Limited Range Spatial Load Balancing for Multiple Robots

Beth Boardman\textsuperscript{1,2}, Troy Harden\textsuperscript{2}, and Sonia Mart\'\i nez\textsuperscript{1}

Abstract—This paper generalizes a spatial load balancing deployment problem for multiple robots in convex environments to handle limited range agent constraints. Here, limited range is understood as imposing a reachability condition or limitation on the distance that an agent can move. The limited range helps in forming a distributed algorithm that restricts the neighbors of an agent. To account for limited ranges, two different cost functions are considered for minimization subject to a variable area constraint. This area constraint leads to a sub-partition of the environment which is dependent on a set of weights and is invariant under translation of the weights. For a fixed sub-partition, an agent position update law is given to decrease the corresponding cost function. Building on the update laws, a class of distributed algorithms are proposed to solve the limited range spatial load balancing problem. We present a result that shows convergence of the algorithm for the area-only cost function. Finally, we include simulations that show the convergent behavior of the algorithms for both cost functions.

I. INTRODUCTION

This paper presents a scalable solution to the limited range, area-constrained, multi-agent spatial load balancing problem in convex environments and for agents subject to differential constraints. The problem of balanced deployment involves a group of agents fairly dividing the load of acquiring information or of servicing tasks spread over an environment. This requirement translates into an optimal region assignment or of servicing tasks spread over an environment. This partition, combined with an appropriately modified objective, is sufficient to enforce agents’ regions to have a desired area. These papers assume the agents are deployed in convex environments with unlimited ranges. While the algorithms have certain distributed properties, in general, they are not implementable over a limited-range communication graph.

This paper defines and analyzes a spatial load balancing problem for robotic agents with limited ranges in a convex environment. The agents minimize one of two cost functions, $\mathcal{H}^{\text{mixed}}(P, W)$ or $\mathcal{H}^{\text{area}}(P, W)$, subject to an area constraint. A limited range sub-partition of the space is defined as $\mathcal{V}^{\text{LR}}(P, \omega)$, dependent on a set of weights, $\omega$. An update law is given that allows agents to find the weights that satisfy the variable area constraint. The agent positions are updated using a gradient law aimed at decreasing the cost function with respect to agent position. We then introduce a class of deployment algorithms for solving the limited range spatial load balancing problem and a result on the algorithm convergence to a solution for the choice of $\mathcal{H}^{\text{area}}(P, W)$ is stated. The limited range can be defined as a maximum allowable travel distance when the agents are in a Euclidean space. The algorithm is shown to be distributed over a connected communication graph in this case. Simulations for agents whose cost to move is given by the Euclidean norm squared, in a convex environment, are shown to converge for $\mathcal{H}^{\text{mixed}}(P, W)$ and $\mathcal{H}^{\text{area}}(P, W)$.

The organization of this paper is as follows. First, the general spatial load balancing problem is defined in Section II. Then, the limited range extension is analyzed in Section III. Next, Section IV examines the distributed algorithm for solving the limited range spatial load balancing problem. Section V follows with simulations of the algorithm in a convex environment. Finally, Section VI contains the conclusion of the paper and future work directions.

II. PROBLEM STATEMENT

To begin some notation is introduced. In general, let $Q \subseteq \mathbb{R}^d$ be the convex configuration space where robots are deployed. 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 of $Q$, $\bigcup_{i=1}^n W_i = \mathcal{Q} \subseteq Q$. Denote the boundary of cell $W_i$ as $\partial W_i$. Define the shared boundary between cells $i$ and $j$ as $\delta_{ij} = W_i \cap W_j$ and denote the unshared boundary of cell $i$ as $\Lambda_i = \partial W_i \setminus \left( \bigcup_{j \neq i} W_j \right)$. Finally, we let $1_n = (1, \ldots, 1)^\top \in \mathbb{R}^n$ and $0_n = (0, \ldots, 0)^\top \in \mathbb{R}^n$.

