

## COMPARISON OF EXPLORATION STRATEGIES FOR MULTI-ROBOT SEARCH

MIROSLAV KULICH\*, TOMÁŠ JUCHELKA, LIBOR PŘEUČIL

*Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic*

\* corresponding author: [kulich@labe.felk.cvut.cz](mailto:kulich@labe.felk.cvut.cz)

**ABSTRACT.** Searching for a stationary object in an unknown environment can be formulated as an iterative procedure consisting of map updating, selection of a next goal and navigation to this goal. It finishes when the object of interest is found. This formulation and a general search structure is similar to the related exploration problem. The only difference is in goal-selection, as search and exploration objectives are not the same. Although search is a key task in many search and rescue scenarios, the robotics community has paid little attention to the problem. There is no goal-selection strategy that has been designed specifically for search. In this paper, we study four state-of-the-art strategies for multi-robot exploration, and we evaluate their performance in various environments with respect to the expected time needed to find an object, i.e. to achieve the objective of the search.

**KEYWORDS:** mobile robots, multi-robot systems, planning, search, exploration.

### 1. INTRODUCTION

Searching for a stationary object of interest in a known or unknown environment is a practical task in everyday life. Almost everyone, for example, has lost keys or has forgotten where he/she put his/her glasses, cellular phone or wallet, and has tried to find them. An important task in the search and rescue scenario is to find a black box flight recorder or debris after a plane crash, or victims/survivors after an accident or a catastrophe.

Although these practical applications and many others exist, the search problem has been addressed only marginally by the robotics community. Some effort has been devoted to the single and multi-robot search problem for environments known a priori. Sarmiento et al. [1] formulate the problem in such a way that the time required to find an object is a random variable induced by the choice of a search path and a uniform probability density function for the location of the object. They propose a two-stage process to solve the problem. First, a set of locations to be visited (known as guards, from the art gallery problem [2]) is determined. An order for visiting these locations while minimizing the expected time to find an object is then sought. The optimal order is determined by a greedy algorithm in a reduced search space, which computes a utility function for several steps ahead. This approach is then used in [3], where robot control is assumed in order to generate smooth and locally optimal trajectories. Hollinger et al. [4] utilize a Bayesian network to estimate the posterior distribution of the position of the target and present a graph search to minimize the expected time needed to capture a non-adversarial object.

A single-robot search in known environments can also be formulated as the Traveling Deliveryman Prob-

lem (TDP, also known as the Traveling Repairman Problem or the Minimal Latency Problem). This problem is studied by the operational research community, and is known to be NP-hard, even for a single robot [5]. Recently, several approximation algorithms have been presented. Salehipour et al. [6] have presented a metaheuristics combining GRASP (General Randomized Adaptive Search) with VND (Variable Neighborhood Descent). Another metaheuristics, called VNS (Variable Neighborhood Search), is introduced in [7], while linear programming is used in [8].

To the best of the authors' knowledge, the problem of a multi-robot search in an unknown environment has not been studied yet. However, methods developed for multi-robot exploration can be adopted for the search problem, as these two problems are similar in a general structure and problem formulation. A popular method for both single-robot and multi-robot exploration is frontier-based exploration, introduced by Yamauchi [9], which has been further extended by many researchers, see for example [10, 11] for an experimental evaluation of several single-robot strategies. For the multi-robot case, Wurm et al. [12] present goal assignment, based on the Hungarian method [13]. Burgard et al. [14] use a decision theory to coordinate the exploration: they estimate the expected information gain of a goal and combine it with a path cost. The method presented in Stachniss et al. [15] takes the structure of the environment into account by detecting rooms and corridors and trying to assign robots to separated rooms. In addition, approaches based on K-means clustering and assigning the clusters to particular robots are presented in [16, 17]

Intuitively, multi-robot search is a process of autonomous navigation of a team of mobile robots in an a-priori unknown environment in order to find an ob-

ject of interest. A natural condition is to perform this process with minimal usage of resources, e.g., time of search, trajectory length, or energy/fuel consumption. Following Yamauchi [9], the search algorithm can be defined (similarly to exploration) as an iterative procedure consisting of model updating with current sensory data, selection of a new goal for each robot based on current knowledge of the environment, and subsequent navigation to this goal. As we discussed in our previous paper [18], focused on a single-robot case, the key difference between search and exploration lies in the way in which the next goals to be visited are chosen at each iteration. We showed that the objectives of the two problems differ, so that trajectories optimal for exploration are not optimal for search in general. Nevertheless, exploration goal-selection strategies may be used for search.

