Autonomous Robots (2018) 42:1731–1748
https://doi.org/10.1007/s10514-018-9713-x

Limited range spatial load balancing in non-convex environments using sampling-based motion planners

Beth Boardman[1] · **Troy Harden**[1] · **Sonia Martínez**[2]

Received: 15 February 2017 / Accepted: 2 February 2018 / Published online: 19 February 2018

Abstract
This paper analyzes the limited range, spatial load balancing problem for agents deployed in non-convex environments and subject to differential constraints which restricts how the agents can move. First, the (unlimited range) spatial load balancing problem is introduced and the minimization problem with area constraints is defined. Then, to extend the problem for limited ranges, two cost functions and a sub-partition are defined. The problems are then analyzed and the results prove the existence of a partition that satisfies the area constraints. The non-convex environment makes the problem difficult to solve in continuous-space. Therefore, a probabilistic roadmap is used to approximate agents' cells via a graph. A distributed algorithm is proven to converge to an approximate solution. Finally, the convergence of the algorithm is shown in simulation.

Keywords Spatial load balancing · Limited range · Multi-agent coverage

1 Introduction

This paper presents coverage algorithms that multiple agents can use to solve area-constrained, limited range, spatial load balancing problems when deployed in non-convex environments. By accounting for how the agents can move via differential constraints, a multi-robot system can more efficiently divide the load of acquiring information or of servicing tasks in a given environment in a fair way. However, non-convex environments and differential constraints generally make the description of assigned regions challenging. Thus, we aim to design and analyze algorithms based on region approximation via sampling which can reduce the communication or computation of agents effectively. Motivated by making the spatial load balancing algorithm distributed over a graph with limited communication ranges, limited traveling ranges for agents are employed.

The gradient-based descent Lloyd algorithm (1982) is the basis of many multi-agent coverage strategies. A limited sample of work building on this approach includes limited sensor footprints (Laventall and Cortés 2008), heterogeneous agents with different sensing radii (Stergiopoulos and Tzes 2011; Pimenta et al. 2008), non-holonomic agents (Kwok and Martínez 2010b; Enright et al. 2008; Savla et al. 2007), and power constraints (Kwok and Martínez 2010a). The area-constrained problem is studied in Cortés (2010), Patel et al. (2014) and Pavone et al. (2011), where a partition of the environment based on weights, is used. This partition, combined with an appropriately modified objective, is sufficient to force agents' regions to have the desired area. In Jiang and Zefran (2013), a sink node is used to aggregate information collected by all of the agents. When the communication cost is being minimized, the agents must all communicate with the sink node. These papers assume the agents have unlimited ranges and are deployed in a convex environment. In Mahboubi et al. (2014), the area of a convex region covered by sensors is maximized by minimizing the area uncovered in the agents' cell. Even though the algorithms have certain

This is one of several papers published in *Autonomous Robots* comprising the "Special Issue on Distributed Robotics: From Fundamentals to Applications".

This work was supported by Los Alamos National Laboratory and is approved for release under LA-UR-16-27817.

✉ Beth Boardman
bboardman@lanl.gov

Troy Harden
harden@lanl.gov

Sonia Martínez
soniamd@ucsd.edu

[1] Applied Engineering and Technology, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

[2] Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093, USA

distributed properties, they cannot, in general, implemented over a limited-range communication graph.

Closely related to this paper is the research on multi-agent coverage in non-convex environments. A first paper is Caicedo-Nunez and Zefran (2008), which applies a diffeomorphism to transform the non-convex environment into an almost convex one, where only a finite number of points have been subtracted. The coverage problem is then solved in the transformed environment and a solution is obtained via the inverse transformation. The diffeomorphism limits this algorithm to two-dimensional environments. Environments with polygonal obstacles are considered in Zhong and Cassandras (2011) and Breitenmoser et al. (2010). A solution to the multi-agent coverage problem for non-point robots in non-convex environments can be found in Pimenta et al. (2008). The approach to solve the non-convex multi-agent coverage problem taken by Kantaros et al. (2014) is to use visibility-based Voronoi diagrams while maximizing the coverage area. The authors of Kantaros et al. (2014) extend their work in Kantaros et al. (2015) to include heterogeneous sensing. Most of the above papers do not present solutions for agents subject to differential constraints. The papers that do account for differential constraints do not consider obstacles.

The following papers are for coverage in an unknown non-convex environment with agents not subject to differential constraints. A Voronoi partition and potential field are used in Renzaglia and Martinelli (2009) to find a solution to the coverage problem in non-convex environments with unknown obstacles. The authors of Bhattacharya et al. (2013) let the agents learn and build a gridded environment map which is used to determine which grid points belong to each agents' generalized Voronoi cells. This grid-based approach is limited to low dimensional spaces and differential constraints are difficult to handle in this manner. The paper Bhattacharya et al. (2014) is an extension of Bhattacharya et al. (2013) to Reimannian manifolds with non-convex boundaries.

This paper builds on the preliminary results presented in our previous conference papers Boardman et al. (2016) and Boardman et al. (2017), to present a unified solution to a more general problem, thus addressing the limitations of each paper. The first paper Boardman et al. (2016) introduces the GRAPH- BASED SPATIAL LOAD BALANCING (GSLB) algorithm for approximately solving a non-convex spatial load balancing problem. Under some conditions, the algorithm constructs a lower-approximation, based on a lower bound of the cost function, of sample assigned regions to enable decentralization and to simplify computations. These lower approximations are dropped in this manuscript; instead we consider limited ranges. The second paper (Boardman et al. 2017) develops a distributed algorithm for solving a continuous-space, limited range, spatial load balancing problem in convex environments. In this manuscript, we go beyond both formulations and consider a limited range, non-convex spatial load balancing problem for dynamically constrained agents. We approach this problem by employing locational optimization techniques that are based on generalized Voronoi partitions of the environment. Due to the obstacles and differential constraints, obtaining an exact description of the Voronoi partition can be difficult. Thus, we expand our objective by using an approximation of the agents' configuration space by means of a probabilistic roadmap star (PRM*).

More precisely, we start from a continuous-space version of the load balancing problem subject to a variable area constraint. With the objective of obtaining distributed algorithms, we introduce limited travel ranges and associated areas and mixed-type performance metric functions. To solve the problem, we introduce "sub-partitions" of the space dependent on a set of additive weights, ω, and agent positions, P. Then, we prove the existence of a set of weights that make the sub-partition satisfy the variable area constraint and thus solve the problem. Based on this result, we provide an update law for agents that allows them to converge to a set of such weights. The agent positions are updated using a gradient law aimed at decreasing the cost function with respect to position. We then define a class of deployment algorithms for solving the problem and show convergence to a solution for the area-type performance metric case.

Building on the continuous-space version, the graph-based algorithm builds a PRM* type graph to describe the cost of an agent subject to differential constraints moving between any two configurations in a known non-convex environment. In this way, the coverage regions are approximated by assigning each agent a subset of nodes from the PRM*. This subset of nodes is further exploited to estimate the coverage regions' corresponding areas. We provide a characterization of how the limited range algorithm is distributed over a communication graph for radially-unbounded cost functions. We provide an analysis of the algorithm's convergence as the number of nodes in the optimal probabilistic roadmap tends to infinity. Finally, we present a simulation of the convergence of agents whose costs to move is given by the Euclidean norm squared.

This paper is organized as follows. Section 2 contains notation and details on the Probabilistic Roadmap construction. The continuous-space (limited range) spatial load balancing is in Sect. 3 followed by the graph-based spatial load balancing in Sect. 4. Analysis for the distributed properties of the algorithm is in Sect. 5 and for the convergence of the algorithm is in Sect. 6. Then, Sect. 7 has the simulations of the limited and unlimited range GRAPH- BASED SPATIAL LOAD BALANCING algorithm. Finally, the conclusion and future work can be found in Sect. 8.

2 Preliminaries

To begin with, we introduce some notations. Let $X \subseteq \mathbb{R}^d$ be the d-dimensional configuration space of the agents and denote the obstacle space as X_{obs}. The free space, $Q = X \backslash X_{\text{obs}}$, is defined as the set of all collision-free agent configurations. In general, Q is a non-convex configuration space that is assumed to be simply connected by dynamic paths of a mobile robot so that all of Q is reachable by all agents. A sub-partition of a subset of Q, $\mathcal{W} = \{W_1, \ldots, W_n\}$, is a collection of n cells, $W_i \subset Q$, $i \in \{1, \ldots, n\}$, whose interiors are disjoint and whose union covers a subset $Q' \subseteq Q$. Denote the boundary of cell W_i as ∂W_i. Define the shared boundary between cells i and j as $\triangle_{ij} = W_i \cap W_j$ and denote the unshared boundary of cell i as $\Lambda_i = \partial W_i \backslash \big(\cup_{j \neq i} W_j \big)$. Finally, we let $\mathbf{1}_n = (1, \ldots, 1)^\top \in \mathbb{R}^n$ and $\mathbf{0}_n = (0, \ldots, 0)^\top \in \mathbb{R}^n$.

2.1 Optimal probabilistic roadmap building

This section briefly describes how to construct an asymptotically optimal probabilistic roadmap (PRM*), denoted by G, for a known environment. Note that the PRM* is limited to dynamics that can be solved with a two point boundary value problem. The graph G allows the agents to approximate the optimal path cost between two configurations in the graph as the sum of the costs of the edges defining the path. The edge cost is the cost an agent incurs while traveling between the nodes that define that edge, e.g. it can be given by length of the edge. The path cost is used in Problem 2 (Sect. 4) as an element of the cost functional being minimized. The graph G is composed of a set of nodes $\mathcal{N}_G$ and a set of edges $\mathcal{E}_G$ constructed in the same was as Karaman and Frazzoli (2011). A node $q \in \mathcal{N}_G$ is a sampled configuration from Q, and each edge, $e \in \mathcal{E}_G$, is an ordered pair, $e = (q_1, q_2)$, which corresponds to an optimal path in Q, satisfies all constraints, and has a cost $J_e(q_1, q_2)$. The cost $J(q_1, q_2)$ is assumed to be additive; given an optimal path from q_1 to q_2, and a node q' in that path, it holds that, $J(q_1, q_2) = J(q_1, q') + J(q', q_2)$. The additive assumption is used in Sect. 4.1.1. We denote the out neighbors of q in G as $\mathcal{N}_G^{\text{out}}(q) = \{q_j \in \mathcal{N}_G \mid (q, q_j) \in \mathcal{E}_G\}$.

Each iteration of the construction of G begins by taking a uniformly sampled random configuration from the free-space, $q_{\text{rand}} \in Q$. Next, all graph vertices that are within a ball centered at q_{rand} with radius, $r = \gamma_G (\log(m)/m)^{1/d}$, are determined. Here, γ_G is a fixed parameter, m is the number of vertices currently in $\mathcal{N}_G$, and d is the dimension of Q. Let the near vertices of q_{rand} be denoted by Q_{near}. If Q_{near} is empty, then the graph vertex that is closest to q_{rand} is added to Q_{near}. The least-cost paths, whose cost is $J_e(q_{\text{rand}}, q_{\text{near}})$, from q_{rand} to $q_{\text{near}} \in Q_{\text{near}}$ are determined; these are outgoing edges of q_{rand}. If the cost depends on direction, as is the case with differential constraints, then the least-cost path from q_{near} to q_{rand} is also determined, these are the incoming edges of q_{rand}. Each collision-free path, e, is added to $\mathcal{E}_G$ as an edge. The application of spatial load balancing requires that G be strongly connected; a necessary condition is that all $q \in \mathcal{N}_G$ must have both an outgoing and incoming edge, so that if an agent reaches that node it may also leave that node. A sufficient condition is to construct a graph which only allows $(q_1, q_2) \in \mathcal{E}_G$ if and only if $(q_2, q_1) \in \mathcal{E}_G$.

The free-space Q is discretized by G while maintaining an asymptotically optimal roadmap of the environment. Each node $q \in \mathcal{N}_G$ has an associated area, $\beta(q)$, which is calculated as follows. Let $\mathcal{N}_X = \mathcal{N}_G \cup \mathcal{N}_{\text{obs}}$, where $\mathcal{N}_{\text{obs}}$ is a set of configurations inside X_{obs}. Then, determine the Voronoi partition of X using $\mathcal{N}_X$. The $\beta(q)$ for each $q \in \mathcal{N}_G$ is the area of its associated cell in this partition. A description of the external boundary of X is needed and we assume this description is available. In this paper, nodes refer to the vertices in the graph, $q \in \mathcal{N}_G$, and not to the n agents that move in the environment.

3 Continuous-space spatial load balancing

The limited range spatial load balancing problem aims to find optimal locations for n agents, with positions $P = \{p_1, \ldots, p_n\}$, $p_i \in Q$, $i \in \{1, \ldots, n\}$, and region assignments $W_i \subseteq Q$, $i \in \{1, \ldots, n\}$, as described below. Let the optimal cost of robot i to move from configuration $q_1 \in Q$ to $q_2 \in Q$ when subject to the differential constraint, $\dot{p}_i = f(p_i, u_i)$ with control input u_i, be $J(q_1, q_2) \geq 0$. A probability density function, $\phi(q)$, defined over Q, $\phi : Q \to \mathbb{R}_{\geq 0}$, describes the likelihood of an event occurring at a configuration in Q.

