International Journal of Control, Automation, and Systems 21(7) (2023) 2338-2349
http://dx.doi.org/10.1007/s12555-022-0182-9

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Cooperative Online Workspace Allocation in the Presence of Obstacles for Multi-robot Simultaneous Exploration and Coverage Path Planning Problem

Vishnu G. Nair, Adarsh Rag S., Jayalakshmi K. P., Dileep M. V., and K. R. Guruprasad*

Abstract: In this paper, a dynamic workspace allocation methodology for coverage path planning using multiple robots in the presence of obstacles is presented. The entire workspace is initially partitioned using the Manhattan Voronoi partitioning method, without considering the obstacles present, and the robots execute Multi-Robot Simultaneous Exploration and Coverage (MRSimExCoverage) using the Spanning Tree Coverage (STC) algorithm and cover the workspace. A dynamic workspace re-allocation strategy to optimize the area covered by each robot, whenever obstacles are detected, so as to avoid certain obstacle-induced coverage issues is studied. Simulation experiments within the Matlab/V-rep environment are used to demonstrate and validate the performance of the proposed algorithm. Though the authors used the STC algorithm for path planning for demonstration, any suitable coverage algorithm may be used.

Keywords: Coverage path planning, dynamic partition boundaries, Manhattan Voronoi, multi-robot systems, workspace allocation.

1. INTRODUCTION

Some of the major applications of mobile robots include cleaning, mine-sweeping, structural inspection, etc., in which the robots execute a complete coverage path passing through the entire accessible workspace. The process of planning such paths in multi-robot scenarios is a challenging problem. The issues such as avoiding duplication of the task, attaining proper coordination between the robots, optimal task allocation, etc, need special attention. The problem of efficient task allocation between the robots in the presence of obstacles is the main focus of this paper. In coverage path planning (CPP) using multiple robots, one of the effective ways of allotting the area to individual robots is the 'divide and conquer' approach in which the entire area is divided into as many cells as the number of robots in the system under consideration [14-18,25,26]. The individual robots then cover the allotted area without requiring any further communication with other robots in the system. In such a methodology, Voronoi partitioning or its variants [18,20,25,27-29] are generally used for partitioning the workspace. In most of the literature, the partitioning is performed based on the initial position of the robots which makes the process less efficient if there is no efficient initial deployment scheme available. But the centroidal Voronoi-based method [26] eliminates this shortcoming. In [27], the authors provided a methodology that combines both exploration and coverage for a multi-robot system. Once the partitioning is completed, the robots start covering the region while simultaneously exploring it. However, the effect of obstacles is not considered in [27]. The presence of obstacles changes the allotted area in terms of its size, which reduces the efficiency of the scheme. Hence, a proper strategy to allot the workspace on the go must be designed. This paper presents a multi-robot coverage path planning strategy by designing a dynamic workspace allocation method, which reduces the obstacle-induced inequality in workspace allocation.

Several multi-robot CPP algorithms are reported in the literature. A review on CPP algorithms is presented in [4]. Generally, the multi-robot coverage (MRC) algorithms re-

Manuscript received February 26, 2022; revised September 21, 2022 and October 20, 2022; accepted November 22, 2022. Recommended by Associate Editor Peng Yao under the direction of Senior Editor Euntai Kim. The authors would like to thank Manipal Academy of Higher Education, Manipal, for providing with the facilities needed for the proposed research work.

Vishnu G. Nair is with the Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal Karnataka 576104, India (e-mail: vishnu.nair@manipal.edu). Adarsh Rag S. is with the Department of Electronics and Communication Engineering, CMR Institute of Technology, Bengaluru, Karnataka 560037, India (e-mail: adarshrag@gmail.com). Jayalakshmi K. P. is with the Department of Electronics and Communication Engineering, St. Joseph Engineering College Mangaluru, Karnataka 575028, India (e-mail: lakshmi.rag08@gmail.com). Dileep M. V. was with the Chungnam National University, Korea (e-mail: dileeppsla@gmail.com). K. R. Guruprasad is with the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, India (e-mail: krgprao@iitk.ac.in).
* Corresponding author.

Springer

ported in the literature fall under one of the following three categories: a) Extended single robot techniques in which the single robot coverage algorithms are extended for accommodating the multi-robot scenario [7-10]; b) Algorithms dedicated for MRC applications [11-13]; and c) Partition and cover (divide and conquer) approach where the workspace is divided into as many cells as the number of robots [14-18,27]. The first two categories of the MRC algorithms require storing the coverage map and continuously communicating between either the robots within the group (in the case of a decentralized/distributed system) or with a central supervisor. The third category reduces these overheads as the area is partitioned and allotted to each robot, which results in task partitioning. The robots cover the allotted cells without any need for communication and the memory requirement is reduced as each robot needs to cover relatively a smaller region. Various strategies were proposed by researchers for partitioning the workspace, such as partitioning into strips of equal size [14,17], dividing into polygonal cells [15,16], and so on. Once the partitioning is completed, each of the partitioned areas (cells) is allotted to individual robots via static [15] or dynamic [14,16] approaches. A distributed Voronoi partitioning considers the initial robot positions thereby improving the partitioning efficiency [18].

The partition and cover approach using Manhattan Voronoi is considered in this paper [20], which converts an MRC problem into a number of single robot coverage problems. In [27], the authors presented the MRSimEx algorithm, which utilizes the advantages of both online and offline coverage strategies. The robots simultaneously explore and cover their allotted area by intermittent exploration during the coverage process, thereby reducing power consumption to a great extent. The exploration sensors need to be on only during the exploration phase. However, the partitioning process is offline, and hence, it is not possible in this approach to accommodate the obstacles during the partitioning process, unless they are known a priori. This affects the optimality due to nonuniform area allotment to the robots. If a large obstacle is inside a robot's allotted cell, that robot needs to cover a smaller area than others, resulting in reduced utilization of the available resources. In offline MRC algorithms using Voronoi partitioning, this problem will not arise since the obstacle scenario is already known a priori. In such cases, only the problem of partitioned cells being topologically disconnected may arise, which can be handled effectively as in [28].

