}$$

Where T is the set of substitutable coalition members, $T \subset COAL$. Set of irreplaceable coalition members is denoted as S, $S=COAL-T$.

In the given case, the expression for coalition fault tolerance (13) can be modified as follows.

$$P_{CFT} = \frac{1}{K}\sum_{i=1/a_i \in S}^{K} P_{CFT}^{(a_i)} + \frac{|T|}{K} \qquad (19)$$

Where $|T|$ denotes the cardinality of set T, $|T| < K$.

According to equation (19), it is possible to conclude that the greater the number of substitutable members in the coalition the greater the coalition fault tolerance. The algorithm allows determining the coalition, which has the largest number of substitutable coalition members as compared to the other coalitions, is presented below. This algorithm gives approximately optimal solution because it doesn't take into account the possibility of the substitution of coalition member a_i by alliance member a_j when expression (17) takes place. The main idea of the algorithm consists in the following. From the perspective of coalition member substitution, it is desirable not to include in the coalition the alliance member that has maximum capabilities as compared to the other alliance members. It will allow to substitute any failed coalition member by this alliance member. However, there may be situations where a coalition does not have sufficient capabilities to perform a task unless it includes an alliance member that has maximum capabilities. Therefore, in these situations, an alliance member that has maximum capabilities is included in the coalition.

At the next step, the alliance member with the second largest capabilities is examined similarly (i.e., either it should or shouldn't be included in the coalition depending on the fact whether the condition $C_{COAL} \geq C_{needed}$ is satisfied or not). This searching procedure continues until the last in order alliance member is examined (see Fig. 3).

It is assumed that before performing the searching procedure the alliance members are sorted in ascending order according to their capabilities. At the preliminary step of searching procedure, the Algorithm 1 is performed to determine the initial coalition *COAL1*. The main goal of Algorithm 1 is to find the coalition *COAL1* capable of performing a task using as few entities as possible. In most cases, this number of entities is not minimal.

Algorithm 1.

Input: Alliance $A = \{a_1,.., a_N\}$, C_{needed}, $C_{a_1}, .., C_{a_N}$, N
int i,j
for $i \leftarrow 0$ to $N-2$
    $A = A - \{a_{N-i}\}$
    $S = \sum_{j=1}^{N-1-i} C_{a_j}$
        if $S < C_{needed}$
            $COAL1 = A \cup \{a_{N-i}\}$
            return $COAL1$
        elseif $i = N-2$
            return a_1
Output: either $COAL1$ or coalition member a_1

If the algorithm doesn't output *COAL1* then there is no need in coalition because one alliance member a_1 can perform a task.

Algorithm 2 is the main part of the searching procedure. It allows finding the coalition that has the largest number of substitutable coalition members, and consequently, the largest value of coalition fault tolerance. Before executing this algorithm, coalition members are sorted in ascending order according to their capabilities. Algorithm 2 corresponds to the block diagram of the searching procedure depicted in Fig. 3.

Algorithm 2.

Input: $COAL1$, C_{needed}, C_{COAL1}, $C_{a_1}, .., C_{a_k}$, K
int i
for $i \leftarrow 0$ to $K-1$
    $COAL \leftarrow COAL1$
    $C_{COAL} \leftarrow C_{COAL1}$
    $COAL \leftarrow COAL - \{a_{K-i}\}$
    $C_{COAL} \leftarrow C_{COAL} - C_{a_{K-i}}$
        if $C_{COAL} \geq C_{needed}$ then
            return $COAL$
        elseif $i = K - 1$ then
            return $COAL1$
Output: either $COAL$ or $COAL1$

If the algorithm doesn't output *COAL* then the initial coalition *COAL1* is the result of the searching procedure. Otherwise, the result of the searching procedure is coalition *COAL*.

