Exploration of Unknown Map for Safety Purposes using Wheeled Mobile Robots

Sara Ashry Mohammed[1,3] and Walid Gomaa[1,2]
[1]*Cyber-Physical Systems Lab (CPS), Computer Science and Engineering Department (CSE), Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt*
[2]*Faculty of Engineering, Alexandria University, Alexandria, Egypt*
[3]*Computers and Systems Department, Electronic Research Institute (ERI), Giza, Egypt*

Keywords: Mobile Robots, Explore Unknown Map using Multi-robot, Rectilinear Obstacles, Gas and Fire Detection, V-REP, Khepera III, Albers Algorithm, Zigzag Algorithm, Heuristic Backtracking SRT Algorithm.

Abstract: Exploring unknown 2-D grid map using multi-robots has a great significance in a vast domain of applications. One possible application is to search for a gas leakage or a fire source which we address in this paper. We propose an algorithm called Zigzag Ray for multi-robot exploration. The aim is to reduce the required time to discover the environment as much as possible to suit the critical applications such as rescue operations. Experiments are done using two, three, and four Khepera robots for exploring a map. The exploration time without the boundary scan offset is ranged from 28.4% to 17.2% of the time taken by Albers algorithm and from 41.2% to 30.7% of the time taken by the Zigzag algorithm for a single robot. Also, the time of four robots by using a Zigzag algorithm for multi-robots is about 46% of Albers time of four robots. A disparity in time existing between the algorithms shows the effectiveness of the new proposed algorithm. Additionally, the Zigzag algorithm of a single robot is compared with the heuristic SRT algorithm. Zigzag time takes about 54.5% to 77.4% from heuristic SRT time. The evaluation is done using the Exploration Index strategy.

1 INTRODUCTION

Historically robots have been limited to industry, where fixed manipulators were used for welding, painting, assembly, product inspection, and testing. Currently, robots are being used more for other purposes. For instance, mobile robots have become more present in our daily life; cleaning our flats, mowing the lawn or searching for gas seepage or fire in an unknown environment.

Robotic exploration algorithms grow because they are crucial in many applications in this field. Recently, looking for olfactory targets using mobile robots has received a considerable interest because of its importance in the detection of chemical seepage as well as rescue and searching operations. Many studies have been done in this field in (Marjovi et al., 2011), (Soldan et al., 2014) and (Soares et al., 2016). Hiroshi used chemical sensors as Noses for mobile robots (Ishida et al., 2016). Multi- robots for odour source localisation in an indoor map is presented by Wang (Wang et al., 2016). A gas leak source detection with mobile robots is introduced by (Martinez et al. 2014).

The motivation of this work is to give a prototype of a robotic system that helps saving people from fires and suffocation resulting from gas seepage. Using a multi-robot in hard operations instead of firefighters ensures the safety of their lives. Similarly, it can also be applied in a smart home. It can be essential for older whose abilities to move and to sense gas seep are weak. This is not limited to only the elderly, but also children who are left alone at home sometimes. This application will have broader prospects if it is applied as a protection factor everywhere. Furthermore, the exploration time and speed are the most significant factors in critical applications.

This paper contributes a novel of exploration algorithm using a multi-robot system. Robots cooperate through a centralised PC station to explore a map. The experiments are done on two, three, and four robots. Also, the results are compared to both Albers (Albers, 2002) and Zigzag algorithm of single-robot (Ashry and Gomaa, 2016) to show the effectiveness of the multi-robot exploration algorithm, especially, in critical applications.

Mohammed, S. and Gomaa, W.
Exploration of Unknown Map for Safety Purposes using Wheeled Mobile Robots.
DOI: 10.5220/0006430903590367
In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017) - Volume 2*, pages 359-367
ISBN: Not Available

Additionally, we compare Zigzag single robot with a Heuristic SRT (Hussieny et al., 2015) algorithm.

Our paper organisation is as following: an introduction is showed in section 1, after that the related work is mentioned in Section 2, Section 3 is discussing the background. Section 4 describes the zigzag algorithm of multi-robots. Section 5 shows the experiments in both a real world and simulation. Section 6 concludes this paper.

2 RELATED WORK

The majority of the multi-robot exploration algorithms have relied on using the concept of frontier cells. The frontier-based exploration was initially introduced by (Yamauchi, 1997) where each grid cell in a 2-D map has a numeric value that shows the existence of objects in the map.