The aim of this paper is to study the behavior of several state-of-the-art exploration strategies and to evaluate their performance in the search task. The rest of the paper is organized as follows. A definition of the problem is presented in Section 2, while the frontier-based framework for search is introduced in Section 3, and strategies are described in Section 4. An evaluation of the results and a discussion are presented in Section 5. Finally, Section 6 is dedicated to concluding remarks.

## 2. PROBLEM FORMULATION

The formulation of a multi-robot search is a direct extension of the single-robot case, introduced in [18]. We assume a team of  $N$  mobile robots equipped with a ranging sensor with a fixed, limited range (e.g., a laser range-finder) operating in an unknown environment. The search problem is defined as navigation of particular robots through this environment in order to find a stationary object placed randomly in the environment. The search is completed when the object is first detected by robot sensors<sup>1</sup>, and the natural goal is to minimize the time for this detection. The objective is to find a tuple of trajectories  $\mathcal{R}^{\text{opt}} = \langle R_i^{\text{opt}} | i = 1 \dots N \rangle$  among all possible tuples of trajectories  $\mathcal{R} = \langle R_i | i = 1 \dots N \rangle$  minimizing the expected (mean) time for detecting the object:

$$\mathcal{R}^{\text{opt}} = \arg \min_R \mathbb{E}(T|\mathcal{R}), \quad (1)$$

where  $R_i$  and  $R_i^{\text{opt}}$  are trajectories of the  $i$ -th robot,  $T$  is the time needed to traverse  $\mathcal{R}$  and

$$T_f = \mathbb{E}(T|\mathcal{R}) = \sum_{t=0}^{\infty} tp(t). \quad (2)$$

$p(t)$  can be generally an arbitrary probability density function, if priori information about the position of the object is available. Nevertheless, we consider that

<sup>1</sup>We do not address in this paper the problem of how to recognize the object to be detected. Instead, we consider that this functionality is available.

this information is not provided, so we define the probability  $p(t)$  as the ratio of the area  $A_t^{\mathcal{R}}$  newly sensed at time  $t$  when the robots follow trajectories  $\mathcal{R}$  and  $A_{\text{total}}$ , the area of the whole environment in which the robots operate:

$$p(t) = \frac{A_t^{\mathcal{R}}}{A_{\text{total}}}.$$

We can therefore rewrite (1) as

$$\mathcal{R}^{\text{opt}} = \arg \min_R \mathbb{E}(T|\mathcal{R}) = \arg \min_{\mathcal{R}} \sum_{t=0}^{\infty} tA_t^{\mathcal{R}}, \quad (3)$$

## 3. FRAMEWORK

The framework for multi-robot search is based on Yamauchi's frontier-based approach [9], successfully used for exploration, which uses an occupancy grid as the environment representation. This approach is centralized, which means that the occupancy grid is global and it is built by a central unit by integrating raw sensor measurements from all robots. Also, all decisions are made centrally and then distributed to the particular robots. The key idea of the approach is to detect *frontier cells*, i.e., reachable grid cells representing free regions adjacent to at least one as yet unexplored cell. The *frontier* is a continuous set of frontier cells such that each frontier cell is a member of exactly one frontier.

The search algorithm consists of several steps that are repeated as long as some unexplored area remains. The process starts with individual robots reading actual sensor information. After some data processing, the existing map is updated with this information. New goal candidates are then determined and goals for particular robots are assigned using a defined cost function. Having assigned the goals to the robots, the shortest path from the robots to the goals are found. Finally, the robots are navigated along the paths. The whole process is summarized in Algorithm 1.

---

### Algorithm 1: Frontier-based search algorithm

---

```

while unexplored areas exist do
  read current sensor information;
  update the map with the obtained data;
  determine new goal candidates;
  assign the goals to the robots;
  plan paths for the robots;
  move the robots towards the goals;

```

---

## 4. EXPLORATION STRATEGIES

Many exploration strategies exist, see for example [19]. Within the exploration framework presented here, we chose and implemented four methods, which are centralized, do not use a distance-based cost for goal evaluation, and are easy to implement. The following paragraphs give an overview of these methods.