In this paper, an efficient strategy is presented to improve the uniformity of the area allocation for MRC implementing MRSimExCoverage algorithm [27]. The allotted area is dynamically re-partitioned and re-allotted based on the obstacle scenario. Simulations are carried out on the V-Rep platform and the proposed methodology is shown to improve the coverage efficiency as compared to a static partitioning of the workspace. The main contribution of the paper is a workspace repartitioning scheme resulting in a more uniform workspace allocation amongst the robots performing online multi-robot simultaneous exploration and coverage, in the presence of initially unknown obstacles. Some of the advantages of the proposed algorithm in comparison with the existing work in the literature include: achieving a non-overlapping complete coverage, lower time taken to complete the coverage task, lower battery consumption, and robustness to failure of a few individual robots.

1.1. Motivation

As discussed in the previous section, the presence of obstacles leads to non-uniform workspace allocation between the robots. If a large obstacle is accommodating the cell allotted to a robot then that robot needs to cover a smaller area than others, resulting in reduced utilization of the available resources. Further, in the case of offline MRC algorithms using Voronoi partitioning, the problem of partitioned cells being topologically disconnected (non-contiguous) may arise. These are illustrated in Fig. 1. Computation of the area to be allotted to each robot using GM-VPC [25] seems to be a better candidate. But since it involves the computation of the geodesic distance around an obstacle, it is an explicit offline strategy. In the MRSimEx algorithm, the Manhattan distance metric is used [20,27], it can be implemented online. Note that in the absence of obstacles, the geodesic distance-based Voronoi partition is identical to the Manhattan distance-based Voronoi partition. In the presence of obstacles, the geodesic metric provides a more meaningful distance measurement. Hence, a combination of both these metrics is expected to result in a better distance metric if it can be computed on the go. The algorithm presented in this paper uses the MRSimex algorithm [27] for the initial partitioning and processing but uses the geodesic distance metric for repartitioning as and when the obstacles are identified. Thus, it is expected to provide a more efficient and uniform area partitioning strategy for multi-robot coverage path planning applications.

The rest of the paper is organized as follows: The problem statement is provided in Section 2. In Section 3, the proposed workspace allocation algorithm is discussed followed by an illustrative example in Section 4. The analysis of the algorithm is carried out in Section 5 and provided results of simulation experiments are in Section 6. The paper is concluded with a brief summary of the contribution and a discussion on the scope of future work in Section 7.

2. PROBLEM STATEMENT

Let $Q \subset \mathbb{R}^2$ be a bounded and contiguous workspace with known boundaries and containing n obstacles, $O_i \subset \mathbb{R}^2$, $i \in \{0, 1, 2, ..., n\}$. The obstacle characteristics such

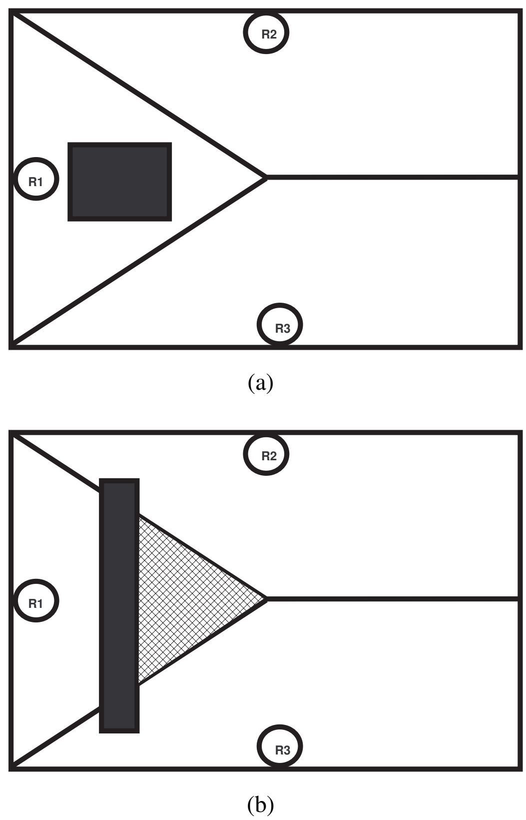


Fig. 1. (a) A large obstacle present in the cell allotted to a robot R1 results in it needing to cover only a (obstacle-free) part of the corresponding Voronoi cell, which is smaller than that allotted to robots R2 and R3, resulting in reduced utilization of the available resources. (b) The presence of an obstacle results in a non-contiguous Voronoi cell corresponding to the robot R1. The shaded region, a part of (obstacle-free) Voronoi cell corresponding to R1 is not accessible to it.

as shape, size, orientation, number, and location are unknown initially. Let N robots, having the capability of performing MRSimExCoverage-STC algorithm [27], are required to cover $Q\setminus O$, $O=\bigcup_{i=1}^{n} O_i$. The robots are attached with a square-sized coverage tool of size D. The sensors mounted on the robots for obstacle detection have relatively a larger range in comparison to their size.

Approximate cellular decomposition schemes (such as in [8]) is considered. The coverage is said to be resolution complete if all the free cells are visited by a robot and is said to be non-overlapping if each cell is visited at most once. The initial partitioning and allocation of the workspace between the robots is based on Manhattan distance [20,27] and centroidal Voronoi partitioning [26] schemes. Obstacles may be detected once the robots start covering their respective Voronoi cells, the presence of which reduces the uniformity in the workspace allocation, as original partitioning does not consider obstacles. The problem discussed in this paper is to find a methodology to efficiently allocate the workspace during the MRSimExCoverage process by accommodating the information about the obstacles, so as to utilize all the robots in the system to their best. The goal is to achieve a more uniformly partitioned workspace, covered completely without any coverage gaps or coverage overlaps. A map of the region is also obtained in the process as a by-product.

3. PROPOSED WORKSPACE ALLOCATION ALGORITHM

In this section, we present the proposed algorithm for online allotment of the workspace during MRSimExCoverage [27] process. MRSimExCoverage is an efficient scheme if power usage is the only criterion to be optimized since it uses the exploration sensors only at fewer instances in comparison with the conventional online coverage process. However, uniform workspace allocation is not guaranteed with this algorithm, particularly when initially unknown obstacles are present in the region. A "Partition and cover" strategy using Manhattan distance-based Voronoi partitioning [20] is followed for initial partitioning. Since geodesic distance-based Voronoi Partitioning [28] provides an effective distance measurement technique in the presence of obstacles, it is considered during the re-partitioning stage. In addition to converting a multi-robot coverage problem into a number of single-robot coverage problems, this will improve the efficiency of task (workspace) allotment in terms of its uniformity. Limited communication is needed since only a few messages related to the availability, task completion, final map, etc., are needed to be communicated between the robots. Now we provide a brief description of the underlying geodesic distance-based Voronoi Partitioning, Manhattan distance-based Voronoi partitioning, and MRSimExCoverage process below.

