Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-20 (2021) 765–770

Multi-Agent Systems Coverage Control in Mixed-Dimensional and Hybrid Environments ⋆

Tairan Liu * **Javad Mohammadpour Velni** *

* *School of Electrical and Computer Engineering, University of Georgia, Athens, GA 30602 USA. (e-mail: Tairan.Liu@uga.edu; javadm@uga.edu).*

Abstract: In this paper, we address the problem of deploying a team of agents over a given environment. The environment we consider here is different from most of the existing works and includes mixed-dimensional and hybrid cases. To deploy the agents, we first find the geodesic Voronoi partitions from the agents' current locations. Then, we let each agent move to the "center" of its partition; this would minimize the worst-case response time for any agent to arrive at an arising event inside the workspace. Due to the non-convexity of the environment under study, it is natural to use geodesic distance instead of the generally used Euclidean distance as the distance measurement metric to keep the trajectory of each agent inside its admissible space. We present algorithms for agents deployment and show that the proposed approaches ensure that each agent converges to the optimal position quantified by a cost function in terms of the geodesic distance. Finally, the results of the simulation study are presented to demonstrate the effectiveness of the distributed deployment algorithms.

Keywords: Coverage control, geodesic distance, non-convex environments.

1. INTRODUCTION

Using a group of agents (robots or sensors) in a planar or spatial space to cover or monitor surrounding environment has been a popular research topic in the last decade. The agents can handle tasks such as area exploration and mapping, surveillance and intruder detection, search and rescue, environmental monitoring and field operation (Cortés and Egerstedt, 2017; Kantaros et al., 2015; Nourbakhsh et al., 2005; Howard et al., 2006; Hameed, 2018; Barrientos et al., 2011). There is no doubt that a team of agents can provide more complete sensing results from different aspects, angles, or locations than a single robot or sensor can. It is unbeatable in speed, efficiency and capability with collaborative agents compared to a single agent. *Multi-agent coverage control* takes care of the agents deployment by maximizing overall performance and minimizing a cost so that collaborative agents can optimally accomplish assigned task(s).

Most of the existing works on multi-agent coverage control problem use convex polytopes to represent the workspace of agents (Cortes et al., 2004; Wang, 2010; Schwager et al., 2009; Li and Cassandras, 2005). Beyond the simplicity in convex shapes, a lot of tools and properties are applicable to solve underlying problems. One commonly used tool is the Voronoi diagram. It is used to divide the environment into sub-regions such that each agent is assigned with a region. The activities or events within the region should be then taken care of by the agent. This method not only guarantees that the events are handled by the closest agent, but also avoids possible conflict between agents since each agent will not get across the border and enter the others' partitions. However, for cases that the environment is not a convex polytope, the centroid of each subregion may not be inside it. In addition, the agents may also need to get across the border in the process of moving to their centroid. In many cases, the space outside the given environment could be obstacles or no-entry regions. So, it is necessary to keep the whole trajectory inside the given environment rather than letting the agents move across the border. Since there is no guarantee that the line segment connecting two points is inside the environment, the commonly used Euclidean distance may not work properly as expected when dealing with non-convex cases. To keep the whole trajectory inside the admissible region, one way to define the distance between two points is using the geodesic metric. Relevant research coping with non-convex environments can be found in (Breitenmoser et al., 2010; Lee et al., 2014; Stergiopoulos et al., 2015; Thanou et al., 2013; Pimenta et al., 2008; Alitappeh and Pimenta, 2016; Alitappeh et al., 2017; Lee et al., 2016). Even though most of the research in multi-agent coverage control studies continuous workspace, another branch uses graph-based methods to handle environments that consist of discrete paths (Alitappeh et al., 2017; Davoodi et al., 2020; Davoodi and Velni, 2020).

