Evaluation of Agents' Management Impact on Performances in Coalition-Based Cooperation

Kaouther Bouzouita, Wided Lejouad Chaari and Moncef Tagina
COSMOS Research Laboratory
National School of Computer Studies
University of Manouba
Manouba, Tunisia
Email: {kaouther.bouzouita, wided.chaari, moncef.tagina}@ensi-uma.tn

***Abstract*—We study within this paper the effect of agents' management on the global performance of a multiagent system (MAS). Cooperative systems working under coalition formation strategies have been considered in this work. Our evaluation approach uses representation tools of the coalition formation process in addition to mathematical modeling techniques using utility functions. An empirical study has been led to validate this approach and present a benchmarking of three different cooperative strategies: *Binary Max-Sum*, *Distributed Stochastic Algorithm* and *Greedy* algorithm. Each strategy aims at solving the problem of task allocation in *search and rescue* scenarios.**

***Index Terms*—MAS evaluation, cooperation, coalition formation, coordination, utility, performance.**

I. Introduction

Many cooperative strategies may be adopted by multi-agent systems (MASs) for the achievement of complex tasks. These tasks either need many agents to be done, or can be executed individually by one agent. In the latter case, a group of agents usually performs a better work. When the system has a limited number of agents, tasks should be treated in order of priority. Besides, an efficient task allocation strategy has to be established. Otherwise, if resources are not very well managed for example, side effects may rise, like losses in time and efficiency. A compromise has to be done between agents good management and overall system's performance. This challenge is not easy to take, especially in dynamic environments where physical agents constantly have to adapt their decisions to varying ambient conditions. Many constraints should be taken into account in such environments. It is certainly beneficial for tasks to get many assigned agents dedicated for their execution. But this may negatively affect other tasks due to their luck of agents. Such an issue is not simple to address in the case of distributed systems, where agents have partial view of their environment. A balanced repartition of the agents will be harder to achieve. For all these reasons, we find it necessary to highlight the effect of agents management on the system's global performance by establishing a utility-based evaluation tool that may indicate eventual organizational weaknesses of the system. Our tool uses a central evaluation technique that compares different coalition strategies.

978-1-5090-5795-5/17/$31.00

A. Related Work

We find in the literature different evaluation approaches of the multiagent systems, such as [1], [2] and [3]. Multiple aspects of the agents society were studied within these works, such as performance, communications, intrinsic characteristics of agents and design methodology of the system. Other works were interested in studying the organizational aspect of such systems, like [4] where the authors used many indicators inspired from the graph theory. After an analogy with the agents' society structure, many independent aspects could be assessed such as centrality in a network of agents. In [5], the proposed approach considers three main evaluation criteria of the agents organization: type of agents, their specialties and the workflow where they act. The adopted technique considers the MAS structure at different complexity levels. It offers however evaluation criteria that are not completely independent, unlike those studied within our work.

The majority of evaluation approaches consider different aspects separately. We try through this work to combine two principle aspects and see a possible relationship between them.

B. Motivation

We chose to consider for this work cooperative agents capable of forming coalitions in order to reach some goal. We remind that a coalition is a group of agents, formed dynamically to achieve some goal of the system. It disappears once its objective is reached. The motivation behind this work is to emphasize the eventual relation between their repartition and the whole system's performance. Thus, such a study may give us a diagnosis tool that analyses the organizational structure of the system and explains, even partially, any deterioration of the MAS performance. In fact, since the considered multi-agent system in our study consists of agents using a decentralized algorithm for deciding which target to choose, there may be some imbalance in agents' assignment, which favors some goals over others. This naturally has an

impact on results, especially in dynamic environments where goals characteristics often change and agents decisions have an essential effect on future states of the environment. In cooperative game theory for instance, coalitions formation process is said to be “super-additive”, “additive” or “sub-additive” when it is a matter of studying its effect on agents performance. In fact, coalition formation may respectively be positive, neutral or negative for the performance of the corresponding goal achievement. In our work, we try to study that effect by considering the additivity of each coalition on all the targets of the system, not only on the considered target, so as to have a more significant judgment on the efficiency of coalition formation process. In fact, especially in the case of physical agents, it is often beneficial to have many agents cooperating for the execution of one goal, if their efforts are additive. But if we consider all targets, and find that some of them are missing agents, that will deteriorate the performance in another side of the system, and may have an impact on global system’s performance, which can then be explained by a bad repartition of agents. A parallel analysis of system’s functional adequacy is then required to see the impact of agents repartition on system’s achievements.
We established an evaluation process based on a centralized tool for evaluating cooperative MAS performance by considering agents’ repartition and its impact on goals achievement. In addition to the assessment of organizational quality, a link with the functional adequacy of the system can then be established in order to find, if any lack of performance is detected, a possible explanation in the repartition of agents. So our contribution, when compared to the studied works in the literature, evaluates cooperative agents by considering different aspects, i.e. organizational an functional ones, and highlights a causal relationship between them, in order to assess the system’s performance.