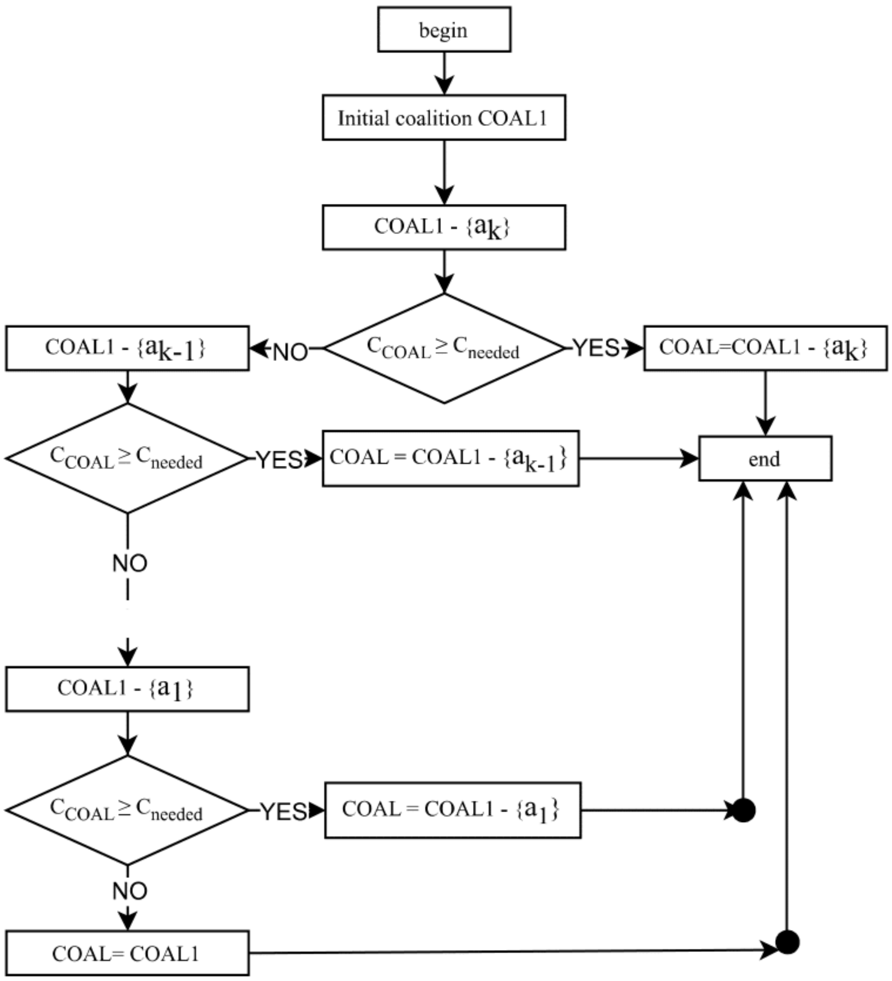


Fig. 3. Block diagram of the searching procedure.

The found coalition *COAL* can contain the redundant coalition members. Algorithm 3 checks the presence of redundant coalition members in the coalition and removes them if it finds some. Redundant coalition member is the coalition member that can be removed from the coalition without affecting execution of the coalition task (i.e., the task will be performed).

Algorithm 3.

Input: *COAL*, C_{needed}, $C_{a_1}, .., C_{a_k}$, K
int i,j,r
for $i \leftarrow 1$ to $K\text{-}1$
    $COAL = COAL - \{ a_i \}$
    $S = \sum_{j=1}^{K} C_{a_j} - \sum_{r=1}^{i} C_{a_r}$
        if $S < C_{needed}$
            $COAL = COAL\ U \{a_i\}$
            $COAL^* = COAL$
            return $COAL^*$
        elseif $i = K - 1$
            return a_K
Output: either $COAL^*$ or a_K

The Algorithm 3 starts with checking the coalition member that has the smallest capabilities. If after removing this coalition member the coalition is still able to perform a task then redundant coalition member is found, and the algorithm continues with checking the coalition member that has the second smallest capabilities. Otherwise, there are no redundant coalition members, and algorithm ends. Checking procedure continues until $(K - 1)$ coalition members are examined. If the algorithm doesn't output $COAL^*$ (i.e., checked and corrected coalition) then there is no need in coalition because one member a_K can perform a task.

3.6. The Third Situation

Finally, the third situation takes place when any coalition member can be substituted by alliance member. It occurs when the execution of Algorithm 2 results in the coalition for which $K=N$. In this case, $|T| = K$ and according to (13) and (18), $P_{CFT} = 1$. The problem may arise when several coalition members fail simultaneously or sequentially one after another. We leave this problem to future research.

4. Application of Proposed Metrics in a Sample Case

Let's consider the alliance $A=\{a_1,..,a_N\}$. To perform a task that requires capabilities C_{needed} , alliance members form a coalition. Generally, there can be formed D different coalitions, where

$$D = \sum_{i=2}^{N} C_i^N \quad (20)$$

It is assumed that in real-world scenarios, a smaller number of coalition members is more appropriate. In view of this, we consider only coalitions that contain a minimum number of coalition members. These coalitions have different capabilities and, consequently, they have different values of coalition fault tolerance. In the next example, we are going to show how coalition fault tolerance depends on the excess of coalition capabilities.

