



AN EVALUATION OF MULTI-ROBOT SYSTEMS EXPLORATION ALGORITHMS

Khalil Mohamed¹, Ayman Elshenawy¹ and Hany Mohy Harb²

¹ Systems and Computers Engineering Department, Al-Azhar University, Cairo, Egypt

² Information Technology College, Dean, Misr University for science and Technology, Giza, Egypt.

ABSTRACT

Multi-robot systems exploration of an environment is an important process that most automated applications depend on. In this paper, common algorithms used to perform the exploration process of an unknown cell-based environment occupied by a set of obstacles using a set of identical robots are studied and their performance are compared depending on three metrics: the total explored area in the map, overall mission time, and the number of hops in the networked robots. The performance of these algorithms is evaluated for different environments and different team sizes using MRESim computer simulator.

Keywords: Multi-robot systems; multi-robot exploration; exploration algorithms.

1. INTRODUCTION

Multi-robot systems (MRS) are a set of mobile robots that are connected through a wireless sensor network to share the sensory information with reconfigurable sensing capabilities [1]. The main goal of MRS is to achieve a complete task simultaneously in a shorter time than a single robot. MRS are characterized by higher fault-tolerance, consolidation of the overlapped information, and the reduction of energy consumption which leads to longer communication time during the task achievement [2,3].

Recently, MRS have been used in several applications such as military applications, search and rescue, surveillance, cleaning, and mine clearing, etc. [4]. In such applications, robots should make a decision whether to perform new tasks or establish cooperative interactions to achieve their individual and collective goals [4,5].

Most of MRS applications depend primarily on the multi-robot exploration process to build an environment map. Multi-robot exploration process encounters several challenges that affect its production. These challenges are such as limitations in the environment that may force robots to move together and robot interference with each other due to missing shared information [5,6].

During multi-robot exploration task, each robot can plan their paths and coordinate their actions. A robot can individually explore different areas of the environment, but without any coordination, it may explore the same area explored by other robots, block other robots, and interpose other robots sensor readings. The absence of coordination in MRS leads to a waste of exploration effort and time [7].

Multi-robot coordination is an essential process of multi-robot exploration task and its performance is affected by its quality [6-8]. It is used to complete the overall task

assigned to the MRS, merge the obtained information by several robots, deal with limited communication, assign tasks to individual robots, specify a set of rules, interact to each individual robots, and overcome the interferences among the robots in order to complete the task efficiently [8,9].

There are many challenges are still present in the process of multi-robot exploration such as cooperation control, concurrent localization, mapping, collision avoidance, task planning, and communication among robots, coordination, navigation. As an example, Fig. 1 shows that three robots try to explore the environment and navigate to their targeted locations. While Robot 3 can navigate to its goal, ignoring the remaining robots, Robots 1 and 2 need to coordinate so as not to cross the narrow doorway simultaneously [9,10].

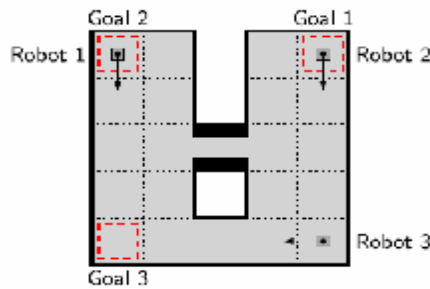
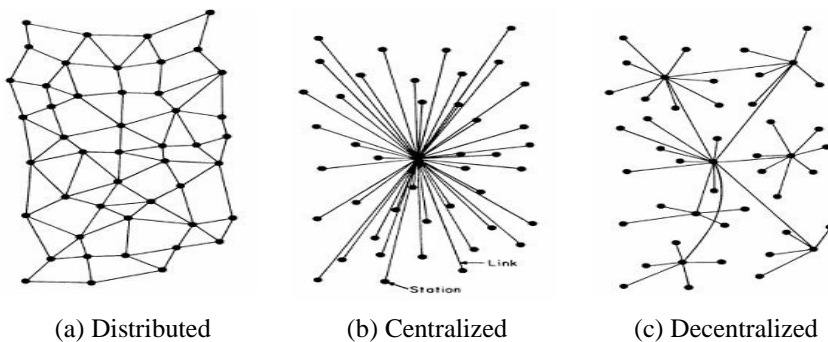


Figure 1 An example of simple navigation task

Task coordination in MRS has been divided into three main categories according to the architecture of the robots team as the following [10,11]:

- i. **Distributed Coordination:** as shown in Fig. 2-a, this category characterized by the absence of the central control, and all robots are completely autonomous in the decision-making process. Each robot can create its map, and exchange it with other robots in order to build a complete map [8,12,13].
- ii. **Centralized Coordination:** as shown in Fig 2-b, this category characterized by the existence of central control robot that communicate with all other robots in order to share the global information about the environment and the robots. It also responsible for generation of the map using the collected information. This architecture is working fast for small number of robots and becomes inefficient for large number of robots due to higher communication overhead, and produces a highly vulnerable system if the central control robot malfunctions [10,14-16].
- iii. **Decentralized Coordination:** as shown in Fig. 2-c, this category robots are organized into clusters with a local controller, and each cluster is responsible for performing sub-task individually. It provides more robust solutions and able to influence the entire team's actions through global goals and plans [11,12,17,18].



(a) Distributed

(b) Centralized

(c) Decentralized

Figure2 Different types of coordination hierarchy

In this paper, the coordination of MRS exploration is studied and a set of common recently used algorithms are presented and compared with a set of different team sizes and different environment structures.

The paper is organized as follows: the problem of exploring an unknown environment is described and formulated in section 2. The multi-robot exploration algorithms are discussed in section 3, a comparison between the performances of coordination strategies are showed and analysed in section 4, and finally, our work is concluded with a suggestion for future works which is presented in section 5.

2. MRS EXPLORATION OF UNKNOWN ENVIRONMENTS

MRS can be used to explore an unknown environment occupied by a set of obstacles. During the exploration process, MRS can gather information, acquire graphical representation, and explore and build the global map of an unknown environment by processing the local maps generated by each robot. MRS exploration task is ended after generating the environment map. It can be represented as graphs (Voronoi diagram, Visibility graph), cells (occupancy grids), polygons or trees (graph without cycles) [11,19]. The main items that affect the exploration process are the environment, obstacles, set of robots and the exploration of algorithm.