3.1. Geodesic VPC

The underlying partitioning scheme used in the partitioning process is Geodesic distance-based Voronoi partitioning (Geodesic VPC) [28]. The geodesic distance between any two points is the length of the shortest path between them. In the context of a mobile robot moving on a flat surface containing obstacles, the geodesic distance is defined as the shortest path between two points in question that avoids the obstacles. The geodesic distance-based Voronoi partitioning is given by [28]

$$V_i^G(\mathcal{P}) = \{q \in Q\setminus O \mid d_G(q,p_i) \le d_G(q,p_j),\ \forall j \in I_N\}, \tag{1}$$

where $d_G(q,p)$ is the geodesic distance between points q and p and Q is the workspace being partitioned.

Unlike in the case of the standard Voronoi partition using Euclidean distance metric, geodesic distance based

Voronoi partition scheme decomposes the free space rather that the whole region, which ensures that the corresponding cells are always topologically connected, or contiguous.

3.2. Manhattan VPC

As the underlying partitioning scheme used in the MRSimExCoverage process is Manhattan distance-based Voronoi partitioning, we provide a brief description of Manhattan VPC in this section [20]. The effectiveness of the coverage algorithm can be improved if the entire workspace is represented as a union of cells of size $2D \times 2D$, where D is the size of the square coverage tool footprint of the robots. If such a provision is not provided in the coverage algorithm, the robot has to retract and restart coverage in some left-out pockets such as in [7]. (see Figs. 2 and 4 in [20] as well as Fig. 2 in [27] for a detailed illustration). Also, as most of the robot motion is either in horizontal or vertical directions, especially in coverage applications, it is logical to use the Manhattan distance metric instead of the standard Euclidean one in the computation of Voronoi cells. The Manhattan distance-based Voronoi partitioning is given by [27]

$$V_i(\mathcal{P}) = \{q \in Q | d_m(q, p_i) \leq d_m(q, p_j), \forall j \in I_N\}, \quad (2)$$

where $d_m(q, p)$ is the Manhattan distance between points q and p and Q is the workspace under consideration.

After the partitioning, the robots execute any single-robot coverage algorithms to cover the allotted region. In this paper we use the Spanning tree-based coverage (STC) algorithm [8] as the underlying single-robot CPP algorithm.

3.3. MRSimExCoverage problem

In the multi-robot simultaneous exploration and coverage (MRSimExCoverage) problem [27], exploration and coverage problems are combined to extract the advantages of both online and offline coverage algorithms.

The robots are assumed to be equipped with relatively longer-range sensors, of the order of the size of the entire workspace. The robots start with generating a coverage path assuming no obstacles are present using the STC algorithm, the exploratory sensors are turned on only when a robot reaches a boundary cell between explored and unexplored cells (exploration windows). The information obtained during these intermittent exploration phases are used to update the map, which is then used for generating the coverage path. The coverage and exploration phases continue sequentially until the entire area is covered. The outcome is the completely covered workspace while the map of the area is obtained as a byproduct.

However, once the robots start the coverage process using the MRSimExCoverage STC algorithm, obstacles may be detected on the go, and the coverage load for the robots could become non-uniform as the obstacle-free regions within the Voronoi cells are now likely to be different depending on the obstacle scenario. To achieve a more uniform workspace allocation some portion of the Voronoi cells allotted to neighboring robots must be reallotted to the robot whose cell is occupied by the obstacle while maintaining contiguity. The detailed procedure is provided in the following section.

3.4. The proposed re-partitioning scheme

Let us consider a multi-robot scenario consisting of N robots covering a workspace as described in Fig. 2. Suppose in the process, i-th robot encounters an obstacle. The detection can be an entire obstacle or a part of it. Suppose that the robot acquired partial information about the obstacle and hence it knows that the part of the obstacle detected is occupying p_i number of major cells in its allotted Voronoi cell. This results in non-uniform task allocation as the i-th robot is now required to cover p_i cells lesser than the other robots, assuming an initial uniform partitioning. To ensure uniform area/task partitioning, the i-th robot now needs to be allotted p_i/N_{bi} number of major cells from the Voronoi cells of neighboring robots. Here, N_{bi} is the number of robots that share the common partition boundary along with the i-th robot itself. The new cell thus formed may no longer be a Voronoi cell, though for

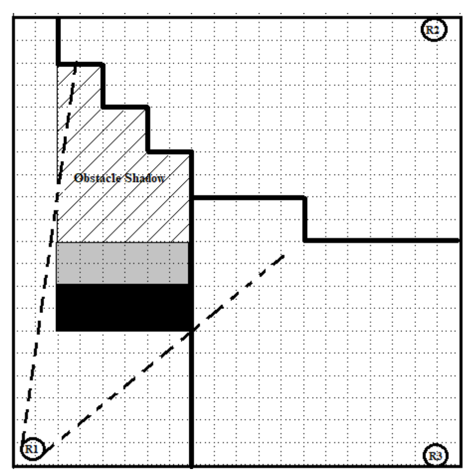


Fig. 2. The initial Voronoi partitioning based on Manhattan distance metric on a $2D \times 2D$ workspace is shown with bold lines. The rectangular region is the obstacle present inside the Voronoi cell of R1. The black region represents the area of the obstacle known to R1 after initial exploration from its current position. The grey area represents the part of the obstacle unknown at present. There are a total of 100 major cells of which 34 are allotted to R1, 33 to R2, and 33 to R3. Also, the striped region represents the portion of the workspace unknown to R1 due to the obstacle shadow.

simplicity we still call it a Voronoi cell. The partitioning efficiency is given

$$\eta = \frac{\max(A_{ri})}{A(Q)/N}, \tag{3}$$

where A_{ri} refers to the area covered by i-th robot (that is, the area of the i-th Voronoi cell), $A(Q)$ is the total area to be covered and N is the number of robots in multi-robot system. For optimal partitioning $\eta = 1$. Sub-optimal solutions are obtained when $\eta > 1$ [26].

Once an obstacle is detected, the MRSimExCoverage process is paused and the process of re-partitioning of Voronoi cells begins. After the completion of the re-partitioning process, the SimExCoverage process resumes. This switching between the processes of MRSimExCoverage and re-partitioning continues whenever the robot(s) encounters an obstacle, and the outcome a completely covered workspace with uniform load allocation to each robot.