4.1. Calculation of P_{CFT} for Coalitions of Robots

Let's consider the coalition of three robots again (see Example 1). It is assumed that values L_1 and L_2 (see expressions (5, 7)) for Robot 1 are equal to 70% and 30% respectively. For Robot 2 are equal to 60% and 20%. For Robot 3 are equal to 50% and 10%. The conditional probabilities of different types of failure for all robots are as follows: P_M=0.85, P_P=0.1, P_S=0.03, P_C=0.02. Now we assume that load has weight of 150kg. It is evidently that three coalitions can perform the task, particularly $COAL^{(1)} = \{R_1, R_2\}$, $COAL^{(2)} = \{R_1, R_3\}$ and $COAL^{(3)} = \{R_1, R_2, R_3\}$.With regard to the assigned task, the third coalition is redundant since it includes members without which the task can be performed. We don't consider redundant coalitions. Thus, only coalitions $COAL^{(1)}$ and $COAL^{(2)}$ are evaluated. The results of calculations which use expressions (10) and (13) are given in Table 2.

Table 2. Results of Computations for Coalitions of Robots

Coalition	Coalition members' capabilities	coalition capabilities	Excess of coalition capabilities ΔC	Coalition tolerance to the failures of its individual members	Coalition fault tolerance
$COAL^{(1)}$	$C_{R1} = 100$ $C_{R2} = 80$	180	30	$P_{CFT}^{(R1)} = 0.85$ $P_{CFT}^{(R2)} = 0.437$	$P_{CFT} = 0.644$
$COAL^{(2)}$	$C_{R1} = 100$ $C_{R3} = 60$	160	10	$P_{CFT}^{(R1)} = 0.283$ $P_{CFT}^{(R3)} = 0.867$	$P_{CFT} = 0.575$

Thus, the probability that the first coalition will tolerate the failures in one of the robots is equal to 0.644, whereas for the second coalition this probability is equal to 0.575.

4.2. Example of Evaluation of Coalition Fault Tolerance

In this example, we are going to show how coalition fault tolerance depends on the excess of coalition capabilities.

Let $A=\{a_1,..,a_6\}$ is alliance consisting of six members. To perform the task, it is needed capabilities equal to C_{needed} = 80. Capabilities of alliance members are as follows: $C_{a_1} = 8, C_{a_2} = 10, C_{a_3} = 17, C_{a_4} = 26, C_{a_5} = 30,\ C_{a_6} = 40$. It is evident that there are no coalitions consisting of two members that are able to perform the task. Thus, the minimum number of coalition members is equals to three. Therefore, the number of possible coalitions is equal to $C_3^6 = 20$, but only four of them are able to perform the task. Particularly, $COAL^{(1)} = \{a_2, a_5, a_6\}$, $COAL^{(2)} = \{a_3, a_4, a_6\}$, $COAL^{(3)} = \{a_3, a_5, a_6\}$ and $COAL^{(4)} = \{a_4, a_5, a_6\}$. In the example under consideration, it is assumed that all coalition members are not substitutable. It is assumed that values of coalition members' failure functions V_1 and V_2, which define the minor, partial and substantial failures, are the same for all coalition members and equal to 0.6 and 0.2 respectively. The conditional probabilities of different types of failure are also the same for all coalition members and equal to P_M = 0.85, P_P = 0.1, P_S = 0.03 and P_C = 0.02. In real-world scenarios, these characteristics and probabilities can be set on the basis of the collected data and subsequent data processing.

Coalition tolerance to the failures of its individual members is computed according to (10), and coalition fault tolerance is computed according to (13). Results of the performed computations are presented in Table 3.

Table 3. Results of Computations of Coalition Fault Tolerance

Coalition	Coalition members' capabilities	coalition capabilities	Excess of coalition capabilities ΔC	Coalition tolerance to the failures of its individual members	Coalition fault tolerance
$COAL^{(2)}$	$C_{a3} = 17$ $C_{a4} = 26$ $C_{a6} = 40$	83	3	$P_{CFT}{}^{(a3)} = 0.748$ $P_{CFT}{}^{(a4)} = 0.493$ $P_{CFT}{}^{(a6)} = 0.323$	$P_{CFT} = 0.521$
$COAL^{(3)}$	$C_{a3} = 17$ $C_{a5} = 30$ $C_{a6} = 40$	87	7	$P_{CFT}{}^{(a3)} = 0.900$ $P_{CFT}{}^{(a5)} = 0.857$ $P_{CFT}{}^{(a6)} = 0.720$	$P_{CFT} = 0.825$
$COAL^{(4)}$	$C_{a4} = 26$ $C_{a5} = 30$ $C_{a6} = 40$	96	16	$P_{CFT}{}^{(a4)} = 0.951$ $P_{CFT}{}^{(a5)} = 0.932$ $P_{CFT}{}^{(a6)} = 0.900$	$P_{CFT} = 0.927$
$COAL^{(1)}$	$C_{a2} = 10$ $C_{a5} = 30$ $C_{a6} = 40$	80	0	$P_{CFT}{}^{(a2)} = 0$ $P_{CFT}{}^{(a5)} = 0$ $P_{CFT}{}^{(a6)} = 0$	$P_{CFT} = 0$

