Distributed Multi-Robot Coverage Control of Non-convex Environments with Guarantees

Armin Sadeghi, *Member, IEEE*, Ahmad B. Asghar, *Member, IEEE*, Stephen L. Smith, *Senior Member, IEEE*

***Abstract*— In this paper, we revisit the problem of distributed coverage with a fleet of robots in convex and non-convex environments. In the majority of approaches for this problem, the environment is partitioned, each robot is assigned to a partition and each robot moves toward a location that improves the service quality in its partition. These approaches converge to a locally optimal solution, however, there is no guarantee on the quality of the locally optimal solution with respect to the globally optimal solution. We propose distributed algorithms for the coverage problem in convex continuous, non-convex continuous and metric graphs. We consider sub-additive sensing functions, which capture scenarios where the service quality of a location is proportional to the distance between the robot and the location. For these sensing functions, we provide the first constant factor approximation algorithms for the distributed coverage problem. We also characterize the time and communication complexity of the proposed algorithm and show that the robots converge to a near-optimal solution in polynomial time. The approximation factor guarantees on the solution quality requires twice the conventional communication range, however, the extensive simulation results show that the proposed algorithm provides close to optimal solution with the conventional communication range as well, and outperforms several existing algorithms in convex, non-convex continuous environments and metric graphs.**

***Index Terms*— Multiple and Distributed Robots, Sensor Networks, Coverage Control**

I. Introduction

Multi-robot teams and mobile sensor networks are used in various applications such as environmental monitoring [1], and surveillance [2]. Deploying the robots in an environment in a distributed manner is a well studied multi-robot problem [3]–[7] known as the distributed coverage control problem. In this problem, the robots are capable of sensing or servicing events arriving in the environment according to a known spatial distribution. The cost of sensing an event is a function of the distance from the robot to that event. The objective is to deploy a set of robots to cover the environment such that the total coverage cost of the environment is minimized. The existing distributed algorithms to solve this problem converge to a locally optimal solution with no guarantees on the quality of the solution. In this paper, we provide distributed approximation algorithms to solve the problem in non-convex continuous and discrete environments.

Contributions: Our main contributions are fourfold. First, given an instance of the coverage problem in a non-convex continuous environment, we generate an instance of the coverage problem for a metric graph and we characterize the quality of solution to the discrete problem for the continuous instance (Section III). Second, we propose a distributed coverage control algorithm for the coverage problem on metric graphs (Section IV) and prove that the proposed algorithm is a constant factor approximation algorithm if the communication range is twice the conventional communication range (Section V). The proposed algorithm is the first deterministic constant factor approximation algorithm for the coverage control problem with multiple robots. In Section VI, we characterize the time and communication complexity of the proposed algorithm and show that the algorithm converges to a near-optimal solution in polynomial time and limited communication. Finally, we evaluate the performance of the proposed algorithm in convex, non-convex continuous and metric graphs and show that the proposed algorithm outperforms the state-of-the-art algorithms with an extensive set of experiments with a significant margin even with conventional communication range. We also implement the proposed algorithm in ROS and compare the solution quality to the existing algorithm with a team of six robots in a simulated real-world scenario.

A preliminary version of this work appeared in [8], which proposed a distributed algorithm for the synchronous case that achieved a constant factor approximation along with a preliminary numerical validation. In this paper, we extend the results in [8] in the following ways. First, we provide bounds on the run-time and the number of messages passed between the robots. In contrast to the existing distributed coverage control algorithms in metric graphs, we prove that the proposed algorithm converges to a near-optimal solution in a polynomial time. Second, we extend the detailed description of the algorithm and approximation algorithm analysis. Third, we provide examples of locally optimal solutions where the conventional algorithms can not escape, while the proposed algorithm achieves the global optimal. Fourth, we show that the proposed algorithm can be implemented in an asynchronous manner while preserving the performance guarantees. Fifth, we extend the experimental results in three

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

A. Sadeghi and S. L. Smith are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (a6sadegh@uwaterloo.ca; stephen.smith@uwaterloo.ca)

A. B. Asghar was with the Department of Electrical and Computer Engineering at the University of Waterloo when this research was conducted (abasghar@umd.edu)

ways: 1) we compare to two state-of-the-art algorithms for non-convex environments, 2) we characterize the amount of communication between the robots, and 3) we perform high-fidelity simulations in the robot simulator ROS/Gazebo to benchmark performance in real-world scenarios.

Related Work: Cortes *et al.* [3] introduced the distributed coverage control problem with mobile sensor networks in convex environments, and proposed a distributed algorithm based on Lloyd's descent method which converges to a locally optimal solution. The partition of each robot is defined using Voronoi partitioning, and each robot senses only the events arriving in its partition. Robots communicate only with the robots in their neighboring partitions to implement the algorithm.

Different variations of the coverage control problem are studied in the literature. In [9], the authors consider the case where the distribution of events is not known by the robots and present a method for the robots to approximate this distribution and to decrease the coverage cost with respect to the approximated distribution. The measurement error in the location of the neighboring robots is considered in [10]. The authors in [11] consider heterogeneous robots that have different maximum velocities but consider the coverage problem on a circle instead of a planar environment. Coverage control with communication limitations is studied in [12] and [13].

Building on Lloyd's descent-based algorithm in [3], there have been extensive studies on the coverage control problem in non-convex environments. Authors in [14] augment Lloyd's algorithm with a local path planning algorithm to avoid obstacles in non-convex polygonal environments. Another approach to tackle non-convex environments is to partition the environment by taking the obstacles into consideration. For instance, in [15] the environment is partitioned considering geodesic distances, and in [16], [17], the partitions are constructed based on the visibility of the robots in the presence of obstacles. In [18], [19], the authors propose a mapping from the non-convex environment to a convex region and utilize the Lloyd's based algorithm [3] to converge to a locally optimal solution in the mapped convex environment. Then, the solution is mapped back to the original environment. In contrast to the aforementioned approaches, we create a discrete representation of the non-convex environment, and solve the corresponding coverage problem on the discrete environment and provide guarantees on the solution quality.

The approach of converting a continuous non-convex environment to a discrete environment is used in [4], [20], [21]. The proposed method in [21] represents the non-convex environment as a graph. The continuous environment is partitioned using the geodesic distances. Each robot moves towards the closest vertex to the centroid of its partition. We also represent the non-convex environment as a graph but are able to characterize the cost of the solution obtained from the discretized environment in the corresponding continuous environment as a function of the sampling density.

The problem of discretized coverage control is the k-median problem [22]–[24] where the objective is to place k robots on vertices of the graph to minimize the total service time. A centralized approximation algorithm was presented for the k-median problem in [22], and the analysis of our distributed approximation algorithm leverages this centralized approximation algorithm. The authors in [24] consider the k-median problem with mobile robots and provide an approximation algorithm for the objective of minimizing a linear combination of the relocation cost of the mobile robots and the expected service time of the demands. In [25], we present an approximation algorithm for the mobile facility location problem with a time horizon, where the demands arrive sequentially over time. The authors in [26] present a distributed algorithm that converges to a local optimum. Their algorithm requires the robots to know the information of the neighbors of their neighbors. This assumption is analogous to the communication range in this paper, however, we establish approximation guarantees on the solution quality of the proposed algorithm. Authors in [27] provide a randomized distributed algorithm for the k-median problem with constant factor approximation in Euclidean environments. In this paper, we consider more general non-convex environments and provide a deterministic approximation algorithm. In [28], the authors consider the problem of maximizing the number of vertices covered by the robots with limited sensor range in undirected graphs with a game-theoretic approach. In contrast, we are considering the problem of maximizing the service quality.

II. Continuous and Discrete Coverage Problems

We begin by reviewing the coverage problems in both continuous [3] and discrete environments [26].

A. Continuous Environment

Consider m mobile robots in a compact environment with obstacles and let $\mathcal{X}$ be the obstacle free subset of the environment. There is an event distribution $\phi : \mathcal{X} \to \mathbb{R}_+$ defined over the environment. Let $d(p,q)$ be the length of the shortest *obstacle free* path between two locations p and q in $\mathcal{X}$. The sensing cost of an event at location p by a robot at q is a strictly non-decreasing function $f : \mathbb{R}_+ \to \mathbb{R}_+$ of $d(p,q)$. Following the non-convex problem formulation in [14], which extends the original formulation in [3], the continuous problem is defined as the problem of finding the set of locations in the environment for the robots that minimizes the sensing cost of the events, i.e.,

$$\min_{Q \in \mathcal{X}^m} \mathcal{H}(\mathcal{Q}) = \min_{Q \in \mathcal{X}^m} \int_{\mathcal{X}} \min_{q_i \in Q} f(d(p,q_i))\phi(p)dp. \tag{1}$$

Without loss of generality, in the rest of the paper we assume that $\int_{\mathcal{X}} \phi(p)dp = 1$. Observe that the best sensing cost for an event is provided by the closest robot to that event location. Then for a given configuration Q, we partition the environment into Voronoi subsets as follows:

$$V_i(Q) = \{p \in \mathcal{X} | d(p,q_i) \le d(p,q_j)\ \forall q_j \in Q \setminus \{q_i\}\}.$$

The robots move according to some dynamics $\dot{q}_i = g(q_i, u_i)$ where the computation of the shortest path between two configurations of the robot is tractable. Typically first order dynamics $g(r_i, u_i) = u_i$ is considered for the robots in coverage control literature [3]. We are interested in the distributed version of the coverage problem, where the robots have local information on the other robots and each robot computes its control input locally.

