# Tasks Allocation for Rescue Robotics: A Replicator Dynamics Approach

Sindy Amaya(✉) and Armando Mateus(✉)

Santo Tomás University, Bogotá, Colombia
{sindyamaya,armandomateus}@usantotomas.edu.co

**Abstract.** Tasks allocation on homogeneous rescue robots has been an important research field in recent years due to the advance in both robotics and artificial intelligence. Nevertheless, catastrophic scenarios still represent a hard challenge because of the complexity and uncertainty of their characteristics and parameters which produce highly heterogeneous tasks as a result of the different nature of problems they are intended for. We propose hereby an approach to the catastrophic condition exposed above by solving the replicator dynamics equation to reduce the effects of uncertainty. A standard metric based on tasks progress is defined and the main elements of game theory like payoff matrix and allocation ratios are computed in order to obtain the number of robots assigned to each task. Finally, software was built for simulation; by using this software some scenarios were defined and simulations were run to compare and validate our approach.

**Keywords:** Rescue robotics · Tasks allocation · Game theory · Replicator dynamics equation

## 1 Introduction

The use of robots in catastrophic scenarios is one of the main research fields in robotics, due to both the need to support people after a disaster and scientific and technical opportunities therein. Previous works on rescue robotics have been developed from specific disciplinary approaches; however, after a catastrophic event it is necessary to perform several tasks such as mapping, navigation, structural health inspection (SHI), life supporting, etc. All these tasks are heterogeneous as a result of the objectives they aim, the operations they perform and the methodology established for their metrics. Game theory is a meta-heuristic method commonly used for negotiation and tasks allocation in multi - agent systems providing optimal solutions for a wide range of disciplines. Nevertheless, there are two great challenges when facing rescue tasks allocation with game theory: the intrinsic uncertainty in the tasks (objectives, resources, size, etc.) and the definition of an appropriate payoff function to compute with. In


L. Rutkowski et al. (Eds.): ICAISC 2019, LNAI 11509, pp. 609–621, 2019.
https://doi.org/10.1007/978-3-030-20915-5_54

**Table 1.** Summary of rescue tasks for robotics [13], their objective and the main metric proposed in this work.

| Task | Objective | Main metrics |
|---|---|---|
| Search | Search victims | Number of found victims |
| Mapping | Map of the disaster zone | Covered area |
| Rubble removal | Clean and secure | Clean area |
| Structural inspection | Determine safe structures | Number of structures |
| In situ medical assessment | First aid assistance | Number of supported victims |
| Extraction and evacuation | Telemedicine | Number of transported victims |
| Mobile repeater | Enlarge communications coverage | Coverage area |
| Serving as a surrogate | Support human tasks | Number of functions |
| Adaptively shoring | Secure structures | Number of structures |
| Providing logistics support | Automation in the supply chain | Number of provisions |

Sect. 2, a summary and description of the main tasks involved in rescue robotics is shown with a proposed method for normalization based on the type and size of the catastrophic event. In Sect. 3 we summarize the game theory and replicator dynamics fundamentals necessary for our formulation of a solution. In Sect. 4 we propose the reward model along with necessary parameters taking into account Nash equilibria considerations. Section 5 presents simulations and results which validate our proposal. Finally, we discuss conclusions and future work in Sect. 6.

## 2 Normalization of Heterogeneous Tasks in Rescue Robotics

There exist 10 rescue tasks which can be assigned to robots as presented in [13]; these tasks can be linked to one or more of the 7 rescue phases also presented in the referred document. In order to allocate rescue tasks in teams of robots, it is necessary to face the problem of having heterogeneity in tasks: objectives, variables, processes, communication requirements, autonomy and control; these are some functions and characteristics which differ among rescue tasks for robots. Table 1 summarizes tasks and the main metric variable for each; note that metrics are different in the way they are calculated and the input parameters they require.