The function $P_{CFT} = f(\Delta C)$ is a discrete increasing function. The domain of this function is the set of integers that correspond to the values of excess of coalition capabilities. Function $P_{CFT} = f(\Delta C)$ can be approximated by function $f(Y)$, where

$$f(Y) = 1 - \frac{1}{\sum_{n=0}^{2} \frac{Y^n}{n!}}\ , Y = -b\Delta C, b = \frac{1}{4} \tag{21}$$

Function $f(Y)$ corresponds to the function $1 - e^{-b\Delta C}$, in which $e^{-b\Delta C}$ is represented by the first three terms of the power series (see Fig. 4). In case when there is only one objective, particularly coalition fault tolerance, the "best" coalition is the coalition that has maximum value of coalition fault tolerance. For the considered example such coalition is $COAL^{(4)}$. When there is also another objective such as efficient using coalition capabilities, the problem arises to solve bi-objective optimization. Bo-objective optimization consists in optimization of two competing objective functions.

4.3. Solving the problem of bi-objective optimization

Excess of coalition capabilities may have some deficiencies, e.g., it is costly, it may cause some coordination overheads, it may cause lack of alliance member capabilities for performing other tasks. The smaller the excess of coalition capabilities ΔC the more effective is the use of these capabilities E. The dependence E on ΔC can be expressed by objective function $E = f_1(\Delta C)$ that should be maximized. The first objective function is function $P_{CFT} = f_2(\Delta C)$, which also should be maximized. Both objective functions depend on the one variable, ΔC. When ΔC increases the objective function $P_{CFT} = f_2(\Delta C)$ also increases, while the second objective function $E = f_1(\Delta C)$ decreases (see Fig. 5). Thus, we have two conflicting objectives.

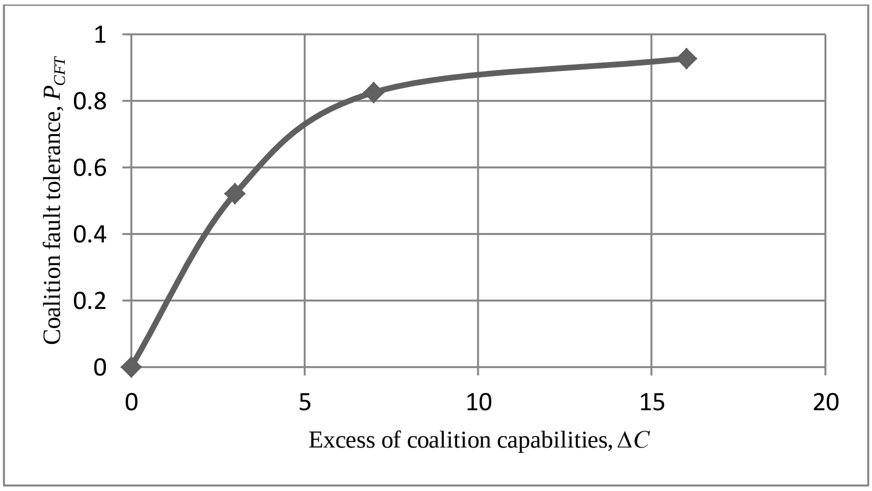


Fig. 4. Dependence of coalition fault tolerance on the excess of coalition capabilities.

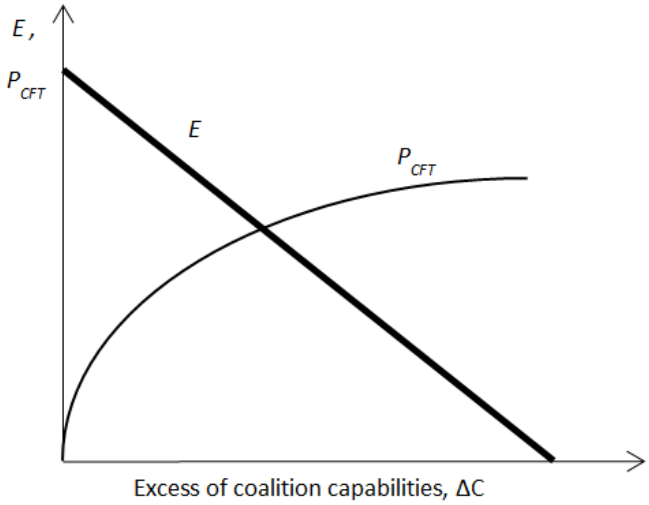


Fig. 5. Two conflicting objective functions.