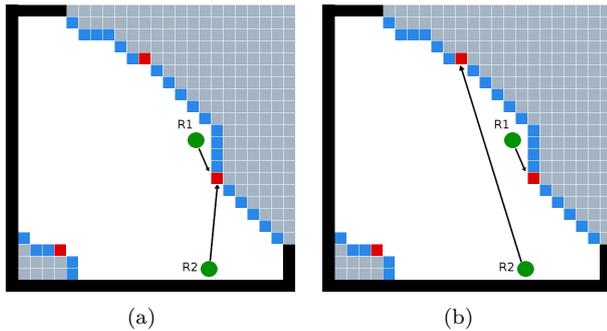


FIGURE 1. Greedy assignment, (a) two robots exploring the same goal, (b) an inefficient assignment of goals.

#### 4.1. GREEDY APPROACH

A simply and easily implementable strategy is described in [9] – each robot greedily heads towards the best goal (according to a cost function) without any coordination between robots. The strategy lacks optimality<sup>2</sup>, since one goal can be selected and explored by many robots as depicted in Fig. 1(a). To avoid this inefficiency already selected goals can be discarded from further selection. This is used in the Broadcast of Local Eligibility (BLE) assignment algorithm, developed by Werger & Mataric [20], see Algorithm 2.

---

#### Algorithm 2: BLE assignment algorithm.

---

```

while any robot remains unassigned do
    find the robot-goal pair  $(i, j)$  with the highest
    utility;
    assign the goal  $j$  to the robot  $i$  and remove
    them from the consideration;

```

---

However, this remains a greedy algorithm, and does not necessarily produce the optimal solution. The solution depends on the order of the robot-goal assignments. Fig. 1(b) depicts an example of an inefficient assignment.

#### 4.2. THE HUNGARIAN METHOD

The Hungarian method, first introduced in [13], is more sophisticated. It is an optimization algorithm which solves the worker-task assignment. The assignment can be written in the form of the  $n \times n$  matrix  $C$ , where element  $c_{i,j}$  represents the cost that the  $j$ -th task has been assigned to the  $i$ -th worker. In our case, we define the cost as the length of the path from the current position of the robot to the goal. The Hungarian method finds the optimal assignment for the given cost matrix  $C$  in  $O(n^3)$ .

<sup>2</sup>Note that we make the decision on the basis of only current knowledge of the map. From this perspective, the approach from Fig. 1(b) seems to be better than the approach in Fig. 1(a). Nevertheless, it might be globally more efficient to leave two robots to explore the same goal in some steps of the exploration process.

The algorithm requires the number of robots to be the same as the number of goals, which cannot be guaranteed throughout the exploration. If the number of robots or goals is lower, imaginary robots or goals can be added to satisfy the assumption. They have a fixed cost assigned to them, so they do not affect the real robots/goals. In the selection, the imaginary robots, and targets are skipped. This strategy does not assign the same goal to different robots and it does not depend on the order of selection.

#### 4.3. K-MEANS CLUSTERING

In the majority of multi-robot tasks, the robots start from the same area, e.g., from the entrance to a building. This leads to exhaustive exploration of the starting area during the first phase of exploration. [16] present a strategy that attempts to spread the robots quickly in the environment, so that each robot focuses on an individual part of the environment, for which it will remain responsible. This is done by using  $K$ -means clustering, which divides the remaining unknown space into the same number of regions as the number of robots. Each particular region is then assigned to the closest robot. After the assignment, each robot chooses a frontier according to a predefined cost function. The cost of frontier  $F_j$  for robot  $R_i$  assigned to region  $\zeta_i$  is defined as:

$$c_{i,j} = \begin{cases} \Delta + e(F_j, C_i) + o_{i,j} & F_j \notin \zeta_i, \\ d(F_j, R_i) + o_{i,j} & F_j \in \zeta_i, \end{cases}$$

where  $\Delta$  is a constant penalization representing the diagonal length of the map,  $e$  is the Euclidean distance,  $C_i$  is the centroid of the region,  $d$  is the real path cost defined by any path planning algorithm, and  $o_{i,j}$  is the accumulated penalization increasing the cost when the frontier has already been selected.

