A Mass Transportation Method of Domain Decomposition for Multi-Agent Teams with Heterogeneous Coverage Capabilities

J.D. Walsh III[1], Sean Maxon[2], Qiuyang Tao[2], and Fumin Zhang[2]

***Abstract*— We propose a mass transport method that decomposes the coverage region in a nonhierarchical manner that is nearly optimal with respect to some user-defined cost function. This method does not rely on any nonlocal information, giving it effectively constant scaling behavior, making it ideal for large swarms. It dynamically adapts to changing conditions and guarantees full region coverage, even when agents are removed from or added to a team. In support of the method, we share simulation and live-test UAV results.**

I. INTRODUCTION

We propose a new method of domain decomposition for heterogeneous multi-agent teams. It generalizes Voronoi cells using concepts from the theory of mass transport. The method describes how to decompose the coverage area and how to identify the boundaries of each region to whatever precision one desires. The decomposition uses completely local information exchange. Two agents need only communicate if they share a boundary. It also offers direct control of arbitrarily-sized region masses, where *mass* is some measure of an agent's coverage capabilities. Overall coverage is guaranteed, and if the region is partitioned the coverage is optimal with respect to some user-defined *cost* function that represents the expense of point-to-point movement. The method requires extremely low bandwidth, as it incorporates a single shared mass transport model, where the state of each agent's relationship to the model is captured by sharing four numbers: a 3D position and a single *shift* value that indicates relative boundary placement.[1]

In the existing literature, many methods for multi-agent teams rely on some method of decomposing the domain to be covered. A common approach uses Voronoi cells to perform such a decomposition; e.g., [1]. Some research describes generalized Voronoi decomposition, but the claim of domain decomposition remains purely theoretical, and changes in coverage over time are not directly addressed; e.g., [2]. All of these Voronoi methods rely on indirect approaches to controlling region mass.

Our method offers direct control over region mass, and explains the connection in an intuitive fashion. It also describes exactly how to decompose the coverage area and how to identify the boundaries of the region to whatever precision one desires. The method evolves naturally over time, and because of its mass transport definition, it forces agents to dynamically adapt to changing conditions, such as losing or gaining team members. This makes it ideal for sustaining long-term operations. The method is nonhierarchical, so the loss of individual agents does not impact team cohesion. At worst the team may split into multiple independent subteams.

In our computer simulations, the method showed rapid convergence to nearly-optimal conditions. When tested on unmanned aerial vehicles (UAVs) in a disruptive environment, the method allowed our agents to adapt dynamically to those changes, even when teammates were lost and regained.

[1]J.D. Walsh III is with the Naval Surface Warfare Center, Panama City Division, 110 Vernon Ave., Panama City, FL 32407, USA joseph.d.walsh@navy.mil

[2]Department of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive NW, Atlanta, GA 30332, USA {sean.maxon, qtao7, fumin}@gatech.edu

[1]in 2D, only three numbers are required.

II. THEORY

A. Problem setup

Suppose we have n agents. These agents will cover a region in $\mathbb{R}^d$, where $d \in \{2, 3\}$, and we know the region is a subset of a compact and convex set S. To measure the expense of coverage, we have defined a cost function $c : S \times S \to [0, \infty)$. The cost function must be continuous and reflexive; that is, $c(\mathbf{x}, \mathbf{y}) = c(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in S$.

The value of regions in S is defined by a probability density function (pdf) $\mu : S \to [0, \infty)$. Higher values of μ indicates higher valued coverage regions. The function μ need not be continuous, but it must be nonatomic (so there are no point-masses) and bounded: there exists some $M \in [0, \infty)$ such that $\mu(\mathbf{x}) \leq M$ for all $\mathbf{x} \in S$.

Define $\mathbb{N}_n := \{1, \ldots, n\}$. For each $i \in \mathbb{N}_n$, we know agent i is at position $\mathbf{p}_i$, and wishes to cover a region whose value is $\nu_i \in (0, 1)$. Assume $\sum_i \nu_i = 1$. Then we can define $\nu(\mathbf{x}) = \nu_i$ if $\mathbf{x} = \mathbf{p}_i$, and otherwise $\nu(\mathbf{x}) = 0$, which makes the function $\nu : S \to [0, 1)$ a discrete pdf. Using this information, we can apply mass transport theory to the problem of dividing S into n coverage regions.