Additionally, (Yamauchi, 1998) contended that to find out the map; each robot moves towards the nearest free frontier cell, and at least one of its neighbouring cells is unexplored. The challenge is how to choose the best frontier cell if more than one robot is included in the exploration, it is important to avoid the situation where two robots move to the cell itself. Nevertheless, He considered that it is probable that more than one robot move to explore the same frontier and then more time is required to accomplish the task. A more advanced technique where the robots start at a known initial points is suggested by Burgard (Burgard et al., 2005). The aim is to minimise the whole time by choosing proper frontier cells for each robot so that they explore different parts of the map and the overlap between them is reduced.

(Sheng et al., 2006) considered that a limitation communication range between the robots is a great problem. In all of the mentioned research, the robot senses the neighbouring cells using laser range sensors. But, what if laser sensors aren't available or expensive. So, the proposed algorithm is suitable.

3 BACKGROUND

3.1 Albers Algorithm

Albers supposed that the robot starts at a corner point and moves along the external border of a rectilinear map in a clockwise direction until it backs to its origin again. So, the robot could know the map dimension and determine the lower segment. Then, it moves up in a northern ray until hitting the boundary.

The robot then goes south on the same column and takes one step east and radiate another northern ray in a recursive operation till hitting the exterior boundary or an obstacle. If the up ray hits an obstacle, then the robot follows obstacle exploring process as described in (Ashry and Gomaa, 2016). Albers algorithm is shown in figure 1.

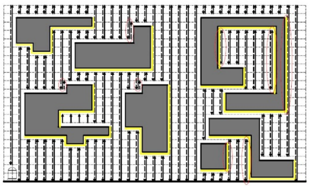


Figure 1: Shows Albers algorithm (Ashry, Gomaa, 2016).

3.2 Heuristic SRT Algorithm

Heuristic SRT algorithm is one of the randomised motion exploring approaches. Robots are directed to gain information through random steps. Those steps should validate certain conditions to maximise the acquired information and cover the entire map. While random techniques have been used intensively for the exploration, they are not efficient for the time critical applications since the robot may visit the place itself more than once during the backtrack operation like (SRT) Sensor Random Tree (Oriolo et al., 2004). (Hussieny et al., 2015) suggested a Heuristic SRT to solve the backtracking problem to find a new frontier cell. It reduced the exploration time to 30 % and 28 % respectively from basic SRT as shown in figure 2.

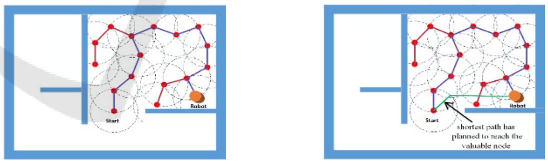


Figure 2: Left figure shows the basic SRT and right figure shows heuristic backtracking algorithm, where a green line (shortest path A*) is planned to the most informative node and then the exploration process starts again. The robot equipped with 360º laser finder (Hussieny et al., 2015).

3.3 Zigzag Algorithm of Single Robot

The problem of Albers algorithm is that the robot moves on the same column twice because it goes up and down on the same line to hit the lower segment. Hence, this algorithm doubles the time of exploration. The Zigzag algorithm proposed an enhancement of this drawback. The robot radiates a ray to the north, and then move one step in the east and emit a ray in the southern direction of the next column (up/east/down/east) (Ashry and Gomaa, 2016). It does not reverse again on the same column like Albers algorithm. If it faces an obstacle, it will follow obstacle exploring process as in figure 3.

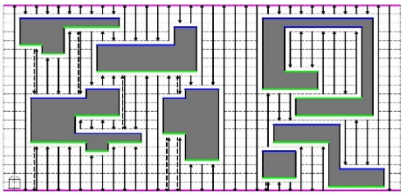


Figure 3: Shows Zigzag Ray lines, lower /upper segments.