Let $a_1, \ldots, a_n \in \mathbb{R}_{>0}$, be targeted cell areas that distribute the load of covering Q among the group of agents. The $a_i, i \in \{1, \ldots, n\}$, are such that $\sum_{i=1}^n a_i = \int_Q \phi(q) dq$, which can be understood as a full load-balancing condition. Ideally, we would like to achieve $\int_{W_i} \phi(q) dq = a_i$, for $i \in \{1, \ldots, n\}$, with the region assignment $\{W_i\}_{i=1}^n$. However, if the $\{W_i\}_{i=1}^n$ strictly satisfy $\cup_{i=1}^n W_i = Q' \subsetneq Q$, full load balancing will not be achievable. Because of this restriction, a new variable area constraint is defined,

$$a_i' = \frac{a_i \int_{Q'} \phi(q) dq}{\int_Q \phi(q) dq}, \quad i \in \{1, \ldots, n\},$$

which corresponds to load-balancing over the restricted space Q'. Of particular interest is the equal area case, $a_i = a_j$ for all i, j, which results in $a_i' = a_j'$, for all $i, j \in \{1, \ldots, n\}$. Note that, when $Q' = Q$, we recover the original, full load-balancing condition. Then, the objective is to find positions, p_i, and regions, $W_i \subseteq Q$, for $i \in \{1, \ldots, n\}$, that solve

the following minimization problem subject to the area and dynamic constraints,

Problem 1 *(Multicenter optimization problem with dynamic and area constraints)*

$$\begin{aligned} \min \quad & \mathcal{H}(P, \mathcal{W}) \\ s.t. \quad & p_i \in Q, \quad \dot{p}_i = f(p_i, u_i), \\ & a_i' = \int_{W_i} \phi(q) dq, \quad i \in \{1, \ldots, n\}. \end{aligned}$$

In other words, the n agents want to minimize a cost functional while making sure each agent's cell, W_i in the partition $\mathcal{W}$, has area a_i'. The agents cannot leave Q and must obey the differential constraints that define their movement. In what follows, we describe specific cost functions, $\mathcal{H}$, and types of subpartitions motivated by coverage control objectives.

3.1 Unlimited range agents in convex spaces

The solution to Problem 1 with limited ranges builds on the previous work in Cortés (2010), Patel et al. (2014) and Pavone et al. (2011) where Q is convex and the agents have unlimited range such that $\cup_{i=1}^n W_i = Q$. Then $a_i = a_i'$ and the cost function that is minimized takes the form,

$$\mathcal{H}^{\text{centroid}}(P, \mathcal{W}) = \sum_{i=1}^{n} \int_{W_i} J(p_i, q) \phi(q) dq.$$

The cost function $\mathcal{H}^{\text{centroid}}(P, \mathcal{W})$ quantifies the network performance and is called centroid because the agent positions that minimize it are the centroids of the cells. This is the problem solved in Cortés (2010).

For trivial first order dynamics, the results in Cortés (2010) state that, given a set of positions, P, there exists a weight assignment, ω, that makes a generalized weighted Voronoi partition, $\mathcal{V}^{\text{weighted}}(P, \omega; J)$, feasible and that this partition is the best among all the partitions $\mathcal{W}$ that satisfy the area constraint. Here, $\mathcal{V}^{\text{weighted}}(P, \omega; J) = \{V_i^{\text{weighted}}(\omega)\}_{i=1}^n$ is such that, for all $i \in \{1, \ldots, n\}$,

$$V_i^{\text{weighted}}(\omega) = \{q \in Q \mid J(p_i, q) - \omega_i \leq J(p_j, q) - \omega_j, \ \forall\, i \neq j\}.$$

The feasible set of weights is $U = \{\omega \in \mathbb{R}^n \mid |\omega_i - \omega_j| \leq J(p_i, p_j) \quad i, j \in \{1, \ldots, n\}\}$. If $\omega \notin U$, then at least one cell is empty. In convex environments, given a partition, the best agent positions are the centroids of their cells. For certain metrics, such as Euclidean metrics, these centroids are given by a closed-form formula. However, in non-convex environments multiple agent positions may minimize the cost function. These positions are referred to as a generalized centroid.

3.2 Limited ranges

When the agents have a limited travel range, referred to as limited range, a sub-partition, $\cup_{i=1}^n W_i \subset Q$, is found and $a_i \geq a_i'$. Here, limited travel range refers to the maximum distance an agent can travel from its final coverage configuration position. Since the area of the sub-partition changes as a function of agent position, the cost function being minimized is modified to account for the current area covered by the agents. This leads to a cost function that either maximizes the area covered by the regions,

$$\mathcal{H}^{\text{area}}(P, \mathcal{W}) = -\sum_{i=1}^{n} \int_{W_i} \phi(q) dq,$$

or combines $\mathcal{H}^{\text{centroid}}(P, \mathcal{W})$ and $\mathcal{H}^{\text{area}}(P, \mathcal{W})$ in the convenient form of,

$$\mathcal{H}^{\text{mixed}}(P, \mathcal{W}) = \sum_{i=1}^{n} \left(\int_{W_i} J(p_i, q) \phi(q) dq - k_i \int_{W_i} \phi(q) dq \right),$$

where $k_i \in \mathbb{R}_{>0}$, are constants, see Sect. 3.3.2 for a particular choice.

3.2.1 Limited range sub-partition

Define the limited ranges of the agents as the reachable sets, $\mathcal{D} = \{D_1, \ldots, D_n\}$, such that, for all $i \in \{1, \ldots, n\}$,

$$D_i = \{q \in Q \mid J(p_i, q) - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k \leq c\}, \tag{1}$$

where $c \in \mathbb{R}$ is the travel range of the agents and is a constant. When $\omega_i = \frac{1}{n} \sum_{k=1}^n \omega_k$, then $J(p_i, q) \leq c$ making c the upper bound, or limited range, on the cost $J(p_i, q)$. When $\omega_i \neq \frac{1}{n} \sum_{k=1}^n \omega_k$, those terms act as a perturbation on the limited range c. These perturbations are why the effective limited range for each agent can be different. There are certain properties of $\mathcal{V}^{\text{weighted}}$ from Cortés (2010) that the limited range sub-partition should maintain in order to obtain analogous results. One such property is that the cells, V_i^{weighted}, are invariant under translation of ω, $V_i^{\text{weighted}}(P, \omega; J) = V_i^{\text{weighted}}(P, \omega + t \cdot \mathbf{1}_n; J)$. This property leads to the area of V_i^{weighted} also being invariant under translations of ω. Including the mean of ω term allows the set D_i to be invariant under translation in ω. The limited range sub-partition is defined as $\mathcal{V}^{\text{LR}}(P, \omega, c) = \{V_i^{\text{LR}}\}_{i=1}^n$, such that, for all $i \in \{1, \ldots, n\}$, $V_i^{\text{LR}} = V_i^{\text{weighted}} \cap D_i$. An example of a limited range sub-partition can be found in Fig. 1. The shared boundaries are from V_i^{weighted} while the unshared arcs are from D_i. Note that one of the agents has no shared boundaries, so its cell $V_i^{\text{LR}} = D_i$. In this example, the D_i

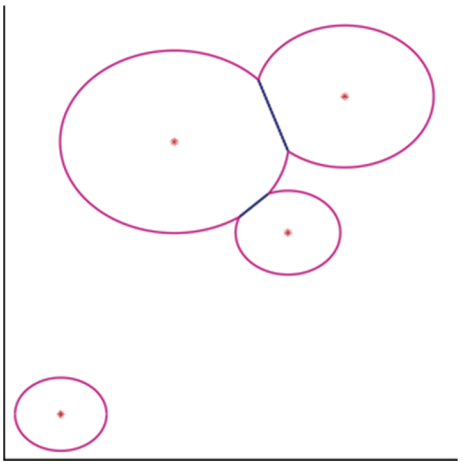

Fig. 1 An example of a limited range sub-partition for four agents, each with a different ω_i

boundaries are circular arcs because the cost is the Euclidean norm squared, $J(p_i, q) = \|p_i - q\|^2$.

The limited range sub-partition defines a graph which is used to describe the algorithm and its properties. This graph, $\mathcal{G}_{\text{LR}}(P, \omega) = (\mathcal{N}, \mathcal{E})$, has vertices, $v_i \in \mathcal{N}$, that correspond to the n agents. In this graph, $e = (v_i, v_j) \in \mathcal{E}$, between agents i and j, if and only if $V_i^{\text{LR}} \cap V_j^{\text{LR}} \neq \emptyset$. If agents i and j share an edge in $\mathcal{G}_{\text{LR}}(P, \omega)$, $(v_i, v_j) \in \mathcal{E}$, then agents i and j are neighbors. Let $\mathcal{N}_i$ denote the set of neighbors for agent i. When $Q' \neq Q$, $\mathcal{G}_{\text{LR}}(P, \omega)$ may not be connected. The edges of $\mathcal{G}_{\text{LR}}(P, \omega)$ change as the agent positions and weights are updated. Because D_i, and hence V_i^{LR}, is dependent upon all agent weights, $\mathcal{G}_{\text{LR}}(P, \omega)$ is not necessarily representative of the agents' communication graph. The remainder of the paper will abbreviate $\mathcal{G}_{\text{LR}}(P, \omega)$ as $\mathcal{G}_{\text{LR}}$. This graph is analyzed further in Sect. 5.2.

Let η be the number of connected components in $\mathcal{G}_{\text{LR}}$ and $\mathcal{G}_l$ be a single connected component of $\mathcal{G}_{\text{LR}}$ such that $\mathcal{G}_{\text{LR}} = \cup_{l=1}^{\eta} \mathcal{G}_l$ and $\mathcal{G}_l \cap \mathcal{G}_{l'} = \emptyset$ for all $l, l' \in \{1, \ldots, \eta\}$ and $l \neq l'$. Denote the number of vertices in $\mathcal{G}_l$ as n_l. Define the vector $\mathbf{v}_l \in \mathbb{R}^n$ such that the i^{th} entry of $\mathbf{v}_l$ is equal to one if agent i is in $\mathcal{G}_l$ and is zero otherwise. Note that $\{\mathbf{v}_l\}$ for $l \in \{1, \ldots, \eta\}$ form an orthogonal basis.

3.2.2 Existence and choice of weights

This section proves the existence of weights that allow $\mathcal{V}^{\text{LR}}$ to satisfy the variable area constraint. Recall, $\mathcal{V}^{\text{LR}}$ is invariant under translations in the weights and define the weights-to-area map as,

$$\mathcal{M}(P, \omega) = \left(\int_{V_1^{\text{LR}}(\omega)} \phi(q) dq, \ldots, \int_{V_n^{\text{LR}}(\omega)} \phi(q) dq \right).$$

For conciseness, in the following $V_i^{\text{LR}} = V_i^{\text{LR}}(\omega)$.

Lemma 1 *The weights-to-area map, $\mathcal{M}$, is the gradient, $\nabla F = -\mathcal{M}$, where $F : \mathbb{R}^n \to \mathbb{R}$,*

$$F(\omega) = \frac{-n}{1-n} \sum_{j=1}^{n} \int_{V_j^{\text{LR}}} (J(p_j, q) - \omega_j + \frac{1}{n} \sum_{k=1}^{n} \omega_k - c) \phi(q) dq.$$

The proof of Lemma 1, and all other proofs, can be found in the "Appendix".

The following equations show how to update an initial weight assignment, ω^0, to new weights that satisfy the area constraints.

Define $A \triangleq \sum_{i=1}^{n} \mathcal{M}_i(\omega^0)$ and

$$\bar{a}_i \triangleq A \frac{a_i}{\int_Q \phi(q) dq}. \tag{2}$$

In the equal area case, $a_i = a_j = \frac{1}{n} \int_Q \phi(q) dq$ and $\bar{a}_i = \frac{A}{n}$. Note that $\mathcal{M}_i(\omega^0) = \bar{a}_i$ does not necessarily hold. Define $U^0 = \left\{ \omega \in U \mid \sum_{i=1}^{n} \mathcal{M}_i(\omega) = \sum_{i=1}^{n} \mathcal{M}_i(\omega^0) \right\}$. Restrict $\mathcal{M}$ to $\omega \in U^0$, and denote this restriction as $\mathcal{M}^0$.

Lemma 2, which gives properties for the Jacobian of $\mathcal{M}^0$, is analogous to Prop. IV.2 from Cortés (2010) but has a more complicated proof due to the sub-partition, $\mathcal{V}^{\text{LR}}$. Note that $\mathcal{M}^0$ is a continuous function of ω.

Lemma 2 *Let $J(\mathcal{M}^0)$ denote the Jacobian matrix of $\mathcal{M}^0 : U^0 \subset \mathbb{R}^n \to \mathbb{R}^n$. Then*

1. *$J(\mathcal{M}^0)$ is symmetric;*
2. *Choose $\omega \in U^0$, and consider the graph $\mathcal{G}_{\text{LR}}(P, \omega)$ with η connected components and associated $\mathbf{1}_n$ and $\mathbf{v}_l$, for all $l \in \{1, \ldots, \eta\}$. Then, these are eigenvectors of $J(\mathcal{M}^0)(\omega)$ with eigenvalue 0;*
3. *The rank of $J(\mathcal{M}^0)(\omega)$ is $n - \eta$.*

From here, the existence of a weight assignment that satisfies the $\bar{a}_i$ constraint can be proven. The proof of Theorem 1 follows that of Prop. IV.4 from Cortés (2010) except that here $\mathcal{G}_{\text{LR}}$ is not necessarily connected.