Let us now set some rules for the major cell allotment to the i-th robot from its Voronoi neighboring robots

1) No covered major cells are allotted as this will result in coverage overlap.
2) No known obstacle-occupied major cells are allotted.
3) Only the major cells on the common boundary of the Voronoi cells (between the i-th robot and its neighboring robots) must be allotted.
4) Cells which are on the intersection of two boundaries of the Voronoi cells are preferred when all the above conditions are satisfied.

The pseudo-codes for the re-partitioning process followed by each robot is given as Algorithms 1, 2, and 3. Consider a scenario shown in Fig. 2 for illustration. Here three robots R1, R2, and R3 are required to cover a rectangular region. The initial Voronoi partitioning is shown with dark-lined borders. The rectangular region with a thick boundary with the Voronoi cell V_1 corresponding to the robot R1 is an obstacle. The black region within the obstacle is the part of the obstacle known to the robot R1 after initial exploration, while the grey area is the part of the obstacle unknown to robots at present. The shaded region beyond the obstacle is the obstacle-free area unknown to the robot R1 due to the obstacle shadow. The candidate major cells within V_2 and V_3, the Voronoi cells corresponding to the robots R2 and R3, to be allotted to the robot R1 on partial detection of obstacle within V_1 is shown in Fig. 3 as a shaded region. Rules 1)-4) above are used to select the right number of major cells among these candidate major cells and allot them to the robot R1.

4. ILLUSTRATIVE EXAMPLE

An illustrative example of the proposed allocation methodology is given in this section. Consider an area

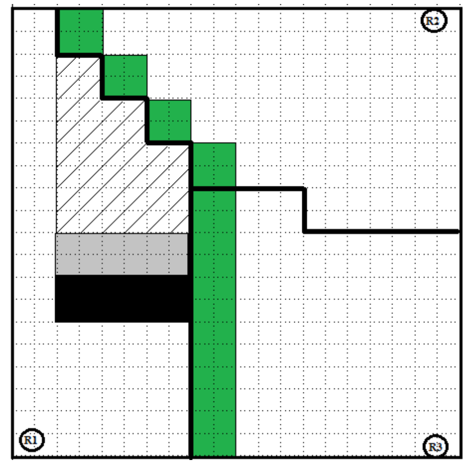


Fig. 3. The possible candidate major cells of the Voronoi cells corresponding to the robots R2 and R3 to be allotted to the robot R1 are shown as a green shaded region.

Algorithm 1: Uniform SimEx coverage.

Partition Q into $V_{2Dmi}(\mathcal{P}(0))$
while 1 **do**
  Explore
  if obstacle detected **then**
    Send "Repart" msg to Neighbors
    Re-Part
  end if
  if Repart msg received from a neighbor **then**
    Wait for Completion of RePart process
    Recreate ST over the explored, new, free major cells //As the Voronoi cell has now changed
  else
    Generate ST over the explored, new, free major cells.
  end if
  Generate CP through subcells circumnavigating ST edges on the right.
  while The starting subcell is reached **do**
    Move along coverage path by one subcell
    if 'exploration window' reached **then**
      Explore
    end if
  end while
end while

Algorithm 2: Explore.

Scan 360° sensor
Identify occupied/free space.
Update the 'occupied cell' list
Update the 'free cells' list
Update the 'unknown cell' list
Update the 'frontier cell' list.

Algorithm 3: Re-Part.

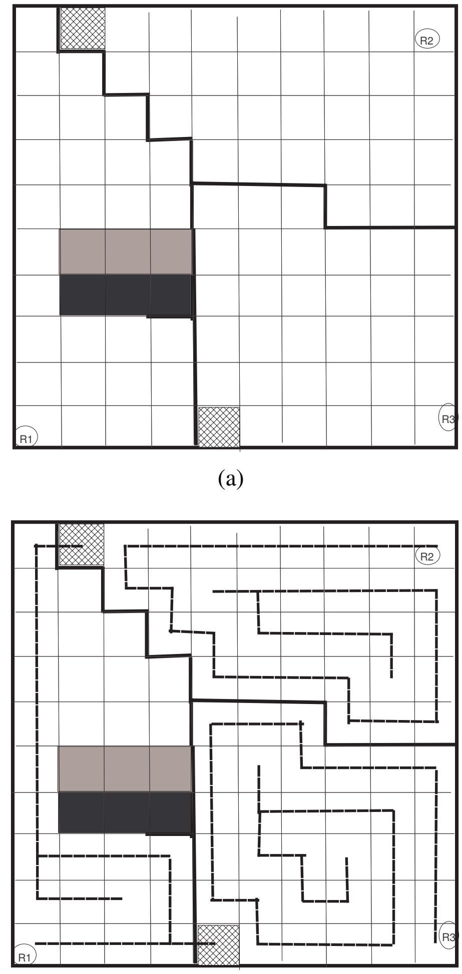
Communicate the obstacle detection information and "repart" msg to neighbors
Identify the number of major cells, p_i, occupied by the obstacle.
Identify p_i/N_{bi} cells from each of the neighboring robots' Voronoi cells which can be allotted based on the set rules.
if No suitable cell **then**
return else
    Allot the respective cells and communicate the new Voronoi cell boundaries to all the robots.
end if

as shown in Fig. 4, consisting of 100 major cells of size $2D \times 2D$ each. The area needs to be covered by three robots R1, R2, and R3. The rectangular obstacle is occupying 6 major cells which reduce the obstacle-free area to 94 major cells. The initial Voronoi partitioning based on the Manhattan distance metric is performed and is shown with bold lines. Out of 100 major cells, 34 are allotted to R1, 33 to R2, and 33 to R3, since the information regarding the obstacle is not considered during the partitioning phase.

The black region represents the part of the obstacle known to R1 after the first exploration. The grey area shows the unexplored region due to the obstacle shadow. Since, after the first exploration phase obstacle is detected, the MRSimExCoverage process is paused and the cell reallocation begins. Here the known area of the obstacle is occupying $p_1 = 3$ major cells. Therefore a total of one major cell ($p_1/N_{b1} = 3/3 = 1$) needs to be allotted from each of the Voronoi cells of R2 and R3. Though any boundary cells can be allocated as the coverage has not yet started, robots chose the major cells shaded in red and allot them to robot R1. Now the number of major cells for R1, R2, and R3 became 33, 32, and 32 respectively. This is shown in Fig. 4(a). At this stage, all the robots independently create ST within their respective Voronoi cells and start moving on the created CP through the sub-cells until the next exploration window is encountered.