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 balanced region assignments 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 $Q$, $\phi : Q \to \mathbb{R}_{>0}$, describes the likelihood of an event occurring at a configuration in $Q$.

Let $a_1, \ldots, a_n \in \mathbb{R}_{>0}$, be desired cell areas such that
\[
\sum_{i=1}^{n} a_i = \int_{Q} \phi(q) dq .
\]
When $\bigcup_{i=1}^{n} W_i = Q' \subseteq Q$, a new variable area constraint is defined,
\[
a'_i = \frac{a_i \int_{Q} \phi(q) dq}{\int_{Q} \phi(q) dq}, \quad i \in \{1, \ldots, n\}.
\]

Of particular interest is the equal area case, $a_i = a_j$, or $a_i' = a_j'$, for all $i, j$. The $n$ agents solve the following minimization problem subject to the area and dynamic constraints.

**Problem 1** (Multicenter Optimization Problem with Dynamic and Area Constraints),
\[
\min \quad H(P, W)
\]
\[
s.t. \quad p_i \in Q, \quad p_i = f(p_i, u_i),
\]
\[
a'_i = \int_{W_i} \phi(q) dq, \quad i \in \{1, \ldots, n\}.
\]

The authors of [2] assume $\bigcup_{i=1}^{n} W_i = Q$, $a_i = a'_i$, and the cost function being minimized takes the form,
\[
H_{\text{centroid}}(P, W) = \sum_{i=1}^{n} \int_{W_i} J(p_i, q) \phi(q) dq.
\]

When the agents have a limited travel range, referred to as limited range, a sub-partition, $\bigcup_{i=1}^{n} W_i \subseteq Q$, is found and $a_i \geq a'_i$. Since the area of the sub-partition changes as a function of 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,
\[
H_{\text{area}}(P, W) = -\sum_{i=1}^{n} \int_{W_i} \phi(q) dq,
\]
or combines $H_{\text{centroid}}(P, W)$ and $H_{\text{area}}(P, W)$ to obtain,
\[
H_{\text{mixed}}(P, W) = \sum_{i=1}^{n} \left( \int_{W_i} J(p_i, q) \phi(q) dq - k_i \int_{W_i} \phi(q) dq \right),
\]
where $k_i \in \mathbb{R}_{>0}$, are constants, see Section III-C.

### A. Unlimited Range Agents in Convex Spaces

The solution to Problem 1 with limited ranges extends the previous work in [2], [9], [10] where $Q$ is convex and the agents have unlimited range, $\bigcup_{i=1}^{n} W_i = Q$. For trivial first order dynamics, the results in [2] state that, given a set of positions, $P$, there exists a weight assignment, $\omega$, that makes a generalized weighted Voronoi partition, $V_{\text{weighted}}(P, \omega; J) = \{V_i^{\text{weighted}}(\omega)\}_{i=1}^{n}$ such that, for all $i \in \{1, \ldots, n\}$,
\[
V_i^{\text{weighted}}(\omega) = \{ q \in Q | J(p_i, q) - \omega_i \leq J(p_j, q) - \omega_j, \forall i \neq j \}.
\]

The feasible set of weights is $U = \{ \omega \in \mathbb{R}^n | |\omega_i - \omega_j| \leq J(p_i, p_j), \quad i, j \in \{1, \ldots, n\} \}$. At least one cell is empty when $\omega \not\in U$. Given a partition in a convex environment, the best agent positions are the centroids of their cells. For certain metrics, such as Euclidean metrics, the centroids are given by a closed-form formula. However, for general metrics multiple agent positions may minimize the cost function, referred to here as generalized centroids.

### III. LIMITED RANGES

This section contains analytical results for the spatial load balancing problem with limited ranges. First, a limited range sub-partition is defined. Then, results pertaining to the existence of a set of weights that make the limited range sub-partition satisfy the area constraint are discussed, followed by how to update the agent positions. For the proofs of the results, we refer the reader to a forthcoming publication.