2.1 Environment

An environment is a finite two-dimensional space with boundary, and it can be represented as

- i. Grid or cell-based maps: is an environment divided into similar cells. Each cell may be unexplored cell, free cell, wall cell, or frontier cell [13, 20].
- ii. Feature or landmark maps, it extracts the features from the environment to represent the signature of the world [21,22].
- iii. Graph-based maps: is described as a graph consists of edges and vertices, it represents the environment as an abstract model with intersected and integrated paths [4,12],
- iv. Semantic maps, is an abstract model that contains the relationships and functions of the objects in the environment

This paper will concentrate on the cell-based map while performing the task.

2.2 Obstacles

Initially, the explored environment is occupied by a set of static obstacles distributed randomly with different shapes and positions.

2.3 Mobile robots

A team of homogenous or heterogeneous robots are used to perform the exploration task. These robots can move from the centre of one cell to another depending on some local information about the other robots or the other cells [1,11,23]. This paper will concentrate on the homogenous robots during the task.

2.4 Exploration algorithm

Exploring algorithm determines the shortest and the lowest-cost path of a robot based on the environment size and type, cell size, and the numbers of robots constitute the exploration team and update the environment map based on the gathered information [3,24]. A set of common exploration algorithms will be discussed in section 4.

3. PROBLEM FORMULATION

MRS exploration problem is formulated as follows: given a set of robots and a set of goals $g_j \in \{G_1, G_2, \dots, G_R\}$. Each robot $r_i \in R$ must travel $L_i \in \{l_1, l_2, \dots, l_M\}$ distance to reach the goal [10,11,25].

Finds the M optimal paths $\zeta^{opt} = (\zeta_i^{opt} | i = 1, 2, \dots, M)$, among sequences of N paths $\zeta = (\zeta_i | i = 1, 2, \dots, N)$ that minimize the travel distance such as:

$$\zeta^{opt} = \operatorname{argmin}_{\zeta} \mathbb{E}(T|\zeta) \tag{1}$$

where $\mathbb{E}(T|\zeta)$ is the expected meantime of the paths ζ and T is traverse time along these paths using the robot r_i .

The factors that affect the performance of exploration process depending on the environment parameters are:

- i. Environment complexity: there are some environment aspects that affect the environment complexity, such as the density of obstacle distribution, the number of walls in the environment and width of the walls [11,26].
- ii. The communication range among: the robots to exchange the information via communication channels such as sensor data and processed data. In this paper, we assume that each robot can communicate with other robots perfectly, without any communication faults in the robot specified sensor range and communication range which are considered as 3 and 10 cell units respectively [10,27].
- iii. Number of robots: is an important factor that affects the exploration time of the environment [9].
- iv. The environment's size [11,23] affects the exploration process, a large environment requires a long exploration time. In this paper we have used three different structural environments but with the same size.

Assumptions:

- i. Each robot initially has no information about other robots and the environment excepts the relative distances with other robots.
- ii. All robots have the same geometrical sizes equal to the size of a grid cell.
- iii. Each robot is able to communicate with the environment with no delay.
- iv. All robots can move upward, downward, leftward, and rightward.
- v.

4. MULTI-ROBOT EXPLORATION ALGORITHMS

Many exploration algorithms exist. Four methods are studied within the presented exploration framework, and the following paragraph gives an overview of these algorithms.

4.1 THE FRONTIER-BASED EXPLORATION ALGORITHM

The basic idea of this algorithm is to gather all the information about the environment by navigating to the boundary that separates the known and the unknown areas. When a robot navigates to a frontier cell, it uses the newly obtained information to construct the mapped area. The robot navigates to the next frontier cell in the assigned path to complete the map building process by adding the obtained information to the map, and increases its knowledge of the environment. If no frontier is found, it will return back to the common station and if a collision is detected, the robot replans the path. It returns left or right to avoid sticking with obstacles [19,22,27]. A summary of this procedure is presented in Algorithm 1.

Algorithm 1. *The frontier-based exploration*

```

While ( Frontiers or Unexplored areas found)
Do
    Measure the surrounding area using current sensor information
    Update the map with the new information
    Determine new goal candidates
    if ( No frontier is found or !The goal is reached)
        Return to the Common Station
    Assign the goals to robots
        if ( No assigned frontier )
            Go back to the Common Station.
        if (undetected overlapping with other robot)
            Replan the path.
    Assign paths for the robots.
    Choose the best frontier and move to it.

```

4.2 THE ROLE-BASED EXPLORATION ALGORITHM

The role-based exploration algorithm deals with the missing of communication capabilities in MRS exploration process for static environments [17,18]. Robots are joined to a network and share the local information with the team [11,13]. Initially, the robot's team is formed as a hierarchical tree of three layers. The robot(s) in the first layer represents the base station of the network.

Robots exist on the second layer work as relays that share information between the base-station robot and the explorer robots in the third layer. The explorer robots explore the environment and send the information to the relay robot at a scheduled time. A summary of the procedure is presented in Algorithm 2.

Algorithm 2. *The role-based exploration*

```

While ( Frontiers or Unexplored areas found)
Repeat
for each explorer robot do
    Initialize explorer
    Explore environment for a time
    if a rendezvous point is detected
        or a relay robot is in range then Share data to relay robot
    end
end
for each relay robot do Initialize relay
    if a rendezvous point is detected
        or an explorer robot is in range then share data from explorer robot
    end
    if a rendezvous point is reached
        or a base station robot is in range then
            share data with the base station robot
    end
end
end

```

4.3 THE LEADER-FOLLOWER EXPLORATION ALGORITHM

This algorithm focuses only on the function of the robot team rather than the environment structure. According to the distance between robots and the corridors, the function of each robot can be changed as well as the detection results. If a robot recognizes the corridor, it acts as a leader, and the remaining robots will be set as followers [10,5,6].

