Comparison of exploration strategies for multi-robot search

Miroslav Kulich, Tomáš Juchelka and Libor Přeučil
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technicka 2, 166 27 Prague
Czech Republic
Email: kulich@fel.cvut.cz, tomas.juchelka@gmail.com, preucil@fel.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 and finishing when the object of interest is found. This formulation and a general structure of search is similar to the related exploration problem with the only exception in a goal-selection as search and exploration objectives are different. Despite search is a key task in many search and rescue scenarios, the robotic community has paid a little attention to the problem and there is no goal-selection strategy designed especially for search. In this paper, we study four state-of-the-art strategies for multi-robot exploration and evaluate their performance in various environments with respect to the expected time needed to find an object, which is the objective of the search.**

I. INTRODUCTION

Searching for a stationary object of interest in known or unknown environments is a practical task of everyday life. Almost everyone, for example, lost keys or forgot where he/she put his/her glasses, cellular phone, or wallet and 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/survived after some accident or catastrophe.

Despite these any many other practical applications, the search problem has been addressed marginally by the robotic community. Some effort has been devoted to the single and multi-robot search problem for a priori known environments. Sarmiento et al. [1] formulates the problem so that the time required to find an object is a random variable induced by a choice of search path and a uniform probability density function for the object's location. They propose two-stage process to solve the problem. Firstly, a set of locations (known as guards from the art gallery problem [2]) to be visited is determined. An order of visiting those locations minimizing the expected time to find an object is found then. 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 for estimating the posterior distribution of target's position and present a graph search to minimize the expected time needed to capture a non-adversarial object.

Single-robot search in known environments can be also formulated as the Traveling Deliveryman Problem (TDP, also as Traveling Repairman Problem or Minimal Latency Problem), which is studied by the operational research community and which is known to be NP-hard even for one robot [5]. Recently, several approximation algorithms were presented. Salehipour et al. [6] present 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 authors' knowledge, the problem of multi-robot search in an unknown environment has not been studied yet. On the other hand, 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 to both single-robot and multi-robot exploration is frontier-based exploration introduced by Yamauchi [9], which was further extended by many researchers, see for example [10], [11] for experimental evaluation of several single-robot strategies. Concerning 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 an 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. Furthermore, 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 a-priory unknown environment in order to find an object 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 the iterative procedure consisting of model updating with actual sensory data, selection of a new goal for each robot based on the 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 how next goals to be visited are chosen at each iteration. We showed that the objective of the both problems differ and thus 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 behavior of several state-of-the art exploration strategies and evaluate their performance in the search task. The rest of the paper is organized as follows. The problem definition is presented in Section II, while the frontier-based framework for search is introduced in Section III and strategies are described in Section IV. Evaluation of the results and discussions are presented in Section V. Finally, Section VI is dedicated to concluding remarks.

‘

II. Problem formulation

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

$$\mathcal{R}^{opt} = \arg\min_{R} \mathbb{E}(T|\mathcal{R}), \tag{1}$$

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

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

$p(t)$ can be generally an arbitrary probability density function if a priory information about object's position is available. Nevertheless, we consider 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 the trajectories $\mathcal{R}$ and A_{total}, the area of the whole environment the robots operate:

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

Equation 1 can be thus rewritten as

$$\mathcal{R}^{opt} = \arg\min_{R} \mathbb{E}(T|\mathcal{R}) = \arg\min_{\mathcal{R}} \sum_{t=0}^{\infty} t A_t^{\mathcal{R}}, \tag{3}$$

III. 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 with at least one not yet explored 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 until some unexplored area remain. The process starts with reading actual sensor information by individual robots. 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;
```

IV. Exploration strategies

Many exploration strategies exist, see for example [19]. We chose and implemented within the presented exploration framework four methods, which are centralized, don't use distance-based cost for goal evaluation and are easy to implement. The following paragraphs give an overview of these methods.

A. Greedy approach

A simply and easily implementable strategy is described in [9] – each robot greedily heads towards the best (according to a cost function) goal without any coordination between robots. The strategy is not much optimal[2] since one goal can be selected and explored by many robots as depicted in Fig. 1(a). To avoid this inefficiency it is possible to discard already selected goals from the further selection. This is used in the Broadcast of Local Eligibility (BLE) assignment algorithm developed by Werger & Mataric [20], see Algorithm 2.

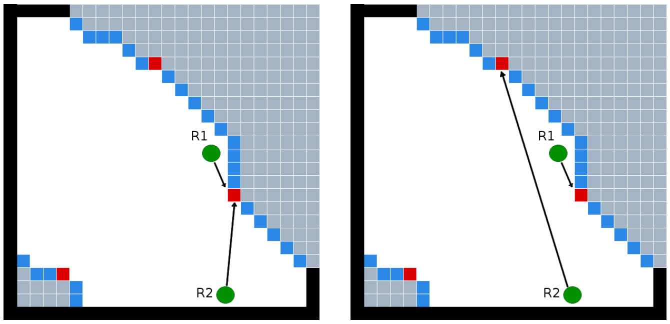
Algorithm 2: BLE assignment algorithm.