Expression for defining the function $f_1(\Delta C)$ can be determined for each particular coalition. In this paper, we assume that function $f_1(\Delta C)$ is a linear discrete decreasing function. In Fig. 5, both objective functions are shown as continuous functions. In general, objective function $E = f_1(\Delta C)$ can be expressed by the decreasing functions such as, for example, $E = k_1 - k_2\Delta C$ or $E = \frac{k_1}{k_2\,(\Delta C)^n+1}$ or $E = k_1 e^{-k_2\Delta C}$, etc.

Assigning a value to variable ΔC is called a solution if it satisfies the imposed constraint $0 \leq \Delta C \leq \Delta C_{max}$. For each solution ΔC there is a point in the objective-space denoted by

$$f(\Delta C) = |f_1(\Delta C), f_2(\Delta C)\,|^T \tag{22}$$

In the given case, bi-objective optimization consists in finding the solution ΔC that optimizes the function $f(\Delta C)$. Objective space is two-dimensional space that can be represented in a plane by Y-axis and X-axis, which correspond to functions $f_1(\Delta C)$ and $f_2(\Delta C)$ respectively (see Fig. 6). In general, there can be many solutions, and consequently, many points in the objective space. In [24], the concept of domination between solutions was introduced. Solution ΔC_i dominates solution ΔC_j if $f_1(\Delta C_i) > f_1(\Delta C_j)$ and $f_2(\Delta C_i) > f_2(\Delta C_j)$. It should be noted that either $f_1(\Delta C_i) \geq f_1(\Delta C_j)$ or $f_2(\Delta C_i) \geq f_2(\Delta C_j)$ can take place but not both. Solution ΔC_P is called Pareto optimal if there is no solutions that dominates solution ΔC_P. The set of all Pareto optimal solutions is called the Pareto front.

If we start viewing the points, which correspond to Pareto optimal solutions, from the point that is placed leftmost, then each subsequent observation point is located further to the right and lower. In the example, the following values of variable ΔC were considered: $\Delta C_1 = 3$, $\Delta C_2 = 7$ and $\Delta C_3 = 16$. These are the values that variable ΔC can take within the defined range $[0, \Delta C_{max}]$.

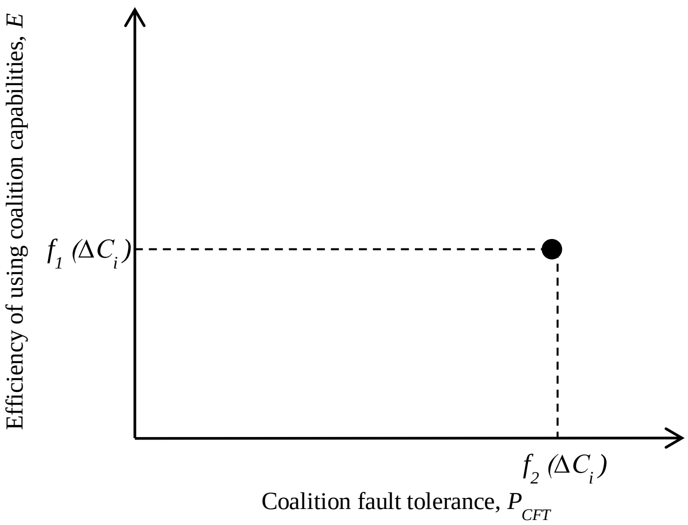


Fig. 6. Two-dimensional objective space.

Two functions was considered to express the efficiency E, particularly

$$E = 2 - \frac{\Delta C}{10} \tag{23}$$

and

$$E = 1 - \frac{\Delta C}{100} \tag{24}$$

The first function (23) can be used when excess of coalition capabilities has substantial impact on efficiency E. The second function (24) can be used when excess of coalition capabilities has minor impact on efficiency E. In case of using the first function, the objective function $f_1(\Delta C)$ has the following values for the solutions ΔC_1, ΔC_2 and ΔC_3 : $f_1(\Delta C_1) = 1.7$, $f_1(\Delta C_2) = 1.3$ and $f_1(\Delta C_3) = 0.4$. In order to select one solution from Pareto optimal set, which includes solutions ΔC_1, ΔC_2 and ΔC_3 , the compromise programming or gain-to-loss ratio [24] can be used. Gain-to-loss ration method is based on the calculation of the maximum units of loss in the value that one objective has due to one unit gain in the value of the other objective in a bi-objective problem. We consider that compromise programming is more acceptable for our task. Compromise programming defines an infeasible ideal point Z (see Fig. 7).