The following robots are looking at two factors, the first one is the Cost ($Cost_j^i = C_{path}(i,j) + C_{orientation}(i,j)$), which is the sum of distance from robot i to reach the frontier ($C_{path}(i,j) = V_j^i$) and the orientation cost ($C_{orientation}(i,j) = O_j^i$) of the robot when it rotates to reach the targeted frontier. The second one is the Frontier utility U_j to compute the proper frontier to the robots, and the difference between the frontier utility and the cost give the reward function of the followers from robot r_1 robot to the frontier f , the reward is shown below.

$$Reward_{\psi(t)}^{R_i} = Utility - Cost_j^i = U_j - (C_{path}(i,j) + C_{orientation}(i,j)) \quad (2)$$

$$= U_j - (\beta V_j^i + \gamma * O_j^i) \quad (3)$$

The markers or labels of the frontier are also taken into consideration to evaluate the suitable corridors assignment to the robots if $L=1$ the robot will assign to the targeted frontier and it sets as a leader if $L=0$ no frontier assigned to the robot. In this case, the optimal task assignment decision can be given as follows:

$$\pi^{opt} = \arg \max_j \sum_j R_f^{r_i} \quad (4)$$

where π^{opt} , is the best solution for task assignment. The details of this procedure are shown in algorithm 3.

Algorithm 3. *The leader-follower exploration*

Input: A cell-based map and the sensor data of robots.
Output: Build the map of an unknown environment using the input capability.
 Evaluate the labels L of frontier J_L .
 Calculate the cost V_j^i for all robots i to reach frontier J_L .
If the frontier J_L is labeled as corridor ($L=1$) without a target robot
 Determine a robot i for a frontier J_L which satisfy $i^* = \operatorname{argmin}_i V_j^i$
 Share the modified robot role with other robots
End if
While there is no robot assigned to a target frontier j which label $L=0$
 Determine a robot i for a frontier J_L according to
 the optimal decision solution
 $(i^*, j^*) = \operatorname{argmax}_{i,j} (V_j^L - b.V_j^i - g.O_j^i)$
 Reduce the remaining utility frontiers' as the
 laser's range of robot i can reach.
End while

4.3 THE HUNGARIAN ALGORITHM

This algorithm is the same as frontier-based exploration, except it use the Hungarian algorithm for the task of the assignment process. For the sake of simplicity, we have called this algorithm Hungarian exploration algorithm. The algorithm attempts to find the best goal state in terms of the given cost function and then finds the optimal route to ensure that the total length of the robot path that moves to its future position is minimized [25,26].

Algorithm 4. *The Hungarian exploration*

Given a cost matrix C of size $n \times n$ to find an optimal assignment.
Step 1. The smallest entry in each row is subtracted from all the entries of this row.
Step 2. The smallest entry in each column is subtracted from all the entries of this column.
Step 3. Covered all zero entry in the cost matrix by drawing appropriate lines of rows and columns.
Step 4. Test for optimality:
 If zero line = n **then**
 An optimal assignment of zero's is possible Exist.
 else if zero line < n **then** Jump to Step 5.
Step 5. Calculate the smallest entry doesn't cover by any lines.
 Subtract it from each uncovered row and add it to all covered column.
Return to Step 3.

Hungarian is an integrative optimization algorithm that solves the problem of mapping in a time limit to enhance the robot-task assignment. The task assignment can be built as a matrix C of size $n \times n$ matrix, where the element C_{ij} represents the path length from the i^{th} robot to the goal j^{th} . For any cost matrix C , the Hungarian finds the

optimal task assignment with a minimal cost and then finds the optimal route to ensure that the total length of the robot path that moves to its future position is minimized. At first, it assumes that the number of targets is the same as the number of robots. In case they are different, virtual goals or virtual robots can be added and assigned to a fixed cost and they are skipped during the exploration process [27]. After deciding the future locations, the future network must be connected. This future location and a global map are broadcasted to other robot teams by the leader. A summary of the procedure is shown in algorithm 4.

5. THE SIMULATION RESULTS

In order to compare the above-listed MRS exploration algorithms, MRESim is used as an exploration framework [1,11,27]. All the simulation experiments are tested on the same machine using the core-i5 processor at 3.8 GHz, 8 GB RAM running under windows 7. The total number of runs are thus (4 exploration algorithms) * (3 maps) * (5 team size configurations) * (average of 12 runs for each experiment) = 720 runs.

The simulator assumes perfect localization and noise-free sensor data [9,10,15]. A set of experiments are applied on three different maps with various structures and fixed sizes as described in Fig. 3. Each environment will be modeled as an occupancy grid of 100×100 cells. All the algorithms are tested to cover the whole environment by a team of identical MRS. In order to get near accurate evaluation results; these experiments will be implemented using a verity size of robot team, two, four, and six robots. Fig.3 shows three maps with the same dimensions and different structures, these maps can be titled as simple, moderate and complex maps. The simple map in Fig.3-a describes the case of a big room with a large free space, no walls in the middle or in other places except the sides and four fixed square obstacles with equal dimensions and black color. The moderate map in Fig. 3-b is more complex as it contains large corridors and divided into a group of small unequal rooms. The complex map in Fig. 3-c. has a large number of small corridors divided into many unequal and small rooms, it represents a real building with many separate rooms.

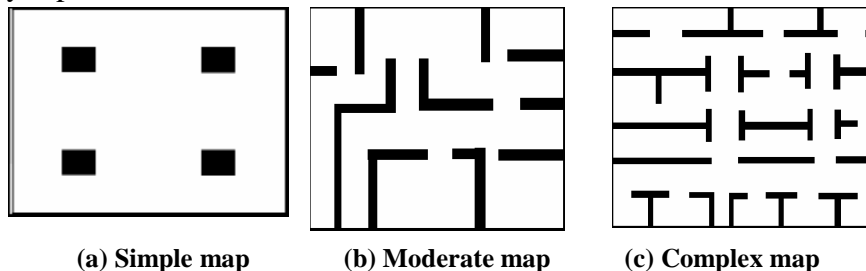


Figure 3 Maps used in exploration process

The simulation is run over a sequence of data such as (map size, map characteristics, size of the robot team and strategies used) which represents the main factors of the assessment process. It is repeated several times to obtain important statistical information about the performance of the task. The number of simulation runs is equal for the three maps to proceed this number of simulation runs, and the statistical assessment that compares the four algorithms are shown in Figs. 5–8 [10,11].