#### A. Limited Range Partition and Associated Graph

Define the limited ranges of the agents as the reachable sets, $D = \{D_1, \ldots, D_n\}$, such that for all $i \in \{1, \ldots, n\}$,
\[
D_i = \{ q \in Q | J(p_i, q) - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k \leq c \},
\]
where $c \in \mathbb{R}$, the maximum range, is a constant.

There are certain properties of $V_{\text{weighted}}$ from [2], that the limited range sub-partition should maintain in order to obtain analogous results. One such property is that the cells, $V_i^{\text{weighted}}$, are invariant under translation of $\omega$, $V_i^{\text{weighted}}(P, \omega; J) = V_i^{\text{weighted}}(P, \omega + t \cdot 1_n; J)$. This property leads to the area of $V_i^{\text{weighted}}$ also being invariant under translations of $\omega$. The mean of $\omega$ term makes the set $D_i$ invariant when $\omega$ is translated by $t$, substitute in $t = (\omega_i + t)$,
\[
J(p_i, q) - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k \leq c,
\]

The limited range sub-partition is defined as $V_{\text{LR}}(P, \omega, c) = \{V_i^{\text{LR}}(\omega)\}_{i=1}^{n}$, such that, for all $i \in \{1, \ldots, n\}$,
\[
V_i^{\text{LR}}(\omega) = V_i^{\text{weighted}}(\omega) \cap D_i.
\]

There is a graph, $G_{\text{LR}}(P, \omega) = (N, E)$, associated with $V_{\text{LR}}$, whose vertices, $v_i \in N$, correspond to the $n$ agents. In this graph, $e = (v_i, v_j) \in E$, 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 $G_{\text{LR}}(P, \omega)$, $(v_i, v_j) \in E$, then agents $i$ and $j$ are neighbors. Let $N_i$ denote the set of neighbors for agent $i$. When $Q' \neq Q$, $G_{\text{LR}}(P, \omega)$ may not be connected. The edges of $G_{\text{LR}}(P, \omega)$ change as the agent positions and weights are updated. The $G_{\text{LR}}(P, \omega)$ is not necessarily representative of the agents’ communication graph because determining $D_i$ depends on knowledge from all agents. The remainder of the paper will abbreviate $G_{\text{LR}}(P, \omega)$ as $G_{\text{LR}}$.

Lemma 1 gives a lower bound on the maximum range $c$ for a general $J(p_i, q)$ so that $\partial V_i^{\text{weighted}} \cap \partial D_i \neq \emptyset$ for each $i$. If the initial agent conditions lead to $\partial V_i^{\text{weighted}} \cap \partial D_i = \emptyset$,
assuming that \( V_{LR} \) satisfies the area constraint, for all \( i \), then Problem 1 is trivially satisfied.

**Lemma 1.** Assume the triangle inequality holds for \( J(p_i, q) \). If \( V_i^{\text{weighted}} \cap \partial D_i \neq \emptyset \) then
\[
c \geq \frac{1}{2} \sum_{k=1}^{n} \omega_k \quad \text{for all } j \in N_i.
\]

If the maximum range, \( c \), is too large such that \( V_i^{\text{LR}} = V_i^{\text{weighted}} \), then the limited range property is lost. A tighter lower bound on \( c \) would help maintain the limited range property and can be determined for specific \( J(p_i, q) \). In particular, Lemmas 2 and 3 compute possible lower bounds when the agents have no dynamics.

