Comparison of Task-Allocation Algorithms in Frontier-Based Multi-robot Exploration

Jan Faigl[1](✉), Olivier Simonin[2], and Francois Charpillet[3]

[1] Czech Technical University in Prague, Prague, Czech Republic
faiglj@fel.cvut.cz
[2] INSA de Lyon, CITI-Inria Laboratory, Université de Lyon, Lyon, France
[3] Inria, Université de Lorraine, Nancy, France

Abstract. In this paper, we address the problem of efficient allocation of the navigational goals in the multi-robot exploration of unknown environment. Goal candidate locations are repeatedly determined during the exploration. Then, the assignment of the candidates to the robots is solved as the task-allocation problem. A more frequent decision-making may improve performance of the exploration, but in a practical deployment of the exploration strategies, the frequency depends on the computational complexity of the task-allocation algorithm and available computational resources. Therefore, we propose an evaluation framework to study exploration strategies independently on the available computational resources and we report a comparison of the selected task-allocation algorithms deployed in multi-robot exploration.

Keywords: Multi-robot exploration · Task-allocation · Planning

1 Introduction

The robotic exploration of unknown environment can be formulated as a problem to create a map of the environment as quickly as possible, e.g., to find eventual victims during search and rescue missions, and the main objective function considered in this paper is the time to create such a map. The fundamental approach to address the exploration problem is based on an iterative determination of possible goal candidates from which new information about the unknown part of the environment can be acquired. These candidates are assigned to the particular exploring units to maximize their utilization regarding the mission objective. This assignment problem can be formulated as the *task-allocation* problem [3]. After the assignment, each robot is navigated towards the assigned goal while its sensor system is used to perceive its surroundings and update the map being built.

J. Faigl—The presented work is supported by the Czech Science Foundation (GAČR) under research project No. 13-18316P. The travel support of the Czech Ministry of Education under the project No. 7AMB14FR019 is also gratefully acknowledged.
O. Simonin and F. Charpillet—This work is supported by the French-Czech PHC "Murotex" project.

N. Bulling (Ed.): EUMAS 2014, LNAI 8953, pp. 101–110, 2015.
DOI: 10.1007/978-3-319-17130-2_7

This process is repeated until the whole map is created, which is indicated by an empty set of the determined goal candidates.

During the exploration, new information about the environment being explored can be exploited by a more frequent determination of the goal candidates and their assignment to the robots that can improve the mission performance [4]. However, it may not necessarily be the case if robots oscillate between the assigned goals and do not explore new areas, because the location of the newly assigned goals are significantly different from the previous one. In such a case, a stable behaviour can be achieved with a less frequent assignment, e.g., after a robot reaches the previously assigned goal. Moreover, in robotics, the performance of the exploration is usually considered in a practical deployment, which even more emphasizes a less frequent decision-making because of limited on-board computational resources that are shared with other tasks like localization. Therefore, a poor behaviour of the exploration strategy might not be observed, while it can be an issue for more computationally powerful systems.

In this paper, we consider five task-allocation algorithms [1,2,6,8] dealing with the multi-robot exploration and we compare their performance under different mission execution constraints. The results indicate the frequency of the decision-making can change conclusions about the performance of the algorithms. Thus a consideration of the limiting cases of the frequency of the decision-making loop allows to provide a more general results and to identify particular constraints for a good expected performance of the algorithms in practical deployments.

Based on these findings, we propose to consider simulation to tackle robotic problems and thus we aim to encourage researchers in the field of multi-agent system and artificial intelligence to consider their task-allocation algorithms also in the multi-robot exploration missions, which can be currently considered as a problem that is more studied by the robotic community.

2 Multi-robot Exploration Framework

Three main decision-making parts can be identified in the exploration approaches based on frontier cells determination [7]. The first is the method to determine new goal candidates from the frontier cells in the actual map of the environment. The second important decision-making process is the assignment of the goal candidates to the robots together with the selection of the next navigational goal for each robot. The third part is the condition when to perform new assignment and how often the first two parts are repeated.