```
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;
```

Nevertheless, it is still a greedy algorithm, which not necessarily produces the optimal solution. The solution depends on the order of the robot-goal assignments. Fig. 1(b) depicts an example of an inefficient targets assignment.

[1]We don't address the problem how to recognize the object to be detected in the paper. Instead, we consider this functionality is available.

[2]Note, that we make decision given only current knowledge of the map. From this perspective, the approach from Fig. 1(b) seems to be better than the one in Fig. 1(b). Nevertheless, it could be globally more efficient to let two robots to explore the same goal at some steps of the exploration process.



(a) (b)

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

B. Hungarian method

The Hungarian method firstly introduced in [13] is more sophisticated. It is an optimization algorithm which solves the worker-task assignment. The assignment can be written in a form of the $n \times n$ matrix C, where the 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 a length of the path from the current robot's position to the goal. The Hungarian method finds the optimal assignment for the given cost matrix C in $O(n^3)$.

The algorithm requires the number of robots to be the same as the number of goals, which can not be guaranteed throughout the exploration. If the number of robots or goals is lower it is possible to add imaginary robots or goals to satisfy the assumption. They have assigned a fixed cost, so they don't 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.

C. K-means clustering

In the majority of multi-robot tasks, robots start from the same area, e.g., from the building entrance. It leads to an exhaustive exploration of the starting area during the first phase of exploration. [16] present a strategy that tries to spread the robots quickly in the environment so that each robot focuses on an individual part of the environment for which it will stay responsible. This is done by using K-means clustering that 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 the frontier F_j for the robot R_i assigned to the region ζ_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 Δ 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 been already selected.