4 ZIGZAG MULTI-ROBOTS

Zigzag of multi robots algorithm is proposed to enhance the performance of zigzag of a single robot. The experiments at section 5 show the effectiveness of this algorithm compared to Albers and zigzag (Ashry and Gomaa, 2016). The following algorithm steps explain the zigzag of multi-robots on two robots to illustrate the concept. Sure, the same algorithm can be applied on multi-robots (3, 4, and so on). The PC works as a centralised station, and the robots work through a client to server's network.

```
Algorithm: Zigzag Multi Robot
switch System State do
    case Initialize Client_Servers do
        Initialize Client();
        for i = 1; i ≤ N; i++ do
            Initialize_Robot_Server(Robot R[i]);
            Check Success Socket Connection(Robot R[i]);
            Adjust Direction(Robot R[i]);
        end
    case Exterior Boundary Scan do
        InitializeRobot_1 Position(x,y) = (0,0); i = 1;
        repeat
            Start Move(Robot R[i]);
            while true do
                Send Position_Direction To Client(Robot R[i]);
                if Robot_i Face Robot_{i+1} then
                    i = i + 1; Break;
            end
        until Robot_{i+1}(x,y) = (0,0)
        Type The Whole Map Boundary Is Explored();
    case Split Map To N Areas do
        Client Split Map To N Areas();
        for i = 1; i ≤ N; i++ do
            Client Send Small Map to Robots(Map m[i], Robot R[i]);
        end
    case Exploring Map do
        for i = 1; i ≤ N; i++ do
            repeat
                Check Collision Avoidance(Robot R[i]);
                if There isn't any common obstacle then
                    Zigzag Ray Traversal(Robot R[i]);
                    Send Position_Direction To Client(Robot R[i]);
                    Client Update Map();
                else
                    Common Obstacle(Robot R[i]);
                    Zigzag Ray Traversal(Robot R[i]);
                    Send Position_Direction To Client(Robot R[i]);
                    Client Update Map();
            until All cells are explored
        end
    case Parking do
        for i = 1; i ≤ N; i++ do
            Go_Nearest Corner_Exterior_ Boundary (Robot R[i]);
            Check Collision Avoidance(Robot R[i]); Stop();
        end
        Send Robot Parked Message To Client();
end
```

Figure 4: Shows the zigzag algorithm for multi-robots.

Protocol

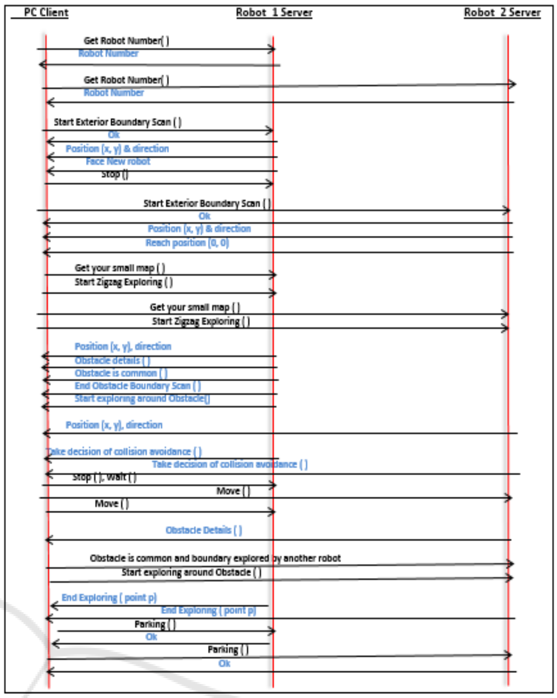


Figure 5: The protocol between the client and servers.

PC centralised station:

1. Initialize a connection between the PC as a client and each robot as a server.
2. The PC asks each robot in the initial state to adjust its direction till the left sensor is occupied and it will receive the position and direction of robots during the *boundary scan process*.
3. After the PC has the exterior boundary, it will apply the divide map process which divides the whole map into small areas equals to the robots number. Then, it sends small maps to each robot.
4. The PC receives information from each robot about visited cells and obstacles and update the map.
5. If the PC receives information from any robot that there is a common obstacle, the "*Common Obstacle process*" will apply.
6. During exploring, each robot applies *Collision Avoidance* technique at each node.

Robots as servers:

1. At the initial state, if we have two robots, the first robot will be put at any corner with any direction (north or east or west or south), and the second robot will be placed at any point on the exterior boundary of the map at any direction.
2. Each robot starts to adjust its direction till the left sensor becomes occupied.