For simplicity, the multi-robot exploration is considered for a homogeneous group of m mobile robots $\boldsymbol{R} = \{r_1, \ldots, r_m\}$, each equipped with an omnidirectional sensor with the sensing range ρ. The control architecture for the exploration is an iterative procedure where new sensor measurements are integrated into the common map represented as the occupancy grid $\mathcal{O}cc$. The procedure can be implemented in a centralized or distributed way as follows:

1. Initialize the occupancy grid $\mathcal{O}cc$ and set the initial plans to $\mathcal{P} = (P_1, \ldots, P_m)$, where $P_i = \{\emptyset\}$ for each robot $1 \leq i \leq m$.

2. `Repeat`
   (a) Navigate robots towards their goals using the plans $\mathcal{P}$, i.e., move each robot to the next cell from the plan;
   (b) Collect new measurements with the range ρ to the occupancy grid $\mathcal{O}cc$;
   `Until` **replanning condition is meet**.
3. Update a navigation map $\mathcal{M}$ from the current occupancy grid $\mathcal{O}cc$.
4. Detect all frontiers $\mathcal{F}$ in the current map $\mathcal{M}$.
5. **Determine goal candidates $\boldsymbol{G}$** from the frontiers $\mathcal{F}$.
6. `If` $|\boldsymbol{G}| > 0$ **assign goals to the robot**
   - $(\langle r_1, g_{r_1}\rangle, \ldots, \langle r_m, g_{r_m}\rangle) = \text{assign}(\boldsymbol{R}, \boldsymbol{G}, \mathcal{M})$, $r_i \in \boldsymbol{R}$, $g_{r_i} \in \boldsymbol{G}$;
   - Plan paths to the assigned goals (as sequences of grid cells) $\mathcal{P} = \text{plan}$ $(\langle r_1, g_{r_1}\rangle, \ldots, \langle r_m, g_{r_m}\rangle, \mathcal{M})$;
   - Go to Step 2.
7. Stop all robots (all reachable parts of the environment are explored).

The navigation part (Step 2(a) and Step 2(b)) is repeated according to the specified condition. Two basic variants of the condition can be distinguished: (1) a robot reaches its goal; (2) a new assignment is performed whenever an assigned goal will no longer be a frontier cell, e.g., a surrounding unknown area becomes explored. In this paper, we call the first variant as the *goal replanning* (**GR**) condition and the second variant the *immediate replanning* (**IR**) condition. The second variant is more computationally demanding as surrounding cells of the frontier can be explored once the robot moves towards the goal about a distance equal to the size of the grid cell, e.g., 0.05 m; hence, new goals and their assignment have to be determined as quickly as possible.

A frequency of the assignment influences the performance of the exploration, but it depends on the computational complexity of the assignment procedure. Therefore, we consider a discrete time simulator to provide an evaluation setup that is independent on available computational power. An average velocity of the robot is assumed and the robot motion is restricted to traverse a single grid cell per one simulation step. Furthermore, we consider the robots have omnidirectional wheels and can move in arbitrary direction in the grid.

3 Exploration Strategies

Five task-allocation algorithms have been used in this evaluation study of the exploration strategies. All assignment procedures assign one or several goal candidates to each robot from which a single goal candidate is then assigned as the navigational goal. Thus, each goal candidate can be assigned only to one robot.

Greedy Assignment (GA) – A modified greedy assignment is utilized rather than the original approach proposed by Yamauchi in [8]. The closest not yet assigned goal is assigned to each robot sequentially; however, the assignment is performed for a random order of the robots to avoid preference of the first robots like in the original Yamauchi's approach.

Iterative Assignment (IA) – is based on the *Broadcast of Local Eligibility* [6], which is implemented in a centralized environment. The assignment is an iterative procedure, where all robot–goal pairs $\langle r, g\rangle$ are ordered by the associated distance cost. Then, the first not assigned goal from the sequence is assigned to the particular robot without an assigned goal.

Hungarian Assignment (HA) – is an optimal task-allocation algorithm for the given $m \times n$ cost matrix in which each cell value is a distance cost of particular robot–goal assignment for m robots and n goal candidates. If $m > n$ the IA algorithm is used, while for $m < n$ the cost matrix is enlarged and virtual robots are added with a very high distance cost for the goals.

