

# MULTIPLE DRONES DRIVEN HEXAGONALLY PARTITIONED AREA EXPLORATION: SIMULATION AND EVALUATION

Ayush Datta  
Rahul Tallamraju  
Kamalakar Karlapalem

Agents and Applied Robotics Group  
International Institute of Information Technology, Hyderabad  
Gachibowli, Hyderabad, India  
{ayush.datta,rahul.t}@research.iiit.ac.in, kamal@iiit.ac.in

## ABSTRACT

In this paper, we simulated a distributed, cooperative path planning technique for multiple drones ( $\sim 200$ ) to explore an unknown region ( $\sim 10,000$  connected units) in the presence of obstacles. The map of an unknown region is dynamically created based on the information obtained from sensors and other drones. The unknown area is considered a connected region made up of hexagonal unit cells. These cells are grouped to form larger cells called sub-areas. We use long range and short range communication. The short-range communication within drones in smaller proximity helps avoid re-exploration of cells already explored by companion drones located in the same subarea. The long-range communication helps drones identify next subarea to be targeted based on weighted RNN (Reverse nearest neighbor). Simulation results show that weighted RNN in a hexagonal representation makes exploration more efficient, scalable and resilient to communication failures.

**Keywords:** Multi Robot Coordination, Drones, Cooperation

## 1 INTRODUCTION

### 1.1 Motivation

These days, drones have been used increasingly for exploration of an unknown area. There are various applications of area exploration like search and rescue, map building, intrusion detection, and planetary exploration. Building co-ordinated multiple basic drones is cheaper than one expensive and complex drone. The idea behind using co-ordinated drones is that if one drone takes a certain amount of time to explore a region, then two co-ordinated drones should take half the time taken by a single drone. Similarly, if more drones are exploring a certain region, they should reduce the exploration time drastically. Assuming that some identical drones are equipped with sensing, localization, mapping and communication capabilities, they need to explore the unknown area efficiently and reliably.

The challenge of exploring an unknown area with obstacles is that the drones tend to explore the same area multiple times, which further tends to increase due to lesser communication between them. Since

centralized coordination algorithms usually suffer from a single point of failure problem, it is desirable to have a distributed co-ordinated system to improve system reliability.

The drones have their own memory in which they keep track of the area explored and the information about the position of the obstacles. If they do not communicate this information among themselves, then they tend to explore the area which is already covered by the other drone. This further increases the exploration time. If they communicate less and exchange less information, then it leads to a more redundant exploration of the cells. At the same time, we need to reduce the amount of communication between the drones and make the system more resilient to communication failures. Further, the mechanism of co-ordination should be such that it is scalable to a large area with multiple drones.

Efficient exploration of the area means that drones do not explore the same area multiple times and they explore the complete area in a minimum amount of time and with minimum drone traveling distance and with minimal communication between them. To address these issues, we propose Hexagonally Partitioned Area Exploration using Reverse Nearest Neighbors. It is a modification of the distributed frontier based algorithm (Wang, Liang, and Guan 2011) with Particle Swarm Optimization (PSO). In this paper, we show that our solution outperforms the PSO model based approach.

The area to be explored is tessellated into hexagonal subareas, unlike the traditional quadrangular grid. First, the drones explore their own subarea using the traditional frontier based algorithm with a limited or short range communication with drones present in the same subarea. After exploration of the own subarea, the drones decide which subarea to explore based on weighed reverse nearest neighbor (RNN). This is aided by the long range communication to figure out which drones are planning to approach which subarea. The RNN assigns the drones to the subareas in such a way that the communication and area exploration time reduces drastically.

## **1.2 Previous Work**

The initial work in the field of area exploration started with a single autonomous robot with sensing, localization and mapping. The work on multi-robot exploration was started by Yamauchi (1997) who introduced distributed frontier algorithm. Since there is no explicit co-ordination among the robots, they tend to move towards the same frontier cells which introduces inefficiency. Based on similar frontier concept, Simmons et al. (2000) developed a semi-distributed multi-robot exploration algorithm which requires a central agent to evaluate the bidding from all the other robots to obtain the most information gain while reducing the cost, or the traveling distance. Berhault et al. (2003) used Combinatorial Auction for the area exploration. The disadvantage of the bidding algorithms is that they have a single point of failure and since the communication cost increases drastically with the increase in the number of robots, these algorithms are not scalable.

There has also been work on heterogeneous robots exploring the area by Singh and Fujimura (1993). However, they do not focus on efficiency. Market based map exploration was also proposed by Zlot et al. (2002). Ant-inspired algorithm to divide the area into square cells on which the robots leave trails of their passage was done by Koenig and Liu (2001). All these strategies require a central shared memory where the robots can leave their marks in the environment. This introduces a single point of failure which makes the system less robust. Fu, Bandyopadhyay, and Ang (2009) made use of local Voronoi decomposition for task allocation in map exploration.