In Fig. 4(b), the spanning tree generated after the first exploration phase is given. Since it is known that the newly allotted major cells are free, ST is generated considering them. The robots now start covering the region as shown in Fig. 5(a). The arrowed lines represent the path of the robots. When the robot R1 reaches an exploration window the coverage stops and the next exploration phase will start.

After the second exploration, the obstacle is completely identified by robot R1. Three more cells are detected to be occupied by obstacles. One each cell ($p_1/N_{b1} = 3/3 = 1$) needs to be allotted to the robot R1 (Fig. 5(b)). Now as robot R2 has already covered some of the major cells



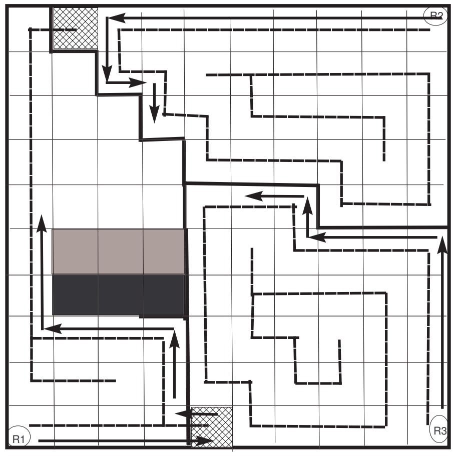
(a)

(b)

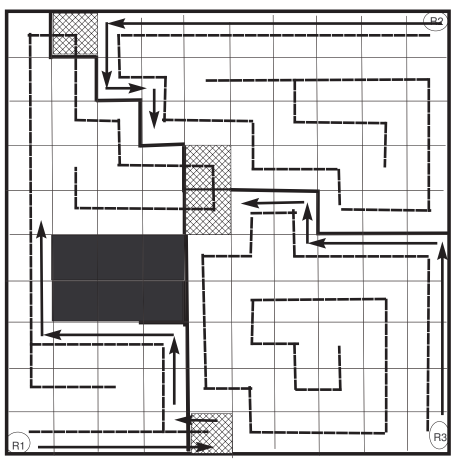
Fig. 4. (a) Re-partitioning Phase. The shaded cells (red) are allotted to R1. The number of free major cells for R1, R2, and R3 are now 33, 32, and 32, respectively. (b) MRSimExCoverage STC begins. The dotted line shows the spanning tree generated after the first exploration phase for all three robots. The empty cells represent the unexplored major cells due to the presence of obstacle shadow.

on the common Voronoi cell boundary, only those which have not yet been covered have to be allotted to robot R1. Robots chose the major cells from the available cells, and one cell each is allotted to R1. The major cells allotted are shown shaded with red in Fig. 5(b). The number of allotted free major cells for R1, R2, and R3 are now changed to 32, 31, and 31, respectively. Coverage resumes after the allocation.

The final scenario after the execution of the exploration, coverage, and the re-partitioning process is shown in Fig. 6. The dotted line shows the spanning tree generated and the arrowed lines represent the path of the robots. The entire workspace is explored and covered without any cov-



(a)



(b)

Fig. 5. (a) MRSimExCoverage STC. The dotted line shows the spanning tree generated after the first exploration phase for all three robots. The arrowed lines represent the path of the robots. R1 reaches an exploration window (boundary cell between explored and unexplored regions) and starts the next exploration phase. (b) Obstacle identified completely. Re-partitioning phase 2 begins. Again, one each cell (O/N) needs to be allotted to R1. From the available cells, one cell each is allotted to R1 (shown as shaded (red)). The lines with arrows represent the robots' path. The number of free major cells for R1, R2, and R3 are now 32, 31, and 31, respectively.

erage gap or overlap. The partitioning efficiency given in (3) with obstacles is same as that without obstacles. This indicates that with the proposed algorithm, the presence of obstacles will not result in the reduction of partitioning efficiency.

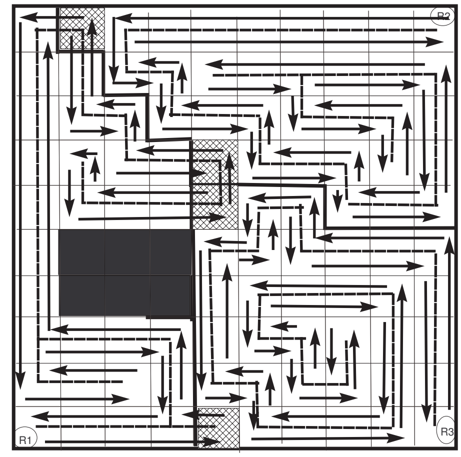


Fig. 6. Final scenario. The dotted line shows the spanning tree generated and the arrowed lines represent the path of the robots.

5. ANALYSIS OF THE PROPOSED ALGORITHM

In this section, a discussion on some of the properties of the proposed algorithm is discussed. As the Voronoi partitioning is recomputed as and when new obstacles are detected the partitioning is dynamic in nature. The following assumptions are made.

Assumption 1: Each of the initial Voronoi cells generated using the Manhattan distance metric in a $2D \times 2D$ workspace is contiguous. But such an assumption may fail in practical scenarios where there may be cases of topologically disconnected Voronoi cells after initial partitioning. Such cases can be handled using the approaches given in [23,24,27].

Assumption 2: In some practical scenarios, based on the shape of the allotted Voronoi cells, some robots may need to perform more turns which may use more battery power than moving along a straight line path. But in this paper, we neglect the battery power needed for turning.

Also, we assume that the time taken by the robots to cover a workspace using STC depends only on the length and not on the number of turns.

Assumption 3: We can regenerate spanning tree edges when some of the major cells through which spanning tree edges were earlier created are given away (allotted) to a neighboring robot.

5.1. Properties of the proposed algorithm

We list a few expected properties of the proposed algorithms without formal proof.

Property 1: With the proposed algorithm the robots successfully explore Q.

Property 2: With the proposed algorithm the robots successfully cover free space $Q \setminus O$ completely without any overlap.

Property 3: The proposed re-partitioning strategy achieves a more uniform task partitioning, provided the initial partitioning is uniform. This result follows from the fact that whenever an obstacle is detected, the number of free major cells shared by the neighboring robots is equal to the number of major cells that the obstacle occupies. This maintains the uniformity of the partitioning.