B. Discrete Environment

Consider a metric graph $G = (\mathcal{S}, E, c)$ where $\mathcal{S}$ is the vertex set, E is the set of edges between the vertices, c is the metric edge cost and let w be the weight on the vertices. Given a team of m robots, the coverage problem on graph G, is the problem of finding a set of m locations to optimally cover the vertices of G, i.e., minimize $\mathcal{D}(Q) = \sum_{v\in\mathcal{S}} \min_{q\in Q} w(v)c(v,q)$. For a given configuration $Q \in \mathcal{S}^m$, we can partition the vertices into m subsets

$$W_i(Q) = \{u \in \mathcal{S} | c(u, q_i) < c(u, q_j)\ \forall q_j \in Q \setminus \{q_i\}\}.$$

If there exists a vertex that has equal distance to two or more robots in Q, then the vertex is assigned to the robot with smaller unique identifier (UID). Robots travel on the edges of the graph and the control input to a robot is a sequence of edges leading to its destination vertex.

Centralized Approximation Algorithm: The centralized version of this problem is a well-known NP-hard problem called the k-median problem [22]. The best known approximation algorithm for this problem on metric graphs is a centralized local search algorithm which provides solutions within a constant factor of the optimal [24]. Starting from a configuration Q, the centralized local search algorithm swaps p vertices in Q at a time with a subset of p vertices in $\mathcal{S} \setminus Q$. If the new configuration improves the coverage by at least some $\epsilon_0 > 0$, then we call this move a valid local move. The procedure terminates if there are no more valid swaps improving the total sensing cost. We will refer to this local search algorithm as CENTRALIZEDALG in the rest of the paper. The solution obtained from CENTRALIZEDALG is within $3 + 2/p + o(\epsilon_0)$ of the globally optimal solution.

We focus on the distributed version of this problem introduced in [26] where the robots use local information to compute their control input. Next, we establish the connection between the continuous and discrete coverage problems.

III. From a Continuous to a Discrete Problem

To establish a connection between the coverage problem in continuous and discrete environments, we first convert the continuous coverage problem to a coverage problem in a discrete environment through sampling of the environment.

Let $\mathcal{S}$ be the set of samples of $\mathcal{X}$ with dispersion ζ [29], where ζ is the maximum distance of any point in the environment $\mathcal{X}$ from the closest point in $\mathcal{S}$, i.e., $\zeta = \max_{p\in\mathcal{X}} \min_{u\in\mathcal{S}} d(p,u)$, (See Figure 1). We construct a metric graph $G = (\mathcal{S}, E, c)$ on sampled locations $\mathcal{S}$, where E is the edge set and c is a function assigning costs to the edges of the graph. The cost of an edge between two sampled locations $u, v \in \mathcal{S}$ is $c(u,v) = f(d(u,v))$. Let $\sigma(v)$ for $v \in \mathcal{S}$ be the points in $\mathcal{X}$ closer to v than other samples in $\mathcal{S}$, i.e.,

$$\sigma(v) = \{p \in \mathcal{X} | d(p,v) \le d(p,u)\ \forall u \in \mathcal{S} \setminus \{v\}\}.$$

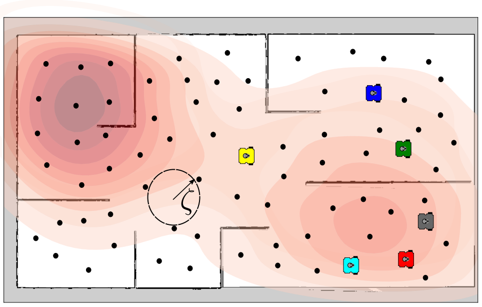


Fig. 1: Sampled locations in an environment with dispersion ζ.

With a slight abuse of notation, let $\sigma^{-1}(p)$ be the closest sample in $\mathcal{S}$ to $p \in \mathcal{X}$. The function $w : \mathcal{S} \to \mathbb{R}_+$ assigning weights to the vertices of the graph is $w(v) = \int_{p\in\sigma(v)} \phi(p)dp$. We assume the following property on the sensing function.

Assumption III.1 (Subadditivity of sensing function): We assume that the sensing cost function f is a sub-additive function, i.e., $f(d(p,u) + d(u,v)) \le f(d(p,u)) + f(d(u,v))$. For instance $f(x) = \sqrt{x}$ and $f(x) = x$ are sub-additive functions.

Remark III.2: Sub-additive sensing functions arise in several applications such as dynamic vehicle routing problems [30], [31] and facility location problems [32], where the sensing cost is the distance traveled by the clients. However, several works [3], [33] consider the square of the distance between the robot and the sensed location (partially due its nice properties in analysis, such as differentiability), which does not satisfy Assumption III.1.

Due to Assumption III.1, the cost on the edges of the graph satisfies the triangle inequality, i.e., for all u, v, z in $\mathcal{S}$

$$\begin{aligned} c(u,v) = f(d(u,v)) &\le f(d(u,z) + d(z,v)) \\ &\le f(d(u,z)) + f(d(z,v)) = c(u,z) + c(z,v). \end{aligned}$$

The following result establishes a connection between the sensing costs of an approximate solution to the discrete coverage problem and the optimal coverage in continuous environment.

Theorem III.3: Consider a continuous coverage problem on environment $\mathcal{X}$ with optimal solution S^*, and its corresponding discrete instance obtained through the set of samples $\mathcal{S}$ with dispersion ζ. Then if Q is the solution obtained from an α-approximation algorithm for the discrete coverage problem instance, the sensing cost of Q on the corresponding continuous problem is $\mathcal{H}(Q) \le \alpha\mathcal{H}(S^*) + o(f(\zeta))$.

Proof: We have,

$$\begin{aligned} \mathcal{H}(Q) &= \int_{\mathcal{X}} \min_{q_i\in Q} f(d(q_i,p))\phi(p)dp \\ &\le \int_{\mathcal{X}} \min_{q_i\in Q} f(d(q_i,\sigma^{-1}(p)) + d(\sigma^{-1}(p),p))\phi(p)dp \\ &\le \int_{\mathcal{X}} \min_{q_i\in Q} [f(d(q_i,\sigma^{-1}(p))) + f(d(\sigma^{-1}(p),p))]\phi(p)dp \end{aligned}$$

Continuing we get

$$\begin{aligned}\mathcal{H}(Q) &\leq \int_{\mathcal{X}} \min_{q_i \in Q} f(d(q_i, \sigma^{-1}(p)))\phi(p)dp \\ &\quad + \int_{\mathcal{X}} f(d(\sigma^{-1}(p), p))\phi(p)dp \\ &\leq \mathcal{D}(Q) + f(\zeta)\int_{\mathcal{X}} \phi(p)dp = \mathcal{D}(Q) + f(\zeta). \qquad (2)\end{aligned}$$

where the first inequality is due to the triangle inequality and the second inequality is due to Assumption III.1. Let $S^* = \{q_i^* | i \in [m]\}$ be the optimal configuration of the continuous problem, and S_G^* be the configuration constructed by moving each robot location in S^* to the closest sampled location in $\mathcal{S}$. Also note that,

$$\begin{aligned}\mathcal{D}(S_G^*) &= \sum_{q_i \in S_G^*} \sum_{u \in W_i} c(u, q_i)w(u) \\ &= \sum_{i \in [m]} \sum_{u \in W_i} c(u, q_i) \int_{p \in \sigma(u)} \phi(p)dp \\ &= \sum_{q_i \in S_G^*} \sum_{u \in W_i} \int_{p \in \sigma(u)} f(d(u, q_i))\phi(p)dp \\ &\leq \sum_{q_i \in S_G^*} \sum_{u \in W_i} \int_{p \in \sigma(u)} \min_{q_j \in S_G^*} f(d(q_j, p)) \\ &\quad + \sum_{q_i \in S_G^*} \sum_{u \in W_i} [2f(d(u, p)) + f(d(p, u))]\phi(p)dp \\ &\leq \mathcal{H}(S_G^*) + 3f(\zeta), \qquad (3)\end{aligned}$$

where the first inequality is due to triangle inequality and Assumption III.1. Furthermore, we have,