Inspired by the works in continuous and discrete environments, we found a combination of them, a hybrid environment composed of continuous parts and discrete parts, and an environment composed of parts with different dimensions, to be also intriguing. These new compounds not only inherit the merits and features of each element, but also

⋆ This research was financially supported by the United States National Science Foundation under award #1934481.

10.1016/j.ifacol.2021.11.264

generate new application scenarios. For instance, several regions connected by a narrow path can be modeled by 2D polytopes connected by a 1D curve; a region whose majority are concentrated in a polytope but several isolated outliers are far away can be modeled by a continuous part connecting discrete points.

In this paper, we introduce the concepts of mixed-dimensional and hybrid environments. We employ the geodesic distance as the metric of distance measurement in this work to solve the locational optimization problem. The environments that can be modeled are combinations of the elements in Figure 1. Some examples of the combination are shown in Figure 2.

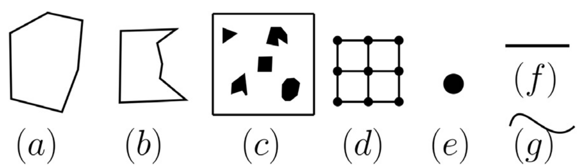

Fig. 1. Allowable elements of an environment.

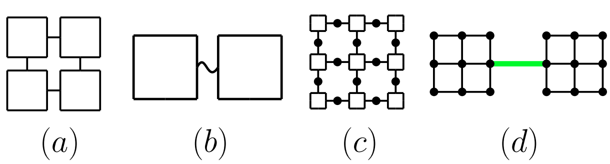

Fig. 2. Some examples of environments composed of basic elements.

The main contributions of this paper are as follows:

(1) We propose the concepts of mixed-dimensional and hybrid environments in coverage control problem.
(2) We develop algorithms to solve the coverage control problem in mixed-dimensional and hybrid environments.
(3) We prove that the proposed method ensures convergence.
(4) We present a novel understanding of the cost function in terms of the geodesic distance.

The paper is organized as follows. The problem statement and some preliminary information are presented in Section 2. In Section 3, we present the methodology and the problem to be addressed. In Section 4, simulation results are presented. Finally, concluding remarks are made in Section 5.

2. PROBLEM STATEMENT AND PRELIMINARIES

2.1 Problem Statement

The problem we intend to address in this paper is the deployment of a team of agents in a bounded space for minimizing the worst-case response time for any agent to arrive at an arising event in the workspace. The bounded space is one of the aforementioned mixed-dimensional, hybrid environments, or a combination of them.

2.2 Environments

We call a region *a mixed-dimensional environment* if it is composed of multiple connected parts where at least one part has different dimension than others. For example, the regions shown in Figure 2(a) and (b) are mixed-dimensional environments. In these mixed-dimensional cases, the square areas are in two-dimensional space and the line segments are in one-dimensional space. Even though the whole connected region is in two-dimensional plane, the line segments actually are one-dimensional geometric shape. This notation can be extended to cases with curves, where curves may appear in 2D or 3D space but still 1D in essence.

We call a region *a hybrid environment* if it is composed of connected parts where at least one part is a continuous space and another part is a set of discrete positions connected through paths. For example, the regions shown in Figure 2(c) and (d) are hybrid environments. In Figure 2(d), the black dots are discrete positions connected through paths (black lines), and the thick green line segment is in 1D space. The difference between the black lines and the thick green line segment is that all the points on the thick green line are considered as admissible workspace for agents to stay; the paths between discrete vertices are only allowed for agents to pass through. Similar to the mixed-dimensional case, a curve is considered as a continuous region in 1D space.

In a hybrid environment, we call the point that connects the continuous region and the discrete part *access point*. All the access points and isolated points in the environment together constitute a weighted graph $G = (V, \mathcal{E}, \mathcal{W})$, where V is the set of vertices, i.e., the access points and isolated points in the environment, $\mathcal{E} \subset V \times V$ is the set of edges, which represent the paths the agents are allowed to go through, and $\mathcal{W}$ is the set of weight values corresponding to each edge, where the values are the distance between the two vertices connected by the edge.