Theorem 1 *Let $a_1, \ldots, a_n > 0$ such that $\sum_{i=1}^{n} a_i = \int_Q \phi(q) dq$ and let $p_1, \ldots, p_n \in Q$. Let there exist some initial weights, ω^0, such that $\mathcal{M}_i(\omega^0) > 0$ for each $i \in \{1, \ldots, n\}$. Let $\{\bar{a}_1, \ldots, \bar{a}_n\}$ be as defined in (2). Then there exists a set of weights $\omega = \{\omega_1, \ldots, \omega_n\}$ such that*

$$\int_{V_i^{\text{LR}}(P, \omega, c)} \phi(q) dq = \bar{a}_i, \quad i \in \{1, \ldots, n\}.$$

3.3 Continuous-space algorithm

This section describes the algorithm used to solve Problem 1. The algorithm alternately updates the agents' weights and

Voronoi cells until the area of the cells has converged to the desired areas. Then, the agents update their positions. Concisely, the algorithm dynamics can be expressed as,

$$\begin{pmatrix}\omega^+\\ P^+\end{pmatrix} = \psi\begin{pmatrix}\omega\\ P\end{pmatrix},$$

where ψ is a function combining (3) and (4). The details of the dynamic weight and position update are presented next.

3.3.1 Weights update

From Theorem 1 and Lemma 2, there exists weights for which $\mathcal{V}^{\mathrm{LR}}(\omega)$ satisfies the area constraints while maintaining constant area in each connected component of $\mathcal{G}_{\mathrm{LR}}$. Instead of maintaining constant areas by numerically solving for the n^{th} weight, the next procedure is followed in our algorithm. All the weights are iteratively updated so they eventually converge to ω^*, such that $V_i^{\mathrm{LR}}(P,\omega^*) = \bar{a}_i$, thus preserving, $\sum_{i=1}^n \mathcal{M}_i(P,\omega^0) = \sum_{i=1}^n \mathcal{M}_i(P,\omega^*)$. First, define $\mathcal{F}(\omega) = -F(\omega) - \sum_{i=1}^n \omega_i\bar{a}_i$, and set $\nabla\mathcal{F}(\omega) = g(\omega) = \mathbf{0}_n$, where $g(\omega) = \mathcal{M}(\omega_1,\ldots,\omega_n) - (\bar{a}_1,\ldots,\bar{a}_n) = \mathbf{0}_n$. The Jacobi algorithm, Bertsekas and Tsitsiklis (1997),

$$\omega^+ = \omega - \gamma\,\mathrm{diag}\left(\frac{\partial g_1}{\partial\omega_1},\ldots,\frac{\partial g_n}{\partial\omega_n}\right)^{-1} g(\omega), \tag{3}$$

is then used to find the ω values that optimize $\mathcal{F}$. The step size can be characterized as in Cortés (2010) Prop. IV.5 to guarantee convergence in the weights. More precisely, let $\mathcal{L}$ be a level set of $\mathcal{F}(\omega)$, and then, $\gamma \in (0,\ Y/B)$, where,

$$Y = \min_{i\in\{1,\ldots,n\}}\min_{\omega\in\mathcal{L}}\frac{\partial g_i(\omega)}{\partial\omega_i} > 0,\ B = \max_{i\in\{1,\ldots,n\}}\max_{\omega\in\mathcal{L}}\frac{\partial g_i(\omega)}{\partial\omega_i} > 0.$$

To implement this algorithm, the agents each need to compute their $g_i(\omega) = \mathcal{M}_i(\omega) - \bar{a}_i$ and $\frac{\partial g_i}{\partial\omega_i}$, where,

$$\frac{\partial g_i(\omega)}{\partial\omega_i} = \frac{\partial\mathcal{M}_i(\omega)}{\partial\omega_i} - \cancelto{0}{\frac{\partial\bar{a}_i}{\partial\omega_i}} = \int_{\Lambda_i}\hat{n}^\top\frac{\partial q}{\partial\omega_i}\phi(q)dq + \int_{\triangle_{ij}}\hat{n}^\top\frac{\partial q}{\partial\omega_i}\phi(q)dq.$$

3.3.2 Gradient function computation

The agents update their positions according to the derivative of $\mathcal{H}(P,\mathcal{V}^{\mathrm{LR}}(P,\omega,c))$ with respect to position to solve Problem 1. The gradient computation details can be found in Cortés et al. (2005). For a general $\mathcal{H}$, the dynamics for agent i are

$$p_i^+ = p_i - h\frac{\partial\mathcal{H}(P,\mathcal{V}^{\mathrm{LR}}(P,\omega,c))}{\partial p_i}, \tag{4}$$

where h is an appropriate step size, found via a line search. Eq. 4 only works for convex Q and when the agents are not subject to differential constraints, $\dot{p}_i = f(p_i,u_i)$. The graph-based algorithm is introduced specifically to handle these issues, see Sect. 4.2.2.

For the area only cost function, the gradient is

$$\frac{\partial\mathcal{H}^{\mathrm{area}}(P,\mathcal{V}^{\mathrm{LR}}(P,\omega,c))}{\partial p_i} = -\int_{\Lambda_i}\phi(q)\hat{n}^\top\frac{\partial q}{\partial p_i}dq. \tag{5}$$

Here, q are at the unshared boundary configurations, $q \in \Lambda_i = \partial V_i^{\mathrm{LR}} \cap D_i$, and $\hat{n}$ is the vector normal to the boundary at q. In other words, the agents move toward the (weighted) center of the unshared boundary of V_i^{LR}, and stay put when $\Lambda_i = \emptyset$.

When using $\mathcal{H}^{\mathrm{mixed}}(P,\mathcal{V}^{\mathrm{LR}})$, the right selection of k_i reduces the gradient computation to moving to a generalized centroid of V_i^{LR},

$$\frac{\partial\mathcal{H}^{\mathrm{mixed}}(P,\mathcal{V}^{\mathrm{LR}}(P,\omega,c))}{\partial p_i} = \int_{V_i^{\mathrm{LR}}}\frac{\partial J(p_i,q)}{\partial p_i}\phi(q)dq - \int_{\Lambda_i}J(p_i,q)\phi(q)\hat{n}^\top\frac{\partial q}{\partial p_i}dq + k_i\int_{\Lambda_i}\phi(q)\hat{n}^\top\frac{\partial q}{\partial p_i}dq.$$

For $J(p_i,q) = \|p_i - q\|^2$ or $J(p_i,q) = \|p_i - q\|$, choose $k_i = \mathcal{R}_i \triangleq c + \omega_i - \frac{1}{n}\sum_{k=1}^n\omega_k$.

4 Graph-based limited range spatial load balancing

This section details the graph-based version of Problem 1. As a preprocessing step, a PRM*, G, is constructed in the non-convex environment Q, see Sect. 2.1. The cost to travel between two configurations q_1 and q_2, $J(q_1,q_2)$, is approximated by the sum of edge costs of the best path in G from q_1 to q_2. Define a sub-partition of a subset of $\mathcal{N}_G$ as $\widetilde{\mathcal{W}} = \{\widetilde{W}_i\}_{i=1}^n$, such that $\cup_{i=1}^n\widetilde{W}_i \subseteq \mathcal{N}_G$ and $\widetilde{W}_i \cap \widetilde{W}_j = \emptyset$.

Let $\tilde{a}_1,\ldots,\tilde{a}_n \in \mathbb{R}_{>0}$, such that $\sum_{i=1}^n\tilde{a}_i = \sum_{q\in\mathcal{N}_G}\phi(q)\beta(q)$, then define the approximate variable area constraint as

$$\tilde{a}_i' = \frac{\tilde{a}_i\sum_{q\in\widetilde{\mathcal{W}}}\phi(q)\beta(q)}{\sum_{q\in\mathcal{N}_G}\phi(q)\beta(q)}, \qquad i\in\{1,\ldots,n\}.$$

The approximated area covered by $q \in \mathcal{N}_G$, $\beta(q)$, is precomputed as described in Sect. 2.1. The $\sum_{q\in\widetilde{\mathcal{W}}}\phi(q)\beta(q)$

requires knowledge from all agents and varies with each algorithm iteration.

The n agents solve graph-based Problem 2, which approximates the integral as a summation over a set of nodes.

Problem 2 *(Graph-based multicenter optimization problem with area constraints)*

$$\begin{aligned} \min \quad & \widetilde{\mathcal{H}}(P, \widetilde{\mathcal{W}}) \\ \text{s.t.} \quad & p_i \in \mathcal{N}_G, \\ & \tilde{a}_i' = \sum_{q \in \widetilde{W}_i} \phi(q)\beta(q), \quad i \in \{1, \ldots, n\}. \end{aligned}$$

Note that the differential constraint on p_i is removed because it is incorporated into G.

The cost function that the agents minimize is an approximation to either $\mathcal{H}^{\text{centroid}}$, $\mathcal{H}^{\text{area}}$, or $\mathcal{H}^{\text{mixed}}$, given as

$$\widetilde{\mathcal{H}}^{\text{centroid}}(P, \widetilde{\mathcal{W}}) = \sum_{i=1}^{n} \sum_{q \in \widetilde{W}_i} J(p_i, q)\phi(q)\beta(q),$$

$$\widetilde{\mathcal{H}}^{\text{area}}(P, \widetilde{\mathcal{W}}) = -\sum_{i=1}^{n} \sum_{q \in \widetilde{W}_i} \phi(q)\beta(q),$$

and

$$\widetilde{\mathcal{H}}^{\text{mixed}}(P, \widetilde{\mathcal{W}}) = \sum_{i=1}^{n} \Big(\sum_{q \in \widetilde{W}_i} J(p_i, q)\phi(q)\beta(q) - k_i \sum_{q \in \widetilde{W}_i} \phi(q)\beta(q) \Big).$$

To solve Problem 2 algorithmically, each agent has a copy of G and it is assumed that the agents know the positions P and the weights ω of the other agents by communicating with each other to interchange this information. The assumption on P and ω can be relaxed in some cases so that it is only necessary to know the positions and weights of a subset of the other agents; this relaxation will be discussed further in Sect. 5.

4.1 Approximate general Voronoi tessellations

Different options are considered for an *approximate generalized Voronoi partition*. For conciseness, $\widehat{\mathcal{V}}$ is used to represent all approximated Voronoi partitions. In other words, if a results pertains to all the approximate Voronoi partitions, we use $\widehat{\mathcal{V}}$ (e.g. $\widehat{\mathcal{V}} = \widetilde{\mathcal{V}}^{\text{weighted}}$ or $\widehat{\mathcal{V}} = \widetilde{\mathcal{V}}^{\text{LR}}$ in the following). When the agents have unlimited range, such that $\cup_{i=1}^{n} W_i = \mathcal{N}_G$, they find the weighted approximate Voronoi partition, $\widetilde{\mathcal{V}}^{\text{weighted}} = \{\widetilde{V}_i^{\text{weighted}}\}_{i=1}^{n}$,

$$\begin{aligned} \widetilde{V}_i^{\text{weighted}} = \{ & q \in \mathcal{N}_G \mid J(p_i, q) - \omega_i \\ & \leq J(p_j, q) - \omega_j, \quad \forall\, j \neq i \}. \end{aligned}$$

Here, $J(p_i, q)$ is the minimum sum of the edge costs of the optimal path in G defined from p_i to q. Due to the random selection of q when building G, the probability that one node belongs to two different cells of $\widehat{\mathcal{V}}$ is zero.

The limited range agents find a sub-partition, $\cup_{i=1}^{n} W_i \subset \mathcal{N}_G$, defined as a limited range Voronoi sub-partition $\widetilde{\mathcal{V}}^{\text{LR}} = \{\widetilde{V}_i^{\text{LR}}\}_{i=1}^{n}$, $\widetilde{V}_i^{\text{LR}} = \widetilde{V}_i^{\text{weighted}} \cap \widetilde{D}_i$, where

$$\widetilde{D}_i = \left\{ q \in \mathcal{N}_G \mid J(p_i, q) - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k \leq c \right\}.$$

In order to calculate the approximation of its own cell, $\widehat{V}_i$, agent i does a Dijkstra graph search, Dijkstra (1959), starting from its current configuration, p_i, and keeps a queue of the vertices it needs to check. To start with, p_i is added to $\widehat{V}_i$, and all the outgoing neighboring nodes of p_i are added to the queue. The agent then takes one of the nodes, q_{check}, from the queue and checks to see if it is part of $\widehat{V}_i$. If q_{check} is a part of $\widehat{V}_i$ then its outgoing neighboring nodes are added to the queue. If q_{check} is not added to $\widehat{V}_i$, then its neighbors are not added to the queue, see Proposition 1 for the result on why this approach is correct. Agent i constructs $\widehat{V}_i$ until the queue is empty.

4.1.1 Properties of $\widehat{\mathcal{V}}$

For $\widetilde{\mathcal{V}}^{\text{weighted}}$ the weights need to belong to $U = \{\omega \in \mathbb{R}^n \mid |\omega_i - \omega_j| \leq J(p_i, p_j) \quad i, j \in \{1, \ldots, n\}\}$. If $\omega \notin U$ then at least one cell is empty. Since $\widetilde{\mathcal{V}}^{\text{LR}}$ is a subset of $\widetilde{\mathcal{V}}^{\text{weighted}}$ then the weights must also belong to the set U.

Lemma 3 gives a lower bound on the constant c for a general $J(p_i, q)$ so that $\partial \widetilde{V}_i^{\text{weighted}} \cap \partial \widetilde{D}_i \neq \emptyset$ for each i. If the initial agent conditions lead to $\partial \widetilde{V}_i^{\text{weighted}} \cap \partial \widetilde{D}_i = \emptyset$ for all i, then, assuming that $\widetilde{\mathcal{V}}^{\text{LR}}$ satisfies the area constraint, Problem 2 is trivially satisfied.

