

Multi-Robot Repeated Area Coverage: Performance Optimization Under Various Visual Ranges

Pooyan Fazli
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
pooyanf@cs.ubc.ca

Alireza Davoodi
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
davoodi@cs.ubc.ca

Alan K. Mackworth
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
mack@cs.ubc.ca

Abstract—We address the problem of repeated coverage of a target area, of any polygonal shape, by a team of robots having a limited visual range. Three distributed *Cluster-based* algorithms, and a method called *Cyclic Coverage* are introduced for the problem. The goal is to evaluate the performance of the repeated coverage algorithms under the effect of changes in the robots' visual range. A comprehensive set of performance metrics are considered, including the distance the robots travel, the frequency of visiting points in the target area, and the degree of balance in workload distribution among the robots. The *Cyclic Coverage* approach, used as a benchmark to compare the algorithms, produces optimal or near-optimal solutions for the single robot case under some criteria. The results show that the identity of the optimal repeated coverage algorithm depends on the metric and the robots' visual range.

Keywords—Multi-Robot Systems; Teamwork; Coordination; Repeated Area Coverage

I. INTRODUCTION

Distributed area coverage as a task for multi-robot systems is a challenging problem in different scenarios such as search and rescue operations, planetary exploration, intruder detection, environment monitoring, floor cleaning and so on. In this task, a team of robots cooperatively visits (observes or sweeps) an entire area, possibly obstructed by obstacles. The goal is to build efficient paths for all the robots which jointly ensure that every point in the environment is visited by at least one of the robots. If there is a need to detect some events in the environment, area coverage guarantees finding all of them in the target area.

There are two classes of coverage problems:

- **Single Coverage:** The aim is to cover the target area until all the accessible points of interest in the environment have been visited at least once, while minimizing the time, distance traversed by the robots, and the number of visits to the points [1], [2], [3], [4].
- **Repeated Coverage:** The goal is to cover all the accessible points of interest in the environment repeatedly over time, while maximizing the frequency of visiting points in the target area, minimizing the sum/maximum length of the paths/tours generated for the robots, balancing the workload distribution among the robots, or maximizing the reward by detecting the maximum number of events, weighted by their importance, in minimum time. Visiting the points in the area can be performed with uniform or non-uniform frequency, depending on the priorities of different parts of the area. In this paper, we use the terms 'coverage' and 'repeated coverage' interchangeably.

Several research communities including robotics/agents [5], [6], sensor networks [7], operations research [8] and computational geometry [9] work on variants of the repeated coverage problem.

In operations research, the *Vehicle Routing Problem* has some similarity to the repeated coverage scenarios [10]. In computational geometry, this problem originates from the *Art Gallery Problem* [11] and its variant for mobile guards, the *Watchman Route Problem* [12]. In the *Art Gallery Problem*, the goal is to find a minimum number of static guards (control points) which can jointly cover a workspace under different restrictions. On the other hand, in the *Watchman Route Problem* the objective is to compute routes (closed curves) watchmen should take to guard an entire area given only the map of the environment. Most research done on the above problem definitions in computational geometry deal with simple polygonal spaces without obstacles, unlimited range of agents' vision, single agent scenarios, and scenarios in which a common initial location is determined for all the agents. *Pursuit-Evasion* is another closely related problem studied in both the computational geometry and the robotics communities. In this task, one or more searchers move throughout a given target area in order to guarantee the detection of all the evaders, which can move arbitrarily fast [13]. In *Pursuit-Evasion* scenarios the searchers do not necessarily cover the entire target area.

In the robotics community, most research in this area is carried out under the rubric of *Area Patrolling*. Elmaliach *et al.* [6] proposed a centralized algorithm which guarantees optimal uniform frequency, *i.e.*, all cells are visited with maximal and uniform frequency in a non-uniform, grid environment. Grid-based representations have limitations since they do not consider the structure of the environment and as a result are unable to handle partially occluded cells or cover areas close to the boundaries in continuous spaces. Our algorithms, on the other hand, guarantee complete coverage of the area. Another problem with some of the existing empirical studies in the field of area patrolling is the lack of a comprehensive population of environment maps in the experiments. In the works by Almeida *et al.* [14] and Machado *et al.* [5] only six maps were used to evaluate the coverage algorithms, two of which have almost 75% similarity. In two other maps called 'circular' and 'corridor', only one representation of the environment (*i.e.* a chain) is possible due to the structure of the environments. Moreover, the authors did not provide details on how the graph is built to represent the environment. They typically presume the existence of a graph which is not a complete model of the environment, just a rough approximation of it. The proposed architectures also consider the agents as points with no extent or limit on visual range, so the problem dealt with is reduced to a graph exploration/coverage task rather than an area coverage scenario. Our paper, on the other hand, builds a complete model of the environment considering the limited visual range of the robots. Moreover, we will investigate the effect of varying the robots' visual range on the performance of the repeated coverage algorithms with extensive experiments using different

metrics including the distance the robots travel, the frequency of visiting points in the target area, and the degree of balance in workload distribution among the robots.

The contributions of this paper are as follows:

- 1) We present an approach to modeling the environment using graph-based methods considering the limited visual range of the robots. To this end, an environment modeling approach is developed based on the *Visibility Graph*.
- 2) Three *Cluster-based* algorithms are introduced for the distributed repeated coverage problem, differing as to how they cluster the graph, namely: the *Uninformed Clustering Coverage*, the *Edge-based Clustering Coverage*, and the *Node-based Clustering Coverage* algorithms.
- 3) An algorithm called *Cyclic Coverage* is introduced and used as a benchmark to compare the performance of the repeated coverage algorithms. The algorithm finds the shortest tour on the graph similar to solving a *Travelling Salesman Problem (TSP)*. We are going to show that even though the *Cyclic Coverage* approach can produce *optimal* or *near-optimal* solutions for a *single robot* case under some particular metrics; however, it is not always the best solution when extending the problem to multi-robot scenarios.
- 4) The effect of varying the robots' visual range on the performance of the repeated coverage algorithms is investigated.