B. Standard Mass Transport

First, consider a specific type of mass transport problem known as the Monge problem. For simplicity, treat the destination set as S, even though ν is zero on $S \setminus \{\mathbf{p}_i\}$.

Definition 2.1 (Monge problem): Let $S \subseteq \mathbb{R}^d$, let μ and ν be probability densities defined on S, and let $c(\mathbf{x}, \mathbf{y}) : S \times S \to \mathbb{R}$ be a measurable *ground cost* function. Define the set of *transport maps*

$$\tau(\mu, \nu) := \{T : T_{\#}(\mu) = \nu\}, \tag{1}$$

where $T_{\#}(\mu)$ indicates the *push forward* of μ by T.

We define the *primal cost* function $P : \tau \to \mathbb{R}$ as

$$P(T) := \int_S c(\mathbf{x},\, T(\mathbf{x}))\, d\mu(\mathbf{x}). \tag{2}$$

The Monge-Kantorovich problem is to find the *optimal primal cost*

$$P^* := \inf_{T \in \tau} P(T), \tag{3}$$

and an associated *optimal transport plan*

$$T^* := \arg\inf_{T \in \tau} P(T). \tag{4}$$

The optimal primal cost is also known as the *Wasserstein distance* or *metric*. When c is the Euclidean distance, the optimal primal cost is sometimes referred to as the *earth mover's distance*.

The constraints we have chosen guarantee that we can find a solution. Moreover, because c is continuous and μ is nonatomic, that solution equals the solution to the (more general) Monge-Kantorovich problem; see Theorem B in [3] for details. In other words, there is at least one transport map $T^* \in \tau$ such that $P(T^*)$ equals the optimal primal cost of the Monge-Kantorovich problem, and so solving the Monge problem is equivalent to solving the Monge-Kantorovich mass transport problem.

Since each $T \in \tau$ is a map, each T partitions S into n sets S_i, where S_i is the set of points in S that are transported by the map T to $\mathbf{p}_i$:

$$S_i := \{\, \mathbf{x} \in S \mid T(\mathbf{x}) = \mathbf{p}_i \,\}. \tag{5}$$

This allows us to rewrite the primal cost equation in a form that will be more useful to us:

$$P(T) := \sum_i \int_{S_i} c(\mathbf{x},\, \mathbf{p}_i)\, d\mu(\mathbf{x}). \tag{6}$$

In order to formulate transport as a map, we have written the transport problem in terms of transporting S to $\mathbf{p}_i$, even though in practice we would move the agent, rather than the region. We can do this because c is reflexive, which allows us to reorder the pdfs as desired without changing the solution.

The Monge (and Monge-Kantorovich) formulations implicitly assume that one will solve the problem by fixing the functions μ and ν, while perturbing the transport plan in some fashion until the infimum is achieved. Indeed, many computational methods take this approach, though as we will see, there are other options.

One of the most common classes of transport-perturbing methods is the simplex methods. They discretize S and treat the problem as fully discrete transport across a directed bipartite graph. Beginning with any $T \in \tau$, the methods repeatedly transform the map into a related, lower cost map, until the minimum is achieved. See Section III of [4] for details.

Because $\{\mathbf{p}_i\}$ is embedded in a continuous space S, one can also use methods developed for partial differential equations. These methods treat the problem as a Monge-Kantorovich solution, using descent techniques to find an optimal transport plan π^*. However, such methods tend to struggle with the discontinuity of ν.

C. Shift Characterization

As noted above, the Monge and Monge-Kantorovich mass transport formulations presume that μ and ν are fixed, and frame the solution in terms of varying the transport plan or map (π or T, respectively). For the semi-discrete problem we are considering, there is another option: We can define the mass transport problem in such a way that the transport map is always optimal, but the destination density ν may not match our desired pdf. In this case, the solution is achieved by perturbing ν, with each perturbation guaranteed to be optimal for some (hopefully more similar) mass transport problem.

Using this idea, we are ready to describe the shift characterization of the semi-discrete optimal transport problem. The definition of the characterization, which follows, is based on one given by Rüschendorf and Uckelmann in [5], [6].

