

Efficient Multi-Robot Coverage of a Known Environment

Nare Karapetyan^{1,2}, Kelly Benson¹, Chris McKinney¹, Perouz Taslakian^{2,3} and Ioannis Rekleitis¹

¹University of South Carolina, Columbia SC, USA,
nare@email.sc.edu, yiannisr@cse.sc.edu

²American University of Armenia, Yerevan, Armenia,

³Element AI, Montreal, Canada,
perouz@elementai.com

Abstract—This paper addresses the complete area coverage problem of a known environment by multiple-robots. Complete area coverage is the problem of moving an end-effector over all available space while avoiding existing obstacles. In such tasks, using multiple robots can increase the efficiency of the area coverage in terms of minimizing the operational time and increase the robustness in the face of robot attrition. Unfortunately, the problem of finding an optimal solution for such an area coverage problem with multiple robots is known to be NP-complete. In this paper we present two approximation heuristics for solving the multi-robot coverage problem. The first solution presented is a direct extension of an efficient single robot area coverage algorithm, based on an exact cellular decomposition. The second algorithm is a greedy approach that divides the area into equal regions and applies an efficient single-robot coverage algorithm to each region. We present experimental results for two algorithms. Results indicate that our approaches provide good coverage distribution between robots and minimize the workload per robot, meanwhile ensuring complete coverage of the area.

Index Terms—Multiple and distributed robots, path planning, coverage.

I. INTRODUCTION

The problem of coverage is common to many application domains. From household robotics, to automation in agriculture, search and rescue, environmental monitoring, and humanitarian de-mining [1], [2]. Formally, the *coverage problem* is defined as the planning of a trajectory for a mobile robot such that an end-effector, often the body of the robot, visits every point in the specified available space while avoiding existing obstacles. Utilizing multiple robots can potentially reduce the coverage time, and, in hazardous situations, such as de-mining, increase robustness by completing the mission even in the event of accidental “robot deaths”. Meanwhile, the utilization of multi-robot teams increases the algorithmic complexity and logistic management.

Rekleitis et al. [3] showed that finding a minimum-cost traveling path for multi-robot systems is NP-hard, while for single robot coverage there exists an efficient polynomial-time algorithm [4] based on the well-studied Chinese Postman Problem (CPP) [5]. The current paper extends the work of Xu et al. [4] on efficient coverage of a known arbitrary environment by a single robot to coverage by multiple robots.

We define optimal coverage as a *MinMax* problem: minimize the length of the longest coverage path over a group

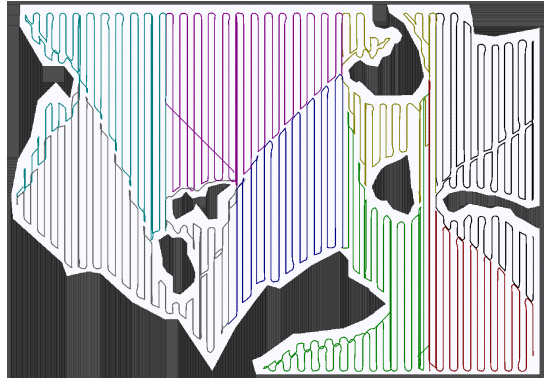


Fig. 1. A team of 8 robots covering a test environment.

of robots. Such a formulation, as opposed to minimizing the overall length of paths, helps us reduce the probability of situations when one or more robots remain idle while the rest do all the work. It is worth noting that the Chinese Postman Problem, utilized by Xu et al. [4], becomes NP-hard for the case of multiple postmen [6] when the objective is to minimize the maximum path length.

While for some applications, such as floor sweeping, scanning of every single point in free space is not crucial, in many situations missing a spot, such as in the case of de-mining operations, could have catastrophic results. With this consideration, coverage algorithms can be categorized as either *complete* or *randomized*. Complete coverage algorithms ensure that each point in the region is visited, while for a randomized approach there is no guarantee on full coverage of the entire area. This work aims to ensure complete coverage with minimum or no-overlapping regions.

Some coverage algorithms require prior information about the environment. Algorithms that operate on a known area utilizing an existing map are called *off-line* algorithms; otherwise they are termed *on-line* [1]. Furthermore, many existing solutions for multi-robot area coverage require constant communication, so the robots can coordinate their tasks. In many applications, however, it is not possible to guarantee continuous communication. In our work, we address the off-line complete coverage problem by utilizing the cellular decomposition approach. We further assume that the robots cannot communicate with each other during the operation and that each robot knows the map of the area it is supposed

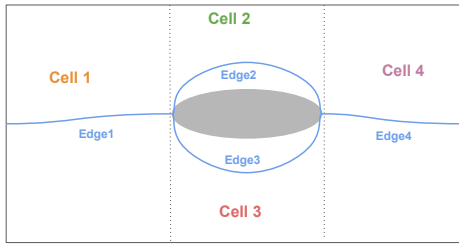


Fig. 2. Cellular Decomposition and Corresponding Reeb Graph to cover; see Fig. 1.