The results show that optimizing each of the metrics depends on the choice of the repeated coverage algorithm, and the robots' visual range.

II. PROBLEM DEFINITION AND PRELIMINARIES

The problem is to cover the environment repeatedly over time using an arbitrary number of robots. To this end, we make the following assumptions.

Assumption 1. *The environment boundary is a known 2D simple polygon containing static polygonal obstacles.*

Assumption 2. *The robots are presumed to have a 360° field of view and a predefined circular limit of visual range.*

Assumption 3. *The robots are homogeneous, with the same speed, and can move in any direction.*

In order to evaluate the coverage mission, some metric criteria need to be determined, but before that we introduce some basic definitions:

- **Full Single Coverage:** all the robots traverse the paths assigned to them just once.
- **Visiting Period (VP):** the time interval between two visits to a point of interest in the target area. A point of interest can have more than one *Visiting Period*, due to the possibility of the point being visited more than once in different time intervals by more than one robot in a *Full Single Coverage*.
- **Average Visiting Period (AVP):** the average of the *Visiting Periods* of a point of interest.
- **Worst Visiting Period (WVP):** the maximum period of time it takes a point of interest to be re-visited in the target area.
- **Visiting Frequency (VF):** the number of visits to a point of interest by a single robot in a *Full Single Coverage*. In case a point of interest is visited by more than one robot in a *Full Single Coverage*, the point will have more than one *Visiting Frequency*, each of which associated with one robot.

The repeated coverage algorithms will be evaluated based on the following metrics:

- **Total Path Lengths (TPL):** the sum of the lengths of the paths assigned to the robots in order to have a *Full Single Coverage*.
- **Total Average Visiting Period (TAVP):** the average of the *Average Visiting Periods* of all the points of interest in the target area.
- **Total Worst Visiting Period (TWVP):** the maximum *Worst Visiting Period* of all the points of interests in the target area.
- **Balance in Workload Distribution (BWD):** the degree of balance in the workload distribution among a team of robots. A workload distribution is completely *balanced* if the standard deviation of the lengths of the constructed paths for the robots is zero: the paths assigned to the robots all have equal lengths.

In this study, the aim is to minimize *TPL*, *TAVP*, and *TWVP* and to maximize *BWD* in the repeated coverage scenario. Interestingly, it is impossible to develop even approximate polynomial time algorithms, when optimizing each of the metrics for the repeated coverage problem, unless $P = NP$ [12]. Furthermore, optimizing all these metrics simultaneously is another challenge, because some are mutually conflicting in the coverage mission. These considerations require us conduct an extensive experimental analysis to evaluate the performance of the algorithms.

The sequential stages of the proposed repeated coverage algorithms are as follows:

- 1) The locations of static *guards* required to cover visually a given 2D environment are determined, allowing for the limited range of the robots' vision (Section III).
- 2) A graph is built on the guards and the obstacles based on the *Visibility Graph* (Section IV-A).
- 3) The graph is reduced to *Reduced-Vis* representation (Section IV-B).
- 4) *Cluster-based Coverage Algorithms:* The *Reduced-Vis* is partitioned into as many clusters as the number of robots. To this end, three different clustering algorithms are introduced, namely: *Uninformed Clustering*, *Edge-based Clustering*, and *Node-based Clustering*. Finally, a tour is built for each robot on the clustered reduced graph. For this purpose, two tour building algorithms are proposed, namely: *Double-Minimum Spanning Tree*, and the *Chained Lin-Kernighan* algorithms (Section V).
- 5) *Cyclic Coverage Algorithm:* Rather than partitioning the *Reduced-Vis* among the robots, the *Cyclic Coverage* algorithm creates a tour on the whole reduced graph, and then distributes the robots equidistantly around it (Section VI).

Since the problem is a repeated coverage scenario, we can ignore the initial cost of moving the robots from their initial locations to their assigned paths in the target area, as that time is negligible compared to the recurring patrol time.

In the following sections, we will explain the different stages of the proposed algorithms for repeated coverage of a target area in detail.

III. LOCATING GUARDS WITH LIMITED VISUAL RANGE

In our problem definition, we presume the robots are equipped with panoramic cameras with a 360° field of view. However, the cameras' visual range is limited. The proposed approach initially locates a set of *guards (Points of Interests)* required to visually cover an entire area. The term *guard* is

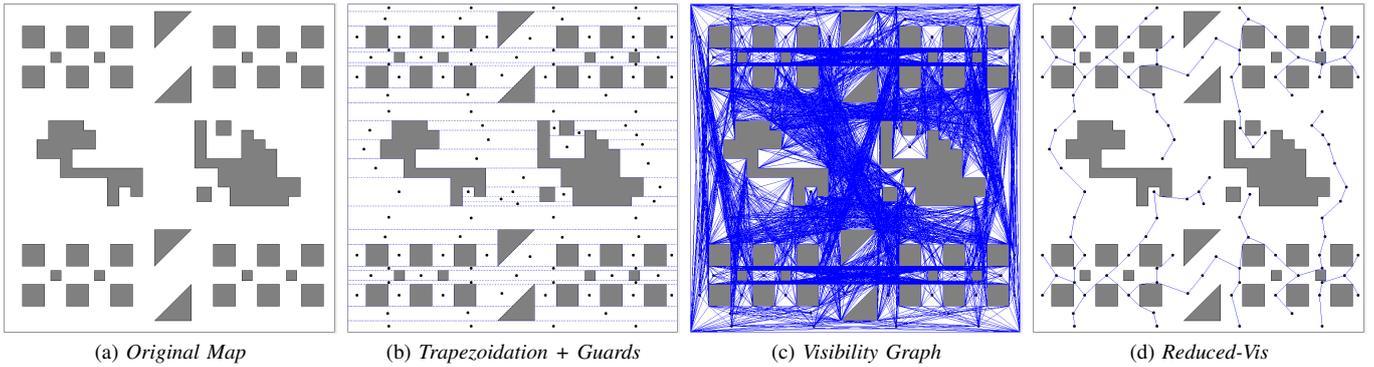


Figure 1: Locating Guards in a Sample Environment

taken from the *Art Gallery Problem* [11]. These static *guards* are control points that can jointly cover the whole environment while satisfying the limited visual range constraint of the robots. In other words, if there are as many robots as there are *guards*, and each robot were stationed on a *guard*, the entire area would be covered visually by the robots.