Kennedy and Eberhart (1995), proposed the PSO model which was inspired by the earlier research by bird flocks. Al-Obaidy and Ayesh (2008) added the power cost as the indicator in PSO model to the WSN optimization. Dasgupta, Cheng, and Fan (2009) built a robot team based on Reynolds' flocking model to improve the efficiency of the multi-robot map exploration. Wang, Liang, and Guan (2011) proposed

PSO model for frontier based exploration. Dornhege and Kleiner (2013) combined the concept of voids with frontier based approach to search for entombed victims in confined structures. Yoder and Scherer (2016) improvises frontier algorithm performance by using only state-changed space in the 3D map in each iteration. Mahdoui, Frémont, and Natalizio (2018) modifies frontier based approach where instead of sharing local maps, robots share their local frontier points with a so-called Leader.

In **Quadrangular RNN**, we show that RNN outperforms the traditional PSO model as it keeps a track of which drones are approaching which subarea instead of keeping track of drones that are in the path of the subarea to be explored. As proposed by Sheng et al. (2006), we need to use a nearness measure so that the agents remain within the communication radius. This approach leads to duplicate exploration of the area. To overcome this, in **Quadrangular RNN with SRC** we show that using two kinds of communication improves the exploration efficiency. Recently, a lot of research has been done on Multi-Robot area exploration and swarm intelligence, but most of these algorithms use a quadrangular grid to divide the area into subareas. In **Hexagonal RNN with SRC**, we show that using a hexagonal grid over a quadrangular grid reduces the exploration time as there are six degrees of directions to move instead of four. Finally, **Weighted HRNN with SRC** also highlights how this system is failure resilient, scalable and avoids the possible deadlocks.

## 2 PROBLEM FORMULATION

We assume that identical drones are equipped with sensing, localization and communication capabilities. The task is to explore an unknown area avoiding the obstacles efficiently and reliably, at the same time making it resilient to communication failure and hardware malfunctioning of the drone. Each drone is assumed to have its own memory which it updates after every step of exploration. In our testing scenario, the obstacles are randomly distributed in the bounded exploring area and each drone is assumed to correctly locate itself on the map. We have divided the complete environment into either square cells or hexagonal cells. Moreover, each drone is equipped to communicate and share information with companion drones.

In case of the map being divided into a quadrangular grid, we assume that the drones are equipped with eight sensors that can sense the environment in eight directions. The sensors are equidistant around the drone, so the drone can detect local environment in eight directions: Front, Right-front, Right, Right-back, Back, Left-Back, Left, and Left-front. The drones can move only in four directions: Front, Right, Back, Left. In case of the map being divided into hexagonal cells, we assume that the drones are equipped with six sensors which can sense the environment. The hexagonal grid allows the drones to move in six directions around it.

Each drone stores the environment in its memory in a cell based map. The cell based map is stored in a  $[n \times m]$  matrix for both hexagonal and square grid. Each cell in the map is either **unexplored**, **obstacle**, **visited** or it is a **frontier cell**. The frontier cell is the cell which has been sensed by drone, but it has not yet been visited. The whole environment is divided into subareas either in rectangular regions composed of square cells in it or in hexagonal subareas composed of hexagonal cells in it.

The drones are initially placed randomly into cells. The goal is to explore all the cells in the area at least once. The drones can communicate with companion drones and co-ordinate to reduce the time to cover the complete area. The communication allows the robots to exchange their local maps with the companion drones.

## 3 MULTI-ROBOT AREA EXPLORATION ALGORITHM

We present the algorithm in four steps where at each step we modify the algorithm in such a way such that the efficiency is progressively improved.

In all the approaches we divide our algorithm into two stages. Initially, the drones are randomly distributed among the subareas. In the first part, **Exploration state** the drones explore their own local subareas and later in the second part, **Moving State** the drones move towards the unexplored cells in the different subarea determined by RNN or PSO model. Each drone runs its own algorithm and maintains exploration or moving state. It is done in such a way that the total time to explore the complete area is minimized.

### 3.1 Quadrangular Reverse Nearest Neighbour (QRNN)

#### 3.1.1 Exploring local subarea (Exploration state)

In the exploration state, the drones keep moving towards the frontier cells recording the terrain of the nearby cells until there are no frontier cells left in the same subarea.

Initially, the cell where a drone is located is marked visited and the neighboring cells are marked frontier cells or obstacle cells in the local memory of each drone. After each time step, the drones move to the nearby frontier cell marking it covered and marking the new neighbor cells as frontier cells. The drone selects the frontier cell to move in this order North, South, East and, West. If there are no more neighboring frontier cells, then the drones move towards the nearest frontier cell. This goes on until there are no more frontier cells left in the subarea. The complete details are written in Algorithm1.

To find the nearest frontier cell in a grid with obstacles in between, we use A\* search to find the nearest frontier cell.