3. The first robot starts to apply the *Boundary scan process*. It moves on the exterior boundary in a clockwise direction until it faces the second robot. Then, it will stop and send the position and orientation of the second robot to the station.
4. The second robot will continue exploring the exterior boundary till it finishes it. So, the map dimension is known. The *boundary scan process* occurs at the first execution of the algorithm in the map, but after that, it is executed without this step.
5. After the PC divided the map, each robot receives information of a small map that should be explored, and it starts exploring from the first column in its area.
6. If a small map does not contain any obstacles, then each robot will apply the same concept of a zigzag ray of the single robot (Ashry and Gomaa, 2016). But, if there is any common obstacle between two regions, the robot will use *common obstacle method*.
7. After each robot completes exploring its area, it will park at any corner on its map boundary.

Divide Map process:

This pattern for the robots number (N = 2 or 3 or 4). If N = 2, then split the large length (LL) by two as shown in figure 6. If N= 3, then divide LL by 3. If N = 4, then divide LL by 2 and cutting the large width by 2.There are general algorithms for dividing a map into small equal maps (Shermer, 1992).

(a)Divide map to 2 small areas.

(b) Divide map to 3 small areas

(c) Divide map to 4 small areas.

Figure 6: The pattern of dividing map to 2, 3, and 4 areas.

Common Obstacle:

This method is applied when there is a common obstacle between the areas. There are two cases:

One robot detects an obstacle and complete travelling around it before other robots.

The other robot when detect the obstacle will start explore it using zigzag process without the need to travel around it.

The robots detect the common obstacle at the same time.

Each robot will move around the obstacle till it reach to the start point of the other robot on the obstacle, then each one will explore it from its area.

Collision Avoidance Technique:

Each robot checks if any robot exists in its circle using ultrasonic sensors as shown in figure 7. The circle radius equalises to the edge length. If a robot detects any other robot in its circle, then the robots will stop and send to the station for taking the decision about which one should move and which one has to stop and wait until no other robot is detected on its circle. The PC chooses any robot randomly to move and ask the others to wait. Then, all robots will check ultrasonic again at each node.

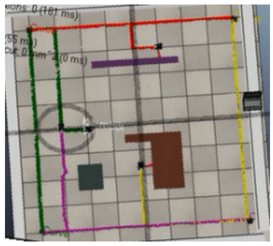


Figure 7: Shows the divided small maps virtually and the circle area indicates the collision avoidance technique.

5 EXPERIMENTS

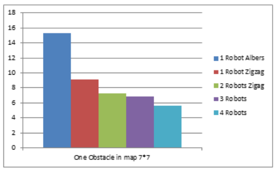
5.1 V-Rep Simulation

V-REP is used for developing algorithms and check verification (Rohmer et al., 2013).

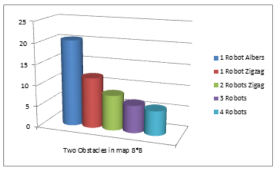
5.1.1 Comparing the Results of Zigzag Algorithms towards Albers for Single Robot

Experiments are done on V-REP to prove the efficiency of the new Zigzag multi-robot vs. Albers algorithm and the zigzag of a single robot. The environments consist of M obstacles, N robots.

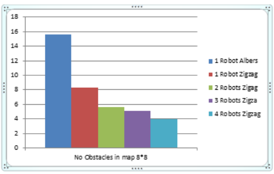
Figure 8: Left figure shows 4 robots on a map with one obstacle. Middle figure shows them on map with two obstacles. But, left shows them on a map with 3 obstacles.



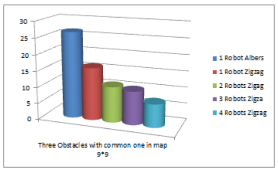
(a)Test1, 1obstacle in map7*7.



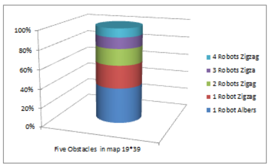
(b) Test2, 2obstacle in map8*8.



(c) Test3, No Obstacle in map8*8.



(d) Test4, 3obstacle in map 9*9.



(e) Test5, 5 obstacles in map 19*39.

Figure 9: Charts of v-rep experiments shows the efficiency of exploration time of the proposed algorithm.