To locate the *guards*, the algorithm decomposes the initial target area, a 2D simple polygon with static obstacles, into a collection of convex polygons using a *Trapezoidal Decomposition* method, and then applies a post-processing approach to eliminate as many trapezoids as possible [15]. The post-processing step is more effective in cluttered areas, and since the number of *guards* located by the algorithm is directly correlated to the number of trapezoids, fewer trapezoids will result in fewer *guards*.

At the next step, a divide-and-conquer method [16] is used to successively subdivide each of the resulting convex polygons (trapezoids) into smaller convex sub-polygons until each of them can be covered visually by one *guard*.

Figure 1b shows the computed guards after trapezoidation on the sample environment of Figure 1a.

IV. BUILDING THE GRAPH

Having located the static guards in the previous step, the *Visibility Graph* is then built on the obstacles' nodes (P) and the computed guards (SG).

A. Visibility Graph

The *Visibility Graph* (VG) is a graph structure used in computational geometry and robot motion planning [17]. In our VG , the computed guards and the corners of the area in the Euclidean plane comprise the nodes of the *Visibility Graph*. If two nodes are mutually visible, they are connected using an edge in the graph. Two nodes of the environment are mutually visible if the line segment joining them does not intersect any obstacle. Figure 1c shows the *Visibility Graph* built on the sample environment.

B. Graph Reduction

The aim of graph reduction is to improve efficiency by minimizing the average or total time taken for the robots to traverse the graph. The input of the algorithm is the VG discussed in Section IV-A, and the output is a graph called *Reduced-Vis*.

The method starts by using the *Floyd-Warshall* algorithm to find the set $MD = \{(c_{ij}, v_i, v_j) | v_i, v_j \in V_{vis}\}$ of minimum distances, c_{ij} , and the set $SP = \{(r_{ij}, v_i, v_j) | v_i, v_j \in V_{vis}\}$ of shortest paths, r_{ij} , between any pair of nodes v_i and v_j of the input graph.

The minimum value of all the minimum distances in MD is then selected provided that both the endpoints of the corresponding shortest path in SP belong to the set of static guards, SG , computed in section III. The chosen path, including all its nodes and edges, forms the initial component of the *Reduced-Vis*.

Next, among all the guards that have not yet been added to the graph, the algorithm finds the closest static guard to the current component, merging the corresponding shortest path with it. Following the same process, the algorithm keeps expanding the *Reduced-Vis* until there are no more guards to be added to the graph. The resultant graph is the final *Reduced-Vis*. The nodes of the reduced graph includes all the guards (SG), as the *Points of Interests* in the target area and the subset of the obstacles' nodes ($P \subset P$). Traversing the *Reduced-Vis* guarantees complete coverage of the target area given the limited visual range of the robots.

Figure 1d illustrates the *Reduced-Vis* computed on the VG of Figure 1c.

V. CLUSTER-BASED COVERAGE ALGORITHMS

Cluster-based coverage algorithms decompose the *Reduced-Vis* into $|R|$ (number of robots) clusters, $T = \{T_1, T_2, \dots, T_{|R|}\}$, such that $\bigcup_{i=1}^{|R|} V_{T_i} = SG$; SG is the set of guards for the reduced graph and V_{T_i} is the set of guards of the cluster T_i . Below, three different *Cluster-based* coverage algorithms are presented. Having built the clusters on the *Reduced-Vis*, a tour is built on the generated cluster for each robot. The tour building algorithms are discussed in Section V-D.

A. Uninformed Clustering Coverage

The *Uninformed Clustering Coverage* algorithm partitions the *Reduced-Vis* into $|R|$ clusters by removing the $|R| - 1$ longest edges of the graph. Thereafter, a tour is built on each cluster generated for the robots. In this algorithm, the evaluation metrics are defined as below:

$$TPL = \sum_{i=1}^{|R|} \text{Length}(\text{Tour}(T_i)). \quad (1)$$

$\text{Tour}(T_i)$ is the tour built on cluster T_i , and $\text{Length}(\text{Tour}(T_i))$ is the length of the tour.

$$TAVP = \frac{\sum_{node \in SG} AVP(node)}{|SG|}, SG = \text{Static Guards}, \quad (2)$$

where $AVP(node) = \frac{Length(Tour(T_i))}{VF_i(node)}$, $i \in \{1, 2, \dots, |R|\}$, and $node \in T_i$. Also, $VF_i(node)$ is the *node Visiting Frequency* in tour T_i .

$$TWVP = \max_{i=1,2,\dots,|R|} (Length(Tour(T_i))). \quad (3)$$

Balance in Workload Distribution is computed after building the tour:

$$BWD(Tour(T)) = \left(1 - \frac{STD(\{Length(Tour(T_i)) | i = 1, 2, \dots, |R|\})}{STD(\{TPL, \alpha_1, \alpha_2, \dots, \alpha_{|R|-1} | \alpha_i = 0\})}\right) \times 100, \quad (4)$$

where $STD(\cdot)$ is the population standard deviation, and $T = \{T_1, T_2, \dots, T_{|R|}\}$ is the $|R|$ clusters created for the $|R|$ robots. For the case of one robot, we assume that $BWD(Tour(T)) = 100$. In BWD 's computation, $STD(\{Length(Tour(T_i)) | i = 1, 2, \dots, |R|\})$ is the population standard deviation of the set of tours created for the robots, and $STD(\{TPL, \alpha_1, \alpha_2, \dots, \alpha_{|R|-1} | \alpha_i = 0\})$ is the worst case scenario, in which one robot is in charge of the whole task, *i.e.* TPL , and the other robots are idle with zero path length $(\alpha_1, \alpha_2, \dots, \alpha_{|R|-1})$.

