

Evaluation of different frontier-based multi-robot exploration strategies

Abdenour Benkrid^{1,a}, Abdelaziz Benallegue² and Noura Achour¹

¹ LRPE Laboratory, USTHB University, B.P 32 El Alia 16111 Bab Ezzouar Algiers, Algeria

² Versailles Engineering Systems Laboratory (LISV), 10/12 Avenue de l'Europe 78140 Velizy, France

Abstract. In this paper, we focus on the problem of exploring an unknown environment by a team of mobile robots. The main objective is to compare four different coordination strategies based on frontier concept (boundaries between unexplored and explored open areas) and analyze their performance in term of assignment quality, overall exploration time and computational complexity. In order to provide a suitable qualitative study we used three optimization criteria. Each strategy has been implemented and tested extensively in computerized simulation.

1 Introduction

One of the major challenges in autonomous mobile robotics is the exploration of an unknown environment, the goal of exploration strategies is to distribute robots in order to cover areas within a minimal time. There are many real world situations that involve exploration algorithms to build a map of inaccessible or dangerous environments for humans, like oceanic exploration [1], planetary exploration [2] and other applications such as cleaning [3] and rescue [4].

The main idea behind the exploration problem is that robots must decide where to go for getting as much new information as possible in order to extend their representation to unknown areas. One of the famous solution is given by Yamauchi [5] who introduced the concept of Frontier, which is the limit between unexplored and explored accessible areas. During an exploration process a robot always moves to the frontier in order to get new information about the world, and so extends his map at the new frontiers. The exploration is complete when there aren't frontiers to be discovered.

For exploring an unknown environment, we use multiple mobile robots. This choice is justified by the fact that the collaboration of several simple and not expensive robots, proves to be faster and more efficient than using a large complex robot [6]. Furthermore, it has been proved that as the different perceptions resources are shared and merged, the reliability of information is increased in addition to the accuracy of the model representing the covered areas [7]. On the other hand, the failure or the loss of one or more robots does not necessarily mean the interruption of the exploration task. However, all of these advantages are conditioned by the choice of an appropriate coordination strategy that assigns each robot to a specific target while minimizing the risk of collisions between members of the team as well as the risk of cross-talk between their sensors [8].

This work investigates several frontier-based multi-robot exploration strategies in simulation. Each strategy differs in term of appropriate target location choice and the method that coordinates the team of robots. To simplify the set-up we consider in simulation, that all robots are identical and can locate themselves independently and build a map with ideal measurements. They can also share and merge their local map without communication problems. Hence, only coordination problem of actions between robots is considered.

This paper is organized as follows. The next section defines the problem and introduces the optimization criteria used throughout the paper. The section 3 describes the primary exploration strategies used in this work while providing the algorithms with their computational complexity and a brief study about assignment quality. In Section 4 and 5, we analyze the performances of exploration algorithms in terms of overall time required to cover the whole environment by the robots. Section 6 is a conclusion.

2 Problem formulation

The explored environment is a finite space in which is deployed a team of mobile robots equipped with sensors that allow them to build a local map of environment and to localize themselves in it. In this paper we use an occupancy grid to represent the environment [9] in which the frontiers cells to be explored are identified. The problem studied in this paper is to affect each robot of the team a frontier to explore, which means, get closer to it for extending the representation of the environment.

To evaluate these affectations, it is imperative to know the cost to moving the robots to the frontiers. This cost is calculated using A* algorithm, the result is a cost matrix associating frontiers and robots.

^a Abdenour Benkrid: abenkrid@usthb.dz

Before presenting the assignment criteria we precise the notations used in this paper.

2.1 Notations

- **R** : Represents a set of robots (r_1 to r_n) with n the number of robots.
- **F** : The set of frontiers (f_1 to f_m) with m the number of frontiers.
- **C** : Cost matrix with c_{ij} the cost associated with assigning robot r_i to frontier f_j .
- **A** : Assignment matrix with $a_{ij} \in [0,1]$ computed as follows:

$$a_{ij} = \begin{cases} 1 & \text{if the robot } r_i \text{ is assigned to the frontier } f_j \\ 0 & \text{otherwise} \end{cases}$$

2.2 Evaluation criteria of an assignment

The affectation process is performed in each iteration or extension of map representing the environment. The quality of assignation is evaluated according to the three criteria detailed after:

- For deploying the robots in the environment we affect to each robot a frontier to explore, this is mathematically expressed as follows.

$$\forall i \sum_{j=1}^m a_{ij} = 1 \quad (1)$$

- Equilibrium of the distribution of the robots on the frontiers: regardless the number of robots used to explore the environment they must be distributed in a balanced manner on the frontiers. This affectation criterion can be resumed by.

$$\lfloor n/m \rfloor \leq \forall j \sum_{i=1}^n a_{ij} \leq \lceil n/m \rceil \quad (2)$$

- Minimum of the cost sum: the minimisation of overall exploration cost is determined by the minimisation of the cost sum.

$$C(A) = \sum_{i=1}^n \sum_{j=1}^m a_{ij} c_{ij} \quad (3)$$

3 Primary exploration strategies

After explaining the problem of exploration and introducing the three evaluation criteria which allow to determine the quality of an assignment, we itemize in this section the primary exploration strategies analysed in this paper. Each will be explained in short together with the quality of its assignment.

3.1 Nearest frontier

Introduced by *Yamauchi* in 1997 for a single robot, this strategy was extended to multiple robots in 1998 [10]. It is also one of the most popular frontier-based exploration strategy.

In this method, each robot broadcasts the collected information of a covered area to another team member, all the other members build a similar representation of the world with the same list of frontiers. The robots move to the nearest frontiers, make new observations and share the results again. The exploration process ends when the entire environment is explored (no frontiers are left).

Even if the information between robots is shared, the decision about where to move is independent. No communication is necessary to coordinate robots. This approach is known to be completely asynchronous (robots are never waiting for other robots), distributed and remain robust to robot loss.

Algorithm 1. Nearest frontier

Input: C_i Cost Matrix of r_i

Output: a_{ij} assignment of robot r_i to frontier f_j

Begin

Find a frontier j which satisfies:

$$j = \arg \min(c_{ij}) \quad \forall f_j \in \mathbf{F}$$

$$a_{ij} = 1$$

end

With its complexity of $O(m)$, this algorithm is considered as a first significant solution to the multi-robot exploration problem, it is optimal to the criteria given by equations 1 and 3, but doesn't respect the criterion given by equation 2. As a consequence, it presents the problem of redundancy in term of frontiers assignment to the robots. In fact, since the control is independent, robots may waste time by navigating to the same frontier.

3.2 Greedy assignment

This approach tries to optimize the total cost of exploration (see equation 3). At each iteration of the allocation loop, greedy algorithm selects the robot-frontier pair with a lowest cost, and both are removed from their respective lists. This process is repeated until all the robots have an assigned frontier.

This algorithm is generally applied in a centralised way and synchronous, but if each robot has the same information about the world including the other robots location, it can compute the affectations of the other members until it finds own assignation. In this paper we present a decentralised and asynchronous version of the greedy allocation algorithm (see Algorithm 2), as explained previously where each robot broadcasts, receives perception and builds a global representation of the world.

Algorithm 2. Greedy assignment (decentralised)

Input: r_i, C Cost Matrix

Output: a_{ij} assignment of robot r_i to frontier f_j
While r_i has no frontier assigned **do**

 Find a robot k and frontier j which satisfy:

$$(k, j) = \arg \min(c_{kj}) \quad \forall r_k \in \mathbf{R}, \forall f_j \in \mathbf{F}$$

$$a_{kj} = 1$$

$$\mathbf{R} = \mathbf{R} \setminus r_k, \mathbf{F} = \mathbf{F} \setminus f_j$$

if $\mathbf{F} = \{ \}$ *then* $\mathbf{F} = \mathbf{F}_{init}$
end

The complexity of this algorithm is $O(n^2m)$. As it tries to optimize the total cost of exploration it respects the criterion 3. The criteria 1 and 2 are also satisfied.

We notice that the majority of frontier-based approaches of the literature are based on greedy algorithm [11].

3.3 Greedy with utility assignment

Simmons et al [12] introduce in standard greedy algorithm the concept of utility which represents the expected gain of information from a target location. This approach takes into account the difference between estimated information and the cost of reaching a target region. Whenever a robot moves to a specific target point, the utility of the unexplored areas visible by the robot sensors are reduced. The greedy with utility algorithm is similar to the previous algorithm, with updating the utility parameter [13].