A mixed-dimensional environment can be considered as a special case of a non-convex environment, in which several regions are connected by narrow tunnels. When reducing the width of the narrow tunnels to zero, the non-convex environment becomes a mixed-dimensional environment. A mixed-dimensional environment can be used to represent the cases that several major areas are connected with paths. It is useful in applications such as city areas connected by highways, distributed large farmlands connected by roads, islands connected by sea bridges or courses. This environment is different from the non-convex one, since all the areas are needed to be covered in the non-convex case, even the narrow tunnels. In a mixed-dimensional environment, the connection segments do not have area and only take time to transport.

Similarly, a hybrid environment can also be considered as a special case of the mixed-dimensional environment. When the areas of some polytopes are reduced to zero, they become discrete points in the space with the original density of a region condensed to a single point. A hybrid environment can be used to represent cases that involve several major areas and discrete outliers. Events are distributed relatively sparse over the major areas, while the events at discrete points are condensed.

2.3 Euclidean Distance, Geodesic Distance, and Shortest Path

Euclidean distance is used in traditional Voronoi partitions, which requires the environment to be convex. The shortest path between two points, which is the line segments connecting them, is straightforward in the convex polytope. However, issues arise when applying the Euclidean distance in non-convex, mixed-dimensional, and hybrid environments. Since there is no guarantee that the direct connection of two points stays in the environment all the time, the Euclidean distance cannot stand for the real travel distance of agents. Thus, it is necessary to use the geodesic distance as the measurement of distance such that the real feasible travel distance is represented.

Finding the shortest path between two points in a 2D non-convex space or space with polygonal obstacles is known as the Euclidean shortest path problem (Li and Klette, 2011). There are many algorithms in the literature that can be used to solve this problem (Li and Klette, 2011; Mitchell, 1996; Hershberger and Suri, 1993; LaValle, 2006; Crane et al., 2020). We adopt the *visibility graph* approach to build the graph (De Berg et al., 1997; Ghosh and Mount, 1991) and then use the Dijkstra algorithm to compute the shortest path and the geodesic distance. This approach also works in our mixed-dimensional and hybrid environments. The details will be provided in Section 3.

2.4 Voronoi Partitioning and Geodesic Voronoi Partitioning

Consider an environment Q and a team of m agents r_i, $i \in \mathcal{R} = \{1, \ldots, m\}$, and denote the position of agent r_i as $p_i \in \mathbb{R}^2$.

In existing body of works, the Voronoi diagram has been widely used to partition a convex polytope such that each cell of the diagram is assigned to one agent. In this way, the agent will stay in its own region and take care of the events in this region. No agent will get into other agents' region and hence collide with each other or conflict with other agents' tasks. Furthermore, the Voronoi diagram ensures that any events in the environment is taken care of by the closest agent.

In a Voronoi diagram, the partition cells are separated by straight lines or line segments such that all the points on the boundary satisfy

$$d(q, p_i) = d(q, p_j),\ i, j \in \mathcal{R} \tag{1}$$

where $q \in Q$ is any point on the boundary, p_i and p_j are the positions of nearby agents, and $d(q, p_i)$ is the Euclidean distance between q and p_i, i.e., $d(q, p_i) = \|q - p_i\|$.

The precise boundary of geodesic Voronoi diagram is much more complicated. Since geodesic distance is used instead of Euclidean distance, the boundary (1) becomes

$$d_{w(p_i,q)} = d_{w(p_j,q)} \tag{2}$$

where $w(p_i, q)$ represents the shortest path connecting them and remains inside the admissible space at all times, $d_{w(p_i,q)}$ is the length of this shortest path $w(p_i, q)$, i.e., the geodesic distance. The geodesic Voronoi partition of agent i is given by

$$\mathcal{V}_i = \{q \in Q \mid d_{w(p_i,q)} \le d_{w(p_j,q)}, \forall i, j \in \mathcal{R}\}. \tag{3}$$