Figure 2 illustrates the tours built for four robots (shown by black, blue, green, and red circles) on the sample environment by the *Uninformed Clustering Coverage* algorithm.

B. Edge-based Clustering Coverage

The *Edge-based Clustering Coverage* approach builds a set of $|R|$ clusters on the reduced graph. The clusters are initiated as follows: the endpoints of the longest path in the original VG graph are selected as the starting points of the first two clusters. For the next cluster, a node in SG is selected such that it maximizes the minimum distance from the starting points of the first two clusters. Similarly, for the next cluster, a node is selected that maximizes the minimum distance from the starting points of the other three clusters. This continues until $|R|$ initial nodes are found for the $|R|$ clusters of the robots.

Starting from the initial nodes, clusters are expanded sequentially, by choosing one guard at a time, until all the guards of the reduced graph have been visited at least once. The guards are visited in a way that satisfies the following constraints:

- Find the nearest immediate (that is, ignoring the corners of the obstacles) guard, add it and the corresponding edge/path to the cluster provided that it does not create a cycle. In case of a tie, choose the guard which maximizes the sum of the distances from the guards most recently selected by the other clusters.
- Do not add a guard which has already been chosen by any other cluster, unless there is no other unvisited immediate guard.
- *Remove Common Nodes*: When all the guards of the graph are selected by at least one cluster, remove as many as possible of the common nodes shared by the clusters of the robots. To this end, starting from the most recently selected guard, discard the guards and their corresponding edges/paths from the cluster if they have been selected sooner by any other cluster. Continue this process until it reaches the most recently selected guard which has not been selected sooner by other clusters.

Finally, a tour is built on the generated cluster for each robot. In the *Edge-based Clustering Coverage* approach, the evaluation metrics are defined as below:

$$TAVP = \frac{\sum_{node \in SG} AVP(node)}{|SG|}, SG = \text{Static Guards}, \quad (5)$$

where $AVP(node) = \frac{1}{\left\{\sum_i \frac{VF_i(node)}{Length(Tour(T_i))}\right\}}$, $i \in \{1, 2, \dots, |R|\}$, and $node \in T_i$. Also, $VF_i(node)$ is the *node Visiting Frequency* in tour T_i .

The computation of the AVP here is different from that in the *Uninformed Clustering Coverage*, because in the *Edge-based Clustering Coverage*, a node can belong to more than one cluster.

$$TWVVP = \max_{node \in SG} (WVP(node)), \quad (6)$$

where $WVP(node) = \min_{i=1,2,\dots,|R|} (\max\{VP_i(node)\})$, $node \in T_i$.

In computing the WVP of a node, we first calculate the maximum *Visiting Period* of the node in each cluster, and then choose the minimum value among those maximum *Visiting Periods*. Recall that each node can belong to more than one cluster, and can also have more than one *Visiting Period* in each cluster.

The computations of TPL , and $BWD(Tour(T))$ are the same as given for the *Uninformed Clustering Coverage* approach.

Figure 3 illustrates the tours built for four robots (shown by black, blue, green, and red circles) on the sample environment by the *Edge-based Clustering Coverage* algorithm.

C. Node-based Clustering Coverage

The *Node-based Clustering Coverage* approach uses the *k-Means* clustering algorithm to divide the guards into $|R|$ disjoint clusters. In the first iteration, the initial centroids are found in the same way as finding the starting points of the clusters discussed in the *Edge-based Clustering Coverage* algorithm. In the next iterations, since the computed centroids may not lie on the nodes of the reduced graph, they are matched to the closest guard in the environment. Distance from the centroids is determined based on the distance in the original VG graph rather than the Euclidean distance.

Having built the $|R|$ clusters on the guards, we connect each pair of guards in each cluster if they have a corresponding edge in the *Reduced-Vis*. Thereafter, we do a connectivity test on all the clusters, meaning that every pair of guards in each cluster should be connected through a path. For this purpose, we first find the disconnected components within the cluster and then compute the *Minimum Spanning Tree* on them based on the edges of the original VG graph. We add the *Minimum Spanning Tree*'s corresponding edges and nodes to the cluster, and finally a tour is built on the cluster. The tour is then assigned to a robot, and the robot repeatedly traverses the tour.

In the *Node-based Clustering Coverage* approach the computations for $TAVP$, $TWVP$, TPL , and $BWD(Tour(P))$ are the same as given for the *Edge-based Clustering Coverage* approach.

Figure 4 illustrates the tours built for four robots (shown by black, blue, green, and red circles) on the sample environment by the *Node-based Clustering Coverage* algorithm.

D. Building the Tour

Having built the clusters on the *Reduced-Vis*, we use two algorithms to build the tours on the clusters:

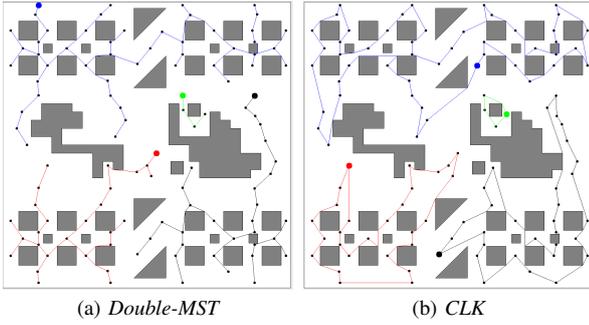


Figure 2: Uninformed Clustering Coverage

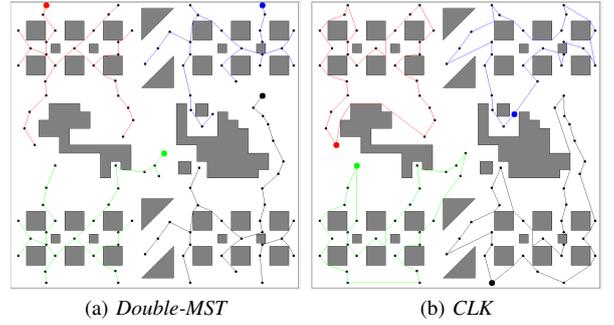


Figure 3: Edge-based Clustering Coverage

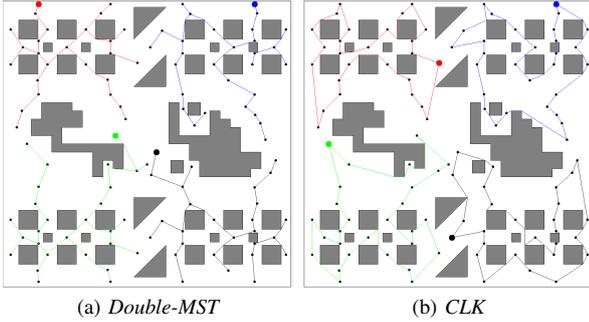


Figure 4: Node-based Clustering Coverage

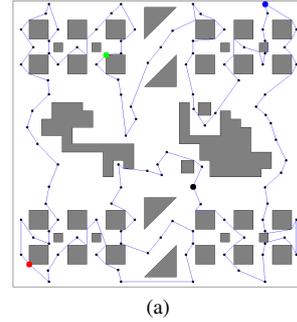


Figure 5: Cyclic Coverage

1) *Double-Minimum Spanning Tree (Double-MST)*: *Double-Minimum Spanning Tree* takes a cluster as an input and returns a cycle whose length is twice the length of the cluster. In this algorithm, every edge of the cluster is visited twice.

2) *Chained Lin-Kernighan (CLK)*: *Chained Lin-Kernighan*, a modification of the *Lin-Kernighan* algorithm, is generally considered to be one of the best heuristic methods for generating *optimal* or *near-optimal* solutions for the *Euclidean Traveling Salesman Problem* [18]. Given the distance between each pair of a finite number of nodes in a *complete graph*, the *Travelling Salesman Problem (TSP)* is to find the shortest tour passing through all the nodes exactly once and returning to the starting node.

The input of the *Chained Lin-Kernighan* algorithm needs to be a *complete graph*. To this end, the *Reduced-Vis* is made complete by adding edges from the original *VG* graph when there does not exist an edge between two nodes in the reduced graph. If there is no edge between the two nodes in the original graph either, a *virtual edge* is added to the reduced graph to connect the two nodes. The *virtual edge* is the shortest path between the two nodes in the *Visibility Graph*. The *Chained Lin-Kernighan* algorithm then finds the shortest tour passing through all the nodes of the *Reduced-Vis*.

VI. CYCLIC ALGORITHM

We develop a method called *Cyclic Coverage* and use it as a benchmark to compare the repeated coverage algorithms. Similar to the *Cluster-based* coverage algorithms, the *Cyclic Coverage* approach locates the guards, builds the graph (*VG*), and then reduces the graph (*Reduced-Vis*). However, rather than partitioning the reduced graph among the robots, it creates a tour on the whole *Reduced-Vis* using the *Chained Lin-Kernighan* algorithm. The proposed algorithm then distributes the robots equidistantly around the tour and moves them repeatedly around it. The *Cyclic Coverage* approach produces *optimal* or *near-optimal* solutions for the *single robot* case under *Total Path Length* and the *Total Worst Visiting Period*.

In this algorithm, the computations for *TAVP*, *TWVP*, *TPL* are the same as given for the *Edge-based Clustering Coverage* and the *Node-based Clustering Coverage* approaches. The notion of *Balance in Workload Distribution* is not defined in this approach, since all the robots traverse the whole tour built on the reduced graph.

Figure 5 illustrates the tours built for four robots (shown by black, blue, green, and red circles) on the sample environment by the *Cyclic Coverage* algorithm.

VII. EVALUATION AND EXPERIMENTS

We have developed a simulator to test the algorithms in different scenarios. The simulator can support different numbers of robots in the target area, different visual ranges for the robots, and varying degrees of clutter in the environment. A random map generator was also developed as a part of the simulator which extends a library [19] to build rectilinear or non-rectilinear polygons with free form polygonal obstacles within the space. Maps can have different numbers of nodes and percentages of clutter. The simulator will be made freely available online.

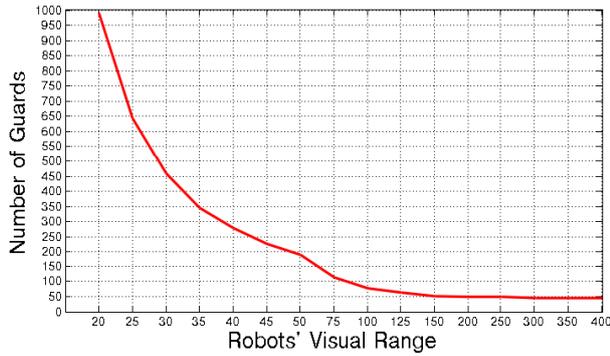
The goal of the experiments is to evaluate the performance of the four repeated coverage algorithms:

- *Uninformed Clustering Coverage (UCC)*
- *Edge-based Clustering Coverage (ECC)*
- *Node-based Clustering Coverage (NCC)*
- *Cyclic Coverage (CC)*

