Auton Robot (2013) 34:251–276
DOI 10.1007/s10514-012-9319-7

Multi-robot repeated area coverage

Pooyan Fazli · Alireza Davoodi · Alan K. Mackworth

Received: 4 November 2010 / Accepted: 29 December 2012 / Published online: 1 March 2013

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 effects of the variables: *Environment Representation*, and *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 can be used as a framework for choosing an appropriate combination of repeated coverage algorithm, environment representation, and the robots' visual range based on the particular scenario and the metric to be optimized.

Keywords Multi-robot systems · Teamwork · Coordination · Area Coverage · Visibility Graph · Constrained Delaunay Triangulation · Uninformed Clustering Coverage · Edge-based Clustering Coverage · Node-based Clustering Coverage · Cyclic Coverage · Chained Lin–Kernighan Algorithm · Double-Minimum Spanning Tree

P. Fazli (✉)· A. Davoodi · A. K. Mackworth
Department of Computer Science, University of British Columbia,
Vancouver, BC V6T 1Z4, Canada
e-mail: pooyanf@cs.ubc.ca

A. Davoodi
e-mail: davoodi@cs.ubc.ca

A.K. Mackworth
e-mail: mack@cs.ubc.ca

1 Introduction

Given the dynamic and uncertain environments in which future robots will have to work compared to those of the familiar, relatively simple industrial robots, the integration of the advanced physical and cognitive systems required by the next generation of robots is a challenging task. It is not feasible to design a single 'universal' robot capable of working within a wide range of applications. PR2 and Nao are examples of the challenges of cost, long product life cycle and limited functionality which will apply to future robots as well. Given these challenges, multi-robot systems may be suitable alternatives to single-robot systems in many real world applications.

It is generally believed that multi-robot systems hold several advantages over single-robot systems. The most common motivations for developing multi-robot system solutions in real world applications are that a single robot cannot deal with task complexity adequately; the task is spatiotemporally distributed; building several niche, resource-bound robots is easier than building a single powerful robot; multiple robots can support parallelism; and finally, redundancy increases robustness.

Distributed *Area Coverage* as a task for multi-robot systems is a challenging problem in different scenarios such as search and rescue operations (Jennings et al. 1997), planetary exploration (Mataric and Sukhatme 2001), intruder detection (Gerkey et al. 2006; LaValle and Hinrichsen 2001), environment monitoring (Lavalle et al. 1997), floor cleaning (Hofner and Schmidt 1994) 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 is confusion in the literature regarding the terms *Exploration* and *Coverage*. To clarify the problem definition, we note that in exploration, we have an unknown environment in which a team of robots is trying to build a map of the area together (Yamauchi 1998; Burgard et al. 2000; Simmons et al. 2000; Howard 2006; Batalin and Sukhatme 2003). On the other hand, in a coverage problem, the map of the environment may be known or unknown and the team aims to cooperatively visit the whole area with their sensors or physical actuators. In other words, building a map of the environment is not the ultimate aim of the coverage mission.

A similar class of problems is *Boundary Coverage* in which the aim is inspection of the boundaries of a target area and the obstacles inside by a team of robots rather than complete coverage of the area (Williams and Burdick 2006; Amstutz et al. 2008).

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 (Fazli et al. 2010a,b; Hazon and Kaminka 2008; Rekleitis et al. 2008).
- *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 weighted average event/intruder detection time, minimizing the sum/maximum length of the paths/tours generated for the robots, or balancing the workload distribution among the robots. 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[1].

Several research communities including robotics/agents (Machado et al. 2002), sensor networks (Gasparri et al. 2008), operations research (Toth and Vigo 2002b) and computational geometry (Carlsson et al. 1993) work on variants of the repeated coverage problem.

In operations research, the *Vehicle Routing Problem* has some similarities to the repeated coverage scenarios (Liu and Shen 1999). In this problem, a number of vehicles deliver goods located at a central depot to a set of geographically dispersed customers. The objective is to minimize the total distance travelled. In the *Vehicle Routing Problem with Time Windows*, the target locations have time windows within which the deliveries (or visits) must be made (Desrochers et al. 1992), and in the *Capacitated Vehicle Routing Problem*, the vehicles have limited carrying capacity for the goods that must be delivered (Toth and Vigo 2002a).

In computational geometry, this problem originates from the *Art Gallery Problem* (O'Rourke 1987; Urrutia 2000) and its variant for mobile *guards*, the *(Multi) Watchman Route Problem* (Chin and Ntafos 1986; Packer 2008; Faigl 2010). In the *Art Gallery Problem*, the goal is to find a minimum number of static *guards* (control points) which can jointly cover *a priori* known simple polygonal workspace under different restrictions. On the other hand, in the *(Multi) Watchman Route Problem* the objective is to compute routes (closed curves) watchmen should take, such that any point inside the polygon is visible from at least one point along one route. *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 (Gerkey et al. 2006; Vidal et al. 2002). 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/Boundary/Perimeter Patrolling*. In Sect. 2 we elaborate on the related literature regarding the *Multi-Agent/Robot Patrolling* scenarios.

Overview and Contributions: In this study, we define four optimization metrics for comparing the efficiency of the repeated coverage algorithms. These metrics are:

- *Total Path Length (TPL)* the robots traverse in the target area.
- *Total Average Visiting Period (TAVP)* of the points of interest in the target area.
- *Total Worst Visiting Period (TWVP)* of the points of interest in the target area.
- *Balance in Workload Distribution (BWD)* among the robots.

These metrics will be discussed in detail in Sect. 3. Interestingly, it is impossible even to develop polynomial approximation algorithms, when optimizing each of the metrics of the repeated coverage problem, unless $P = NP$ (Packer 2008). 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 contributions of this paper are as follows:

1. We present an approach to modeling the environment using graph-based methods considering the limited visual

[1] In this paper, we use the terms *'coverage'* and *'repeated coverage'* interchangeably.

range of the robots. To this end, two environment modeling approaches are developed based on the *Visibility Graph* and the *Constrained Delaunay Triangulation*.
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 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 different environment representations on the performance of the repeated coverage algorithms are examined.
5. The effect of varying the robots' visual range on the performance of the repeated coverage algorithms is investigated.

The results can be used as a framework for choosing an appropriate combination of repeated coverage algorithm, environment representation, and the robots' visual range based on the particular scenario and the metric to be optimized.

2 Background and review

Choset (2001) divides the approaches for area coverage based on the methods they employ for decomposing the area: *Exact Cellular Decomposition*, and *Approximate Cellular Decomposition*.

In the *Exact Cellular Decomposition* (e.g. graph-based methods), the area is divided to a set of non-overlapping regions whose union covers the whole environment. In the *Approximate Cellular Decomposition* (e.g. grid-based methods), the environment is divided into cells which are all the same size and shape. However, cells that are partially occluded by obstacles or close to the boundary are discarded, therefore the union of the cells only *approximates* the target area.

The methods based on the *Approximate Cellular Decomposition* 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. In contrast, methods based on the *Exact Cellular Decomposition* do not suffer those restrictions. However, while traversing the graph guarantees covering the whole environment in continuous spaces, the path may include many redundant movements.

Although there is a wide body of literature for single coverage scenarios (Choset 2001; Agmon et al. 2008a; Rekleitis et al. 2008; Gabriely and Rimon 2011; Zheng et al. 2005; Kurabayashi et al. 1996; Batalin and Sukhatme 2002), repeated coverage has not received the same attention. Two classes of the repeated coverage problem in the literature are: *1) Area Patrolling*, and *2) Boundary or Perimeter Patrolling (Open or Closed Polylines)*, and each is divided into:

- *Optimization-based repeated coverage*, in which the team's goal is to optimize some criteria, for example minimizing the average or worst frequency of visiting the points of interests in the target area, minimizing the total path traversed by the robots in the environment, or balancing the workload distribution among the robots. *Optimization-based repeated coverage* is the focus of this paper.
- *Adversarial repeated coverage*, in which the team's goal is to maximize the probability of detecting an adversary or multiple adversaries trying to penetrate the environment mainly through the patrol paths.

2.1 Optimization-based repeated coverage

Machado et al. (2002) studied several architectures for repeated coverage in non-weighted graphs (i.e. the distance between two adjacent nodes is uniform). The goal was to minimize the time gap between two visits to the same node. The proposed architectures differ on various parameters such as agent type (reactive or goal-oriented), agent communication (centralized, peer-to-peer, flag-based, or no-communication), coordination scheme (centralized vs. distributed), agent perception (local vs. global), and decision-making (random selection vs. goal-oriented selection). They showed that the *Conscientious Reactive Agents* architecture outperforms the other multi-agent architectures. An agent in *Conscientious Reactive Agents* chooses a node to visit from its neighbourhood with the highest time of being unvisited relative to the agent's own visits rather than all the other agents' visits. There is no communication among the agents. The approach is generalized to weighted graphs in Almeida et al. (2004)

Santana et al. (2004) studied adaptive agents that learn to patrol weighted graphs to minimize the time intervals between visits to the nodes, using *Reinforcement Learning (RL)* techniques. A *Markov Decision Process (MDP)* formalism was used to model the patrolling problem, and the challenge was to define a state and action space for each agent individually, and to develop proper models of instantaneous rewards which could lead to satisfactory long term performance. The *Q-Learning* algorithm was used to train the agents, which proved to be computationally expensive.

A 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. (2004) and Machado et al. (2002) 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. Santana et al. (2004) also used two very similar maps to evaluate their proposed patrolling algorithm. Moreover, in none of the above patrolling tasks, did the authors 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. This 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 different representations of the target area and varying the robots' visual range on the performance of the repeated coverage algorithms with extensive experiments on a set of carefully selected maps.

Elmaliach et al. (2009) 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. As mentioned above, grid-based representations have limitations in handling partially occluded cells or cover areas close to the boundaries in continuous spaces. Also, one of the limitations of the proposed approach is the requirement for a corridor's size in the environment to be at least twice the size of the robot in order to be covered. Our algorithms, on the other hand, guarantee complete coverage of the area.

Elmaliach et al. (2008) also addressed the problem of frequency-based patrolling of open polylines (e.g., as in open-ended fences), where the two endpoints of the polylines are not connected. Jensen et al. (2011) extended Elmaliach et al.'s work on patrolling open polylines, with a focus on maintaining the patrol over the long-term. They accomplish this task by replacing the robots having power level below a threshold with some reserve robots. Patrolling open perimeters is challenging because robots must revisit the just visited areas when they reach an endpoint and turn back. Boardman et al. (2010) presented a distributed boundary tracking controller for multi-robot systems. Within this system, the boundary is partitioned into sub-segments, each allocated to a robot, such that the workload is balanced among the robots. They also aimed to minimize the phase difference between the robots, to limit the size of the gap created between the robots.

2.2 Adversarial repeated coverage

Some work considered the existence of adversary agents in the workspace, where the goal is to maximize the detection rate of the intruder(s) in the work space. The main idea behind these patrolling strategies is to use non-deterministic, probabilistic algorithms in order to avoid static patrolling patterns which, in an adversarial scenario, could be exploited by intruders.

Sak et al. (2008) considered the case of multi-agent patrol in adversarial environments in general graphs. The authors assumed three types of intruders in the environment: a random intruder, an intruder that waits until the patrolling agent leaves a node to penetrate the area, and an statistical intruder that collects statistics on the period between visits to a random node and predicts the timing of the next safe visit to the node. Some patrolling algorithms were experimentally evaluated by simulation. The results showed that no patrol strategy was optimal for all the possible adversaries.

Ahmadi et al. (2006) addressed the problem of multi-robot repeated coverage of a grid-based target area in order to detect a set of events of interest. The frequency of the events occurrence in the environment can possibly be non-uniform. Thus, the robots should visit the points with non-uniform frequency. The main contribution of the paper is an online area partitioning method among the robots through a negotiation mechanism, which is adaptive to non-uniform frequency of event occurrence in the target area.

Guo et al. (2007) studied a centralized multi-robot system for patrolling continuous environments. In this paper, the area is partitioned into sub-regions using a *Voronoi Diagram*. Robots are then distributed from their initial positions to their sub-regions, and finally, each robot patrols its sub-region in order to detect a possible intrusion of an adversary agent into that sub-region.

Agmon et al. (2008b) studied patrolling a cyclic boundary, in which the robots' goal is to maximize their rewards by detecting an adversary agent, which attempts to penetrate through a point on the boundary unknown to the robots. In their scenario, the full-knowledge adversary knows the location of the robots and the patrol strategy and needs a time interval of length t to accomplish the intrusion. They also examined the case of a zero-knowledge adversary (Agmon et al. 2008c) and a partial-knowledge adversary (Agmon et al. 2009b) in perimeter patrolling. The uncertainty in the robots' perception was investigated in (Agmon et al. 2009a), in which the ability to detect the intruders decreased as the distance grew. Czyzowicz et al. (2011) addressed the same problem using a team of verable speed robots.

Fazli and Mackworth (2012a,b) addressed the problem of repeated coverage by a team of robots of the boundaries of a target area and the structures inside it. The robots have limited visual and communication range. Events may occur on any parts of the boundaries and may have different importance weights. In addition, the boundaries of the area and the structures are heterogeneous, so that events may appear with varying probabilities on different parts of the boundary, and these probabilities may change over time. The goal is to maximize the reward by detecting the maximum number of events, weighted by their importance, in minimum time. The reward a robot receives for detecting an event depends on how early the event is detected.

Girard et al. (2004) studied a centralized system composed of multiple unmanned air vehicles patrolling a border area. The border is represented as a continuous two-dimensional region divided in sub-regions. Each sub-region is assigned to an air vehicle that repeatedly patrols it with a spiral trajectory to detect the possible intrusions in that sub-region.

3 Problem definition and preliminaries

The problem is to cover the environment repeatedly over time using a given 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 Each robot, in the set of robots R, is represented by a point in the environment.

This assumption reduces the problem to covering a polygonal environment by a team of point robots without loss of generality. This follows from using the standard approach to robot motion planning in an environment with polygonal obstacles for the computation of the configuration space. In this case, configuration space is the set of all admissible positions of the robot, i.e. the *Minkowski sum* (Skiena 1998) of the set of obstacles and the boundary of the environment with the shape of the robot

Assumption 3 The robots are presumed to have a 360° field of view and a predefined circular limit of visual range.

Assumption 4 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 that the point may be visited more than once in different time intervals by one or more than one robot in a *Full Single Coverage*. For example, in Fig. 1, point A (shown by the red dot) has three *Visiting Periods*, two are determined by the black robot/tour and one is determined by the blue robot/tour [2].