The frontier that does not belong to the assigned region receives a high penalization  $\Delta$ , so it can happen that there is no frontier in the assigned region, in which case, the robot selects the closest frontier to its region. As a result, robots tend to work separately in their assigned regions. If the assigned region is not directly accessible, other regions are explored on the way to the assigned region. The robots explore all these separate regions simultaneously, because each robot heads to its own region. This disperses the robots in the environment, and different parts of the environment are explored at similar speeds.

In general, the  $K$ -means algorithm consists of the following steps.

- (1.) Randomly choose  $K$  centroids  $C_i$  where  $1 \leq i \leq K$ .
- (2.) Classify each not yet explored cell in the environment to the class  $\zeta_i$  of its closest centroid  $C_i$ .
- (3.) Determine a new centroid for each class.
- (4.) If all the centroids did not change, finish. Otherwise, continue with the step 2.

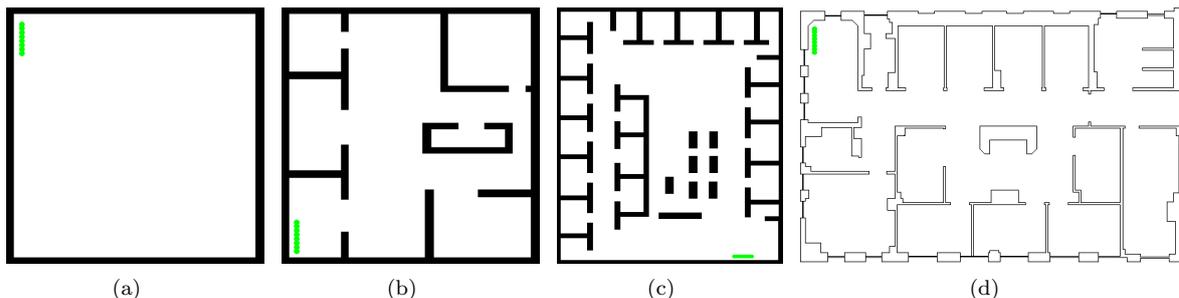


FIGURE 2. Maps used in the experiments. The starting positions are marked with the green circles: (a) *Empty* map with dimensions  $50 \times 50$  m; (b) *Arena* map with dimensions  $50 \times 50$  m; (c) *Jh* map with dimensions  $52.5 \times 60$  m; (d) *Hospital* map with dimensions  $138 \times 110.75$  m.

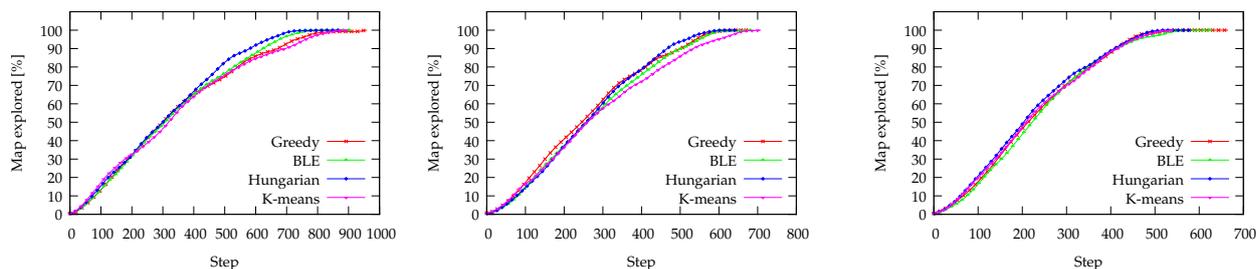


FIGURE 3. *Empty* map size progress: for 4 robots (left), for 6 robots (center), for 8 robots (right).

## 5. RESULTS

The strategies mentioned above (Greedy, BLE, the Hungarian method, and K-means clustering) were implemented in a framework for search/exploration in a polygonal domain [21]. The framework uses ROS [22] as a communication middleware. Sensor measurements are represented by polygons and combined together by polygon operations. This way, a polygonal representation of the environment is built. The framework therefore enables search/exploration to be performed with a larger number of robots, in larger experiments, and re-planning can be carried out faster than is possible in a grid-based approach. Experiments to compare the strategies were performed in simulation, using maps with various sizes and structures, see Figure 2.