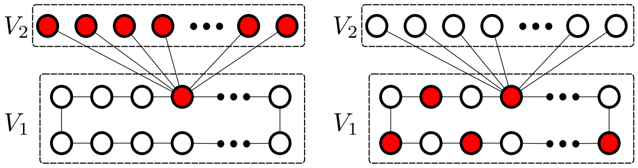
$$\begin{aligned}&\mathcal{H}(S_G^*) = \sum_{q_i \in S_G^*} \int_{V_i(S_G^*)} f(q_i, p)\phi(p)dp \\ &\leq \sum_{q_i \in S_G^*} \int_{V_i(S^*)} f(q_i, p)\phi(p)dp \\ &\leq \sum_{q_i \in S_G^*} \int_{V_i(S^*)} f(d(q_i^*, p))\phi(p)dp + \int_{\mathcal{X}} f(d(q_i, q_i^*))\phi(p)dp \\ &\leq \mathcal{H}(S^*) + f(\zeta), \qquad (4)\end{aligned}$$

where the second inequality is due to triangle inequality and Assumption III.1. Let Q_G^* be the optimal solution to the discrete coverage problem on graph G, then $D(Q) \leq \alpha\mathcal{D}(Q_G^*) \leq \alpha\mathcal{D}(S_G^*)$. Therefore, by Equations (2), (3) and (4), we have, $\mathcal{H}(Q) \leq \alpha\mathcal{H}(S^*) + (4\alpha + 1)f(\zeta)$. ■

A 5-approximation algorithm for the centralized coverage on metric graphs is provided in [22]. In the following section, we provide the first distributed approximation algorithm for the coverage in metric graphs.

IV. Distributed Algorithm

In distributed coverage control algorithms for continuous environments, and their adaptations to discrete environments, the algorithm drives each robot to the position inside its partition such that the sensing cost of its partition is minimized, i.e., the centroid of its Voronoi cell in the continuous problem. Although these algorithms converge to locally optimal solutions, there are no global guarantees on the quality of the solution. The following example provides a graph construction where such "move to centroid" algorithms perform poorly.



(a) Locally optimal configuration (b) Globally optimal configuration

Fig. 2: Example environment with $3n + 1$ vertices and $n + 1$ robots

Example IV.1: Consider the environment shown in Figure 2 with $3n + 1$ vertices, $n + 1$ robots and unit costs for the shown edges. We consider the metric completion of the shown graph. The vertices are partitioned into two subsets: 1) V_1 with $2n + 1$ vertices and unit weights on the vertices and 2) V_2 with n vertices of weights ϵ for some $0 < \epsilon \ll 1$. The highlighted vertices show the configuration of the robots. The configuration in Figure 2(a) is a locally optimal solution under the move to centroid control law with global cost of $n(n + 1)$. However, the configuration shown in Figure 2(b) provides a global cost of $n + n\epsilon$. Therefore, the locally optimal solution provided by the move to centroid algorithm provides a solution with cost at least $\frac{n+1}{1+\epsilon}$ of the optimal cost on the shown instance. •

Observe that Example IV.1 depicts a scenario where the locally optimal solution reached by conventional methods is at least a factor of n of the optimal solution. The conventional method may get stuck in local solutions far from the optimal in more common scenarios. For instance, in scenarios where two areas of an environment are connected by a corridor, and a robot is positioned in the corridor. The robot can block the movement of robots between the two areas, preventing improvement to coverage using conventional methods.

A. High-level Idea

The basic idea of our distributed coverage algorithm is to imitate the local-search algorithm for the k-median problem (See Section II-B), namely CentralizedAlg, in a distributed manner. The challenge in performing a local move in the distributed manner is that the robots are only aware of the partitions of their neighboring robots, therefore, the effect of a local move on the global objective is not known to the robots. However, we break down a local move in CentralizedAlg into a sequence of moves between neighbors. Let robot j with position q_j and neighbors $\mathcal{N}(j)$ be the closest robot to vertex v. Then a local move of CentralizedAlg swapping the position q_i of robot i with vertex v is equivalent to a sequence of swaps inside Q between the neighboring robots and a move from q_j to v. Figure 3 shows an example of a local move in the centralized algorithm performed by a sequence of local moves.

For this distributed coverage algorithm, we define the minimum communication range and neighboring robots as follows:

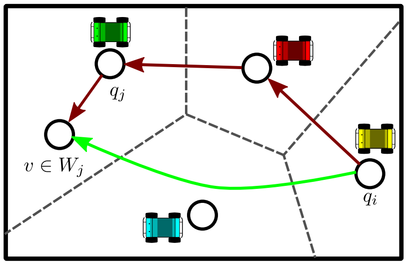


Fig. 3: Local move in the centralized algorithm (green) and its equivalent sequence of moves in the distributed algorithm (red).

Definition IV.2 (Neighbor robots): Given a configuration Q, the set of neighbors of robot i is defined as

$$\mathcal{N}(i) = \{j \in [m] | d(q_i, q_j) \leq 4d_{ij}^{\max}\}$$

where V_i is the Voronoi partition of robot i in the continuous environment and

$$d_{ij}^{\max} = \max\{\max_{p \in V_i} d(p, q_i), \max_{p \in V_j} d(p, q_j)\}.$$

Remark IV.3: The conventional definition of neighbors in the literature [3] is that two robots are neighbors if the intersection of their Voronoi cell boundaries is not empty. Therefore, the distance of two neighboring robots i and j can be $2\max\{\max_{p \in V_i} d(q_i, p), \max_{p \in V_j} d(q_j, p)\}$. In [26], authors show that in environments represented as graphs, with the conventional communication range, a move inside a robot's partition might change the partition of the neighbors of their neighbors. Therefore, they assume that the robots communicate with the neighbors of their neighbors, which is analogous to twice the communication range needed to implement Lloyd's based algorithms in continuous environments. In Section VII, we evaluate the performance of the algorithm with the conventional and our definition of neighbors.

We extend the definition of neighboring robots in a continuous environment to capture the cases where the graph instance for the discrete coverage problem is given and the underlying continuous environment is unknown. Consider a graph instance $G = (V, E, c)$, then for each edge $(u, v) \in E$ we add a dummy vertex z with zero weight and replace edge (u, v) by two edges (u, z) and (z, v) such that $c(u, v) = 2c(u, z) = 2c(z, v)$. We let $c(u, v)$ for $(u, v) \notin E$ be the length of the shortest path in G between u, v. Let W_i be the partition of robot i, in the resulting graph. Then the equivalent definition of the neighboring robots is given as follows:

Definition IV.4 (Neighbors in graphs): Given a configuration Q, the set of neighbors of robot i is defined as

$$\mathcal{N}(i) = \{j \in [m] | c(q_i, q_j) \leq 4c_{ij}^{\max}\},$$

where $c_{ij}^{\max} = \max\{\max_{p \in W_i} c(p, q_i), \max_{p \in W_j} c(p, q_j)\}\}$.

B. Detailed Description

We are now ready to provide a detailed description of the proposed algorithm.

Algorithm Framework (Algorithm 1): For the ease of presentation, we provide a description of the algorithm in which the robots perform local moves sequentially. Each robot is assigned a unique identifier UID. Starting from an active robot, say robot i, the robot will make the possible local moves using Algorithm 2. If it can not make a local move, the robot will become inactive and will send a completion message to the neighboring robots via SENDCOMPLETIONMESSAGE (line 6 of Algorithm 1). After execution of the local move by a robot, the robot becomes inactive. The next active robot to execute the local move can be selected in a distributed manner using a token passing algorithm [34], or any other method that ensures each robot gets a turn at making a local move. Algorithm 1 is repeated iteratively until none of the robots find a valid local move and all of the robots have become inactive.