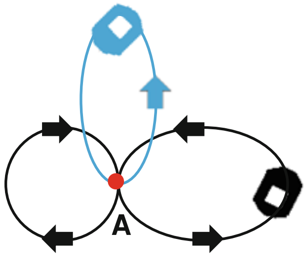


Fig. 1 *Visiting Period* and *Visiting Frequency*

- *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*. If 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 associated with a different robot. For example, in Fig. 1, point A has two *Visiting Frequencies*, one is determined by the black robot/tour and the other is determined by the blue robot/tour. The *Visiting Frequency* of point A on the black tour is 2 and on the blue tour is 1.

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

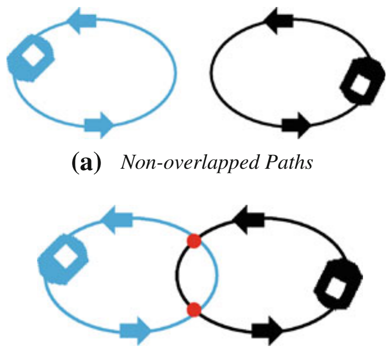
- *Total Path Lengths (TPL):* the sum of the lengths of the paths assigned to the robots in order to have a *Full Single Coverage*.
- *Total Average Visiting Period (TAVP):* the average of the *Average Visiting Periods* of all the points of interest in the target area.
- *Total Worst Visiting Period (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.

3.1 Computing the evaluation metrics

The paths of the robots in the target area may either not overlap, that is, there is no common *Point of Interest* among

[2] All figures in this paper are best viewed in color.



(a) *Non-overlapped Paths*

(b) *Overlapped Paths*

Fig. 2 Non-overlapped versus overlapped paths for two robots

the robots' paths (as shown in Fig. 2), or may overlap, that is, there are some common *Points of Interest* among the robots' paths (e.g. red dots shown in Fig. 2b). Considering this, the evaluation metrics are defined as below:

$$TPL = \sum_{i=1}^{|R|} Length(Path(r_i)). \tag{1}$$

$|R|$ is the number of robots, $Path(r_i)$ is the path built for robot r_i, and $Length(Path(r_i))$ is the length of the path.

$$TAVP = \frac{\sum_{node \in PoI} AVP(node)}{|PoI|}, \quad PoI = \textit{Points of Interest}, \tag{2}$$

where

$AVP(node) = \frac{Length(Path(r_i))}{VF_i(node)}$, $i \in \{1, 2, \ldots, |R|\}$, $node \in Path(r_i)$, if *node* is not a common *Point of Interest* among the robots' paths, or

$AVP(node) = \frac{1}{\left[\sum_i \frac{VF_i(node)}{Length(Path(r_i))}\right]}$, $i \in \{1, 2, \ldots, |R|\}$, $node \in Path(r_i)$, if *node* is a common *Point of Interest* among the robots' overlapped paths.

$VF_i(node)$ is the *node Visiting Frequency* in $Path(r_i)$.

$$TWVP = max_{node \in PoI}(WVP(node)), \tag{3}$$

where

$WVP(node) = max\,\{VP_i(node)\}$, $node \in Path(r_i)$, if *node* is not a common *Point of Interest* among the robots' paths, or

$WVP(node) = min_{i=1,2,\ldots,|R|}\{max\,\{VP_i(node)\}\}$, $node \in Path(r_i)$, if *node* is a common *Point of Interest* among the robots' overlapped paths.

$\{VP_i(node)\}$ is the set of all the *Visiting Periods* of the *node* in $Path(r_i)$.

In computing the *WVP* of a *Point of Interest* which is common among the robots' paths, we first calculate the maximum *Visiting Period* of the point in each robot's path, and then choose the minimum of the maximum values. Recall that each *Point of Interest* can belong to the path of more than one robot, and can also have more than one *Visiting Period* in each robot's path.

$$BWD(Paths) = \left(1 - \frac{STD(\{Length(Path(r_i))|i=1, 2, \ldots, |R|\})}{STD(\{TPL, \alpha_1, \alpha_2, \ldots, \alpha_{|R|-1}|\alpha_i=0\})}\right) \times 100, \tag{4}$$

where $STD(.)$ is the population standard deviation, and $Paths = \{Path(r_1), Path(r_2), \ldots, Path(r_{|R|})\}$ is the $|R|$ paths created for the $|R|$ robots. For the case of one robot, we assume that $BWD(Paths) = 100$. In BWD's computation, $STD(\{Length(Path(r_i))|i = 1, 2, \ldots, |R|\})$ is the population standard deviation of the set of paths created for the robots, and $STD(\{TPL, \alpha_1, \alpha_2, \ldots, \alpha_{|R|-1}|\alpha_i = 0\})$ 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 ($\alpha_1, \alpha_2, \ldots, \alpha_{|R|-1}$).

3.2 Stages of the repeated coverage algorithms

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

1. The locations of static *guards* (*Points of Interest)* required to cover visually a given 2D environment are determined, allowing for the limited range of the robots' vision (Sect. 4).
2. A graph is built on the guards and the obstacles based on either the *Visibility Graph* or the *Constrained Delaunay Triangulation* (Sect. 5).
3. The graph is reduced to either the *Reduced-Vis* or the *Reduced-CDT* representation (Sect. 6).
4. *Cluster-based Coverage Algorithms:* The *Reduced Graph* 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 (Sect. 7).
5. *Cyclic Coverage Algorithm: Cyclic Coverage* finds the shortest tour on the whole *VG* or *CDT* graph, passing through all the static *guards*, and then distributes the robots equidistantly around it (Sect. 8).

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.

4 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 taken from the *Art Gallery Problem* (O'Rourke 1987). These static *guards* are control points that can jointly cover the whole environment while satisfying the limited visual range constraint of the robots (Faigl et al. 2011). 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 (Zalik and Clapworthy 1999). 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 (Kazazakis and Argyros 2002) 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 3c, d show the computed guards after trapezoidation (Fig. 3b) on the sample environment of Fig. 3a.

5 Building the graphs

Having located the static guards in the previous step, the *Visibility Graph* and the *Constrained Delaunay Triangulation* are then built on the obstacles' nodes and the computed guards.

5.1 Visibility Graph

The *Visibility Graph (VG)* is a graph structure used in computational geometry and robot motion planning (Latombe 1991). 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 *VG*. Two nodes of the environment are mutually visible if the line segment joining them does not intersect any obstacle (De Berg et al. 2008). Figure 4a illustrates the *Visibility Graph* built on the sample environment.

5.2 Constrained Delaunay Triangulation

The *Delaunay Triangulation* of a set of nodes in the Euclidean plane is a triangulation such that the circumcircle of any triangle in the triangulation does not contain nodes other than the three that define it (Lee and Schachter 1980). The *Delaunay Triangulation* corresponds to the dual graph of the *Voronoi Tessellation* (Okabe et al. 2000).

The *Constrained Delaunay Triangulation (CDT)* is a variant of the standard *Delaunay Triangulation* in which a set of pre-specified edges (in our case, the edges of the obstacles) must lie in the triangulation (Chew 1987). A *Constrained Delaunay Triangulation* is not truly a *Delaunay Triangulation*. Some of its triangles might not be *Delaunay*, but they are all *Constrained Delaunay*. Figure 4 illustrates the *Constrained Delaunay Triangulation* built on the sample environment.

6 Graph Reduction

The aim of the Graph Reduction method is to improve efficiency by minimizing the average or total time taken for the robots to traverse the graph. *Algorithm 1* describes the steps of the construction of a *Reduced Graph* (*Reduced-Vis* or *Reduced-CDT*) on a given environment. The input to the algorithm is the *VG* or the *CDT* discussed in Sect. 5.

The method starts by using the *Floyd-Warshall* algorithm to find the set $MD = \left\{(c_{ij}, g_i, g_j) | g_i, g_j \in V_{vis-cdt}\right\}$ of

Algorithm 1: Graph Reduction

```
Input:
Graph $G_{vis-cdt}(V_{vis-cdt}, E_{vis-cdt})$, where $V_{vis-cdt} = SG \bigcup P$
/* VG or CDT */
$SG = \{g_1, g_2, \ldots, g_m\}$  /* Static Guards */
$P = \{p_1, p_2, \ldots, p_n\}$ /* Endpoints of Obstacles */

Output:
$G_{r-vis-cdt}(V_{r-vis-cdt}, E_{r-vis-cdt})$ where
$V_{r-vis-cdt} = SG \bigcup \widetilde{P}, \widetilde{P} \subset P$  /* Reduced Graph */

begin
    $V_{r-vis-cdt} \longleftarrow \phi$
    $E_{r-vis-cdt} \longleftarrow \phi$
    $(MD, SP) \longleftarrow FloydWarshall(G_{vis-cdt})$
    $(i, j) \longleftarrow \arg\min_{(i,j)}\{c_{ij} | (c_{ij}, g_i, g_j) \in MD \text{ and } g_i, g_j \in SG\}$
    $r_{ij} \longleftarrow GetCorrespondingShortestPath(i, j)$
    $G_{r-vis-cdt}(V_{r-vis-cdt}, E_{r-vis-cdt}) \longleftarrow$
    $InitialReducedGraph(r_{ij})$
    while ¬ all the guards added do
        $g \longleftarrow FindClosestGuardTo(G_{r-vis-cdt})$
        $Expand(G_{r-vis-cdt}, g)$
    end
    return $G_{r-vis-cdt}(V_{r-vis-cdt}, E_{r-vis-cdt})$
end
```