The *empty* map 2(a) was created to simulate the trivial case of a big room without obstacles. The *arena* map 2(b) represents a slightly structured environment with large corridors and rooms. The *jh* map 2(c) represents a real administrative building with many separated rooms. The *hospital* map 2(d) is a part of the hospital-section map from the Stage simulator representing another building.

All the simulations were examined on the same hardware with a quad-core processor on 3.30 GHz running x86\_64 GNU/Linux Kubuntu 3.0.0-20, ROS electric with the Stage simulator and gcc 4.6.1.

The considered numbers of robots are  $m = \{4, 6, 8\}$ , while the sensor range is set to  $\rho = 5$  meters with a  $270^\circ$  field of view. The robots are controlled using our implementation of the SND algorithm [23] as

a ROS node, and the planning period was set to 1 second.

Although all the strategies are deterministic, other parts of the exploration process (especially robot control) and thus the whole process are not deterministic. Each experimental setup determined by a tuple  $\langle \text{map}, \text{number of robots}, \text{strategy} \rangle$  was therefore repeated several times to obtain statistical characteristics of the exploration. The number of runs differs for different maps, as does the time demand for performing a certain number of experiments. The number of repetitions for *empty*, which is the easiest map, was 30. For *arena* there were 22 runs, while 17 runs were performed for each setup for *jh* and 10 for *hospital*.

As the computations are not time-consuming, the experiments were speeded up 3 times in the Stage configuration file. This has the same effect as when the planning period is set to 3 seconds and the simulation speed is normal. The benefit is that we can perform the experiments faster. This is crucial, as we performed about 700 experiments, each taking 5 to 15 minutes.

A statistical evaluation of the strategies is shown in Figs. 3–8. The first two figures depict the progress of a newly explored area averaged for each map and strategy over all runs, while Figs. 7 and 8 display five-number summaries of the expected time for finding object —  $T_f$  — as defined in (2).  $A_t^R$  is computed as the difference of the volumes of the already explored areas at times  $t$  and  $t - 1$ , while  $A_{\text{total}}$  is the volume of the explored area in the final map.

For the *empty* environment and 4 to 6 robots, all

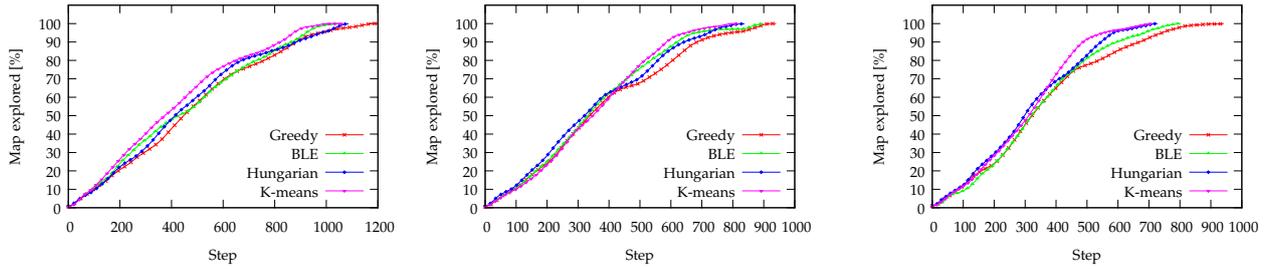


FIGURE 4. Arena map size progress: for 4 robots (left), for 6 robots (center), for 8 robots (right).

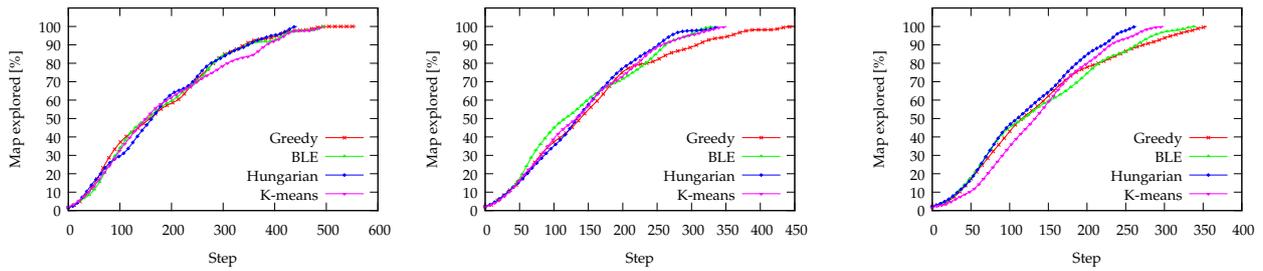


FIGURE 5. Jh map size progress: for 4 robots (left), for 6 robots (center), for 8 robots (right).