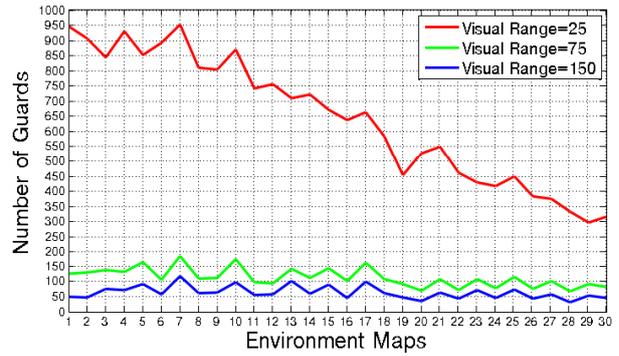
under the effect of changes in the *Visual Range of the Robots*. The performance of the coverage algorithms is evaluated based on these criteria:

- *Total Path Length (TPL)*
- *Total Average Visiting Period (TAVP)*
- *Total Worst Visiting Period (TWVP)*
- *Balance in Workload Distribution (BWD)*

We consider three types of environments in the experiments: sparse (0 – 25% cluttered), semi-cluttered (25 – 50% cluttered),



(a) Average Number of Guards Computed on the Selected Maps as a Function of Robots' Visual Range



(b) Number of Guards Computed on Each Selected Map as a Function of the Chosen Visual Ranges

Figure 6: Number of Guards vs. the Robots' Visual Range

cluttered (50 – 75% cluttered). Ten different maps are used in the experiments for each of the three environment types (30 in total). The clutter percentage of an environment is the ratio of the area of the obstacles to the whole target area (*i.e.* obstacles + free space). The size of the environments is $15m \times 15m$.

Furthermore, three levels of visual range (*low, medium, and high*) are determined in order to investigate the impact of the visual range on the performance of the coverage algorithms. Figure 6a shows the average number of guards computed on the 30 maps used in the experiments under visual ranges up to 400cm. As shown in the figure, the number of computed guards declines with the increase in the visual range of the robots, and after visual range of 150cm, the number of guards becomes essentially fixed. Based on this distribution and some initial experiments to find the proper interval between the visual ranges in order to show the difference of the results more effectively, we chose visual ranges 25cm (*low*), 75cm (*medium*) and 150cm (*high*) to evaluate the algorithms. Figure 6b shows the number of guards computed on each of the selected maps (sorted from least cluttered to most cluttered) under different visual ranges of the robots. The trend of the figure implies that as the clutter of the environment increases, the number of guards declines in general.

In order to eliminate the dependency of the results on specific maps, we use the *Cyclic Coverage* approach as the benchmark and the results of the repeated coverage algorithms on different criteria are re-calculated as ratios to this reference solution. For each optimization metric (*i.e.* *TPL, TAVP, TWVP, BWD*), the average values of ratios over all the maps are shown respectively in Figures 7, 8, 9, 10 for different numbers of robots (1, 2, ..., 15), under the selected visual ranges.

In sum, we have collected data from $5400 = 4 \text{ CoverageAlgorithms} \times 15 \text{ Robots} \times 3 \text{ VisualRanges} \times 30 \text{ Maps}$ runs of the simulator. The results can be used as a guideline for selecting an appropriate combination of repeated coverage algorithm and the choice of robots' visual range based on the particular scenario and the metric to be optimized.

A. Results Under Total Path Length

Figure 7 shows the performance of the coverage algorithms under *Total Path Length* on the basis of the pre-determined visual ranges of the robots.

In all the tested visual ranges for the robots, at least one of the *Cluster-based* approaches (*i.e.* *Uninformed Clustering Coverage*) outperforms the *Cyclic Coverage* algorithm, and interestingly as the visual range of the robots increases, there

are more *Cluster-based* algorithms that dominate the *Cyclic Coverage* approach, especially in the scenarios in which more robots are involved. The *Node-based Clustering Coverage* approach also dominates the *Edge-based Clustering Coverage* algorithm. This dominance intensifies with the increase in the robots' visual range.

Under this criterion, the choice of tour building algorithm significantly affects the coverage approaches, in which using *CLK* lead to shorter paths for the robots compared with *Double-MST*.

The results imply that although the *Cyclic Coverage* algorithm produces *optimal* or *near-optimal* solutions for *single-robot* cases, it is not the best solution when extending the problem to multi-robot scenarios.

B. Results Under Total Average Visiting Period

Figure 8 shows the performance of the coverage algorithms under *Total Average Visiting Period* on the basis of the pre-determined visual ranges of the robots.

As the visual range of the robots increases, the *Cluster-based* approaches show better performance. For the medium and high visual ranges, both the *Node-based Clustering Coverage* and the *Edge-based Clustering Coverage* algorithms dominate the *Cyclic Coverage* algorithm, especially in the scenarios in which more robots are involved. However, for the low visual range, the *Cyclic Coverage* algorithm is the best choice of the coverage mission.

Under this criterion, the choice of the tour building algorithm significantly affects the coverage approaches, such that using *Double-MST* outperforms *CLK* in minimizing the *Total Average Visiting Period* of the points in the target area.

C. Results Under Total Worst Visiting Period

Figure 9 shows the performance of the coverage algorithms under *Total Worst Visiting Period* on the basis of the pre-determined visual ranges of the robots.

An interesting observation is the decrease of the difference in performance among the *Cluster-based* approaches with the increase in the robots' visual range. Nonetheless, The *Cyclic Coverage* approach is the best choice for minimizing the *Total Worst Visiting Period*, regardless of the visual range of the robots. The *Node-based Clustering Coverage* approach also outperforms the other *Cluster-based* approaches.

Under this criterion, the choice of tour building algorithm affects the coverage approaches especially for the low visual range, such that using *CLK* outperforms the *Double-MST* in minimizing the *Total Worst Visiting Period* of the points in the target area.