The frontier that does not belong to the assigned region receives a high penalization Δ so it can happen that there is no frontier in the assigned region, in that case, the robot selects the closest frontier to its region. As the 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 one. Robots explore all these separated regions simultaneously because each robot heads to its own region. This leads to a dispersion of 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 ζ_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.

V. Results

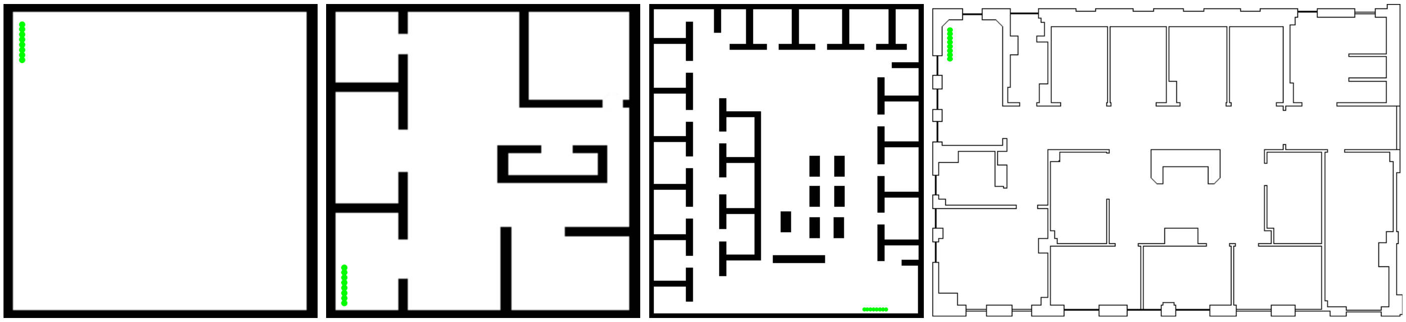
The aforementioned strategies (Greedy, BLE, Hungarian method, and K-means clustering) were implemented in a framework for search/exploration in a polygonal domain [21]. The framework uses ROS [22] as a communication middleware and the mapping process based on polygon operations as sensor measurement are represented by polygons so it enables to perform search/exploration with a higher number of robots, in larger experiments and make re-planning faster than possible in a grid-based approach. The experiments comparing the strategies have been performed in simulation using maps with various sizes and structures, see Figure 2.

The *empty* map 2(a) has been created to simulate a 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 the 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, 8 GB RAM 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 $270°$ 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. Each experimental setup determined by a tuple $\langle$map, number of robots, strategy$\rangle$ was therefore repeated several times to obtain statistical characteristics of the exploration. The number of runs differs for different maps as time demands to perform such number experiments. The number of repetitions for *empty*, which is the easiest map, was 30, for *arena* it was 22, while 17 runs were performed for each setup for *jh* map and 10 for *hospital* map.



(a) (b) (c) (d)

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

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

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