C. MAS Used for The Validation of the Approach

The RoboCup rescue Simulation System (*RCSS*) [6] has been used for the validation of our apporach (see Figure 1). It is an urban search and rescue system. Further details about that simulation platform can be found in [7]. Within the same paper, a summary of the main open research problems has been elaborated. We study the cooperation problem in the *RCSS* using two classes of agents: Fire brigades and police patrols who interfere after a disaster occurring in a city and are intended to respectively extinguish fires and remove blockades in order to minimize the damages affecting that city. Many works have considered such a system such as [8] and [9], especially in studying its cooperative aspect. In [10], authors used a cooperative approach based on inter-team coordination.

A benchmark, called *RMASBench* developed by Pujol et al. [11], in order to compare different strategies of the RCSS system: coordinating strategies and non-coordinating ones. Utility functions have been used too in order to assess some aspects of the Distributed Constraints Optimization Problem (DCOP), such as the number of required agents per target. Constraint violations are then raised when the number of

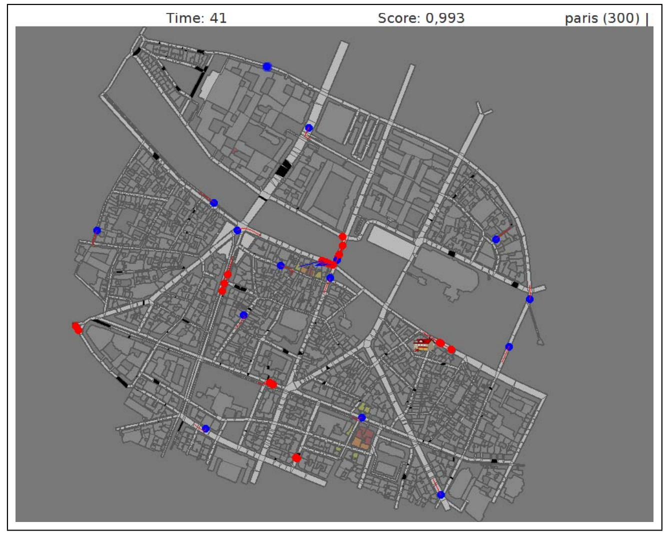


Fig. 1: A running simulation of the *RCSS*. Agents are represented by dots. Buildings on fire are colored and blocked roads are blacked out.

assigned agents exceeds that number. This operation has two major disadvantages. First, it considers exceeding numbers of agents as a negative point, where, in many cases it can be beneficial for the corresponding goal achievement and thus for the whole system’s performance when no lack of agents is detected elsewhere. Second, this computation is only declared to the benchmark user, and is not used for the system’s global evaluation. We try, in our work, to treat these two issues. A score function, implemented within the benchmark, estimates the percentage of the buildings kept safe in the city and thus reflects the performance of the MAS in the rescue operation. This last value will be used for the validation of our utility function to make a matching between agents repartition, studied by our evaluation approach, and its impact on the functional adequacy of the system, represented by the score function.

D. Paper Organization

The remainder of the paper is organized as follows. Section 2 presents the evaluation approach and its different steps. Section 3 is dedicated to the empirical validation of our work. Section 4 presents an interpretation of the main results followed by a discussion in section 5. Section 6 concludes and presents future work.

II. Evaluation Approach Description

Our evaluation approach has two main aspects. First, it offers compact representation tools to present the states of the environment, so as to give a compact clear view of it, and thus allows an easier study of the problem. Second, modeling tools, based on utility functions, are used to compute performance measures allowing then an overview of the system’s efficiency.

A. Representation Tools

Agents of the studied MAS evolve in a highly dynamic environment. They are autonomous agents trying at each iteration to adapt their decision to the perceived changes. If we try to represent all possible states of the environment in order to evaluate them by deducing a personalized utility value, we will get an infinite set of possible execution scenarios. Thus, utility estimation will not be feasible. We opt then for the use of a finite-state automaton (FSA) representing all states relevant to the utility computation, i.e. states relative to the coalition formation process in the case of the organizational evaluation. Such a tool allows a compact representation of the studied MAS.

Transitions correspond to significant events that make the system move from one state to another and may have an impact on the utility variation. Such a method has been used in [2] where the evaluation approach consisted in assessing the system's output by considering its agents' communications.

We defined the automaton states by analyzing the coalition formation process during one iteration. Such a process can be divided into three main steps:

1) Beginning of coalition formation: This step represents the grouping of a number of agents into one coalition for reaching one target. This number may be inferior to the required number of agents for the specified target. It may also be equal to it. In the first case, we increment the number of missing agents to use it in the evaluation phase.
2) Superfluous agents assignment (optional): Some coalitions may have a number of agents that exceeds their real needs. In our approach we designate such entities by *superfluous agents*.
3) Ending of coalition formation: Such a state is detected when no more agents are assigned to a coalition at the end of the considered iteration.