In order to evaluate and compare all individual exploration strategies, it is beneficial to have a common set of measurable performance values that indicate the relative achievement of each algorithm used. The proposition metrics are as follows:

- Maximize the total explored area in the map: as shown in Fig. 7 the area that the robot senses using the measuring device sensor is collected from all the robots team.

- Minimize the mean exploration time: by calculating the robots trajectories with all algorithms as shown in Fig. 8.
- Reduction of total hop counts: the link bridges that are established to connect a pair of robots may have multiple hops, each of them may raise some delay in the communication network between the robot team. This metric is shown in Fig. 10, for simplicity, the test is run only in case of 10-robot-team with complex environment.

All the robots are supposed to move from the top left corner to the bottom right corner. The four algorithms are run on all three maps. Performance of each of the algorithms is measured in terms of (a) number of movement steps, (b) path length or mean exploration time, (c) number of hop counts. One movement step is the robot's moving from one cell to any of its adjacent neighbouring cells, and one step's length may be 1 or 1.414 [11]. An example of how to calculate these parameters is shown in Fig. 4 [10,11].

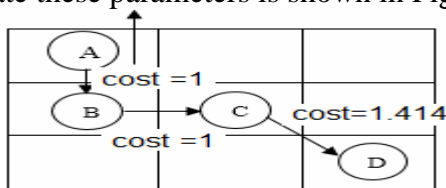
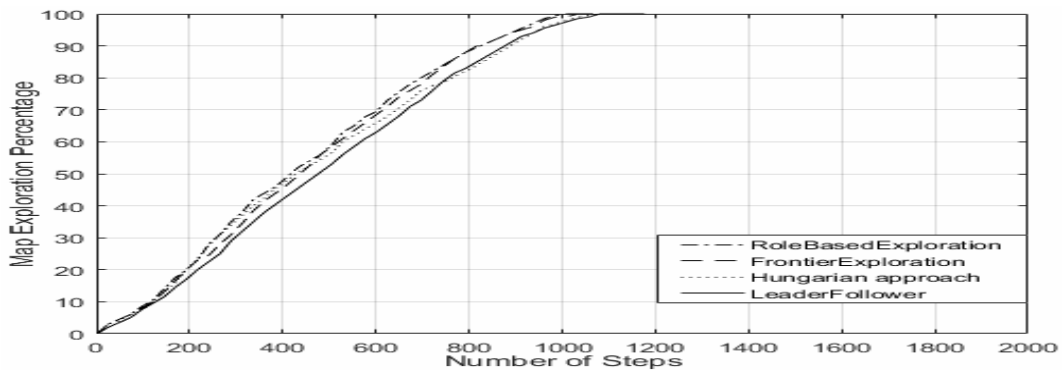
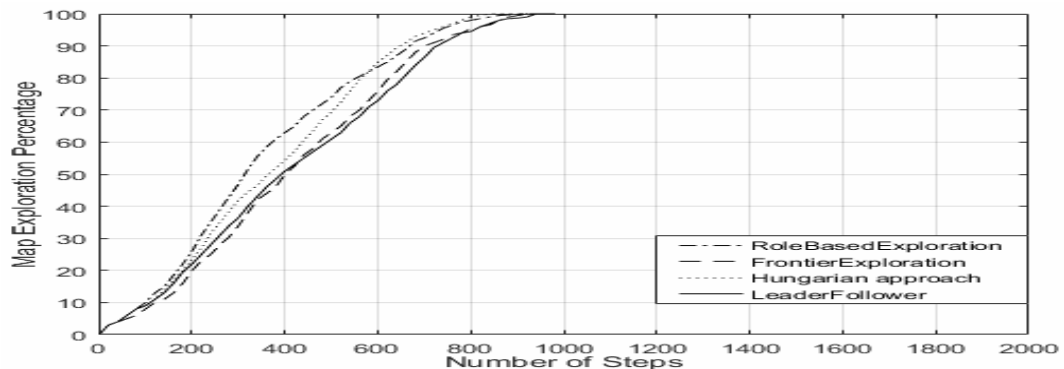


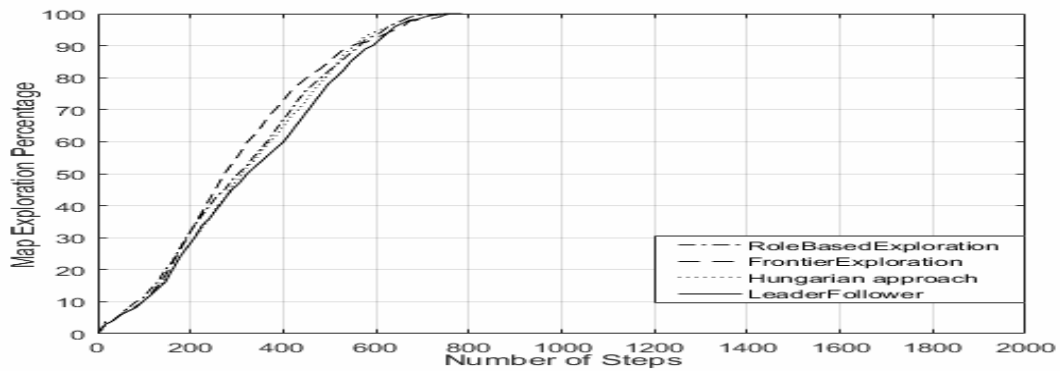
Figure 4 The performance parameter calculation example, the small circle represents robot's current location