### 2.1 Brief Description of Robotic Rescue Tasks

Tasks in Table 1 include several sub - tasks, e.g., "acting as a mobile repeater" involves the setup tasks, positioning according to available maps and the optimization of power transmission [18]. Reconnaissance and mapping task is one of the most addressed subjects in researching: in [3], coordination of homogeneous robots is achieved by defining "frontier cells" and "exploredness cells" in an optimization problem stated to minimize the exploration time; on the contrary [9] proposes distributed autonomous coordination and centralized semi - autonomous coordination for exploration. [1,6] also work on exploratory tasks

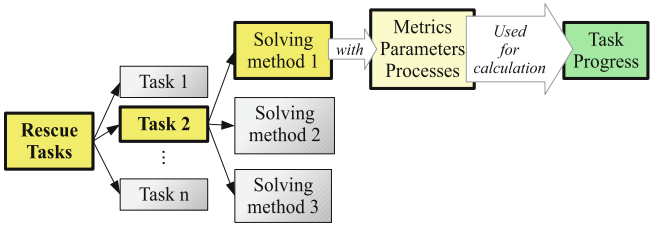


**Fig. 1.** Perspective of rescue tasks and their characteristics used in this work for tasks progress calculation.

with different approaches. In general, simultaneous localization and mapping (SLAM) techniques are the most adopted method for exploratory tasks.

SHI, structure shoring and rubble removal are tasks involving robotics engineering and civil engineering [5,7,10]. "Logistic support" and "serving as surrogate" tasks are not directly seen as a topic in rescue robotics but they are in service robotics as presented in [16], [8] and [12]. In Fig. 1 a relation chart shows the perspective used in this work: for task 2 of the n rescue tasks, three methods can be used, each one with particular metrics, parameters and processes (used for task progress calculation). For instance, search victims (task 1) could be done by image processing (solving method 1), speech recognition (solving method 2) or a mixed method of both (solving method 3); each method requires specific parameters as input (image, audio, etc) and produces results (probability of a person in a photo, text, etc) having a number of victims as a target. On the contrary, mapping (task) may require LIDAR maps as input and a 3d map as output.

### 2.2 Normalization of Tasks Metrics

We point out that a set of methods for each rescue task can be selected for carrying it out. Every method, due to the working principle, has its own requirements, algorithms and techniques which lead to specific metrics. This is a problem for tasks allocation because of the need for comparison of tasks which are defined in different dimensions. To deal with the problem above described we propose to normalize metrics, i.e., to assess the performance of the set of robots assigned to a task in a scale [0, 1] insofar as the advance in execution and the progress of the task is performed. The response in catastrophic scenarios is a race against time, then the performance metric should be an appropriate (and linear as far as possible) indicator of completeness per time. As indicated in [11,13] after a disaster event there is a general pattern in which rescue tasks are executed. This issue introduces an additional consideration: to prioritize. We gather tasks in three stages (in the beginning not directly linked to phases) and assign priorities for each group in a stage; note that stages are sequential as follows:

- Early stage: this stage starts when the catastrophic event is notified and information about it is available; victims characterization, disaster zone definition and planning, these are the activities carried out at this stage.
- Stabilization stage: a complete identification process of the catastrophic scenario, victims and risks is performed.
- Operation stage: victims support/evacuation and scenario stabilization tasks.

Length of the stages is directly linked to the scale and nature of the event then it is not possible to set a priori; this is the main reason to use a task progress metric.

## 3 Game Theory for Tasks Allocation

The evolutionary game theory studies the behavior of a population of agents that interact strategically and reevaluate their choices according to payoff opportunities [15]. By using this approach, a large number of applications have been developed in economics, social sciences, biology (where the evolutionary game theory was originated), engineering, control theory and computer science [2], [17] and [4]. Although different, these scenarios have several characteristics in common: first, they contain a large number of agents that can make decisions; second, each agent is small, that is, their choices have a low impact on the decisions of others and third, agents are anonymous, which means that the outcome of an agent depends on its own strategy and the distribution of the strategies of others. To model scenarios from the perspective of evolutionary game theory, two elements are necessary: first, to define the population game and the second one is the choice of a review protocol [14].

In population games, it must be defined at least one population of agents or players, which are grouped in a mass noted as $m \in \mathbb{R}^+$. Then, each individual of the population corresponds to a quantum of that mass. Agents can choose their actions to play from a finite set of available pure strategies denoted as $\mathcal{S}$. In general, the strategies are listed by using natural numbers, i.e., a set with $n$ available strategies, the set of strategies is given by $\mathcal{S} = \{1, .., n\}$. Thus, all players have the same set of strategies.