Figure 2 presents the FSA associated to coalition formation process.

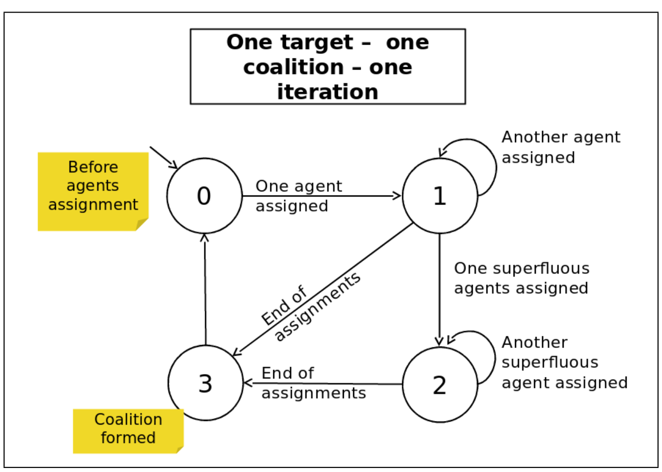


Fig. 2: Representation of system states corresponding to one coalition formation for one target reaching at one iteration.

We consider all formed coalitions at the end of the timestep to deduce a utility value evaluating instant global agents repartition. Two values are computed relatively to all coalitions by considering differences between their respective "number of affected agents", which is the number of agents in the coalition, and "number of required agents" corresponding to that of agents required for the extinction of the aimed fire. The deduced values are:

- *Number of superfluous agents*: The total number of agents exceeding the required one in all coalitions. We summarize the numbers of superfluous agents in each coalition to get this value;
- *Number of missing agents*: The total number of missing agents for the extinction of fires. Numbers of eventually missing agents in each coalition are summarized.

  Utility values are computed as follows: If some agents are missing in one coalition, we cannot decrease the utility of the system unless some others are superfluous in other coalitions. A problem in **instant global agents' repartition** is then detected.
- *Number of unassigned fire Agents*: This parameter refers to the number of agents having no assigned target in the considered iteration. The computation of that parameter interferes in the computation of the instant global utility if there are missing agents in some coalitions. This case corresponds to a problem in **agents' exploitation**.

After considering all iterations, and computing utilities corresponding to agents repartition and exploitation, we deduce the final global utility of the system which is an average value of all instant global utilities.

B. Modeling Tools

The second part of our evaluation approach consists in measuring penalties associated to agents organization. A mathematical model has thus been established in order to assess formed coalitions during the execution of the studied system. We start by estimating coalitional utility, also called instant global utility, which evaluates agents repartition and exploitation in every iteration. Then we compute the final global utility of the whole execution.

Besides the organizational aspect, our contribution considers the functional adequacy of the MAS. Such an aspect is represented by a score function in the system considered for the validation of our approach. It is given by the used benchmarking platform: It represents the percentage of the city's buildings not damaged by fires during the execution.

By matching the two scales, repartition penalty and execution score, we can deduce the relation between agents good management and their efficiency in reaching the system's goals.

1) Coalitional Utility Function: This function represents the penalty of coalitions formation during one iteration i. We use the following terminology:

- $num^{c}_{FireAgents}$: is the number of fire agents in a coalition c;

- $num^{c}_{RequiredAgents}$: is the number of required fire agents in a coalition c;
- $num_{FireAgents}$: The number of fire agents in the system;
- $num^{i}_{SuperfluousAgents}$: The total number of agents exceeding the required one in all coalitions in iteration i;
- $num^{i}_{MissingAgents}$: The number of missing agents for the extinction of targets in iteration i;
- $num^{i}_{UnassignedFireAgents}$: Number of agents having no assigned target in iteration i.

For simplicity reasons, the aforementioned parameters will be designed respectively by: num^{c}_{FA}, num^{c}_{RA}, num_{FA}, num^{i}_{SA}, num^{i}_{MA} and num^{i}_{UA}.

$num^{i}_{SuperfluousAgents}$ and $num^{i}_{MissingAgents}$ will be computed as follows:

For every coalition c, if $(num^{c}_{FA} > num^{c}_{RA})$ *then* num^{i}_{SA} *is incremented by* $(num^{c}_{FA} - num^{c}_{RA})$ *else* num^{i}_{MA} *is incremented by* $|num^{c}_{FA} - num^{c}_{RA}|$

Two cases should be distinguished when computing the utility of agents repartition in an iteration i:

1) **First case**: Number of unassigned agents is null. In other words, all fire agents are assigned to targets (see Formula 1).

$$U^{i}_{AgentsRepartition} = Penalty^{i}_{coalitionFormation} = \begin{cases} \dfrac{min(num^{i}_{MA}, num^{i}_{SA})}{num_{FA}} \\ \text{if } ((num^{i}_{MA} > 0) \wedge (num^{i}_{SA} > 0)), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

$\dfrac{min(num^{i}_{MA}, num^{i}_{SA})}{num_{FA}}$ is the ratio of misplaced agents to the total number of agents in the system.