Definition 2.2 (Shift Characterization): Let $S \subseteq \mathbb{R}^d$. Let μ be a nonatomic probability density defined on S. Let ν be a probability density on S with exactly n non-zero values on $\mathbf{p}_i$. Let $c(\mathbf{x},\, \mathbf{y}) : S \times S \to [0,\, \infty)$ be a continuous and measurable *ground cost* function.

Assume there exists $\sigma_i \in \mathbb{R}$ for all $i \in \mathbb{N}_n$. Define

$$F(\mathbf{x}) := \max_i \{\sigma_i - c(\mathbf{x},\, \mathbf{p}_i)\} \tag{7}$$

and

$$S_i := \{\, \mathbf{x} \in S \mid F(\mathbf{x}) = \sigma_i - c(\mathbf{x},\, \mathbf{p}_i) \,\}. \tag{8}$$

Then $\cup_{i=1}^n S_i = S$.

The problem of determining an optimal transport map T^* is equivalent to determining shifts $\{\sigma_i\}$ such that for all $i \in \mathbb{N}_n$, the total mass transported from S_i to $\mathbf{p}_i$ equals $\nu(\mathbf{p}_i)$.

As [7] states, the special case $c = \|\cdot\|_2^2$ has two well-established methods that are specifically designed for solving semi-discrete problems directly: the Oliker-Prussner algorithm and the damped Newton methods proposed in papers like [8]. Both rely on some variant of Oliker and Prussner's seminal 1988 paper; see [9].

A significant amount of research effort is currently being devoted to developing methods for shift characterized mass transport problems using other cost functions. See [10] for an example that works for all p-norms with $p \in (1,\, \infty)$, as well as a detailed bibliography of other methods.

D. Boundary Regions and Partitioning

If the sets S_i partition S a.e., then T, defined as

$$T(\mathbf{x}) = \{\, \mathbf{p}_i \mid \mathbf{x} \in S_i \,\}, \tag{9}$$

is a map a.e. Partitioning is guaranteed by many cost functions, including all of the p-norms with $p \in (1,\, \infty)$; see [11]. If the sets S_i do not constitute a partition a.e., the overlapping region can be split in any way desired, subject to mass restrictions.

Consider the overlap regions. For all $i,\, j \in \mathbb{N}_n$ such that $i \neq j$, let

$$S_{ij} := S_i \cap S_j. \tag{10}$$

The *boundary set* is defined as

$$B := \bigcup_{1 \le i < n} \bigcup_{i < j \le n} S_{ij}. \tag{11}$$

For all $i, j \in \mathbb{N}_n$ such that $i \neq j$, define $g_{ij} : S \to \mathbb{R}$ as

$$g_{ij}(\mathbf{x}) := c(\mathbf{x}, \mathbf{p}_i) - c(\mathbf{x}, \mathbf{p}_j). \tag{12}$$

As described in [10], $B \neq \varnothing$ and for each $\mathbf{x} \in B$ there exist $i, j \in \mathbb{N}_n$, $i \neq j$, such that $\mathbf{x} \in S_{ij}$. Because $\mathbf{x} \in S_i$, we have $F(\mathbf{x}) = \sigma_i - c(\mathbf{x}, \mathbf{p}_i)$, and because $\mathbf{x} \in S_j$, we have $F(\mathbf{x}) = \sigma_j - c(\mathbf{x}, \mathbf{p}_j)$. Combining and rearranging these, gives us

$$g_{ij}(\mathbf{x}) = \sigma_i - \sigma_j \qquad \forall \mathbf{x} \in S_{ij}. \tag{13}$$

Thus, (13) implies that S_{ij} is a subset of a level set of g_{ij}; the value $\sigma_i - \sigma_j$ is constant, regardless of which $\mathbf{x} \in S_{ij}$ is chosen. Using this information, for each $i, j \in \mathbb{N}_n$, $i \neq j$, such that $S_{ij} \neq \varnothing$, we can define the constant *shift difference*

$$\sigma_{ij} := \sigma_i - \sigma_j = g_{ij}(\mathbf{x}) \qquad \forall \mathbf{x} \in S_{ij}. \tag{14}$$

E. Domain Decomposition

Standard approaches to the Wasserstein distance problem presume global information, and use that information to find a global solution. We propose a different approach: decompose the mass transport problem and use purely local information to quickly approximate a global solution. To do this, we need a way of decomposing S, and a meaningful local approach to perturbing individual shifts.