For *empty* environment and 4 to 6 robots all the methods except K-means report similar results with difference 0.3% for 4 robots and 1% for 6 robots. A worse behavior of K-means strategy (6% worse than Hungarian for 4 robots and 8.5% for 6 robots) is caused by a not appropriate distribution of robots at the first stages of the search and need of their redistribution later. The situation is more balanced for 8 robots but the greedy and Hungarian approaches behave slightly better. Performance of the methods in other maps shows similar characteristics. Greedy is the worst in almost all cases followed by BLE. In general, the best results were achieved by Hungarian method followed by K-means in *arena* and *hospital*, which even outperforms Hungarian method in some cases. Notice bad 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 which slows down the search as the robots have to visit same rooms in many cases.

VI. Conclusion

Although search in unknown environments has many practical applications it has been addressed marginally by the robotic community. In this paper, we define the problem for a team of robots and present results of several standard exploration approaches for the search. Despite the Hungarian method outperforms the other approaches in majority of cases, these results are tight on the first sight and differ for different maps and numbers of robots. In our further research we therefore plan to study a behaviour of the methods in more details in order to clarify reasons of their performance. Although the number of performed experiments is not small, more experiments are needed to make statistically reasonable conclusions, especially to provide statistical analysis of variance. We believe that this study will help to design novel methods for search in future.

Acknowledgment

This work has been supported by the Technology Agency of the Czech Republic under the project no. TE01020197 "Centre for Applied Cybernetics".

References

[1] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson, "A multi-robot strategy for rapidly searching a polygonal environment," in *Proc. 9th Ibero-American Conf. on AI on Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, C. Lematre, C. A. Reyes, and J. A. Gonzlez, Eds., vol. 3315. Springer, 2004, pp. 484–493.

[2] T. Shermer, "Recent results in art galleries [geometry]," *Proc. of the IEEE*, vol. 80, no. 9, pp. 1384 –1399, sep 1992.