If the environment is mixed-dimensional, we use W_i to denote the continuous partition, i.e.,

$$\mathcal{V}_i = W_i,\ i \in \mathcal{R}. \tag{4}$$

If the environment is hybrid, we use W_i to denote the continuous portion, V_i to denote the discrete points, g_i to represent the graph connecting all the vertices, and the pair (W_i, g_i) to denote all the elements in this partition, i.e.,

$$\mathcal{V}_i = W_i \cup \{q_j \mid j \in V_i\},\ i \in \mathcal{R} \tag{5}$$

where q_j is the position of the jth vertex in V_i.

3. METHODOLOGY

3.1 Cost Function

If the environment is mixed-dimensional, we employ the following locational cost for agent r_i, $i \in \mathcal{R}$ (Alitappeh and Pimenta, 2016)

$$\mathcal{H}_i(p_i, W_i) = \int_{W_i} f(d_{w(p_i,q)})\phi(q)dq, \tag{6}$$

where

$$f(x) = x^2 \tag{7}$$

is used throughout the following content. The total cost is the sum of all the agent costs defined as

$$\mathcal{H}(p, W) = \sum_{i=1}^{n} \int_{W_i} f(d_{w(p_i,q)})\phi(q)dq, \tag{8}$$

where $p = \{p_1, \ldots, p_n\}$, $W = \{W_1, \ldots, W_n\}$, $\phi(q)$ is the density function at $q \in Q$, which represents the possibility of events happening at that location.

If the environment is hybrid, we employ the locational optimization function for agent r_i, $i \in \mathcal{R}$ in its partition (W_i, g_i) (Alitappeh and Pimenta, 2016; Alitappeh et al., 2017)

$$\mathcal{H}_i(p_i, g_i, W_i) = \int_{W_i} f(d_{w(p_i,q)})\phi_1(q)dq + \sum_{q \in V_i} f(d_{w(p_i,q)})\phi_2(q), \tag{9}$$

where g_i is the graph connecting the continuous region W_i and isolated points in V_i. The total cost is given by

$$\mathcal{H}(p, G, W) = \sum_{i=1}^{n} \int_{W_i} f(d_{w(p_i,q)})\phi_1(q)dq + \sum_{i=1}^{n} \sum_{q \in V_i} f(d_{w(p_i,q)})\phi_2(q), \tag{10}$$

where G is the graph connecting all the continuous regions and isolated points.

Remark 1. It is noted that the cost function (8) is similar to the standard cost function initially proposed by Cortes et al. (2004). The only difference is replacing the Euclidean distance with the geodesic distance.

Remark 2. Here, we specifically use (7) for its merit in distance-related integration. With this form, the parallel axis theorem can be applied to decompose the integration into two parts: (a) integration about the centroid, and (b) a value only concerned about the distance from any point to the centroid of the region. This decomposition greatly simplifies the integration and reduces the amount of calculations.

Remark 3. With the consideration in Remark 2, we can have a new understanding of the effect of the geodesic distance on the cost function. As shown in Figure 3(a), the cost function of this region when p_i is located at the red dot equals to the integration of the geodesic distance from the red dot to all the points in the region in blue. Then, we can decompose the whole concave shape into two parts as shown in Figure 3(b). All the points in the green region can "see" the red dot directly, and hence the distance between any point and the red dot equals to their Euclidean distance. All other points (i.e., in the pink region) cannot "see" the red dot directly, and hence the geodesic distance is needed. Since the total cost is the sum of the two parts, we can actually move the pink triangle to anywhere if the Euclidean distance from the centroid of the triangle to the red dot equals to the geodesic distance of it (Figure 3(c)). After that, the total cost remains the same value. With proper partitioning, it is possible to transform the concave shape to a convex shape. However, this decomposition does not remain the same for every point in the region. At a different location, the partition might be different – this is in fact the source of complexity in using geodesic distance.