As described above, the boundaries of each S_i are determined by the shift differences of agent i and its immediate neighbors. Hence, we can consider decomposing the domain S based on those neighbor relationships.

Let $N_i \subseteq \mathbb{N}_n \setminus \{i\}$ and suppose that, for all agent j such that $S_{ij} \neq \varnothing$, $j \in N_i$. If agent i knows $\mathbf{p}_j$ and σ_j for all $j \in N_i$, then agent i can compute g_{ij} for all of its immediate neighbors. This is sufficient to partition S into S_i and $S \setminus S_i$. Furthermore, if we know that agents beyond a certain range are too far away to be neighbors, we can choose to exclude them from computations.

Suppose some neighbor $j \notin N_i$. For all $\mathbf{x} \in S_j \setminus S_i$, it is possible that agent i will believe $F(\mathbf{x}) = \sigma_i - c(\mathbf{x}, \mathbf{p}_i)$, while in fact $F(x) = \sigma_j - c(\mathbf{x}, \mathbf{p}_j) > \sigma_i - c(\mathbf{x}, \mathbf{p}_i)$. (We say possible here because there may be some agent $k \in N_i$ such that $\sigma_k - c(\mathbf{x}, \mathbf{p}_k) > \sigma_i - c(\mathbf{x}, \mathbf{p}_i)$.) Hence, i will approximate S_i by computing the larger region $\hat{S}_i \supseteq S_i$. Since this is true for each agent i, it follows that $\cup_i \hat{S}_i = S$, and coverage is guaranteed even if some N_i does not contain all the neighbors of agent i. If, however, every N_i contains all the neighbors of agent i, then $\hat{S}_i = S_i$ for all i. In this case, $\{\hat{S}_i\}$ partitions S μ-a.e. if $\{S_i\}$ does.

For any agent i, a maximum number of neighbors must exist, and that number is independent of the total number of agents n. The actual maximum number would be difficult to compute, as it is related to the minimum radius of the region surrounding each agent, which is itself a function of μ and ν. However, we can gain an intuition about the maximum by considering the unequal packing problem, which suggests that the upper bound is a few dozen agents. In practice, since the computational complexity of determining S_i is dependent on the number of neighboring agents, the complexity of determining $\hat{S}_i$ has a fixed upper bound which is independent of n.

This is relevant because we can use $\hat{S}_i$ to compute a good perturbation for σ_i. There is a monotonic relationship between S_i and σ_i, and when μ is continuous that relationship is nearly linear in c.

Let $\Delta\hat{S}_i$ be the region for agent i computed with the shift $\sigma_i + \Delta\sigma_i$, with all other conditions unchanged. Let R be the $(d-1)$-dimensional measure of the boundary surface S_{ij}. If $\Delta\sigma_i$ is sufficiently small and μ is continuous in some neighborhood of S_{ij}, then

$$\mu(\Delta\hat{S}_i) \approx \mu(\hat{S}_i) + \Delta\sigma_i R m, \tag{15}$$

where m is some constant in $[0, M]$.

III. METHOD

The centerpiece of our mass transport method is an algorithm for updating the position and shift of an individual agent i. For ease of notation, we assume $S_i = \hat{S}_i$ except where an explicit distinction is needed.

Data: c, μ, ν, $(\mathbf{p}_i, \sigma_i)$, $\{ (\mathbf{p}_j, \sigma_j) \mid j \in N_i \}$
Result: updated $(\mathbf{p}_i, \sigma_i)$
1 determine S_i;
2 compute $\mu(S_i)$;
3 find $\overline{\mathbf{p}}_i$, the centroid of S_i;
4 compute $\Delta\sigma_i \propto \nu_i - \mu(S_i)$;
5 $\sigma_i \to \sigma_i + \Delta\sigma_i$;
6 move $\mathbf{p}_i$ in the direction of $\overline{\mathbf{p}}_i$;
7 broadcast updated $\mathbf{p}_i$ and σ_i;

Algorithm 1: Region Decomposition algorithm

The most complex part of the algorithm is actually the first step: determining S_i. We deal with it by exploiting the level-set structure of the boundary. If we can find a set of well-chosen boundary points, the level set equation gives us an analytic formula for the lines (surfaces) in between. Obtaining the boundaries allows us to complete steps 2 and 3: computing the integrals necessary to evaluate the mass $\mu(S_i)$ and centroid $\mathbf{p}_i$. From there, the other steps are relatively straightforward.