During the game, each player selects a pure strategy from $\mathcal{S}$. The population states are the result of the distribution of the $m$ players mass among the available strategies. The state of the population is defined with the non-negative scalars $p_1, ..., p_n$, where $p_i$ indicates the mass portion of players choosing the strategy $i \in S$. The vector $\mathbf{p} = [p_1, ..., p_n]$ denotes the state of the population. The set of possible population states, which corresponds to all the possible distributions of the agents between the strategies, this is given by the simplex in the Eq. (1):

$$\Delta = \left\{ \mathbf{p} \in \mathbb{R}^n_{\geq 0} : \sum_{i \in \mathcal{S}} p_i = m \right\} \tag{1}$$

Individuals, who choose the $i$ th strategy to play get a payment that depends on the state of the population. This payment is characterized by a payoff function $f_i : \Delta \rightarrow \mathbb{R}$. For a given population state $\mathbf{p}$, $f_i(\mathbf{p})$ determines the reward associated with the strategy $i \in \mathcal{S}$. A population game is completely defined by the payment vector $\mathbf{f} : \Delta \rightarrow \mathbb{R}^n$, where $\mathbf{f}(\mathbf{p}) = [f_1(\mathbf{p}), ..., f_n(\mathbf{p})]$.

The average fitness represents the average payment obtained by the members of the population:

$$\overline{f}\,(p) = \frac{\sum_{i \in \mathcal{S}} p_i f_i(\mathbf{p})}{m} \tag{2}$$

A population state $\mathbf{p}$ is a *Nash Equilibria*, if in each population no agent can improve their payment unilaterally by changing strategy.

Under a game theory approach, the problem of tasks allocation in a system of multiple robots can be seen as a strategic game, where the set of robots is considered as the population of agents and the set of tasks is the set of strategies, which has the game. The objective is to assign a portion of the population of agents to each one of the strategies so that some index of performance of the complete system can be minimized or maximized. The theory of evolutionary games is proposed to achieve an optimal plan of assignment of tasks for each robot.

### 3.1 Mean Dynamic and Revision Protocol

The function $\rho : \mathbb{R}^{\mathrm{n}} \times \Delta \rightarrow \mathbb{R}_{+}^{\mathrm{n} \times \mathrm{n}}$ is known as the review protocol that describes the times and the result of the decisions made by the agents about how to behave in a repetitive strategic interaction. The scalar $\rho_{\mathrm{ij}}$ captures the switch from the i strategy to j strategy.

*Mean Dynamics* allows to obtain population dynamics that describe the evolution of behavior in each population, programmed with an specific *protocol of revision*. The Eq. (3), know as *Mean Dynamic*, is:

$$\dot{p}_{\mathrm{i}} = \sum_{j \in \mathcal{S}} p_{\mathrm{j}} \rho_{\mathrm{ji}} - p_{\mathrm{i}} \sum_{i \in \mathcal{S}} \rho_{\mathrm{ij}} \tag{3}$$

The *Replicator Dynamic equation* (RD) can be deduced from the Eq. (3) and the Pairwise Proportional Imitation. Each agent imitates the strategy of an opponent previously selected, as long as the utility for that opponent is bigger than its own; this is done with a proportional probability to the difference of the utilities.

$$\dot{p}_{\mathrm{i}} = p_{\mathrm{i}} \left( f_{\mathrm{i}}\,(p) - \bar{f}\,(p) \right) \tag{4}$$

## 4 Defining the Payoff Function

For using game theory here inside, the set of robots is represented as agents and the set tasks as strategies so our main concern is to find an optimal allocation of robots for each task to reduce uncertainty effects.

Choosing a payoff function to represent the payment for specific player's strategy is one of the difficult topics in game theory. Next, we will discuss the methodology put into practice in this work.

### 4.1 Classic Definition of Payoff Matrix

Classic game theory problems like rock - scissors - paper are solved by applying Nash Equilibria. These following steps define how to obtain the portion in which each strategy must be selected for agents on a infinite time-line of game repetitions (iterations).