The function of the A\* search is given by:

$$f(i, j) = g(i, j) + h(i, j) \quad (1)$$

In equation (1),  $g(i, j)$  is the distance along the shortest path that connects the two cells 'i' and 'j'. Here  $h(i, j)$  is the displacement or crow fly distance between the two cells as explained by Nosrati, Karimi, and Hasanvand (2012). Algorithm 1 explains the Exploration State.

---

#### Algorithm 1 Subarea Coverage

---

```

1: procedure SUB-AREA COVERAGE
2:   Mark the current cell as a visited cell.
3:   Mark the neighbor unexplored cells in the same subarea as frontier cells or obstacles.
4:   if there are frontier cells around then
5:     go to one of them
6:   else if frontier cells in the same subarea present then
7:     go to nearest frontier cell along the shortest path
8:   else Moving state
9:   end if
10:  go to 2
11: end procedure

```

---

#### 3.1.2 RNN based co-ordination (Moving State)

In this state, the drones have to move towards an appropriate subarea. The drones should move towards a subarea ensuring that a minimal number of companion drones are approaching the same subarea.

Previous papers use PSO based cooperation model to avoid multiple drones heading towards the same cell. The PSO model uses the velocity which is updated according to previous best positions and the global best position achieved by its neighbors. The global best position is defined as the cell with a minimum number of other drones in its direction.

The disadvantage of the PSO model and can be explained using Figure 1.

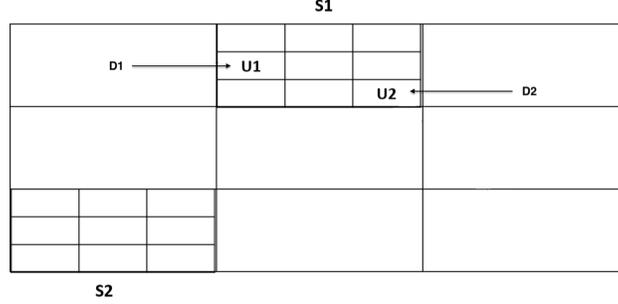


Figure 1: Quadrangular Grid of size 9X9 with 9 subareas.

In the above quadrangular grid,  $D_1$  and  $D_2$  represent the drones,  $S_1$  and  $S_2$  are unexplored subareas and  $U_1$ ,  $U_2$  are the unit cells of subarea  $S_1$ . Considering a case where only subareas  $S_1$  and  $S_2$  have unexplored cells left and rest all area is explored. If we follow the PSO model, then both drones  $D_1$  and  $D_2$  will move towards  $S_1$ , as unexplored cells  $U_1$ ,  $U_2$  are more closer to them than the cells of subarea  $S_2$  but ideally, since there is nobody to move to unexplored region  $S_2$ , the  $D_1$  should move towards  $S_2$  while the drone  $D_2$  should move towards  $S_1$ .

To overcome this problem, we use the concept of reverse nearest neighbor. The intuition behind this concept is to keep a track of the number of drones approaching that subarea. If a greater number of drones are approaching that subarea then the cost to travel to that subarea should be increased. RNN outperforms the traditional PSO model as it keeps a track of which drones are approaching which subarea instead of keeping track of drones that are approaching a particular cell. We call it reverse nearest neighbor because instead of focusing on the nearest area to a drone we focus on the nearest drone to a subarea.

The cost for the Drone  $D_i$  at  $i$  to travel to a cell  $U_j$  at  $j$  is calculated as shown in (2):

$$Cost(D_i, U_j) = g(i, j) + h(i, j) + \alpha * S_{(U_j)} \quad (2)$$

Here,  $S_{(U_j)}$  is the number of drones that were approaching the subarea in which  $U_j$  is located.  $\alpha$  is the constant just to scale this factor. Functions 'g' and 'h' are part of A\* search as explained earlier.

Using RNN, we can see that initially, both the drones will move towards the nearby unexplored region  $S_1$  but after a time step, the cost in RNN for  $D_1$  to move towards  $S_1$  will increase thereby, pushing  $D_1$  towards  $S_2$  and  $D_2$  towards  $S_1$ .

Long range communication facilitates that all the drones exchange information towards which subarea they tend to approach after each time stamp. Even if the communication fails with some far away drones, then also the algorithm keeps working based on the previous information. Hence, this algorithm is distributed and resilient to communication failure.

In Algorithm 2, the drone in **Moving State** makes a step towards selected subarea. After making a step it comes back to **Exploration State** if there are frontier cells around it.