A. Finding Boundary Points

To identify the boundary of S_i, we look for intersections of $d+1$ boundary regions. This number is sufficient to guarantee that the intersection, if it exists, is a point. The boundary regions in question can be the boundary of S, or of one or more agent regions. Since each of these intersections includes S_i, we consider every combination of up to d neighbors from N_i. The number of neighbors required depends on the type of boundary constraint. For example, when using the unit cube for S, we considered combinations constrained to:

- a corner (no neighbors required),
- an edge (plus one neighbor),
- a face (plus two neighbors),
- the interior (with three neighbors).

We search for an intersection via a descent method.

Data: $\mathbf{p}_i$, zero or more $\mathbf{p}_j$s, constraints
Result: (if it exists) $\mathbf{x}$ in constrained region, at the intersection of ∂S_i and zero or more boundaries ∂S_j
Initialize: choose constrained $\mathbf{x}$ at the center of the convex hull defined by $\mathbf{p}_i$ and the $\mathbf{p}_j$s
do
  foreach $\mathbf{p}_j$ *with* $\mathbf{p}_i \neq \mathbf{p}_j$ **do**
    if $g_{ij}(\mathbf{x}) > \sigma_i - \sigma_j$ **then**
      move $\mathbf{x}$ in constrained direction $\mathbf{p}_j - \mathbf{p}_i$;
    else if $g_{ij}(\mathbf{x}) < \sigma_i - \sigma_j$ **then**
      move $\mathbf{x}$ in constrained direction $\mathbf{p}_i - \mathbf{p}_j$;
    end
  end
while $\mathbf{x}$ *making progress and within* S;
if $\mathbf{x} \notin$ *desired intersection* **then**
  discard $\mathbf{x}$;
end

Algorithm 2: Descent algorithm

B. From Points to Region

Once we have our well-chosen boundary points, we evaluate the general form of g_{ij} to determine how best to approximate the boundary in between. Because we know we will be integrating over the region, we naturally prefer to approximate S_i as a polygon (in 3D, a polyhedron), and we choose accordingly.

For example, when c is the Euclidean distance, g_{ij} defines one of the two pieces of a hyperbola (in 3D, a hyperboloid of two sheets). In our experiments, we approximate the boundary regions in between our points by this method:

For each $j \in N_i$:
  Find a point in S_{ij} on the line segment $\overline{\mathbf{p}_i\mathbf{p}_j}$ (if such a point exists), and add it to the existing set of well-chosen boundary points.

Finding these constrained points is done with Algorithm 2. We then form a convex hull using these "region points," giving us a reasonably good polygonal (or polyhedral) approximation of S_i

C. From Region to Mass/Centroid (Quadrature)

One we have a polygonal (or polyhedral) approximation of S_i, we use quadrature to approximate the mass and centroid of the region. If desired, the Wasserstein distance $\int_{S_i} c\, d\mu$ may also be computed.

Here, the ideal method depends on the pdf μ. For example, when μ is the uniform density, the approximation of $\mu(S_i)$ is the volume of the polygon (polyhedron), and the centroid is the center of mass. In our experiments, we assumed μ uniform and broke the approximated S_i into triangles (or trapezoids) for computation. The mass is the sum of these sub-masses, and the centroid is the weighted average of the sub-centroids.

D. Computing Shift Update

We expect shift updates to respond well to a linear approximation, but the exact method will be highly dependent on c and μ. For our experiments, we assumed the most obvious relation:

$$\Delta\sigma_i = \nu_i - \mu(S_i).$$

However, for the sake of stability, we subjected $\Delta\sigma_i$ to two additional constraints: the new region

1) cannot leave the agent behind, and
2) cannot overrun any of agent i's neighbors.