Lemma 3 *Assume the triangle inequality holds for $J(p_i, q)$. If $\partial \widetilde{V}_i^{\text{weighted}} \cap \partial \widetilde{D}_i \neq \emptyset$, then $c \geq \frac{J(p_i, p_j) + \frac{2}{n} \sum_{k=1}^{n} \omega_k + \omega_i - \omega_j}{2}$.*

A tighter bound can be found for particular $J(p_i, q)$. Lemmas 4 and 5 compute such bounds.

Lemma 4 *Let $J(p_i, q) = \|p_i - q\|^2$. If $\partial \widetilde{V}_i^{\text{weighted}} \cap \partial \widetilde{D}_i \neq \emptyset$, then $c \geq \frac{(\|p_i - p_j\|^2 + \omega_i - \omega_j)^2}{4\|p_i - p_j\|^2} - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k$.*

Lemma 5 *Let $J(p_i, q) = \|p_i - q\|$. If $\partial \widetilde{V}_i^{\text{weighted}} \cap \partial \widetilde{D}_i \neq \emptyset$ then $c \geq \frac{(\|p_i - p_j\| + \omega_i - \omega_j)}{2} - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k$.*

The proof of Lemma 5 is similar to that of Lemma 4, and therefore omitted for brevity.

Algorithm 1 $(P^*, \widehat{\mathcal{V}}(P^*, \omega^*; J)) \leftarrow \text{GSLB}(P_0, \omega_0, \widehat{\mathcal{V}}, Q)$

$G \leftarrow \text{PRM}^*(Q)$;
$(P, \omega) \leftarrow \text{Initialize}(P_0, \omega_0)$;
for all $\{\text{Agent } i\}_{i=1}^n$ **do**
  while $P \neq P^+$ **do**
    $P = P^+$
    $\widehat{V}_i(P, \omega; J) \leftarrow \text{VoronoiPartition}(P, \omega, c, G)$;
    $\tilde{A} \leftarrow \text{getArea}(\widehat{\mathcal{V}}(P, \omega))$;
    while $\|\omega - \omega^+\| > \text{error}$ **do**
      $\omega = \omega^+$;
      $\omega_i^+ \leftarrow \text{UpdateWeights}(P, \omega, \widehat{V}_i, \tilde{A}, G)$;
      $\omega^+ \leftarrow \text{TransmitAndReceive}(\omega_i^+)$;
      $\widehat{V}_i(P, \omega; J) \leftarrow \text{VoronoiPartition}(P, \omega, c, G)$;
    end while
    $p_i^+ \leftarrow \text{UpdateAgentPosition}(p_i, \widehat{V}_i, G)$;
    $P^+ \leftarrow \text{TransmitAndReceive}(p_i^+)$;
  end while
end for
return $(P, \widehat{\mathcal{V}}(P, \omega; J))$;

The additive property of J allows the agents to only check a connected subset of nodes, $q \in \mathcal{N}_G$, to find the approximated regions $\widetilde{V}_i^{\text{weighted}}$.

Proposition 1 *Assume that J is an additive cost and let there exist an optimal path from p_i to q passing through q'. Then, if q' is not part of $\widetilde{V}_i^{\text{weighted}}$ then q is not part of $\widetilde{V}_i^{\text{weighted}}$.*

Note that $\widetilde{\mathcal{V}}^{\text{LR}}$ may not be connected but still only needs to check the same q as $\widetilde{\mathcal{V}}^{\text{weighted}}$.

Another useful property of $\widetilde{\mathcal{V}}^{\text{weighted}}$ and $\widetilde{\mathcal{V}}^{\text{LR}}$ is that they are invariant under translation in the weights, $\widehat{\mathcal{V}}(\omega + t\mathbf{1}) = \widehat{\mathcal{V}}(\omega)$, which follows from the invariance property of their continuous-space counterparts.

4.2 Discrete-space algorithm

Algorithm 1 briefly outlines the general algorithm procedure for Problem 2 with limited ranges that leads to an approximate solution. Note that $\widehat{\mathcal{V}}$ refers to a generic approximation of a Voronoi sub-partition in terms of graph nodes; we refer to $\widehat{\mathcal{V}}$ as approximate generalized Voronoi partitions. First, the agents each determine such a partition, (see Sect. 4.1 for definition and details). Then the weights are updated to reflect the error in the area constraint, the details of which are in Sect. 4.2.1. These two steps are alternated until the area constraints are satisfied to within a specified error, which can be reduced by increasing the number of nodes in G. Next, the agents move to a neighboring node that will decrease $\widetilde{\mathcal{H}}$, see Sect. 4.2.2. The steps are repeated until none of the agents are able to update their positions, $P = P^+$.

4.2.1 Updating the agent weights

Recall, for Problem 2, each agent's cell should satisfy an approximate area constraint, $\tilde{a}_i'$. When the agents have unlimited ranges, $\tilde{a}_i' = \tilde{a}_i$. Agents with limited range follow the procedure outlined in Sect. 3.2.2 to find a weight assignment for $\widetilde{\mathcal{V}}^{\text{LR}}$. Let ω_0 be the set of weights, then define $\tilde{A} = \sum_{q \in \widetilde{\mathcal{V}}^{\text{LR}}(\omega_0)} \phi(q)\beta(q)$. The new variable area constraint is $\bar{a}_i = \frac{\tilde{a}_i \tilde{A}}{\sum_{q \in \mathcal{N}_G} \phi(q)\beta(q)}$. Let $\hat{a}_i$ indicate either $\tilde{a}_i$ or $\bar{a}_i$.

Define the error between the current and specified area as

$$\widetilde{g}(\omega) = \left(\sum_{q \in \widehat{V}_1(\omega)} \phi(q)\beta(q) - \hat{a}_1, \ldots, \sum_{q \in \widehat{V}_n(\omega)} \phi(q)\beta(q) - \hat{a}_n \right).$$

Next, each agent updates ω to reduce the area error.

From Bertsekas and Tsitsiklis (1997), the Jacobian update used to minimize $\widetilde{g}(\omega)$ is approximated as

$$\omega_i^+ = \omega_i - \gamma \left(\frac{\partial \widetilde{g}(\omega)}{\partial \omega_i} \right)^{-1} \widetilde{g}_i(\omega),$$

which converges for a small enough $\gamma > 0$. The partial derivatives of $\widetilde{g}$ are approximated as,

$$\frac{\partial \widetilde{g}}{\partial \omega_i}(\omega) \approx \frac{\sum_{q \in \widehat{V}_i^i} \phi(q)\beta(q) - \sum_{q \in \widehat{V}_i} \phi(q)\beta(q)}{\Delta\omega_i}, \tag{6}$$

where $\widehat{V}_i^i$ is the Voronoi cell for agent i with $\omega = \{\omega_1, \ldots, \omega_i + \Delta\omega_i, \ldots, \omega_n\}$. Note that $\Delta\omega_i > 0$ needs to be small enough to guarantee convergence but also large enough that $\widehat{V}_i^i \neq \widehat{V}_i$. The $\widehat{V}_i^i$ can be computed with a single Dijkstra graph search so agent i can easily compute (6).

The algorithm loops through determining $\widehat{\mathcal{V}}$ and updating ω until the area constraint is satisfied to within a specified error.

4.2.2 Updating the agent positions

After $\widehat{V}_i$ and ω have been determined, agent i decides where to move. Ideally, each agent minimizing $\widetilde{\mathcal{H}}^{\text{centroid}}$ or $\widetilde{\mathcal{H}}^{\text{mixed}}$ would move to a position in the generalized centroid set on $\widehat{V}_i$, which is computationally intensive. Instead, agent i moves in the direction of one of the generalized centroids by moving to a neighboring node of p_i such that

$$p_i^+ \in \text{argmin}_{p \in \mathcal{N}_G^{\text{out}}(p_i)} \sum_{q \in \widehat{V}_i} J(p, q)\phi(q)\beta(q).$$

If the agents have limited range, and are solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{area}}$, they update their position by approximating (5),

$$\frac{\partial \widetilde{\mathcal{H}}^{\text{area}}(P, \widetilde{\mathcal{V}}^{\text{LR}})}{\partial p_i} = \sum_{i=1}^{n} \sum_{q \in \lambda_i} \phi(q)\hat{n}^\top(q) \frac{\partial q}{\partial p_i}.$$

The agent then moves to a neighboring node in the graph that is in a direction as close as possible to $\frac{\partial \widetilde{\mathcal{H}}^{\text{area}}(P,\widetilde{\mathcal{V}}^{\text{LR}})}{\partial p_i}$.

Note that because the agent is moving to another node in the graph, the new agent position will automatically be in Q. The new agent position is shared among the Voronoi neighbors of agent i, who need it to calculate their Voronoi cell. The algorithm loops through these steps until the agents' positions become fixed; indicating that convergence has been reached. Note that when iterating these steps in the continuous-space version, Problem 1, agents are locally minimizing the functional with respect to their positions.

5 Distributed algorithm properties

This section looks at the distributed nature of the GRAPH-BASED SPATIAL LOAD BALANCING algorithm using $\widetilde{\mathcal{V}}^{\text{weighted}}$ and $\widetilde{\mathcal{V}}^{\text{LR}}$. The following discussion focuses on the discrete-space case, but analogous considerations hold for the continuous-space counterpart.

The information that agent i needs for the implementation of the GRAPH- BASED SPATIAL LOAD BALANCING using $\mathcal{V}^{\text{weighted}}$ is limited to those other agents j whose approximated regions are connected to the approximated region of i via boundary nodes (or $V_i^{\text{weighted}}(P,\omega;J) \cap V_j^{\text{weighted}}(P,\omega;J) \neq \emptyset$ in the continuous-space counterpart). In the case where no obstacles are present, this property generally involves a limited number of agents which depends on the specific metric. In the Euclidean metric case, where, for equal weights, the generic number of neighbors is six (Okabe et al. 2000). When obstacles are present, this characterization is more difficult. The distributed properties of the GRAPH- BASED SPATIAL LOAD BALANCING using $\widetilde{\mathcal{V}}^{\text{LR}}$ are discussed below, but first an alternate definition of $\widetilde{\mathcal{D}}$ using a maximum radius is introduced.

5.1 Alternate definition of $\widetilde{\mathcal{D}}$

Under certain costs, $J(p_i,q)$, D_i can be defined as a ball with radius R_i. This definition is intuitive in a way that the c definition, (1), is not. Assume $J(p_i,q) = \|p_i - q\|^2$ or $J(p_i,q) = \|p_i - q\|$, then the radius of the ball defined by $\widetilde{D}_i$ is denoted by R_i for all $i \in \{1,\ldots,n\}$. Recall $\mathcal{R}_i \triangleq c + \omega_i - \frac{1}{n}\sum_{k=1}^n \omega_k$, then, using the definition of D_i at the boundary, $J(p_i,q) = \|p_i - q\|^2$ implies $R_i = \mathcal{R}_i^{1/2}$ and $J(p_i,q) = \|p_i - q\|$ implies $R_i = \mathcal{R}_i$.

Depending upon the physical system, an upper bound may be imposed on R_i, denoted by $R_{\max}$. We want to remove c from the definition of D_i and replace it with the fixed $R_{\max}$. To remove c, let c be a function of $R_{\max}$ and ω, instead of a constant. Then, define $c_{\max} = R_{\max} - \max_k(\omega) + \frac{1}{n}\sum_{k=1}^n \omega_k$. Substituting $c_{\max}$ into $\mathcal{R}_i \triangleq c + \omega_i - \frac{1}{n}\sum_{k=1}^n \omega_k$, gives,

$$\begin{aligned}\mathcal{R}_i &= R_{\max} - \max_k(\omega) + \frac{1}{n}\sum_{k=1}^n \omega_k + \omega_i - \frac{1}{n}\sum_{k=1}^n \omega_k,\\ &= R_{\max} - \max_k(\omega) + \omega_i.\end{aligned}$$

Notice that now $\mathcal{R}_i$, and hence R_i, are no longer dependent on c nor the mean of ω, but on $R_{\max}$ and the maximum value of ω. The new definition of D_i is,

$$D_i = \{q \in Q \mid J(p_i,q) \leq R_{\max} - \max_k(\omega) + \omega_i\}, \tag{7}$$

when $J(p_i,q) = \|p_i - q\|$, and is

$$D_i = \{q \in Q \mid J(p_i,q) \leq \sqrt{R_{\max} - \max_k(\omega) + \omega_i}\},$$

when $J(p_i,q) = \|p_i - q\|^2$. Algorithm 1 needs some minor modifications to account for the maximum radius constraint. First, in Lines 6 and 12, the primitive VoronoiPartition now takes inputs $R_{\max}$ and R_i instead of c. Then, after Line 11, R_i is determined.

Let D_i denote the continuous-space counterpart of $\widetilde{D}_i$. Then defining D_i using $R_{\max}$ causes problems in Lemma 1 because $\max_k \omega$ is not differentiable. However, given the max operator's properties, one can conjecture that an analogous result exists using generalized gradients. As a consequence, assuming the analogous result leads to $\mathcal{M}(\omega)$ being gradient (i.e. that the weights-to-area map is in the generalized gradient of F,) then all other results follow. In particular, in Lemma 2, the new D_i still satisfies the conditions on the partial derivatives. With the assumption that Lemma 1 holds, a weight assignment exists that satisfies the area constraint. Then, the convergence result in Theorem 2 still holds for the D_i definition using $R_{\max}$, (7).