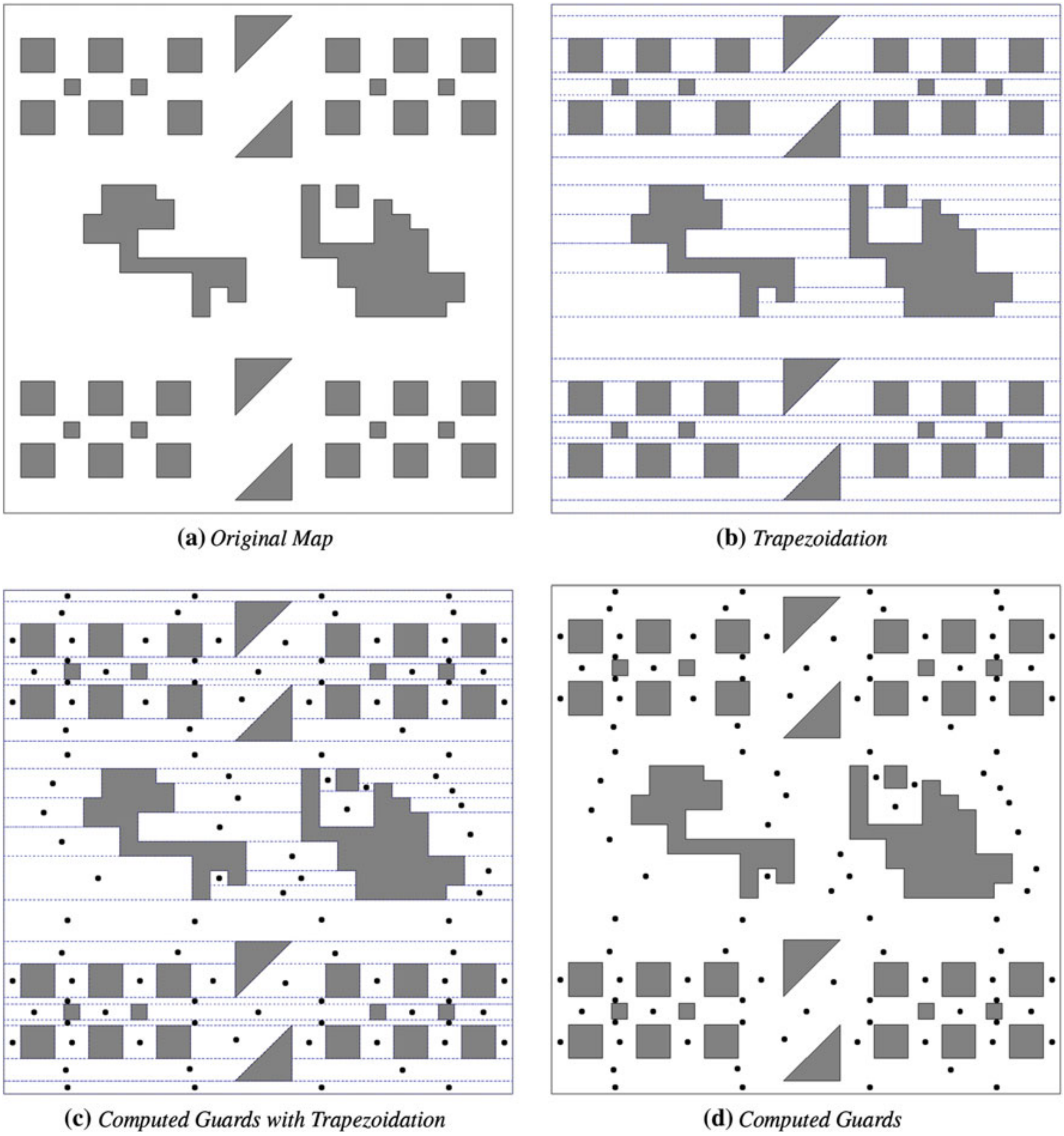


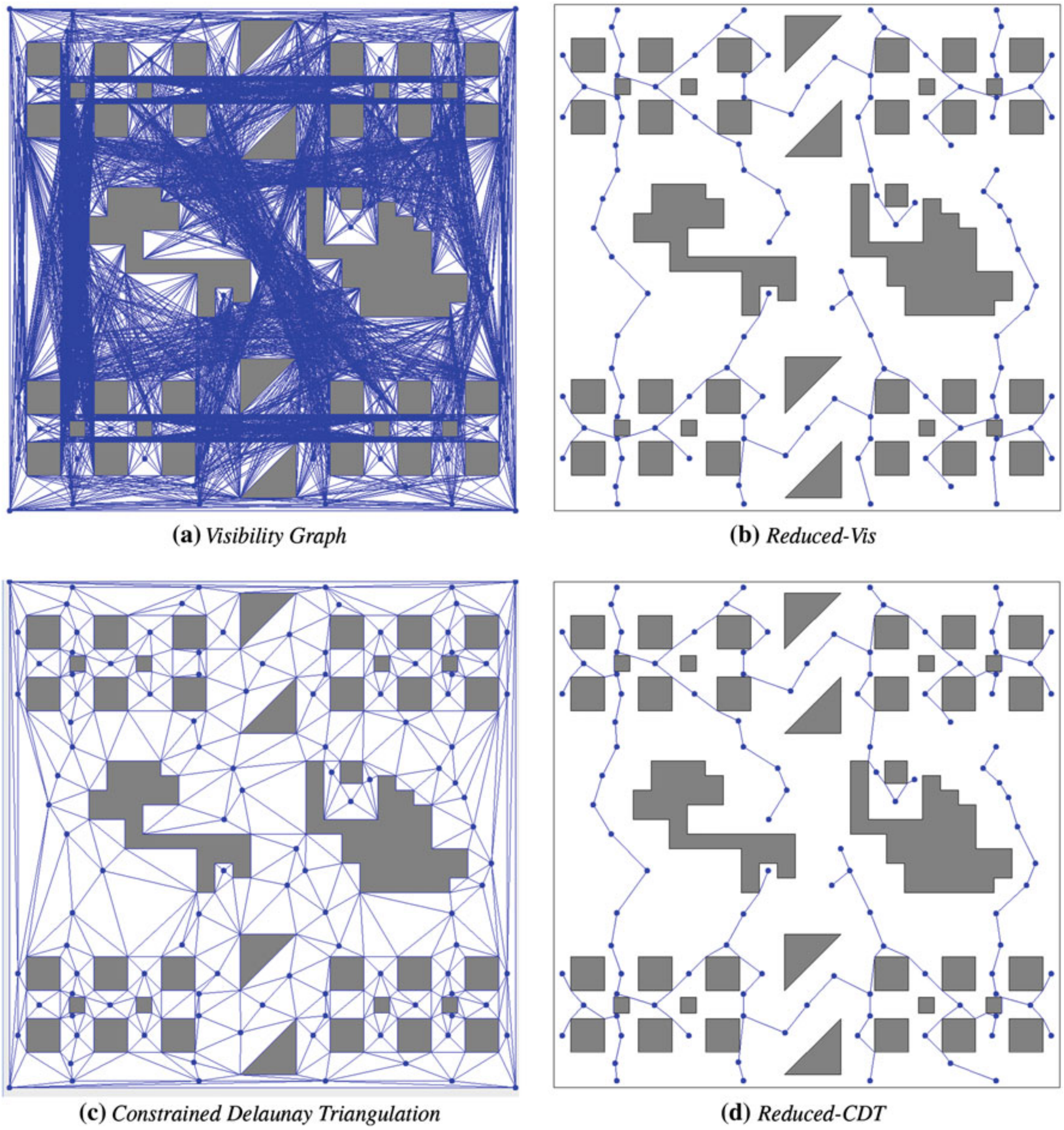
Fig. 3 Locating guards in a sample environment

minimum distances, c_{ij}, and the set $SP = \{(r_{ij}, g_i, g_j) | g_i, g_j \in V_{vis-cdt}\}$ of shortest paths, r_{ij}, between any pair of guards g_i and g_j of the input graph *(line 4)*.

The minimum value of all the minimum distances in MD is then selected *(line 5)* and its corresponding shortest path in SP *(line 6)*, including all its nodes and edges, forms the initial component of the *Reduced Graph (line 7)*.

Next, among all the guards that have not yet been added to the graph, the algorithm finds the closest guard to the current component *(line 9)*, merging the corresponding shortest path with it *(line 10)*. Following the same process, the algorithm keeps expanding the component until there are no more guards to be added to the graph *(lines 8–11)*. The resultant graph is the final *Reduced Graph (line 12)*. The nodes of the graph includes all the guards (SG), as the *Points of Interests* in the target area and the subset of the obstacles' nodes ($\widetilde{P} \subset P$). Traversing the *Reduced Graph* guarantees complete coverage of the target area given the limited visual range of the robots.

Figure 4b, d illustrate, respectively, the *Reduced-Vis* and the *Reduced-CDT* computed on the *VG* and the *CDT* of Figure 4a, c.


(a) *Visibility Graph*

(b) *Reduced-Vis*

(c) *Constrained Delaunay Triangulation*

(d) *Reduced-CDT*

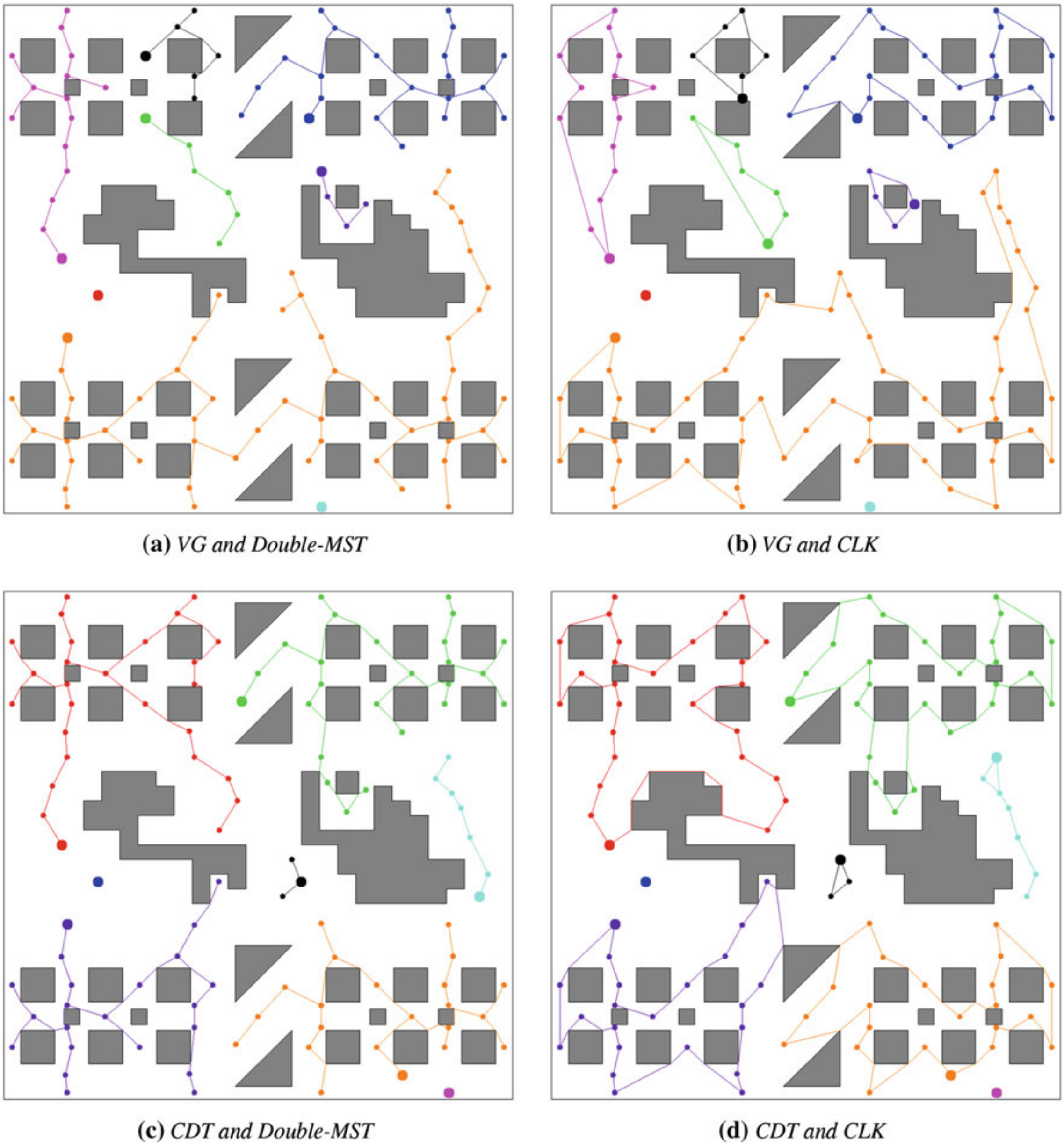
Fig. 4 Graphs and *Reduced Graphs*

7 Cluster-based coverage algorithms

Cluster-based coverage algorithms decompose the *Reduced Graph* into $|R|$ (number of robots) clusters, $T = \{T_1, T_2, \ldots, T_{|R|}\}$, such that $\bigcup_{i=1}^{|R|} V_{T_i} = SG$; SG is the set of all guards and V_{T_i} is the set of guards of the cluster T_i. Below, three different *Cluster-based* coverage algorithms are presented. The input of the proposed algorithms can be either of the *Reduced Graphs*. Having built the clusters on the *Reduced Graphs*, a tour is built on the generated cluster for each robot. The tour building algorithms are discussed in Sect. 7.4.

7.1 Uninformed Clustering Coverage

Uninformed Clustering Coverage (Algorithm 2) partitions the *Reduced Graph* into $|R|$ clusters by removing the $|R| - 1$ longest edges of the graph *(line 2)*. Thereafter, a tour is built on each cluster generated for the robots *(lines 3–5)*. There is no overlap (existence of common guards) among the tours built by *Uninformed Clustering Coverage*.


(a) *VG and Double-MST*

(b) *VG and CLK*

(c) *CDT and Double-MST*

(d) *CDT and CLK*

Fig. 5 Tours built by eight robots by *Uninformed Clustering Coverage*

Figure 5 illustrates the tours built for eight robots on the sample environment by the *Uninformed Clustering Coverage* algorithm.

7.2 Edge-based Clustering Coverage

Edge-based Clustering Coverage (Algorithm 3) builds a set of $|R|$ clusters on the *Reduced Graph*. The clusters are initiated as follows: the endpoint guards of the longest path in the original graph (*VG* or *CDT*) are selected as the starting points of the first two clusters. For the next cluster, a guard is selected such that it maximizes the sum of the distances from the starting points of the first two clusters. Similarly, for the next cluster, a guard is selected that maximizes the sum of the distances from the starting points of the first three clusters. This continues until $|R|$ initial guards are found for the $|R|$ clusters of the robots *(line 2)*. The aim of this part and the reason to initiate the clusters in the original *VG* or *CDT* graph is to distribute the clusters spatially as much as possible far away from each other in the target area. Distance between the guards in the original graph, in contrast with the Euclidean distance, takes into account the obstacles in the area, and is more accurate than the distance in the *Reduced Graph*, because there are many edges between the nodes in the original graph which were removed from the *Reduced Graph*.

Starting from the initial guards, clusters are expanded in the *Reduced Graph* sequentially by choosing one guard at a time, until all the guards of the *Reduced Graph* have been selected at least once *(lines 3–11)*. Each cluster selects guards in a way that satisfies the following constraints:

- Find the nearest immediate (that is, ignoring the corners of the obstacles) unselected guard in the *Reduced Graph*, add it and the corresponding edge/path to the cluster. 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 *(line 5)*. This aims to avoid the overlap among the clusters during their expansion as much as possible.
- Do not add a guard which has already been chosen by any other cluster, unless there is no other unselected immediate guard *(lines 6–8)*.

Remove Common Guards: When all the guards of the graph have been selected by at least one cluster, remove as many as possible of the common guards shared by the clusters of the robots. To this end, for each cluster, discard the guards and their corresponding edges/paths from the cluster if they have been selected earlier by any other cluster *(lines 13–19)*. There are cases where the overlap among the clusters cannot be resolved. If removing the overlap among the clusters disconnects any of them, then the overlap is left in all the clusters.

Finally, a tour is built on the generated cluster for each robot *(line 20)*.

Figure 6 illustrates the tours built for eight robots on the sample environment by the *Edge-based Clustering Coverage* algorithm.

Algorithm 2: Uninformed Clustering Coverage

```
Input:
$G_{r-vis-cdt}$ : the Reduced-Vis or the Reduced-CDT Graph
$|R|$: Number of Robots

Output:
A set of $|R|$ tours, $Tours = \{T_1, T_2, \ldots, T_{|R|}\}$ where
$\bigcup_{i=1}^{|R|} V_{T_i} = SG$, $SG$ is the set of all guards and $V_{T_i}$ is the set of
guards of the tour $T_i$

begin
    /*Remove $|R|-1$ longest edges of $G_{r-vis-cdt}$*/
    $Tours \longleftarrow ConstructClusters(G_{r-vis-cdt})$
    foreach $T_i \in Tours$ do
        $T_i \longleftarrow BuildTour(T_i)$
    end
    return $Tours$
end
```