We quantified these constraints by requiring that:

1) $\Delta\sigma_i < m_i$, where m_i is agent i's speed per step, and
2) $\Delta\sigma_i < \frac{1}{2}c(\mathbf{p}_i, \mathbf{p}_j)$.

Whenever $\Delta\sigma_i$ violates one of these constraints, we halve its value (repeatedly, if necessary).

IV. EXPERIMENTS

The algorithm was implemented for 2D and 3D using MATLAB. We used the Euclidean distance for c and uniform density for μ. The region to cover, S, was designated to be a unit cube, and agent speed per step was constrained to a maximum distance of 0.01 (with respect to c).

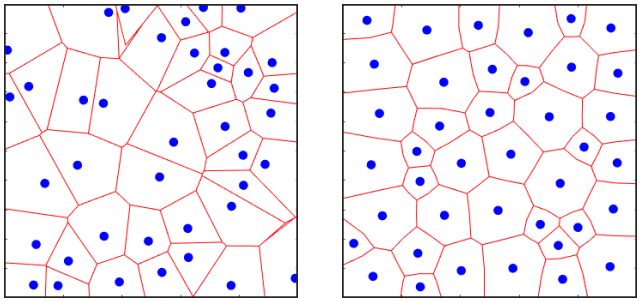
Communications are simulated using a global routine. Each agent receives only the position and shift information, and only from those agents it can see, as determined by the global communications routine. The agent then iterates through the algorithm in its sandbox. For simplicity, agent communications were synchronized. Since the method does not require any specific exit condition, tests were performed using a predetermined number of iterations.

A. Computer Simulations

Computer simulation tests were performed in 2D and 3D, with up to 1280 agents running for up to 5000 iterations. The algorithm scaled quadratically, and computation time was dominated by the global communications function. A typical run, using 40 heterogeneous agents, started as shown in Figure 1(a) and stabilized to the arrangement shown in Figure 1(b) after 128 iterations.

All shifts were initially identical, so the initial regions form a Voronoi partition, with no control over region masses. Due to the limited communication range, this caused some errors in the initial region approximations. Those errors were resolved by the next iteration.

Four different types of agents were used, with four distinct ν_i values. The final region masses match those specified in each individual agent's design. Initial positions were random. Final positions were found by each agent following



(a) Initial (frame 1) (b) Final (frame 128)

Fig. 1. Initial and final frames for 40 agents

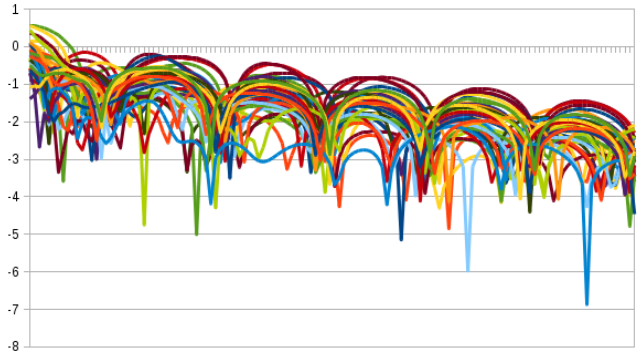


Fig. 2. Mass errors over time (semilog percentage plot)

its region's centroid, as specified in the algorithm, over successive iterations.

Figure 2 shows how the Mass errors for the agents decreased over time. The x-axis indicates the iteration number. Because the ν_is have multiple values, all errors are percentages. These percentages are given in a semilog plot (so 0 corresponds to 1%, -1 to 0.1%, and so on). Each region's error is a distinct line. By the last iteration, the worst mass error was 0.816%.

Figure 3 shows the change in the Wasserstein distance. The red upper line is the distance computed over time. Like most mass transport problems, the one created by our agents cannot be solve exactly, so the actual minimum solution is unknown. In order to gauge the quality of our Wasserstein distance, we computed an *idealized Wasserstein distance*: the value that would be achieved if each agent's region was a circle with center $\mathbf{p}_i$ and the exact radius necessary to contain $\mu(\hat{S}_i)$. (Since the idealized distance can only be achieved if the space between circles has measure zero, we know we cannot actually reach this value.) The blue line at the bottom of the graph shows the idealized Wasserstein distance for this set of agents. As we change transport problems, perturbing the destination pdf toward ν, the idealized Wasserstein distance also changes. The idealized Wasserstein distance was effectively constant after 72 iterations, and by the final iteration our system had stabilized to a Wasserstein distance within 1.75% of the idealized value. Further iterations did not noticeably alter the configuration of regions or further