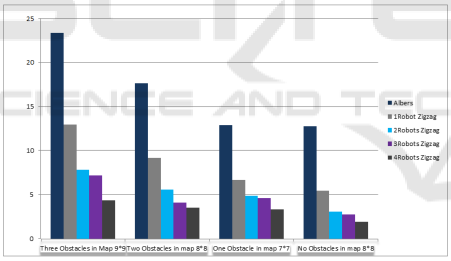


Figure 10: It shows the exploration time at V-REP without a boundary scan time, so, the speedup by 4 robots ranged from 71% to 85%.

5.1.2 Comparing the Results of Zigzag Multi-robots versus Albers Multi-robots

We applied the same concept of a Zigzag algorithm for multi robots, as explained in section 4, to Albers algorithm to make a fair comparison. So, Albers is tested on 2 robots and 4 robots. They are tested on the same maps at test 1, 2, 3 and 4. Figure 11 shows the result of this comparison.

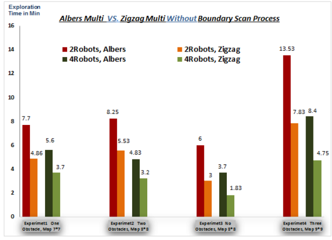


Figure 11: It shows the exploration time of Albers vs. Zigzag multi-robot at the V-REP without boundary offset when each robot knows its start position and orientation. The speed up percentage ranged from 33% to 50.5%.

5.1.3 Albers and Zigzag on a Rectilinear Map

The experiments are also done on a rectilinear map to show another environments type.

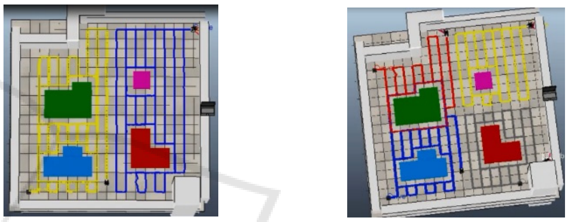


Figure 12: It shows experiments on a rectilinear map. Albers with 2 robots on the left figure while Zigzag with 4 robots on the right figure.

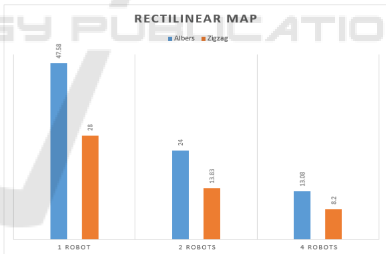


Figure 13: It shows the exploration time of Albers vs. Zigzag on a rectilinear map without boundary offset. The speed-up percentage for 1, 2, and 4 robots ranged from 37% to 42.5%. Additionally, using the Zigzag on 4 robots save 82.76% of the Albert time of single robot.

5.2 Controlled Real World Experiments

The Experiments are done in a controlled environment with 6*6 nodes and surrounded by a wooden and cartoon panels as walls. Figure 14 shows a snapshot from the experiments. Also, figure 15 shows the time of Albert, Zigzag on a single robot, and Zigzag for multi-robot at 2, 3, and 4 robots on a map with zero, one and two obstacles.

Figure 14: A snapshot from the controlled environment on the real world of map consist of 6*6 nodes.

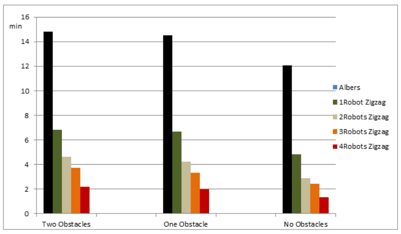


Figure 15: It shows the real world experiments without the boundary time offset. So, the speed-up percentage by using the Zigzag to Albers approach, for 4 robots ranged from 85% to 89%.

5.3 Searching for Gas or Fire using Multi-robots

We extended the application of searching for gas source using one master robot (Ashry, 2016). The extension includes searching for gas or a fire source using multi-robots for saving people from gas suffocations and fires. The master robots apply the proposed exploration algorithm of Zigzag multi-robots. Experiments are done on two master robots equipped with high sensitive gas and flame sensors as and two slave robots equipped with a gripper to capture the potential victim by taking the Dijkstra's algorithm as shown in figure 16. The candle is used as a fire source. The sensor module name is MQ-6. It is sensitive to butane, propane, and natural gas.