2) **Second case**: There are agents that are not assigned to any target. If all agents have no assigned target, that means, for the considered system, that all fires have been extinguished, and the end of the execution has been reached. Otherwise, if only some agents have no assigned target, that corresponds to a bad exploitation of system's agents, since these agents could have joined coalitions to raise their efficiency in fire extinction (see Formulas 2 and 3).

$$U^{i}_{AgentsRepartition} = Penalty^{i}_{coalitionFormation} = \begin{cases} Penalty^{i}_{AgentsExploitation} & \text{if } num^{i}_{UA} < num_{FA}, \\ 0 & \text{if } num^{i}_{UA} = num_{FA}. \end{cases} \tag{2}$$

where

$$Penalty^{i}_{AgentsExploitation} = \begin{cases} \dfrac{min((num^{i}_{UA} + num^{i}_{SA}), num^{i}_{MA})}{num_{FA}} & \text{if } num^{i}_{MA} > 0, \\ \dfrac{num^{i}_{UA}}{num_{FA}} & \text{if } num^{i}_{MA} = 0. \end{cases} \tag{3}$$

2) Global Utility Function: This function represents the utility of the whole execution. It is an average value of instant utility measures (see Formula 4).

$$U = \frac{\sum_{i} U^{i}_{AgentsRepartition}}{num_{iterations}} \tag{4}$$

III. Validation of the Evaluation Approach

In this section we describe the sets of experiments that were launched for the validation of our approach. Then we present their main results and interpretations.

A. Description of Experiments

We run many experiments using well-known state-of-the-art algorithms: *Binary Max-Sum* (*BMS*), *Distributed Stochastic Algorithm* (*DSA*) and *Greedy*, since all of them use agents cooperation. *BMS* and *DSA* use different coordination strategies whereas *Greedy* algorithm uses no agents coordination. We can, through this study, compare the different task allocation techniques in terms of performance and also evaluate the impact of coordination on such an aspect:

1) ***BMS*** is an extension of the *Max-Sum* (*MS*) allocation strategy that uses binary variables. *Max-Sum* is a decentralized algorithm that aims at solving social welfare maximization problems [12]. It provides an approximate solution to the problem of maximizing a function (team utility in our case) that decomposes additively as a sum of functions, also referred to as *factors*. A bipartite graph is used for exchanging messages between variables (fire assignments variables) and factors to find out the best assignment for each variable [10]. *MS* and *BMS* use inter-team agents coordination.
2) ***DSA*** is a local search algorithm executed in parallel by agents. It uses a stochastic parameter DSA_p that controls the amount of parallelism allowed during the execution of the algorithm. Agents choose some fire and communicate their choice to other agents. Each brigade re-evaluates its choices depending on others', and may change its target depending on DSA_p value, which represents the probability of changing the target. In the launched experiments, we kept the default value of the RMASBench, i.e. $DSA_p = 0.1$. Agents, at the beginning of each iteration, choose the target of the last iteration. This algorithm has the advantage of requiring low computation and communication efforts [10].
3) ***Greedy*** is a simple allocation method that does not use any agents coordination. Each agent chooses one target by considering the one that maximizes his own individual utility.

A standard scenario of the *RoboCup Rescue* competition has been considered. It uses a map of Paris. Twenty experiments of each selected algorithm were launched.

B. Results

Four sets of experiments were run. We generate penalties and scores plots corresponding to each one of them.

1) First Set of Experiments: We started our experimental study by considering one execution at a time. Each figure shows the evolution of the score (dotted line) and the average value of penalties computed so far (continuous line). We used a translation of 0.8 of the penalties values to make the comparison of the plots possible. Different cases of system's performance could then be distinguished (see Figure 3).

We can see in the three examples of the Figure 3 that there is some proportionality between system's performance (score value) and the inverse of its agents' repartition penalties.

In order to have a global overview of the MAS performance, we computed the average value of each execution's penalties, and generated a plot that compares these values to final scores, which constitutes the second set of experiments.

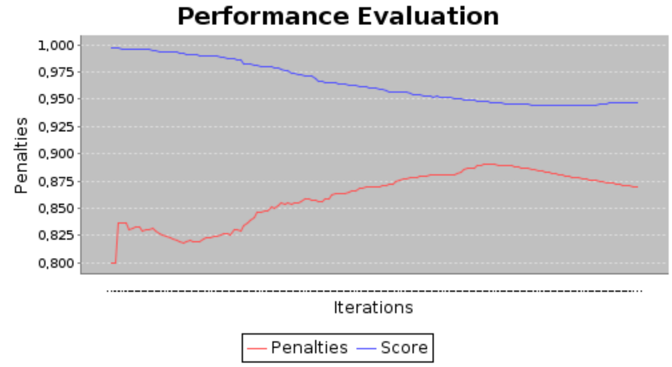
2) Second Set of Experiments: We launched many times the simulation in order to have different values of average execution penalty and compare it to the corresponding score value. We can easily detect many obvious similarities between progressions of the score and the penalty's inverse. They are quiet proportional. This gives a concrete proof of the existence of a relationship between agents' repartition and the global performance of a system.