Algorithm 3: Edge-based Clustering Coverage

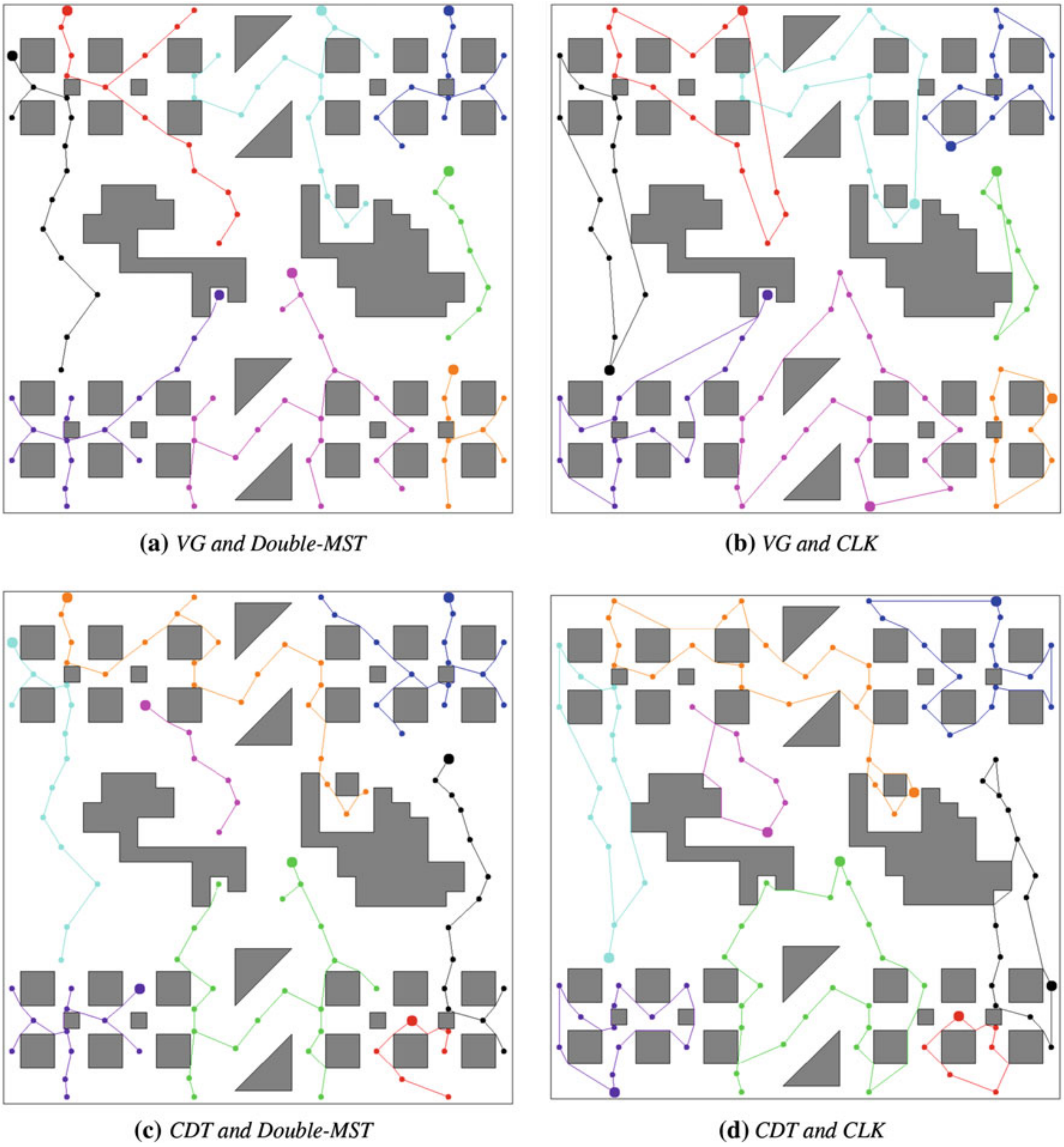
```
Input:
$G_{vis-cdt}(V_{vis-cdt}, E_{vis-cdt})$, where $V_{vis-cdt} = SG \bigcup P$  /* VG or CDT */
$SG = \{g_1, g_2, \ldots, g_m\}$  /* Static Guards */
$P = \{p_1, p_2, \ldots, p_n\}$ /* Endpoints of Obstacles */
$G_{r-vis-cdt}$ : the Reduced-Vis or the Reduced-CDT Graph
$|R|$: Number of Robots

Output:
A set of $|R|$ tours, $Tours = \{T_1, T_2, \ldots, T_{|R|}\}$ where
$\bigcup_{i=1}^{|R|} V_{T_i} = SG$, $SG$ is the set of all guards and $V_{T_i}$ is the set of
guards of the tour $T_i$

1  begin
2      $Tours \longleftarrow InitiateClusters(G_{vis-cdt})$
3      while ¬ all the guards of the graph $G_{r-vis-cdt}$ visited do
4          foreach $T_i \in Tours$ do
5              find $g \in V$ which is the nearest immediate guard to $T_i$
               in $G_{r-vis-cdt}$ and ¬ visited
6              if there is no such a guard $g$ then
7                  find $g \in V$ which is the nearest immediate guard
                   to $T_i$ in $G_{r-vis-cdt}$
8              end
9              $T_i.add(g)$
10         end
11     end
12     foreach $T_i \in Tours$ do
13         if there are common guards between $T_i$ and another
           cluster then
14             if the guards have been selected earlier by the other
               cluster then
15                 if ¬ removing the common guards from $T_i$
                   disconnects either cluster then
16                     $RemoveCommonGuardsFrom(T_i)$
17                 end
18             end
19         end
20         $T_i \longleftarrow BuildTour(T_i)$
21     end
22     return $Tours$
23 end
```

7.3 Node-based Clustering Coverage

Node-based Clustering Coverage (Algorithm 4) initially uses the *k-Means* clustering algorithm (Hartigan and Wong 1979) to divide the guards into $|R|$ disjoint clusters $\{T_1, T_2, \ldots, T_{|R|}\}$, in which each guard belongs to the cluster with the nearest mean *(line 2)*. In other words, given the set of guards $\{g_1, g_2, \ldots, g_m\}$, *k-Means* minimizes the within-cluster sum of squares:

$$\underset{T}{\operatorname{argmin}} \sum_{i=1}^{|R|} \sum_{g_j \in T_i} \left\| g_j - \mu_i \right\|^2, \tag{5}$$

where μ_i is the mean/centroid of the guards in cluster T_i. $\left\| g_j - \mu_i \right\|^2$ is the distance between a guard and the centroid



(a) *VG and Double-MST*

(b) *VG and CLK*

(c) *CDT and Double-MST*

(d) *CDT and CLK*

Fig. 6 Tours built by eight robots by *Edge-based Clustering Coverage*

of the cluster determined based on their distance in the original *VG* or *CDT* graph. Distance between the guards in the original graph, in contrast with the Euclidean distance, takes into account the obstacles in the area, and is more accurate than the distance in the *Reduced Graph*, because there are many edges between the nodes in the original graph which were removed from the *Reduced Graph*.

In the first iteration of *k-Means*, the initial means are found in the same way as finding the starting points of the clusters discussed in *Edge-based Clustering Coverage*. This aims to distribute the clusters spatially as much as possible far away from each other in the target area. Given this initial set of $|R|$ means, the algorithm proceeds by alternating between two steps: *(1) Assignment Step:* Assign each guard to the cluster with the closest mean, *(2) Update Step:* Calculate the new means to be the centroid of the guards in the cluster. Since the computed centroids may not lie on the guards of the graph, they are matched to the closest guard in the environment.

Having built the $|R|$ clusters on the guards *(line 3)*, we connect each pair of guards in each cluster if they have a corresponding path (including the intermediate nodes of the obstacles) in the *Reduced Graph (line 5)*. Thereafter, we do a connectivity test on all the clusters, meaning that every pair

Algorithm 4: Node-based Clustering Coverage

```
Input:
$G_{vis-cdt}(V_{vis-cdt}, E_{vis-cdt})$, where $V_{vis-cdt} = SG \bigcup P$   /* VG or CDT */
$SG = \{g_1, g_2, \ldots, g_m\}$   /* Static Guards */
$P = \{p_1, p_2, \ldots, p_n\}$ /* Endpoints of Obstacles */
$G_{r-vis-cdt}$ : the Reduced-Vis or the Reduced-CDT Graph
$|R|$: Number of Robots

Output:
A set of $|R|$ tours, $Tours = \{T_1, T_2, \ldots, T_{|R|}\}$ where
$\bigcup_{i=1}^{|R|} V_{T_i} = SG$, $SG$ is the set of all guards and $V_{T_i}$ is the set of
guards of the tour $T_i$

1  begin
2      $InitialCentroids \longleftarrow FindInitialCentroids(G_{vis-cdt}, |R|)$
3      $Tours \longleftarrow kMeans(G_{vis-cdt}, |R|, InitialCentroids)$
4      foreach $T_i \in Tours$ do
5          $T_i \longleftarrow ConnectGuards(T_i, G_{r-vis-cdt})$
6          $DCComponents \longleftarrow FindDCComponents(T_i)$
7          $T_i \longleftarrow BuildMST(G_{vis-cdt}, DCComponents)$
8          $T_i \longleftarrow BuildTour(T_i)$
9      end
10     return $Tours$
11 end
```

of guards in each cluster should be connected through a path. To this end, we first find the disconnected components within the cluster *(line 6)* and then compute the *Minimum Spanning Tree* on them by getting help from the edges of the original *VG* or *CDT* graph, and the nodes of the obstacles *(line 7)*. The *Minimum Spanning Tree* will not add any other new guard to the set of guards existing in the cluster. This prevents the overlap among the clusters; however, the tree can add any nodes of the obstacles in the environment to the cluster. We add the *Minimum Spanning Tree*'s corresponding edges and nodes to the cluster *(line 7)*, and finally a tour is built on the cluster *(line 8)*. The tour is then assigned to a robot, and the robot repeatedly traverses the tour. In *Node-based Clustering Coverage*, there is no overlap among the tours generated by the algorithm,

Figure 7 illustrates the tours built for eight robots on the sample environment by the *Node-based Clustering Coverage* algorithm.

7.4 Building the tour

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

7.4.1 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.

7.4.2 Chained Lin–Kernighan (CLK)

Chained Lin–Kernighan, a modification of the *Lin–Kernighan* algorithm (Lin and Kernighan 1973), is generally considered to be one of the best heuristic methods for generating *optimal* or *near-optimal* solutions for the *Euclidean Traveling Salesman Problem* (Applegate et al. 2003). 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 (Applegate et al. 2007).

Lin–Kernighan is a *local search* algorithm (Hoos and Sttzle 2004) and a generalization of the *k-opt* algorithm (Chandra et al. 1999). A *k-opt* algorithm explores all the *TSP* tours which can be obtained by removing *k* edges from the original tour and adding *k* different edges such that the resulting tour is feasible. In order to improve the efficiency, Lin and Kernighan introduce a variable *k-opt* algorithm, which adaptively decides at each iteration what value of *k* to use (Lin and Kernighan 1973). Given the computation time limit, the process is repeated by generating new initial tours and applying the *Lin–Kernighan* algorithm to possibly find a tour shorter than the best one thus far. Martin et al. (1991, 1992) suggest that instead of repeatedly starting from new tours, which is inefficient, the alternative is to perturb the *Lin–Kernighan* tour, and then reapply the algorithm. If this leads to a shorter tour, then discard the old tour, and start with the new one. Otherwise, continue with the old tour and perturb it again. The implementation of the *Chained Lin–Kernighan* method which we use in our study is based on the *Concorde* TSP library (Applegate et al. 2003).

Having built the clusters on the *Reduced Graph*, the *Chained Lin–Kernighan* algorithm takes the distance matrix of the guards of each cluster in the initial *VG* or *CDT* graph as an input, and finds the shortest tour passing through all the guards of the cluster. The matrix represents the shortest path distances between all pairs of guards of the cluster in the initial *VG* or *CDT* graph, without taking into account any additional guards other than the ones existing in the cluster. However, the shortest path between the guards in the cluster can include any nodes of the obstacles in the environment. This guarantees that if there is not an overlap among the guards of the generated clusters, then there will not be any overlap among the guards of the tours either after applying the *Chained Lin–Kernighan* algorithm on the clusters. The input to the *Chained Lin–Kernighan* algorithm should be a complete graph and the distance matrix is indicative of a

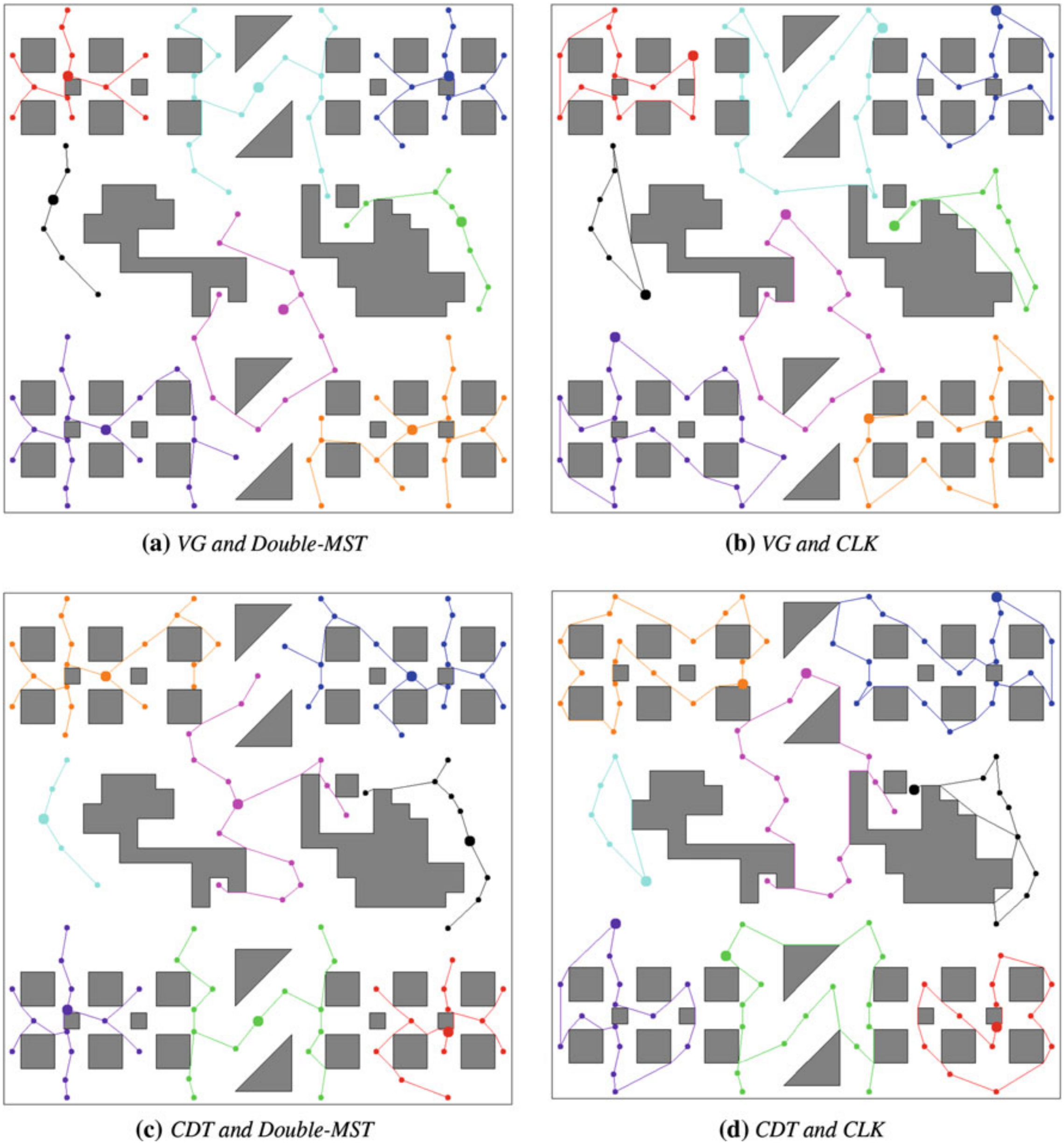


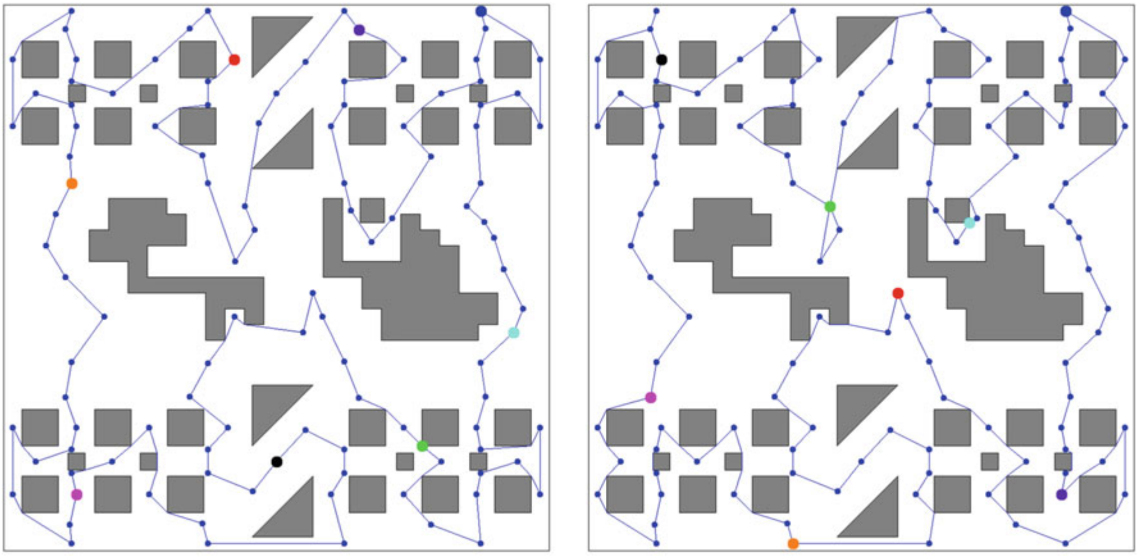
Fig. 7 Tours built by eight robots by *Node-based Clustering Coverage*

complete graph, even though the clusters themselves are not complete.