Algorithm 1 DISTRIBUTEDCOVERAGEALGORITHM

```
1:  Each robot sets itself to active
2:  while there exists an active robot do
3:      for any active robot $i \in \{1, \ldots, m\}$ do
4:          if $\sum_{u \in W_i} c(u, q_i) > 0$ then
5:              LOCALMOVE($i$)
6:              SENDCOMPLETIONMESSAGE()
7:          end if
8:          Robot $i$ deactivates
9:      end for
10: end while
```

Local Move of Robot i (Algorithm 2): At an iteration of Algorithm 2, let the current configuration of robots be given by $Q = \{q_1, \ldots, q_m\}$ where the vertices in the partition of robot i are given by W_i. The robot i considers moving to a vertex $v \in W_i$ from its current vertex q_i. This move can only change the sensing cost of the vertices in the neighboring robots' partitions (See Lemma V.1 in Section V). Hence, the robot i can calculate the new neighboring partitions after a potential move to v. In line 2 of Algorithm 2, robot i calculates the change in local objective δ_v due to this move for all $v \in W_i$. Since only robot i is executing Algorithm 2 at the current time, δ_v also represents the change in the global objective function. If $\min_{v \in W_i} \delta_v \leq -\epsilon_0$, robot i moves to $q_i' = \arg\min \delta_v$ and the iteration terminates (local move type 1). If there is no valid local move of type 1, then robot i calculates the change in local objective if it moves to v and a new robot appears at q_i, i.e.,

$$\rho_v = \sum_{j \in \mathcal{N}(i)} \sum_{u \in R_j(v)} w(u)[c(u, v) - c(u, q_j)], \tag{5}$$

where $R_j(v) = \{u \in W_j | c(u, q_j) > c(u, v)\}$ represents the vertices in the partition of robot j that are closer to v than the robot j at q_j. Robot i then passes the message with the set $\Gamma_i = \{\rho_v | v \in W_i\}$ and a counter set to 1 to all its neighbors (line 7 of Algorithm 2) and waits for a response (line 8 of Algorithm 2). If the response is a rejection from all the neighbors, Algorithm 2 terminates. Otherwise it selects the acceptance message with the largest change in the objective and moves to the corresponding vertex. It also sends an acknowledgement message to the neighbor k whose message was selected so that robot k can move to q_i.

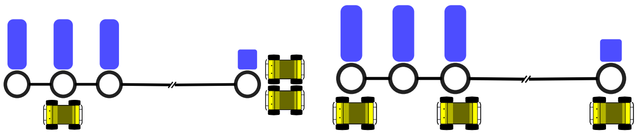
Response of Other Robots (Algorithm 3): When a robot k receives messages from its neighbors, it follows Algorithm 3.

Algorithm 2 LOCALMOVE(i)

```
 1: while ∃ a local move do
 2:     Calculate $\delta_v$ for all vertices in $W_i$
 3:     if $\min_v \delta_v \leq -\epsilon_0$ then
 4:         Move to $v$
 5:     else
 6:         Calculate $\Gamma_i = \{\rho_v | v \in W_i\}$        ▷ Equation (5)
 7:         SENDMESSAGE($\Gamma_i$, 1)
 8:         RECEIVEMESSAGE()
 9:         SENDACKNOWLEDGEMENT()
10:     end if
11: end while
```

Algorithm 3 RECEIVER

```
    Input: message = ($\Gamma_i$, MessageCounter)
 1: if MessageCounter = 1 then
 2:     Calculate $\ell_v$ for all $v$ in the message        ▷ Equation (6)
 3:     if $\exists v$ in the message with $\rho_v + \ell_v \leq -\epsilon_0$ then
 4:         send acceptance message to parent
 5:     else
 6:         SENDMESSAGE($\Gamma_i$, MessageCounter + 1)
 7:     end if
 8: else
 9:     Calculate $\ell$                                      ▷ Equation (7)
10:     if $\exists v$ in the message with $\rho_v + \ell \leq -\epsilon_0$ then
11:         send acceptance message to parent
12:     else
13:         SENDMESSAGE($\Gamma_i$, MessageCounter + 1)
14:     end if
15: end if
16: RECEIVEMESSAGE()
```



(a) A test environment for algorithm in [26]

(b) Final configuration with proposed algorithm

Fig. 4: A discrete test environment and the final configurations of algorithm in [26] and the proposed algorithm

Since messages from only one sender robot are propagating through the system at any time, it can select the message with the smallest counter value if it receives messages from multiple neighbors. The neighbor who sent the message with the smallest counter value is called the parent of robot k. It sends back a rejection message to all other neighbors. If the counter value of the message was one, it means that the message originated from its neighboring robot, say i. Then robot k calculates the change in the sensing costs of the vertices in $W_k \setminus R_k(v)$ if it moves to vertex q_i and robot i moves to vertex v resulting in configuration $Q' = \{q_j | j \in \mathcal{N}(k)\} \cup \{v\}$, i.e.,

$$\ell_v = \sum_{u \in W_k \setminus R_k(v)} \min_{q \in Q'} [c(u,q) - c(u,q_k)] w(u), \tag{6}$$

If $\min_v(\rho_v + \ell_v) \leq -\epsilon_0$, robot k decides to move to q_i and sends an acceptance message to robot i with the vertex $\arg\min_v \rho_v + \ell_v$ and the change associated with this move. Otherwise, it increments the counter and sends the message with Γ_i to its neighbors. If the counter value in a message is greater than one, robot k calculates ℓ as follows:

$$\ell = \sum_{u \in W_k} \min_{j \in \mathcal{N}(k)} [c(u,q_j) - c(u,q_k)] w(u) \tag{7}$$

If $\min_v(\rho_v + \ell) \leq -\epsilon_0$, robot k sends an acceptance message back to its parent. Otherwise, it increments the counter and sends the message to its neighbors.

In function RECEIVEMESSAGE in line 8 of Algorithm 2 and line 16 of Algorithm 3, if any robot receives at least one acceptance message from its neighbors, it passes the message with lowest increase in sensing cost value to its parent. If it receives rejection messages from all its neighbors, it sends back a rejection message to its parent. Robot i selects the move with maximum improvement in the sensing cost and sends back an acknowledgment using SENDACKNOWLEDGEMENT to the accepted messages. Robots that receive the acknowledgment move to their parent's location. If there is no more local move available, robot i sends a completion message using SENDCOMPLETIONMESSAGE to the neighboring robots which will be propagated to all the robots. Then the next active robot executes LOCALMOVE.

Observe that in the proposed distributed coverage control algorithm, the robots only move to their neighbors partition. To do so, each robot may calculate the local change in the objective function using Equations (6) and (7). The information required to calculate the changes in the objective is limited to the obstacle and event distributions in the neighboring partitions.

C. Examples of Locally Optimal Solutions To Escape

Now we provide two examples for the algorithms in [26] and [14]. We illustrate the locally optimal solution reached by these algorithms, and the moves considered in the proposed algorithm which helps escape these sub-optimal solutions.

Consider a discrete coverage problem with four vertices and three robots initialized at the configuration shown in Figure 4(a). The bars on the vertices of the graph represent the weight of the vertices. By the communication model in [26], all the robots are neighbors of each other. The local move in algorithm in [26] moves the robots inside their partitions if the move improves the sensing cost of its partition and the neighboring partitions. Note that the initialized configuration of the robots is a locally optimal solution for the algorithm in [26]. However, in the proposed algorithm the robots will improve on the current configuration by performing single-hop move type II. Figure 4(b) shows the final configuration with the proposed algorithm.

Figure 5 shows a continuous environment with a single robot. The sensing cost of an event is a function of the geodesic distance from the robot. The high-level idea of the algorithm

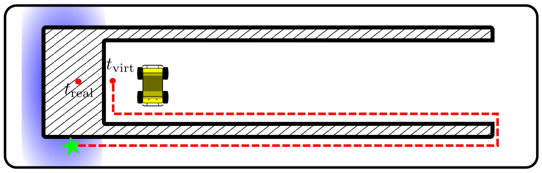


Fig. 5: A test environment for Algorithm in [14]

in [14] is to find the centroid in the environment without obstacles (see $t_{\rm virt}$ in Figure 5) and if the centroid is inside an obstacle, then the algorithm projects the centroid to a face of the obstacle (see $t_{\rm real}$ in Figure 5) and moves the robot towards $t_{\rm real}$. Observe that in the scenarios where the sensing cost is a function of the length of the shortest path between the robot and the event location, the projection of the centroid may result in sub-optimal solutions. However, the proposed algorithm avoids these scenarios by solving the coverage problem on a discrete representation of the environment, and for a sufficiently small ζ, the proposed algorithm converges to a near-optimal solution.

V. Analysis of the Algorithm

In this section, we provide an analysis on the quality of the solutions provided by the proposed algorithm.

A. Correctness and Approximation Factor

Prior to providing the main results on the correctness and approximation factor of the algorithm, we provide two results on the change in the sensing cost of vertices in the partitions of the neighboring robots with a move of a robot.

The following result shows that a move by a robot inside its partition can only change the sensing cost of the vertices in its neighboring partitions.

Lemma V.1: Consider a vertex $z \in W_j$ where robot j at position q_j is the closest robot to z. Then robot j is closer to z than any vertex in the partition of a non-neighbor robot, i.e., $c(z, q_j) \leq \min_{i \notin \mathcal{N}(j)} \min_{u \in W_i} c(z, u)$.