Figure 4 presents scores and penalties plots relative to different executions.

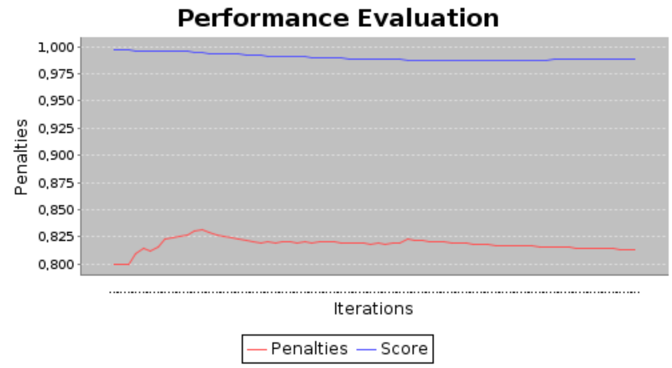
3) Third Set of Experiments: We launched a third set of experiments using *DSA* algorithm in order to compare the two coordinating strategies (see Figure 5).

4) Fourth Set of Experiments: Another set of experiments using a non-communicating strategy, *Greedy* algorithm, has been launched (see Fig 6). When comparing with the aforementioned strategies, impact of coordination between agents could then be studied.

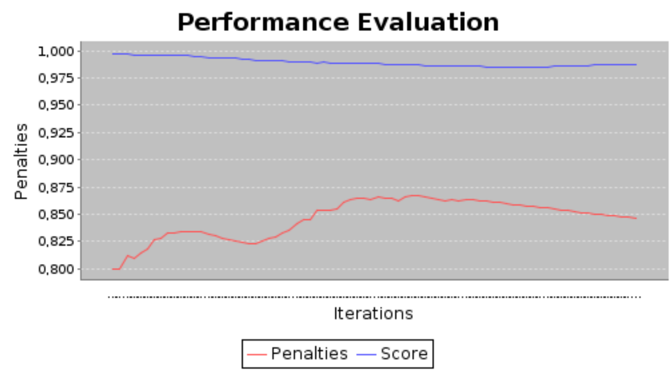
5) Main Results Summary: In Table I we present the highest, lowest and average value of scores and penalties of executions relative to the considered algorithms. We have established such a summary in order to compare results of the strategies *BMS* and *DSA* using agents coordination and to see its eventual impact by comparison with a non-communicating algorithm, i.e. *Greedy* algorithm.



(a) One execution.



(b) A second execution.



(c) A third execution.

Fig. 3: Penalties and score plots in three different executions of the *RCSS* using the *BMS* algorithm.

IV. Results Interpretation

We present in this section main interpretations of the obtained results.

A. Validation of the Utility Function as an Evaluation Tool of Agents' Organization

We remind that social welfare is represented, in our work, by the value of the score function which computes the percentage of the number of buildings that were not damaged by fires, or that have been saved. This value reflects the MAS efficiency regarding fire extinction problem.

We can see that such a value is decreasing at different rates in Figure 3. When compared to the plot of the system's penalties, we can see that it is inversely proportional to it in most cases. The more agents are badly distributed, the more the score decreases.

Second set of experiments gives a more global overview on such a relationship by considering multiple experiments. Average penalties value allows us to have a general estimation of agents repartition during one execution of the system. We study the influence that it may have on the final reached