Property 4: With the proposed re-partitioning strategy ensures each Voronoi cell is made of contiguous collection of free major cells. This result follows from the fact that only those major cells which are connected to obstacle-free space are reallocated, thus not changing the local topology.

Property 5: With the proposed algorithm the coverage is achieved in lesser time.

6. RESULTS AND DISCUSSIONS

In this section, the results of simulation experiments carried out in the V-Rep simulation environment to demonstrate the proposed algorithm are presented. A differential wheeled DR12 robot model having an exploration sensor is used in the simulation. It is assumed that localization is available to the robot. For achieving localization there are several techniques available such as the use of blue-tooth, gyroscopes, odometry, and algorithmic techniques such as SLAM. In this paper, an office-like environment is considered and the simulation is carried out using five robots. Figs. 7(a)-7(e) show different stages of exploration and robot coverage path along with the updated Voronoi cell boundaries as the robots detect obstacles. Fig. 8 shows the final explored and covered workspace. It is observed that the entire workspace is explored and covered with a uniform workspace area (number of major cells) allocated to robots, without any coverage gaps or overlaps. The algorithm being distributed in nature, its performance is independent of the number of robots in the system.

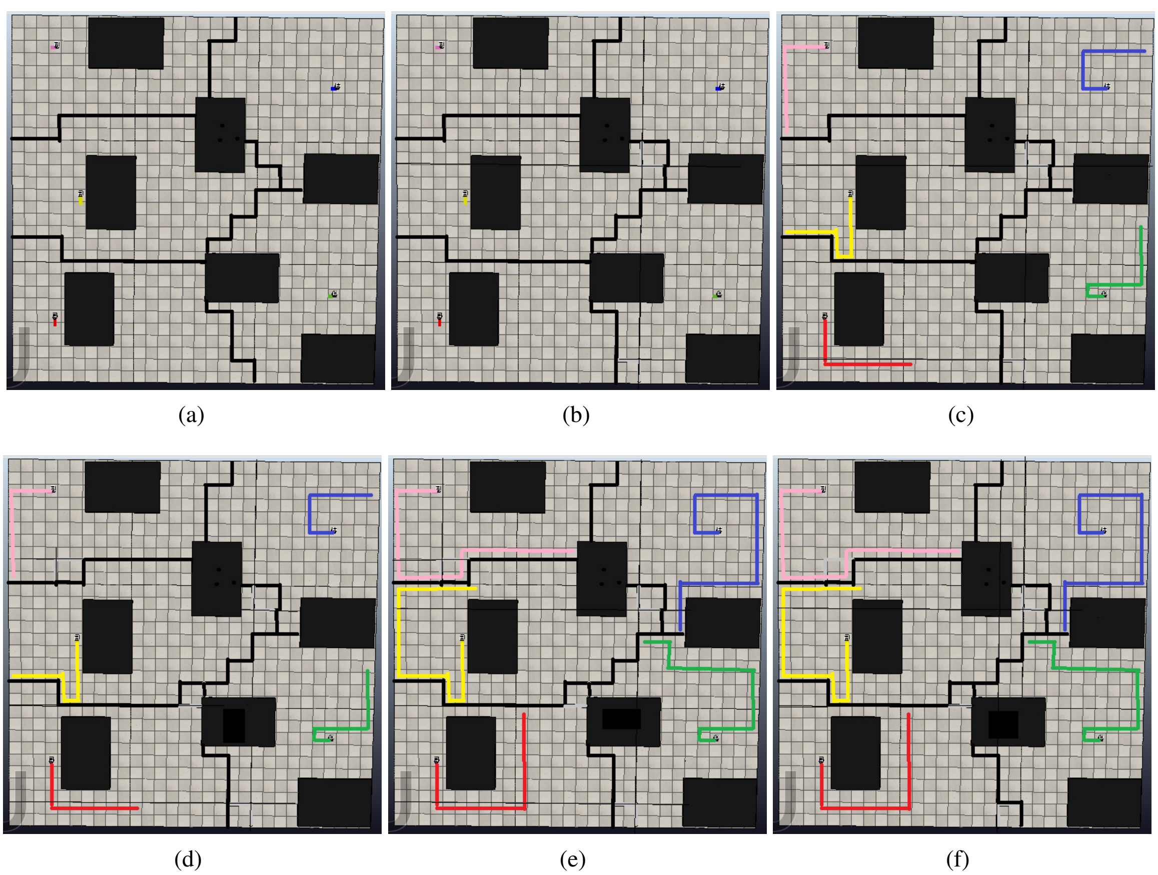


Fig. 7. Snapshots of various stages of MRSimExCoverage with five DR2 robots in V-rep simulation environment. Obstacles are shown with black rectangles, black thick lines represent the partition boundaries. (a) Initial partitioning. (b) Re-partitioned region after the first exploration. (c) The robots start covering after re-partitioning is complete. The robot paths are shown with colored (thin) lines. (d) Second re-partitioning. (e) coverage continues after the second re-partitioning. (f) Third re-partitioning.

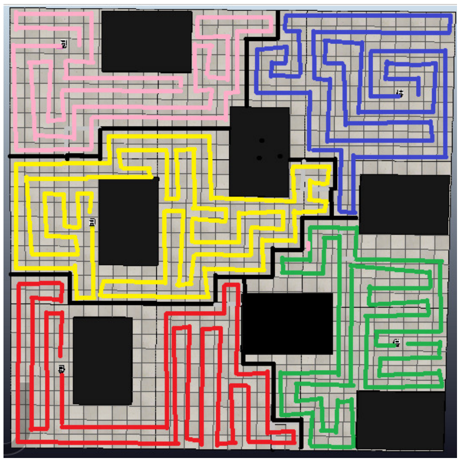


Fig. 8. Final coverage path by five robots showing complete, non-overlapping coverage with uniform area partitioning.

6.1. Comparison with other multi-robotic coverage strategies

This section gives a brief comparison of the proposed algorithm with uniform online workspace allocation with that of the standard MRSimExCoverage STC [27] and other CPP algorithms reported in the literature. All the algorithms provide a complete and non-overlapping coverage path. As with the standard MRSimExCoverage STC algorithm, as the range of the exploratory sensors increases, the number of exploration instances will decrease. Being STC-based algorithms, the proposed algorithm, the standard MRSimExCoverage STC [27], multi-robot spanning tree coverage (MSTC) [9] and multi-robot forest coverage (MFC) [10] achieve complete and non-overlapping coverage. But this is not the case with algorithms such as the spreading out technique [11], where the maximum reported coverage is 97.3% with overlap depending on the number and position of the robots.