7.5 Overlap among the tours

Overlap (existence of common guards) among the tours generated by the *Cluster-based* algorithms affects the performance of the algorithms in some cases. The affected cases will be discussed in Sect. 10.

In summary, there is no overlap among the tours built by the *Uninformed Clustering Coverage* and the *Node-based Clustering Coverage* algorithms, using either *Double-MST* or *CLK*. The reason is that the original clusters (before building the tours on them) generated by these two coverage algorithms have no overlap on the guards. However, there may be some unresolved overlaps among the clusters and as a result among the tours generated by *Edge-based Clustering Coverage*, using either *Double-MST* or *CLK*. In the event of an overlap among the tours, the robots assigned to each of the overlapped tours, all have to traverse the common parts among themselves.



(a) *VG*

(b) *CDT*

Fig. 8 Tour built by eight robots by *Cyclic Coverage* algorithm

8 Cyclic algorithm

We develop a method called *Cyclic Coverage (Algorithm* 5) and use it as a benchmark to compare the repeated coverage algorithms. Similar to the *Cluster-based* coverage algorithms, *Cyclic Coverage* locates the guards, and builds the graph (*VG* or *CDT*). However, rather than reducing and partitioning the graph among the robots, it creates a tour passing through all the guards of the graph, using the *Chained Lin–Kernighan* algorithm *(line 2)*. The input of the *Chained Lin–Kernighan* algorithm is the distance matrix of the guards in the *VG* or *CDT* graph. The proposed algorithm then distributes the robots equidistantly around the tour and moves them repeatedly around it *(line 3)*. In this algorithm, in some rare cases, there may be some overlaps among the segments assigned to the robots.

Algorithm 5: Cyclic Coverage Algorithm

```
Input:
$G_{vis-cdt}(V_{vis-cdt}, E_{vis-cdt})$, where $V_{vis-cdt} = SG \bigcup P$   /* VG or CDT */
$SG = \{g_1, g_2, \ldots, g_m\}$  /* Static Guards */
$P = \{p_1, p_2, \ldots, p_n\}$ /* Endpoints of Obstacles*/
$|R|$: Number of Robots

Output:
A tour, $dTour$, distributed among the robots, passing through all the guards of the VG or CDT graph.

1 begin
2     $tour \longleftarrow BuildTour(G_{vis-cdt})$
3     $dTour \longleftarrow DistributeRobots(tour, |R|)$
4     return $dTour$
5 end
```

Cyclic Coverage produces *optimal* or *near-optimal* solutions for the *single robot* case under *Total Path Length* and the *Total Worst Visiting Period*. The notion of *Balance in Workload Distribution* is not defined in this approach, since all the robots traverse the whole tour built on the original graph.

Figure 8 illustrates the tours built for eight robots on the sample environment by the *Cyclic Coverage* algorithm.

9 Complexity analyses of the algorithms

The basic version of the coverage problem with just one robot with unlimited visual range operating in a simple polygon without obstacles has an exact polynomial time solution (Carlsson et al. 1999; Tan 2001). But, extending the problem to support robots' limited visual range, obstacles in the environment, or allowing multiple robots make the corresponding decision problems *NP-hard* (Nilsson 1995). Interestingly, it is impossible even to develop polynomial approximation algorithms, when optimizing each of the metrics (*TPL, TAVP, TWVP, and BWD*) defined for the repeated coverage problem, unless $P = NP$ (Packer 2008). Furthermore, optimizing all the metrics simultaneously is another challenge, because some are mutually conflicting in the coverage scenario.

In summary, the time complexity of the stages of the proposed coverage algorithms are shown in Table 1.

10 Evaluation and experimental simulations

We have developed a simulator to test the algorithms in different scenarios. The simulator can support different numbers

Table 1 Time complexity of different stages of the repeated coverage algorithms

Stages of the algorithms		Time complexity
Locating Guards		$O(n^2 \log n^2)$
Building Graph	Visibility Graph	$O((n+m)^3)$
	Constrained Delaunay Triangulation	$O((n+m)log(n+m))$
Graph Reduction		$O(m^3)$
Clustering the Reduced Graph	Uninformed Clustering	$O(\|R\|(n'+m))$
	Edge-based Clustering	$O(\|R\|mlog(m))$
	Node-based Clustering	$O(\|R\|Im) + O(m^2)$
Cyclic		$O(m^{2.2})$
Building Tours	Double-MST	$O(n'+m)$
	Chained Lin–Kernighan	$O(m^{2.2})$[a]

n number of obstacles' nodes, n' number of obstacles' nodes in the reduced graph, m number of guards, I number of iterations of the algorithm, $|R|$ number of robots

[a] This entry is based on experimental results (Lin and Kernighan 1973; Applegate et al. 2003). The worst case time complexity of the *Chained Lin–Kernighan* algorithm is apparently not available in the literature (Helsgaun 2009)

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 (Tomás and Bajuelos 2004) 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 the following independent variables:

- Visual range of the robots.
- Environment representation (i.e. the combination of the graph representation and the tour building algorithms).

For the environment representation, we have four combinations of the graph representation and the tour building algorithms:

- *Visibility Graph (VG)* and *Double-MST*
- *Visibility Graph (VG)* and *Chained Lin–Kernighan (CLK)*
- *Constrained Delaunay Triangulation (CDT)* and *Double-MST*
- *Constrained Delaunay Triangulation (CDT)* and *Chained Lin–Kernighan (CLK)*

The performance of the coverage algorithms is evaluated based on these criteria:

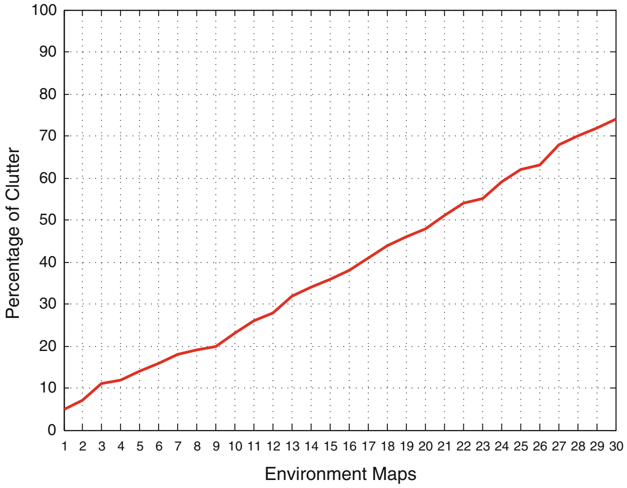
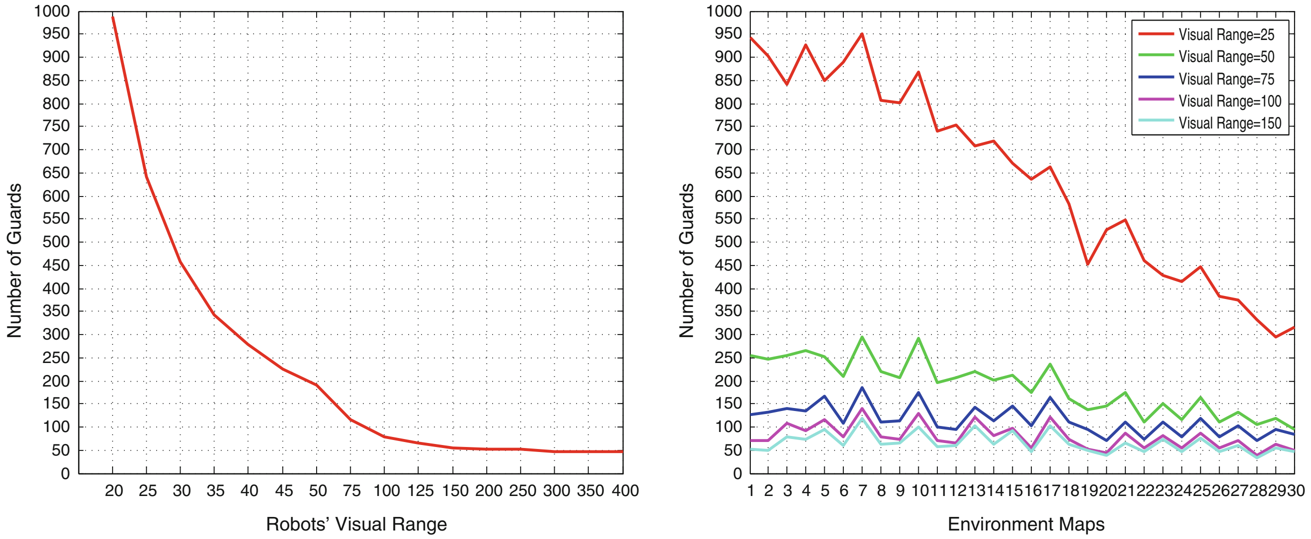


Fig. 9 Percentage of clutter in each map

- *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), 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: $\frac{Obstacles}{Obstacles + FreeSpace}$. Figure 9 represents the percentage of clutter in each of the maps used in the experiments. The size of the environments is 15 m × 15 m.

Figure 10a shows the average number of guards computed on the 30 maps used in the experiments under visual ranges up to 4 m. As shown in the figure, the number of computed



(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*

Fig. 10 Number of guards versus the robots' visual range

guards declines with the increase in the visual range of the robots, and after visual range of 1.5 m, the number of guards becomes essentially fixed. Based on this distribution and some initial experiments to find the proper interval between the visual ranges, so that the difference of the results between two subsequent visual ranges is shown more clearly, we chose visual ranges 0.25 m, 0.5 m, 0.75 m, 1 m and 1.5 m to evaluate the algorithms. Figure 10b shows the number of guards computed on each of the selected maps (sorted from least cluttered to most cluttered) under the chosen visual ranges for 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 results of *Cyclic Coverage* with *Visibility Graph* under *TPL, TAVP*, and *TWVP* as the reference and the results of the coverage algorithms on each map, under *TPL, TAVP*, and *TWVP*, are normalized as ratios to the reference solution.

For optimization metrics *TPL, TAVP*, and *TWVP*, the average values of ratios over all the maps are shown respectively in Figs. 11, 12, and 13 for different numbers of robots $(1, 2, \ldots, 15)$, under the selected visual ranges. Note that the figures do not demonstrate the average of the actual values of the coverage algorithms but the average normalized ratios to the reference solution over all the maps. That is why the values of *Cyclic Coverage* with *Visibility Graph*, shown by the red line in Figs. 11, 12, and 13, is fixed (equal to 1) for different number of robots. For *BWD*, since the values are bounded between 0 and 100, we show the average of the actual values over all the maps in Fig. 14.

In sum, we have collected data from $9000 = 4\ \mathit{CoverageAlgorithms} \times 15\ \mathit{Robots} \times 5\ \mathit{VisualRanges} \times 30\ \mathit{Maps}$ runs of the simulator. The results can be used as a framework for choosing an appropriate combination of repeated coverage algorithm, environment representation, and the robots' visual range based on the particular scenario and the metric to be optimized.