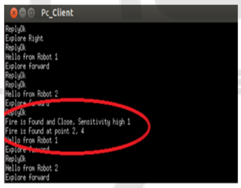
After the exploration process and detecting the target, the map and the target point (object found beside gas/fire node) are sent to the two slave robots (R2, R3) by a centralised station. Slaves receive the map information, and they take the Dijkstra's shortest path algorithm to the target point. They equipped with grippers to capture the object. R3 took the same route of R2 as it is the shortest one, but its path was increased with last steps to face R2. To avoid the collision between robots, R2 precedes R3 with an edge. R2 checks are arriving of R3 with front ultrasound sensors. Then, the robots catch the object as shown in figure 17. The following algorithm illustrates the approach for safety purposes.

Algorithm: Extension for Safety Purposes

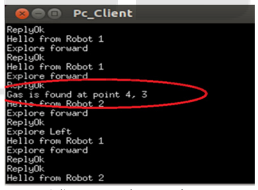
```
switch RobotsState do
    case Exploring do
        Masters robots apply Zigzag algorithm
         of multi-robots;
        foreach Robot do
            Check gas odor at each node;
            if GasOdor == true then
                Mark this node as gas source ;
                 Inform Centralized Station;
            if FireDetected == true then
                Mark this node as fire source ;
                 Inform Centralized Station;
            if (Object Beside Gas/Fire node ==
             true) then
                Mark this node as a target point
                 (potential victim);
                Inform Centralized Station;
                Update Map;
                After Ending Exploration, Stop
                 Master Robots;
        end
        RobotsState = TakingPath;
    case TakingPath do
        Slave robots start working;
        Apply Dijkstra algorithm;
        if (Robot1-ReachTarget == true) &
         (Robot2-ReachTarget == true) then
            Move gripper down;
            Catch Object, move gripper up;
end
```

(a)Khepera III robots.

(b) Experiment in the real world.



(c) Fire detection.



(d) Gas detection.

Figure 16: Gas or fire source detection in the system.

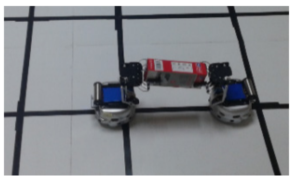


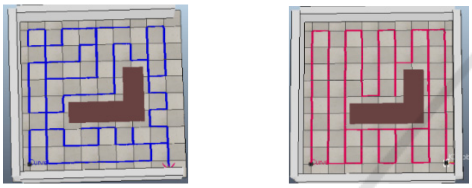
Figure 17: The slaves with the gripper on face to face.

5.4 Comparing the Results of Zigzag Algorithm towards Heuristic SRT for Single Robot

Haitham proposed a heuristic SRT backtracking as an enhancement of the basic SRT exploration algorithm while the robot reaches valuable nodes instead of backtracking all the previous as in the basic sensor-based techniques (Hussieny et al., 2015).

The contribution of this approach is the selection of a valuable node. It is done with the help of the ray-casting algorithm that estimates the number of cells to be explored. The robot is equipped with 360º laser range finder. It starts by choosing any node to explore randomly depending on a sensor reading such that not to choose an obstacle node or a node visited before. If the robot reaches a close area, it chooses the nearest node by calculating a heuristic function, and it will take the A* path to it.

We compare the result of the heuristic SRT backtracking with a zigzag algorithm for a single robot. Considering the limitation of Khepera III robot capabilities, we use infrared sensors instead of laser range sensor. The robot can move in 4 directions (N, S, E, and W) directions. The experiments are done on a 2-D grid map on V-REP simulation as shown in Figures 18, 19 and 20.



(a) Heuristic SRT. (b) Zigzag on the map with 1 obstacle.

Figure 18: Shows the result of heuristic backtracking SRT and zigzag algorithm on 9*9 map with one obstacle.

Figure 19: It shows the result of simulation on a rectilinear map with 4 obstacles between heuristic SRT on the left figure and zigzag on the right figure using a single robot.

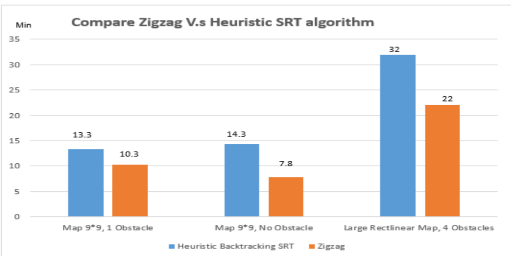


Figure 20: The result of comparison between heuristic SRT and zigzag algorithm of a single robot.