**Lemma 2.** Let \( J(p_i, q) = ||p_i - q||^2 \). If \( V_i^{\text{weighted}} \cap \partial D_i \neq \emptyset \) then
\[
c \geq \frac{1}{2} \sum_{k=1}^{n} \omega_k \quad \text{for all } j \in N_i.
\]

**Lemma 3.** Let \( J(p_i, q) = ||p_i - q|| \). If \( V_i^{\text{weighted}} \cap \partial D_i \neq \emptyset \) then
\[
c \geq \frac{1}{2} \sum_{k=1}^{n} \omega_k \quad \text{for all } j \in N_i.
\]

**B. Existence and Choice of Weights**

This section deals with proving the existence of a set of weights that allow \( V_i^{LR} \) to satisfy the variable area constraint. Recall, \( V_i^{LR} \) is invariant under translations in the weights and define the weights-to-area map as
\[
M(P, \omega) = \left( \int_{V_i^{LR}(\omega)} \phi(q) dq, \ldots, \int_{V_i^{LR}(\omega)} \phi(q) dq \right).
\]

Lemma 4 is a property of the weights-to-area map which are needed to prove Theorem 1. For conciseness, the following assumes \( V_i^{LR} = V_i^{LR}(\omega) \).

**Lemma 4.** The weights-to-area map, \( M \), is gradient, \( \nabla F = -M \), where \( F: \mathbb{R}^n \rightarrow \mathbb{R} \),
\[
F(\omega) = -\frac{1}{n} \sum_{j=1}^{n} \int_{V_i^{LR}} (J(p_j, q) - \omega_j + \frac{1}{n} \sum_{k=1}^{n} \omega_k - c) \phi(q) dq.
\]

To prove Lemma 4, take the derivative of \( F(\omega) \) with respect to \( \omega_j \), using the Leibniz rule [4].

The following shows how to update an initial weight assignment, \( \omega^0 \), to new weights that satisfy the area constraints. Define \( A = \sum_{i=1}^{n} M_i(\omega^0) \) and \( \bar{a}_i = \frac{A_i}{\int_Q \phi(q) dq} \).

In the equal area case, \( a_i = \bar{a}_i = \frac{1}{n} \int_Q \phi(q) dq \) and \( \bar{a}_i = \frac{A_i}{\int_Q \phi(q) dq} \)

Define, \( U^0 = \{ \omega \in U \mid \sum_{i=1}^{n} M_i(\omega) = \sum_{i=1}^{n} M_i(\omega^0) \} \).

Restrict \( M \) to \( \omega \in U^0 \), and denote this restriction as \( M^0 \).

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 [2] except \( G_{LR} \) 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 \( \omega^0 \) be some initial weights such that \( M_i(\omega^0) > 0 \) for each \( i \in \{1, \ldots, n\} \). Let \( \{a_1, \ldots, a_n\} \) be as defined in (1). Then there exists a set of weights \( \omega = \{\omega_1, \ldots, \omega_n\} \) such that
\[
\int_{V_i^{LR}(P, c)} \phi(q) dq = \bar{a}_i, \quad i \in \{1, \ldots, n\}.
\]

From Theorem 1, we know that weights exist for which \( V_i^{LR}(\omega) \) satisfies the area constraints while maintaining constant area in each connected component of \( G_{LR} \). Instead of maintaining constant areas by numerically solving for the \( n \)th weight, the following procedure is used in our algorithm. All the weights are iteratively updated so they eventually converge to \( \omega^* \), such that, \( V_i^{LR}(P, \omega^*) = \bar{a}_i \), thus preserving,
\[
\sum_{i=1}^{n} M_i(P, \omega^0) = \sum_{i=1}^{n} M_i(P, \omega^*). \]

Let the error between the current and desired cell areas be \( g(\omega) = M(\omega_1, \ldots, \omega_n) - (\bar{a}_1, \ldots, \bar{a}_n) \) and define, \( F(\omega) = -F(\omega) - \sum_{i=1}^{n} \omega_i \bar{a}_i \), then,
\[
\nabla F(\omega) = g(\omega) = 0. \]