Comparing the time taken for task completion, it was observed that the spreading out technique [11] as well as MSTC is on the higher side. The MFC algorithm takes less completion time than MSTC but more than that of MRSimExCoverage STC as well as the proposed one in this paper. Since the proposed algorithm needs to do re-partitioning and re-allocation of cells, the time of completion is more than that of MRSimExCoverage-STC since this is not included in it. To compare the expected battery usage we need to consider the same area of coverage and with the same number of robots for all the algorithms. In the case of online MSTC, and MFC, a similar amount of battery usage can be expected. In offline MSTC and MFC, lower battery consumption is expected as robots have to only move along the planned path unlike the online implementation, where robots need to turn physically and detect obstacles at every major cell. In the case of MRSimExCoverage-STC, as the number of exploration instances is much lower compared to the online MSTC or MFC algorithms, the battery consumption will be much lower, though the exact amount depends on the obstacle scenario. With the proposed methodology, in addition to having the advantage of lower exploration instances due to MRSimExCoverage, the battery consumption is expected to further reduce owing to reduced time of coverage due to uniform task (area) partitioning.

Some versions of MSTC have the property of robustness to failure of a few robots, while MFC as well as spreading out techniques do not have this property. Though MRSimExCoverage STC and the proposed algorithms do not have the robustness property, it can be incorporated with minor modifications of the algorithm.

As a final parameter, uniformity in task allocation, in terms of the area allotted to each robot is compared. For comparison, an area of 30 x 30 (900 major cells) is considered, out of which 182 cells are occupied by obstacles. Thus, the free space is 718 major cells. For a 5-robot system, the ideal area allocation will be 20%. The MFC and MSTC algorithms inherently achieve a more uniform task/area partitioning. In the case of MRSimExCoverage STC, robot 1 is allotted with 18.94% (i.e., 1.06% less from ideal), robot 2 is allotted with 17.27% (i.e., 2.73% less from ideal), robot 3 is allotted with 23% (i.e., 3% higher from ideal), robot 4 is allotted with 17.68% (i.e., 2.32% less from ideal) and robot 5 is allotted 15.45% (i.e., 4.5% less than ideal). But when the same conditions are implemented using the the proposed algorithm, the area allocation for robots 1 to 5 is 19.35%, 19.8%,19.6%, 19.1%, and 19.43% which are close to the optimal value of 20 as compared to MRSimExCoverage STC method. The result is shown graphically in Fig. 9. The η values for the same are

$\eta_1 = 0.965$,
$\eta_2 = 0.99$,
$\eta_3 = 0.98$,
$\eta_4 = 0.955$, and
$\eta_5 = 0.9715$.

A summary of the comparison of different algorithms is Tabulated in Table 1. It demonstrates the advantages of the proposed methodology with the existing algorithms. The major characteristics of the proposed algorithm are enumerated below.

- Uniform workspace allocation to the robots in the MRS.
- Provides complete coverage without any overlap.
- Less time to complete the coverage task.
- Less battery usage.
- Robustness to robot failures can be easily included without much changes in the algorithm.

Table 1. Comparison between the proposed algorithm and other similar ones.

	MRSimEx [27]	MSTC [9]	MFC [10]	Spreading out [11]	Proposed
Complete non-overlap coverage	Yes	Yes	Yes	No	Yes
Time taken	Low	High	Low	High	Low
Battery usage	Low	High	High	High	Low
Robust	Yes	Yes	No	No	Yes
Uniform task sharing	High	Low	Low	Low	Very high
Memory usage	Low	High	High	High	Low

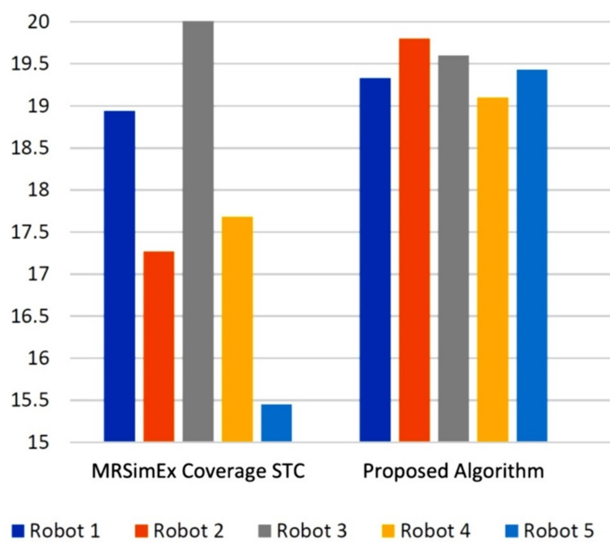


Fig. 9. Comparison between the proposed algorithm and MRSimEx Coverage STC. A 5-robot system is considered in a workspace of 900 cells out of which 718 cells are free. The uniformity in workspace allocation is evident. The vertical axis represents the percentage of the workspace allotted.

7. CONCLUSION

A novel methodology for online task allocation for multi-robot systems, performing MRSimExCoverage strategy for coverage path planning applications is proposed in this paper. The initial partitioning is based on Manhattan distance-based Voronoi partitioning scheme assuming a free workspace. As the robots perform coverage, and exploration simultaneously, obstacles may be encountered, and based on the information obtained, the Voronoi cell boundaries are reallocated ensuring the contiguity of the Voronoi cells. It was shown that the proposed uniform online workspace allocation scheme for solving the multi-robot simultaneous exploration and coverage achieves a more uniform task allocation among the robots. The simulations are carried out in a Matlab/V-Rep simulation environment using the DR12 robot model. It was observed that the proposed methodology guarantees partitioning efficiency the same as that of the initial partitioning without considering obstacles. The robots successfully cover and explore the area without any coverage gap or overlap.

CONFLICTS OF INTERESTS

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper. The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robot and Automation*, vol. 6, no. 4, pp. 462-472, 1990).

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

[3] V. R. Jisha and D. Ghose, "Frontier based goal seeking for robots in unknown environments," *Journal of Intel and Robot Systems*, vol. 67, pp. 229-254, 2012.

[4] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258-1276, 2013.