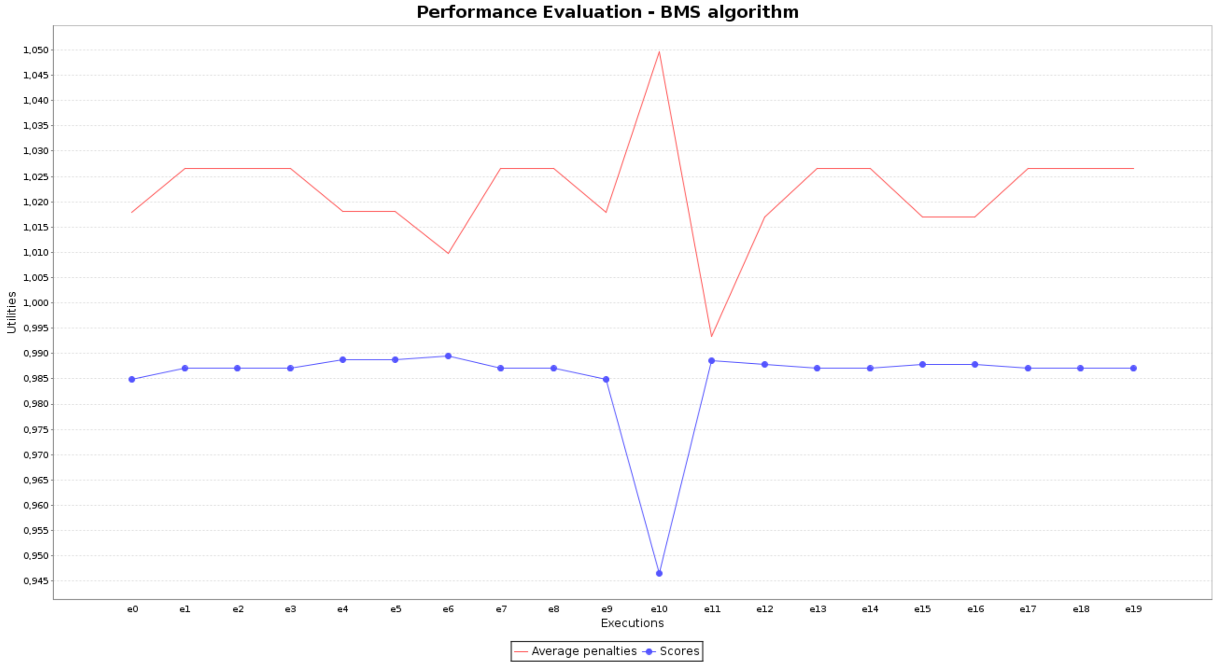


Fig. 4: Average penalties vs corresponding scores in 20 executions of the *RCSS* using *BMS* algorithm.

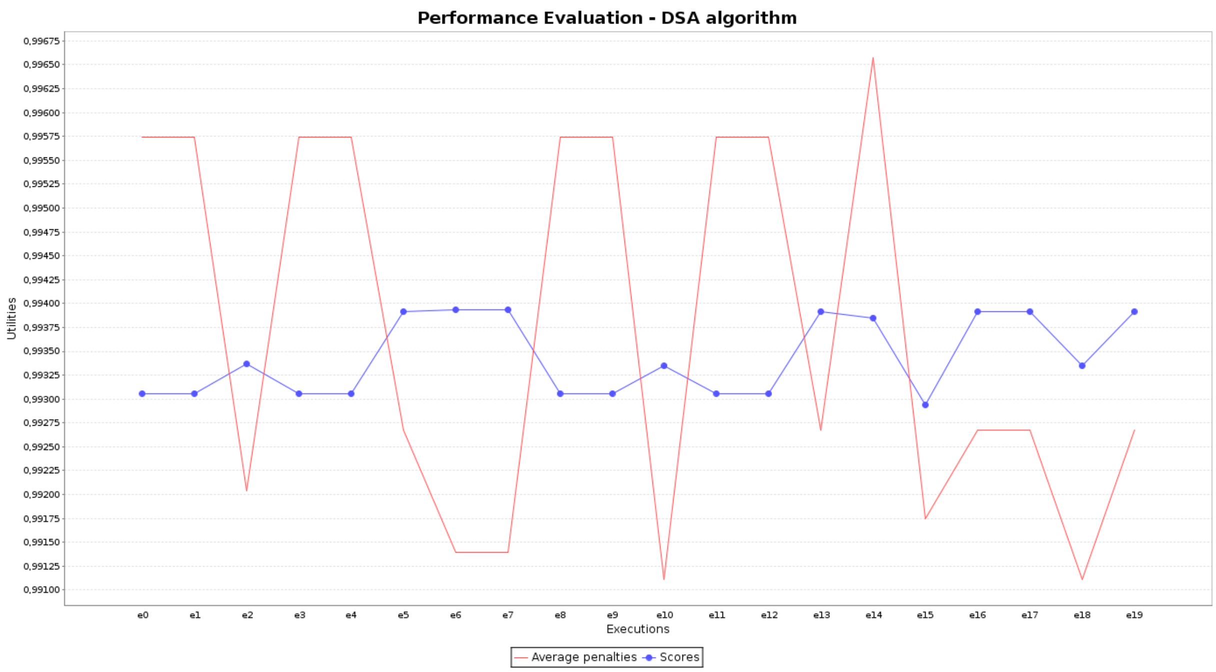


Fig. 5: Average penalties vs corresponding scores in 20 executions of the *RCSS* using *DSA* algorithm.