(a) using 2 robots team size



(b) using 4 robots team size



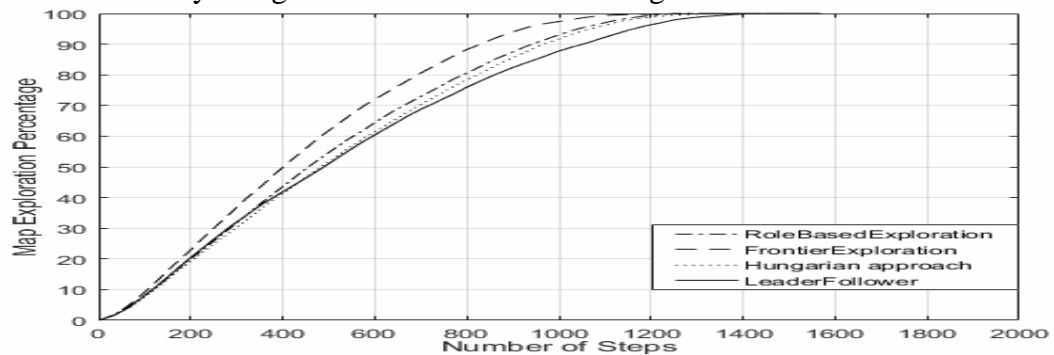
(c) using 6 robots team size

Figure 5 The simple map exploration results for (a),(b) and (c)

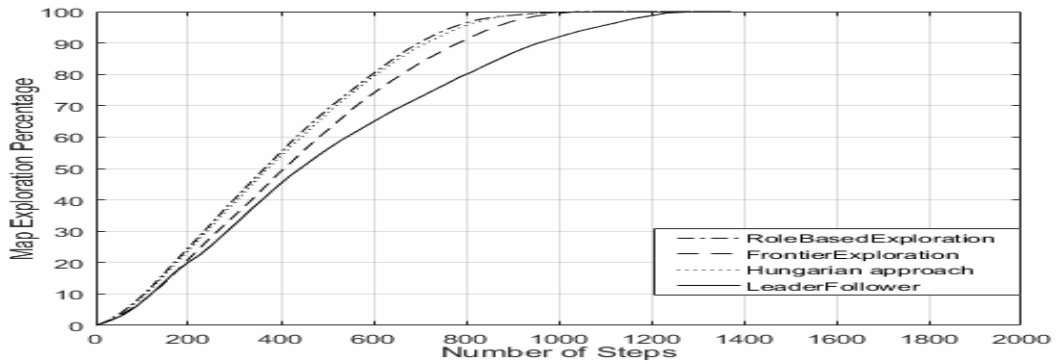
The graphs shown from Figs. 5-8 compare the performance of the four exploration algorithms based on (i) the average number of movement steps of each robot to accomplish the task, and (ii) the meantime (average path length) of exploration required by each robot to fully explore the environment's maps.

For the simple map environment, the simulation results for all algorithms using different robots sets of sizes: 2, 4 and 6 robots are shown in Fig. 5. It indicates that the four algorithms give approximately the same results for all team sizes; except for the leader-follower algorithm which has a slightly different behaviour. Also, the number of steps needed by each team to completely explore the entire map decreases while the number of the robot in each team increases as shown in Fig. 6. For example, the maximum number of steps required by a team of 2 robots is 1008 steps and 736 steps for a team of 6 robots when using the role-based algorithm.

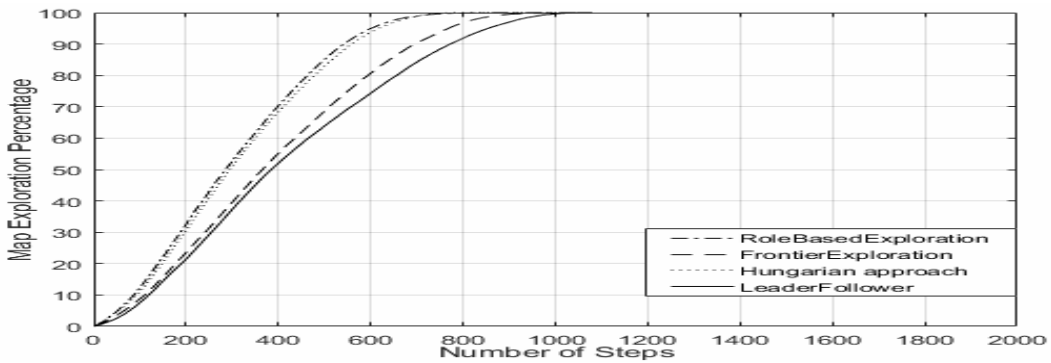
The same experiment is tested for the moderate map described in Fig. 3-b and the results are plotted as shown in Fig. 6. The moderate map is somewhat more complex than the simple map. Therefore, the results of this experiment indicate that there are some differences among the four algorithms. Some of these differences are reflected in the frontier-based and leader-follower algorithms; which are the worst two algorithms especially in case of four and six team sizes. The role-based algorithm yields better results followed by Hungarian method as shown in Fig. 6.