**Algorithm 2** RNN based coordination

- 
- 1: **procedure** RNN BASED COORDINATION
  - 2:   Initialise, drones are not approaching any subarea in present iteration.
  - 3:   Communicate(Long Range) with companion drones to find which subarea they were approaching in previous iteration.
  - 4:   Calculate the cost  $Cost(D_i, U_j)$  to approach each cell based on number of drones  $S_{(U_j)}$  moving towards subarea of  $U_j$  in previous iteration.
  - 5:   Move towards the cell with minimum cost.
  - 6:   Update which subarea drone is approaching.
  - 7:   go to Exploration State (Algorithm 1)
  - 8: **end procedure**
- 

**3.2 QRNN with Short Range Communication (QRNN-SRC)**

This approach is similar to the previous approach, showing and improving upon the demerits of QRNN. In QRNN during the exploration state, it is possible to have multiple drones in the same subarea. This is more likely if the number of drones exploring the area is high. In QRNN, communication was only taking place in the movement state. The drones might follow one another to explore their own subarea. As all the drones follow the order of direction to move that is North, South, East and, West, the drones might end up covering a lot of cells multiple times.

As shown in Figure 2 if the drones are in the same subarea, then one drone might follow the path of a companion drone as the other drone is unaware of what the first drone has already explored.

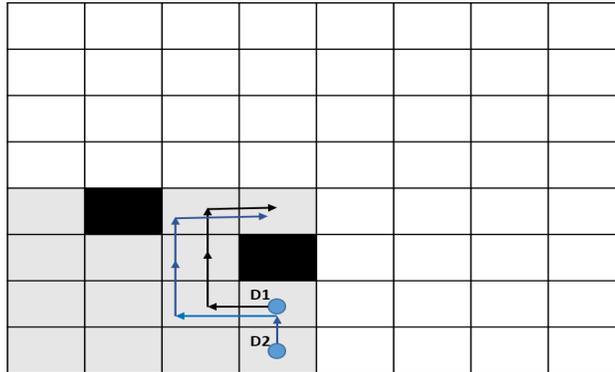


Figure 2: Quadrangular Grid.

Therefore, drones within a subarea need to communicate more frequently. The short range communication amongst the drones in smaller proximity helps avoid obstacles and re-exploration of cells already explored by companion drones located in the same subarea. The short range communication can also be multi-hop communication between the nearby drones leading to cascading information exchange.

The long range communication like in QRNN helps in deciding optimal next sub area to be targeted by individual drones based on RNN (Reverse nearest neighbor).

### 3.3 Hexagonal RNN with SRC (HRNN-SRC)

One of the most common and simplest representations of the area to be explored is the quadrangular grid map. In a quadrangular grid, the diagonal distance between two squares is bigger than the horizontal or vertical distance between two squares. Thus, it is harder to manage distances when drones move in diagonal, vertical and horizontal ways.

To simplify that problem, drones are frequently restricted to only make vertical or horizontal moves when quadrangular map representation is used. In a hexagonal grid, diagonal and vertical distances between two hexagons are the same. Thus, there is no need to worry about managing different distances scales or restrict drones to only make vertical or horizontal moves. Therefore, in this approach, we use the hexagonal cells and hexagonal subareas to represent the area to be explored.

The hexagonal approach gives drones six directions to move. Later simulation results show that there is an improvement in the efficiency using the hexagonal grid cells.

### 3.4 Weighted RNN-SRC (WHRNN-SRC)

In this approach, we improve upon the suggested RNN method in QRNN-SRC.

In the scenario, as shown in Figure 3, if there are drones  $D_{1-4}$  and there are two subareas  $S_1$  and  $S_2$ . More drones are approaching subarea 1, therefore using Reverse Nearest Neighbor  $D_4$  will move towards  $S_2$ . But, if  $S_2$  has only one cell left to be unexplored and the complete  $S_1$  is unexplored then  $D_4$  should move towards subarea 1. Figure 4 explains the deadlock scenario which is explained later.

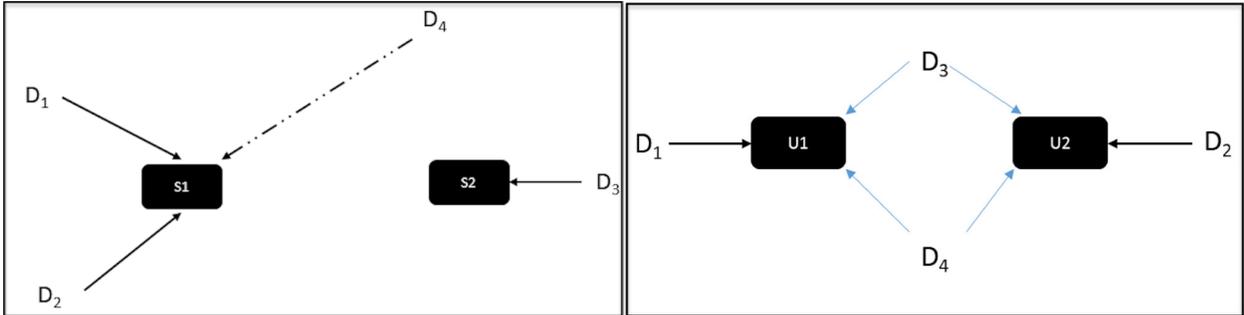


Figure 3: Drones are approaching subareas.