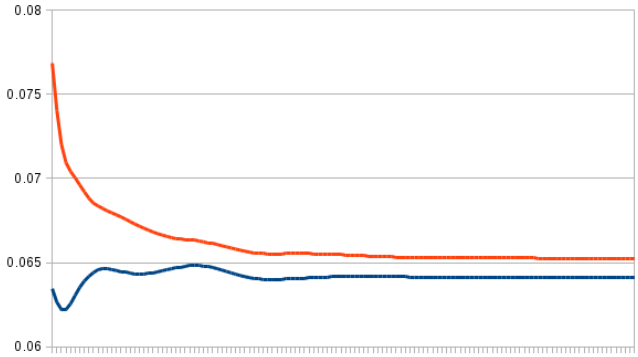


Fig. 3. Wasserstein distance over time

reduce the Wasserstein distance.

It is well known that the Wasserstein distance for a specific mass transport problem is always unique, and [10] shows that if $\mu(B) = 0$, the transport map is unique μ-a.e. However, these results presume that the agent positions are fixed.

The existence of the idealized Wasserstein distance guarantees that a globally minimum Wasserstein distance, gained by relaxing the requirement of fixed agent positions, must exist. If S has rotational symmetry under c, as our cube does, rotating the agent positions must result in the same optimal Wasserstein distance, even though it results in a new transport map. Hence, allowing agent movement, at least certain types of movement, can take away the μ-a.e. uniqueness of the transport map.

We performed multiple experiments where we changed the initial positions of our agents. It seemed that each instance generated a new and different transport map. However, in every case where we changed the agent positions, final Wasserstein distance achieved was approximately identical. These experimental results strongly suggest that the Wasserstein distances achieved are globally minimum, even when the fixedness of $\{\mathbf{p}_i\}$ is relaxed.

The Monge problem can have local minima. For example, in unit square, giving every $\mathbf{p}_i$ an x-value of ½ should generate a collinear agent distribution that is suboptimal for the 2D space (but would be optimal for a 1D projection onto $x = ½$). However, these lower-dimensional local minima are highly unstable. In practice, approximation errors pushed our set of agents out of the lower-dimensional configuration in every computational experiment we tried. Once a higher dimensional configuration was achieved, the method then progressed inexorably to the (presumed global) solution.

B. UAVs

We successfully performed laboratory tests with systems of two, three, and four unmanned aerial vehicles (UAVs). The MATLAB code for the algorithm was identical to that used in simulation. We ran the method's MATLAB software in real-time at a speed of 10 Hz for a period of two minutes (120 total iterations).

The UAVs we used were GT-MABs: Georgia Tech Miniature Autonomous Blimps [12]. These are small lighter-than-

air robots intended for indoor experiments and designed prioritizing safety, endurance, and omnidirectional maneuverability. Their lightweight design makes them susceptible to wind forces, so we tested our method in a laboratory where strong, intermittent drafts from the air conditioning system would cause destabilizing agent movement. The region S was designated as a $4\text{m} \times 4\text{m} \times 3\text{m}$ box. Agents leaving that region were considered "lost" and could not send or receive communications from other agents. If an agent reentered the region, communications would resume, resulting in "new" teammates (new from the perspective of both the lost agents and those who had stayed inside).

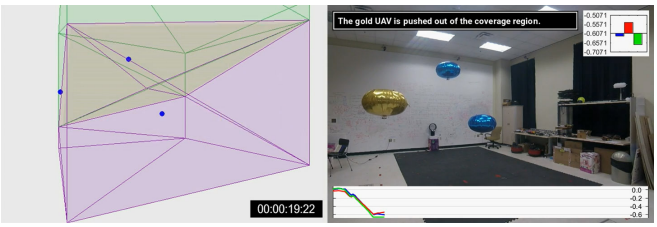


Fig. 4. Gold agent leaves S at 19 seconds

For a specific example, consider the results of one of our three agent tests (see video). At about 00:06 the air conditioning system turned on. By 00:19 the air currents pushed the gold agent (visible in Figure 4) outside of S. At that point, the two blue agents stopped receiving communications from the gold agent, and vice versa. Since the two blue agents could no longer hear their gold counterpart, they immediately set about dividing S between themselves. Meanwhile, the gold agent was no longer receiving any messages, so it assumed the other two agents were lost, and attempted to move to the centroid of S, in order to cover the entire region by itself.