6 EVALUATION

There is a trade-off between the metrics evaluating the exploration approach performance like distance, time and area completeness percentage which comesover the exploration cost. A single “ExplorationIndex” is used to judge the performance of differentexploration strategies (Hussieny et al., 2015).

EI is directly proportional to the completeness C, and is inversely proportional to the exploration time(ET) and the travelled distance (TD), and the normalised number of nodes (F). The larger the values of this index, the better the performance of a strategy. The index is defined by equation 1.

$$EI = \frac{W_c * C}{(W_t * ET) * (W_d * TD) * (W_f * F)} \quad (1)$$

Where $F = N_{tot} / N_{actual\ visited}$ and Wc, W_t, W_d and W_f are the proportional weights added to measure the contribution of each factor to a metric. Different environments were tested to show a high Exploration index (EI) as shown in figure 21. Assume $W_c = W_t = W_d = W_f = W$ in equation 1.

Table 1: Shows the EI of Zigzag and heuristic SRT.

Test	C	ET(min)	TD(m)	F	EI
T1:Zigzag	1	10.33	42	1	$2.3*10^{-3}$
T1: SRT	1	13.25	56	1	$1.3*10^{-3}$
T2:Zigzag	1	8	40	1	$3.1*10^{-3}$
T2: SRT	1	14.3	58	1	$1.2*10^{-3}$
T3:Zigzag	1	22	70.5	1	$6.5*10^{-4}$
T4: SRT	0.97	32	83	1.03	$3.5*10^{-4}$

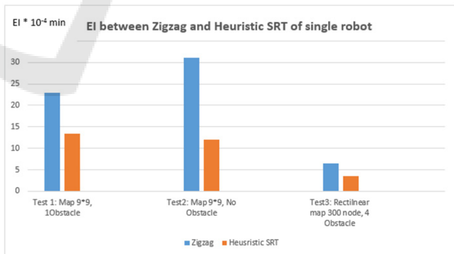


Figure 21: Chart the EI of Zigzag and heuristic SRT.

We evaluate the Zigzag Algorithms vs. Albers by measuring the exploration time on different case studies on an experimental work and calculating the tight bound time complexity for each algorithm as shown in Table 2 where Ɵ is the tight bound function, n is number of nodes in a column, Obstacles (n) is function determines total complexity for all obstacles and C is a constant which include nodes hidden inside obstacles to remove them from the calculation.

Table 2: Shows the tight bound complexity of Albers VS. Zigzag algorithm for single and multi-robots.

Algorithm	Tight bound time complexity
Albers single robot	Θ ((2 n^2 + 5n) + Obstacles (n) - C)
Zigzag single robot	Θ ((n^2 – n -2) + Obstacles (n) - C)
Albers multi-robots	Θ ((4n +($\frac{2n^2}{NR}$) +n) + Obstacles(n) - C)
Zigzag multi-robots	Θ (($\frac{n^2}{NR}$) + (n-2)+ Obstacles(n) - C)

7 CONCLUSION

The paper proposes an algorithm for exploring unknown grid map using a centralised multi-robot system called Zigzag Multi-robots for safety purposes. If the map is large, and the common obstacles are minimum, the exploration time of 4 robots on a rectilinear map is equal to 0.172 of Albers time. Nevertheless, in the worst case, when there are common obstacles, the time of 4 robots in the map is equal to 0.284 of Albers time. Also, we used this algorithm as a part of the combined approach for the gas/ fire searching using two robots as masters and two robots as slaves to hold the potential victim. Additionally, we compare Zigzag with heuristic SRT, and we measure the evaluation index where the higher the EI, the better performance of the algorithm.

In future work, we will apply Zigzag multi-robots algorithm on a decentralised system in bigger maps with 8 and 16 robots and compare the results with an extension for heuristic SRT multi-robots approach.

ACKNOWLEDGMENTS

The author would like to thank Dr. Mohammed Hamdy from Fayoum University, Dr. Reda El-Bssuieny, Dr. Haitham El-Hussieny from Banha University and Dr.Alaa sheta from ERI. The first author is supported by a scholarship from the Egypt Scientific Research which is gratefully thankful.