1. Compute expected value for player i for each strategy j; expected value is calculated.
2. State and solve the n + 1 linear system by applying Nash Equilibria (same expected value for every player).
3. Obtained results indicate the portion of choosing strategies apart from j for a number of game repetitions which ensures Nash Equilibria and optimal values.
4. To compute the portion of agents choosing strategy it is necessary to get the complement of the obtained value in step 2.

For a given payoff matrix **f** of $n$ strategies, the optimal set of agents playing strategies is found by solving the n + 1 linear system produced by steps above. For

$$\mathbf{f} = \begin{Bmatrix} f_0^0 & f_1^0 & \cdots & f_n^0 \\ f_0^1 & f_1^1 & \cdots & f_n^1 \\ \vdots & \vdots & . & \vdots \\ f_0^n & f_1^n & \cdots & f_n^n \end{Bmatrix}$$

Where $f_j^i$ represents the payoff for agent i playing strategy j. For a diagonal payment matrix, the linear system produced by applying Nash Equilibria leads to the optimal assignment **p** for each agent:

$$p_i^j = \frac{\frac{\prod_{k=1}^{n} f_k^k}{f_i^j}}{\sum_{i=1}^{n} \frac{\prod_{k=1}^{n} f_k^k}{f_i^j}} \tag{5}$$

The last expression is the basis for the development of the payoff function that is proposed below.

### 4.2 Proposal of Metrics to Evaluate Progress

As previously mentioned, we defined three metrics for task progress evaluation:

1. Advance percentage

$$a_i(\%) = \frac{N_i * T_s}{w_i} * 100 \tag{6}$$

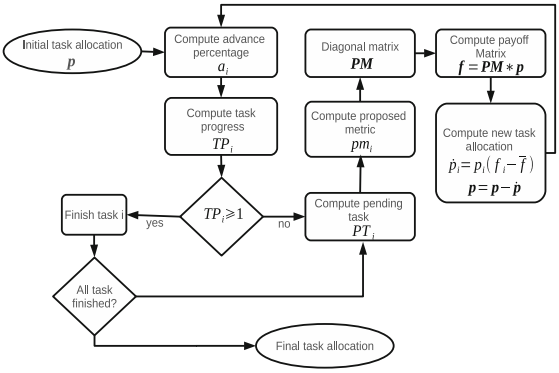


**Fig. 2.** Proposed flow chart for computing the allocation vector $\mathbf{p}$

$a_i$: Advance for task $i$
$N_i$: Number of agents dedicated to task $i$
$T_s$: Sample time
$w_i$: Total robot hour of the task

2. Task pogress percentage

$$TP_i(\%) = TP_i + a_i(\%) \tag{7}$$

3. Pending task

$$PT_i(\%) = 1 - TP_i(\%) \tag{8}$$

Above metrics define a linear model of the tasks which in most cases is inaccurate. We propose as payoff function the diagonal matrix constructed with elements by computing Eq. (9), $PM$; this expression is based on Eq. (5). For each task there is an exact set of robots assigned to execute it; it is possible for that to be an empty set. On Figure 2, we depict the algorithm we used to compute the payoff function which is computed using the Eq. 10.

$$pm_i = \frac{\frac{\prod_{j=1}^{n} w_j}{w_i}}{\sum_{k=1}^{n} \frac{\prod_{j=1}^{n} w_j}{w_k}} * PT_i \tag{9}$$

$$\mathbf{f}(\mathbf{p}) = PM * \mathbf{p} \tag{10}$$

## 5 Simulations and Results

In order to validate the approach, a simulator software with structure as depicted in Fig. 3 was built. Two scenarios are presented here to validate the proposed approach. The first one is intended for validation under ideal conditions, i.e., no stochastic behavior and no modification on the workload. In contrast, the second scenario includes an stochastic behavior for the tasks progress represented by an