[3] A. Sarmiento, R. Murrieta, and S. Hutchinson, "An efficient motion strategy to compute expected-time locally optimal continuous search paths in known environments," *Advanced Robotics*, vol. 23, no. 12-13, pp. 1533–1560, 2009.

[4] G. Hollinger, J. Djugash, and S. Singh, "Coordinated search in cluttered environments using range from multiple robots," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, C. Laugier and R. Siegwart, Eds. Springer Berlin Heidelberg, 2008, vol. 42, pp. 433–442.

[5] K. Trummel and J. Weisinger, "The complexity of the optimal searcher path problem." *Operations Research*, vol. 34, no. 2, pp. 324 – 327, 1986.

[6] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy, "Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem," *A Quarterly Journal of Operations Research (4OR)*, vol. 9, pp. 189–209, 2011.

[7] N. Mladenović, D. Urošević, and S. Hanafi, "Variable neighborhood search for the travelling deliveryman problem," *A Quarterly Journal of Operations Research (4OR)*, pp. 1–17, 2012.

[8] I. Méndez-Díaz, P. Zabala, and A. Lucena, "A new formulation for the traveling deliveryman problem," *Discrete Appl. Math.*, vol. 156, no. 17, pp. 3223–3237, Oct. 2008.

[9] B. Yamauchi, “Frontier-based exploration using multiple robots,” in *Proc. 2nd Int. Conf. on Autonomous Agents*, 1998, pp. 47–53.

[10] F. Amigoni, “Experimental evaluation of some exploration strategies for mobile robots,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2818–2823.

[11] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, “Evaluating the efficiency of frontier-based exploration strategies,” in *Proc. Int. Symp. on Robotics/German Conf. on Robotics*, 2010, pp. 1–8.

[12] K. Wurm, C. Stachniss, and W. Burgard, “Coordinated multi-robot exploration using a segmentation of the environment,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, sept. 2008, pp. 1160 – 1165.

[13] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[14] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–378, 2005.

[15] C. Stachniss, O. M. Mozos, and W. Burgard, “Efficient exploration of unknown indoor environments using a team of mobile robots,” *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 205–227, 2008.