Multiple Traveling Salesman Assignment (MA) – is an extension of the TSP distance cost approach [5] in which the next robot goal is selected as the first goal on the route found as a solution of the Traveling Salesman Problem (TSP). In MA, this distance cost is utilized in the multiple traveling salesman problem (MTSP) that is addressed by the $\langle$*cluster first, route second*$\rangle$ heuristic [2]. First, the goal candidates are clustered by the K-means algorithm to m clusters. Then, each cluster is assigned to a particular robot and the next robot goal is determined according to the TSP distance cost [5].

MinPos – is based on a computation of the rank $r_{i,j}$ for each goal i and robot j [1]. The rank $r_{i,j}$ is the number of robots that are closer to the goal candidate i than the robot j. Then, each robot selects the goal for which its rank is minimal. If several goal candidates have the same minimal rank for the robot i, the closest goal candidate to the robot is selected as the goal.

3.1 Proposed Goal Candidates Determination

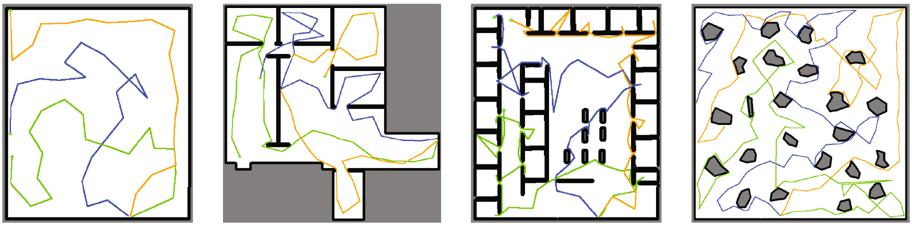
The proposed goal candidates determination method is an extension of the method [5] developed for a single robot exploration. The method is based on selection of representatives of the frontiers cells from which all frontier cells can be covered. However, we found out that for a group of robots, the original procedure [5] can provide less representatives than the number of robots, which may decreases the mission performance. Therefore, we modified the procedure to adaptively adjust the number of the determined goal candidates and call the new procedure as the *Adaptive Number of Representatives* (ANR) method.

It is assumed the freespace cells in the map $\mathcal{M}$ of the environment always form a single connected component and all frontier cells $\boldsymbol{F}$ are organized into a set of o sets (called free edges) of the single connected components $\mathcal{F} = \{\boldsymbol{F}_1, \ldots, \boldsymbol{F}_o\}$ such that $\boldsymbol{F} = \bigcup_{i=1}^{o} \boldsymbol{F}_i$ and $\boldsymbol{F}_i \cap \boldsymbol{F}_j = \emptyset$ for $i \neq j$, $1 \leq i, j \leq o$. Then, representatives are determined by the K-means clustering algorithm. n_r clusters are determined for each free edge $\boldsymbol{F}_i$ and the mean of each cluster is one goal candidate. In [5], authors determined n_r as $n_r = 1 + \lfloor |\boldsymbol{F}_i|/1.8\rho_g + 0.5\rfloor$, where ρ_g is the sensor range (in the number of grid cells). However, for many robots in the team and small n_r a goal candidate can be assigned to several robots or there will be a robot without the assigned goal. Therefore, we propose to adjust particular n_r of the largest free edges to have at least m goal candidates in total.

We experimentally verified improvement of this method over the original method of the goal candidates determination [5] for all scenarios considered in this paper. Due to limited space we consider ANR as the only goal candidates determination method and do not present the supporting results here.

4 Results

The task-allocation algorithms have been studied in four environments: *em*, *autolab*, *jh*, and *potholes*; with dimensions 21 m × 24 m, 30 m × 30 m, and 21 m × 24 m, and 40 m × 40 m, respectively, that represent office-like and open space environments, see Fig. 1. The studied performance indicator is the required time to explore the whole environment that is measured using the proposed discrete-time simulator as the number of the simulation steps denoted as T. Notice that for this criterion, it does not make sense if one robot stop its activity sooner while other robots still need to visit remaining frontiers.