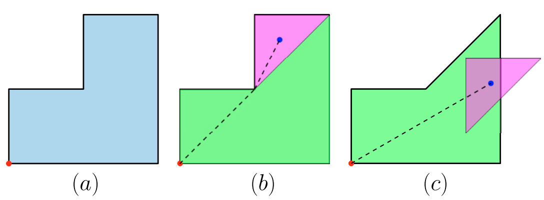

Fig. 3. Transformation of a concave environment.

3.2 Proposed Algorithms and Analysis

We first present an algorithm to calculate the geodesic distance in mixed-dimensional or hybrid environments. As mentioned in Section 2, we adopt the visibility graph approach to build a graph and applying the Dijkstra algorithm to find the geodesic distances in these environments. In these algorithms, we assume that

- each agent has full knowledge of the environment;
- each agent only has the knowledge of its neighboring agents' location information;
- there exists a path between any two positions.

Remark 4. Above assumptions are not against the distributed requirement. On one hand, the agents can continuously communicate with their neighboring agents to exchange their status data without a centralized control station to manage the information. On the other hand, each agent only requires information from its neighbors rather than all the agents.

Next, we present the distributed coverage control algorithm over mixed-dimensional or hybrid environment. We show that the geodesic Voronoi partitioning ensures the minimum value of the total cost function $\mathcal{H}$ with given positions of agents, and the distributed coverage control algorithm ensures convergence to the minimum value of the total cost function $\mathcal{H}$.

Remark 5. Using the proposed algorithm, the agents do not collide with each other since each agent would only move inside its own Voronoi partition and not enter other agents' partitions.

Algorithm 1 Find the shortest path and geodesic distance in a mixed-dimensional or hybrid environment.

Input: Start point q_{start}, end point q_{end}, admissible region W (mixed dimensional) or (W, G) (hybrid)
Output: $w(q_{start}, q_{end})$ and $d_{w(q_{start},q_{end})}$

q_{start}, q_{end}, q_{env} ← Collect all the vertices involved, where q_{env} is the set of vertices composing the environment polygon in the mixed-dimensional case or the polygon of the continuous region W and the isolated points in the graph G in the hybrid case
G ← Build a complete graph from vertices q_{start}, q_{end}, and q_{env}
G_c ← Remove all the edges in G which are not totally inside W (or not edges in G in the hybrid case), i.e., create the visibility graph
$w(q_{start}, q_{end})$ ← Use Dijkstra's algorithm to find the shortest path from q_{start} to q_{end} in G_c
d_{ij} ← Calculate the length of each edge in $w(q_{start}, q_{end})$
$d_{w(q_{start},q_{end})} \leftarrow \sum_{(i,j)\in w(q_{start},q_{end})} d_{ij}$

Algorithm 2 Implementation of the distributed coverage control algorithm.

Input: Initial position of agents (p_i) and the information of the environment (Q)

while True **do**
  Acquire locational information of neighbor agents
  g_i ← Compute the geodesic Voronoi partition based on the drone and its neighbors' current location
  c_i ← Find the point with the minimum cost for each agent, i.e., $\arg\min_{p_i} \mathcal{H}_i(p_i, W_i)$ or $\arg\min_{p_i} \mathcal{H}_i(p_i, g_i, W_i)$
  $w(p_i, c_i)$ ← Find the shortest path from current location to c_i
  Move to c_i following the path $w(p_i, c_i)$
  Update all the states
end while

Theorem 1. For a given bounded environment Q, the geodesic Voronoi partitioning renders the minimum value of the total cost $\mathcal{H}$.

Proof. Denote the regions generated by geodesic Voronoi partitioning by $\{W_i\}$, $i \in \mathcal{R}$ (or $\{(W_i, g_i)\}$ in hybrid case). Denote the regions generated by any other partitioning methods by $\{\hat{W}_i\}$ (or $\{(\hat{W}_i, \hat{g}_i)\}$ in hybrid case).