5.2 Distributed properties using $\widetilde{\mathcal{V}}^{\text{LR}}$

While the algorithm in Cortés (2010) for solving the spatial load balancing problem is distributed in the sense that only information is needed for neighboring agents, these neighboring agents may be significantly far away from one another. Especially when Euclidean norms are used, the limited range constraint forces the agents to only consider neighbors within a certain distance of each other.

More precisely, the computation of $\widetilde{V}_i^{\text{LR}}(P,\omega)$ requires knowledge of the positions and weights of agent i's neighbors and knowledge of the mean of the weights for $\widetilde{D}_i$. The latter can be computed using a distributed consensus algorithm performed over a *connected* communication graph.

This communication graph might not be $\mathcal{G}_{\text{LR}}$, since $\mathcal{G}_{\text{LR}}$ is not necessarily connected. If $\widetilde{D}_i$ is defined as in Sect. 5.1, the agents need the maximum ω value instead of the mean. The maximum value is found using a max operation over a connected system, requiring fewer communications between the agents.

Before each weight update, the area covered by $\widetilde{\mathcal{V}}^{\text{LR}}$ needs to be determined. Again, a distributed consensus algorithm over a connected communication graph is needed. Once inside the weights update loop, agents only need the information from their cell, $\widetilde{V}_i^{\text{LR}}$, to determine ω_i^+. Likewise, the position update using the gradient of $\widetilde{\mathcal{H}}^{\text{area}}(P, \widetilde{\mathcal{V}}^{\text{LR}})$ or the centroid of $\widetilde{V}_i^{\text{LR}}$ for $\widetilde{\mathcal{H}}^{\text{mixed}}(P, \widetilde{\mathcal{V}}^{\text{LR}})$, only requires knowledge from the agent's own cell. In all, the algorithm is distributed over the smallest connected graph containing $\mathcal{G}_{\text{LR}}$.

The $R_{\max}$ constraint can be used to define a disk graph $\mathcal{G}_{2R_{\max}}$ that is sufficient for agents to determine neighbors in $\mathcal{G}_{\text{LR}}$. When using a general J, the balls are defined as a reachable set. If J is radially unbounded, the balls are compact sets. In the Euclidean metric case, the balls are circles whose radii are related to $R_{\max}$ and correspond to the standard r-disk graph. Define $\mathcal{G}_{2R_{\max}}$ over the set of agents, where a (communication) edge exists between agents i and j if and only if the balls centered at the agents' position with radius $R_{\max}$ intersect, $\mathbb{B}(p_i, R_{\max}) \cap \mathbb{B}(p_j, R_{\max}) \neq \emptyset$. The $\mathcal{G}_{2R_{\max}}$ can be used to determine an over approximation of the sets of neighboring agents in $\mathcal{G}_{\text{LR}}$. To see this approximation, note that $\widetilde{V}_i^{\text{LR}} \subseteq \widetilde{D}_i \subseteq \mathbb{B}(p_i, R_{\max})$, for all $i \in \{1, \ldots, n\}$. Therefore, $\widetilde{V}_i^{\text{LR}} \cap \widetilde{V}_j^{\text{LR}} \neq \emptyset$ only if $\widetilde{D}_i \cap \widetilde{D}_j \neq \emptyset$, which happens only if $\mathbb{B}(p_i, R_{\max}) \cap \mathbb{B}(p_j, R_{\max}) \neq \emptyset$. This result implies that agent i can compute its cell, $\widetilde{V}_i^{\text{LR}}$, communicating only with neighbors j in $\mathcal{G}_{2R_{\max}}$. In other words, an agent only needs to communicate with other agents within a distance of $2R_{\max}$ of itself, $\|p_i - p_j\| \leq 2R_{\max}$.

6 Convergence

The spatial load balancing algorithm in Cortés (2010) is shown to converge to a $(P^*, \mathcal{V}^{\text{weighted}}(P^*, \omega^*; J))$ solution of Problem 1 for convex environments. A similar result holds for non-convex Q since there exists ω that allows a generalized Voronoi partition $\mathcal{V}^{\text{weighted}}(P, \omega; J)$ to satisfy the constraints. Lemma 6 is used to extend Prop. IV.4 from Cortés (2010) to non-convex Q; Prop. IV.4 sates there exists a set of weights that make $\mathcal{V}^{\text{weighted}}$ satisfy the area constraints in a convex continuous-space.

Lemma 6 *Let $\mathcal{V}^{\text{weighted}}$, $J(p_i, q)$, $\phi(q)$, and ω be defined as above. Define the weights-to-area map as*

$$\mathcal{M}(P, \omega) = \left(\int_{V_i^{\text{weighted}}(\omega)} \phi(q)dq, \ldots, \int_{V_n^{\text{weighted}}(\omega)} \phi(q)dq \right).$$

Then, $\mathcal{M}$ is gradient, $\nabla F = -\mathcal{M}$, where $F : \mathbb{R}^n \to \mathbb{R}$,

$$F(\omega) = \sum_{j=1}^{n} \int_{V_j^{\text{weighted}}(\omega)} (J(p_j, q) - \omega_j)\phi(q)dq.$$

The proof of Lemma 6 is similar to that of Lemma 1 and therefore omitted.

For the discrete case note that for any $\widetilde{\mathcal{W}} = \{\widetilde{W}_i\}_{i=1}^n$ partition of $\mathcal{N}_G$, one can find (many) continuous-space partitions $\mathcal{W} = \{W_i\}_{i=1}^n$ such that $\widetilde{W}_i = W_i \cap \mathcal{N}_G$ and $\mathcal{W}$ satisfies the constraints. As the number of nodes in $\mathcal{N}_G$ goes to infinity, $\widetilde{\mathcal{W}}$ will converge to a $\mathcal{W}$. Due to integration properties, and because $\mathcal{H}^{\text{centroid}}(P, \mathcal{W})$ is continuous,

$$|\mathcal{H}^{\text{centroid}}(P, \mathcal{W}) - \widetilde{\mathcal{H}}^{\text{centroid}}(P, \widetilde{\mathcal{W}})| \leq \epsilon, \tag{8}$$

is true for a sufficiently small sample dispersion.

The GRAPH- BASED SPATIAL LOAD BALANCING algorithm with $\mathcal{V}^{\text{weighted}}$ cannot converge to the exact partition and centroids of the continuous problem, but an approximate solution is guaranteed, see Theorem 2.

Theorem 2 *The unlimited range* GRAPH- BASED SPATIAL LOAD BALANCING *algorithm is guaranteed to converge to an approximate solution $(P^*, \widetilde{\mathcal{V}}^{\text{weighted}}(P^*, \omega^*))$ which is in a continuous-space set defined by $\|\mathcal{H}^{\text{centroid}}(P^*, \mathcal{V}^{\text{weighted}})(P, \omega) - \mathcal{H}^{\text{centroid}}(P^*, \mathcal{V}^{\text{weighted}}(P^*, \omega^*))\| \leq 2\epsilon$.*

Proving convergence of the limited range GRAPH- BASED SPATIAL LOAD BALANCING relies on the continuous-space convergence Theorem 3 which is only shown to converge for $\mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}})$.

Theorem 3 *The continuous-space version of Algorithm 1, used to solve the limited range spatial load balancing problem, converges to a solution $(P^*, \mathcal{V}^{\text{LR}}(P^*, \omega^*))$ when $\mathcal{H} = \mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}})$.*

Then, convergence to an approximate solution is guaranteed by Lemma 7.

Lemma 7 *The limited range* GRAPH- BASED SPATIAL LOAD BALANCING *using $\widetilde{\mathcal{H}}^{\text{area}}$ is guaranteed to converge to an approximate solution $(P^*, \widetilde{\mathcal{V}}^{\text{LR}}(P^*, \omega^*))$ which is in a continuous-space set defined by $\|\mathcal{H}^{\text{area}}(P^*, \mathcal{V}^{\text{LR}}(P), \omega) - \mathcal{H}^{\text{area}}(P^*, \mathcal{V}^{\text{LR}}(P^*, \omega^*))\| \leq 2\epsilon$.*

The proof of Lemma 7 is similar to that of Theorem 2 and therefore omitted.

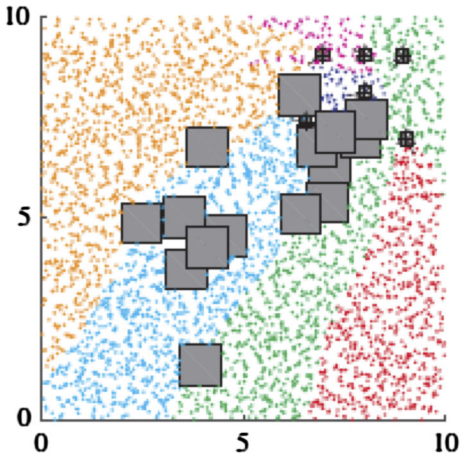

Fig. 2 The initial 5000 node graph $\widetilde{\mathcal{V}}^{\text{weighted}}$ for six agents before solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{centroid}}$

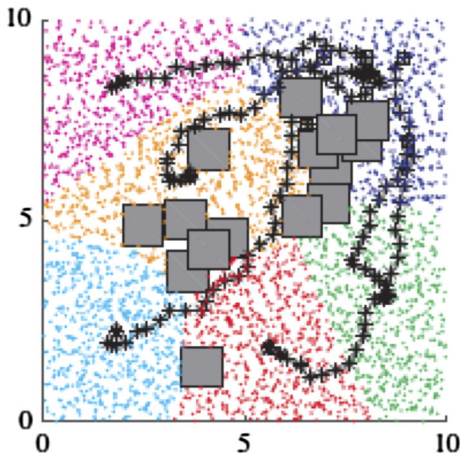

Fig. 3 The final 5000 node graph $\widetilde{\mathcal{V}}^{\text{weighted}}$ for six agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{centroid}}$

7 Simulations

The simulations below examine agents without differential constraints whose edge cost, J_e, is Euclidean distance and for Dubins vehicle agents whose edge cost is the distance traveled. There are simulations for six agents each of them solving graph-based Problem 2 with an equal area constraint and a uniform probability density function, $\phi(q) = 1, \forall\, q \in \mathcal{N}_G$. Each agent is initialized with $\omega_i = 5$. The simulations compare the limited range defined by a constant $c = 3$ and $R_{\max} = 3.5$.

All simulations have the same initial agent positions; clustered together in the top right corner. The simulations were run for a graph constructed with 2000 samples and a more dense graph constructed using 5000 samples. The graph should be constructed such that the edge lengths are less than $\partial \widetilde{D}_i$. This will ensure that there exists neighboring nodes of p_i within $\widetilde{V}_i^{\text{LR}}$ for the agent to move to.

7.1 Voronoi graph partitions

This section compares the weighted Voronoi graph partition, $\widetilde{\mathcal{V}}^{\text{weighted}}$ for six agents solving Problem 2. Figures 2 and 3 are the initial and final $\widetilde{\mathcal{V}}^{\text{weighted}}$ partitions, respectively, in the non-convex environment using the graph with 5000 samples.

The initial and final limited range Voronoi graph partitions for agents solving Problem 2 with $\widetilde{\mathcal{H}} = \widetilde{\mathcal{H}}^{\text{area}}$ in the non-convex environment using the 5000 sample graph are in Figs. 4, 5 and 6. Figure 5 is the final $\widetilde{\mathcal{V}}^{\text{LR}}$ partition when $c = 3$. When $R_{\max} = 3.5$, the agents' final $\widetilde{\mathcal{V}}^{\text{LR}}$ partition is in Fig. 6. The final $\widetilde{\mathcal{V}}^{\text{LR}}$ partitions for the 5000 sample graph in the non-convex environment under $\widetilde{\mathcal{H}} = \widetilde{\mathcal{H}}^{\text{Mixed}}$ are Fig. 7 when $c = 3$, and Fig. 8 when $R_{\max} = 3.5$.