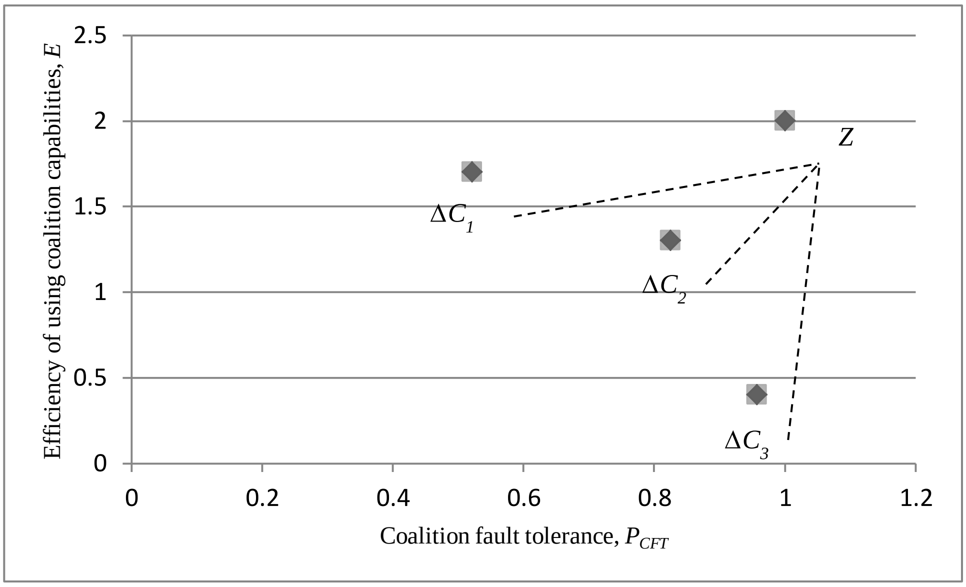


Fig. 7. The infeasible ideal point Z.

Point Z is formed by combining the individual best values of each of the objective functions. The coordinates of ideal point Z are represented by ordered pair (X, Y), where $X = f_2^{max}(\Delta C)$ and $Y = f_1^{max}(\Delta C)$. It is necessary to compute for each point, which represents the solution, its distance from ideal point Z. This can be done by using Manhattan distance (or $L1$-norm), i.e.,

$$dist(P_i, P_j) = |f_2(P_i) - f_2(P_j)| + |f_1(P_i) - f_1(P_j)| \tag{25}$$

When function (23) is used, the following values of distances are obtained: $dist(\Delta C_1, Z) = 0.779$, $dist(\Delta C_2, Z) = 0.875$ and $dist(\Delta C_3, Z) = 1.673$. Thus, the best solution is solution ΔC_1 (i.e., $COAL^{(2)} = \{a_3, a_4, a_6\}$) since the distance between the point that represents solution ΔC_1 and ideal point Z is the smallest. In case of using the function (24), the following values of distances are obtained: $dist(\Delta C_1, Z) = 0.509$, $dist(\Delta C_2, Z) = 0.245$ and $dist(\Delta C_3, Z) = 0.233$. In the given case, the best solution is solution ΔC_3 (i.e., coalition $COAL^{(4)} = \{a_4, a_5, a_6\}$).

The complexity of bi-objective optimization depends on the number of solutions that constitute the Pareto front. The number of solutions is defined by the number of possible coalitions that can perform a mission. This number of possible coalitions can be evaluated as follows.

Let's consider an alliance of N members. At the first step, it is needed to sort the alliance members in ascending order according to their capabilities. The sorting procedure has complexity $O(NlogN)$. At the second step, it is needed to exclude the alliance members that have capabilities more than required capabilities C_{needed}. It is necessary because each such member doesn't need a coalition since it can perform a mission by itself. This excluding procedure has complexity $O(N)$. Total number of excluded members is denoted as R. At the next step, it is needed to determine all possible coalitions that can be formed by $(N\text{-}R)$ alliance members and to choose only those coalitions that satisfy condition (3). The number of coalitions that must be examined can be calculated according to (20). This procedure has complexity

$O(N_1^{\frac{N_1}{2}})$, where $N_1=N\text{-}R$. We consider that coalitions with minimum number of members are preferable (when taking into account coordination and communication costs). In view of this, we change the searching procedure so that first will be examined all coalitions with two members. If at least one of such coalitions satisfies condition (3) then the searching procedure will not examine the coalitions which have more than two members. Otherwise, all coalitions with three members will be examined. If at least one of the coalitions with three members satisfies condition (3) then the searching procedure will not examine the coalitions which have more than three members. The same analysis is applied to the coalitions with larger number of members. Such arranged searching procedure allows reducing the number of examined coalitions approximately by two times. Since the searching procedure has polynomial complexity, it could be used when the number of alliance members doesn't exceed a few dozen.

