This is a preprint version of the article: Kulich, M., Přeučil, L., and Miranda Bront, J. J. (2014). **Single robot search for a stationary object in an unknown environment**. *ICRA2014: Proceedings of 2014 IEEE International Conference on Robotics and Automation*, which has been published in a final form at https://doi.org/10.1109/ICRA.2014.6907716.

Single Robot Search for a Stationary Object in an Unknown Environment

Miroslav Kulich[1], Libor Přeučil[1] and Juan José Miranda Bront[2]

***Abstract*— In this article we introduce the problem of finding an optimal path in order to find a stationary object placed in the environment whose map is not a-priory known. At first sight the problem seems to be similar to exploration which has been thoroughly studied by the robotic community. We show that a general framework for search can be derived from frontier-based exploration, but exploration strategies for selection of a next goal to which navigate a robot cannot be simply reused. We present three goal selection strategies (greedy, traveling salesmen based, and traveling deliveryman based) and statistically evaluate and discuss their performance for search in comparison to exploration.**

I. Introduction

Single-robot search, similarly to exploration, can be understood as a process of autonomous navigation of a mobile robot in an unknown environment in order to find an object of interest. The search algorithm can be formulated 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. A natural condition is to perform the search with an expected minimal usage of resources, e.g., trajectory length, time of search, or energy consumption.

While the research of exploration by a single or multiple mobile robots has been quite intensive (see e.g. [1], [2], [3], or our previous research [4], [5]), the search problem has been addressed marginally by the robotic community. On the other hand, the general structure of the search problem is the same as that of the exploration problem. The only difference lies in goal selection as the objectives of search and exploration are dissimilar, i.e a trajectory that is optimal in exploration does not necessarily minimize the expected value of the time to find an object along it (see Fig 1).

Some effort has been devoted to the single and multi-robot search problem for a-priori known environments which can be straightforwardly used as a goal selection strategy in the iterative procedure of the search problem in an unknown space. Sarmiento et al. [6] 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 [7]) 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 [8], where robot control is assumed in order to generate smooth and locally optimal trajectories. Hollinger et al. [9] 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.

Operational research formulates single-robot search in known environments as Traveling Deliveryman Problem (TDP), which is known to be NP-hard [10]. Recently, several approximation algorithm were presented. Salehipour et al. [11] present a meta-heuristic combining General Randomized Adaptive Search (GRASP) with Variable Neighborhood Descent (VNS), meta-heuristic called GVNS (General Variable Neighborhood Search) is introduced in [12], while integer linear programming is used in [13].

In this paper we pursue two objectives: (1) *to formally formulate the search problem* in a-priory unknown environment and *to define the criterion to be optimized* and (2) *to discuss behavior of several goal-selection strategies*, two of them already utilized for exploration, while the third one is based on approximation algorithm of TDP and designed especially to be applied for a search scenario. In other worlds, the paper does not aim to introduce novel methods that perfectly solve the presented problem (although the TDP-based strategy is new), instead it tries to show specifics of the search task and to highlight distinction of search and exploration.

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

Suppose a mobile robot 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 the process of robot's navigation through this environment in order to find a stationary object placed randomly in

*This work has been supported by the Technology Agency of the Czech Republic under the project no. TE01020197 "Centre for Applied Cybernetics" and Czech Ministry of Education, Youth and Sports under the project no. 7AMB14AR015 "Multi-Robot Autonomous Systems".

[1] Miroslav Kulich, Libor Přeučil are with Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27 Prague 6, Czech Republic{kulich, preucil}@labe.felk.cvut.cz

[2] Juan José Miranda Bront is with Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina jmiranda@dc.uba.ar

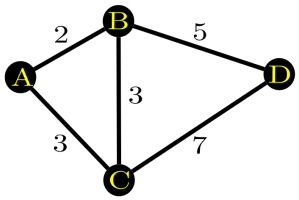


Fig. 1. Distinction of exploration and search. Given the robot starts at node A and the numbers represent the times to traverse the corresponding edges, the optimal exploration solution is $ACBD$ (the time to perform it is 3+3+5=11, while the time to traverse the path $ABCD$ is 2+3+7=12). On the other hand, the best solution for search is $ABCD$: the time to visit B is 2, C is 2+3=5, and D is 2+3+7=12, so the average time of visiting a node is $(2 + 5 + 12)/3 = 6.33$. For the path $ACBD$ we get: time to visit C is 3, B is 3+3=6, and D is 3+3+5=11, so the average time is $(3 + 6 + 11) = 6.66$.