(a) using 2 robots team size



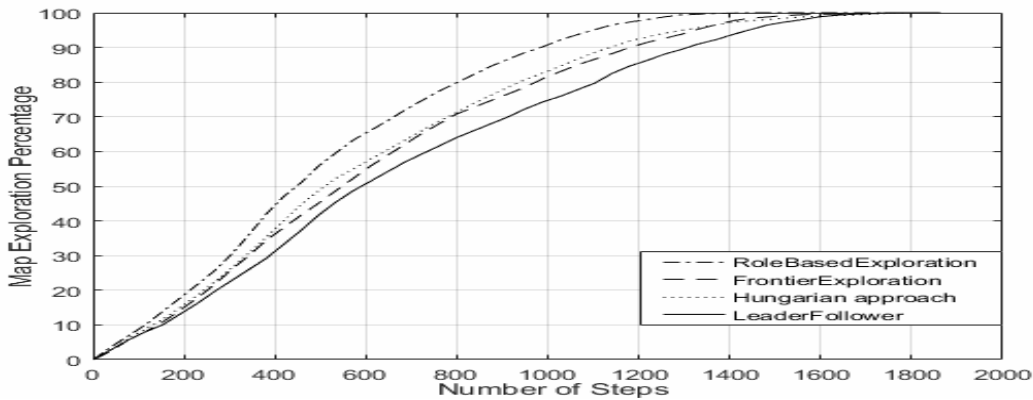
(b) using 4 robots team size



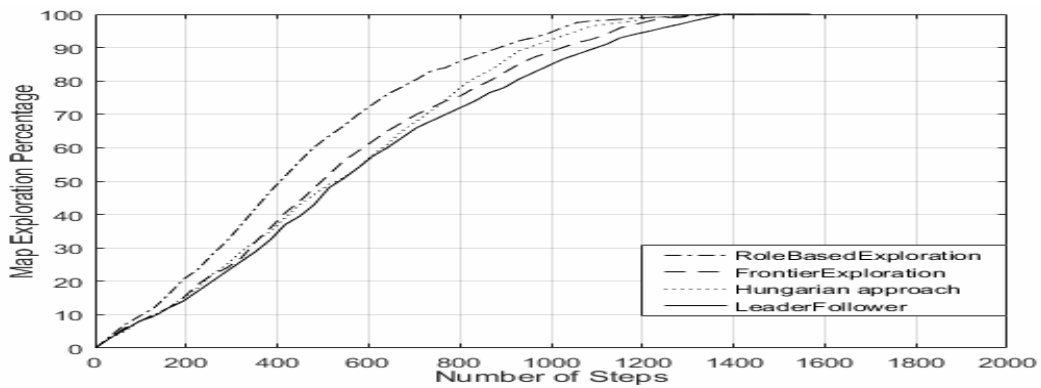
(c) using 6 robots team size

Figure 6 The moderate map exploration results for (a),(b) and (c)

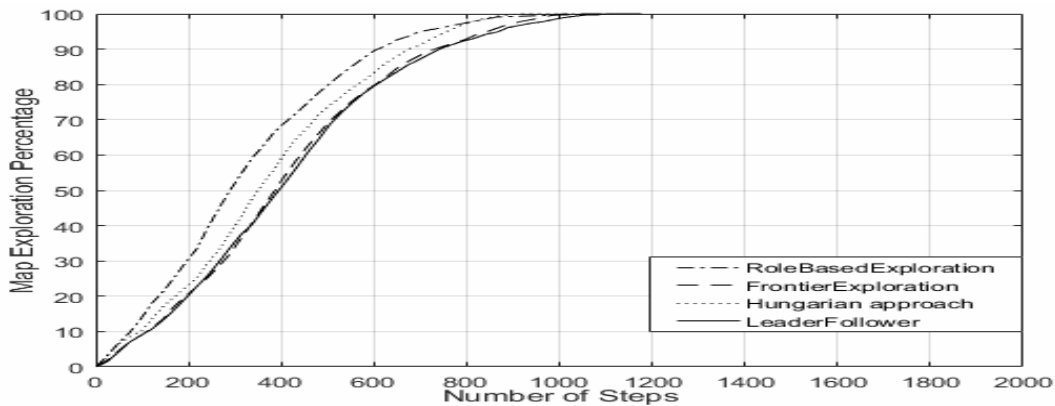
The only result that the frontier-based superiority of the remaining algorithms is in case of using a team of 2 robots. As shown in Fig. 6 the number of steps in case of moderate map exceeded the number of steps required to completely explore the entire simple map, this is normal because this map is more complex than the simple map, also the number of steps decreases as the number of the robots. For example, the maximum number of steps required by a team of 2 robots is 1404 steps, while in a team of 6 robots it is 825 steps when using the role-based algorithm.



(a) using 2 robots team size



(b) using 4 robots team size



(c) using 6 robots team size

Figure 7 The complex map exploration results for (a),(b) and (c)

Finally, the same experiment is tested for the more complex environment as described in Fig. 3-c and the simulation results are plotted in Fig. 7. We have found that the performance of the role-based and Hungarian algorithms are better, but the performance of the leader-follower and frontier-based is less than them the reason of this is that the more complex the map is, the less their performance and the number of steps decreases as the number of the robot increases. For example, the maximum number of steps required by a team of 2 robots is 1444 steps, while in a team of 6 robots it is 912 steps when using the role-based algorithm.

From all the above results we can conclude that as the number of robots increases the number of steps required to explore the entire maps decreases. Finally, it is possible to say that the best results are achieved by the role-based algorithm followed by Hungarian which even outperforms the role-based algorithm in some cases. The worst two algorithms are frontier-based followed by leader-follower Algorithm.

The number of steps that are required for each size of a robot team to accomplish MRS exploration task using the four strategies in the three different maps of an environment is shown in Table 1. The table shows that each algorithm contains a row that splits into three rows of numerical data information; these data are arranged from a top that corresponds to a simple map, moderate map and complex maps.

Table 1 Number of steps for task completion

algorithm	Number of a robot team		
	2	4	6
Frontier-based	980	870	702
	1208	1070	842
	1654	1282	1010
Role-based	920	794	666
	1324	925	725
	1390	1200	818
Leader-follower	1010	880	682
	1504	1190	883
	1730	1378	1034
Hungarian	980	770	682
	1375	984	712
	1635	1250	914

Table 2 Meantime of the exploration process

algorithm	Number of a robot team		
	2	4	6
Frontier-based	1032	940	784
	1248	1092	1056
	1748	1376	1104
Role-based	1008	900	736
	1404	1034	825
	1444	1320	912
Leader-follower	1080	920	752
	1596	1344	1056
	1824	1472	1128
Hungarian	1080	840	752
	1455	1064	792
	1786	1344	1008

This paper also discusses the relationship between the team size and the meantime of exploration by comparing the robots trajectories of all algorithms as shown in Fig. 8. For the simple map, the simulation results show that the role-based algorithm has less exploration meantime compared to the other algorithms for all team sizes followed by the Hungarian algorithm, and the frontier-based followed by the leader-follower needs more time to completely explore the entire map.

For the moderate map, the role-based and Hungarian algorithms needed less time than the frontier-based and leader-follower algorithms for all team sizes of robot except in case of 2 robots frontier-based algorithm appear better. Similarly, for a complex map, the role-based and Hungarian algorithms needed less time than the frontier-based and leader-follower algorithms for all team sizes of robots.