The next section presents an overview of existing solutions for the complete coverage problem for both single and multiple robotic systems. Furthermore, we define area coverage in terms of *arc routing* in graph theory and discuss some of its variations. In Section III, we present the problem statement and in the next two sections outline the proposed algorithmic solutions. Then experimental results of the graph routing phase and simulated experiments of the multi-robot area coverage algorithms are presented in Section VI. Finally, future work is highlighted in Section VII together with a discussion on lessons learned.

II. RELATED WORK

Most approaches to the area coverage problem first decompose the free area in an approach-specific fashion then proceed to compute a global solution. As such, coverage problems can be classified based on the approach taken in the decomposition step. The two most popular approaches are grid-based decompositions and cellular decompositions.

In grid-based approaches, the area to be covered is represented as a grid map. Each grid cell's value indicates whether or not it contains an obstacle. One of the earliest grid-based solutions, introduced by Gabriely and Rimon [7], was building a spanning tree from the map representation. Spanning tree based coverage was adapted for multi-robot systems as well [8]–[11]. The main challenge of grid-based approaches is that the robustness and efficiency of these methods depends on the resolution of the input representation. As the footprint increases the discretization of the grid becomes too coarse.

Another popular approach to coverage problems is based on the *cellular decomposition* of the free area. The idea behind cellular decomposition is to split the total free space into non-overlapping cells such that their union is the original area. The resulting decomposition is often represented by an adjacency graph, where either each node is a cell, and the nodes of neighbouring cells are connected by an edge; or each cell is an edge and the points where connectivity changes represent the vertices, the graph then is termed *Reeb graph*; see Fig. 2.

One of the widely used exact cellular decomposition algorithms is the *boustrophedon decomposition* introduced by Choset in 2000 [12]–[14]. The algorithm does not require any prior knowledge of the environment and is designed for single-robot systems. Each cell in a boustrophedon decomposition is defined by points along obstacles, called *critical points*. In contrast to other cellular decomposition algorithms,

the boustrophedon approach provides a more effective space division.

In 2010, Manadiar and Rekleitis [15] introduced a new efficient algorithm for single robot coverage based on the boustrophedon decomposition. However, in contrast to the original algorithm, the environment is assumed known and the problem is cast as an instance of the *Chinese postman problem (CPP)*: given a weighted graph, find a minimum-cost cycle that traverses each edge at least once. With this definition, the coverage problem was solved using a polynomial-time algorithm. That approach was extended to the field of aerial robotic coverage producing efficient coverage solutions taking into account real world considerations such as wind direction [4]. This paper extends the efficient coverage algorithm proposed above to multiple robots.

The work of Rekleitis et al. [3] extended the boustrophedon decomposition approach from single robots [12] to multi-robot systems for covering unknown arbitrary environments. Two types of algorithms were presented based on the communication restrictions, called *team-based coverage* and *distributed coverage*, both assuming an unknown environment and utilizing a sensor-based approach.

A recent study by Avellar et al. [16] on multi-robot coverage that is similar to that addressed by Xu et al. [4] operates in two phases. First the area is decomposed into line-sweeping rows, thus constructing a complete graph where the vertices consist of the borders of coverage rows and the launching point. The resulting graph is used in the second phase to solve the vehicle routing problem [17]. The authors claim this solution ensures minimum-time coverage for Unmanned Aerial Vehicles (UAV), in contrast to other existing solutions. One of the main objectives of this approach is to minimize the number of robots that are required for completing the assigned work. However, this algorithm is only applicable if there are no obstacles in the area of coverage.

Multi-robot coverage using the environment as an implicit communication device has been proposed under the term *Ant Robotics*; see for example the work by Wagner et al. [18]. More recently repartitioning the area was proposed for covering the space by a team of robots [19]. It is worth noting that, the problem of coverage is similar to exploration, especially when considering a much larger-footprint sensor; see for example the work by Quattrini Li et al. [20], [21]. Most of the existing studies rely heavily on the existence of communication between robots, however, in some domains it is infeasible to maintain communications. To overcome this problem in the agricultural domain, Janani et al. [22], [23] proposed a solution that used indirect communication, observing the visual trails of the other robots to guide the coverage. This kind of a solution is impractical in aerial or marine explorations, as any trail disappears very fast in such environments.