Proof: Proof by contradiction. Suppose there exists a move to vertex $v \in W_i$ by robot i that changes the sensing cost of a vertex $z \in W_j$ of a non-neighbor robot j, i.e., $c(v, z) < c(q_j, z)$ which is equivalent to $d(v, z) \leq d(q_j, z)$ by the monotonicity of function f. By the triangle inequality,

$$d(q_j, v) \leq d(v, z) + d(q_j, z) \leq 2d(q_j, z). \tag{8}$$

Observe that $v \in W_i$, then $c(q_j, v) \leq c(q_i, v)$ which implies $d(q_j, v) \leq d(q_i, v)$. By Equation (8), we have $d(q_i, q_j) \leq d(q_j, v) + d(q_i, v) \leq 2d(q_j, v) \leq 4d(q_j, z)$. Then by the definition of the neighboring robots in Section 4.1, robots i and j are neighbors. This is a contradiction. ■

Observe that the proof of LemmaV.1 holds for both definitions of neighboring robots in continuous and discrete environments. Also, the following result shows that if a robot i moves anywhere in the graph, then the vertices previously in W_i will be assigned to robot in $\mathcal{N}(i)$.

Lemma V.2: For any vertex $z \in W_i$, there exists a robot $j \in \mathcal{N}(i)$ at q_j where $c(z, q_j) \leq c(z, q_k)$ for all $k \notin \mathcal{N}(i)$.

Proof: First we prove the result using definition of the neighbor robots in Definition 1. Suppose there exists $k \notin \mathcal{N}(i)$ and vertex $z \in W_i$ such that $c(z, q_k) < \min_{j \in \mathcal{N}(i)} c(z, q_j)$. Let P be the shortest path from z to q_k. Let p be the point on the path where P intersects the boundary of Voronoi cell of robot i. The point p is not on the boundary of robot k, otherwise the distance between $d(q_i, q_k) \leq 4 \max\{\max_{p \in V_i} d(q_i, p), \max_{p \in V_k} d(q_k, p)\}$ which is a contradiction. Hence, the point p is on the boundary of a neighboring robot j. Therefore, there is a path from q_j to z shorter than P, then $c(q_j, z) = f(d(q_j, z)) \leq f(d(q_k, z)) = c(q_k, z)$.

Now we prove the result for the case where the underlying continuous coverage problem is unknown and neighbors are defined according to Definition IV.4. Suppose there exists $k \notin \mathcal{N}(i)$ and vertex $z \in W_i$ such that $c(z, q_k) < \min_{j \in \mathcal{N}(i)} c(z, q_j)$. Observe that if two vertices of partitions W_i and W_k share an edge in E, then by the definition of neighboring robots in Definition IV.4, robots i and k are neighbors. Therefore, since $k \notin \mathcal{N}(i)$, then there is no shared edge between the vertices in W_i and W_k. Let P be the path on G from q_k to z. Then the path P should contain a vertex u in a partition of another robot j which is a neighbor of robot i. Then by the metric property of the graph,

$$c(q_j, z) \leq c(q_j, u) + c(u, z) \leq c(q_k, u) + c(u, z) = c(q_k, z),$$

where the second inequality is due to $u \in W_j$ and $c(u, q_j) \leq c(u, q_k)$. This is a contradiction. ■

Then we provide the following result on the change in the global objective with a successful move in the algorithm.

Lemma V.3: If a local move is accepted by the robot, then the global objective improves by at least ϵ_0.

Proof: A local move falls under the following cases:

(i) Since, by Lemma V.1, a move of type 1 can only change the sensing cost of the neighboring robots. Then the result is trivial for the local moves of type 1.

(ii) If the local move consists of a move by the robot i that is executing LocalMove to a vertex v in its partition and a neighboring robot j moving to vertex q_i. Let Q' be the configuration after the local move, then the change in the global objective $\Delta\mathcal{D} = \mathcal{D}(Q') - \mathcal{D}(Q)$ is given by the following:

$$\Delta\mathcal{D} = \sum_{k \in [m]} \sum_{u \in W_k} w(u)[\min_{q \in Q'} c(u, q) - c(u, q_k)]. \tag{9}$$

They by Lemma V.1 and V.2, the sensing cost changes only for vertices in $u \in \cup_{k \in \mathcal{N}(i)} W_k$, therefore,

$$\begin{aligned} \Delta\mathcal{D} = & \sum_{k \in \mathcal{N}(i)} \sum_{u \in W_k} w(u)[\min_{q \in Q'} c(u, q) - c(u, q_k)] \\ = & \sum_{k \in \mathcal{N}(i)} \sum_{u \in R_k(v)} w(u)[c(u, v) - c(u, q_k)] \\ & + \sum_{k \in \mathcal{N}(i)} \sum_{u \in W_k \setminus R_k(v)} w(u)[\min_{q \in Q'} c(u, q) - c(u, q_k)]. \end{aligned}$$

Observe that the sensing cost for vertex $u \in W_k \setminus R_k(v)$ for robot k at $q_k \in Q' = \mathcal{N}(i) \cup \{v\} \setminus \{j\}$ does not change, therefore, we have

$$\Delta\mathcal{D} = \sum_{k\in\mathcal{N}(i)} \sum_{u\in R_k(v)} w(u)[c(u,v) - c(u,q_k)] + \sum_{u\in W_j\setminus R_j(v)} w(u)[\min_{q\in Q'} c(u,q) - c(u,q_j)].$$

Hence, the result follows immediately as $\Delta\mathcal{D} = \rho_v + l_v$.

(iii) Suppose the local move consists of a move by the robot i that is executing LOCALMOVE to a vertex v in its partition and a sequence of moves between the neighboring robots. Without loss of generality, let $\langle v, q_i, q_{i+1}, \ldots, q_{j-1}, q_j\rangle$ be the sequence of moves between the neighboring robots where each robot moves to the previous vertex of the preceding robot in the sequence. Let Q' be the configuration after the local move, then the change in the global objective is given by Equation (9). Observe that each robot accepts only the message from the parent robot. Therefore, among the neighbors of the robots in the sequence only the parent of each robot moves.

First we show that the change in the sensing cost under this sequence of moves only occurs for vertices in $\cup_{k\in\mathcal{N}(i)} W_k$ and vertices in W_j. Let u be a vertex assigned to robot $k' \in [m]\setminus\{j \cup \mathcal{N}(i)\}$ in configuration Q, i.e., $u \in W_{k'}$. Since $k' \notin \mathcal{N}(i)$, then by Lemma V.1 a move to vertex v will not improve the sensing cost of u. Also if k' is among the robots moving in the sequence, then there is a robot moving to its previous location, therefore, each vertex in $W_{k'}$ will be sensed by another robot with the same sensing cost. Therefore, the total change $\Delta\mathcal{D}$ in the sensing cost of the vertices becomes

$$\Delta\mathcal{D} = \sum_{k\in\mathcal{N}(i)} \sum_{u\in R_k(v)} w(u)[c(u,v) - c(u,q_k)] + \sum_{u\in W_j} w(u)[\min_{q\in Q'} c(u,q) - c(u,q_j)].$$

Hence, the result follows immediately as $\Delta\mathcal{D} = \rho_v + l$. ■

Now we show the following result on the valid local moves in CENTRALIZEDALG given the final configuration of the proposed distributed algorithm.

Lemma V.4: If the proposed distributed algorithm terminates, then there is no single swap move in the centralized local search algorithm CENTRALIZEDALG (See Section II-B) that improves the objective function.

Proof: Suppose that there exists a centralized local move of robot j at vertex q_j to a vertex $v \in W_i$ that improves the objective function by ϵ_0. Therefore, adding a robot to v improves the sensing cost of the vertices in $\cup_{k\in\mathcal{N}(i)} W_k$ by $\rho_v \leq -\epsilon_0$. Therefore, by construction of the algorithm, the robot i would have suggested the move to its neighboring robots. Suppose after l communications, robot j at q_j receives the message for the first time. In Line 2 (resp. Line 9) of Algorithm 3 if $j \in \mathcal{N}(i)$ (resp. $j \notin \mathcal{N}(i)$), robot j calculates the increase in the sensing cost ℓ_v (resp. ℓ) for the vertices in $\cup_{k\in\mathcal{N}(j)} W_k$ by the move from q_j to the parent of robot j. Since robot j has rejected this offer, by Lemma V.3 the change in the global sensing is less than ϵ_0. This is a contradiction. ■

Theorem V.5: The proposed distributed coverage control algorithm provides a solution within $5 + o(\epsilon_0)$ factor of the optimal configuration.

Proof: The result follows immediately from Lemma V.4. The final configuration in the distributed algorithm is a locally optimal solution for the CENTRALIZEDALG with single swap at each iteration, therefore, the configuration provides a coverage within a factor $5 + o(\epsilon_0)$ of the global optimal. ■

Corollary V.6: Given an environment $\mathcal{X}$ with m mobile robots and a sampling of $\mathcal{X}$ with dispersion ζ, the solution Q obtained from the proposed distributed coverage control algorithm provides coverage cost $\mathcal{H}(Q) \leq 5\mathcal{H}(S^*) + o(f(\zeta) + \epsilon_0)$, where S^* is the optimal solution of the continuous problem.