Figure 4: Deadlock Scenario.

To address the above demerit, we propose a weighted RNN algorithm, wherein each subarea has a weight proportional to the number of cells presently unexplored in it. Now, the cost for the drone  $D_i$  to travel to a cell  $U_j$  is calculated as shown in formula (3):

$$Cost(D_i, U_j) = g(i, j) + h(i, j) + \alpha * S(U_j) - \beta * \frac{W_{S_j}}{subarea.height * subarea.width} \quad (3)$$

Here,  $W_{S_j}$  represents the number of cells unexplored in the subarea containing cell  $U_j$  in the previous iteration. The above formula penalizes agents more if they move towards a subarea where drones are already approaching by increasing cost. It also reduces the penalty if they move towards a completely unexplored subarea.  $\beta$  is the constant which is multiplied by the percentage of unexplored cells in that subarea.

The weighted RNN algorithm improves the efficiency keeping the system resilient to failure of communication. The efficiency is defined as minimizing number of times the cells are covered.

In this algorithm, we can see that towards the end of the simulation when only a few unexplored cells are left then, all the drones move towards few remaining unexplored cells unnecessarily. Hence, towards the end when the number of unexplored cells is comparable to the number of drones, we apply the bidding algorithm to further optimize the results.

Since the RNN works at every time step, many times this can result in deadlocks.

In the scenario, as shown in Figure 4, initially the drones  $D_3$  and  $D_4$  move towards unexplored region  $U_1$ . In the next time step, since both of the drones proceed towards  $U_1$ . Now, using RNN the cost to the  $U_1$  increases as two drones are trying to approach it. So, both the drones start moving towards unexplored area  $U_2$ . This way both the drones get stuck in the cycle causing a deadlock. To avoid this problem, we have used a timestamp mechanism, wherein the decision to move is delayed by an iteration. This way the deadlocks are also handled.

#### 4 EXPERIMENTAL RESULTS

We have tested our proposed approaches and compared them with the traditional PSO model. The experiments were simulated with a large number of grid cells (Max 10,000) and with a maximum of 200 drones. In the simulation experiment, we compared our approaches with the traditional PSO based model.

The various parameters for the simulation experiment are shown in Table 1:

Table 1: Parameters of the Simulation Experiment.

Parameters	Values
Grid-size{height, width}	40x40, 60x60, 80x80, 100x100
Size of Sub-area taken	{4x4},{5x5},{8x8},{10x10}
Number of Agents	5, 10, 25, 50, 100, 150, 200
Number of Obstacles	20%
Radius of short range communication (Euclidean Distance)	5, 10, 15, 20
Constant ' $\alpha$ ' which gives weight to RNN	2-10

The traditional PSO based model, QRNN and QRNN-SRC are simulated on the quadrangular grid as shown in Figure 5. The HRNN-SRC and WHRNN-SRC are simulated on the hexagonal grid as shown in Figure 6.

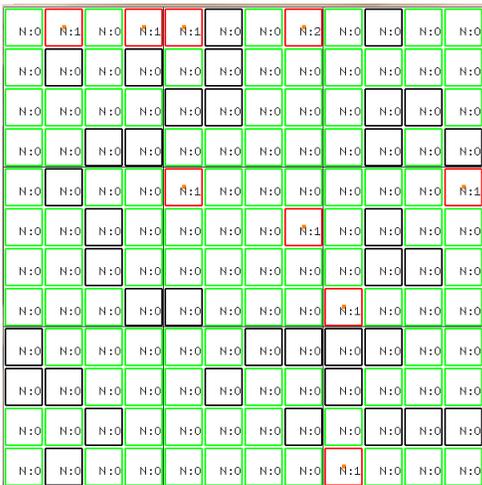


Figure 5: Quadrangular grid.

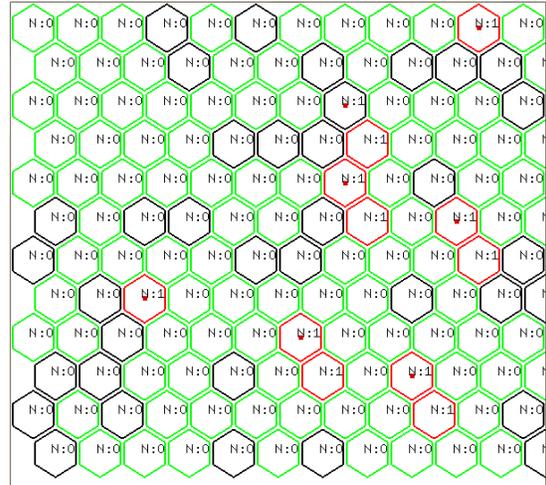


Figure 6: Hexagonal Grid.