Most of the studies presented the decomposed areas as graphs to perform planning, e.g. graph vertices represent the destination points in the area, and the edges are paths connecting them. With this graph representation, the optimal

area coverage problem can be modelled as a modified version of the *travelling salesman problem (TSP)* or the *k-travelling salesman problem* for multiple robots. Meanwhile, when the task is to traverse all the edges of a given graph at least once, then the problem is known as the Chinese Postman Problem (CPP) and k-CPP. While the CPP was proven to have a polynomial-time solution by Edmonds and Johnson [5], others variants are NP-hard [6].

III. STATEMENT OF THE PROBLEM

In the following section the problem of multi-robot area coverage is formally presented together with relevant definitions. Then two approximation algorithms solving the multi-robot area coverage problem are described. Please note that the proposed multi-robot algorithms are based on the single robot efficient area coverage algorithm proposed by Xu et al. [4], which we refer to as *Efficient Complete Coverage algorithm* or *ECC* for brevity.

The multi-robot area coverage problem, with k robots, is the problem of dividing an input area into k non-overlapping sub-areas such that each robot performs the same amount of work while covering its assigned region.

While information exchange between robots may help to avoid overlap [3], [24], in some real-life operations it is impossible to ensure the availability of communication, either because of sensor range or unavailability of a communication network. This work discusses the communication-less, complete, and optimal coverage problem for a multi-robot system. The optimality of the coverage is defined as a MinMax problem: minimizing the maximum coverage cost over a group of robots. The environment that we examine is a 2D bounded rectangular region with obstacles with arbitrary shapes. We assume that there exists prior knowledge about the environment as a map representation. We are also given a single starting point from where all robots must start the coverage and end at that point. In Section III-A we will define some basic terminology used throughout this paper.

A. Terminology

Under the assumption that prior knowledge about the environment exists, the input to the proposed algorithms is a binary image of the coverage area, where obstacles are represented in black and free space in white. The algorithms are also provided with a starting point v_s and the size of the robot team k . For the given inputs both algorithms are producing k non overlapping subregions and the tours that connect from the starting point to each subregion and back.

In both algorithms the ECC algorithm is used: from the image map the area is represented as a graph by applying the boustrophedon cellular decomposition (BCD) algorithm [12] and then the CPP algorithm [5] is applied to find an optimal route over the complete graph or parts of it. The output of the boustrophedon decomposition is an undirected weighted Reeb graph $G_r = (V_r, E_r)$: the set of vertices V_r are points on the corners of obstacles that change the connectivity of the region, and the set of edges E_r represent single non-dividable obstacle free cells C_i [14]. We are assuming that a single cell is the smallest unit and thus if the number of cells is less

than the number of robots k we will have $E_r - k$ idle robots. The CPP algorithm operates in two stages: it first generates an *Eulerian graph* G_r , which is a graph where every vertex has even degree [25]; then it calculates an *Eulerian path*, which is a path that visits every edge of the graph only one time. An *Eulerian tour* is an Eulerian path that starts and ends at the same vertex.

To create an Eulerian graph some edges in the G_r graph must be duplicated. To perform efficient duplication Xu et al. [4] define the edge cost for every edge $e = (v_i, v_j)$ in the E_r as $c_e = \frac{(\text{cell width})^2}{\text{cell area}}$. Additionally, in this paper, two new weights are introduced to differentiate between the actual coverage cost performed by a robot, and the travel cost required for traveling between cells in a subregion. In what follows, the traversal weight w_t of an edge $e = (v_i, v_j)$ is defined as the geodesic distance between the vertices v_i and v_j , for all $1 \leq i, j \leq n$; the coverage weight w_c is defined as the size of the corresponding cell area. The starting point is denoted as $v_s = v_1$. The shortest path between vertices v_i and v_j is defined with respect to both traversal and coverage weights: traversal shortest path $sp_t(v_i, v_j)$ and coverage shortest path $sp_c(v_i, v_j)$. The coverage cost of a subregion is defined as the sum of the coverage costs over all cells contained in the subregion and the traversal cost of the paths that connect the cells.

We next present the two k -coverage algorithms. The first algorithm splits an optimal tour between multiple robots and is called *Coverage with Route Clustering (CRC)*. The second algorithm first divides the area between robots and then starts route planing, and is called *Coverage with Area Clustering (CAC)*.

IV. COVERAGE WITH ROUTE CLUSTERING (CRC)

The first algorithm is a direct extension of the efficient single robot area coverage algorithm (ECC) proposed by Xu et al. [4] for a multi-robot system. The pipeline of the algorithm execution is presented in Fig. 3. As discussed in the previous section, similar to the original work, the boustrophedon cellular decomposition algorithm (BCD) [12] computes a graph representing the free area and encoding the position of obstacles. Because the graph is always connected and all edges are required, the efficient coverage route-finding problem is defined as the MinMax k -Chinese postman problem (k -CPP).