TABLE I: Scores and Penalties Overview of Three Algorithms: *BMS*, *DSA* and *Greedy*

Algorithm	Highest of average penalties	Lowest of average penalties	**Average of average penalties**	Lowest score	Highest score	**Average score**
DSA	0.0166	0.0111	**0.0137**	0.9929	0.9939	**0,9934**
BMS	0,0696	0,0134	**0,0421**	0,9465	0,9895	**0.9853**
Greedy	0,1002	0,0196	**0,0556**	0,9119	0,9700	**0,9576**

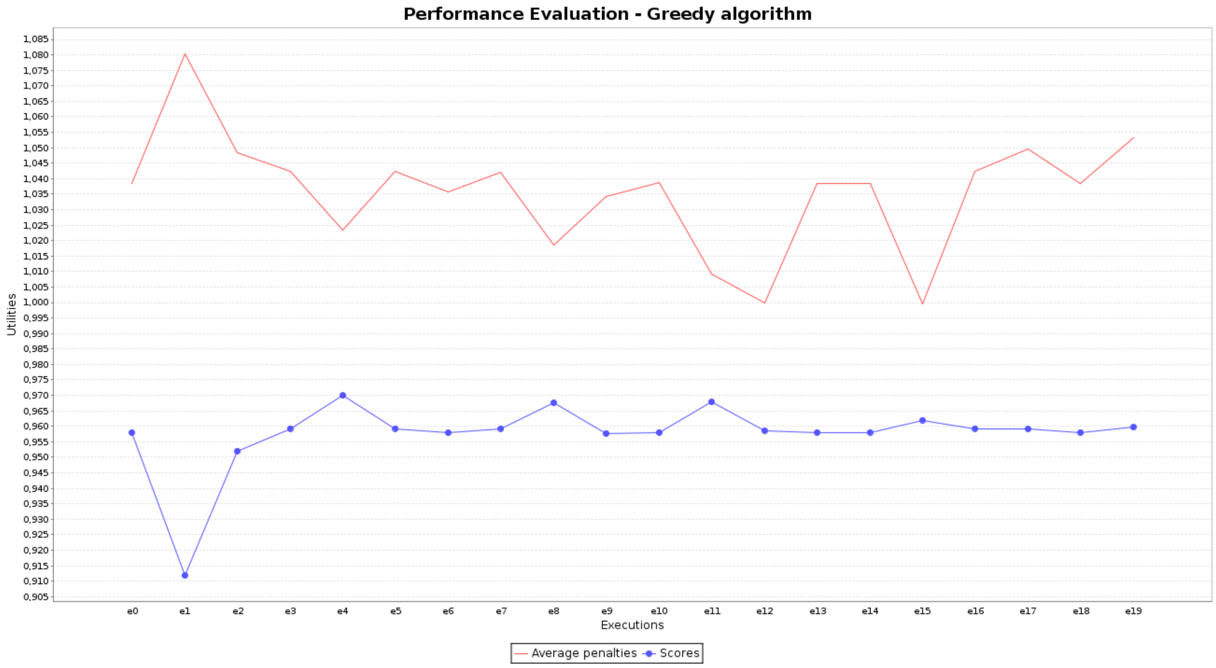


Fig. 6: Average penalties vs corresponding scores in 20 executions of the *RCSS* using *Greedy* algorithm.

score. In most cases, we can, once again, see that these values are inversely proportional, especially in cases where great differences in scores are detected.

This makes plausible the hypothesis of the existence of a relation between the two aspects: social welfare and agents repartition during coalitions formation. If many factors are contributing to the system's degree of performance, agents repartition strategy is, for sure, one of them.

Experiments relative to *DSA* and *Greedy* strategies show the same result, and thus confirm what we have just concluded.

Our approach is thus providing reliable evaluation tools of agents' effectiveness when a cooperative strategy using coalitions is adopted.

B. Benchmarking of Coalition Formation Strategies

As shown in Table I, we retained the best and worst values of respectively penalties and scores recorded for the three algorithms. We also computed the average value of each one of them in order to have a global idea about agents efficiency in the corresponding cases. We may notice the following results:

- *Comparison of all strategies*: The best strategy in terms of average score and average penalties is *DSA*. *BMS* is the second and *Greedy* is the worst one. An important result is to be mentioned: the more the strategy is good at agents repartition, the better it is in goals achievement. Indeed, we may notice that scores and penalties are inversely proportional not only when considering results of the same strategy but also when comparing the three algorithms.
- *Comparison of communicating strategies*: *DSA* is better than *BMS* in agents cooperation techniques. Its worst score for the considered set of experiments is higher than the best score of *BMS*.
- *Coordinating vs non-coordinating strategies*: *Greedy* algorithm uses no communication between agents. Such a strategy has led to worse results than other communicating strategies.

V. Discussion

The produced results make plausible the assumption that agents good repartition has an impact on the whole system's performance. So it should be taken into account as an important element in the design of agents coalition formation strategies.

The experiments we conducted evaluate the considered MAS in two different ways. First, by judging its functional adequacy by considering the score value computed by the *RMASBench* platform. Second, by computing the penalty associated to agents repartition in different coalitions during the execution of the system. A correspondence between the two aspects could then be established.

The different strategies, *BMS*, *DSA* and *Greedy*, have been used to deduce such results.

Two different cases can be distinguished :

- Agents are fairly distributed among targets according to their degree of criticality: Penalty associated to such a case is low and the reached score is relatively high.
- Some agents are badly assigned to targets, as their number exceeds the required one for the considered target. At the same time, other coalitions have fewer agents than needed. A high penalty value is then attributed to the system and the score is relatively low.