10.1 Running time of the algorithms

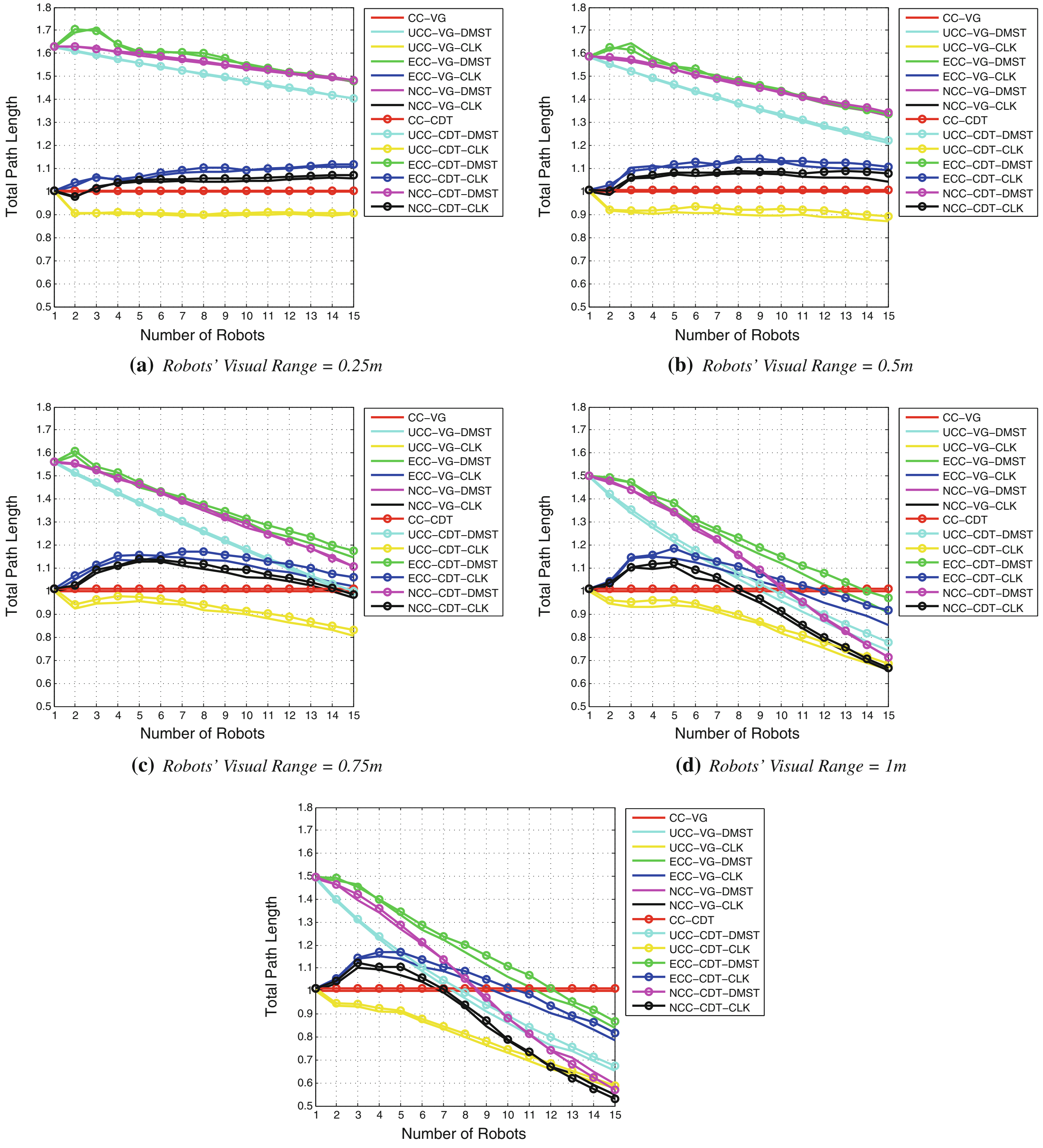
Table 2 shows the average running time of all the stages of building a *Reduced Graph* over all the maps, under the selected visual ranges of the robots. The stages include: computing the guards, building the graph (*VG* or *CDT*), and reducing the graph.

Tables 3, 4, 5, and 6 respectively show the average running time of the *Uninformed Clustering Coverage (UCC)*, *Edge-based Clustering Coverage (ECC)*, *Node-based Clustering Coverage (NCC)*, and the *Cyclic Coverage (CC)* algorithms over all the maps and all number of robots, under the selected visual ranges of the robots and the two tour building algorithms.

The simulations were run on a single-core Pentium 4 (3.2 Ghz) desktop computer, with 3GB of memory.

10.2 Results for Total Path Length

Figure 11 shows the performance of the coverage algorithms for *Total Path Length* on the basis of the pre-determined visual ranges of the robots, and the choice of environment representation.



(a) *Robots' Visual Range = 0.25m*

(b) *Robots' Visual Range = 0.5m*

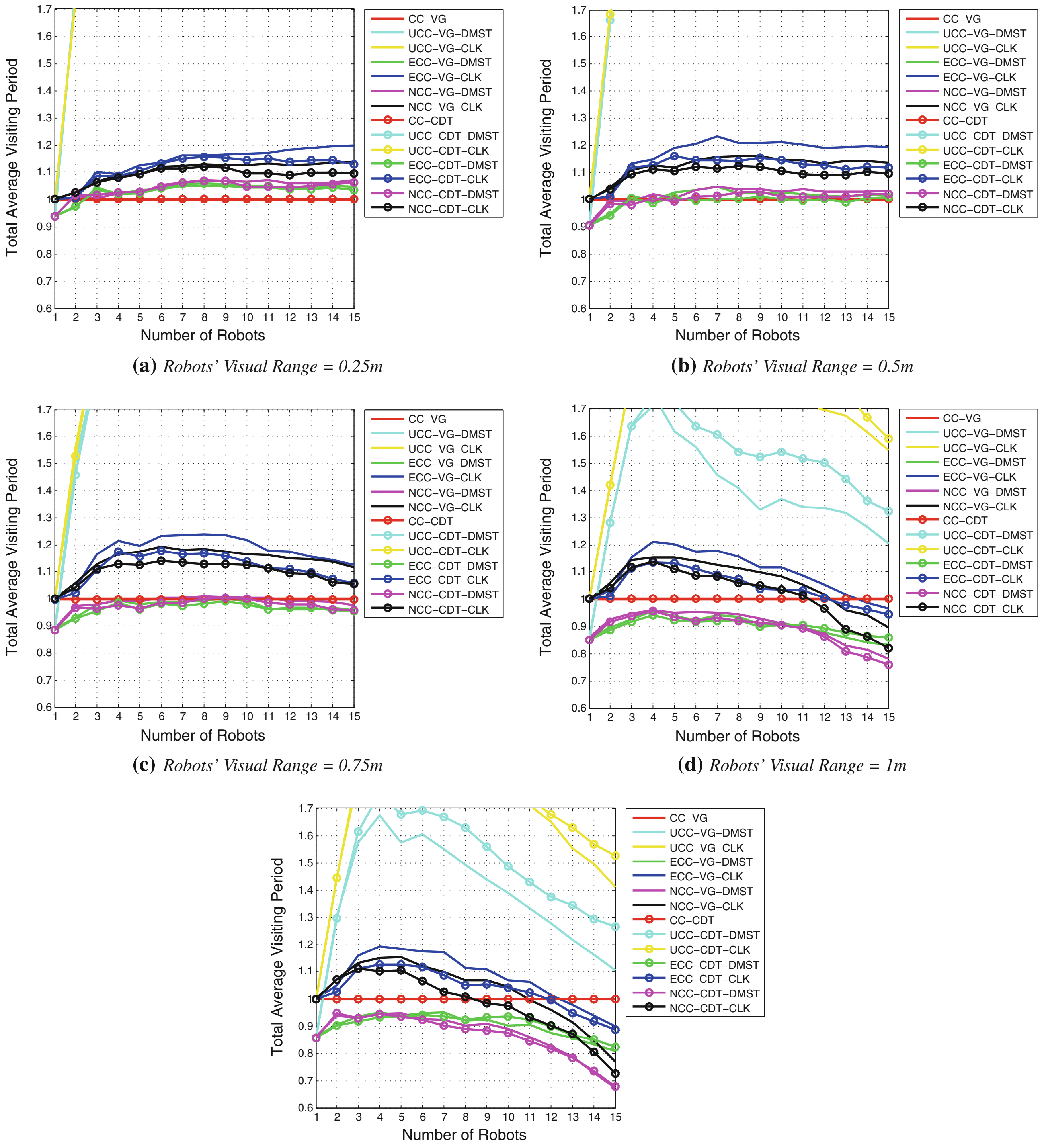
(c) *Robots' Visual Range = 0.75m*

(d) *Robots' Visual Range = 1m*

(e) *Robots' Visual Range = 1.5m*

Fig. 11 Total Path Length. CC = Cyclic Coverage, UCC = Uninformed Clustering Coverage, ECC = Edge-based Clustering Coverage, NCC = Node-based Clustering Coverage, DMST = Double-MST, CLK = Chained Lin–Kernighan, VG = Visibility Graph, CDT = Constrained Delaunay Triangulation

Effect of Robots' Visual Range: In all the tested visual ranges for the robots, at least one of the *Cluster-based* algorithms (i.e. *Uninformed Clustering Coverage*) outperforms *Cyclic Coverage* ($p < 0.01$), and interestingly, as the visual range of the robots increases, there are more *Cluster-based* algorithms dominating *Cyclic Coverage*, especially in the scenarios in which more robots are involved.



(a) *Robots' Visual Range = 0.25m*

(b) *Robots' Visual Range = 0.5m*

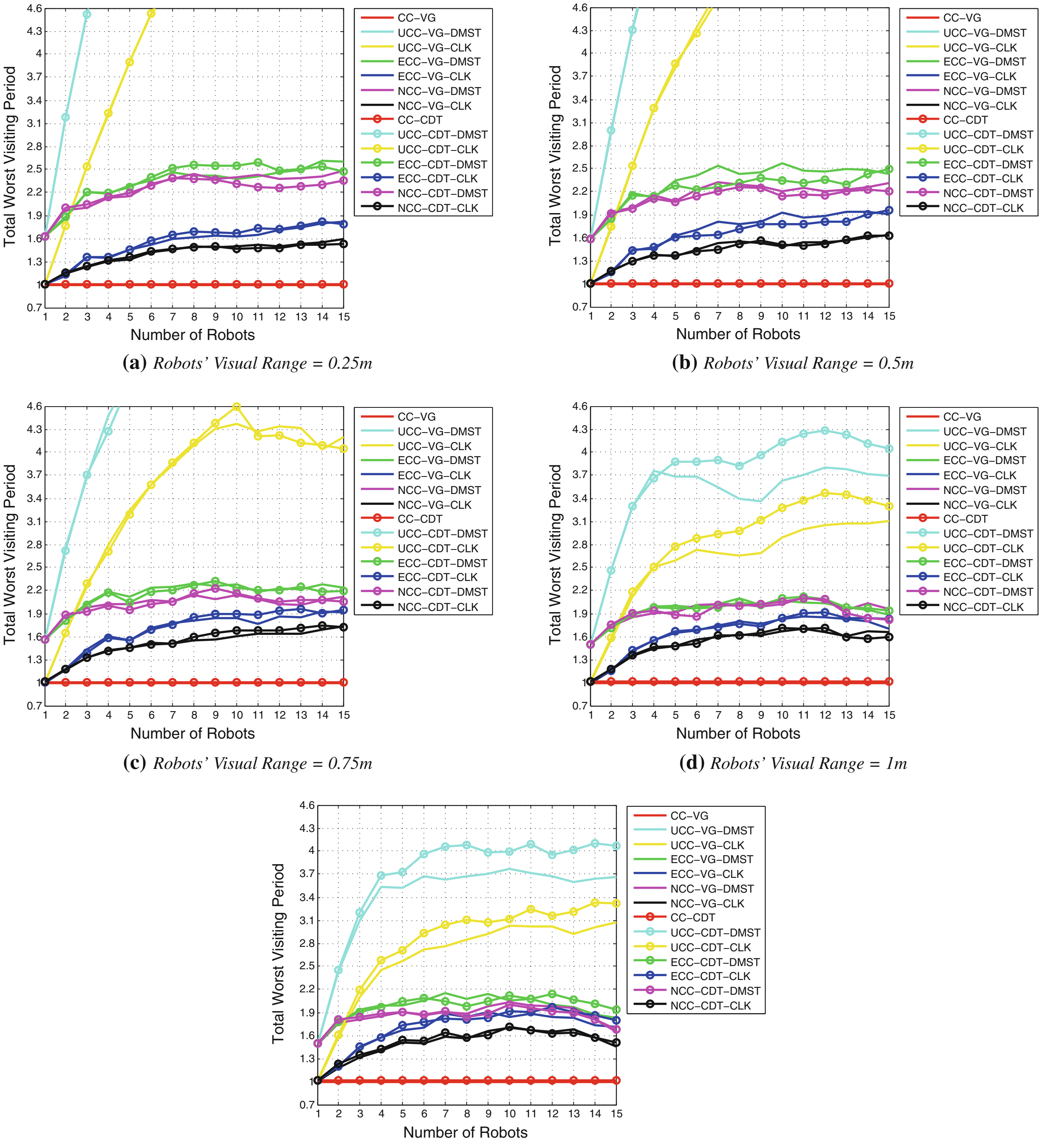
(c) *Robots' Visual Range = 0.75m*

(d) *Robots' Visual Range = 1m*

(e) *Robots' Visual Range = 1.5m*

Fig. 12 *Total Average Visiting Period*. CC = Cyclic Coverage, UCC = Uninformed Clustering Coverage, ECC = Edge-based Clustering Coverage, NCC = Node-based Clustering Coverage, DMST = Double-MST, CLK = Chained Lin–Kernighan, VG = Visibility Graph, CDT = Constrained Delaunay Triangulation

Increasing the robots' visual range leads to increases in the distance between the guards computed in the environments, and as a result increases in the the length of the edges of the graph built on the environment. Subsequently, *Cluster-based* algorithms, while building the clusters, remove the edges in between them, unless the resultant clusters overlap one another, in case of using *Edge-based Clustering Coverage*, and this overlap can not be resolved. Increasing the visual