[5] D. C. Guastella, L. Cantelli, G. Giammello, C. D. Melita, G. Spatino, and G. Muscato, "Complete coverage path planning for aerial vehicle flocks deployed in outdoor environments," *Computers & Electrical Engineering* , vol. 75, pp. 189-201, 2019.

[6] P. Maini, K. Sundar, M. Singh, S. Rathinam, and P. B. Sujit, "Cooperative aerial-ground vehicle route planning with fuel constraints for coverage applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3016-3028, 2019.

[7] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247-253, 2000.

[8] Y. Gabriely and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry*, vol. 24, pp. 197-224, 2003.

[9] N. Agmon, N. Hazon, and G. A. Kaminka, "The giving tree: Constructing trees for efficient offine and online multi-robot coverage," *Annals of Math and Artificial Intelligence*, vol. 52, pp. 143-168, 2008.

[10] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada, pp. 3852-3857, 2005.

[11] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," *Proc. of the 6th International Symposium on Distributed Autonomous Robotics Systems*, Fukuoka, Japan, pp. 373-382, 2002.

[12] L. Ludwig and M. Gini, "Robotic swarm dispersion using wireless intensity signals," Gini, M., Voyles, R. (eds), *Distributed Autonomous Robotic Systems*, vol 7, Springer, Tokyo, pp. 135-144, 2006.

[13] Z Wilson, T Whipple, and P Dasgupta, "Multi-robot coverage with dynamic coverage information compression," *Proc. of the 8th International Conference on Informatics in Control Automation and Robotics*, Noordwijkerhout, The Netherlands, pp. 236-241, 20011.

[14] T. W. Min and H. K. Yin, "A decentralized approach for cooperative sweeping by multiple mobile robots," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Innovations in Theory, Practice and Applications*, Victoria, BC, Canada, pp. 380-385, 1998.

[15] S. Hert and V. Lumelsky, "Polygon area decomposition for multiple robot workspace division," *International Journal of Computational Geometry and Applications*, vol. 8, pp. 437-466, 1998.

[16] M. Jager and B. Nebel, "Dynamic decentralized area partitioning for cooperating cleaning robots," *Proc. of IEEE International Conference on Robotics and Automation*, Washington, DC, USA, pp. 3577-3582, 2002.

[17] I. Rekleitis, A. P. New, P. Rankin, E. Samuel, and H. Choset, "Efficient boustrophedon multi-robot coverage: An algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, pp. 109-142, 2008.

[18] K. R. Guruprasad, Z. Wilson, and P. Dasgupta, "Complete coverage of an initially unknown environment by multiple robots using Voronoi partition," *Proc. of the 2nd International Confence on Advances in Control and Optimization of Dynamical Systems*, 2012.

[19] J. Song and S. Gupta, "$\varepsilon^\star$: An online coverage path planning algorithm," *IEEE Transactions on Robotics*, vol. 34, pp. 526-533, 2018.

[20] V. G. Nair and K. R. Guruprasad, "Manhattan distance based Voronoi partitioning for efficient multi-robot coverage," Shreesha, C., Gudi, R. (eds), *Control Instrumentation Systems. Lecture Notes in Electrical Engineering*, Springer, Manipal, India, vol. 581, pp. 81-90, 2020.

[21] K. R. Guruprasad and T. D. Ranjitha, "ST-CTC: A spanning tree-based competitive and truly complete coverage algorithm for mobile robots," *Proc. of the 2nd International Conference of Robotics*, Society of India, Goa, India, 2015.

[22] R. E. Tarjan, "Data structures and network algorithms," *CBMS-NSF Regional Conference Series in Applied Mathematics CB44 Society for Industrial and Applied Mathematics*, Philadelphia, USA, ISBN:978-0-89871-187-5, 1983.

[23] K. R. Guruprasad and P. Dasgupta, "Distributed spatial partitioning of an initially unknown region for a multi-robot coverage application," A. Lomuscio, P. Scerri, A. Bazzan, and M. Huhns (eds.), *Proc. of the 13th International Conference on Autonomous Agents and Multiagent Systems*, Paris, France, pp. 1453-1454, 2012.

[24] H. Kurt, P. Dasgupta, and K. R. Guruprasad, "A repartitioning algorithm to guarantee complete, non-overlapping planar coverage with multiple robots," N. Y. Chong and Y. J. Cho, (eds), *Distributed Autonomous Robotic Systems*, Springer Tracts in Advanced Robotics, vol. 112, pp. 33-48, 2016.

[25] V. G. Nair and K. R. Guruprasad, "GM-VPC: An algorithm for multi-robot coverage of known spaces using generalized Voronoi partition," *Robotica*, vol. 38, no. 5, pp. 845-860, 2020.

[26] V. G. Nair and K. R. Guruprasad, "Centroidal Voronoi partitioning using virtual nodes for multi robot coverage," *International Journal of Engineering and Technology*, vol. 7, no. 2.21, pp. 135-139, 2018.

[27] V. G. Nair and K. R. Guruprasad, "MR-SimExCoverage: Multi-robot simultaneous exploration and coverage," *Computers and Electrical Engineering*, vol. 88, 106680, 2020.

[28] V. G. Nair and K. R. Guruprasad, "Geodesic-VPC: Spatial partitioning for multi-robot coverage problem," *International Journal of Robotics and Automation*, vol. 33, no. 3, pp. 189-198, 2020.

[29] V. G. Nair and K. R. Guruprasad, "2D-VPC: An efficient coverage algorithm for multiple autonomous vehicles," *International Journal of Control, Automation, and Systems*. vol. 19, no. 8, pp. 2891-2901, 2021.

Vishnu G. Nair received his Ph.D. degree from the National Institute of Technology Karnataka India. He is with the Department of Aeronautical and Automobile Engineering, Manipal Academy of Higher Education. His research interests include multi robotic systems.

Adarsh Rag S. received his Ph.D. degree from the Manipal Institute of Technology India. He is with CMR Institute of Technology, Bengaluru India. His research interests include algorithms and nanoelectronics.

Jayalakshmi K. P. is an Assistant Professor with the Department of Electronics and Communication Engineering at St. Joseph Engineering College, Mangaluru. Her research interests include digital control and artificial intelligence.

Dileep M. V. was with the Chungnam National University, Korea. His research interests include control systems and robotics.

K. R. Guruprasad received his Ph.D. degree from the Indian Institute of Science, Bengaluru, India. He is with the Department of Mechanical Engineering, IIT Kanpur India. His research interests include control engineering, robot motion planning, and multi-robotic systems and applications.