The Jacobi algorithm, [1],
\[
\omega^+ = \omega - \gamma \text{diag}(\frac{\partial g_1}{\partial \omega_1}, \ldots, \frac{\partial g_n}{\partial \omega_n})^{-1} g(\omega),
\]

is then used to find the \( \omega \) values that optimize \( F \). The step size can be characterized as in [2] Prop. IV.5 to guarantee convergence in the weights. More precisely, let \( L \) be a level set of \( F(\omega) \), and then, \( \gamma \in (0, Y/B) \), where
\[
Y = \min_{i \in \{1, \ldots, n\}} \min_{\omega \in L} \frac{\partial g_i(\omega)}{\partial \omega_i} > 0, \quad B = \max_{i \in \{1, \ldots, n\}} \max_{\omega \in \mathbb{R}^n} \frac{\partial g_i(\omega)}{\partial \omega_i} > 0.
\]

To implement this algorithm, the agents update their positions with respect to the area error \( g_i(\omega) = M_i(\omega) - \bar{a}_i \) and \( \frac{\partial g_i}{\partial \omega_i} \), where,
\[
\frac{\partial g_i(\omega)}{\partial \omega_i} = \frac{\partial M_i(\omega)}{\partial \omega_i} - \frac{\partial \bar{a}_i}{\partial \omega_i} = \int_{\Lambda_i} \tilde{n}^T \frac{\partial \phi}{\partial \omega_i} dq + \int_{\Delta_{ij}} \tilde{n}^T \frac{\partial \phi}{\partial \omega_i} dq.
\]

**C. Gradient Function Computation**

The agents update their positions according to the derivative of \( H(P, V_i^{LR}(P, c)) \) with respect to position to solve Problem 1. For a general \( H \), the dynamics for agent \( i \) are
\[
p_i^+ = p_i - h \frac{\partial H(P, V_i^{LR}(P, c))}{\partial p_i},
\]

where \( h \) is an appropriate step size, found via a line search. The agents’ dynamic constraints are used to determine how to move to the new position.
Algorithm 1 \((P^*, \mathcal{V}^{LR}(P^*, \omega^*, c)) \leftarrow \text{LRSLB}(P^0, \omega^0, c, Q)\)

1: \((P, \omega) \leftarrow \text{Initialize}(P^0, \omega^0)\);  
2: for all \{Agent \(i\}\}_{i=1}^{n} do  
3: \(\text{while } P \neq P^+ \text{ do}\)  
4: \(P = P^+\);  
5: \(V_i^{LR}(P, \omega; J) \leftarrow \text{VoronoiPartition}(P, \omega, c)\);  
6: \(A \leftarrow \text{getArea}(V_i^{LR}(P, \omega))\);  
7: \(\text{while } \|\omega - \omega^+\| > \text{error} \text{ do}\)  
8: \(\omega = \omega^+\);  
9: \([\omega^+] \leftarrow \text{UpdateWeights}(P, \omega, V_i^{LR}, A)\);  
10: \([\omega^+] \leftarrow \text{TransmitAndReceive}(\omega^+)\);  
11: \(V_i^{LR}(P, \omega; J) \leftarrow \text{VoronoiPartition}(P, \omega, c)\);  
12: end while  
13: \(p_i^+ \leftarrow \text{UpdatePosition}(p_i)\);  
14: \(P^+ \leftarrow \text{TransmitAndReceive}(p_i^+)\);  
15: end while  
16: end for  
17: return \((P, \mathcal{V}^{LR}(P, \omega; J))\); 

For the area-only cost function the derivative is
\[
\frac{\partial H^{\text{area}}(P, \mathcal{V}^{LR}(P, \omega, c))}{\partial p_i} = -\int_{\Lambda_i} \phi(q) \hat{n}_i \cdot \frac{\partial \hat{q}}{\partial p_i} \, dq.
\]

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

For \(H^{\text{mixed}}(P, \mathcal{V}^{LR})\), the right selection of \(k_i\) can reduce the gradient computation to a generalized centroid of \(V_i^{LR}\):
\[
\frac{\partial H^{\text{mixed}}(P, \mathcal{V}^{LR}(P, \omega, c))}{\partial p_i} = \int_{V_i^{LR}} \phi(p) \hat{n}_i \cdot \frac{\partial \hat{q}}{\partial p_i} \, dq + k_i \int_{\Lambda_i} \phi(q) \hat{n}_i \cdot \frac{\partial \hat{q}}{\partial p_i} \, dq.
\] (3)

For agents subject to the Euclidean metric, the costs \(J(p_i, q) = \|p_i - q\|^2\) or \(J(p_i, q) = \|p_i - q\|\), choosing \(k_i = R_i \triangleq c + \omega_i - \frac{1}{n} \sum_{k=1}^{n} \omega_k\) reduces the gradient to a generalized centroid.