[16] A. Solanas and M. A. Garcia, “Coordinated multi-robot exploration through unsupervised clustering of unknown space,” in *Proc. Int. Conf. on Intelligent Robots and Systems*, 2004.

[17] D. Puig, M. Garcia, and L. Wu, “A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation,” *Robotics and Autonomous Systems*, vol. 59, pp. 635–653, 2011.

[18] M. Kulich, L. Preucil, and C. M. B. Bront, “Single robot search for a stationary object in an unknown environment,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014.

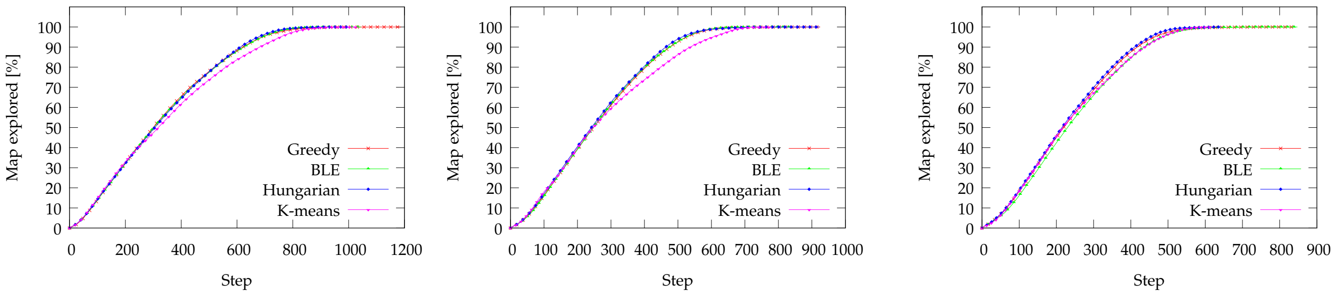
[19] M. Juliá, A. Gil, and O. Reinoso, “A comparison of path planning strategies for autonomous exploration and mapping of unknown environments,” *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9298-8

[20] B. B. Werger and M. J. Mataric, “Broadcast of local eligibility for multi-target observation,” in *Distributed Autonomous Robotic Systems 4*. Springer-Verlag, 2001, pp. 347–356.

[21] T. Juchelka, M. Kulich, and L. Preucil, “Multi-robot exploration in the polygonal domain,” in *Int. Conf. on Automated Planning: Workshop on Planning and Robotics*, 2013.

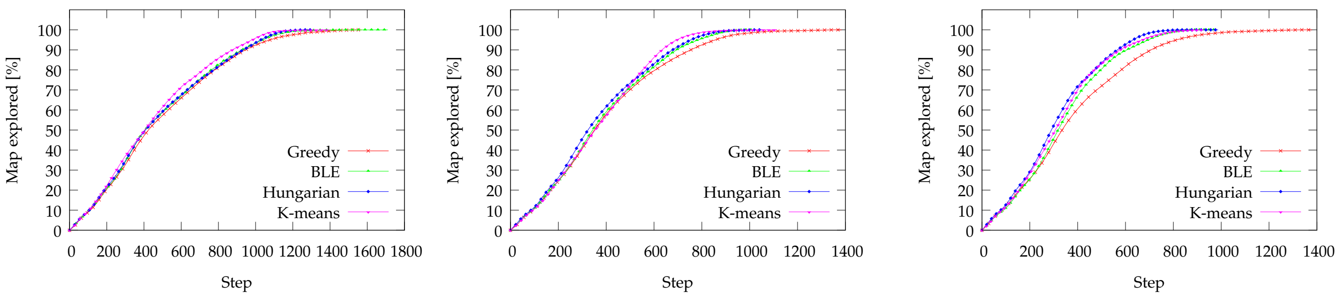
[22] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *Proc. IEEE Int. Conf. on Robotics and Automation: Workshop on Open Source Software*, 2009.

[23] J. W. Durham and F. Bullo, “Smooth nearness-diagram navigation,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 690–695.



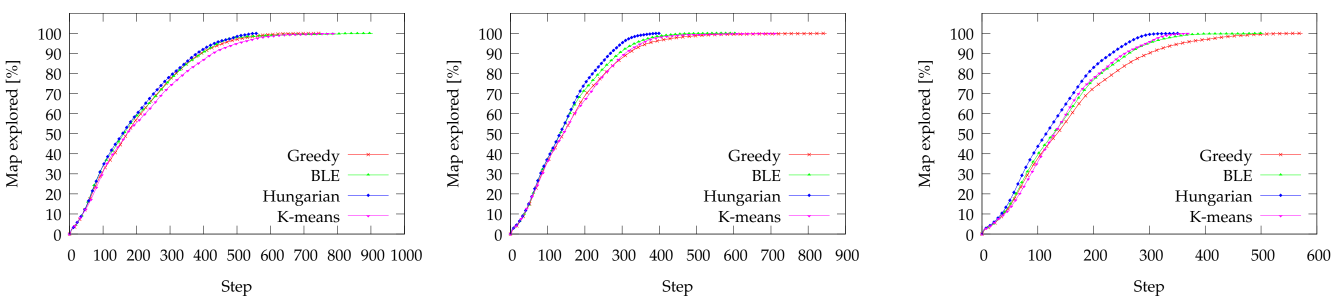
(a) (b) (c)

Fig. 3: *Empty* map size progress: (a) for 4 robots, (b) for 6 robots, (c) for 8 robots.



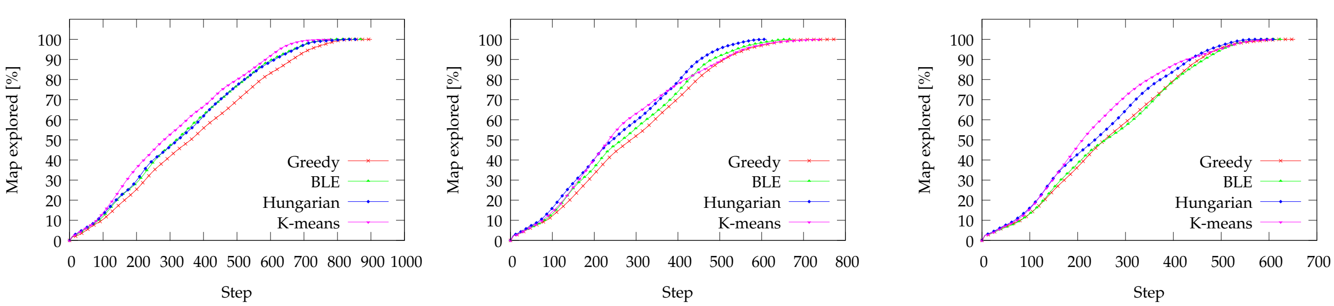
(a) (b) (c)

Fig. 4: *Arena* map size progress: (a) for 4 robots, (b) for 6 robots, (c) for 8 robots.



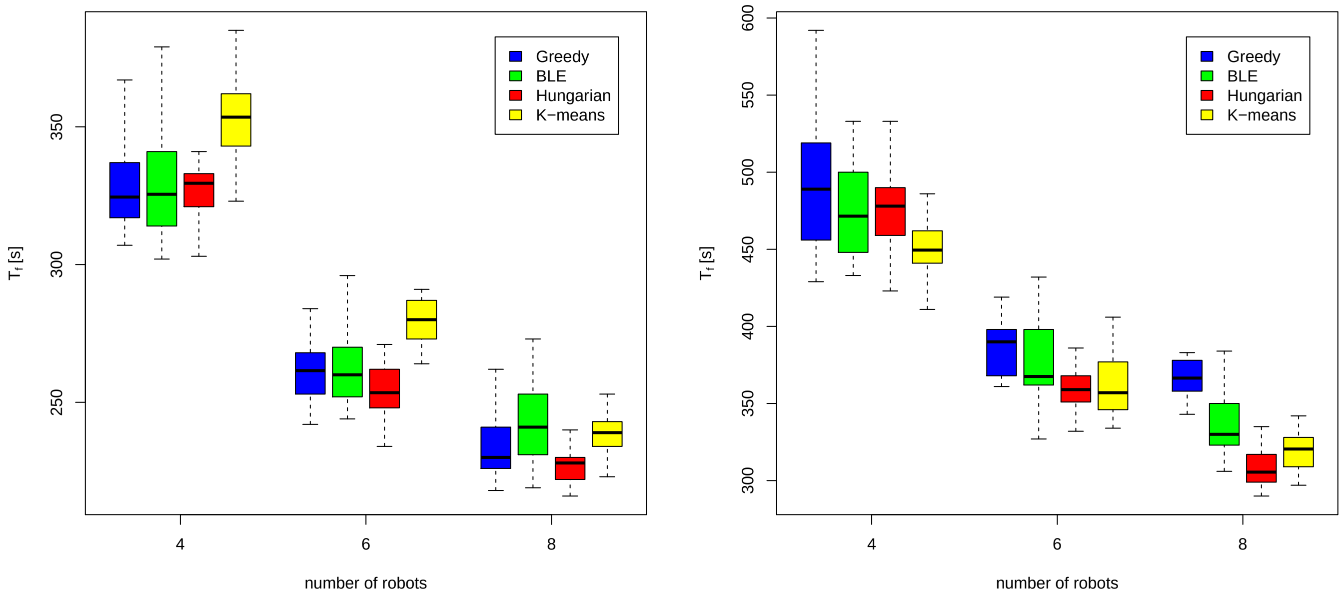
(a) (b) (c)

Fig. 5: *Jh* map size progress: (a) for 4 robots, (b) for 6 robots, (c) for 8 robots.



(a) (b) (c)

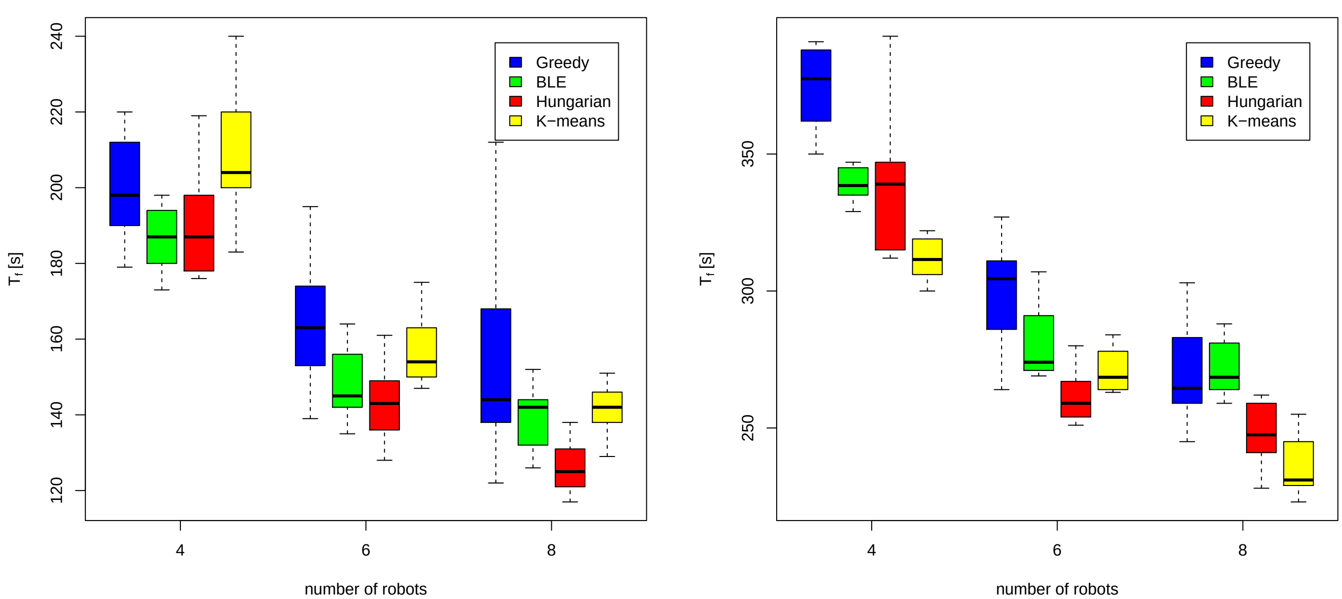
Fig. 6: *Hospital* map size progress: (a) for 4 robots, (b) for 6 robots, (c) for 8 robots.



(a)

(b)

Fig. 7: Comparison of the strategies, i.e. the five-number summaries of T_f (a) for *empty* and (b) for *arena* map.



(a)

(b)

Fig. 8: Comparison of the strategies, i.e. the five-number summaries of T_f (a) for *jh* and (b) for *hospital* map.