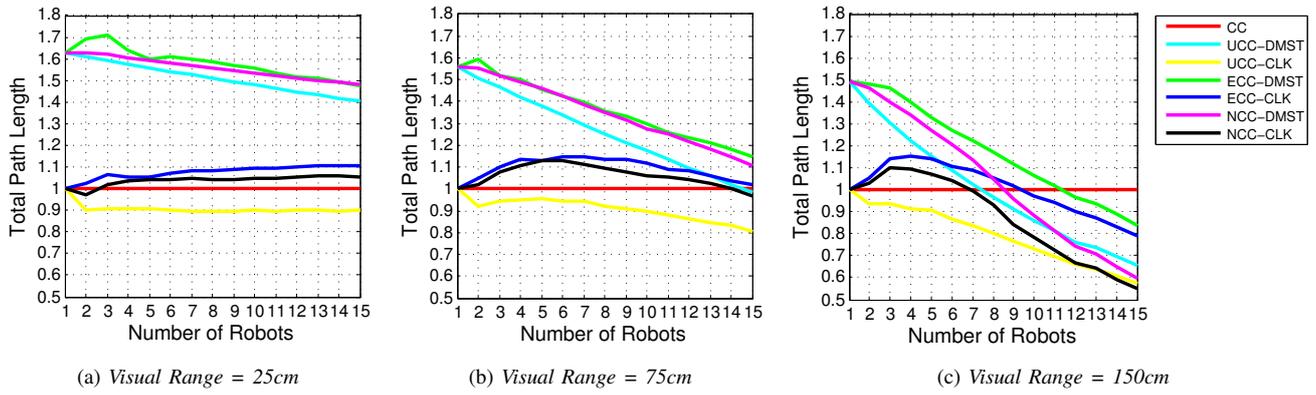


Figure 7: Total Path Length.

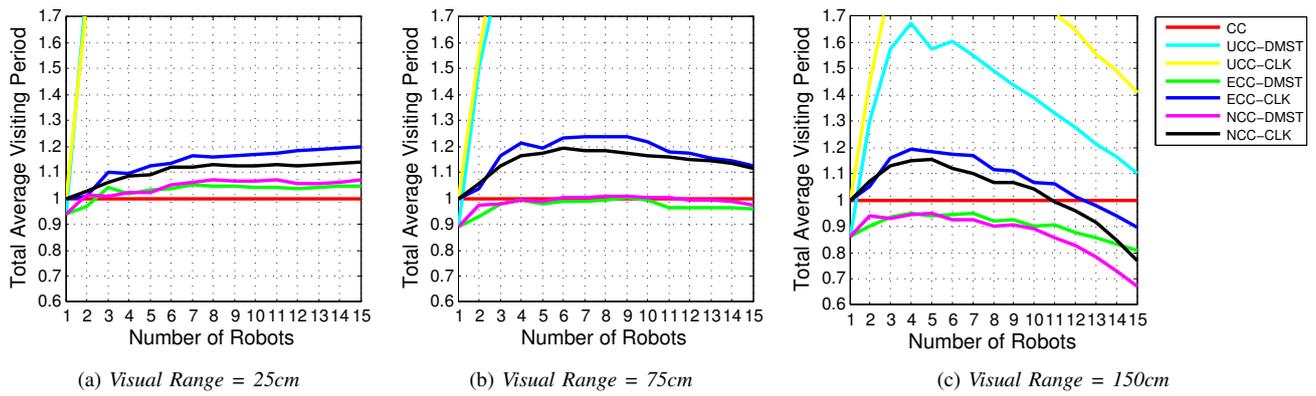


Figure 8: Total Average Visiting Period.

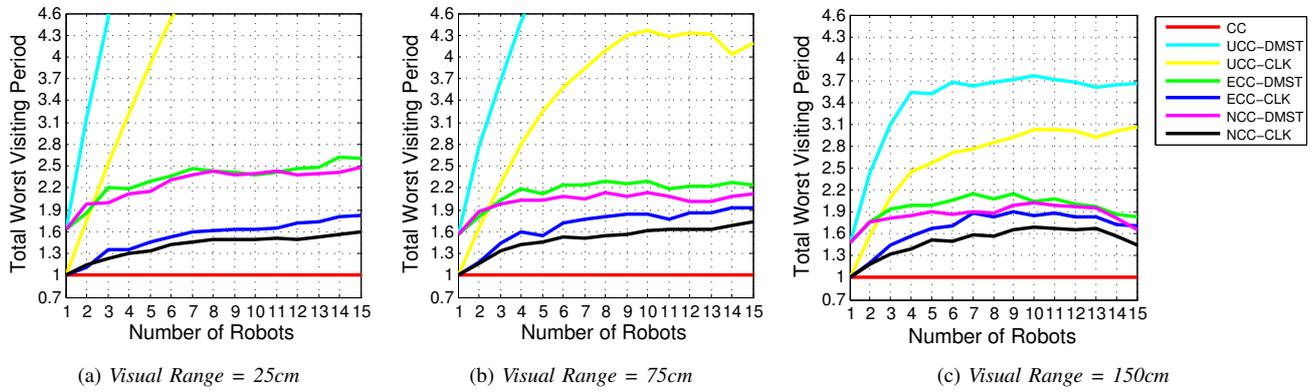


Figure 9: Total Worst Visiting Period.

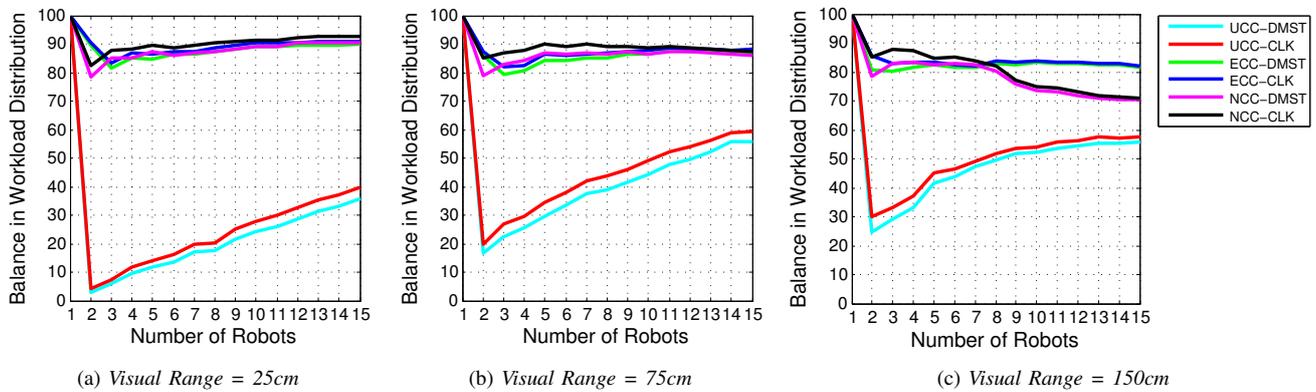


Figure 10: Balance in Workload Distribution.