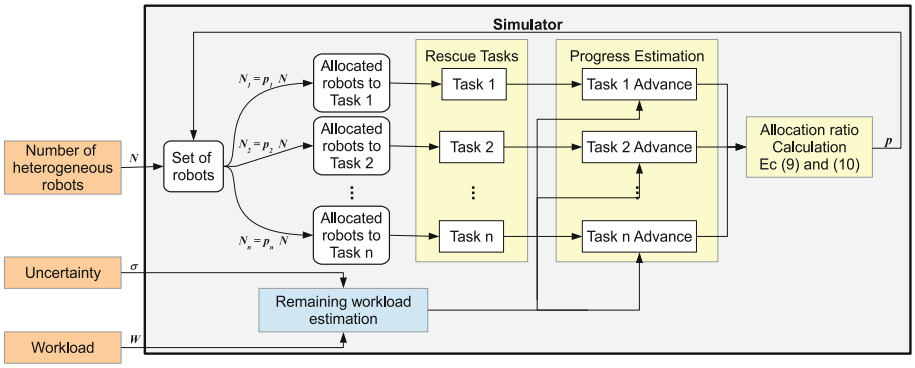


**Fig. 3.** Overall structure of the built simulator software.

additive random function v(t) (disturbance variable) modeled as a normal continuous probability distribution with parameters in Table 2. A modification on the workload is also programmed to take place at iteration 50; this modification reflects the availability of new information about tasks from that iteration.

For both, scenarios 1 and 2, the initial allocation ratio p[0] is set according to Table 2.

All scenarios are simulated with a sample time ($T_s$) of 1 h, an integer number of robots and allocated robots. A quantity of 100 homogeneous robots is set to be divided in the three tasks.

### 5.1 Scenario 1: Ideal Conditions

With the settings described above for scenario 1, the workload for tasks was defined as shown in Fig. 4 (left). After simulation was run, tasks progress, payoff function and robots allocation percentage were computed producing the results in Figs. 4 (right) and 6 (left and right).

On simulation, tasks were finished on iterations (hours) 68, 85 and 79. Note that, because in this work we define workload in robot-hours, the last iteration $it_{last}$ for ideal conditions can be calculated as follows:

$$it_{last} = \frac{\sum w_i * T_s}{N} \tag{11}$$

For settings in Table 2 for scenario 1 $it_{last} = 85$.

**Table 2.** Parameters used for simulation (* only used for scenario 2).

| Task | W (in robot hours) | W[50] (in robot hours)* | v(t)* | p[0] |
|---|---|---|---|---|
| 0 | 800 | 1000 | $N(\mu = 0, \sigma = 1)$ | 0,7 |
| 1 | 5300 | 500 | $N(\mu = 0, \sigma = 1)$ | 0,1 |
| 2 | 2400 | 4000 | $N(\mu = 0, \sigma = 1)$ | 0,2 |

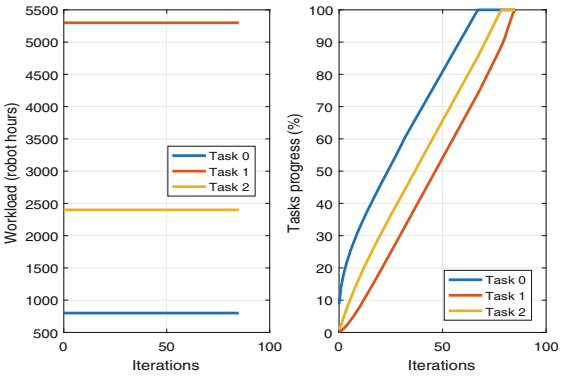


**Fig. 4.** Workload as income (left) and tasks progress as outcome (right).

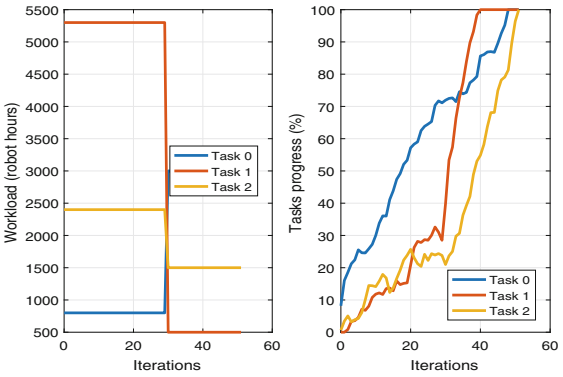


**Fig. 5.** Workload with change due to new information of tasks on iteration 50 (left) and tasks progress with stochastic behavior (right).