In Fig. 8 the time of exploration decreases when the number of robots in the team increases. Also, the exploration time for the same team size in the three maps increases and this is the result in the simple map. There are only four obstacles and there are no limits in detection range, but gradually obstacles increase and the number of walls in moderate and complex maps which limits detection ranges for the robot and then limits the robot navigation in the environment.

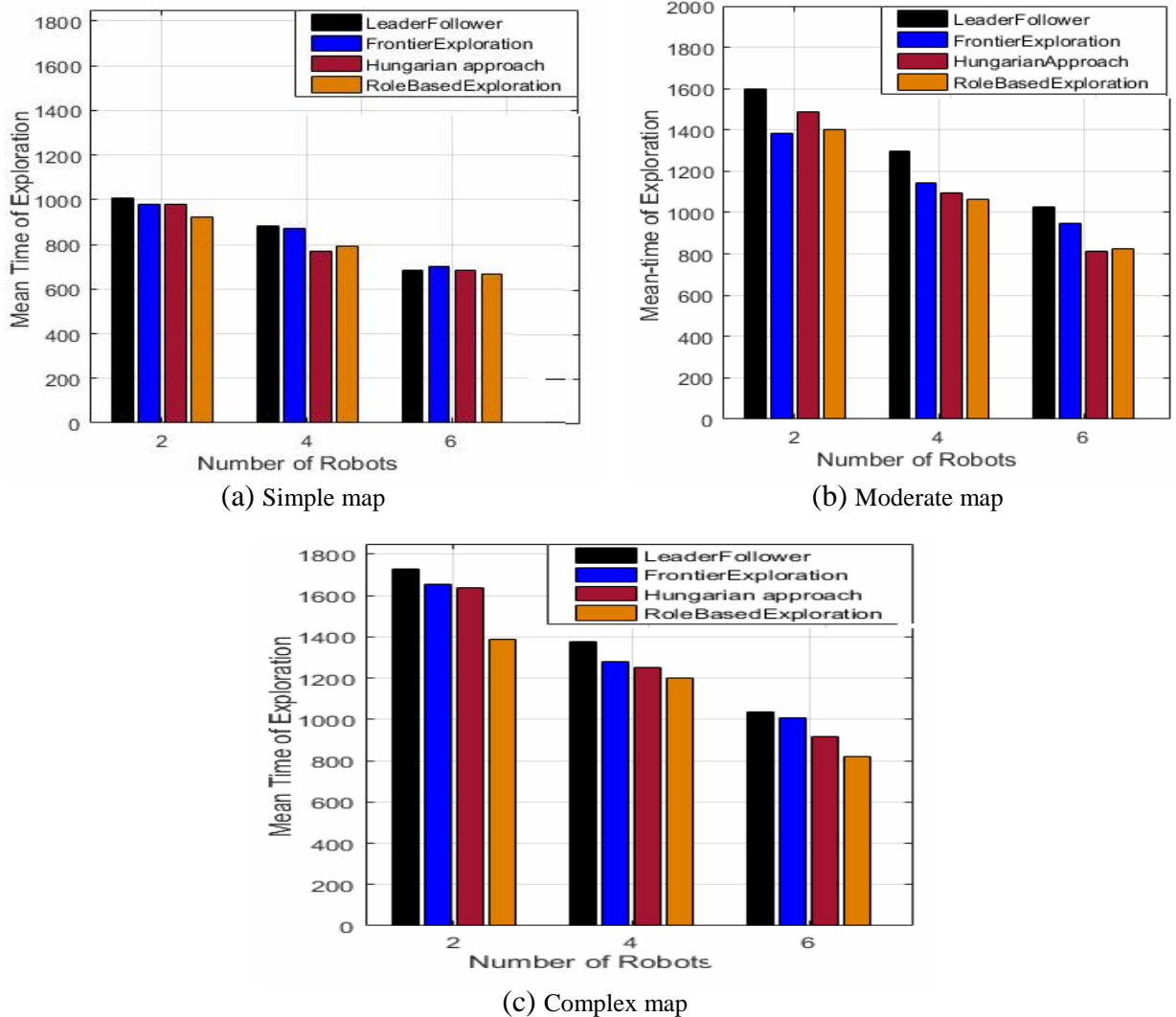


Figure 8 Exploration time mean vs robot team scale in (a), (b) and (c)

The meantime of the exploration process required by each size of a robot team to complete MRS exploration and coordination task using the four strategies in the three different environment's maps is shown in Table 2. The table shows that each algorithm contains a row that splits into three rows of numerical data information, these data are arranged from a top that corresponds to a simple map, a medium that corresponds to a moderate map and down that is corresponds to a complex map [10,11].

In order to reduce the communications overlapping between the robot team and some of them, it is necessary to ensure that the robots in the area that are already explored are compatible with each other at all times. Therefore, a communication mechanism is created so that robots can share their local information with each other at each movement step. The four algorithms mentioned in this paper are communicated using a centralized manner, Fig. 9 shows an example of a multi-hop network that demonstrates the centralized communication systems. If we consider the hop count values for all paths among all robot pairs the result is shown in Table 3, in another way when we take any robot as a central node (e.g robot 3) the hop count values for all path established for robot 3 is shown in Table 3.

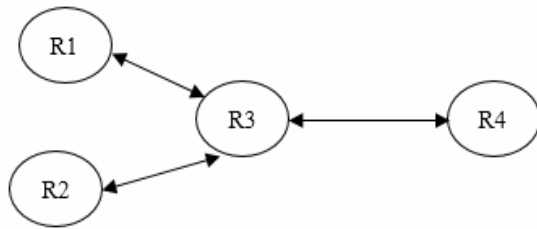


Table 3 The hop count values for all path among all robot pairs to leader R3

Robot pair	Hop counts
R1-R3	1,1
R2-R3	1,1
R4-R3	1,1
Total	6

Figure 9 Example of multi-hop network

Fig. 10 shows that the total number of hop counts needed at every step in all algorithms based exploration method with 10 robots to make the test simple. This paper based on [21,27,28], therefore, the min and max hop counts for the complex map using role-based, Hungarian, frontier-based and leader-follower algorithms remain within the band of [(20-40), (27-48), (38-56), (42-58)] respectively, except in some cases where the Hungarian and leader-follower algorithms exceed this band ,especially when using a complex environmental map.