5. Conclusion

In this paper, a failure function for coalition members was proposed. This function takes into account different types of failure of a coalition member such as minor, partial, substantial and complete failure. By using coalition members' failure functions, a metrics for measuring coalition fault tolerance was proposed. As distinct from the existing metrics (such as FTC and maximum allowable number of coalition member failures), the proposed metrics takes into account different types of coalition member failures and also the possibility to continue the work without replacing the failed coalition member (in case of minor, partial and substantial failures).

In the paper, we have considered the situation when only one coalition member can fail at a time. Such situation is the most probable. The situations when more than one coalition members fail at a time have essentially lower probability. Nevertheless, they should be studied as a separate research. We are planning consideration of this problem in the future. In the future, we also plan to relax the restrictions on the dimension of a coalition member's capability vector.

As distinct from multi-agent systems, in which the autonomous agents are self-interested, the alliance members considered in the paper are unselfish. In view of this, the procedure of formation of coalitions by alliance members is significantly distinct from the procedure of coalition formation used in multi-agent systems.

For the case when substitution of the failed coalition members is possible, the algorithm of formation of a coalition by alliance members was developed. The algorithm searches the coalition that has maximum value of coalition fault tolerance. The algorithm provides an approximate solution to the problem. Finding the exact solution is a difficult task with high complexity.

For the case when substitution of the failed coalition members is impossible or very complicated, the method for comparing the possible coalitions and choosing the coalition that has the largest value of coalition fault tolerance was proposed. In the paper, the value of coalition fault tolerance is used for solving the problem of bi-objective optimization, in which coalition fault-tolerance is one of the objectives.

References

[1] M. Pechoucek, V. Marik, J. Barta, "A knowledge-based approach to coalition formation," In IEEE Intelligent Systems, vol. 17, No. 3, pp. 17–25, 2002, doi: 10.1109/MIS.2002.1005627.

[2] O. Shehory, S. Kraus, "Method for task allocation via agent coalition formation," Artificial Intelligence, vol. 15, No.3, pp. 218-251, 1998, doi: 10.1016/S0004-3702(98)00045-9.

[3] J. Koning, "Algorithms for translating interaction protocols into a formal description," IEEE International Conference on Systems (SMC-99), Tokyo, 1999.

[4] B. Bauer, J. Muller, J. Odell, "An extension of UML by protocols for multi-agent interaction," International Conference on Multi-agent Systems (ICMAS'00), Boston, USA, pp. 207-214, 2000.

[5] J. Fiser, V. Mashkov, "Alliance and coalition formation," Int. Journal of Applied Compute Science, vol. 18, No. 1, pp. 19-38, 2010.

[6] A. Ledmi, H. Bendjenna, S. M. Hemam, "Fault tolerance in distributed systems: A survey," 3th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Algeria, 2018, doi: 10.1109/PAIS.2018.8598484.

[7] A. Sari, M. Akkaya, "Fault tolerance mechanisms in distributed systems," International Journal of Communications, Network and System Sciences, vol. 8, No. 12, 2015, doi: 10.4236/ijcns.2015.812042.

[8] V. Mashkov, "Restricted alliance and coalition formation," Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Beijing, pp. 329-332, 2004, doi: 10.1109/IAT.2004.1342963.

[9] O. Shehory, K. Sycara, S. Jha, "Multi-agent coordination through coalition formation," Lecture Notes 1365, pp. 143-154, 2006, doi: 10.1007/BFb0026756.

[10] R. Glinton, P. Scerri, K. Sycara, "Agent-based sensor coalition formation," Proceedings of 11th International Conference on Information Fusion (FUSION'08), pp. 173-179, 2008.

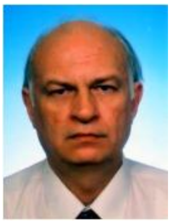
[11] C. Li, U. Rajan, S. Chawia, K. Sycara, "Mechanisms for coalition formation and cost sharing in an electronic marketplace," Proceedings of the 5th International Conference on Electronic Commerce, ICEC, pp. 68-77, 2003, doi: 10.1145/948005.948015.