REFERENCES

A. Marjovi and L. Marques. (2011). "Multi-robot olfactory search in structured environments". *Robotics and Autonomous Systems*, (pp. 867–881, no. 11.).

Ashry, S., & Gomaa, W. (2016). "Exploration Of Unknown Map For Gas Searching And Picking Up Objects Using Khepera Mobile Robots". *the 13th International Conference on Informatics in Control, Automation and Robotics* (pp. 294 – 302. (Vol 2)). In Portugal,July 29-31: ICINCO.

B.Yamauchi. (1998). “Frontier-based exploration using multiple robots”. *in Proceedings of the 2th international conference on Autonomous agents. ACM,*, (pp. 47–53).

B.Yamauchi. (1997). “A frontier-based approach for autonomous exploration”. *In Computational Intelligence in Robotics and Automation* (pp. 146–151). IEEE, 1997.

D. Martinez, T. Pallej`a, J. Moreno, M. Tresanchez, M. Teixid´o, D. Font, A. Pardo, S. Marco, and J. Palac´ın. (2014). “A mobile robot agent for gas leak source detection”. *In Trends in Practical Applications of Heterogeneous Multi-Agent Systems* (pp. 19–25). The PAAMS Collection. Springer, 2014.

E.W.Dijkstra. (1959). " A note on two problems in connexion with graphs. Numerische mathematik". 269–271, vol 1.

G. Oriolo, M. Vendittelli, L. Freda, and G. Troso. (2004). “The srt method: Randomized strategies for exploration”. *In Proceeding of IEEE International Conference on Robotics and Automation* (pp. 4688–4694, vol. 5). ICRA.

H. Ishida, A. J. Lilienthal, H. Matsukura, V. H. Bennetts, and E. Schaffernicht. (2016). “Using chemical sensors as noses for mobile robots”. *Essentials of Machine Olfaction and Taste*, (pp. 219–245).

Haitham El-Hussieny, Samy Assal, and Mohamed Abdellatif. (2015, Apr 1). "Robotic exploration: new heuristic backtracking algorithm, performance evaluation and complexity metric". *International Journal of Advanced Robotic Systems*, 12(4), p.33.

J. M. Soares, A. Marjovi, J. Giezendanner, A. Kodiyan, A. P. Aguiar, A. M. Pascoal, and A. Martinoli. (2016). “Towards 3-d distributed odour source localisation: an extended graph-based formation control algorithm for plume tracking”. *In Intelligent Robots and Systems (IROS)* (pp. 1729–1736). IEEE/RSJ International Conference.

M. Turduev, G. Cabrita, M. Kırtay, V. Gazi, and L. Marques. (2014). “Experimental studies on chemical concentration map building by a multi-robot system using bio-inspired algorithms”. *Autonomous agents and multi-agent systems*, (pp. 72–100, vol. 28, no. 1).

S. Albers, K. Kursawe, and S. Schuierer. (2002). “Exploring unknown environments with obstacles”. *Algorithmica*, vol. 32, pp. 123–143.

S. Soldan, J. Welle, T. Barz, A. Kroll, and D. Schulz. (2014). “Towards autonomous robotic systems for remote gas leak detection and localization in industrial environments”. *In Field and Service Robotics* (pp. 233–247). Springer, 2014.

Shermer, T. C. (1992). “A linear algorithm for bisecting a polygon”. *Information processing letters*, (pp. 135–140, vol. 41, no. 3.).

W. Wang, M. Cao, S. Ma, C. Ren, X. Zhu, and H. Lu. (2016). “Multi-robot odor source search based on cuckoo search algorithm in ventilated indoor environment”. *n Intelligent Control and Automation*

(WCICA), 12th World Congress (pp. 1496–1501). IEEE, 2016.

W.Burgard, M.Moors, C.Stachniss, and F.E.Schneider. (2005). “Coordinated multi-robot exploration”. *Robotics*, (pp. 376–386, vol. 21, no. 3).

W.Sheng, Q.Yang, J.Tan, and N.Xi. (2006). “Distributed multi-robot coordination in area exploration”. *Robotics and Autonomous Systems*, (pp. 945–955, vol. 54, no. 12).

Rohmer, E., Singh, S. P., and Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. pp. 1321–1326.