By 00:30 the gold agent had successfully reversed course, and was heading back toward S. Shortly thereafter the air conditioning shut off. At 00:43 the gold agent reentered S, and communications were reestablished. The agents immediately began renegotiating an optimal 3-way split, which stabilized around 01:11. The swaying of the agents during this time stems from the reuse of our simulation code; the agents were attempting changes in position which were simply too precise to achieve, given their inertia.

At approximately 01:20 the air conditioning system turned on again. (The draft can be seen pushing the agents downward.) This destabilizes the arrangement enough that the agents rotate their positions, but they remain in a nearly-optimal configuration until the test ends at 02:00.

V. CONCLUSIONS

Our future plans include more extensive laboratory tests on a significantly larger fleet of UAVs. We are already looking at more complex cost and density functions, particularly those with practical applications in the aerospace and undersea domains. Used naively, asynchronous communication risks temporary gaps in our coverage of S. We are looking at approaches that will address that issue, should it occur.

An as-yet-untapped area of significant interest is approaches for shift negotiations that take advantage of information gathered from neighbors over time. The question of identifying persistant neighbors, given only position and shift information, is already being addressed.

Finally, we are looking at approaches that change the density μ over time in order to move the region of control while avoiding obstacles. This will allow the team to traverse a region, following a designated path, without moving in an easily predictable "formation." Our initial experiments suggest that the resulting behavior will closely resemble that found in nature, such as schooling fish and swarming gnats.

REFERENCES

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

[2] K.R. Guruprasad and D. Ghose, "Heterogeneous locational optimisation using a generalised Voronoi partition," *International Journal of Control*, vol. 86, issue 6, pp. 977–993, 2013.

[3] A. Pratelli, "On the equality between Monge's infimum and Kantorovich's minimum in optimal mass transportation," Annales de l'Institut Henri Poincare (B): Probability and Statistics, vol. 43, no. 1, pp. 1–13,2007.

[4] V. Chvátal, *Linear Programming*, W.H. Freeman: New York, 1983, pp. 289–402.

[5] L. Rüschendorf and L. Uckelmann, "Numerical and analytical results for the transportation problem of Monge-Kantorovich," Metrika, vol. 51, no. 3, pp. 245–258, 2000.

[6] L. Rüschendorf, "Monge-Kantorovich transportation problem and optimal couplings." Jahresbericht der Deutschen Mathematiker-Vereinigung, vol. 109, no. 3, pp. 113–137, 2007.

[7] J. Kitagawa, Q. Mérigot, and B. Thibert, "A Newton algorithm for semi-discrete optimal transport," arXiv:1603.05579 [cs], Mar. 2017.

[8] F. Aurenhammer, F. Hoffmann, and B. Aronov, "Minkowski-type theorems and least-squares clustering," Algorithmica, vol. 20, no. 1, pp. 61–76, 1998.

[9] V.I. Oliker and L.D. Prussner, "On the numerical solution of the equation $\frac{\partial^2 z}{\partial x^2}\frac{\partial^2 z}{\partial y^2} - \left(\frac{\partial^2 z}{\partial x \partial y}\right)^2 = f$ and its discretizations, I," Numerische Mathematik, vol. 54, pp. 271–293, 1988.

[10] J.D. Walsh, "The boundary method for semi-discrete optimal transport partitions and Wasserstein distance computation," arXiv:1702.03517 [cs], May 2017.

[11] J.D. Walsh, "Uniqueness of optimal solutions for semi-discrete transport with p-norm cost functions," arXiv:1705.09383 [cs], May 2017.

[12] S. Cho, V. Mishra, Q. Tao, P. Vamell, M. King-Smith, A. Muni, W. Smallwood,and F. Zhang, "Autopilot design for a class of miniature autonomous blimps," 2017 IEEE Conference on Control Technology and Applications (CCTA), 2017.