Let us first consider a mixed-dimensional environment. The total cost of the geodesic Voronoi partitioning and an arbitrary partitioning is given by

$$\mathcal{H}(p, W) = \sum_{i=1}^{n} \int_{W_i} f(d_{w(p_i,q)})\phi(q)dq, \tag{11}$$

and

$$\hat{\mathcal{H}}(p, \hat{W}) = \sum_{i=1}^{n} \int_{\hat{W}_i} f(d_{w(p_i,q)})\phi(q)dq, \tag{12}$$

respectively.

The geodesic Voronoi partitioning ensures that

$$d_{w(p_i,q)} \leq d_{w(p_j,q)},\ \forall q \in W_i,\ i, j \in \mathcal{R}, i \neq j. \tag{13}$$

This means that all the points in W_i ($i \in \mathcal{R}$) are closer to r_i than any other agents.

For the arbitrary partitioning, $\exists i, j \in \mathcal{R}$ $(i \neq j)$, and $q \in \hat{W}_i$ such that

$$d_{w(p_i,q)} \geq d_{w(p_j,q)}. \tag{14}$$

Consider that $q_0 \in W_j$ is the unique point partitioned into $\hat{W}_i$ with the arbitrary partitioning. Then, we have

$$\hat{\mathcal{H}}_i(p_i, \hat{W}_i) = \mathcal{H}_i(p_i, W_i) + f(d_{w(p_i,q_0)})\phi(q_0) \tag{15}$$

and

$$\hat{\mathcal{H}}_j(p_j, \hat{W}_j) = \mathcal{H}_j(p_j, W_j) - f(d_{w(p_j,q_0)})\phi(q_0). \tag{16}$$

Adding (15) to (16), we obtain

$$\begin{aligned}\hat{\mathcal{H}}_i(p_i, \hat{W}_i) + \hat{\mathcal{H}}_j(p_j, \hat{W}_j) = \\ \mathcal{H}_i(p_i, W_i) + \mathcal{H}_j(p_j, W_j) \\ + f(d_{w(p_i,q_0)})\phi(q_0) - f(d_{w(p_j,q_0)})\phi(q_0)\end{aligned} \tag{17}$$

Due to (14), we have

$$f(d_{w(p_i,q_0)})\phi(q_0) \geq f(d_{w(p_j,q_0)})\phi(q_0). \tag{18}$$

Substituting (18) into (17), we have

$$\hat{\mathcal{H}}_i(p_i, \hat{W}_i) + \hat{\mathcal{H}}_j(p_j, \hat{W}_j) \geq \mathcal{H}_i(p_i, W_i) + \mathcal{H}_j(p_j, W_j) \tag{19}$$

Then, the total cost functions satisfy

$$\mathcal{H}(p, W) \leq \hat{\mathcal{H}}(p, \hat{W}). \tag{20}$$

This means that the value of $\mathcal{H}_i(p_i, W_i)+\mathcal{H}_j(p_j, W_j)$ either increases or does not change due to q_0 with the arbitrary partitioning. If q_0 is the only difference between $\{W_i\}$ and $\{\hat{W}_i\}$, the value of the total cost function also either increases or does not change.

If multiple points are assigned to other regions with the arbitrary partitioning, (20) still holds. Then, we know that the total cost either increases or does not change with the arbitrary partitioning method, i.e., the geodesic Voronoi partition renders the minimum value of the total cost function $\mathcal{H}$.

For a hybrid environment, the above conclusion still holds. The proof is very similar and hence omitted here. ■

Theorem 2. For a given environment, Algorithm 2 ensures that the cost function $\mathcal{H}$ converges to the minimum value.

Proof. The algorithm involves a loop with two steps at each iteration. In step A, the geodesic Voronoi partitions are acquired based on the agents' current location. In step B, each agent moves to the optimal location of its partition. We will prove that the value of the total cost function $\mathcal{H}$ will not increase in both steps of each iteration.