Algorithm 3. Greedy with utility assignment

Input: r_i, C Cost Matrix

Output: a_{ij} assignment of robot r_i to frontier f_j

 Set the utility U_j of all frontiers to 1

While r_i has no frontier assigned **do**

 Find a robot k and frontier j which satisfy:

$$(k, j) = \arg \max(U_{j'} - \beta \cdot c_{kj}) \quad \forall r_k \in \mathbf{R}, \forall f_j \in \mathbf{F}$$

$$a_{kj} = 1; \mathbf{R} = \mathbf{R} \setminus r_k, \mathbf{F} = \mathbf{F} \setminus f_j$$

 Reduce the utility of each frontier j' in the visibility of the robot sensors :

$$U_{j'} \leftarrow U_{j'} - D(\|j - j'\|)$$

if $\mathbf{F} = \{ \}$ *then* $\mathbf{F} = \mathbf{F}_{init}$
end

The value of β determines the weight of frontier utility compared to the cost needed to reach it.

3.4 Minimum position (MinPos)

Bautin et al [14] proposed a new frontier allocation method, which not only takes into account the distance between robots and frontiers, but also the position of a robot towards a frontier. This position is determined by counting how many robots are closer to the frontiers than the considered one. To compute the cost matrix *Bautin* introduced a wavefront propagation algorithm [15] from each frontier. This method gives an interesting alternative to path planning used in previous strategies.

Algorithm 4. MinPos

Input: r_i, C Cost Matrix

Output: a_{ij} assignment of robot r_i to frontier f_j
For each $f_j \in \mathbf{F}$ **do**

$$P_{ij} = \sum_{\forall r_k \in \mathbf{R}, k \neq i, c_{kj} < c_{ij}} 1$$

end

 Find a frontier j which satisfies:

$$j = \arg \min(P_{ij}) \quad \forall f_j \in \mathbf{F}; a_{ij} = 1$$

if there is an equality choose the minimum cost among $\min P_{ij}$

The complexity of the MinPos algorithm is $O(nm)$, it also respects the criteria given by equations 1 and 2, but does not always respect the criterion given by equations 3.

4 Simulation

To analyzing the performance of the previously mentioned algorithms, we realised several simulations using MATLAB software.

These simulations allowed to simulate the exploration process in two different environments depicted in figure 1 which represent offices (Fig.1.a) and randomly generated mazes (Fig.1.b). This simulated environments are discretised into occupancy grids. Each cell of this grid is considered as a unit with resolution of 0.5mx0.5m and all the simulated environments previously described have a size of 25mx20m which represents 50 x 40 units. Furthermore, we assume that all robots are identical and have the same size as the cell of the grid, so, each of them represents one unit in the map. We also assume that their location is known with certainty and all of them can scan their neighbourhood at 360° around them inside a parameterised sensor radius which is fixed in our experimental simulation to 4 units. The inaccuracies during map generation are not considered. We predefine the maximum speed of the robots and fix it to 2 units

/second, we also assume a fully connected communication network between them. Hence, a local new information about the world is shared and stored in a common global map.

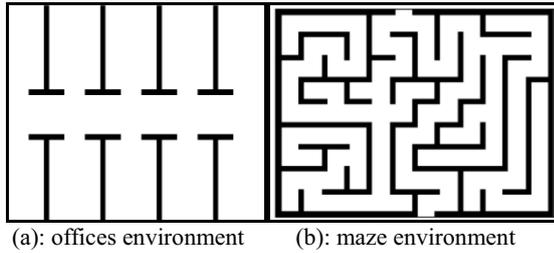


Figure 1. Environments used for the simulation experiments.

In each analyse of the four strategies, all of the cells are initially defined as unexplored and the robots started at the same randomly chosen location. Each grid cell which is in the course of simulation totally covered by the sensors of the robot will be marked as free or occupied. The objective of a simulation run was to completely explore the environment which means that all units should be classified as obstacle, accessible, or inaccessible cell. The inaccessible cell is an unexplored cell and considered unreachable for all the team members.