(a) *em* (b) *autolab* (c) *jh* (d) *potholes*

Fig. 1. Final exploration paths in the evaluated environments and for the number of robots $m = 3$ and sensor range $\rho = 3\,\mathrm{m}$

The comparison of algorithms performance is made for a set of scenarios, where each scenario consists of the environment with the defined starting positions of the robots, the number of robots m selected from the set $m \in \{3, 5, 7\}$, and the sensor range ρ from the set $\rho \in \{3\,\mathrm{m}, 5\,\mathrm{m}, 7\,\mathrm{m}\}$. A small random perturbation (in tenths of meters) is introduced to the initial positions of the robots to consider sensitivity of the algorithms to the initial conditions. Therefore, 20 variants of each environment are considered, which gives $4 \times 3 \times 3 \times 20 = 720$ variants, and T is evaluated as the average value.

Two limiting replanning conditions **GR** and **IR** of the exploration procedure are considered. For the **GR** condition new goal candidates are determined once a robot reaches its goal, while **IR** is the fastest replanning possible as new goal candidates are determined and assigned to the robots immediately once an assigned goal is no longer frontier. Besides, based on the results in [2], we consider replanning after 7 discrete steps, or sooner when a robot reaches its goal, which is denoted as **S7R**.

Table 1. Reference exploration times T_{ref}

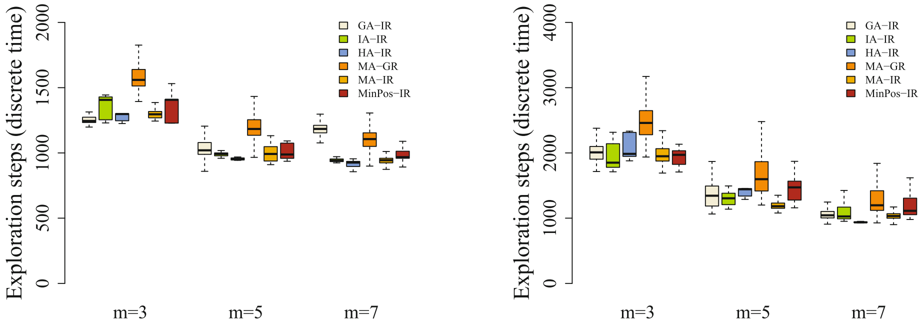
Map	$\rho = 3$ m			$\rho = 5$ m			$\rho = 7$ m		
	$m = 3$	$m = 5$	$m = 7$	$m = 3$	$m = 5$	$m = 7$	$m = 3$	$m = 5$	$m = 7$
autolab	1204	854	827	837	719	654	686	624	601
em	726	527	475	456	405	410	366	343	356
jh	864	660	613	857	654	588	782	624	588
potholes	2578	1679	1205	1678	1058	916	1301	928	829

The evaluation framework is deterministic and also IA, HA, and MinPos algorithms are deterministic procedures and thus only a single trial of each algorithm for a particular scenario is performed. On the other hand, GA and MA strategies are stochastic, and therefore, 20 trials are performed for these strategies and each scenario. Thus, the total number of performed trials is 92 880.

Evaluation Methodology – The performance of each exploration strategy may differs in a particular scenario, which requires an individual comparison for the particular combination of the map, m, ρ, and replanning condition. This leads to an excessive number of comparisons without a straightforward generalization of the results. A summary of the overall performance indicator cannot be simply computed as an average value of the required exploration time, because its absolute value depends on the size of the environment and sensor range ρ. Therefore, a reference value for each particular scenario is required to compute a global competitive ratio of the strategy. A reference can be an optimal solution of the exploration; however, it cannot be easily found because it is a computationally intractable problem due to a huge search space. That is why we propose to determine the reference value as the best found solution from the large set of the results we computed. The found references are depicted in Table 1.