For the k -CPP problem solution, we adapt the k -postman approximation algorithm proposed by Frederikson et al. (FHK) [6]. The following is a sketch of the algorithm:

- 1) The efficient Eulerian tour is obtained by the CPP algorithm [5].
- 2) The most time consuming edge in the Eulerian tour to cover is identified as follows:

$$s_{max} = \max_{2 \leq j \leq m-1} \{s(v_1, v_j) + w_c(v_j, v_{j+1}) + s(v_{j+1}, v_1)\}$$

where $s(v_i, v_j)$ is defined as the shortest path between the vertices v_i and $v_j \forall i, j, 1 \leq i, j \leq n$.

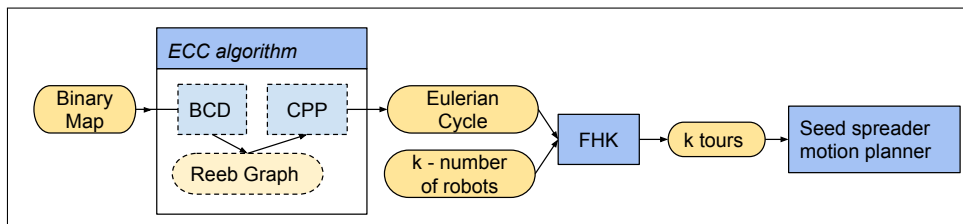


Fig. 3. Coverage with Route Clustering (CRC) execution pipeline

- 3) s_{max} is defined as the least amount of work that each robot is assigned to perform. The remaining coverage area is distributed between robots by a fraction of $\frac{j}{k}$.

The FHK algorithm was used to solve the k -RPP problem by Xu [24]. In their work, there is no differentiation between the coverage and traversal weights of the graph, as the problem domain discussed is different. In contrast to our problem statement, Xu considers road maps, where the coverage and traversal costs can be considered the same. With our problem definition, a modification to the original algorithm is made by introducing the shortest path $sp_t(v_i, v_j)$ to travel between vertex v_i and vertex v_j . The experimental results show that the proposed modification – using the sp_t cost instead of the sp_c coverage path cost in the computation – improves the area coverage distribution between robots and decreases the maximum coverage cost.

Complexity Analysis: The Exact Cellular decomposition of an area by the BCD algorithm can be performed in $O(n \log n)$ time, where n is the total number of resulting vertices in the free area [26]. The CPP algorithm finds an Eulerian cycle in polynomial time [5], while FHK algorithm takes $O(n^3)$ time to divide the Eulerian cycle into k -tours. Thus, the overall complexity of finding k -tours is $O(n^3)$. Although the FHK algorithm might not ensure that all robots are busy, it provides an optimal solution with a $2 - \frac{1}{k}$ approximation factor [6] for the MinMax constraints.

V. COVERAGE WITH AREA CLUSTERING (CAC)

The CAC approach is the following: divide the area into approximately equal partitions and apply the Efficient Complete Coverage (ECC) algorithm [4] for each of the sub-areas. As opposed to the *Cluster First, Route second* algorithm proposed by Xu [24], our algorithm uses a different clustering technique and takes as input an area map instead of a graph. A complete pipeline of the algorithm is presented in Fig. 4.

The proposed algorithm is discussed next, together with the introduction of some additional terminology relevant only to this algorithm.

The *dual graph* of a planar Reeb graph is a weighted graph $G_d = (U_d, E_d)$ with vertex set $U_d = \{u_1, u_2, \dots, u_m\}$, where each vertex is located at the center of a cell of the Reeb graph; two vertices in G_d are connected with an edge if and only if their corresponding cells in the Reeb graph are neighbors (share a critical point). The weight $w_d(u_i, u_j)$ of each edge (u_i, u_j) in E_d is defined as the Euclidean distance between u_i and u_j ; see Fig. 5. As every cell is bounded by two critical points, then we can associate every leftmost

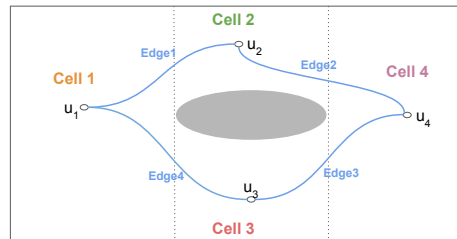


Fig. 5. Dual graph representation based on boustrophedon decomposition critical point with the vertex corresponding to that cell in the dual graph.

Similar to the CRC approach, the input to the algorithm is an area map that we decompose into a set of cells utilizing the BCD algorithm. Along with the Reeb Graph creation, the dual graph $G_d = (U_d, E_d)$ is also built. The dual graph is used to divide the area and allocate it to the k robots using a breadth-first-search-like (BFS) algorithm for clustering. We perform a BFS traversal and include every vertex into a cluster as long as the weight limit of the current cluster is not exceeded. Two vertices u_i and u_j in the dual graph are considered *nearest neighbors* if and only if the Euclidean distance $w_d(u_i, u_j)$ between these vertices has the smallest value for all vertices u_i, u_j , where $1 \leq i, j \leq n$. The *cluster size* of a cluster is the sum of the areas of all the cells in the cluster, plus the shortest travel distance from the starting vertex to the nearest cell in the cluster.