the environment[1]. By finding an object we understand the situation when it is firstly detected by robot's sensors. A natural condition is to minimize the time when this situation occurs. More formally, this condition can be expressed as minimization of the expected (mean) time T the object is firstly detected, when the robot follows the trajectory R:

$$\mathbb{E}(T|R) = \int_0^\infty tp(t)\ \mathrm{d}t, \tag{1}$$

where $p(t)$ is the probability of finding the object at time t.

Sensing as well as planning is performed in discrete times, therefore (1) can be rewritten as

$$\mathbb{E}(T|R) = \sum_{t=0}^{\infty} tp(t), \tag{2}$$

where $p(t) = \frac{A_t^R}{A_{total}}$ is the ratio of the area A_t^R newly sensed at time t when the robot follows the trajectory R and A_{total}, the area of the whole environment the robot operates.

The objective is to find the trajectory R^{opt} minimizing (2):

$$R^{opt} = \arg\min_R \mathbb{E}(T|R) = \arg\min_R \sum_{t=0}^{\infty} tA_t^R.$$

III. Framework

The framework for single-robot search is based on Yamauchi's frontier based approach [1] successfully used for exploration, which uses an occupancy grid as the environment representation. The key idea of the approach is to detect *frontier cells*, i.e. reachable grid cells representing free regions adjacent with at least one no 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 is an iterative procedure that is terminated once the object to be found is detected or all the reachable space has been searched. We assume that the searched object lies entirely in one grid cell and it is detected when the cell is within visibility range of the sensor. At each step of the algorithm, the map is updated and frontiers are detected. After that, the most appropriate frontier cell is selected as the new robot goal and the robot is navigated towards it, see Algorithm 1.

[1]In general, a-priory information about object's position can be given in the form of a probability density function, but a uniform distribution is expected in the paper.

Algorithm 1: Frontier based search.

repeat
- Get the updated map built from sensor readings;
- Detect all frontiers from the actual map;
- Select representatives (goal candidates);
- Determine the next goal from the set of goal candidates;
- Plan a path to the next goal;
- Perform the plan;

until *the object found*;

Determination of the next goal is performed in two steps. At first, frontier cells are filtered to get a set of representatives approximating the frontier cells such that each frontier cell is detectable by the robot sensor from at least one representative. This is done by k-means clustering for each frontier, where the representative of each cluster is the closest frontier cell to the cluster's mean. The number of clusters in a frontier is defined similarly to [5] as

$$n_f = 1 + \lfloor \frac{N_f}{1.8\rho} + 0.5 \rfloor,$$

where N_f is the number of cells forming the frontier and ρ is the sensor range. This guarantees that all frontier cells will be explored after visiting all representatives (goal candidates). Moreover, reduction of goal candidates dramatically decreases computational burden of more sophisticated and time-consuming goal-selection strategies and therefore allows their usage for non-trivial environments.

In the second step, the most suitable goal candidate according to the defined criteria is determined.

IV. Goal selection strategies

The key component of the search algorithm and the only part different from exploration is selection of the next goal candidate to be visited, i.e. goal selection strategy. We present three strategies, two of them were already presented within the concept of exploration, while the third one is designed especially for search.

A. Greedy approach

The idea of the greedy approach proposed for exploration in [1] is to optimize the immediate next payoff by selecting the candidate that is reached with the lowest cost, i.e. the candidate nearest to the current robot's position. Although this approach does not reflect the effect of actions beyond the next step, it is used in many exploration implementations.

B. Traveling salesman strategy

Contrary to the greedy approach, which plans one-step ahead, the other two strategies work with a larger planning horizon assuming that all the goal candidates will be visited.