In Figure 5 and Figure 6, the cells outlined in black color represents the obstacles, visited cells are outlined in red color and the unvisited cells are outlined in green color. The simulation environment shown above was created in SWIFT using OpenGL for visualization. The number inside the cell represents how many times that cell has been explored.

Figure 7 compares the performance of PSO, QRNN-SRC, HRNN-SRC and, WHRNN-SRC, with the increase in drones keeping the area to be explored constant. Here the grid size is 1600(40 x 40) cells. It is clear from the figure that, for Weighted RNN, with increasing drones, the average number of cells re-explored is much less compared to the traditional PSO model. This proves that the algorithm is efficient and scalable.

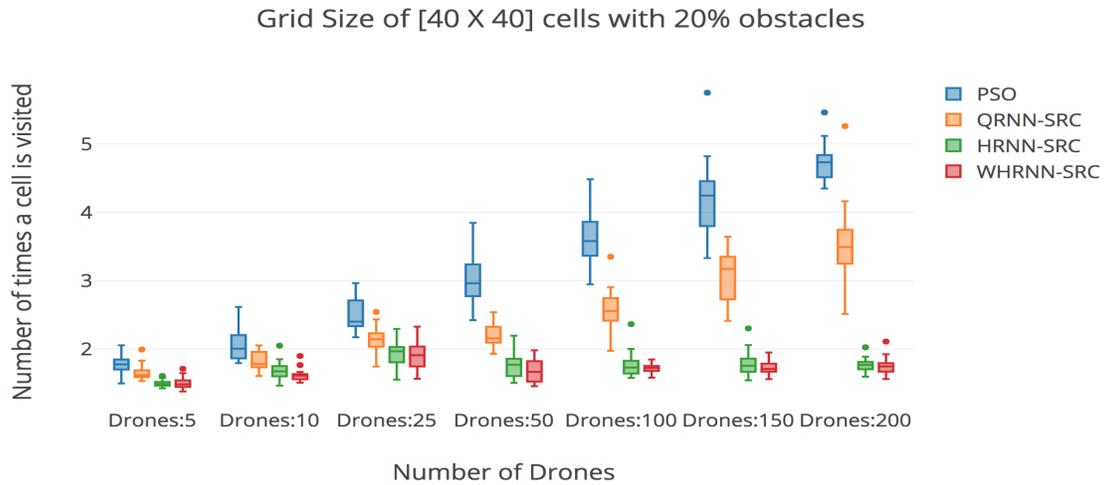


Figure 7: Number of drones vs Average Re-explored Cells.

Figure 8 represents the performance of Weighted RNN vs traditional PSO model, with an increase in grid size keeping the number of drones as constant. Here the approaches are compared taking 50 drones with an increase in the grid size. In this case also we can see that the weighted RNN outperforms the PSO based model. Our solution performs better not only in terms of the average number of times a cell is visited but also it has much less variance.

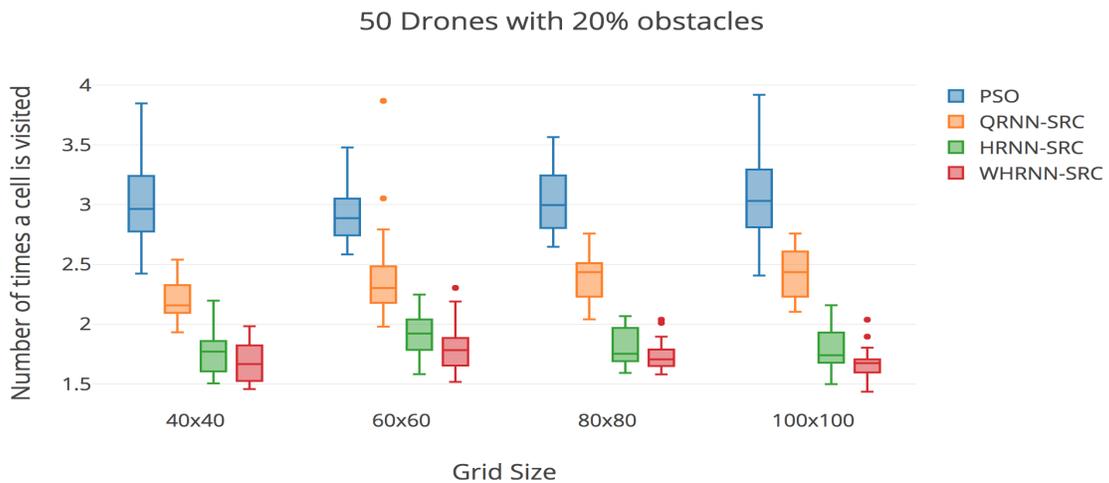


Figure 8: Grid Size vs Average Re-Explored Cells.

The above box plots are generated with extensive grid search on the parameters as shown in Table 1. For each combination of number of drones and size of grid, the simulation environment is created 20 times with 20% random obstacles. For each of the simulation environment grid search on parameters as shown in Table 1 is performed. For each combination of algorithm type, number of drones and size of grid the best parameters achieved will be different. For instance, with an increase in the number of drones, the value of ‘ $\alpha$ ’ should increase to spread the drones farther away.