Area clustering: Clustering of the areas is performed following a greedy approach. Each cell in the cluster corresponds to a vertex in the dual graph G_d , thus the clustering becomes a vertex clustering problem in a graph. The clustering schema is the following:

- 1) Sort the vertices/cells in increasing order, first by the y -coordinate and then by the x -coordinate in S_{cell} array. The sorting has to be stable, to ensure that each vertex is processed from bottom to top and from left to right. Set the current vertex u_{curr} to be the first vertex in the sorted list.
- 2) Perform breadth-first search starting from u_{curr} . The order of visiting adjacent vertices is defined by the nearest neighbourhood, i.e. adjacent vertices are processed from nearest to farthest. While traversing each vertex included in the path, the corresponding cell is included in the current cluster as long as the size of the cluster does not exceed a predefined limit.
- 3) The next u_{curr} is the next vertex from S_{cell} that is not included in any cluster, i.e. not marked as visited. Then the process repeats from Step 2.

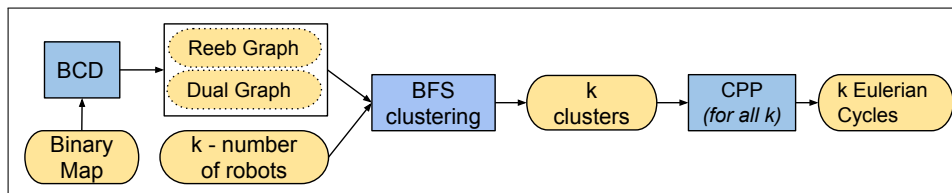


Fig. 4. Coverage with Area Clustering (CAC) execution pipeline

Cluster limit: The limit for each cluster is defined iteratively. The general idea is to compensate the work of the farther-distance travelling robot by assigning the nearest ones more coverage area instead of simply dividing the area into k equal regions. So the procedure for computing the limit is the following:

- 1) Initialize S to be equal to the area size. Sort all the vertices in decreasing order in $ST(u_i)$ according to the shortest distance from the source vertex v_1 to the associated vertex v_j of the Reeb graph.
- 2) For every initial vertex u_h of the cluster h -th, $L = ST(u_h)$. The h -th cluster size must be no greater than $L + \frac{S}{k-h+1}$.
- 3) Extract from S the cluster size. If the number of clusters has not reached k (the number of robots) and S is greater than 0, continue to Step 2.

Complexity analysis: The running time for the cell decomposition stage with the dual graph creation remains $O(n \log n)$ [26]. Area partitioning has a $O(n^2)$ running time, as the greedy approach for cluster expansion goes over every cell only once. Finally, the running time for finding an Eulerian cycle for every cluster is equal to $\sum_{i=1}^k O(n_i) = O(n)$, where n_i is the size of i -th cluster. Thus the overall complexity of the CAC algorithm is $O(n^2)$.

VI. EXPERIMENTAL RESULTS

The experimental validation of the proposed algorithms consists of two components: first a large number of randomly generated Reeb graphs were tested for analysing the performance of the area partitioning stage of proposed CRC and CAC algorithms. Second, multi-robot simulations were run in stage mobile robot simulator [27] for different environments to validate statistical results. In both cases, in addition, to show the advantage of our methods, we compare them with simple equal partitioning and original FHK algorithm with only one edge cost. We will refer to last one as FHK. As for simple equal partitioning, we partition a single optimal route using cluster cost equal to $1/k$ of that optimal route cost, where k is the number of robots. We call this naive route clustering and we will henceforth refer to it as NRC.

A. Statistical Analysis

For statistical analysis we compare the area partitioning phase performance of proposed algorithms, i.e. k -tour splitting FHK algorithm used in CRC algorithm with two different edge costs and DFS-like graph clustering used in CAC. Moreover we show the performance of these algorithms compared to the baseline, i.e. the original FHK algorithm [6] and NRC. Before presenting the actual results, the testing framework is described in detail.

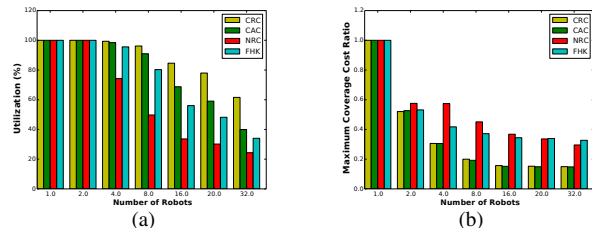


Fig. 6. CRC, CAC utilization and max coverage cost average results compared to NRC and original FHK algorithms for a variable number of robots. (a) Robot utilization, highest number is better. (b) Maximum area covered, lowest number is better.