[12] V. Mashkov, "Tasks allocation among agents of restricted alliance," Proceedings of 8th IASTED International Conference on Intelligent Systems and Control, ACTA Press, Cambridge, MA, USA, pp. 13-18, 2005.

[13] M. Klush, A. Gerber, "Dynamic coalition formation among rational agents," Intelligent Systems, IEEE, vol. 17, No. 3, 2002, doi: 10.1109/MIS.2002.1005630.

[14] V. Mashkov, J. Barilla, P. Simr, J. Bicanek," Modeling and simulation of coalition formation," SIMULTECH 2015, 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, 2015, pp. 329-336.
[15] A. Caparros, A. Hammoudi, T. Tazdait," On coalition formation with heterogeneous agents," Nota di Lavoro, No. 70, 2004, available on: https://hdl.handle.net/10419/117951.
[16] T. Tang, L. E. Parker, "Coalescing multi-robot teams through AsyMTRe: A formal analysis," In Proceedings of IEEE International Conference on Adv. Robot, pp. 817-824, 2004.
[17] L. Vig, J. Adams," Multi-robot coalition formation," IEEE Transactions on Robotics, vol. 22, No. 4, pp. 637-649, 2006, doi: 10.1109/TRO.2006.878948.
[18] S. Sebo, B. Stoll, B. Scassellati, M. F. Jung, "Robots in groups and teams: A literature review," Proceedings of the ACM on Human Computer Interaction, vol. 4, issue CSCW2, Article no. 176, pp. 1-36, 2020, doi: 10.1145/3415247.
[19] B. P. Gerkey, M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," Int. J. Robot, Res. Vol. 23, No. 9, pp. 939-954, 2004, doi: 10.1177/0278364904045564.
[20] V. Mashkov, "Coalition formation with unreliable agents," Proceedings of the Forth IASTED International Conference Knowledge Sharing and Collaborative Engineering, St. Thomas, US Virgin Islands, pp. 138-143, 2006.
[21] A. Fedoruk, R. Deters, "Improving fault-tolerance by replicating agents," Proceedings of the First International Conference on Autonomous Agents and Multi-agent Systems, AAMAS 2002, pp. 737-744, 2002, doi: 10.1145/544862.544917.
[22] F. Carreras, A. Magana, "Stability for coalition structures in terms of the proportional partitional Shapley value," Springer, Homo Oeconomicus, 2024, doi: 10.1007/s41412-024-00143-8.
[23] A. Piazza, J. P. Torres-Martinez, "Coalition stability in matching problems with externalities and random preferences," Games and Economic Behavior, 2024, doi: 10.1016/j.geb.2023.12.002.
[24] K. Deb, "Multi-objective optimization using evolutionary algorithms," New York, NY USA, John Wiley & Sons Inc. 2001, 536 pages.
[25] M. Agarwal, N. Kumar, L. Vig, "Non-additive multi-objective robot coalition formation, " Expert Systems with Applications, vol. 41, Issue 8, pp. 3736-3747, 2014, doi: 10.1016/j.eswa.2013.11.044.

Authors' Profiles



Viktor Mashkov is Doctor of Science in Engineering, Docent in Cybernetics. He has been teaching at Universities since 1984. From 2007 to present, docent at the University of J. E. Purkyne in Usti nad Labem (Czech Republic). Supervised four PhD students and advised two DSc. Theses. Participated in more than ten International Conferences of IEEE, IASTED, ICSOFT, HIS that were held in Hungary (Budapest-1996, Spain (Malaga-2002, Barcelona-2007), Chile (Santiago-2002), Austria (Innsbruck-2003), China (Beijing-2004), the USA (Boston-2005), Virgin Islands (St. Tomas-2006), Cyprus (Pafos-2011), Czech Republic (Prague-2014), France (Colmar-2015), etc., where his contribution papers were published in the proceedings of the conferences. Member of the program committees of conferences, such as Central and Eastern European Conference on Multi-agent Systems CEEMAS, Intelligent Decision Making Systems and Applied Aspects of Information Technologies ISDMIT, etc. He is author of more than fifty science papers as well as several books and several chapters in books on the subject of system level self-diagnosis. His major research focuses on self-diagnosis at system level, multi-agent systems, alliance and coalition formation.

How to cite this paper: Viktor Mashkov, "Evaluation of Coalition Fault Tolerance", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.18, No.1, pp. 42-55, 2026. DOI:10.5815/ijieeb.2026.01.03