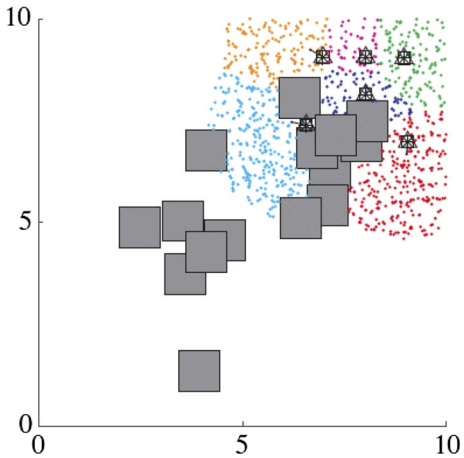

Fig. 4 The Initial 5000 node graph $\widetilde{\mathcal{V}}^{\text{LR}}$ with $c = 3$ for six agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{area}}$

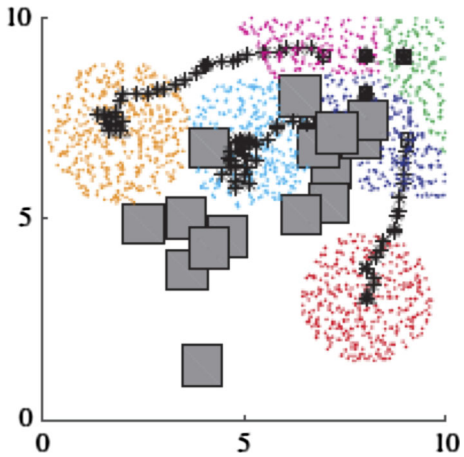

Fig. 5 The final 5000 node graph $\widetilde{\mathcal{V}}^{\text{LR}}$ with $c = 3$ for six agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{area}}$

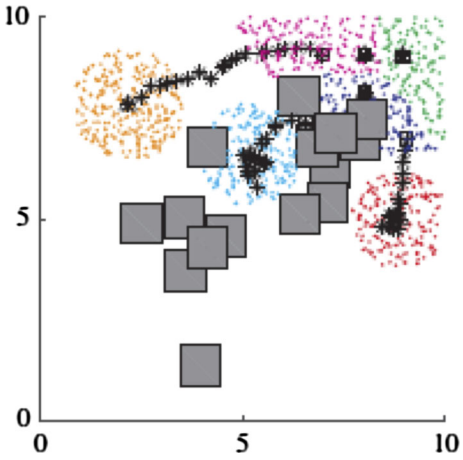

Fig. 6 The final 5000 node graph $\widetilde{\mathcal{V}}^{\mathrm{LR}}$ with $R_{\max} = 3.5$ for six agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\mathrm{area}}$

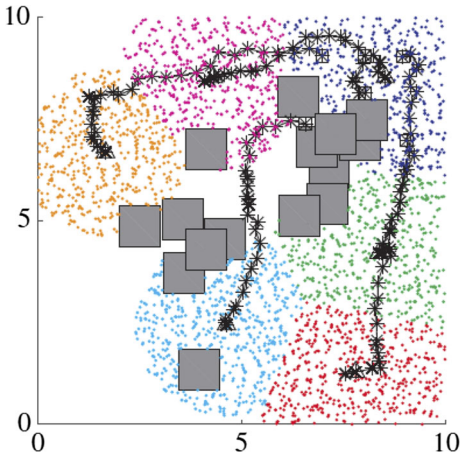

Fig. 7 The final 5000 node graph $\widetilde{\mathcal{V}}^{\mathrm{LR}}$ for six agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\mathrm{mixed}}$

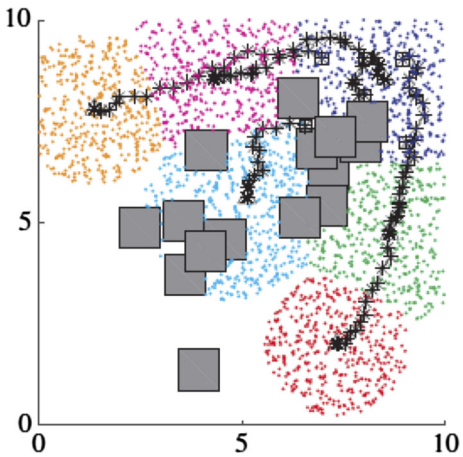

Fig. 8 The final 5000 node graph $\widetilde{\mathcal{V}}^{\mathrm{LR}}$ for six agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\mathrm{mixed}}$

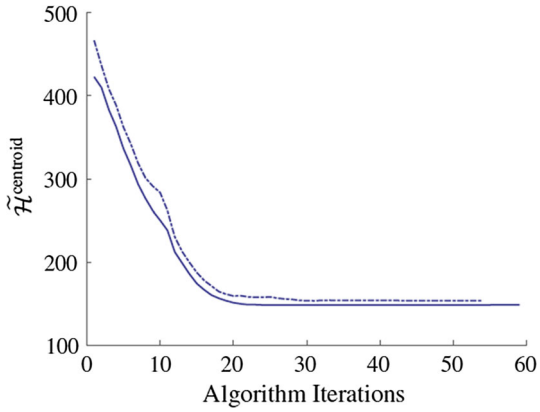

Fig. 9 The evolution of $\widetilde{\mathcal{H}}^{\mathrm{centroid}}$ obtained by solving Problem 2 using $\widetilde{\mathcal{V}}^{\mathrm{weighted}}$, where the solid line is the 5000 node graph and the dashed is the 2000 node graph

7.2 Evolution of $\widetilde{\mathcal{H}}$

The following simulations are a comparison of the evolution of the different cost functions for the various problems. The plots show the algorithm converges to a solution.

The evolution of $\widetilde{\mathcal{H}}^{\mathrm{centroid}}$ for agents solving Problem 2 in the non-convex environment using partition $\widetilde{\mathcal{V}}^{\mathrm{weighted}}$ is in Fig. 9. The cost function decreases monotonically as expected.

Figure 10 is the evolution of $\widetilde{\mathcal{H}}^{\mathrm{area}}$ for the agents solving Problem 2 with limited ranges in the non-convex environment. The evolution of $\widetilde{\mathcal{H}}^{\mathrm{mixed}}$ for the limited range agents in the convex environment solving Problem 2 is in Fig 11. Both the 2000 and 5000 sample graphs produce costs that decrease smoothly until the algorithm gets close to convergence then chatters slightly. It is important to note that when the algorithm uses $\widetilde{\mathcal{H}}^{\mathrm{area}}$ or $\widetilde{\mathcal{H}}^{\mathrm{mixed}}$, the position update is approximate. The agents do not follow the gradient exactly, but rather a close approximation based on the neighboring nodes. The more nodes there are in the graph, the less error there will be in following the gradient. This can be seen by comparing the 2000 and 5000 node graphs. The 5000 node graph produces a cost function plot with less pronounced increases and less chattering.

The following simulation is with seven agents in a hallway type environment. The initial and final agent partitions are shown in Figs. 12 and 13. The area-only cost function is plotted in Fig. 14 for both the c and $R_{\max}$ definitions of the limited range sub-partition. The cost function decreases and then chatters due to the error in the gradient. Figure 15 is the mixed cost function. The c definition decreases monotonically while the $R_{\max}$ definition increases first then decreases monotonically.

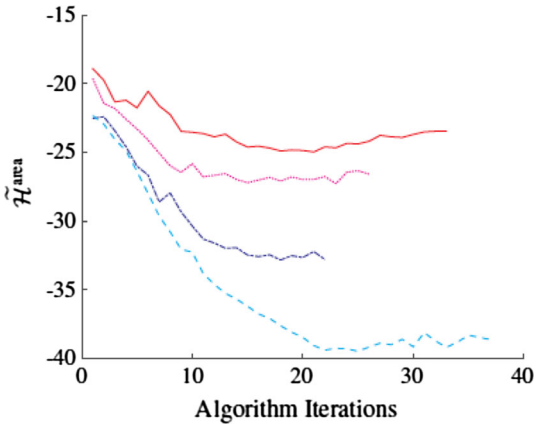

Fig. 10 The evolution of $\widetilde{\mathcal{H}}^{\text{area}}$ obtained by solving Problem 2 using $\widetilde{\mathcal{V}}^{\text{LR}}$ from the 2000 node graph with $c = 3$ (blue dash-dot line), $R_{\max} = 3.5$ (red solid line), from the 5000 node graph with $c = 3$ (cyan dashed line), $R_{\max} = 3.5$ (magenta dotted line) (Color figure online)

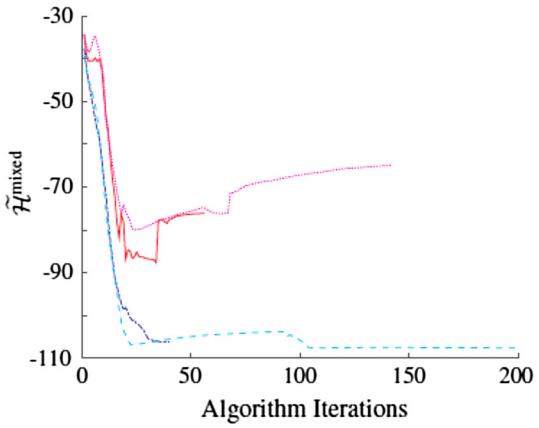

Fig. 11 The evolution of $\widetilde{\mathcal{H}}^{\text{mixed}}$ obtained by solving Problem 2 using $\widetilde{\mathcal{V}}^{\text{LR}}$ from the 2000 node graph with $c = 3$ (blue dash-dot line), $R_{\max} = 3.5$ (red solid line), from the 5000 node graph with $c = 3$ (cyan dashed line), $R_{\max} = 3.5$ (magenta dotted line) (Color figure online)

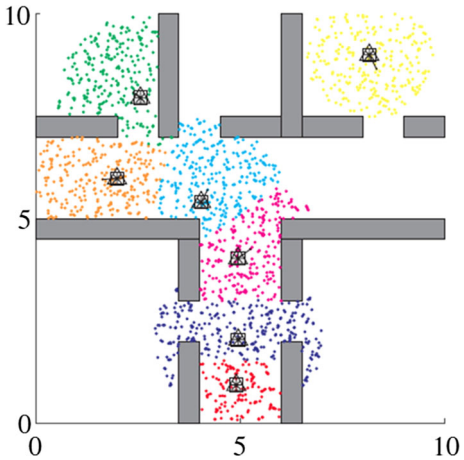

Fig. 12 A 5000 node graph initial $\widetilde{\mathcal{V}}^{\text{LR}}$ for seven agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{area}}$

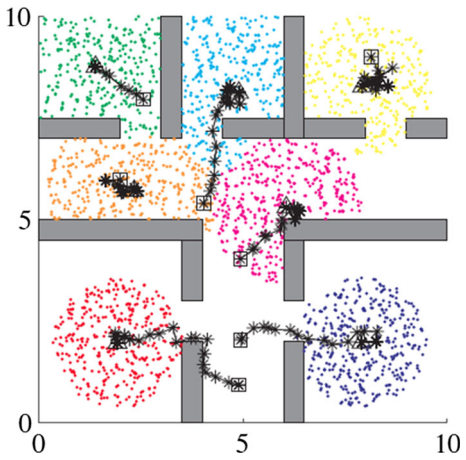

Fig. 13 A 5000 node graph final $\widetilde{\mathcal{V}}^{\text{LR}}$ for seven agents obtained by solving Problem 2 with $\widetilde{\mathcal{H}}^{\text{area}}$

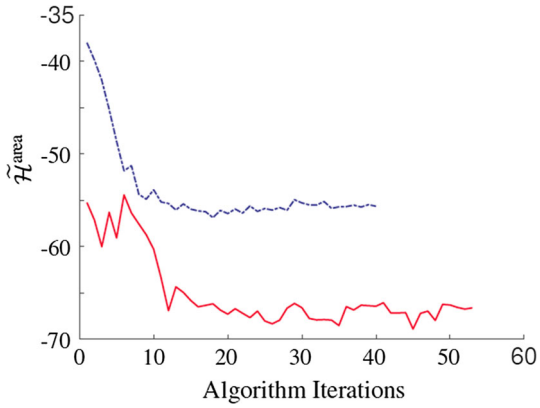

Fig. 14 The evolution of $\widetilde{\mathcal{H}}^{\text{area}}$ obtained by solving Problem 2 using $\widetilde{\mathcal{V}}^{\text{LR}}$ from a 5000 node graph with $c = 3$ (blue dash-dot line), $R_{\max} = 3.5$ (red solid line) (Color figure online)

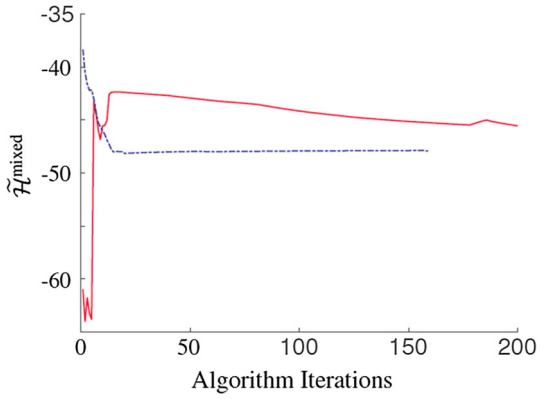

Fig. 15 The evolution of $\widetilde{\mathcal{H}}^{\text{mixed}}$ obtained by solving Problem 2 using $\widetilde{\mathcal{V}}^{\text{LR}}$ from a 5000 node graph with $c = 3$ (blue dash-dot line), $R_{\max} = 3.5$ (red solid line) (Color figure online)

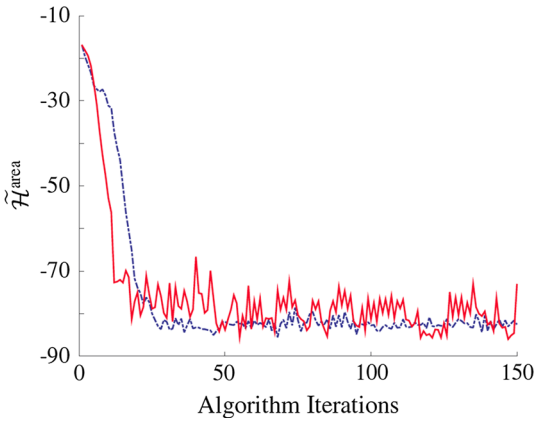

Fig. 16 The evolution of $\widetilde{\mathcal{H}}^{\text{area}}$ obtained by solving Problems 2 for Dubins' vehicle using $\widetilde{\mathcal{V}}^{\text{LR}}$, where the blue dashed line is with $c = 7$ and solid red line is with $R_{\max} = 8$ (Color figure online)

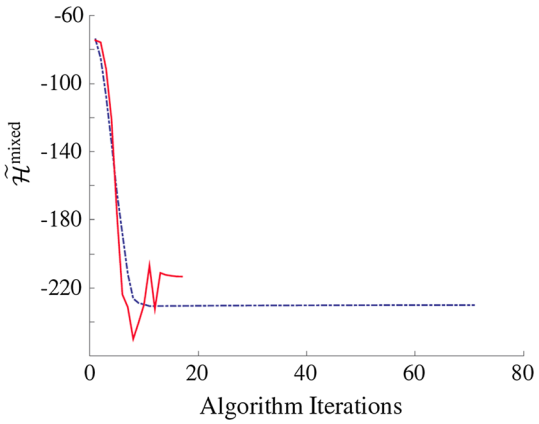

Fig. 17 The evolution of $\widetilde{\mathcal{H}}^{\text{mixed}}$ obtained by solving Problems 2 for Dubins' vehicle using $\widetilde{\mathcal{V}}^{\text{LR}}$, where the blue dashed line is with $c = 7$ and solid red line is with $R_{\max} = 8$ (Color figure online)

7.2.1 Dubins' vehicle results

Simulations for Dubins' vehicle agents, subject to limited ranges, solving Problem 2 using $\widetilde{\mathcal{V}}^{\text{LR}}$ were run for a 2000 node graph. Figure 16 is the evolution of $\widetilde{\mathcal{H}}^{\text{area}}$ and Fig. 17 is the evolution of $\widetilde{\mathcal{H}}^{\text{mixed}}$ in the non-convex environment. The blue dashed lines are for $\widetilde{\mathcal{V}}^{\text{LR}}$ with a constant $c = 7$ and the red solid lines are for $R_{\max} = 8$. The $\widetilde{\mathcal{H}}^{\text{area}}$ and $\widetilde{\mathcal{H}}^{\text{mixed}}$ decrease initially and then chatters until convergence is reached. The $\widetilde{\mathcal{H}}^{\text{mixed}}$ decreases smoothly for $\widetilde{\mathcal{V}}^{\text{LR}}$ with a constant $c = 7$. The chattering produced by the Dubins' vehicle is due to the resolution in the graph. If the graph were to have more nodes, the position update would have less error and therefore less chattering. Because an increase in the number of nodes in the graph causes the algorithm to increase in run time, a balance between the run time and the smoothness of the cost function decrease is needed.