TABLE I

INPUT GRAPHS INFORMATION

	MIN	MAX	MEAN
Number of Vertices	10	41	25
Number of Edges	12	57	35
Length of Euler Tour	16	73	43
Cost of Euler Tour	2414	11176	5123

Test Data: Testing data represent randomly generated 200 images, with arbitrarily distributed obstacles. The boustrophedon decomposition is performed on each of these artificial environments and as a result a Reeb graph is produced. In the resulting graph the Eulerian path is constructed by applying the algorithm solving the Chinese postman problem. The following input is required for the comparison of the algorithms: a Reeb graph with coverage cost and travel cost assigned to each edge, an Eulerian tour of that graph, and a number of tours the graph is partitioned into. Size information about the input graph and the single coverage tour costs are shown in Table I. The number of robots or tours that the graph will be partitioned into have the following values $k = \{1, 4, 8, 16, 20, 32\}$. All data examples are used with all possible k values. It is worth noting that single cell coverage is considered an atomic action. As such, it is only divided where an edge (cell) has to be duplicated (at most once).

Measurement Metrics: For every call of the algorithm, we measure the number of idle robots and the maximum coverage cost. In Fig. 6 the maximum coverage cost is represented as fraction of single optimal coverage path, and utilization as the percentage of coverer robots. We present average results over all input data for each number of robots Fig. 6. In addition to measure overall effectiveness we averaged results over all k robots, when $k > 1$ II.

Results: The area coverage problem is defined in this work as a MinMax problem. The maximum coverage cost is minimized by ensuring that no robot stays idle. However, even with this definition there is a possibility to have idle robots when the size of cells are not balanced.

All algorithms were applied on the datasets which were

TABLE II
AVERAGE RESULTS FOR $k=\{2, 4, 8, 16, 20, 32\}$ ROBOTS

	Utilization (%)	Maximum Coverage Cost ratio
CRC	86.6	0.248
CAC	76.2	0.245
NRC	52.0	0.433
FHK	69.0	0.388

generated as described at the beginning of this section. Based on the presented results the CRC and CAC algorithms show similar performance on solving MinMax problem and on utilizing the robots; see Fig 6. But on average for larger number of robots CRC compared to CAC shows 10% better utilization. Meanwhile both CRC and CAC algorithms outperform NCR and FHK. In average CRC and CAC outperform NRC and FHK algorithms by maximizing the utilization by 20.5% and minimizing the coverage cost by 39.8%; see Table II.

B. Experimental Validation in different environments

For testing complete pipeline of CRC and CAC algorithms different environments were used. In Fig. 7 the top row presents a variant of the well known cave environment from stage [27]; see Fig. 7(a). A complex environment with many obstacles is presented in Fig. 7(b). An indoor environment is shown in Fig. 7(c), and the large environment from rural Quebec from the work of Xu et al. [4] is presented in Fig. 7(d). The second row of Fig. 7 presents the coverage path for four robots performing the CRC coverage algorithm, while the third row of Fig. 7 presents the respective coverage paths utilizing the CAC algorithm. Each of the robots utilizes the Boustrophedon coverage pattern proposed by Acar et al. [13], [14]. As can be seen from the different results in the different environments, the distribution of areas among the robots vastly varies. To a large extent, the fact that each cell represents an atomic coverage action it is responsible for the uneven distribution of tasks.

To validate statistical results same four environments were also used to test the performance of the proposed algorithms in comparison with the NRC, FHK algorithms for different number of robots. Fig. 8 presents the utilization and maximum coverage cost, calculated the same way as it was described in the previous section, for covering each of the environments presented in Fig. 7 (a)-(d). Different number of robots ($\{1,2,4,8,16,20,32\}$) were used. The results once again show that both CRC and CAC in comparison with naive clustering (NRC) and FHK provide better minimization and utilization, even though they have the same convergence for minimizing the maximum coverage cost. It is worth noting that the scale of each environment is arbitrary and the distances measured only serve as relative measurements. What is significant, however, is the comparison among different number of robots in a single environment.

VII. DISCUSSION AND CONCLUSION

Two solutions to the multi-robot area coverage problem were presented. The optimal coverage in this paper is formulated as a MinMax optimization problem. The ability to

communicate between robots was excluded in the formulation of our problem to ensure that the problem is applicable to environments with minimal or no communication.