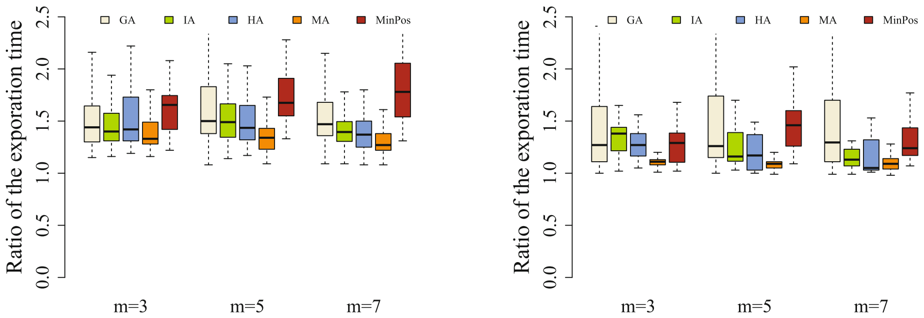
Influence of the Replanning Condition – Two scenarios are selected to show the influence of the replanning conditions and particular five-point summaries are depicted in Fig. 2. The MA strategy is considered with **GR** and **IR** conditions (see Sect. 2) denoted as MA–GR and MA–IR, respectively. These results clearly show that with the IR condition, the performance of all exploration strategies (including the greedy assignment) is better than the MA strategy with a lower replanning frequency under the GR condition. Therefore, a deployment of the exploration strategy on different computational platforms may results to different conclusions about the algorithms' performance.

Overall Comparison of the Exploration Strategies – is computed as the average value of the competitive ratio between the required exploration time T and the reference time, see Table 1. Selected aggregated results over all environments are presented in Figs. 3, 4, and 5 for particular sensor range ρ.



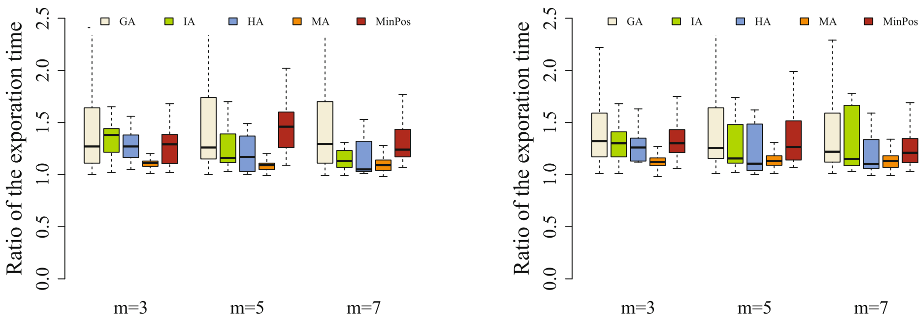
(a) *autolab* environment, ρ=3 m (b) *potholes* environment, ρ=5 m

Fig. 2. Required exploration time to explore the *autolab* and *potholes* environments with m robots, sensor range $\rho = 3$ m and **IR** or **GR** replanning condition



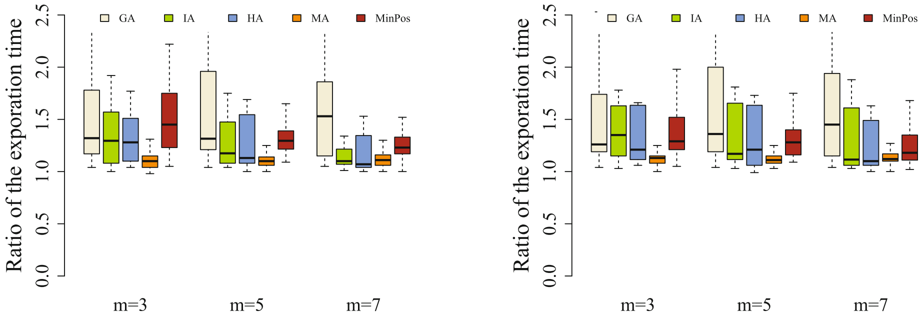
(a) **GR**, ρ=3 m (b) **GR**, ρ=5 m

Fig. 3. Overall summary of the exploration strategies performance for **GR** condition



(a) **IR**, ρ=5 m (b) **S7R**, ρ=5 m

Fig. 4. Overall summary of the exploration strategies performance for $\rho = 5$ m



(a) **IR**, ρ=7 m (b) **S7R**, ρ=7 m

Fig. 5. Overall summary of the exploration strategies performance for $\rho = 7$ m

4.1 Discussion

The results indicate the **GR** condition provides longer exploration times than the **IR** condition. The difference between **IR** and **S7R** is not significant and thus one can expect similar performance while computational requirements are significantly lower for **S7R**. The main benefit of the immediate replanning is in a lower standard deviation, which is a premise of a more reliable estimation of the average performance. This is especially noticeable for the MA strategy, which seems to provide the fastest exploration for the IR condition.