### 5.2 Scenario 2: Change on Workload and Stochastic

On iteration 50, due to new information about tasks workload, W is redefined as indicated in Table 2. In addition, a stochastic behavior (disturbance variable) is introduced in the model as described in Eq. 12.

$$a_i = \frac{N_i * T_s}{w_i} + v_i(t) \tag{12}$$

Figures 5 and 7 present the results of simulation for this scenario with a particular realization; tasks are finished on iterations 48, 40 and 51.

### 5.3 Results

From Figs. 4, 5, 6 and 7 it is possible to point out the following:

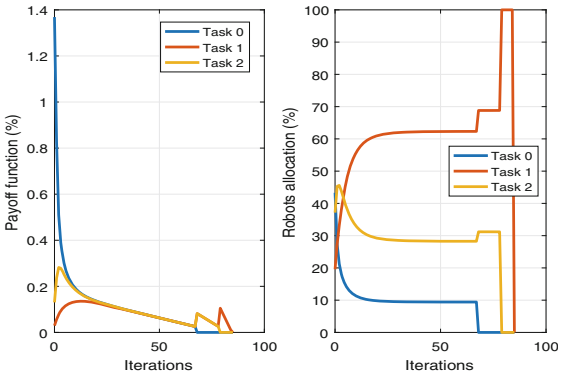


**Fig. 6.** Payoff computed for each task based on (left) and the obtained robot allocation percentage per tasks (right).

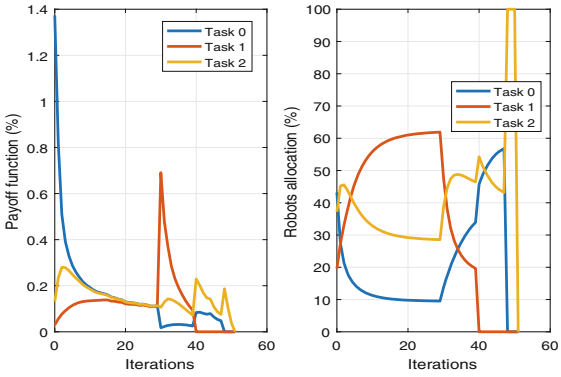


**Fig. 7.** Payoff computed for each task based on (left) and the obtained robot allocation percentage per tasks (right).

1. Payoff function of every task becomes the same as time goes by; this produces that tasks progress has the same relative speed for each task.
2. There is a lag in iterations for tasks which can be explained by the initial allocation p[0].
3. After all the payoff functions corresponding to tasks equals their values, they rapidly decrease to 0. This confirms conditions defined for Nash Equilibria and Replicator Dynamics ensuring the common variation in the payoff function.
4. Robots allocation goes from the initial p to the optimal. Note that after every tasks is finished all allocation must go to 0.
5. Stochastic behavior added to the task advance is not reflected in the task allocation, but in the payoff function. This is the desired result which holds the number of allocated robots independent of random advances.

6. In addition, the proposed approach is able to react to workload change by adjusting payoff functions and robots allocation ratios. This can be also observed by the change of Payoff function and robots allocation after a task is finished.

## 6 Conclusions

Tasks allocation in rescue robotics is characterized for the lack of solutions which integrate highly heterogeneous tasks. It is possible to divide a full set of tasks into phases according to the different times in which each task is required (related to the time from the catastrophic event occurrence). The use of advance and progress in task execution was proposed here as a common metric for tasks and the definition of workloads (necessary robot time to finish a task) as cost for every task. Other possible definitions can be done as well in future works adding uncertainty and new information availability.

We proposed a linear behavior for task progresses but real ones can be defined in other ways as exponential behaviors. Another topic to improve for the method here proposed is the inclusion of task - changing costs to represent the lost in time and energy produced when a robot is re-allocated for a different task; in this work we have supposed that robots can immediately stop to advance in a task and start to move on to another so the re-allocation cost is null.

We have taken advantage of the replicator dynamics equation to find the optimal allocation of homogeneous robots into sets of heterogeneous rescue tasks; we also obtained a reduction in the impact of uncertainty in the allocation. This method has proved to be flexible because it does not need a formal mathematical model but only available information. It is necessary the definition of a payoff function for which we propose here as a diagonal matrix, in which the payment for the portion robots dedicated to the execution of task $i$ is a reciprocal function of the task progress.