5 Results

In this work, the time in seconds needed to achieve the objective to explore all the environment is used as a main parameter to analyze the performance of the studied algorithms. For each strategy in each environment we evaluate the average of 50 values of different simulation experiments in seconds which the team of robots need to complete the exploration.

The results are presented according to the type of environment used in simulation (Fig.2, 3). As we can see, for all strategies, increasing the number of robots in order

to explore the environment is faster than using a single robot, in all environments used for our experimental study, Minimum Position (MinPos) and Greedy with Utility Assignment strategy are more performant than the two other strategies, especially when a number of member team is between 3 and 4. The performances of MinPos and Greedy with Utility Assignment are the same with a slight advantage for MinPos strategy when a number of robots is small, however, when this number increases, greedy with utility assignment have the better performances.

6 Conclusion

In this paper we addressed the problem of exploring an unknown environment with multiple autonomous mobile robots. Four different exploration strategies based on frontiers approach were studied and analyzed in terms of assignment quality, computational complexity and overall time needed to explore all unknown areas of the environment.

Performance measures in simulation demonstrated that minimum position (MinPos) and greedy with utility approaches are significantly more efficient in terms of total exploration time than nearest frontier or greedy assignment. To be more precise, minimum position strategy performs better than greedy with utility assignment when the number of robots is small, if this number grows the greedy with utility assignment has the best performances.

It's clear that we assumed in our work ideal measurements and thus neglected the localisation problem to simplify the setup, and supposed that robots can communicate between them anywhere in the environment without problem which is not the case in practice. Identifying the impact of these is left for future work.

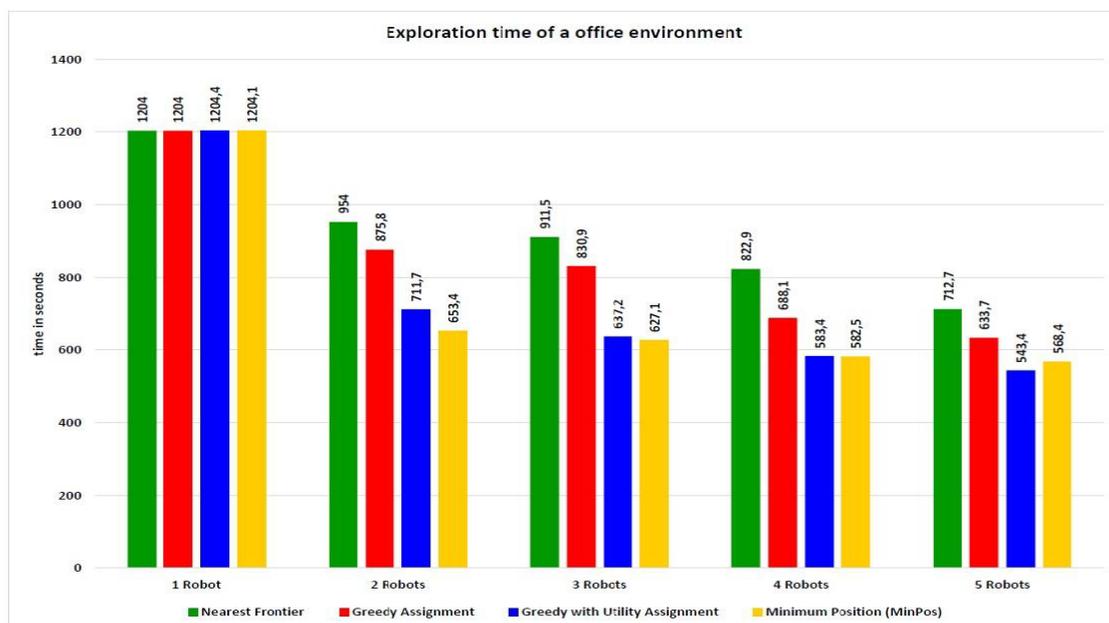


Figure 2. Time in seconds required to explore office environment.