Proof follows immediately from Theorems III.3 and V.5.

Remark V.7 (Collision Avoidance): In the practical implementation of the proposed algorithm, each robot is equipped with a local planner to avoid collisions with obstacles and other robots. At at each iteration, the robots move to their new discretized location while avoiding the collisions using a local planner. Since the theoretical guarantees of this paper are based on location of the robots at the end of each iteration, therefore the robots preserve the theoretical guarantees while using a local planner to avoid collisions. We illustrate this with an implementation of the proposed algorithm in Section VII-C.

B. Parallel Execution of Local Moves

The proposed algorithm in Section IV requires the robots to take turns in performing the local moves and improve the solution quality. However, we can modify the implementation such that some robots can operate in parallel. To do this, instead of the robots performing local moves in turn, the robots calculate the change in local objectives and send and receive messages in parallel. If a robot receives messages originating from multiple active robots, it selects the message from one of them (for instance, from the robot with lowest UID) and runs Algorithm 3 for that message. When a robot decides to move to a new location, it moves only if none of its neighbors are currently moving. If any of its neighbors are currently moving, it waits until all of its neighbors stop moving and runs Algorithm 2 and Algorithm 3 again.

Theorem V.8: The asynchronous implementation of the proposed distributed coverage control algorithm provides a solution within $5 + o(\epsilon_0)$ factor of the optimal configuration.

Proof: By Lemmas V.1 and V.2, if the robot i moves in its partitions or to a vertex v in the partition of robot j, the cost of only the vertices in $\mathcal{N}(i)$ and $\mathcal{N}(j)$ is affected. Therefore, if a robot only moves when none of its neighbors are moving, the change in local cost calculated by that robot is correct and the rest of the proof follows identical to the proof of Theorem V.5. Hence the distributed algorithm presented in the paper can be implemented asynchronously. ■

Also, note that since the asynchronous implementation performs the same local moves as the proposed algorithm in Section IV, therefore, the time complexity analysis of the algorithm holds for the asynchronous implementation.

VI. Time Complexity

In this section we characterize the runtime and the communication complexity of the proposed algorithm and show that the proposed algorithm converges to a solution within $5+o(\epsilon_0)$ factor of the optimal solution in a polynomial number of iterations. This convergence result shows that the proposed algorithm is a polynomial time distributed approximation algorithm for the coverage planning problem. To the best of our knowledge, this is the first distributed approximation algorithm for the multi-robot coverage problem.

Let Q_0 be the starting configuration of the robots and Q^* be the optimal configuration for the coverage problem on graph G. Now we show that the proposed algorithm has a polynomial runtime under the following assumption:

Assumption VI.1: The $\max_{u,v\in\mathcal{S}} c(u,v)$ is polynomial in the number of samples $|\mathcal{S}|$.

For instance, if a graph is constructed via a grid sampling of the continuous environment, where each cell is a $d\times d$ square, then we have

$$\max_{e\in E} c(e) = f(\sqrt{2|\mathcal{S}|}d) = O(\sqrt{|\mathcal{S}|}),$$

where the second equality follows from sub-additivity of the sensing function f.

Lemma VI.2: For a given $\epsilon > 0$, the proposed algorithm with $\epsilon_0 \leq \epsilon$ achieves a $5+o(\epsilon)$ approximation factor with at most

$$\log(\mathcal{D}(Q^*)/\mathcal{D}(Q_0))/\log\left(1-\frac{\epsilon}{\max_{e\in E} c(e)}\right)$$

iterations, which is polynomial in the input size and $1/\epsilon$.

Proof: Let Q_i be the configuration of the robots at step i of the algorithm. As the global sensing cost improves by at least ϵ_0 after each iteration, we have,

$$\mathcal{D}(Q_{i+1}) - \mathcal{D}(Q_i) \leq -\epsilon_0.$$

Observe that $\mathcal{D}(Q_i) \leq \max_{e\in E} c(e)$. Therefore,

$$\mathcal{D}(Q_{i+1}) \leq \mathcal{D}(Q_i) - \frac{\epsilon}{\max_{e\in E} c(e)}.$$

Therefore, at each iteration of the distributed coverage algorithm the sensing cost improves by the factor of $1-\frac{\epsilon}{\max_{e\in E} c(e)}$. Hence, the algorithm terminates in $\log(\mathcal{D}(Q^*)/\mathcal{D}(Q_0))/\log\left(1-\frac{\epsilon}{\max_{e\in E} c(e)}\right)$ iterations which is polynomial in the input size and $1/\epsilon$. ■

Observe that by Theorem V.5 and Lemma VI.2, for a given $\epsilon > 0$ we can achieve a polynomial run-time and an approximation factor of $5+o(\epsilon)$ by setting $\epsilon_0 \leq \beta\epsilon$ for any $\beta \leq 1$.

Remark VI.3: At each iteration of the algorithm, at most m^2 messages are sent with size at most $n\log(\max_{e\in E} c(e)) + \log(m)$ bits by the robot executing LocalMove. Then at most m^2 messages are sent back between the robots in the acceptance/rejection step. Finally, the SendAcknowledgement step requires at most m messages. Hence, the communication complexity of the proposed algorithm is

$$O\left(\log(\mathcal{D}(Q^*)/\mathcal{D}(Q_0))/\log(1-\frac{\epsilon}{\max_{e\in E} c(e)})m^2\right).$$

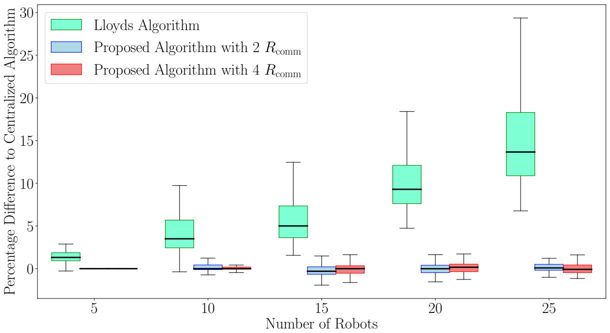


Fig. 6: Percentage difference of the solutions of different algorithms to the solution of the centralized algorithm

VII. Simulation Results

In this section, we evaluate the performance of the proposed distributed algorithm and compare it to convex and non-convex distributed coverage algorithms and the centralized algorithm in [24]. To construct the discrete problem described in Section 3, we use a grid sampling of the environment. We denote the maximum distance inside a Voronoi cell of a robot i by $R_{\text{comm}} = \max_{p\in V_i} d(q_i,p)$ and we evaluate the performance of the proposed algorithm with communication ranges of $4R_{\text{comm}}$ (See Definition 1) and $2R_{\text{comm}}$ which is analogous to the conventional communication model in the continuous coverage literature. In the rest of this section, we use $f(x)=x$ as the sensing cost function.

A. Convex Environments

In this experiment, we compare our algorithm to distributed Lloyd's algorithm [3] in convex environments. We use the Euclidean distance as the metric between two points. The comparison is conducted in a 1500×850 environment with 100 different event distributions. The event distributions are truncated multivariate normal distributions with mean $[1400,800]$ and covariance matrices $\Sigma=[\sigma,0;0,\sigma]$ where σ is uniformly randomly selected from interval $[5,10]\times10^4$. In this experiment, the robots are initialized in the bottom left corner of the environment. Figure 6 illustrates the percentage difference of the solutions provided by the algorithms with respect to the centralized algorithm. Observe that the proposed achieves solution quality very close to the centralized algorithm, even with a large number of robots, while Lloyd's algorithm provides solutions with approximately 15% deviation. Note that the underlying centralized problem is an NP-hard problem and the algorithm used for solving the centralized instances is an approximation algorithm. Therefore, the centralized algorithm provides sub-optimal solutions and in the few of these instances the proposed algorithm provides a better solution, resulting in negative percentages.

Figure 7 shows the percentage share of the different local move types. The dashed lines show the results with $2R_{\text{comm}}$ communication range. The single hop Move type II is a local move which involves a robot and its neighbors in comparison to the multi-hop local move which involves non-neighbor robots. Observe that the majority of the local moves are Move Type I which communicates with only the neighboring

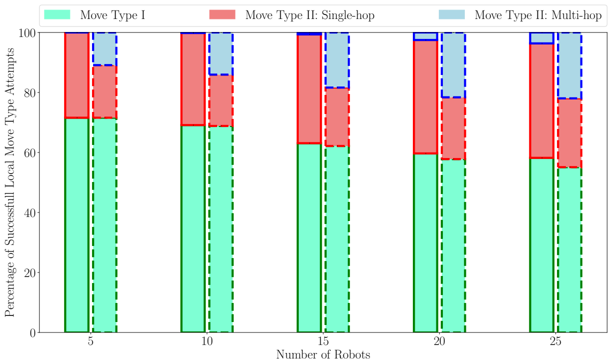


Fig. 7: Percentage share of attempts for different local move types which resulted in improving the objective.