In the first CRC algorithm, the ECC algorithm was used to calculate single optimal tour that covers entire area of free space. On the resulting tour, a modification of the k -tour splitting FHK algorithm—termed CRC coverage algorithm—is applied. We show that defining the longest distance in the FHK algorithm, by taking into account the travel cost of edges instead of just the coverage cost, improves the original algorithm. The running time of the tour splitting stage is $O(V^3)$, with an approximation factor of $2 - \frac{1}{k}$ [6]. A second greedy approximation algorithm CAC with $O(V^2)$ complexity is also presented. It first splits the free space in smaller areas and then performs efficient single robot coverage on each of the cells.

Presented results for the CRC and CAC algorithms illustrate the improvement in robot utilization and minimization of max coverage cost. Even though the proposed algorithms show good performance, they still lack high utilization for a larger number of robots. The latter is due to the unbalanced cell sizes. Thus our future work will study new ways of partitioning the area to achieve better utilization for a larger number of robots. Furthermore, we are currently constructing a fleet of six Autonomous Surface Vehicles (ASVs). The proposed algorithms will be deployed on the ASVs to perform bathymetric studies in the nearby Lake Murray SC, USA.

ACKNOWLEDGMENT

The authors would like to thank the generous support of the Google Faculty Research Award and the National Science Foundation grants (NSF 1513203, 1637876).

REFERENCES

- [1] H. Choset, "Coverage for robotics a survey of recent results," *Annals of Mathematics and AI*, vol. 31(1-4), pp. 113–126, 2001.
- [2] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61(12), pp. 1258 – 1276, 2013.
- [3] I. Rekleitis, A. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and AI*, vol. 52(2-4), pp. 109–142, 2008.
- [4] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36(4), pp. 365–381, 2014.
- [5] J. Edmonds and E. L. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming*, vol. 5, pp. 88–124, 1973.
- [6] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in *Proc. of the 17th Annual Symposium on Foundations of Computer Science (SFCS)*. IEEE Computer Society, 1976, pp. 216–227.
- [7] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Mathematics and AI*, vol. 31(1-4), pp. 77–98, 2001.
- [8] N. Hazon and G. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 735–741.
- [9] N. Agmon et al., "The giving tree: constructing trees for efficient offline and online multi-robot coverage," *Annals of Mathematics and AI*, vol. 52(2-4), pp. 143–168, 2008.
- [10] P. Fazli, A. Davoodi, P. Pasquier, and A. Mackworth, "Complete and robust cooperative robot area coverage with limited range," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 5577–5582.

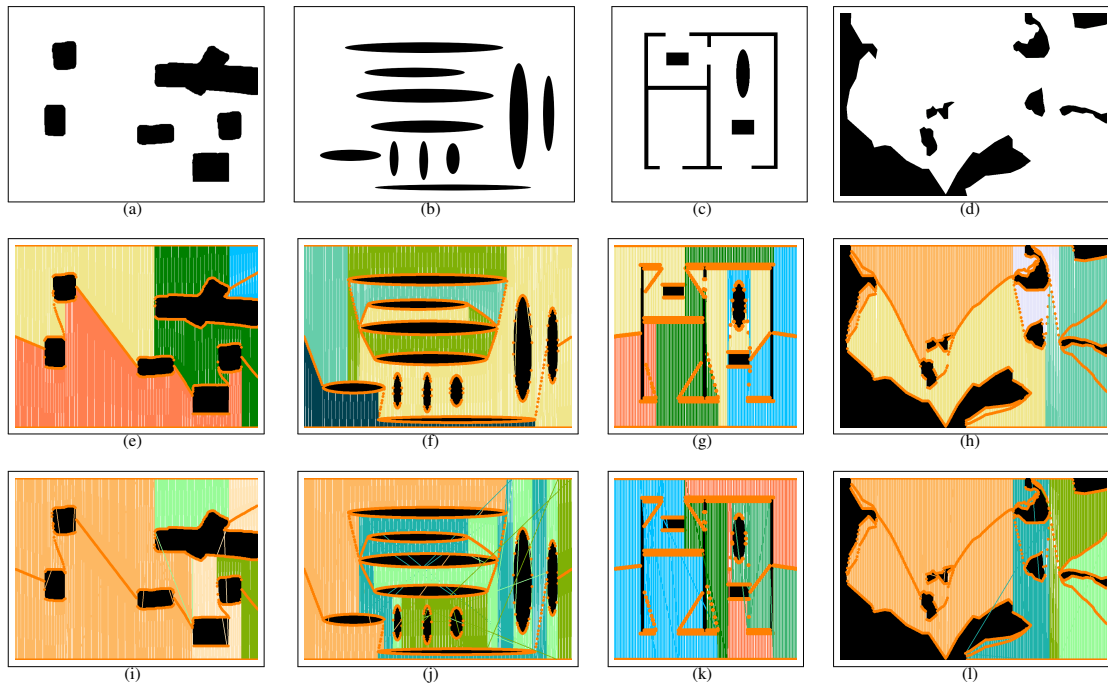


Fig. 7. The four environment where the proposed multi-robot coverage algorithms were tested. (a) Cave; (b) Multi-cell; (c) Indoor; (d) Rural Quebec. (e-h) Coverage paths for four robots utilizing the CRC coverage algorithm. (i-l) Coverage paths for four robots utilizing the CAC coverage algorithm.