It is objective of an additional work to test and propose other payoff definitions for non linear scenarios which are closer to real ones. We have proposed for this work a metric which allows a relatively fast equalization of payoff functions for tasks, producing low uncertainty in robot allocation ratios. Obtained results encourage the definition of metrics and more payoff functions in order to have a wide range of choices which can be used to fit real scenarios requirements.

A formal convergence and stability study is encouraged after the results of the work. Some issues like Karush Kuhn Tucker conditions should be ensured for convergence. We have supposed an enormous number of robots so the allocations, quantities between 0 and 1, are not problematic, however for a lower number of robots this could lead to an integer allocation problem which can cause not convergence or instability.

## References


1. Amigoni, F., Basilico, N., Quattrini Li, A.: How much worth is coordination of mobile robots for exploration in search and rescue? In: Chen, X., Stone, P., Sucar, L.E., van der Zant, T. (eds.) RoboCup 2012. LNCS (LNAI), vol. 7500, pp. 106–117. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39250-4_11
2. Arif, M.U., Haider, S.: An evolutionary traveling salesman approach for multi-robot task allocation. In: ICAART, no. 2, pp. 567–574 (2017)
3. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Trans. Robot. **21**(3), 376–386 (2005). https://doi.org/10.1109/TRO.2004.839232
4. Chen, R., Julian, G., Bernd, M.: Social learning in a simple task allocation game. arXiv preprint arXiv:1702.05739 (2017)
5. Chen, Y., Dai, J., Mao, X., Liu, Y., Jiang, X.: Image registration between visible and infrared images for electrical equipment inspection robots based on quadrilateral features. In: 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), December, pp. 126–130 (2017). https://doi.org/10.1109/ICRAE.2017.8291366
6. Grayson, S.: Search & rescue using multi-robot systems (2014)
7. Jeon, H., Bang, Y., Myung, H.: Structural inspection robot for displacement measurement. In: 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), May, pp. 188–191 (2011). https://doi.org/10.1109/DEST.2011.5936623
8. Koubaa, A. (ed.): Robot Operating System (ROS): The Complete Reference (Volume 2). SCI, vol. 707. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54927-9
9. La Cesa, S., Farinelli, A., Iocchi, L., Nardi, D., Sbarigia, M., Zaratti, M.: Semi-autonomous coordinated exploration in rescue scenarios. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 286–293. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68847-1_27
10. Lins, R.G., Givigi, S.N., Freitas, A.D.M., Beaulieu, A.: Autonomous robot system for inspection of defects in civil infrastructures. IEEE Syst. J. **12**(2), 1414–1422 (2018). https://doi.org/10.1109/JSYST.2016.2611244
11. Luo, L., Chakraborty, N., Sycara, K.: Distributed algorithms for multirobot task assignment with task deadline constraints. IEEE Trans. Autom. Sci. Eng. **12**(3), 876–888 (2015). https://doi.org/10.1109/TASE.2015.2438032
12. Murphy, R.R., Tadokoro, S., Kleiner, A.: Disaster robotics. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 1577–1604. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1_60
13. Murphy, R.R., et al.: Search and rescue robotics. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 1151–1173. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-30301-5_51
14. Quijano, N., Ocampo-Martinez, C., Barreiro-Gomez, J., Obando, G., Pantoja, A., Mojica-Nava, E.: The role of population games and evolutionary dynamics in distributed control systems: the advantages of evolutionary game theory. IEEE Control Syst. Mag. **37**(1), 70–97 (2017). https://doi.org/10.1109/MCS.2016.2621479
15. Sandholm, W.H.: Population Games and Evolutionary Dynamics. MIT Press, Cambridge (2010)

16. Siciliano, B., Khatib, O. (eds.): Springer Handbook of Robotics. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1
17. Sun, C., Wang, X., Liu, J.: Evolutionary game theoretic approach for optimal resource allocation in multi-agent systems. In: 2017 Chinese Automation Congress (CAC), October, pp. 5588–5592 (2017). https://doi.org/10.1109/CAC.2017.8243778
18. Zheng, Y., Ling, H., Xue, J.: Disaster rescue task scheduling: An evolutionary multiobjective optimization approach. IEEE Trans. Emerg. Top. Comput. **6**(2), 288–300 (2018). https://doi.org/10.1109/TETC.2014.2369957