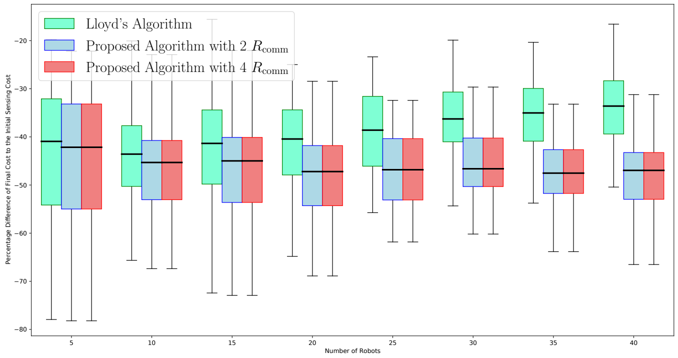


Fig. 8: Percentage improvement of algorithms with random initial configurations.

robots. However, the local Moves of Type II help the proposed algorithm to leave locally optimal solutions.

Figure 8 illustrates the improvement in the sensing cost for the Lloyd's and the proposed algorithm in a convex environment with uniformly random initial configurations. The results are obtained for 50 initial configurations in the environment. Observe that even with the large number of robots where the random configuration provides a relatively good sensing cost, the proposed algorithm improves the solution by 50% on average. In a system of 40 robots, our proposed algorithm provided $\approx 15\%$ additional improvement on the sensing cost as compared to the Llyod's algorithm.

B. Non-Convex Environments

In this section, we compare the solution quality of the proposed algorithm with two different communication ranges to the algorithms in [14], [26] and the centralized algorithm [24]. The experiment is conducted in a 1500×850 environment that contains obstacles (See Figure 10), and using 100 different event distributions. The distributions are generated in the same manner as in the convex environment experiments with uniformly random mean and covariance matrices. The communication model in the non-convex studies are different, for instance, two robots are neighbors in [14], if the intersection of the Voronoi cells of the robots in the environment without obstacles is non-empty, and two robots are neighbors in [26] if the two partitions of the robots share an edge in the discrete representation of the environment.

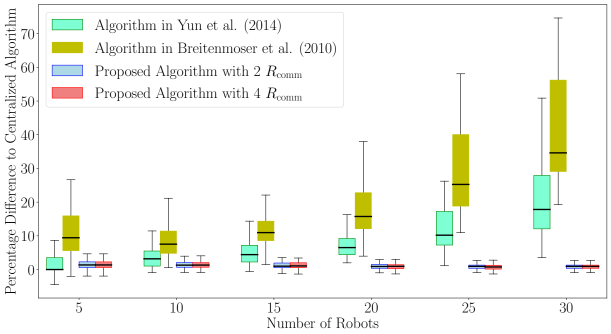


Fig. 9: Percentage difference of the solutions of different algorithms to the solution of the centralized algorithm

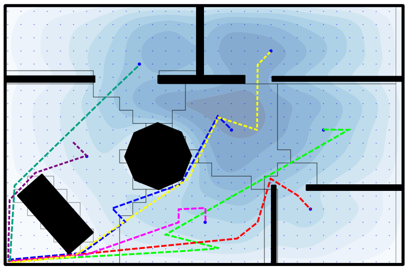


Fig. 10: Robot movements in a non-convex environment using the proposed distributed algorithm

Therefore in the implementations of algorithms in [14] and [26], we assume that robots are connected to every other robot.

Figure 9 shows the percentage difference between the solutions of algorithms compared to the centralized algorithm. Observe that the proposed algorithm with the conventional communication range out-performs both other algorithms by $\approx 20\%$ on average in a system with 30 robots and matches the solution quality of the centralized algorithm. Observe that the run-time of the proposed algorithm and [26] are a function of the number of samples in the discretized environment. In an environment with 869 samples, the average total run-time of the proposed algorithm in an environment with 30 robots is 38.6 seconds, and the run-times of [26] and [14] are 171.5 and 3.8 seconds, respectively. Figure 10 illustrates the final configuration and the movement of the robots using the proposed algorithm in a non-convex environment.

C. ROS Implementation

We have also implemented the proposed algorithm in ROS [35] and evaluated the performance in a non-convex environment with 6 Turtlebot robots. The Turtlebots are equipped with LIDAR for sensing and their motions is governed by differential drive dynamic model. The robots are augmented with TEB local planner [36] for motion planning and avoiding dynamic and static obstacles. Figure 11 illustrates the testing environment, and Figure 13(a) shows the event distribution. Figure 12 illustrates the global objective for the proposed algorithm and the global objective where the robots only perform move type I, which is the adaptation of Lloyd's algorithm to discrete environments. The proposed method provides approximately 27% improvement by performing 2 multi-hop moves and converges to a near-optimal locally

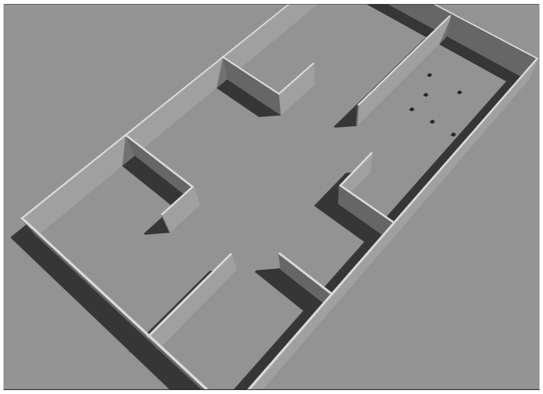


Fig. 11: The test environment in Gazebo simulator [37]

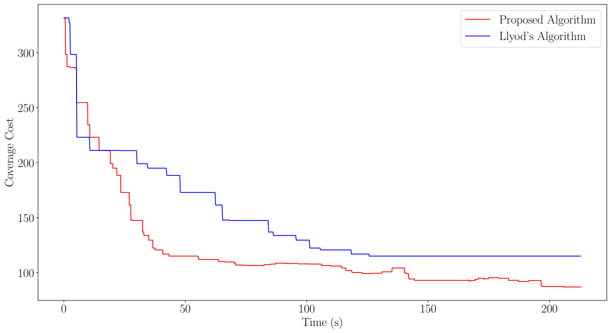


Fig. 12: The coverage cost of the proposed algorithm compare to the Lloyd's algorithm

optimal solution. Figure 13 illustrates the trajectories of the robots performing the proposed algorithm. The adaptation of Lloyd's algorithm, which performs only type I moves reaches the configuration of the robots shown in Figure 13(d) and cannot find any move of type I to improve the solution. However, the proposed algorithm performs a multi-hop move to escape the locally optimal solution which improves the solution quality significantly. A video this experiment illustrating the configurations of the six Turtlebots at different time steps is available online[1]. These simulation results show that the performance of the proposed algorithm is independent of the algorithms used for path planning and collision avoidance and shines a light on the practicality of the proposed algorithm in real-world scenarios.

VIII. Conclusion

This paper considers the problem of distributed coverage control in convex and non-convex environments. A connection is established between the solution quality of the continuous coverage problem and the solution to the coverage problem on a discrete representation of the environment. We also propose the first distributed approximation algorithm for the coverage problem in discrete and continuous environments and provide a bound on the quality of the solution. We also characterize the run-time and communication complexity of the proposed algorithm. For future work, we consider adapting the proposed method to capture heterogeneous robots with multiple sensors and multiple event types in the environment.

[1] ece.uwaterloo.ca/~sl2smith/papers/2021TCNS-Distributed_Coverage.mp4

References

[1] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, "Autonomous oceanographic sampling networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.

[2] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *ACM International Conference on Mobile Computing and Networking*, 2001, pp. 139–150.

[3] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[4] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 364–378, 2012.

[5] A. Sadeghi and S. L. Smith, "Coverage control for multiple event types with heterogeneous robots," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3377–3383.

[6] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[7] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, pp. 1–53, 2011.

[8] A. Sadeghi, A. B. Asghar, and S. L. Smith, "Approximation algorithms for distributed multi-robot coverage in non-convex environments," in *Algorithmic Foundations of Robotics XIV*. Springer International Publishing, 2021, pp. 330–346.

[9] M. Schwager, M. P. Vitus, S. Powers, D. Rus, and C. J. Tomlin, "Robust adaptive coverage control for robotic sensor networks," *IEEE Trans. on Control of Network Systems*, vol. 4, no. 3, pp. 462–476, 2015.