(a) *Robots' Visual Range = 0.25m*

(b) *Robots' Visual Range = 0.5m*

(c) *Robots' Visual Range = 0.75m*

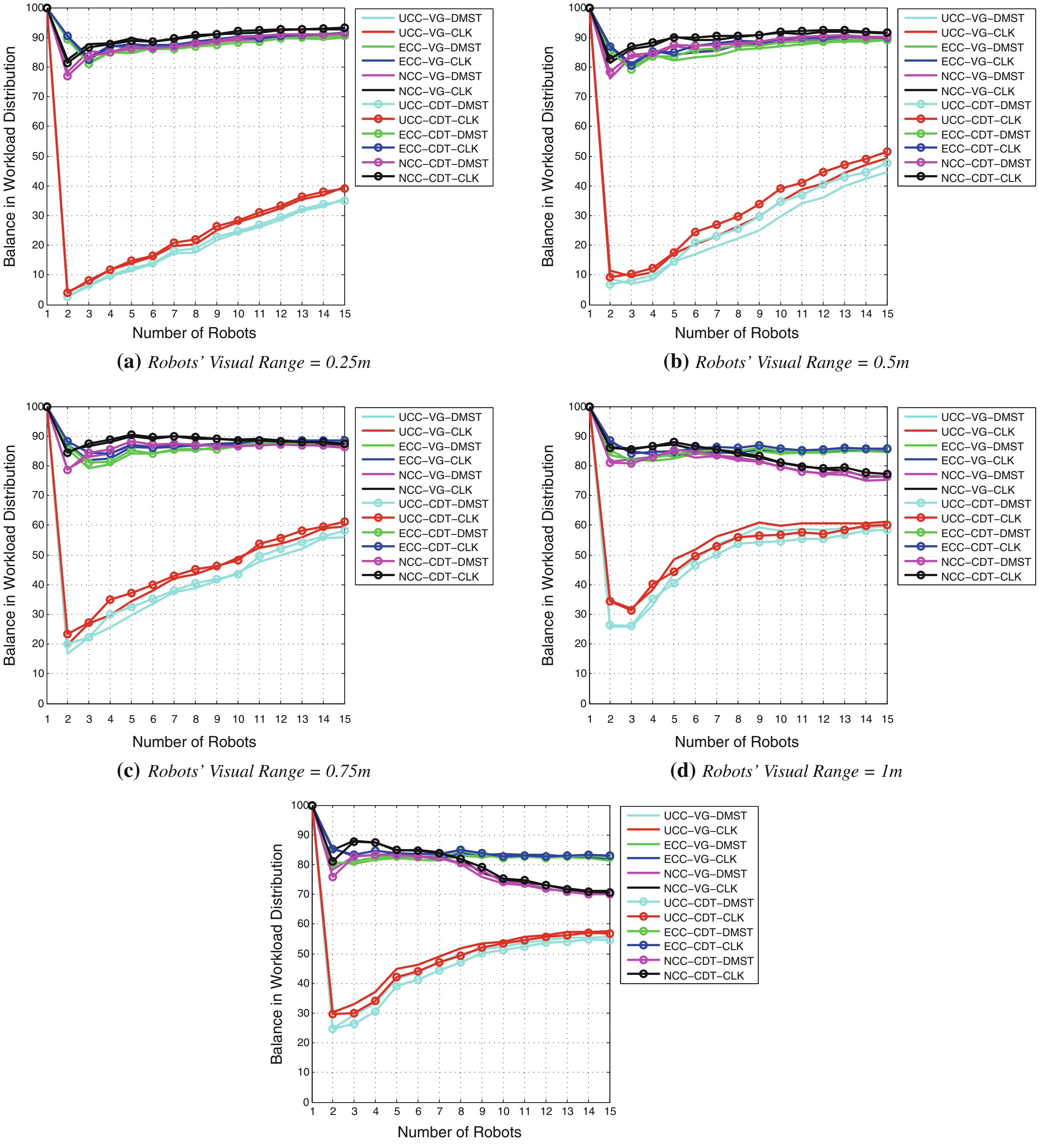
(d) *Robots' Visual Range = 1m*

(e) *Robots' Visual Range = 1.5m*

Fig. 13 *Total Worst Visiting Period.* CC = Cyclic Coverage, UCC = Uninformed Clustering Coverage, ECC = Edge-based Clustering Coverage, NCC = Node-based Clustering Coverage, DMST = Double-MST, CLK = Chained Lin–Kernighan, VG = Visibility Graph, CDT = Constrained Delaunay Triangulation

range and the number of robots leads to, respectively, longer and more edges being removed from the *Reduced Graph*, and consequently improving the *Total Path Length* by the *Cluster-based* algorithms.

Effect of Environment Representation: As for the impact of the *VG* and the *CDT* on the performance of the algorithms, we did not find significant differences between the two under this criterion. However, as for the impact of the tour build-



(a) *Robots' Visual Range = 0.25m*

(b) *Robots' Visual Range = 0.5m*

(c) *Robots' Visual Range = 0.75m*

(d) *Robots' Visual Range = 1m*

(e) *Robots' Visual Range = 1.5m*

Fig. 14 *Balance in Workload Distribution*. UCC = Uninformed Clustering Coverage, ECC = Edge-based Clustering Coverage, NCC = Node-based Clustering Coverage, DMST = Double-MST, CLK = Chained Lin–Kernighan, VG = Visibility Graph, CDT = Constrained Delaunay Triangulation

ing algorithms, all the coverage mechanisms perform significantly better under *CLK* than Double-MST ($p < 0.01$), but this superiority declines with the increase of the robots' visual range and the number of robots. Increasing the robots' visual range leads to fewer guards being computed in the environment, and as a result decreases in the size of the graph and the *Reduced Graph* (in terms of the number of edges) built on the environment. Increasing the number of robots also leads

Table 2 Building Reduced Graph running time (msec)

Visual Range	0.25 m	0.5 m	0.75 m	1 m	1.5 m
Reduced Graph	13352	1494	599	496	492

Table 3 UCC running time (msec)

Visual Range	0.25 m	0.5 m	0.75 m	1 m	1.5 m
UCC-DMST	109	80	71	71	71
UCC-CLK	48628	3060	383	238	232

Table 4 ECC running time (msec)

Visual Range	0.25 m	0.5 m	0.75 m	1 m	1.5 m
ECC-DMST	43725	3306	483	180	169
ECC-CLK	95369	9229	847	336	320

Table 5 NCC running time (msec)

Visual Range	0.25 m	0.5 m	0.75 m	1 m	1.5 m
NCC-DMST	2048	295	102	81	79
NCC-CLK	49598	5434	481	260	240

Table 6 CC running time (msec)

Visual Range	0.25 m	0.5 m	0.75 m	1 m	1.5 m
CC	711	210	129	121	114

to smaller clusters being built out of the *Reduced Graph*, and as the size of the clusters declines, the difference between *Double-MST* and *CLK* built on the clusters declines as well.

Conclusion: Uninformed Clustering Coverage, with no overlap among the tours it builds and removing the longest edges of the *Reduced Graph*, outperforms the other algorithms including *Cyclic Coverage* under *Total Path Length*. *Node-based Clustering Coverage* also dominates *Edge-based Clustering Coverage* ($p < 0.01$). Possible overlaps among the tours generated by *Edge-based Clustering Coverage* dilutes the algorithm's performance under *Total Path Length*.

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

Under this criterion, the coverage algorithms show similar performance under the two graph representation algorithms (i.e. *VG* and *CDT*), but the choice of tour building algorithm significantly affects the coverage approaches, in that using *CLK* lead to shorter paths for the robots compared with *Double-MST*.

10.3 Results for Total Average Visiting Period

Figure 12 shows the performance of the coverage algorithms for *Total Average Visiting Period* on the basis of the predetermined visual ranges of the robots, and the choice of environment representation.

Effect of Robots' Visual Range: As the visual range of the robots increases, the *Cluster-based* algorithms show better performance. For visual ranges of 0.75 m, 1 m, and 1.5 m, both the *Node-based Clustering Coverage* ($p < 0.01$) and the *Edge-based Clustering Coverage* ($p < 0.01$) algorithms dominate *Cyclic Coverage*, especially in the scenarios in which more robots are involved. However, for small visual ranges (i.e. 0.25 m), *Cyclic Coverage* is the best choice of the coverage mission ($p < 0.01$). For visual range of 0.5 m, *Node-based Clustering Coverage*, *Edge-based Clustering Coverage* and *Cyclic Coverage* are all in balance.

Similar to the *Total Path Length*, increasing the visual range and the number of robots lead to respectively longer and more edges being removed from the *Reduced Graph* by the *Cluster-based* algorithms, enabling them to outperform *Cyclic Coverage* under *Total Average Visiting Period*. *Node-based Clustering Coverage* and *Edge-based Clustering Coverage* also outperform *Uninformed Cyclic Coverage* ($p < 0.01$) by building more balanced clusters (see Sect. 10.5), helping improve the *Total Average Visiting Period*.

Effect of Environment Representation: As for the impact of the *VG* and the *CDT* on the performance of the algorithms, we did not find a significant difference between the two under this criterion. However, as for the impact of the tour building algorithms, all the *Cluster-based* algorithms perform better under *Double-MST* than *CLK* ($p < 0.01$). *Double-MST* leverages overlapped paths for the robots, and so improves the performance under *Total Average Visiting Period* compared to *CLK* which discourages overlapped paths.

Conclusion: The *Node-based Clustering Coverage* and the *Edge-based Clustering Coverage* algorithms are the best options when working with robots having medium and large visual ranges, and *Cyclic Coverage* is the choice of the coverage mission for robots with small visual ranges.

Under this criterion, the coverage algorithms show similar performance under the two graph representation algorithms (i.e. *VG* and *CDT*), but 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.

10.4 Results for Total Worst Visiting Period

Figure 13 shows the performance of the coverage algorithms for *Total Worst Visiting Period* on the basis of the pre-determined visual ranges of the robots, and the choice of environment representation.

Effect of Robots' Visual Range: Cyclic Coverage dominates the *Cluster-based* algorithms under *Total Worst Visiting Period* in all the tested visual ranges for the robots ($p < 0.01$). *Node-based Clustering Coverage* also outperforms the other *Cluster-based* algorithms ($p < 0.01$). *Node-based Clustering Coverage* builds more balanced clusters compared to *Uninformed Clustering Coverage* (see Sect. 10.5), and builds clusters with no overlap compared to *Edge-based Clustering Coverage*. More balanced and less overlapped clusters helps improve the *Total Worst Visiting Period*.

Effect of Environment Representation: As for the impact of the *VG* and the *CDT* on the performance of the algorithms, we did not find significant difference between the two under this criterion. However, as for the impact of the tour building algorithms, all the coverage mechanisms perform significantly better under *CLK* than Double-MST ($p < 0.01$), but this superiority declines with the increase of the robots' visual range and the number of robots. Increasing the robots' visual range leads to fewer guards being computed in the environment, and as a result decreases in the size of the graph and the *Reduced Graph* (in terms of the number of edges) built on the environment. Increasing the number of robots also leads to smaller clusters being built out of the *Reduced Graph*, and as the size of the clusters declines, the difference between *Double-MST* and *CLK* built on the clusters declines as well.

Conclusion: Cyclic Coverage is the best choice for minimizing the *Total Worst Visiting Period*, regardless of the visual range of the robots.

Under this criterion, the coverage algorithms show similar performance under the two graph representation algorithms (i.e. *VG* and *CDT*), but the choice of tour building algorithm affects the coverage approaches especially in small visual ranges, such that using *CLK* outperforms *Double-MST* in minimizing the *Total Worst Visiting Period* of the points in the target area.

10.5 Results for Balance in Workload Distribution

Figure 14 shows the performance of the coverage algorithms for *Balance in Workload Distribution* on the basis of the pre-determined visual ranges of the robots, and the choice of environment representation.

Effect of Robots' Visual Range: For small visual ranges (i.e. 0.25 m, 0.50 m), we noticed slight improvements in the *Balance in Workload Distribution* with the increase in the number of robots in the environment. However, for visual range of 0.75 m, this improvement disappears for the *Node-based Clustering Coverage* and the *Edge-based Clustering Coverage* algorithms, and for visual ranges of 1 m and 1.5 m, the *Balance in Workload Distribution* declines in both the algorithms, especially in *Node-based Clustering Coverage* and in scenarios in which more robots are involved.

As discussed before, as the robots' visual range increases, the number of guards being computed in the environment and as a result the size of the built graph (in terms of the number of edges) declines. Hence, increasing the number of robots in the environment also makes building balanced clusters out of a graph smaller in size even more difficult. *Edge-based Clustering Coverage* is less affected by this issue, as it maintains some overlaps among the clusters.

Effect of Environment Representation: As for the impact of the *VG* and the *CDT* on the performance of the algorithms, we did not find any significant difference between the two under this criterion. However, as for the impact of the tour building algorithms, all the *Cluster-based* algorithms perform slightly better under *CLK* than *Double-MST* ($p < 0.01$).

Conclusion: For small visual ranges, *Node-based Clustering Coverage* is the best choice for the coverage mission; however, with the increase of the visual range, *Edge-based Clustering Coverage* dominates the *Node-based Clustering Coverage* algorithm for maximizing the *Balance in Workload Distribution*.

Under this criterion, the coverage algorithms show similar performance under the two graph representation algorithms (i.e. *VG* and *CDT*), but 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.

11 Conclusions and future work

We have addressed 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, namely: *Uninformed Clustering Coverage*, *Edge-based Clustering Coverage*, *Node-based Clustering Coverage*, and a method called *Cyclic Coverage* are introduced for the problem. We evaluated the performance of the repeated coverage algorithms under the effects of the following variables: *(1) Environment Representation*, and the *(2) Robots' Visual Range*. A comprehensive set of performance metrics were considered, including *Total Path Length*, *Total Average Visiting Period*, *Total Worst Visiting Period*, and the *Balance in Workload Distribution*. *Cyclic Coverage*, used as a benchmark to compare the algorithms, produces *optimal* or *near-optimal* solutions for the *single robot* case under some metrics, however, it is not always the best solution when extending the problem to multi-robot scenarios. Moreover, *Cyclic Coverage* may not be an appropriate approach compared with the *Cluster-based* algorithms, when for example, the robots have different speeds, or the target area consists of regions with different priorities for coverage. The results can be used as a framework for choosing an appropriate combination of repeated coverage algorithm, environment representation, and the robots' visual range based on the particular scenario and the metric to be optimized.

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. *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.
6. *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.

Acknowledgments We are grateful to Dr. Will Evans and Dr. David Kirkpatrick of the Department of Computer Science, University of British Columbia for useful discussions on the work reported here. This work was supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

Agmon, N., Hazon, N., & Kaminka, G. A. (2008a). The giving tree: Constructing trees for efficient offline and online multi-robot coverage. *Annals of Mathematics and Artificial Intelligence*, *52*(2–4), 143–168.

Agmon, N., Kraus, S., & Kaminka, G. (2008b). Multi-robot perimeter patrol in adversarial settings. In *Proceedings of the IEEE international conference on robotics and automation, ICRA 2008* (pp. 2339–2345).

Agmon, N., Kraus, S., & Kaminka, G. A. (2009a). Uncertainties in adversarial patrol. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems, AAMAS '09* (Vol. 2, pp. 1267–1268).

Agmon, N., Kraus, S., Kaminka, G. A., & Sadov, V. (2009b). Adversarial uncertainty in multi-robot patrol. In *Proceedings of the 21st international joint conference on artificial intelligence, IJCAI'09* (pp. 1811–1817).

Agmon, N., Sadov, V., Kaminka, G., & Kraus, S. (2008c). The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems* (Vol. 1, pp. 55–62). Japan: International Foundation for Autonomous Agents and Multiagent Systems.

Ahmadi, M., & Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In *Proceedings of the IEEE international conference on robotics and automation, ICRA 2006* (pp. 1724–1729).

Almeida, A., Ramalho, G., Santana, H., Tedesco, P. A., Menezes, T., Corruble, V., et al. (2004). Recent advances on multi-agent patrolling. In *Proceedings of the Brazilian symposium on artificial intelligence, SBIA 2004* (pp. 474–483).