The problem then lies in determination of the order in which the candidates will be visited. Traveling salesman strategy formulates this task as the problem of finding the shortest Hamiltonian path in the complete graph of all goal candidates: given the robot position s_0, the set of goal candidates $\mathcal{S} = \{s_1, \ldots, s_n\}$, and $d(a, b)$ the length of the path between cells a and b, the aim is to find the permutation $\Pi = \{\pi_1, \ldots, \pi_n\}$ of indexes $\mathcal{I} = \{1, \ldots, n\}$ such that

$$C^{TSP}(\Pi) = d(s_0, s_{\pi_1}) + \sum_{i=1}^{n-1} d(s_{\pi_i}, s_{\pi_{i+1}})$$

is minimal over all permutations of $\mathcal{I}$. The goal candidate s_{π_1} is then selected as the next robot goal.

The problem of finding the Hamiltonian path of the minimal length (HPML) is known to be NP-hard [14], so finding the optimal solution can be computationally demanding. On the other hand, there exist many approximation algorithms. One of the most powerful ones is the Chained Lin-Kernighan heuristic, which gives near-optimal results (up to 1% of the optimum) in a reasonable time [15] for a similar problem - the Traveling salesman problem (TSP). In our previous work [4] we present a straightforward conversion of HPML to TSP and show that the TSP-based approach significantly outperforms the greedy selection in the exploration scenario.

C. Traveling deliveryman strategy

Motivated by a good performance of the TSP approach for exploration, we introduce a similar method for search. Again, the most appropriate order of the detected candidates is determined, which minimizes the expected time to find the object as defined in (2).

Given the graph $G(V, E)$, where V stands for the set of goal candidates, and E is the set of all shortest paths between the goal candidates and assuming that (1) $p(t)$ is the same for all the candidates and (2) time needed to reach a candidate is proportional to the distance to the candidate (note that precise determination of this time is a complex and time consuming task)[2], we can derive:

$$\Pi^{opt}_{G(V,E)} = \arg\min_{\Pi} \sum_{k=0}^{n} t_{\pi_k} = \arg\min_{\Pi} \sum_{k=0}^{n} d_{\Pi}(k), \qquad (3)$$

where Π is a permutation of indexes and $d_{\Pi}(k)$ is a length of the trajectory from s_0 to s_{π_k} through all goal candidates s_{π_j}, $0 < j < k$:

$$d_{\Pi}(k) = \sum_{j=1}^{k} d(s_{pi_{j-1}}, s_{pi_j}) \qquad (4)$$

The equation (3) is a formulation of the Traveling deliveryman problem (TDP) [11] which finds an open tour minimizing the total waiting time of all customers to be visited. Although TSP and TDP have similar formulations, they are in fact different and an optimal solution for one problem is generally not a good solution for the second one [11], see also Fig. 2.

[2]Both these assumptions are really strong and don't hold in majority of cases. Consequences of violation of these conditions will be discussed later.

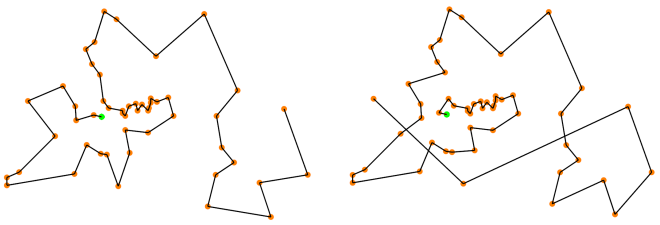


Fig. 2. Illustration of the difference between TSP and TDP on berlin52 problem from TSP Library [16]. Tours start at the green point. Left: the solution found by the Chained Lin-Kernighan heuristic has $C^{TSP} = 7305.72$ and $C^{TDP} = 166573$. Right: the solution found by our implementation of GVNS [12], $C^{TSP} = 8534.29$ and $C^{TDP} = 134852$.

No fast approximation algorithm giving near-optimal results exist for TDP, therefore we propose a depth-first search on $G(V, E)$. In order to speed-up the algorithm, several improvements have been made:

- Only n nearest (using Euclidean distance) goal candidates form the graph.
- Only α nearest (using the distance computed by Dijkstra's algorithm) neighbors for each node are expanded, i.e. the branching factor of the algorithm is α.
- The sums in the equations (3) and (4) can be rewritten as:

  $$C^{TDP}(\Pi) = \sum_{k=1}^{n} (n - k + 1) d(s_{pi_{j-1}}, s_{pi_j}),$$

  so the contribution of each edge/node to the cost function can be determined when the node is expanded and thus processing of one node during search is constant.