D. Results Under Balance in Workload Decomposition

Figure 10 shows the performance of the coverage algorithms under *Balance in Workload Decomposition* on the basis of the pre-determined visual ranges of the robots.

For the low visual range, we noticed improvements in the *Balance in Workload Distribution* with the increase in the number of robots in the environment. However, for the medium visual range, this improvement disappears for the *Node-based Clustering Coverage* and the *Edge-based Clustering Coverage* algorithms, and for the high visual range, the *Balance in Workload Distribution* declines drastically in the *Node-based Clustering Coverage* algorithm, especially in the scenarios in which more robots are involved.

In summary, for the low visual range, the *Node-based Clustering Coverage* approach is the best choice for the coverage mission; however, with the increase of the visual range, the *Edge-based Clustering Coverage* approach dominates the *Node-based Clustering Coverage* algorithm for maximizing the *Balance in Workload Distribution*.

Under this criterion, the choice of tour building algorithm affects the coverage approaches, such that using *CLK* slightly outperforms the *Double-MST* in maximizing the *Balance in Workload Distribution* among the robots.

VIII. FUTURE WORK

There are many challenging future research directions for this problem, including:

- 1) **Heterogeneity:** In a coverage scenario, various forms of heterogeneity can be allowed such as different motion or sensing capabilities of the robots.
- 2) **Open Systems:** A new robot could be added to the team during execution. To this end, the robots should recalculate their paths so that the coverage mission could incorporate the newly added robot.
- 3) **Priority:** In some applications, parts of the target area should be visited or covered more often than others.
- 4) **Robustness:** There are many robustness criteria that need to be dealt with in the real world, such as robot action failure, communication failure, message loss, and the like.
- 5) **Communication:** The robots could have a limited range of communication, meaning a message sent by a robot is transmitted only to robots within a certain distance.
- 6) **Uncertainty:** Noisy sensors of the robots, action uncertainty, unknown obstacles, and the like. can lead to different challenging problems. In the case of noisy sensors, the accuracy of the area information achieved by a robot varies with the distance of the area from the robot.
- 7) **Dynamic Environments:** The robot team should have the ability to change its behavior over time in response to a changing environment, with dynamic obstacles, either to improve performance or to prevent unnecessary degradation in performance.

REFERENCES

- [1] P. Fazli, A. Davoodi, P. Pasquier, and A. K. Mackworth, "Complete and robust cooperative robot area coverage with limited range," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010*, 2010, pp. 5577–5582.
- [2] H. Choset, "Coverage for robotics – a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.
- [3] N. Hazon and G. A. Kaminka, "On redundancy, efficiency, and robustness in coverage for multiple robots," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1102–1114, 2008.
- [4] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 109–142, Apr. 2008.
- [5] A. Machado, G. Ramalho, J.-D. Zucker, and A. Drogoul, "Multi-agent patrolling: an empirical analysis of alternative architectures," in *Proceedings of the 3rd International Workshop on Multi-agent-based Simulation II, MABS 2002*. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 155–170.
- [6] Y. Elmaliach, N. Agmon, and G. A. Kaminka, "Multi-robot area patrol under frequency constraints," *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3-4, pp. 293–320, 2009.
- [7] A. Gasparri, B. Krishnamachari, and G. Sukhatme, "A framework for multi-robot node coverage in sensor networks," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2, pp. 281–305, 2008.
- [8] P. Toth and D. Vigo, *The vehicle routing problem*. Philadelphia, PA: Society for Industrial Mathematics, 2002.
- [9] S. Carlsson, B. J. Nilsson, and S. C. Ntafos, "Optimum guard covers and m-watchmen routes for restricted polygons," *International Journal of Computational Geometry and Applications*, vol. 3, no. 1, pp. 85–105, 1993.
- [10] F. Liu and S. Shen, "A method for vehicle routing problem with multiple vehicle types and time windows," *Natural Science Council*, vol. 23, no. 4, pp. 526–536, 1999.
- [11] J. O'Rourke, *Art gallery theorems and algorithms*. New York, NY, USA: Oxford University Press, 1987.
- [12] E. Packer, "Robust geometric computing and optimal visibility coverage," Ph.D. dissertation, Stony Brook, NY, USA, 2008.
- [13] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.
- [14] A. Almeida, G. Ramalho, H. Santana, P. A. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyre, "Recent advances on multi-agent patrolling," in *Proceedings of the Brazilian Symposium on Artificial Intelligence, SBIA 2004*, 2004, pp. 474–483.
- [15] B. Zalik and G. J. Clapworthy, "A universal trapezoidation algorithm for planar polygons," *Computers & Graphics*, vol. 23, no. 3, pp. 353 – 363, 1999.
- [16] G. D. Kazazakis and A. A. Argyros, "Fast positioning of limited visibility guards for inspection of 2D workspaces," in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2002*, 2002, pp. 2843–2848.
- [17] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
- [18] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems," *INFORMS Journal on Computing*, vol. 15, pp. 82–92, January 2003.
- [19] A. Tomás and A. Bajuelos, "Quadratic-time linear-space algorithms for generating orthogonal polygons with a given number of vertices," in *International Conference on Computational Science and Its Applications, ICCSA 2004*, 2004, pp. 117–126.