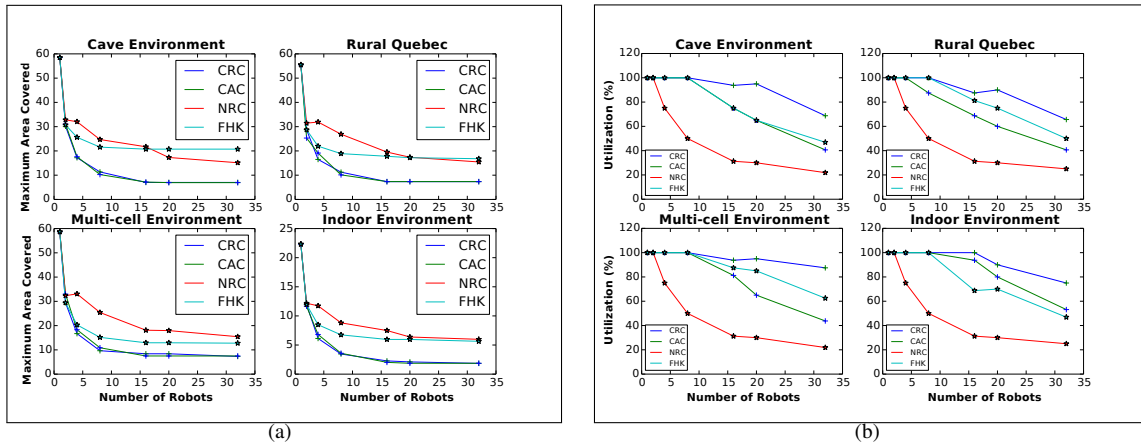


Fig. 8. The effectiveness of the multi-robot coverage algorithms for different environments and for different number of robots.

- [11] A. Renzaglia, L. Doitsidis, S. A. Chatzichristofis, A. Martinelli, and E. B. Kosmatopoulos, "Distributed multi-robot coverage using micro aerial vehicles," in *Proc. Med. Conf. on Control Automation (MED)*, June 2013, pp. 963–968.
- [12] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9(3), pp. 247–253, 2000.
- [13] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse Decompositions for Coverage Tasks," *The Int. J. of Robotics Research*, vol. 21(4), pp. 331–344, 2002.
- [14] E. U. Acar and H. Choset, "Sensor-based Coverage of Unknown Environments: Incremental Construction of Morse Decompositions," *The Int. J. of Robotics Research*, vol. 21(4), pp. 345–366, 2002.
- [15] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 5525–5530.
- [16] G. S. C. Avellar, G. A. S. Pereira, L. C. A. Pimenta, and P. Iscold, "Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time," *Sensors*, vol. 15(11), p. 27783, 2015.
- [17] P. Toth and D. Vigo, Eds., *The Vehicle Routing Problem*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.
- [18] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Trans. on Robotics and Automation*, vol. 15(5), pp. 918–933, 1999.
- [19] K. Hungerford, P. Dasgupta, and K. Guruprasad, "A repartitioning algorithm to guarantee complete, non-overlapping planar coverage with multiple robots," in *Distributed Autonomous Robotic Systems*. Springer, 2016, pp. 33–48.
- [20] A. Quattrini Li, "Study, design, and evaluation of exploration strategies for autonomous mobile robots," Ph.D. dissertation, Poli. Milano, 2015.
- [21] A. Quattrini Li, R. Cipolleschi, M. Giusto, and F. Amigoni, "A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings," *Autonomous Robots*, vol. 40(4), pp. 581–597, 2016.
- [22] A. Janani, L. Alboul, and J. Penders, "Multi-agent cooperative area coverage: Case study ploughing," in *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems*, 2016, pp. 1397–1398.
- [23] —, *Multi Robot Cooperative Area Coverage, Case Study: Spraying*. Sheffield, UK Cham: Springer Int. Publishing, Jun. 2016, pp. 165–176.
- [24] L. Xu, "Graph planning for environmental coverage," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, USA, Aug. 2011.
- [25] N. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory, 1736-1936*. New York, NY, USA: Clarendon Press, 1986.
- [26] N. H. Sleumer and N. Tschichold-Gärman, "Exact cell decomposition of arrangements used for path planning in robotics," 1999.
- [27] Brian Gerkey et al., "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," in *Proc. of the Int. Conf. on Advanced Robotics*, 2003, pp. 317–323.