Greedy algorithm uses no communication at all, which led to bad results. Communication between agents is then necessary in cooperative systems in order to exchange each others' preferences and take decisions in a better way. Indeed, communicating strategies, *BMS* and *DSA*, led to better results. In the literature, *BMS* is more efficient than *DSA*, unlike the results that we found. This may be due to the fact that the DSA_p parameter was set to 0.1; so agents are not likely to change their targets. Which leads to a lighter communication load and thus a faster execution. This last strategy with the specified value of DSA_p combines efficiency and effectiveness in goals achievement.

VI. Conclusion and Perspectives

In this work, we highlighted the impact of agents good management on the overall performance of a MAS. Relation between the two aspects, not explicit at the beginning of this study, can be established now, thanks to a theoretical study completed by an empirical one in order to confirm the interest of establishing an efficient strategy of agents repartition in coalition formation process. When designing systems, communication methods between agents should be used in order to coordinate their actions and thus make them more efficient. Besides, communication should not be too costly regarding system's resources consumption, since that may have an inverse effect on its performance. Restrictions on agents assignments, as the ones defined by our evaluation process, should then be taken into account.

As a future work, we propose to use our evaluation function to enhance MAS performance by designing a better task allocation strategy. By optimizing it, we can deduce a better repartition of agents among coalitions.

Our evaluation process may also be improved by considering other criteria impacting MAS performance. In fact, we have so far considered homogeneous coalitions formation, while the system uses two different kinds of agents, fire brigades and police patrols. Actions of these agents are complementary and coordination between the different classes has a certain impact on the execution output. As a future work, we can study such heterogeneous coalitions and their effect on the global performance.

References

[1] E. Kaddoum, M. P. Gleizes, J.-P. Georg, P. Glize, and G. Picard, "Analyse des critres d'valuation de systmes multi-agents adaptatifs," in *Systmes Multi-Agents, Gnie logiciel multi-agents - JFSMA 09 - Dix Septimes Journes Francophones sur les Systmes Multi-Agents, Lyon, France, October 19-21, 2009*, 2009, pp. 123–132.

[2] K. Bouzouita, W. Lejouad Chaari, and M. Tagina, "Utility-based approach to represent agents' conversational preferences," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part II*, vol. 443 CCIS, no. PART 2. Springer Berlin Heidelberg, 2014, pp. 454–463.

[3] F. B. Hmida, W. L. Chaari, and M. Tagina, "Graph theory to evaluate communication in industrial multiagent systems," *IJIIDS*, vol. 5, no. 4, pp. 361–388, 2011.

[4] F. B. Hmida, W. L. Chaari, R. Dupas, and A. Seguy, "Graph-based evaluation of organization in multiagent systems," in *9th International Conference of Modeling, Optimization and Simulation-MOSIM'12*, 2012.

[5] R. Mandiau, E. L. Strugeon, and G. Agimont, "Study of the efficiency of multi-agent organizational structure: an approach based on an experiment," in *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*. IEEE, 2000, pp. 413–414.

[6] M. Ohta, T. Koto, I. Takeuchi, T. Takahashi, and H. Kitano, "Design and implementation of the kernel and agents for the robocup-rescue," in *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*. IEEE, 2000, pp. 423–424.

[7] C. Skinner and S. Ranchurn, "The RoboCup rescue simulation platform," in *9th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, 2010, pp. 1647–1648.

[8] K. Tumer and A. Agogino, "Distributed Agent-Based Air Traffic Flow Management," in *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*, vol. 5, 2007, pp. 330–337. [Online]. Available: http://dl.acm.org/citation.cfm?id=1329434

[9] M. K. Colby, W. Curran, C. Rebhuhn, and K. Tumer, "Approximating difference evaluations with local knowledge," in *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems*, ser. AAMAS '14. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1577–1578. [Online]. Available: http://dl.acm.org/citation.cfm?id=2615731.2616070

[10] M. Pujol-Gonzalez, J. Cerquides, A. Farinelli, P. Meseguer, and J. A. Rodriguez-Aguilar, "Efficient Inter-Team Task Allocation in RoboCup Rescue," in *Proc. 14th Int. Conf. Auton. Agents Multiagent Syst.*, 2015, pp. 413–421. [Online]. Available: http://www.aamas2015.com/en/AAMAS_2015_USB/aamas/p413.pdf

[11] M. Pujol-Gonzalez, A. Kleiner, A. Farinelli, S. Ramchurn, B. Shi, F. Mafoletti, and R. Reffato. (2012-2014) Rmasbench: Multi-agent coordination benchmark. [Online]. Available: http://github.com/RMASBench/RMASBench

[12] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings, "Decentralised coordination of low-power embedded devices using the max-sum algorithm," in *AAMAS '08 Proc. 7th Int. Jt. Conf. Auton. agents multiagent Syst., Estoril, Portugal, May 12-16, 2008, Volume 2*, no. Aamas, 2008, pp. 639–646.