IV. ALGORITHM

This section outlines the limited range spatial load balancing (LRSLB) Algorithm 1 used to find the solution to Problem 1 with limited ranges. A convergence result for Algorithm 1 using \(H^{\text{area}}(P, \mathcal{V}^{LR})\) is also presented.

Algorithm 1 takes inputs \(Q\) and \(c\), where \(c\) is chosen so that one of the Lemmas 1–3 hold. Then, it is initialized with \(P^0\) and \(\omega^0\). In VoronoiPartition, each agent determines its cell, \(V_i^{LR}\): A distributed consensus algorithm over a connected communication graph is run via the primitive getArea to approximate \(A\), the area of \(V_i^{LR}\). The consensus algorithm is run sufficiently long to achieve convergence within some error. The weight convergence is done in Lines 7 – 12. The UpdateWeights primitive uses (2) to incrementally update the agent’s weight. The new weight is then transmitted to the neighboring agents in \(G_{LR}\), TransmitAndReceive, so the new \(V_i^{LR}\) can be determined. Once all the weights converge to within a specified error, the algorithm moves onto Line 13, UpdatePosition, which uses one of the two gradients from Section III-C. The agents then share their new positions with their neighbors. Algorithm 1 terminates, returning \((P, \mathcal{V}^{LR}(P, \omega))\), once all the agents’ positions become static.

A. Algorithm Convergence

This section discusses the convergence of Algorithm 1 with respect to the area-only cost function,
\[
H^{\text{area}}(P, \mathcal{V}^{LR}(P, \omega)) = -\sum_{i=1}^{n} \int_{V_i^{LR}} \phi(q) \, dq.
\]
First, define the mapping from agent positions to the weight assignment as, \(A : Q \rightarrow \mathbb{R}^n\) such that, \(A_i(P) = (a_1, \ldots, a_n)\). Then, define the trajectory of the agents as the mapping \(T : Q \rightarrow Q\), such that
\[
T(P) = (\text{UpdatePosition}(V_1^{LR}(P, A(P))), \ldots, \text{UpdatePosition}(V_n^{LR}(P, A(P)))).
\]

Convergence can be guaranteed if \(H^{\text{area}}(P, \mathcal{V}^{LR})\) decreases monotonically during each iteration of the algorithm. The result in Lemma 5 assumes computation of \(A\) is exactly obtained from the distributed consensus algorithm and that the weight assignment exactly satisfies the area constraints.

Lemma 5. Algorithm 1 converges to a solution \((P^*, \mathcal{V}^{LR}(P^*, \omega^*))\) when \(H = H^{\text{area}}(P, \mathcal{V}^{LR})\).

A convergence proof for Algorithm 1 using \(H^{\text{mixed}}(P, \mathcal{V}^{LR})\) is not available. The monotonic decrease of \(H^{\text{mixed}}(P, \mathcal{V}^{LR})\) cannot be guaranteed, because the area of the cell is not guaranteed to be non-decreasing. Simulations in Section V-B show that Algorithm 1 converges in a convex case, and result in an overall decrease of the function, but there are iterations when \(H^{\text{mixed}}(P, \mathcal{V}^{LR})\) increases.