In our implementation, we set $n = 15$, $\alpha = 10$, which allows to compute the depth-first search algorithm in milliseconds on a standard computer.

V. Experiments

Performance of the proposed strategies has been evaluated in three types of environments in simulations using the Player/Stage framework [17]. The first environment (*empty*) is a space without obstacles, the second one (*potholes*) represents an empty area with several obstacles, and the *jh* environment is a map of a part of a real administrative building. All the environments were scaled so the represent an area of 24×21 m and they are visualized in Fig. 3.

All experiments were performed within the same computational environment: a workstation with the Intel®Core i7-3770 CPU at 3.4 Ghz running OS Sabayon with the Linux kernel 3.7.0. The algorithms have been implemented in C++ as Player client programs. Simulation of the S1R robot equipped with Hokuyo URG-04LX [18] with 270° field of view has been used as the robotic platform, while the occupancy grid with cell size 0.05×0.05 m has been chosen to represent the working environment. Smooth Nearnest Diagram (SND) algorithm [19] implemented in the Player

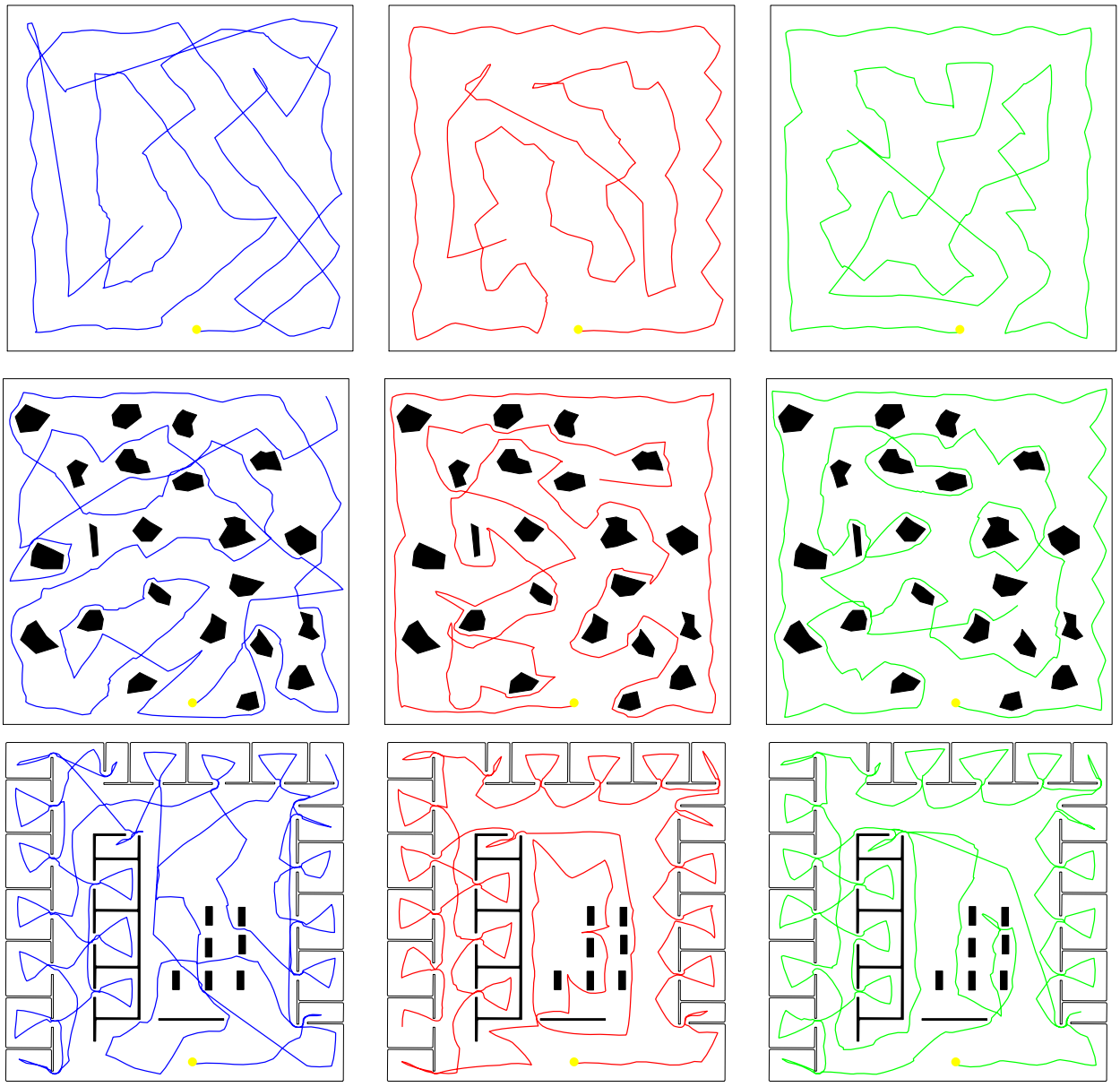


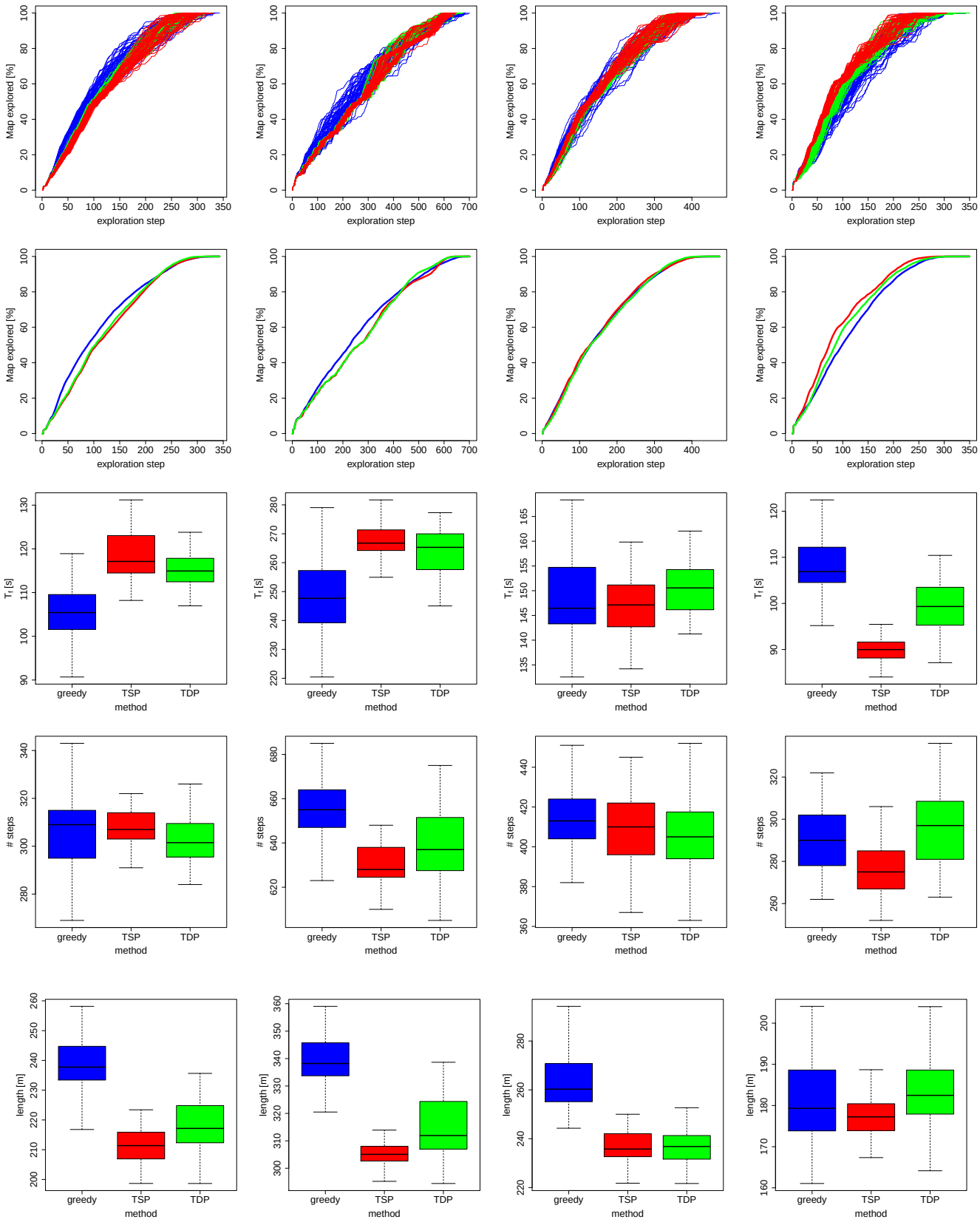
Fig. 3. The testing environments and the best results found given the sensor range $\rho = 2\ m$. From the top: *empty* environment (from the left) $T_f^{greedy} = 90.69$, $T_f^{TSP} = 108.20$, $T_f^{TDP} = 106.97$, *potholes* environment: $T_f^{greedy} = 132.55$, $T_f^{TSP} = 134.20$, $T_f^{TDP} = 141.25$, *jh* environment: $T_f^{greedy} = 220.44$, $T_f^{TSP} = 254.95$, $T_f^{TDP} = 245.03$. The robot starts at the black point in the bottom part of the map.