Figure10 Number of steps versus total hop counts in the case of 10 robot for complex map

6. COCLUSIONS AND FUTURE WORKS

In this paper, a four common MRS exploration algorithms are presented, and its performances are analyzed and compared for different team sizes and different environments with respect to three metrics total time needed to explore the three environments, number of steps to accomplish the exploration task. Role-based exploration algorithm yields better results than the other used algorithms followed by Hungarian. In the future, the role-based exploration algorithm can be used as the main exploration algorithm to design a framework for task coordination in MRS.

REFERENCES

- [1] A. Pal, R. Tiwari and A. Shukla, “Coordinated Multi-Robot Exploration under Connectivity Constraints”, *Journal of Information Science and Engineering*, 29(4), **2013**, pp. 711-727.
- [2] K. Cesare et al., “Multi-UAV Exploration with Limited Communication and Battery”, Proc. IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26-30 May 2015.
- [3] T. Gunn, J. Anderson, “Effective Task Allocation for Evolving Multi-Robot Teams in Dangerous Environments”, Proc. IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), Atlanta, GA, USA 114(4), 17-20 Nov 2013, pp. 82-97.
- [4] S. C. Nagavarapu et al., “Multi-Robot Graph Exploration and Map Building with Collision Avoidance: A Decentralized Approach”, Springer, Science+Business Media Dordrecht, 83(3-4), **2015**, pp. 503-523.
- [5] M. Andries, F. Charpillet, “Multi-robot exploration of unknown environments with identification of exploration completion and post-exploration rendezvous using ant algorithms”, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3-7 Nov 2013.
- [6] A. Farinellia et al., “Advanced Approaches for Multi-Robot Coordination in Logistic Scenarios”, *Elsevier, Robotics and Autonomous Systems*, 90(3), **2017**, pp. 34-44.

- [7] N. Palmieri et al., “Coordination Techniques of Mobile Robots with Energy Constraints”, Proc. IEEE International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Montreal, QC, Canada, 24-27 July 2016.
- [8] D. Fox et al., “Distributed Multi-robot Exploration and Mapping”, *IEEE Transactions on Robotics and Automation*, 94(2), **2006**, pp. 1325-1339.
- [9] W. Burgard et al., “Coordinated Multi-Robot Exploration”, *IEEE Transactions on Robotics*, 21(3), **2005**, No. 3, pp. 376-386.
- [10] K. Mohamed, A. Elshenawy, H. M. Harb, “Exploration Strategies of Coordinated Multi-Robot Systems: A Comparative Study”, *International Journal of Robotics and Automation*, 7(1), **2018**, pp. 48-58.
- [11] K. Mohamed, A. Elshenawy, H. M. Harb, “A Hybrid Decentralized Coordinated Approach for Multi-robot Exploration Task”, *The Computer Journal*, 3(4), **2018**, pp. 473-487
- [12] F. S. Melo, M. Veloso, “Decentralized MDPs with sparse interactions”, *Elsevier, B.V, Artificial Intelligence*, 175(11), **2011**, pp. 1757–1789.
- [13] A. Pal et al., “Multi-Robot Exploration in Wireless Environments”, *Springer, Science+Business Media in Autonomous Robots*, 4(4), **2012**, pp. 526–542.
- [14] T. Andre, C. Bettstetter, “Collaboration in Multi-Robot Exploration: To Meet or not to Meet?”, *Springer, Journal of Intelligent & Robotic Systems*, 82(2), **2016**, pp. 325–337.
- [15] S. Saeedi et al., “Multiple-Robot Simultaneous Localization and Mapping: A Review”, *Journal of Field Robotics*, 33(1), **2016**, pp. 3-46.
- [16] X. Dail et al., “Cooperative exploration based on supervisory control of multi-robot systems”, *Springer, Media in Autonomous Robots*, 45(1), **2016**, pp. 18–29.
- [17] A. D. Haumann, et al., “DisCoverage: A new Paradigm for Multi-Robot Exploration”, Proc. IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14-18 May 2010.
- [18] S. Omidshafiei et al., “Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions”, *IEEE International Journal of Robotics Research*, 36(2), **2017**, pp. 231–258.
- [19] A. Marjovi et al., “Multi-Robot Exploration and Fire Searching”, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10-15 Oct 2009.
- [20] M. Turpin et al., “CAPT: Concurrent assignment and planning of trajectories for multiple robots”, *The International Journal of Robotics Research*, 33(1), **2014**, pp. 98–112.
- [21] Z. Yan et al., “A Survey and Analysis of Multi-Robot Coordination”, *International Journal of Advanced Robotic Systems*, **2013**.
- [22] J. Faigl, M. Kulich, “On Benchmarking of Frontier-Based Multi-Robot Exploration Strategies”, Proc. IEEE European Conference on Mobile Robots (ECMR), Lincoln, UK, 2-4 Sept 2015.
- [23] J. Banfi et al., “Strategies for Coordinated Multirobot Exploration with Recurrent Connectivity Constraints”, *Springer, Media in Autonomous Robots, LLC*, 42(4), **2017**, pp. 875–894.
- [24] C. Amato et al., “Policy search for multi-robot coordination under uncertainty”, *International Journal of Robotics Research*, 35(14), **2016**, pp. 1760–1778.
- [25] F. Caccavale et al., “Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators”, *IEEE/ASME Transactions on Mechatronics*, 13(5), **2008**, pp. 152-165.
- [26] Y. Kantaros, M. M. Zavlanos, “Global Planning for Multi-Robot Communication Networks in Complex Environments”, *IEEE Transactions on Robotics*, 32(4), **2016**, pp. 1045-1061.
- [27] G. Lozenguez et al., “Punctual versus continuous auction coordination for multi-robot and multi-task topological navigation”, *Springer, Science+Business Media in Autonomous Robots*, 40(4), **2015**, pp. 599-613.