B. Alternate Definition of \(D\) using Maximum Radius

This section assumes the robots have no dynamics and that the cost is either \(J(p_i, q) = \|p_i - q\|^2\) or \(J(p_i, q) = \|p_i - q\|\). The reachable set, \(D_i\), then becomes a ball with radius \(R_i\) for all \(i \in \{1, \ldots, n\}\). Recall \(R_i \triangleq c + \omega_i - \frac{1}{n} \sum_{k=1}^{n} \omega_k\), then
\[
J(p_i, q) = \|p_i - q\|^2 \text{ implies } R_i = R_i^{1/2} \text{ and } J(p_i, q) = \|p_i - q\| \text{ implies } R_i = R_i^c.
\]

When the physical system require an upper bound, \(R_{\text{max}}\), be imposed on \(R_i\), \(c\) becomes a function of \(R_{\text{max}}\) and \(\omega^0\)
\[
c_{\text{max}} = R_{\text{max}} - \max_{\omega}(\omega) + \frac{1}{n} \sum_{k=1}^{n} \omega_k.
\]
Substituting \(c_{\text{max}}\) into the equation for \(R_i\) gives, \(R_i = R_{\text{max}} - \max_{\omega}(\omega) + \omega_i\). Notice that \(R_i\), and hence \(R_i\), are now dependent on \(R_{\text{max}}\) and the maximum value of \(\omega\). Algorithm 1 needs two modifications when subject to the maximum radius constraint: The inputs of Lines 5 and 11 change, VoronoiPartition\((P, \omega, R_{\text{max}}, R_i)\) and \(R_i\) is determined after Line 9.

Defining \(D_i\) using \(R_{\text{max}}\) causes problems in Lemma 4 because \(\max_{\omega} \omega\) is not differentiable. However, given the max
operator properties, one can conjecture that an analogous result exists using generalized gradients. As a consequence, assuming the analogous result leads to \( M(J) \) being gradient (i.e. that the weights-to-area map is in the generalized gradient of \( F \)) then all other results follow. With the assumption that Lemma 4 holds, then a weight assignment exists that satisfies the area constraint. Then, the convergence result in Lemma 5 still holds for the \( D_i \) definition for \( R_{\text{max}} \).

C. Distributed Properties of Algorithm 1

While the algorithm in [2] 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. 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 \( V^\text{LR}_i(P, \omega) \) requires knowledge of the positions and weights of agent \( i \)'s neighbors and knowledge of the mean of the weights for \( D_i \). The latter can be computed using a distributed consensus algorithm performed over a connected communication graph, not necessarily \( G_{\text{LR}} \). If \( D_i \) is defined as in Section IV-B, then the agents need the maximum value instead of the mean. The maximum value is found using a max operation over a connected multi-agent system, requiring fewer communications between the agents.

Before each weight update, the area covered by \( V^\text{LR} \) 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, \( V^\text{LR}_i \) to determine \( \omega^+_i \). Likewise, the position update using the gradient of \( H^\text{area}(P, V^\text{LR}) \) or the centroid of \( V^\text{LR}_i \) for \( H^\text{mixed}(P, V^\text{LR}) \), only requires knowledge from the agent’s own cell. In all, the algorithm is distributed over the smallest connected graph containing \( G_{\text{LR}} \).

The \( R_{\text{max}} \) constraint is used to define a disc graph \( G_{2R_{\text{max}}} \) 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 \( R_{\text{max}} \) intersect, \( B(p_i, R_{\text{max}}) \cap B(p_j, R_{\text{max}}) \neq \emptyset \). The \( G_{2R_{\text{max}}} \) is used to determine an over-approximation of the sets of neighboring agents in \( G_{\text{LR}} \). To see this, note that \( V^\text{LR}_i \subseteq D_i \subseteq B(p_i, R_{\text{max}}) \), for all \( i \in \{1, \ldots, n\} \), and therefore, \( V^\text{LR}_i \cap V^\text{LR}_j \neq \emptyset \) only if \( D_i \cap D_j \neq \emptyset \), which happens only if \( B(p_i, R_{\text{max}}) \cap B(p_j, R_{\text{max}}) \neq \emptyset \). This implies that by communicating only with neighbors \( j \) in \( G_{2R_{\text{max}}} \), an agent \( i \) can compute its cell \( V^\text{LR}_i \). An agent only needs to communicate with other agents within a radius of \( 2R_{\text{max}} \) of itself.