[10] J. Habibi, H. Mahboubi, and A. G. Aghdam, "Distributed coverage control of mobile sensor networks subject to measurement error," *IEEE Trans. on Automatic Control*, vol. 61, no. 11, pp. 3330–3343, 2016.

[11] L. Dou, C. Song, X. Wang, L. Liu, and G. Feng, "Coverage control for heterogeneous mobile sensor networks subject to measurement errors," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3479–3486, 2018.

[12] R. Patel, P. Frasca, J. W. Durham, R. Carli, and F. Bullo, "Dynamic partitioning and coverage control with asynchronous one-to-base-station communication," *IEEE transactions on control of network systems*, vol. 3, no. 1, pp. 24–33, 2015.

[13] W. Luo and K. Sycara, "Voronoi-based coverage control with connectivity maintenance for robotic sensor networks," in *International Symposium on Multi-Robot and Multi-Agent Systems*. IEEE, 2019, pp. 148–154.

[14] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4982–4989.

[15] M. Thanou, Y. Stergiopoulos, and A. Tzes, "Distributed coverage using geodesic metric for non-convex environments," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 933–938.

[16] Y. Kantaros, M. Thanou, and A. Tzes, "Visibility-oriented coverage control of mobile robotic networks on non-convex regions," in *IEEE International Conf. on Robotics and Automation*, 2014, pp. 1126–1131.

[17] H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. Zhang, "Distributed coordination of multi-agent systems for coverage problem in presence of obstacles," in *IEEE American Control Conference*, 2012, pp. 5252–5257.

[18] C. H. Caicedo-Nunez and M. Zefran, "Performing coverage on nonconvex domains," in *IEEE International Conference on Control Applications*, 2008, pp. 1019–1024.

[19] C. H. Caicedo-Núnez and M. Zefran, "A coverage algorithm for a class of non-convex regions," in *IEEE Conference on Decision and Control*, 2008, pp. 4244–4249.

[20] R. J. Alitappeh, G. A. Pereira, A. R. Araújo, and L. C. Pimenta, "Multi-robot deployment using topological maps," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 3-4, pp. 641–661, 2017.

[21] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed coverage and exploration in unknown non-convex environments," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 61–75.

[22] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM Journal on computing*, vol. 33, no. 3, pp. 544–562, 2004.

[23] S. Li and O. Svensson, "Approximating k-median via pseudo-approximation," *SIAM Journal on Computing*, vol. 45, no. 2, pp. 530–547, 2016.

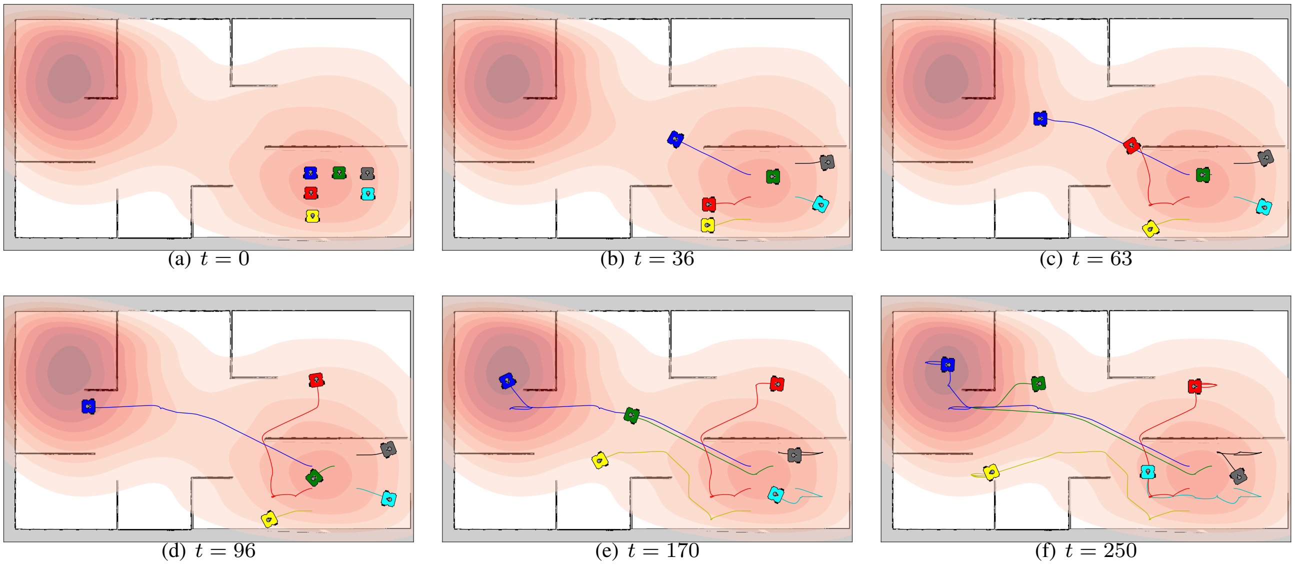


Fig. 13: Trajectories of the robots for the ROS implementation of the proposed algorithm

[24] S. Ahmadian, Z. Friggstad, and C. Swamy, “Local-search based approximation algorithms for mobile facility location problems,” in *Proceedings of the 24th annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2013, pp. 1607–1621.

[25] A. Sadeghi and S. L. Smith, “Re-deployment algorithms for multiple service robots to optimize task response,” in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2356–2363.

[26] S.-k. Yun and D. Rus, “Distributed coverage with mobile robots on a graph: locational optimization and equal-mass partitioning,” *Robotica*, vol. 32, no. 2, pp. 257–277, 2014.

[27] M.-F. F. Balcan, S. Ehrlich, and Y. Liang, “Distributed k-means and k-median clustering on general topologies,” in *Advances in Neural Information Processing Systems*, 2013, pp. 1995–2003.

[28] A. Y. Yazıcıoglu, M. Egerstedt, and J. S. Shamma, “Communication-free distributed coverage for networked systems,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 499–510, 2016.

[29] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[30] D. J. Bertsimas and G. Van Ryzin, “A stochastic and dynamic vehicle routing problem in the Euclidean plane,” *Operations Research*, vol. 39, no. 4, pp. 601–615, 1991.

[31] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, “Dynamic vehicle routing for robotic systems,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011.

[32] D. B. Shmoys, “Approximation algorithms for facility location problems,” in *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 2000, pp. 27–32.

[33] A. Sadeghi and S. L. Smith, “Heterogeneous task allocation and sequencing via decentralized large neighborhood search,” *Unmanned Systems*, pp. 1–17, 2017.

[34] T. Lemaire, R. Alami, and S. Lacroix, “A distributed tasks allocation scheme in multi-uav context,” in *IEEE International Conference on Robotics and Automation*, vol. 4, 2004, pp. 3622–3627.

[35] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system.” [Online]. Available: https://www.ros.org

[36] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[37] C. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. Rivero, J. Manzo, E. Krotkov, and G. Pratt, “Inside the virtual robotics challenge: Simulating real-time robotic disaster response,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 494–506, April 2015.

Armin Sadeghi (M′16) received dual B.S. degrees in mechanical engineering and aerospace engineering from the Sharif University of Technology, Tehran, Iran, in 2013, the M.S. degree in electrical engineering from the University of Waterloo, Waterloo, Canada, in 2016, and the Ph.D. degree in electrical engineering from the University of Waterloo, in 2020. He is currently a Postdoctoral Fellow in the Autonomous Systems Laboratory at the University of Waterloo. His research interest includes multi-robot task allocation, approximation algorithms, and autonomous driving.

Ahmad Bilal Asghar received the B.S. degree in electrical engineering from Lahore University of Management Sciences, Pakistan, in 2012, the M.A.Sc. degree in electrical engineering from the University of Waterloo, Canada, in 2015, and the Ph.D. degree in electrical engineering from the University of Waterloo, in 2020. He is currently a postdoctoral associate in the Maryland Robotics Center at the University of Maryland. His research interests include multi-robot path planning for persistent monitoring, and risk-aware planning.

Stephen L. Smith (S′05 – M′09 – SM′15) received the B.Sc. degree in engineering physics from Queen’s University, Canada, in 2003, the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Canada, in 2005, and the Ph.D. degree in mechanical engineering from the University of California, Santa Barbara, USA, in 2009.

He is currently a Professor in the Department of Electrical and Computer Engineering at the University of Waterloo, Canada, where he holds a Canada Research Chair in Autonomous Systems. From 2009 to 2011 he was a Postdoctoral Associate in the Computer Science and Artificial Intelligence Laboratory at MIT. His main research interests lie in control and optimization for autonomous systems, with a particular emphasis on robotic motion planning and coordination.