The overall results also indicate that considering a longer planning horizon in the MA strategy based on a solution of the multiple traveling salesman problem provides the lowest expected exploration time regardless the replanning frequency, i.e., in comparison to other strategies with the same replanning frequency. The MinPos strategy is sensitive to the replanning frequency; however for the **S7R** condition, it provides better or competitive performance to the IA and HA strategies. The main advantage of the MinPos strategy is the ability to be implemented in a distributed environment, which is not straightforward for implementation of the goal candidates clustering in MA. MinPos is also less computationally demanding, which can be an additional benefit.

A relative comparison of the IA and HA strategies can be concluded that both approaches provide competitive overall performance. Here, we can also highlight an ability to implement decision-making procedure in a distributed environment that is straightforward for IA using only local information, while HA may need complete information about the robots positions and all goal candidates.

5 Conclusion

A comparison of five task-allocation algorithms employed in multi-robot exploration of unknown environment is presented in this paper. The algorithms are accompanied with a new improved goal candidates determination called *adaptive number of representatives* (ANR). The used evaluation methodology is based on a reference solution of the particular exploration scenario that allows to aggregate results among different scenarios and evaluate the performance indicators statistically. Moreover, we propose to evaluate the exploration strategies using precisely defined computational environment that does not depend on the available computational resources, and which allows to obtain statistically significant results using thousands of trials.

The presented results indicate the performance of the exploration strategy depends on the frequency of replanning, and therefore, an evaluation methodology that is not dependent on a particular setup of the evaluation environment may provide a more general results and conclusions. In particular, we consider a limit case with the immediate replanning condition to validate scalability of the decision-making procedure with a more powerful computational resources. Although this may not be achieved in a practical deployment, such an evaluation allows to identify if the exploration strategy is "stable" in the taken decisions with increasing replanning frequency or if it needs a specific limit to exhibit the taken decision before another decision will be made.

Our future work can be divided into two research streams. The first stream aims to deliver a methodology for benchmarking exploration algorithms that will allow to compare different approaches in a unified and easily replicable setup, which will not only compare algorithm performance using particular hardware setup but will also provide a more general conclusion about the expected performance. The proposed evaluation framework, task-allocation algorithms, and the limiting replanning conditions are the initial building blocks for such benchmarking. The second research stream aims to consider the exploration problem in a distributed setup with a limited communication. Here, we aim to employ the proposed evaluation methodology and extend it for distributed task-allocation algorithms and their evaluation using the proposed simulator and practical verification using real mobile robots.

Acknowledgments. Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

References

1. Bautin, A., Simonin, O., Charpillet, F.: *MinPos*: a novel frontier allocation algorithm for multi-robot exploration. In: Su, C.-Y., Rakheja, S., Liu, H. (eds.) ICIRA 2012, Part II. LNCS, vol. 7507, pp. 496–508. Springer, Heidelberg (2012)
2. Faigl, J., Kulich, M., Přeučil, L.: Goal assignment using distance cost in multi-robot exploration. In: IEEE/RSJ International Conference of the Intelligent Robots and Systems (IROS), pp. 3741–3746 (2012)
3. Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. Int. J. Robot. Res. **23**(9), 939–954 (2004)
4. Holz, D., Basilico, N., Amigoni, F., Behnke, S.: Evaluating the efficiency of frontier-based exploration strategies. In: ISR/ROBOTIK, pp. 1–8 (2010)

5. Kulich, M., Faigl, J., Přeučil, L.: On distance utility in the exploration task. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4455–4460 (2011)
6. Werger, B.B., Mataric, M.J.: Broadcast of local eligibility: behavior-based control for strongly cooperative robot teams. In: 4th International Conference on Autonomous Agents, AGENTS 2000, pp. 21–22. ACM, New York (2000)
7. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp. 146–151 (1997)
8. Yamauchi, B.: Decentralized coordination for multirobot exploration. Robot. Auton. Syst. **29**, 111–118 (1999)