V. SIMULATIONS

The simulations are 15 agents minimizing \( H^\text{area}(P, V^\text{LR}) \) and \( H^\text{mixed}(P, V^\text{LR}) \) with \( J(p_i, q_i) = ||p_i - q_i||^2 \), a convex \( Q \), and a uniform \( \phi(q) \). The simulations also compare defining \( D_i \) with a constant maximum radius \( c \) and a maximum radius \( R_{\text{max}} \). In general, the maximum range is not intuitive to determine. But, when the cost \( J(p_i, q_i) = ||p_i - q_i||^2 \), the becomes the radius of a ball squared minus some error that is a function of \( \omega, c = ||p_i - q_i||^2 - \omega_i + \frac{1}{n} \sum_{k=1}^{n} \omega_k = R_i^2 + \epsilon(\omega) \).

In other words, the radius of \( D_i \) is \( R_i = \sqrt{c - \epsilon(\omega)} \).

For all simulations, the agents are initially positioned together in the lower left corner and the weights are initially all one. Note that the initial positions of the agents do not allow for the existence of a feasible set of weights, \( \omega \in U \), and therefore the area constraint, \( a' \), is not satisfied. The algorithm moves the agents away from one another and eventually the area constraint is satisfied. The convergence of the simulations is unaffected by these initial conditions.

A. Area-Only Cost Function

This section compares the limited range algorithm results using \( H^\text{area}(P, V^\text{LR}) \). The final \( V^\text{LR} \) partition for \( c = 50 \) is in Fig. 1a and for \( R_{\text{max}} = 15 \) is in Fig. 1b. How \( H^\text{area}(P, V^\text{LR}) \) evolves is shown for the \( c = 50 \) partition in Fig. 2a and the \( R_{\text{max}} = 15 \) case in Fig. 2b. As discussed in Section IV-A, the cost monotonically decreases at each partition update and stays constant during the agent position update.

Next, the limited range radii are compared, Fig. 3a for \( c = 50 \) and Fig. 3b for \( R_{\text{max}} = 15 \). The \( c = 50 \) radii eventually converge to a radius of \( R \approx 7 \), while the \( R_{\text{max}} \) radii converge to values close to the maximum allowable radius.

B. Mixed Cost Function

Below are the results for the limited range algorithm using \( H^\text{mixed}(P, V^\text{LR}) \). Fig. 4a and Fig. 4b contain the final \( V^\text{LR} \) partition for \( c = 50 \) and \( R_{\text{max}} = 15 \), respectively. The evolution of \( H^\text{mixed}(P, V^\text{LR}) \) for a \( c = 50 \) partition and
The limited range sub-partition satisfies the variable area constraint and an weights. A weight assignment exists that makes the limited range sub-partition defined and shown to be invariant under translation of the load balancing problem. A limited range sub-partition is proven to converge.

An algorithm to solve the limited range spatial load balancing problem for the area-only cost function is given. Agent position update laws are derived. Simulations confirm the analytical results.

The agents’ limited range radii are compared in Fig. 6a, $c = 50$, and Fig. 6b, $R_{\text{max}} = 15$. The radii when $c = 50$ initially are spread out and then converge to a common $R_i \approx 7$. When the maximum radius is used there is always at least one agent whose radius is the maximum value. As the algorithm progresses the agents radii converge to different values close, but not equal, to the maximum radius.

Future research directions include a convergence result for the mixed cost function algorithm and simulations for a non-convex environment. To solve the problem in a non-convex environment, the results of this paper will be combined with those from the authors’ previous work on graph-based spatial load balancing which uses sampling-based motion planners to determine the optimal paths.

ACKNOWLEDGMENT

This work was supported by Los Alamos National Laboratory and is approved for release under LA-UR-16-27078.

REFERENCES