Step A: At each iteration, the partitioning renders the minimum value of the total cost function $\mathcal{H}$ at current location of agents (this was proven in Theorem 1). It means that the value of the total cost function should be less than or equal to the value from the last iteration if this is not the initial partitioning.

Step B: The optimal position is the point that has the minimum cost in each region. This leads to $\mathcal{H}_i(c_i) \leq \mathcal{H}_i(p_i)$. Then, we can obtain $\mathcal{H}(c) = \sum_1^n \mathcal{H}_i(c_i) \leq \sum_1^n \mathcal{H}_i(p_i) = \mathcal{H}(p)$.

In summary, both steps will reduce the value of the total cost function, until the iteration that $\mathcal{H}(p, W) = \mathcal{H}(p, \hat{W})$, and $\mathcal{H}(c) = \mathcal{H}(p)$, where W and $\hat{W}$ are partitions from current step and last step, respectively. This means that the agents already arrived at the positions that have the minimum cost. ■

4. NUMERICAL SIMULATION RESULTS

4.1 Basic Configurations

In the numerical validation, we employ a uniform density distribution to the environment. Specifically, we set $\phi(q) = 1$ in the mixed-dimensional environment. Due to the limited space, the simulation results of the hybrid case is omitted here.

4.2 Mixed-dimensional Environment

In this simulation, the environment consists of a square and a triangle connected by a line segment. Four agents were used and initially started from the square region as shown in Figure 4(a). In each step, each agent moved to the position with the minimum value of the cost function in its partition. The trajectories are shown in Figure 4(b). Finally, all the agents moved to the centroid of their own partition, where three agents covered the square area and one agent moved to the centroid of the triangle through the one-dimensional path. The total cost function is decreasing all the time until converging to a constant value as shown in Figure 4(c).

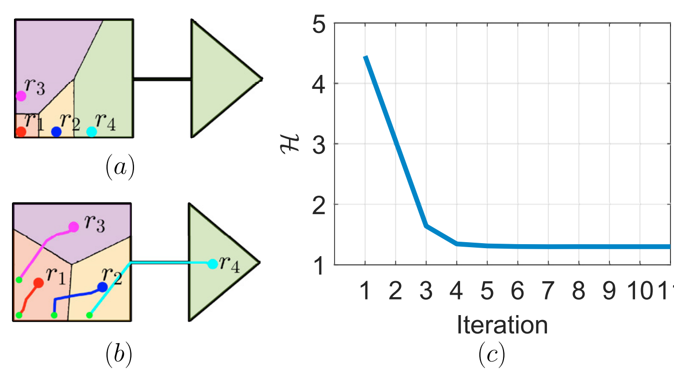

Fig. 4. Simulation results in a mixed-dimensional environment: (a) initial positions of agents and the corresponding geodesic Voronoi partitions, (b) final positions of agents and the corresponding geodesic Voronoi partitions, and (c) total cost function.

5. CONCLUSIONS

In this work, we proposed the concepts of mixed dimensional and hybrid environments, and the coverage control problem in those environments was addressed. We provided distributed solution methods to address the coverage control problem in those irregular environments by employing geodesic distance and geodesic Voronoi partitions. We proved that the proposed methods can ensure that the agents converge to their optimal positions quantified by a cost function. A novel understanding of the cost function in terms of the geodesic distance was also presented. Finally, we showed the effectiveness of the proposed algorithm using numerical simulations on a mixed-dimensional environment. It is worth mentioning that the concepts and algorithm we proposed here do not rely on a specific number of agents or the configuration of the environment.

REFERENCES

Alitappeh, R.J., Pereira, G.A., Araújo, A.R., and Pimenta, L.C. (2017). Multi-robot deployment using topological maps. *Journal of Intelligent & Robotic Systems*, 86(3-4), 641–661.