has been used to control the robot motion and to avoid obstacles. The TSP solver used is the Chained Lin-Kernighan heuristic from the Concorde package [14].

The goal selection algorithm (i.e. the body of the loop in Algorithm 1) is run every 1500 ms. It means that a new goal can be selected before the old one is reached. Moreover, the sensor range has been limited to 2, 3, and 5 meters. For each experimental setup consisting of the map, and the range, 50 runs have been performed for all the approaches.

The best solutions found during experiments together with the expected time of finding the object (T_f) for the sensor range $\rho = 2\ m$ are depicted in Fig. 3 to illustrate behavior of the strategies. In general, the TSP approach explores the environment systematically: it starts with going around the environment border, which is nicely seen in Fig. 3 especially for the *jh*, where the rooms are visited first. Also the number of crossings is low, as the method tries to minimize the total length traveled, which is better without crossings. The similar behavior is observed for the TDP, although the generated trajectories are more chaotic. On the contrary, it is not possible to pick out any intention in the behavior of the greedy approach (GA). The robot navigated by this strategy returns to already visited places as it "forgets" to explore places nearby. For example, this situation is visible on the right-bottom corner of the empty map in Fig. 3.

Statistical evaluation of the strategies according to several criteria is shown in Fig. 4: progress of the amount of the explored area and the expected time of finding the object (T_f)



(a) empty (b) jh (c) potholes (d) potholes ($\rho = 3$ m)

Fig. 4. Comparison of the strategies (greedy in blue, TSP in red, and TDP in green). The first three columns show results for the *empty*, *jh*, and *potholes* maps and the visibility range $\rho = 2$, the last column shows results for the *potholes* map and $\rho = 3$. The first row depicts a progress of an explored area for all runs, the second row the same averaged over the runs, and the other rows display the five-number summaries of T_f, the number of exploration steps, and the total length traversed.

describe the behavior from the search perspective, while the number of exploration steps (i.e. the time in which the whole area is explored) and the total length of the traversed path are there for comparison as they are relevant to exploration.

Due to the limited space, only statistics for $\rho = 2\ m$ and for the *potholes* environment with $\rho = 3\ m$ are presented. The results for other cases are similar to the ones presented.

Not surprisingly (see also [4]), the described methods' behavior causes that both the TSP and TDP generate about 10-15% shorter trajectories than the GA in average, see the fifth row of the Fig. 4. The TSP and TDP outperform the GA in the total time needed to search/explore the whole environment, although the difference is not so big as for the lengths. This discrepancy is caused by robot control: while the robot moves fast in free spaces, it takes time to navigate in narrow corridors. Therefore, the fact that the robot navigated by the GA has to return to distant places (fast in an empty space) is not so crucial comparing the time needed to turn or pass doors. The solution to this is to consider the real time needed to reach the goal in the objective function, instead of approximate it by a path length.