For all the experiments, parameters which give the best result after applying grid search are used.

#### 4.1 EVALUATION OF FAILURES

In real time, it is very probable that failures might occur in the exploration process which can lead to massive delays in task completion.

There are two kinds of failures that can occur: **1. Temporary Failures:** Drones are affected by temporary failures due to weather and wind conditions, terrain features, communication latency, etc. The system can recover from such failures by incorporating a robust control algorithm into the drones **2. Permanent Failures in Drones:** Permanent failures in communication between drones can lead to massive task delays or permanent sensor failures can lead to partial exploration and mapping or motor and controller failures can lead to odometry/state estimation errors and accidents causing mid-air drone collisions.

Communication failures are usually simulated using MTBF (Mean Time Between Failures). At the time ‘ $t$ ’ the reliability of communication system  $R(t)$  for a drone is given by (4):

$$R(t) = e^{-\frac{t}{MTBF}} \tag{4}$$

Failure in communication leads drones to re-exploration of the areas already explored by the companion drones. The plot as shown in Figure 9 is simulated taking MTBF of 3 for all drones.

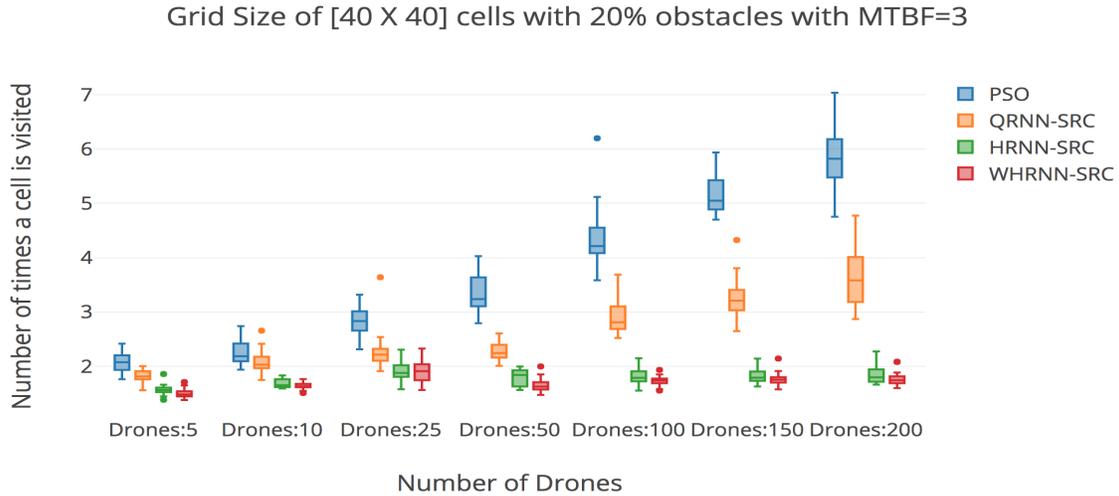


Figure 9: Number of drones vs Avg Re-explored Cells with MTBF=3.

After a lapse in communication as soon as the communication starts working the drone gets the latest information regarding the area explored by the companion drones. The results above show that our algorithm outperforms the traditional PSO even in case of communication failures.

## 5 SUMMARY

In this paper, we improved upon the existing PSO algorithm for coordinating a swarm of drones while they are exploring an unknown environment. The goal of the paper is to reduce the redundancy and achieve complete map coverage. Our proposed algorithm is scalable and outperforms even in case of failure in communication among drones. Using two kinds of communication on a hexagonal grid with the weighted RNN algorithm improved the efficiency drastically. In the future, the same approach can also be applied to the continuous region rather than the discrete cells taken presently.