8 Conclusions

A limited range spatial load balancing problem for agents subject to differential constraints in a non-convex environment is defined and discussed. To handle differential constraints, the problem is redefined using a probabilistic roadmap star (PRM*) and an algorithm is given that finds an approximate solution to the original problem. We discuss how introducing the limited range sub-partition and determining a subset of agents containing the Voronoi neighbors of a specific agent limits the amount of communication between agents. A convergence proof is given for the GRAPH- BASED SPATIAL LOAD BALANCING algorithm. All other defined approximated problems are shown to converge in simulation. Future work includes more extensive differential constraint simulations as well as implementation of the algorithm on a set of mobile robots.

Appendix

Proof of Lemma 1 Take the derivative of $F(\omega)$ with respect to ω_i using the Leibniz rule (Flanders 1973), which applies over general domains,

$$\begin{aligned}\frac{\partial F(\omega)}{\partial \omega_i} = \frac{-n}{1-n}\Bigg(&\sum_{j=1}^{n}\int_{V_j^{\text{LR}}} -\left[\frac{\partial q}{\partial \omega_i}\right.\\ &\left.\times\left(\frac{\partial}{\partial q}\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k - c\right)\phi(q)\right)\right]\cdot dq\\ &+\sum_{j=1}^{n}\int_{\partial V_j^{\text{LR}}}\frac{\partial q}{\partial \omega_i}\cdot\left(\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k - c\right)\phi(q)\right)dq\\ &+\sum_{j=1}^{n}\int_{V_j^{\text{LR}}}\frac{\partial}{\partial \omega_i}\left(\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k - c\right)\phi(q)\right)dq\Bigg).\end{aligned}$$

The first term,

$$\begin{aligned}&\sum_{j=1}^{n}\int_{V_j^{\text{LR}}} -\left[\frac{\partial q}{\partial \omega_i}\right.\\ &\quad\left.\times\left(\frac{\partial}{\partial q}\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k - c\right)\phi(q)\right)\right]\cdot dq = 0.\end{aligned}$$

There are two vectors in the (q_1, q_2) plane being crossed, resulting in a vector perpendicular to the (q_1, q_2) plane, in dot product with a vector in the (q_1, q_2) plane, thus resulting in a zero value. The second term becomes

$$\sum_{j=1}^{n}\int_{\partial V_j^{\mathrm{LR}}}\frac{\partial q}{\partial \omega_i}\cdot\left(\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k-c\right)\phi(q)\right)dq$$
$$=\sum_{j=1}^{n}\int_{\triangle_{ij}}\hat{n}^{\top}\frac{\partial q}{\partial \omega_i}\left(\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k-c\right)\phi(q)\right)dq$$
$$+\sum_{j=1}^{n}\int_{\Lambda_j}\hat{n}^{\top}\frac{\partial q}{\partial \omega_i}\left(\left(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k-c\right)\phi(q)\right)dq.$$

This term then vanishes along $\triangle_{ij}$ because the opposing normal vectors are multiplying the same terms which then cancel each other out. For all $q\in\Lambda_j$, $J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k-c=0$. The third term,

$$\sum_{j=1}^{n}\int_{V_j^{\mathrm{LR}}}\frac{\partial}{\partial \omega_i}\left((J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k-c)\phi(q)\right)dq$$
$$=\sum_{j=1}^{n}\int_{V_j^{\mathrm{LR}}}\Bigg(\cancelto{0\,\forall\, j}{\frac{\partial J(p_j,q)}{\partial\omega_i}}\phi(q)-\cancelto{1,\ i=j}{\frac{\partial\omega_j}{\partial\omega_i}}\phi(q)$$
$$+\frac{1}{n}\sum_{k\neq 1}^{n}\cancelto{1,\ i=k}{\frac{\partial\omega_k}{\partial\omega_i}}\phi(q)-\cancelto{0\,\forall\, j}{\frac{\partial c}{\partial\omega_i}}\phi(q)$$
$$+(J(p_j,q)-\omega_j+\frac{1}{n}\sum_{k=1}^{n}\omega_k-c)\cancelto{0\,\forall\, j}{\frac{\partial\phi(q)}{\partial\omega_i}}\Bigg)dq$$
$$=\frac{1-n}{n}\int_{V_i^{\mathrm{LR}}}\phi(q)dq.$$

Putting everything together results in

$$\frac{\partial F}{\partial\omega_i}=-\int_{V_i^{\mathrm{LR}}}\phi(q)dq=-\mathcal{M}_i(\omega),\quad i\in\{1,\dots,n\}.$$

□

Proof of Lemma 2 Fact 1 follows from $\mathcal{M}$, and thus $\mathcal{M}^0$, being gradient. For Fact 2, by definition, the sum of $\mathcal{M}^0$ stays constant when ω is updated, therefore, the range of $\mathcal{M}^0$ is in $\{m\in\mathbb{R}^n_{\geq 0}\mid \mathbf{1}_n^{\top}m=A\}$, and thus, $\mathbf{1}_n^{\top}J(\mathcal{M}^0)=\mathbf{0}_n$. Because $J(\mathcal{M}^0)$ is symmetric, $\mathbf{1}_n$ is a right eigenvector with eigenvalue 0. Consider $\omega\in U^0$ and let η be the number of connected components to $\mathcal{G}_{\mathrm{LR}}\equiv\mathcal{G}_{\mathrm{LR}}(P,\omega)$, such that each connected component, $\mathcal{G}_l$, has a corresponding vector $\mathbf{v}_l$ associated with it. Then, because each $\mathcal{G}_l$ maintains a constant area during the weights update, additional eigenvectors with eigenvalue zero can be defined, $\mathbf{v}_l^{\top}J(\mathcal{M}^0)(\omega)=\mathbf{0}_n$. For Fact 3, first show that for $i\in\{1,\dots,n\}$ and $(\omega_1,\dots,\omega_i,\dots,\omega_n)$, $(\omega_1,\dots,\omega_i',\dots,\omega_n)$ such that $\omega_i'\geq\omega_i$,

$$\mathcal{M}_i^0(\omega_1,\dots,\omega_i',\dots,\omega_n)\geq\mathcal{M}_i^0(\omega_1,\dots,\omega_i,\dots,\omega_n),$$
$$\mathcal{M}_j^0(\omega_1,\dots,\omega_i',\dots,\omega_n)\leq\mathcal{M}_j^0(\omega_1,\dots,\omega_i,\dots,\omega_n),\ j\neq i.$$

Recall, $V_i^{\mathrm{LR}}=V_i^{\mathrm{weighted}}\cap D_i$. First, how $\partial V_i^{\mathrm{weighted}}$ changes with respect to ω is examined. When $q\in\triangle_{ij}$ then $J(p_i,q)-\omega_i=J(p_j,q)-\omega_j$. If ω_i increases, then the boundary $\triangle_{ij}$ moves further away from p_i which increases the area of V_i^{weighted} and decreases the area of V_j^{weighted}. Second, examine the change in the boundary of D_i. Let $J_{\max}(p_i,q)$ be the cost at the boundary of D_i, then $J_{\max}(p_i,q)=c+\omega_i-\frac{1}{n}\sum_{k=1}^{n}\omega_k$. If ω_i increases then so does $J_{\max}(p_i,q)$ which in turn increases the area of D_i. For D_j, if ω_i increases then $J_{\max}(p_j,q)$ decreases and hence the area of D_j decreases. Then, the partial derivatives of $\mathcal{M}^0$ for $\mathcal{V}^{\mathrm{LR}}$ at ω satisfy

$$\frac{\partial\mathcal{M}_i^0}{\partial\omega_i}\geq 0,\quad \frac{\partial\mathcal{M}_j^0}{\partial\omega_i}\leq 0,\quad j\neq i.$$

The above expression, when combined with Facts 1 and 2, leads to $J(\mathcal{M}^0)(\omega)$ being the Laplacian of $\mathcal{G}_{\mathrm{LR}}$. Because $\mathcal{G}_{\mathrm{LR}}$ has exactly η connected components, the $\mathrm{rank}(J(\mathcal{M}^0))(\omega)=n-\eta$. □

Proof of Theorem 1 First, consider the function $G:U^0\to\mathbb{R}$ defined as

$$G(\omega)=\frac{1}{2}\|\mathcal{M}^0(\omega)-(\bar{a}_1,\dots,\bar{a}_n)\|^2. \tag{9}$$

This function is continuous and invariant under translations. Therefore, it induces a continuous function on $U^0/\sim$, the quotient set of U^0 for which $\omega_1\sim\omega_2$ if they differ in a translation. It can be proven that $U^0/\sim$ is a compact set and there is a minimizer of G on $U^0/\sim$. Evaluate the derivative of (9) with respect to ω at ω^*,

$$0=\frac{\partial}{\partial\omega_i}\Big|_{\omega=\omega^*}\left(\frac{1}{2}\|\mathcal{M}^0(\omega)-(\bar{a}_1,\dots,\bar{a}_n)\|^2\right)$$
$$=\sum_{k=1}^{n}\left(\mathcal{M}_k^0(\omega^*)-\bar{a}_k\right)\frac{\partial\mathcal{M}_k^0}{\partial\omega_i}\Big|_{\omega=\omega^*},\ \forall\, i\in\{1,\dots,n\},$$

which can then be expressed as $(\mathcal{M}^0(\omega^*)-(\bar{a}_1,\dots,\bar{a}_n))J(\mathcal{M}^0)(\omega^*)=\mathbf{0}_n$. The weights-to-area map, $\mathcal{M}^0(\omega)$, is differentiable in the same way that $F(\omega)$ from Lemma 1 is differentiable. Note that ω^* is in the interior of $U^0/\sim$ because $\bar{a}_i>0$ (boundary points are those for which the cell of an agent is empty). Recall that, over U^0, there are eigenvectors such that $\mathbf{v}_l^{\top}J(\mathcal{M}^0)(\omega^*)=\mathbf{0}_n$ for all $l\in\{1,\dots,\eta\}$ and that the $\mathrm{rank}(J(\mathcal{M}^0)(\omega^*))=n-\eta$. From here, deduce that $\mathcal{M}^0(\omega^*)-(\bar{a}_1,\dots,\bar{a}_n)=\sum_{l=1}^{n}b_l\mathbf{v}_l$ for some $b_l\in\mathbb{R}$. Next, notice that $0=\mathbf{v}_l^{\top}(\mathcal{M}^0(\omega^*)-(\bar{a}_1,\dots,\bar{a}_n))=b_ln_l$, and therefore $b_l=0$ for all $l\in\{1,\dots,\eta\}$, or $\mathcal{M}^0(\omega^*)=(\bar{a}_1,\dots,\bar{a}_n)$. □

Proof of Lemma 3 Because $q \in \partial \widetilde{D}_i$, $J(p_i, q) = c + \omega_i - \frac{1}{n}\sum_{k=1}^{n} \omega_k$. This same q is also in $\partial \widetilde{V}_i^{\text{weighted}}$,

$$J(p_i, q) - \omega_i + \omega_j = J(p_j, q), \text{ for some } j.$$

Substituting the above expression into the triangle inequality, $J(p_i, q) + J(p_j, q) \geq J(p_i, p_j)$, gives

$$2c - \frac{2}{n}\sum_{k=1}^{n} \omega_k + \omega_i + \omega_j \geq J(p_i, p_j).$$

Therefore, the c must be larger than

$$c \geq \frac{J(p_i, p_j) + \frac{2}{n}\sum_{k=1}^{n} \omega_k - \omega_i - \omega_j}{2}.$$

□

Proof of Lemma 4 Because $q \in \widetilde{V}_i^{\text{weighted}}$ then

$$\|p_i - q\|^2 - \omega_i = \|p_j - q\|^2 - \omega_j \text{ for some } j. \tag{10}$$