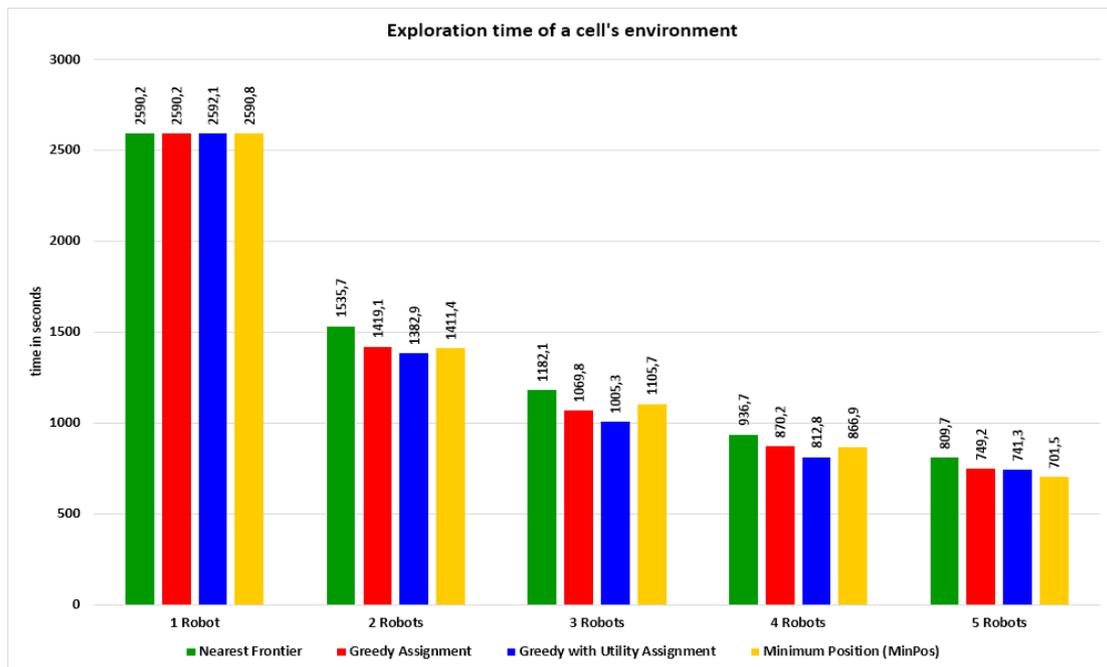


Figure 3. Time in seconds required to explore maze environment.

References

1. A. Pascoal, P. Oliveira, C. Silvestre, C. Sebastião, M. L. Rufino, V. Barroso, and P. Dando, *Oceans 2000 MTS/IEEE Conference and Exhibition*, **1**, 409-415 (2000)
2. D. S. Apostolopoulos, L. Pedersen, B. N. Shamah, K. Shillcutt, M. D. Wagner, and W. L. Whittaker, *IEEE International Conference on Robotics and Automation*, **4**, 4174-4179 (2001)
3. H. Endres, W. Feiten, and G. Lawitzky, *IEEE International Conference on Robotics and Automation*, 1779-1781 (1998)
4. R. Murphy, *IEEE Trans. Syst. Man Cybern. C Appl. Rev.*, 138-153 (2004)
5. B. Yamauchi, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 146-151 (1997)
6. O. Simonin, O. Grunder, *Mechatronics*, **19**, 463-470 (2009)
7. D. Fox, W. Burgard, H. Kruppa, and S. Thrun, *The 23rd German Conference on Artificial Intelligence*, 325-340 (1999)
8. D. Goldberg and M. Mataric, *Rob. Auton. Syst. J.*, **8**, 637-642 (1997)
9. A. Elfes, *Computer*, **22**, 46-57 (1989)
10. B. Yamauchi, *Proceedings of the second international conference on Autonomous agents*, 47-53 (1998)
11. W. Burgard, M. Moors, and F. Schneider, *IEEE Trans. Robot.*, **21**, 376-386 (2005)
12. R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, S. Thrun, and H. Younes, *Proceedings of the National Conference on Artificial Intelligence*, 852-858 (2000)
13. R. Zlot, A. Stentz, M. B. Dias, S. Thayer, *IEEE International Conference on Robotics and Automation*, 3016-3023 (2002)
14. A. Bautin, P. Lucidarme, R. Guyonneau, O. Simonin, S. Lagrange, N. Delanoue, and F. Charpillet, *5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 210-215 (2013)
15. J. Barraquand, B. Langlois, and J. C. Latombe, *IEEE Trans. Syst. Man Cybern.*, **22**, 224-241 (1992)