Amstutz, P., Correll, N., & Martinoli, A. (2008). Distributed boundary coverage with a team of networked miniature robots using a robust market-based algorithm. *Annals Mathematics Artificial Intelligence*, *52*(2–4), 307–333.

Applegate, D., Cook, W., & Rohe, A. (2003). Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, *15*, 82–92.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2007). *The traveling salesman problem: A computational Study*. Princeton, NJ: Princeton University Press.

Batalin, M. A., & Sukhatme, G. S. (2002). Spreading out: A local approach to multi-robot coverage. In *Proceedings of 6th international symposium on distributed autonomous robotic systems, DARS 2002* (pp. 373–382).

Batalin, M., & Sukhatme, G. (2003). Efficient exploration without localization. In *Proceedings of the IEEE international conference on robotics and automation, ICRA 2003* (Vol. 2, pp. 2714–2719).

Boardman, M., Edmonds, J., Francis, K., & Clark, C. (2010). Multirobot boundary tracking with phase and workload balancing. In *IEEE/RSJ international conference on intelligent robots and systems, IROS 2010* (pp. 3321–3326).

Burgard, W., Moors, M., Fox, D., Simmons, R., & Thrun, S. (2000). Collaborative multi-robot exploration. In *Proceedings of the IEEE international conference on robotics and automation, ICRA 2000* (pp. 476–481).

Carlsson, S., Jonsson, H., & Nilsson, B. J. (1999). Finding the shortest watchman route in a simple polygon. *Discrete and Computational Geometry*, *22*, 377–402.

Carlsson, S., Nilsson, B. J., & Ntafos, S. C. (1993). Optimum guard covers and m-watchmen routes for restricted polygons. *International Journal of Computational Geometry and Applications*, *3*(1), 85–105.

Chandra, B., Karloff, H., & Tovey, C. (1999). New results on the old k-opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, *28*, 1998–2029.

Chew, L. P. (1987). Constrained delaunay triangulations. In *Proceedings of the symposium on computational geometry, SCG* (pp. 215–222).

Chin, W., & Ntafos, S. (1986). Optimum watchman routes. In *Proceedings of the second annual symposium on computational geometry, SCG 1986* (pp. 24–33). New York, NY: ACM.

Choset, H. (2001). Coverage for robotics—a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, *31*(1–4), 113–126.

Czyzowicz, J., Gasieniec, L., Kosowski, A., & Kranakis, E. (2011). Boundary patrolling by mobile agents with distinct maximal speeds. In *Proceedings of the 19th European conference on algorithms, ESA 2011* (pp. 701–712).

De Berg, M., Cheong, O., Van Kreveld, M., & Overmars, M. (2008). *Computational geometry: Algorithms and applications*. New York: Springer.

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operation Research*, *40*(2), 342–354.

Elmaliach, Y., Agmon, N., & Kaminka, G. A. (2009). Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence*, *57*(3–4), 293–320.

Elmaliach, Y., Shiloni, A., & Kaminka, G. A. (2008). A realistic model of frequency-based multi-robot polyline patrolling. In *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems, AAMAS 2008* (pp. 63–70).

Faigl, J. (2010). Approximate solution of the multiple watchman routes problem with restricted visibility range. *IEEE Transactions on Neural Networks*, *21*(10), 1668–1679.

Faigl, J., Kulich, M., & Přeučil, L. (2011). A sensor placement algorithm for a mobile robot inspection planning. *Journal of Intelligent and Robotic Systems*, *62*, 329–353.

Fazli, P., Davoodi, A., Pasquier, P., & Mackworth, A. K. (2010a). 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* (pp. 5577–5582).

Fazli, P., Davoodi, A., Pasquier, P., & Mackworth, A. K. (2010b). Multi-robot area coverage with limited visibility. In *Proceedings of the 9th international conference on autonomous agents and multiagent systems, AAMAS 2010* (pp. 1501–1502).

Fazli, P., & Mackworth, A. K. (2012a). The effects of communication and visual range on multi-robot repeated boundary coverage. In *proceedings of the 10th IEEE international symposium on safely, security, and rescue robotics, SSRR 2012*. College Station, TX.

Fazli, P., & Mackworth, A. K. (2012b). Multi-robot repeated boundary coverage under uncertainty. In *Proceedings of the IEEE international conference on robotics and biomimetics, ROBIO 2012*. Guangzhou, China.

Gabriely, Y., & Rimon, E. (2001). Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, *31*(1–4), 77–98.

Gasparri, A., Krishnamachari, B., & Sukhatme, G. (2008). A framework for multi-robot node coverage in sensor networks. *Annals of Mathematics and Artificial Intelligence*, *52*(2), 281–305.

Gerkey, B. P., Thrun, S., & Gordon, G. (2006). Visibility-based pursuit-evasion with limited field of view. *International Journal of Robotics Research*, *25*(4), 299–315.

Girard, A., Howell, A., & Hedrick, J. (2004). Border patrol and surveillance missions using multiple unmanned air vehicles. In *IEEE conference on decision and control, CDC 2004* (Vol. 1, pp. 620–625).

Guo, Y., Parker, L. E., & Madhavan, R. (2007). Collaborative robots for infrastructure security applications. In *Mobile robots, studies in computational intelligence* (Vol. 50, pp. 185–200). New York: Springer.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100–108.

Hazon, N., & Kaminka, G. A. (2008). On redundancy, efficiency, and robustness in coverage for multiple robots. *Robotics and Autonomous Systems*, *56*(12), 1102–1114.

Helsgaun, K. (2009). General k-opt submoves for the Lin-Kernighan TSP heuristic. *Mathematical Programming Computation*, *1*(2), 119–163.

Hofner, C., & Schmidt, G. (1994). Path planning and guidance techniques for an autonomous mobile cleaning robot. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, IROS 1994* (pp. 610–617).

Hoos, H., & Sttzle, T. (2004). *Stochastic local search: Foundations applications*. San Francisco: Morgan Kaufmann.

Howard, A. (2006). Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, *25*(12), 1243–1256.

Jennings, J., Whelan, G., & Evans, W. (1997). Cooperative search and rescue with a team of mobile robots. In *Proceedings of 8th international conference on advanced robotics, ICAR 1997* (pp. 193–200).

Jensen, E., Franklin, M., Lahr, S., & Gini, M. (2011). Sustainable multi-robot patrol of an open polyline. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 4792–4797).

Kazazakis, G. D., & Argyros, A. A. (2002). 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* (pp. 2843–2848).

Kurabayashi, D., Ota, J., Arai, T., Ichikawa, S., Koga, S., Asama, H. et al. (1996). Cooperative sweeping by multiple mobile robots with relocating portable obstacles. In *Proceedings of the 1996 IEEE/RSJ international conference on intelligent robots and systems, IROS 1996* (Vol. 3, pp. 1472–1477).

Latombe, J. C. (1991). *Robot motion planning*. Norwell: Kluwer.

LaValle, S., & Hinrichsen, J. (2001). Visibility-based pursuit-evasion: The case of curved environments. *IEEE Transactions on Robotics and Automation*, *17*(2), 196–202.

Lavalle, S. M., Lin, D., Guibas, L. J., Claude Latombe, J., & Motwani, R. (1997). Finding an unpredictable target in a workspace with obstacles. In *Proceedings of the IEEE international conference on robotics and automation, ICRA 1997* (pp. 737–742).

Lee, D. T., & Schachter, B. J. (1980). Two algorithms for constructing a delaunay triangulation. *International Journal of Computer Information Science*, *9*(3), 219–242.

Lin, S., & Kernighan, B. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, *21*(2), 498–516.

Liu, F., & Shen, S. (1999). A method for vehicle routing problem with multiple vehicle types and time windows. *Natural Science Council*, *23*(4), 526–536.

Machado, A., Ramalho, G., Zucker, J. D., & Drogoul, A. (2002). Multi-agent patrolling: An empirical analysis of alternative architectures. In *Proceedings of the 3rd international workshop on multi-agent-based simulation II, MABS 2002* (pp. 155–170). Berlin: Springer.

Martin, O., Otto, S., & Felten, E. (1991). Large-step markov chains for the traveling salesman problem. *Complex Systems*, *5*(3), 299–326.

Martin, O., Otto, S. W., & Felten, E. W. (1992). Large-step markov chains for the tsp incorporating local search heuristics. *Operations Research Letters*, *11*, 219–224.

Mataric, M. J., & Sukhatme, G. S. (2001). Task-allocation and coordination of multiple robots for planetary exploration. In *Proceedings of the 10th international conference on advanced robotics, ICAR 2001* (pp. 61–70).

Nilsson, B. J. (1995). Guarding art galleries—methods for mobile guards. PhD Thesis, Lund University.

Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2000). *Spatial tessellations: Concepts and applications of voronoi diagrams. Series in probability and statistics* (2nd edn.). New York: John Wiley.

O'Rourke, J. (1987). *Art gallery theorems and algorithms*. New York: Oxford University Press.

Packer, E. (2008). Computing multiple watchman routes. In C. C. McGeoch (Ed.), *7th International workshop on experimental algorithms, WEA 2008. Lecture notes in computer science* (Vol. 5038, pp. 114–128). New York: Springer.

Packer, E. (2008). Robust geometric computing and optimal visibility coverage. Ph.D. thesis, Stony Brook, NY, USA (2008).

Rekleitis, I., New, A. P., Rankin, E. S., & Choset, H. (2008). Efficient boustrophedon multi-robot coverage: An algorithmic approach. *Annals of Mathematics and Artificial Intelligence*, *52*(2–4), 109–142.

Sak, T., Wainer, J., & Goldenstein, S. K. (2008). Probabilistic multiagent patrolling. In *Proceedings of the 19th Brazilian symposium on artificial intelligence: Advances in artificial intelligence, SBIA 2008* (pp. 124–133). Berlin: Springer (2008).

Santana, H., Ramalho, G., Corruble, V., & Ratitch, B. (2004). Multi-agent patrolling with reinforcement learning. *International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, *2004*, 1122–1129.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, D. Moors, M., Thrun, S., et al. (2000). Coordination for multi-robot exploration and mapping. In *Proceedings of the national conference on artificial intelligence, AAAI 2000, Austin, TX* (pp. 852–858).

Skiena, S. S. (1998). *The algorithm design manual*. New York: Springer.

Tan, X. (2001). Fast computation of shortest watchman routes in simple polygons. *Information Processing Letter*, *77*(1), 27–33.

Tomás, A., & Bajuelos, A. (2004). Quadratic-time linear-space algorithms for generating orthogonal polygons with a given number of vertices (pp. 117–126). New York: Springer.

Toth, P., & Vigo, D. (2002a). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, *123*(1–3), 487–512.

Toth, P., & Vigo, D. (2002b). *The vehicle routing problem*. Philadelphia, PA: Society for Industrial Mathematics.

Urrutia, J. (2000). Art gallery and illumination problems. In Handbook of computational geometry (pp. 973–1027). Amsterdam: North-Holland (2000).

Vidal, R., Shakernia, O., Kim, H., Shim, D., & Sastry, S. (2002). Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, *18*(5), 662–669.

Williams, K., & Burdick, J. (2006). Multi-robot boundary coverage with plan revision. In *Proceedings of the IEEE international conference on robotics and automation, ICRA 2006* (pp. 1716–1723).

Yamauchi, B. (1998). Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on autonomous agents, AGENTS 1998* (pp. 47–53).

Zalik, B., & Clapworthy, G. J. (1999). A universal trapezoidation algorithm for planar polygons. *Computers & Graphics*, *23*(3), 353–363.

Zheng, X., Jain, S., Koenig, S., & Kempe, D. (2005). Multi-robot forest coverage. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, IROS 2005* (pp. 3852–3857).

Author Biographies

Pooyan Fazli is a PhD Candidate in the Department of Computer Science at the University of British Columbia and a member of the Laboratory for Computational Intelligence. His research interests lie in the areas of Multi-Agent/Robot Systems, Human-Robot Collaboration, Computer/Robot Vision, Machine Learning, and Computational Geometry. He was co-organizer of the 2009 and 2010 Semantic Robot Vision Challenge, and was awarded first place in the 2009 competition as a member of the UBC team. He has extensive experience in building and mentoring RoboCup teams (Soccer and Rescue Robots), and co-founded four student teams which have achieved significant results in national and international competitions. He received his Bachelor's and Master's degrees in Computer Science (Artificial Intelligence) in the Department of Computer Science at Amirkabir University of Technology (Tehran Polytechnic).

Alireza Davoodi is a graduate student in the Department of Computer Science at the University of British Columbia where he works on User Modeling, Intelligent Interfaces, and Adaptive Agents. Previously, he was in the Metacreation, Agents and Multi-Agent Systems Group in the School of Interactive Arts and Technology at Simon Fraser University. He has achieved significant results in national and international RoboCup competitions (Soccer and Rescue Robots). He is also the recipient of the best paper award of ACCDS 2009. He received his Bachelor's and Master's degrees in the Department of Computer Science at Amirkabir University of Technology (Tehran Polytechnic).

Alan K. Mackworth is a Professor of Computer Science and Canada Research Chair in Artificial Intelligence at the University of British Columbia. He was educated at Toronto (B.A.Sc.), Harvard (A.M.) and Sussex (D.Phil.). He works on constraint-based artificial intelligence with applications in vision, robotics, situated agents, assistive technology and sustainability. He is known as a pioneer in the areas of constraint satisfaction, robot soccer, hybrid systems and constraint-based agents. He has authored over 100 papers and co-authored two books: *Computational Intelligence: A Logical Approach (1998) and Artificial Intelligence: Foundations of Computational Agents (2010)*. He was President and Trustee of International Joint Conferences on AI (IJCAI) Inc.; he is on the IJCAI Executive Committee. He has served on many editorial boards and program committees. He was VP and President of the Canadian Society for Computational Studies of Intelligence (CSCSI). He served as President of the Association for the Advancement of Artificial Intelligence (AAAI). He has received the ITAC/NSERC Award for Academic Excellence, the Killam Research Prize, the CSCSI Distinguished Service Award, the Association for Constraint Programming Award for Research Excellence and the Lifetime Achievement Award of the Canadian AI Association (CAIAC). He served as the founding Director of the UBC Laboratory for Computational Intelligence. He is a Fellow of AAAI, the Canadian Institute for Advanced Research and the Royal Society of Canada.