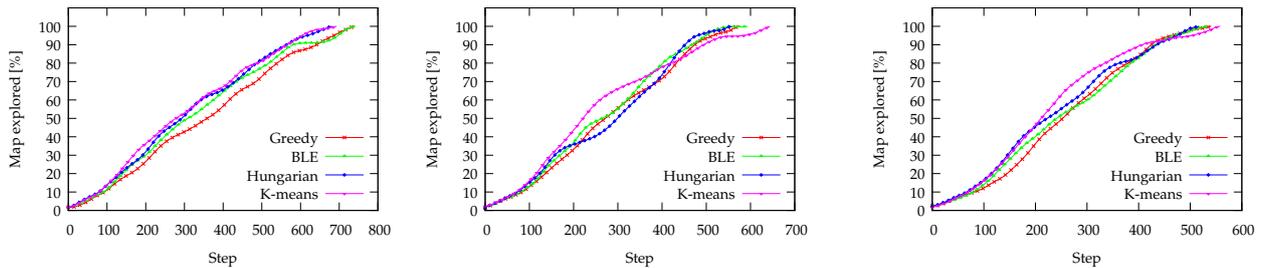


FIGURE 6. Hospital map size progress: for 4 robots (left), for 6 robots (center), for 8 robots (right).

the methods, except K-means, report similar results with a difference of 0.3% for 4 robots and 1% for 6 robots. The worse behavior of the K-means strategy (6% worse than the Hungarian method for 4 robots and 8.5% worse for 6 robots) is caused by a non appropriate distribution of robots in the first stages of the search, and the need to redistribute them later. The situation is more balanced for 8 robots, but the greedy approach and the Hungarian approach behave slightly better. The performance of the methods in the other maps shows similar characteristics. Greedy performs worst in almost all cases, followed by BLE. In general, the best results were achieved by the Hungarian method, followed by K-means in *arena* and *hospital*. K-means even outperforms the Hungarian method in some cases. Note the poor results of K-means for *jh*. This environment contains many small rooms, and its partitioning forces robots to explore small regions partially spread over two or three rooms. This slows down the search as more than one robot visits the same room, in many cases.

## 6. CONCLUSION

Although search in unknown environments has many practical applications, it has until now been addressed only marginally by the robotics community. In this paper, we define the problem for a team of robots, and present the results of several standard exploration approaches for a search. Although the Hungarian method outperforms the other approaches in most cases, the differences in results appear to be marginal, and differ for different maps and for different numbers of robots. In our further research we therefore plan to study the behaviour of the methods in greater detail in order to clarify the reasons for their performance. Although the number of experiments that have been performed is not small, more experiments are needed in order to draw statistically reasonable conclusions, and especially to provide a statistical analysis of variance. We believe that this study will help in the design of novel methods for search in future.