## REFERENCES

- Al-Obaidy, M., and A. Ayeshe. 2008. "Optimizing autonomous mobile sensors network using PSO algorithms". In *Computer Engineering & Systems, 2008. ICCES 2008. International Conference on*, pp. 199–203. IEEE.
- Amigoni, F., N. Basilico, and A. Q. Li. 2013. "How much worth is coordination of mobile robots for exploration in search and rescue?". In *RoboCup 2012: Robot Soccer World Cup XVI*, pp. 106–117. Springer.
- Berhault, M., H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. 2003. "Robot exploration with combinatorial auctions". In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, Volume 2, pp. 1957–1962. IEEE.
- Burgard, W., M. Moors, D. Fox, R. Simmons, and S. Thrun. 2000. "Collaborative multi-robot exploration". In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, Volume 1, pp. 476–481. IEEE.
- Burgard, W., M. Moors, C. Stachniss, and F. E. Schneider. 2005. "Coordinated multi-robot exploration". *IEEE Transactions on robotics* vol. 21 (3), pp. 376–386.
- Dasgupta, P., K. Cheng, and L. Fan. 2009. "Flocking-based distributed terrain coverage with dynamically-formed teams of mobile mini-robots". In *Swarm Intelligence Symposium, 2009. SIS'09. IEEE*, pp. 96–103. IEEE.
- Dornhege, C., and A. Kleiner. 2013. "A frontier-void-based approach for autonomous exploration in 3d". *Advanced Robotics* vol. 27 (6), pp. 459–468.
- Fu, J. G. M., T. Bandyopadhyay, and M. H. Ang. 2009. "Local Voronoi decomposition for multi-agent task allocation". In *2009 IEEE International Conference on Robotics and Automation*, pp. 1935–1940. IEEE.
- Kennedy, J and Eberhart, R 1995. "Particle swarm optimization, IEEE International of first Conference on Neural Networks".
- Koenig, S., and Y. Liu. 2001. "Terrain coverage with ant robots: a simulation study". In *Proceedings of the fifth international conference on Autonomous agents*, pp. 600–607. ACM.
- Mahdoui, N., V. Frémont, and E. Natalizio. 2018. "Cooperative Frontier-Based Exploration Strategy for Multi-Robot System". In *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pp. 203–210. IEEE.
- Matignon, L., L. Jeanpierre, and A.-I. Mouaddib. 2012. "Distributed value functions for multi-robot exploration". In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1544–1550. IEEE.
- Middleton, L., and J. Sivaswamy. 2006. *Hexagonal image processing: A practical approach*. Springer Science & Business Media.
- Nieto-Granda, C., J. G. Rogers, and H. I. Christensen. 2014. "Coordination strategies for multi-robot exploration and mapping". *The International Journal of Robotics Research*, pp. 0278364913515309.

- Nosrati, M., R. Karimi, and H. A. Hasanvand. 2012. "Investigation of the\*(star) search algorithms: Characteristics, methods and approaches". *World Applied Programming* vol. 2 (4), pp. 251–256.
- Rogers III, J. G., C. Nieto-Granda, and H. I. Christensen. 2013. "Coordination strategies for multi-robot exploration and mapping". In *Experimental Robotics*, pp. 231–243. Springer.
- Senthilkumar, K., and K. Bharadwaj. 2012. "Multi-robot exploration and terrain coverage in an unknown environment". *Robotics and Autonomous Systems* vol. 60 (1), pp. 123–132.
- Sheng, W., Q. Yang, J. Tan, and N. Xi. 2006. "Distributed multi-robot coordination in area exploration". *Robotics and Autonomous Systems* vol. 54 (12), pp. 945–955.
- Simmons, R., D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. 2000. "Coordination for multi-robot exploration and mapping". In *AAAI/IAAI*, pp. 852–858.
- Singh, K., and K. Fujimura. 1993. "Map making by cooperating mobile robots". In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 254–259. IEEE.
- Wang, Y., A. Liang, and H. Guan. 2011. "Frontier-based multi-robot map exploration using particle swarm optimization". In *Swarm Intelligence (SIS), 2011 IEEE Symposium on*, pp. 1–6. IEEE.
- Wurm, K. M., C. Stachniss, and W. Burgard. 2008. "Coordinated multi-robot exploration using a segmentation of the environment". In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1160–1165. IEEE.
- Yamauchi, B. 1997. "A frontier-based approach for autonomous exploration". In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pp. 146–151. IEEE.
- Yoder, L., and S. Scherer. 2016. "Autonomous exploration for infrastructure modeling with a micro aerial vehicle". In *Field and service robotics*, pp. 427–440. Springer.
- Zlot, R., A. Stentz, M. B. Dias, and S. Thayer. 2002. "Multi-robot exploration controlled by a market economy".

## **AUTHOR BIOGRAPHIES**

**AYUSH DATTA** is a research student of "Agents and Applied Robotics Group" at "International Institute of Information Technology, Hyderabad". His interests lie in Robotics, Simulation and Path Planning. His email address is [ayush.datta@research.iit.ac.in](mailto:ayush.datta@research.iit.ac.in).

**RAHUL TALLAMRAJU** is a Ph.D. student of "Agents and Applied Robotics Group" at "International Institute of Information Technology, Hyderabad". His interests lie in Multi-Robot Systems, Multi-Agent Motion Planning. His email address is [rahul.t@research.iit.ac.in](mailto:rahul.t@research.iit.ac.in).

**KAMALAKAR KARLAPALEM** is a Full Professor at "International Institute of Information Technology, Hyderabad". He is the head of Data Sciences and Analytics Centre and Agents and Applied Robotics Group. He holds a Ph.D. from (Georgia Institute of Technology, USA), His research interests include Database system, data visualization, data analytics, multi agent systems, workflows and electronic contracts. His email address is [kamal@iit.ac.in](mailto:kamal@iit.ac.in).