Alitappeh, R.J. and Pimenta, L.C. (2016). Distributed safe deployment of networked robots. In *Distributed autonomous robotic systems*, 65–77. Springer.

Barrientos, A., Colorado, J., Cerro, J.d., Martinez, A., Rossi, C., Sanz, D., and Valente, J. (2011). Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5), 667–689.

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

Cortés, J. and Egerstedt, M. (2017). Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6), 495–503.

Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2), 243–255.

Crane, K., Livesu, M., Puppo, E., and Qin, Y. (2020). A survey of algorithms for geodesic paths and distances. *arXiv preprint arXiv:2007.10430*.

Davoodi, M., Faryadi, S., and Velni, J.M. (2020). A graph theoretic-based approach for deploying heterogeneous multi-agent systems with application in precision agriculture. *Journal of Intelligent & Robotic Systems*, 101(1), 1–15.

Davoodi, M. and Velni, J.M. (2020). Heterogeneity-aware graph partitioning for distributed deployment of multiagent systems. *IEEE Transactions on Cybernetics*.

De Berg, M., Van Kreveld, M., Overmars, M., and Schwarzkopf, O. (1997). Computational geometry. In *Computational geometry*, 1–17. Springer.

Ghosh, S.K. and Mount, D.M. (1991). An output-sensitive algorithm for computing visibility graphs. *SIAM Journal on Computing*, 20(5), 888–910.

Hameed, I.A. (2018). A coverage planner for multi-robot systems in agriculture. In *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 698–704. IEEE.

Hershberger, J. and Suri, S. (1993). Efficient computation of euclidean shortest paths in the plane. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, 508–517. IEEE.

Howard, A., Parker, L.E., and Sukhatme, G.S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6), 431–447.

Kantaros, Y., Thanou, M., and Tzes, A. (2015). Distributed coverage control for concave areas by a heterogeneous robot–swarm with visibility sensing constraints. *Automatica*, 53, 195–207.

LaValle, S.M. (2006). *Planning algorithms*. Cambridge university press.

Lee, S.K., Fekete, S.P., and McLurkin, J. (2014). Geodesic topological voronoi tessellations in triangulated environments with multi-robot systems. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3858–3865. IEEE.

Lee, S.K., Fekete, S.P., and McLurkin, J. (2016). Structured triangulation in multi-robot systems: Coverage, patrolling, voronoi partitions, and geodesic centers. *The International Journal of Robotics Research*, 35(10), 1234–1260.

Li, F. and Klette, R. (2011). *Euclidean Shortest Paths: Exact or Approximate Algorithms*. Springer Publishing Company, Incorporated.

Li, W. and Cassandras, C.G. (2005). Distributed cooperative coverage control of sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 2542–2547. IEEE.

Mitchell, J.S. (1996). Shortest paths among obstacles in the plane. *International Journal of Computational Geometry & Applications*, 6(03), 309–332.

Nourbakhsh, I.R., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S. (2005). Human-robot teaming for search and rescue. *IEEE Pervasive Computing*, 4(1), 72–79.

Pimenta, L.C., Kumar, V., Mesquita, R.C., and Pereira, G.A. (2008). Sensing and coverage for a network of heterogeneous robots. In *2008 47th IEEE conference on decision and control*, 3947–3952. IEEE.

Schwager, M., Rus, D., and Slotine, J.J. (2009). Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3), 357–375.

Stergiopoulos, Y., Thanou, M., and Tzes, A. (2015). Distributed collaborative coverage-control schemes for non-convex domains. *IEEE Transactions on Automatic Control*, 60(9), 2422–2427.

Thanou, M., Stergiopoulos, Y., and Tzes, A. (2013). Distributed coverage using geodesic metric for non-convex environments. In *2013 IEEE international conference on robotics and automation*, 933–938. IEEE.

Wang, B. (2010). *Coverage control in sensor networks*. Springer Science & Business Media.