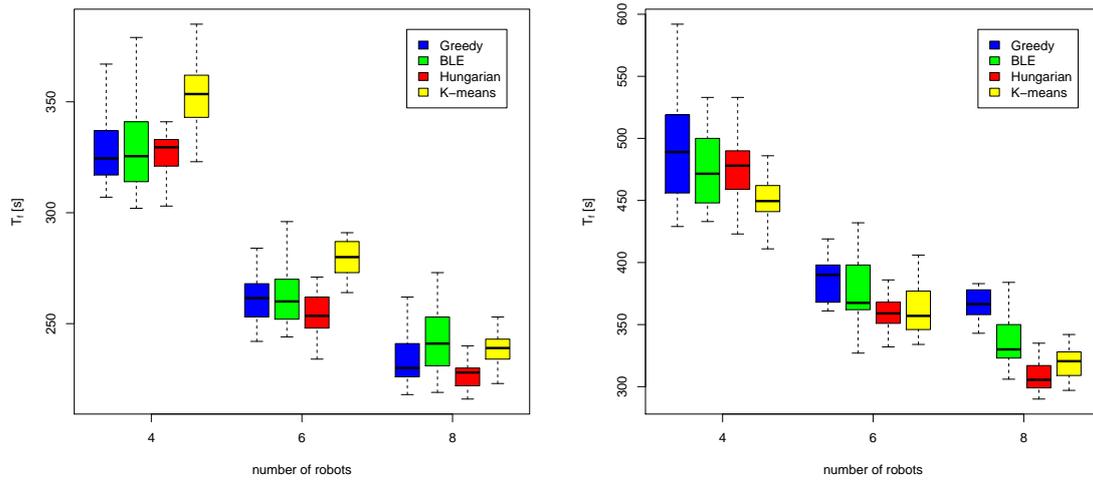


FIGURE 7. Comparison of the strategies, i.e. the five-number summaries of  $T_f$  for *empty* (left) and for *arena* (right) map.

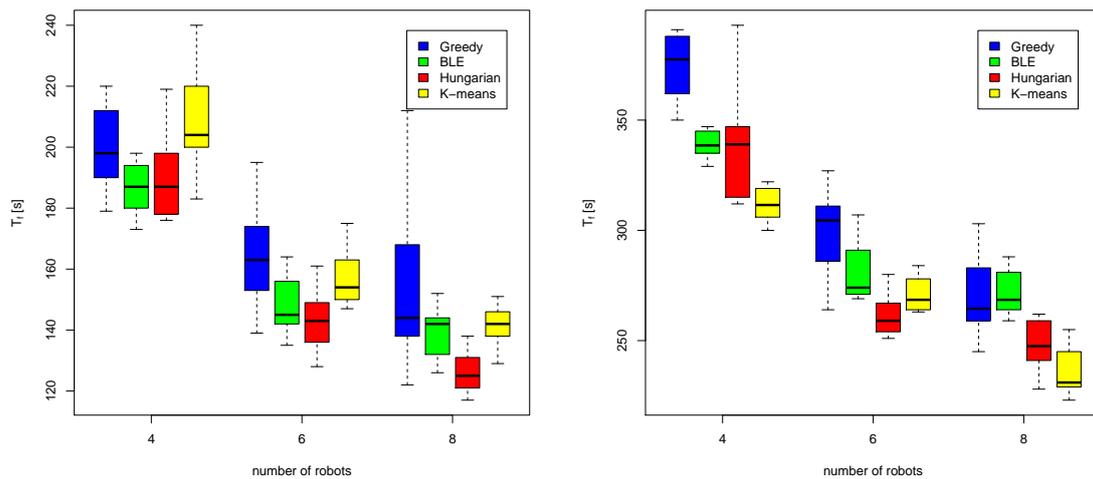


FIGURE 8. Comparison of the strategies, i.e., the five-number summaries of  $T_f$  for *jh* (left) and for *hospital* (right) map.

#### ACKNOWLEDGEMENTS

This work has been supported by the Technology Agency of the Czech Republic under project no. TE01020197.

#### REFERENCES

- [1] A. Sarmiento, R. Murrieta-Cid, S. Hutchinson. A multi-robot strategy for rapidly searching a polygonal environment. In C. Lemaître, C. A. Reyes, J. A. González (eds.), *Proc. 9th Ibero-American Conf. on AI on Advances in Artificial Intelligence*, vol. 3315 of *Lecture Notes in Computer Science*, pp. 484–493. Springer, 2004. DOI:10.1007/978-3-540-30498-2\_48.
- [2] T. Shermer. Recent results in art galleries [geometry]. *Proc of the IEEE* **80**(9):1384–1399, 1992. DOI:10.1109/5.163407.
- [3] A. Sarmiento, R. Murrieta, S. Hutchinson. An efficient motion strategy to compute expected-time locally optimal continuous search paths in known environments. *Advanced Robotics* **23**(12-13):1533–1560, 2009. DOI:10.1163/016918609X12496339799170.
- [4] G. Hollinger, J. Djughash, S. Singh. Coordinated search in cluttered environments using range from multiple robots. In C. Laugier, R. Siegwart (eds.), *Field and Service Robotics*, vol. 42 of *Springer Tracts in Advanced Robotics*, pp. 433–442. Springer Berlin Heidelberg, 2008. DOI:10.1007/978-3-540-75404-6\_41.
- [5] K. Trummel, J. Weisinger. The complexity of the optimal searcher path problem. *Operations Research* **34**(2):324–327, 1986. DOI:10.1287/opre.34.2.324.
- [6] A. Salehipour, K. Sørensen, P. Goos, O. Bräysy. Efficient GRASP+VND and GRASP+VNS