Performance of the strategies according to the main criterion, the mean time to find an object, is presented in the third row of the Fig. 4. The GA outperforms the other methods in the *empty* and *jh* environments. The GA forces the robot into an empty space and because the robot moves fast in an empty space, the environments are searched rapidly. By contrast, the TSP and TDP waste time at the beginning of the mission by inefficiently exploring small pieces of the space, i.e. by exploration of *empty* map borders or offices in the *jh* map. This is clearly shown in the first and second rows of the Fig. 4 for *empty* and *jh* maps, where the amount of searched area at the beginning grows significantly faster for the GA than for the TSP and TDP. Although the situation is opposite in the final phases (as the GA returns to "forgotten" places), advantage of the GA from the beginning is enough to outperform the other approaches in T_f.

Interesting results were measured for *potholes* environment and $\rho = 3\ m$ for which both TSP and TDP perform better than the GA. Obstacles in the environment are placed so that variance of areas sensed from various positions is small, which corresponds with the assumption (1) in section IV-C that the probability of sensing the object is equal for all places. Therefore, these methods give good results.

The results show that incorporating the gain of visiting the goal is crucial for designing effective strategies. This is different to exploration, for which strategies can be efficient assuming distance cost only. On the other hand, incorporating the gain into search strategies is straightforward as it is expressed as $p(t)$ in the Eq. 2, contrary to exploration, for which no consistent mathematical derivation exist.

VI. Conclusion

The problem of searching for a stationary object in apriory unknown environment is formulated and several strategies for this problem based on a distance cost only are presented and statistically evaluated. The key result is that contrary to exploration, sophisticated methods have to incorporate the gain of visiting the goal to outperform the greedy strategy. The second reason for strategies' ineffectiveness lies in inappropriate approximation of the time needed to reach a goal by a distance to the goal. Study of these two issues and design of strategies taking them into account will be subject of the future work. Furthermore, it is presupposed that the probability of finding the object is equal for all places. Generalization of the problem assuming various probabilities will be therefore also researched.

References

[1] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. of the Second International Conference on Autonomous Agents*, 1998, pp. 47–53.

[2] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for searching environments in rescue operations," *Autonomous Robots*, vol. 31, no. 4, pp. 401–417, 2011.

[3] K. S. Senthilkumar and K. K. Bharadwaj, "Multi-robot exploration and terrain coverage in an unknown environment," *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 123–132, 2012.

[4] M. Kulich, J. Faigl, and L. Preucil, "On distance utility in the exploration task," in *Robotics and Automation (ICRA), 2011 IEEE Int. Conf. on*, 2011, pp. 4455–4460.

[5] J. Faigl, M. Kulich, and L. Preucil, "Goal assignment using distance cost in multi-robot exploration," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conf. on*, oct. 2012, pp. 3741 –3746.

[6] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson, "A multi-robot strategy for rapidly searching a polygonal environment," in *Advances in Artificial Intelligence - IBERAMIA 2004, 9th Ibero-American Conference on AI, Puebla, México, November 22-26, 2004, Proceedings*, ser. Lecture Notes in Computer Science, C. Lematre, C. A. Reyes, and J. A. Gonzlez, Eds., vol. 3315. Springer, 2004, pp. 484–493.

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

[8] 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.

[9] 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.

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

[11] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy, "Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem," *4OR*, vol. 9, pp. 189–209, 2011.

[12] N. Mladenović, D. Urošević, and S. Hanafi, "Variable neighborhood search for the travelling deliveryman problem," *4OR*, pp. 1–17, 2012.

[13] 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.

[14] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. (2010, Sep.) Concorde TSP Solver. [Online]. Available: http://www.tsp.gatech.edu/concorde.html

[15] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems," *Informs J. on Computing*, vol. 15, no. 1, pp. 82–92, 2003.

[16] G. Reinelt. (2013, Apr.) TSP Library. [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[17] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proc. of the 11th Int. Conf. on Advanced Robotics*, 2003, pp. 317–323.

[18] M. Kulich, J. Chudoba, K. Kosnar, T. Krajnik, J. Faigl, and L. Preucil, "Syrotek – distance teaching of mobile robotics," *Education, IEEE Transactions on*, vol. 56, no. 1, pp. 18–23, 2013.

[19] J. W. Durham and F. Bullo, "Smooth nearness-diagram navigation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008, pp. 690–695.