Now, squaring the triangle inequality of the Euclidean norm,

$$\|p_j - q\|^2 \geq \|p_i - p_j\|^2 + \|p_i - q\|^2 - 2\|p_i - p_j\|\|p_i - q\|.$$

Substitute this equation into (10), and rearranging, gives

$$\|p_i - q\| \geq \frac{\|p_i - p_j\|^2 + \omega_i - \omega_j}{2\|p_i - p_j\|}.$$

Using $q \in \partial \widetilde{D}_i$, leads to $\|p_i - q\| = (c + \omega_i - \frac{1}{n}\sum_{k=1}^{n} \omega_k)^{1/2}$, which is substituted into the above expression giving the bound on c,

$$c \geq \frac{(\|p_i - p_j\|^2 + \omega_i - \omega_j)^2}{4\|p_i - p_j\|^2} - \omega_i + \frac{1}{n}\sum_{k=1}^{n} \omega_k \text{ for all } j.$$

□

Proof of Proposition 1 The existence of an optimal path from p_i to q passing through q' leads to $J(p_i, q') + J(q', q) = J(p_i, q)$. From $q' \notin \widetilde{V}_i^{\text{weighted}}$, we have $J(p_i, q') - \omega_i \geq J(p_j, q') - \omega_j$ for some $j \neq i$. Putting these two equations together, along with the triangle inequality $J(p_j, q) \leq J(p_j, q') + J(q', q)$ gives

$$\begin{aligned} J(p_i, q') - \omega_i &\geq J(p_j, q') - \omega_j \\ J(p_i, q) - J(q', q) - \omega_i &\geq J(p_j, q) - J(q', q) - \omega_j \\ J(p_i, q) - \omega_i &\geq J(p_j, q) - \omega_j. \end{aligned}$$

which implies that $q \notin \widetilde{V}_i^{\text{weighted}}$. □

Proof of Theorem 2 Convergence can be proved by showing that $\widetilde{\mathcal{H}}^{\text{centroid}}$ decreases monotonically at every step. Recall that the agent positions are updated specifically so that $\widetilde{\mathcal{H}}^{\text{centroid}}$ decreases, $\widetilde{\mathcal{H}}^{\text{centroid}}(P, \widetilde{\mathcal{V}}^{\text{weighted}}(P)) \geq \widetilde{\mathcal{H}}^{\text{centroid}}(P^+, \widetilde{\mathcal{V}}^{\text{weighted}}(P))$. Next, using (8),

$$\begin{aligned} &\widetilde{\mathcal{H}}^{\text{centroid}}(P^+, \widetilde{\mathcal{V}}^{\text{weighted}}(P)) \\ &\quad \geq \mathcal{H}^{\text{centroid}}(P^+, \mathcal{V}^{\text{weighted}}(P)) - \epsilon \\ &\quad \geq \mathcal{H}^{\text{centroid}}(P^+, \mathcal{V}^{\text{weighted}}(P^+)) - \epsilon \\ &\quad \geq \widetilde{\mathcal{H}}^{\text{centroid}}(P^+, \widetilde{\mathcal{V}}^{\text{weighted}}(P^+)) - 2\epsilon. \end{aligned}$$

Thus, $\widetilde{\mathcal{H}}^{\text{centroid}}$ decreases as long as $\mathcal{H}^{\text{centroid}}(P^+, \mathcal{V}^{\text{weighted}})(P)$ is 2ϵ larger than $\mathcal{H}^{\text{centroid}}(P^+, \mathcal{V}^{\text{weighted}}(P^+))$. □

Proof of Theorem 3 Note that Q be compact and invariant under the dynamics T, and define $\mathcal{H}_{\text{v}}(P) = \mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}}(P, \mathcal{A}(P))$. Because the limited range spatial load balancing algorithm alternately updates the agents' positions and weight assignments, the evolution of $\mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}})$ is,

$$\begin{aligned} \mathcal{H}_{\text{v}}(P) &= \mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}}(P, \mathcal{A}(P))) \\ &= \mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(P, \mathcal{A}(P))) \geq \mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(P))) \\ &= \mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(T(P)))) = \mathcal{H}_{\text{v}}(T(P)), \end{aligned}$$

where $\mathcal{V}^{\text{LR}}(P, \omega)$ and $\mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(T(P)))$ both satisfy the variable a' constraint. When the partition $\mathcal{V}^{\text{LR}}$ is kept constant and the position is updated ($\mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}}(P, \mathcal{A}(P)))$ $= \mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(P, \mathcal{A}(P)))$), the area covered by $\mathcal{V}^{\text{LR}}$ is constant. Because $T(P)$ is found to specifically increase the area, the new sub-partition, $\mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(P))$, has a greater area than $\mathcal{V}^{\text{LR}}(P, \mathcal{A}(P))$ ($\mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(P, \mathcal{A}(P))) \geq \mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(P)))$). Finally, the $\mathcal{A}(P)$ values are updated in such a way that the area is kept constant during the update ($\mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(P))) = \mathcal{H}^{\text{area}}(T(P), \mathcal{V}^{\text{LR}}(T(P), \mathcal{A}(T(P))))$). Theorem 1 says that $\mathcal{A}(T(P))$ exists. Note that if $T(P) \neq P$ then $\mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}})$ decreases implying $\mathcal{H}_{\text{v}}(P) > \mathcal{H}_{\text{v}}(T(P))$. Therefore, $\mathcal{H}_{\text{v}}(P) = \mathcal{H}_{\text{v}}(T(P))$ if and only if $T(P) = P$. From here, apply the LaSalle invariance principle to guarantee the trajectories of T converge to the largest invariant set in $\mathcal{Z} = \{P \in Q \,|\, \mathcal{H}_{\text{v}}(P) = \mathcal{H}_{\text{v}}(T(P))\}$. It can be concluded from the above discussion that $\mathcal{Z}$ is the set of limited range generalized Voronoi configurations where the gradient of $\mathcal{H}^{\text{area}}(P, \mathcal{V}^{\text{LR}})$ is zero. □

References

Bertsekas, D. P., & Tsitsiklis, J. N. (1997). *Parallel and distributed computation: Numerical methods*. Athena Scientific. ISBN 1886529019.

Bhattacharya, S., Ghrist, R., & Kumar, V. (2014). Multi-robot coverage and exploration on riemannian manifolds wth boundaries. *The International Journal of Robotics Research*, *33*, 113–137.

Bhattacharya, S., Michael, N., & Kumar, V., (2013). Distributed coverage and exploration in unknown non-convex environments. In *International Symposium on Distributed Autonomous Robotic Systems*, pages 61–75. Springer.

Boardman, B., Harden, T., & Martínez, S. (2016). Spatial load balancing in non-convex environments using sampling-based motion planners. In *American Control Conference*.

Boardman, B., Harden, T., & Martínez, S. (2017). Limited range spatial load balancing. In *American Control Conference*, Submitted.

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

Caicedo-Nunez, C. H., & Zefran, M. (2008). Performing coverage on nonconvex domains. In *IEEE Conference Control Applications*, pp. 1019–1024.

Cortés, J. (2010). Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control*, *55*(3), 749–754.

Cortés, J., Martínez, S., & Bullo, F. (2005). Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation & Calculus of Variations*, *11*(4), 691–719.

Dijkstra, E. W. (1959). A note on two problems on connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Enright, J., Salva, K., & Frazzoli, E. (2008). Coverage control for nonholonomic agents. In *IEEE International Conference on Decision and Control*, pp. 4250–4256.

Flanders, H. (1973). Differentiation under the integral sign. *The American Mathematical Monthly*, *80*(6), 615–627.

Jiang, W., & Zefran, M. (2013). Coverage control with information aggregation. In *IEEE International Conference on Decision and Control* (pp. 5421–5426). IEEE.

Kantaros, Y., Thanou, M., & Tzes, A. (2014). Visibility-oriented coverage control of mobile robotic networks on non-convex regions. In *IEEE International Conference on Robotics and Automation* (pp. 1126–1131). IEEE.

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

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, *30*(7), 846–894.

Kwok, A., & Martínez, S. (2010a). Deployment algorithms for a power-constrained mobile sensor network. *International Journal on Robust and Nonlinear Control*, *20*(7), 725–842.

Kwok, A., & Martínez, S. (2010b). Unicycle coverage control via hybrid modeling. *IEEE Transactions on Automatic Control*, *55*(2), 528–532.

Laventall, K., & Cortés, J. (2008). Coverage control by robotic networks with limited-range anisotropic sensory. In *American Control Conference*, pp. 2666–2671, Seattle, WA.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137.

Mahboubi, H., Moezzi, K., Aghdam, A. G., Sayrafian-Pour, K., & Marbukh, V. (2014). Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors. *IEEE Transactions on Industrial Informatics*, *10*(1), 163–174.

Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2000). *Spatial tessellations: Concepts and applications of voronoi diagrams* (2 ed.). Wiley Series in Probability and Statistics. Wiley. ISBN 0471986356.

Patel, R., Frasca, P., & Bullo, F. (2014). Centroidal area-constrained partitioning for robot networks. *ASME Journal on Dynamic Systems, Measurement, and Control*, *136*(3), 031024-1–031024-8.

Pavone, M., Arsie, A., Frazzoli, E., & Bullo, F. (2011). Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Transactions on Automatic Control*, *56*(8), 1834–1848.

Pimenta, L., Kumar, V., Mesquita, R. C., & Pereira, G. (2008). Sensing and coverage for a network of heterogeneous robots. In *IEEE International Conference on Decision and Control*, pp. 3947–3952, Cancun, Mexico.

Renzaglia, A., & Martinelli, A. (2009). Distributed coverage control for a multi-robot team in a non-convex environment. In *IEEE/RSJ International Conference on Intelligent Robots & Systems*.

Savla, K., Bullo, F., & Frazzoli, E. (December 2007). The coverage problem for loitering Dubins vehicles. In *IEEE International Conference on Decision and Control*, pp. 1398–1403, New Orleans, LA.

Stergiopoulos, Y., & Tzes, A. (2011). Coverage-oriented coordination of mobile heterogeneous networks. In *Mediterranean Conference on Control and Automation*, pp. 175–180.

Zhong, M., & Cassandras, C. G. (2011). Distributed coverage control and data collection with mobile sensor networks. *IEEE Transactions on Automatic Control*, *56*(10), 2445–2455.

Beth Boardman is a Research and Development Engineer with Los Alamos National Laboratory. In 2013, she joined the Process Hardware Automation and Robotics team within the Process Automation and Control (AET-5) group at Los Alamos National Laboratory. At Los Alamos National Laboratory, she participates in research for manipulator collision avoidance and path planning. She is also involved in augmented reality research projects. She received her Ph.D. in Aerospace Engineering from the University of California, San Diego in October 2017, where her research focused on robotic motion planning and multi-agent networks. She received her Masters and Bachelors in Aeronautics and Astronautics from the University of Washington in 2012 and 2010.

Troy Harden Research and Development Engineer, Los Alamos National Laboratory. Troy Harden has worked for the past 14+ years in the Process Hardware Automation and Robotics team within the Process Automation and Control (AET-5) group (and its predecessor organizations) at Los Alamos National Laboratory. At Los Alamos, he has participated in research and development in the areas of manipulator obstacle avoidance, human–robot interaction, and robot motion planning. He has also provided technical leadership on a variety of automation projects. Example automation projects include development of a robotic system to clean legacy spherical containment vessels, integrating a probing system into a lathe controller, designing/building end effectors and tooling for the Advanced Recovery and Integrated Extraction System (ARIES) disassembly lathe and robot, development

of on-machine gauging software for machine tools, development of force guided placement software for the ARIES disassembly lathe robot, development of control systems for a variety of machine tools, and the design and integration of custom automation hardware. Troy currently serves as the first line manager for his team. He came to Los Alamos as a post-doc in 2002 after receiving his Ph.D. in mechanical engineering from The University of Texas at Austin (UT). While at UT, he worked in the Robotics Research Group (RRG), where his major research topic was manipulator obstacle avoidance. Other RRG responsibilities included software development, hardware and software integration, and troubleshooting and repairing a variety of robotic manipulators (from 6 Degree of Freedom (DOF) industrial manipulators to a 17 DOF dual-arm research robot). Troy has sixteen published conference papers.

Sonia Martínez is a Professor at the Department of Mechanical and Aerospace Engineering at the University of California, San Diego. Prof. Martínez received her Ph.D. degree in Engineering Mathematics from the Universidad Carlos III de Madrid, Spain, in May 2002. Following a year as a Visiting Assistant Professor of Applied Mathematics at the Technical University of Catalonia, Spain, she obtained a Postdoctoral Fulbright Fellowship and held appointments at the Coordinated Science Laboratory of the University of Illinois, Urbana-Champaign during 2004, and at the Center for Control, Dynamical systems and Computation (CCDC) of the University of California, Santa Barbara during 2005. From January 2006 to June 2010, she was an Assistant Professor with the department of Mechanical and Aerospace Engineering at the University of California, San Diego. From July 2010 to June 2014, she was an Associate Professor with the department of Mechanical and Aerospace Engineering at the University of California, San Diego. Dr Martínez' research interests include networked control systems, multi-agent systems, and nonlinear control theory with applications to robotics and cyber-physical systems. In particular, she has focused on the modeling and control of robotic sensor networks, the development of distributed coordination algorithms for groups of autonomous vehicles, and the geometric control of mechanical systems. She is a Senior Editor of the IEEE Transactions on Control of Networked Systems.