- metaheuristics for the traveling repairman problem. *A Quarterly Journal of Operations Research (4OR)* **9**:189–209, 2011. DOI:10.1007/s10288-011-0153-0.
- [7] N. Mladenović, D. Urošević, S. Hanafi. Variable neighborhood search for the travelling deliveryman problem. *A Quarterly Journal of Operations Research (4OR)* pp. 1–17, 2012. DOI:10.1007/s10288-012-0212-1.
- [8] I. Méndez-Díaz, P. Zabala, A. Lucena. A new formulation for the traveling deliveryman problem. *Discrete Appl Math* **156**(17):3223–3237, 2008. DOI:10.1016/j.dam.2008.05.009.
- [9] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents, AGENTS '98*, pp. 47–53. ACM, New York, NY, USA, 1998. DOI:10.1145/280765.280773.
- [10] F. Amigoni. Experimental evaluation of some exploration strategies for mobile robots. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2818–2823. 2008. DOI:10.1109/ROBOT.2008.4543637.
- [11] D. Holz, N. Basilico, F. Amigoni, S. Behnke. Evaluating the efficiency of frontier-based exploration strategies. In *Proc. Int. Symp. on Robotics/German Conf. on Robotics*, pp. 1–8. VDE Verlag, 2010.
- [12] K. Wurm, C. Stachniss, W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1160–1165. 2008. DOI:10.1109/IROS.2008.4650734.
- [13] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**:83–97, 1955. DOI:10.1002/nav.3800020109.
- [14] W. Burgard, M. Moors, C. Stachniss, F. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on* **21**(3):376–386, 2005. DOI:10.1109/TRO.2004.839232.
- [15] C. Stachniss, O. M. Mozos, W. Burgard. Efficient exploration of unknown indoor environments using a team of mobile robots. *Annals of Mathematics and Artificial Intelligence* **52**(2-4):205–227, 2008. DOI:10.1007/s10472-009-9123-z.
- [16] A. Solanas, M. A. Garcia. Coordinated multi-robot exploration through unsupervised clustering of unknown space. In *Proc. Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 717–721 vol.1. 2004. DOI:10.1109/IROS.2004.1389437.
- [17] D. Puig, M. Garcia, L. Wu. A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation. *Robotics and Autonomous Systems* **59**(9):635–653, 2011. DOI:10.1016/j.robot.2011.05.004.
- [18] M. Kulich, L. Přeučil, C. M. B. Bront. Single robot search for a stationary object in an unknown environment. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 5830–5835. 2014. DOI:10.1109/ICRA.2014.6907716.
- [19] M. Juliá, A. Gil, O. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots* **33**(4):427–444, 2012. DOI:10.1007/s10514-012-9298-8.
- [20] B. B. Werger, M. J. Mataric. Broadcast of local eligibility for multi-target observation. In L. Parker, G. Bekey, J. Barhen (eds.), *Distributed Autonomous Robotic Systems 4*, pp. 347–356. Springer-Verlag, 2001. DOI:10.1007/978-4-431-67919-6\_33.
- [21] T. Juchelka, M. Kulich, L. Přeučil. Multi-robot exploration in the polygonal domain. In *Int. Conf. on Automated Planning: Workshop on Planning and Robotics*. 2013.
- [22] M. Quigley, K. Conley, B. P. Gerkey, et al. ROS: an open-source robot operating system. In *Proc. IEEE Int. Conf. on Robotics and Automation: Workshop on Open Source Software*. 2009.
- [23] J. W. Durham, F. Bullo. Smooth nearness-diagram navigation. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 690–695. 2008. DOI:10.1109/IROS.2008.4651071.