Probabilistic Swarm Guidance using Optimal Transport

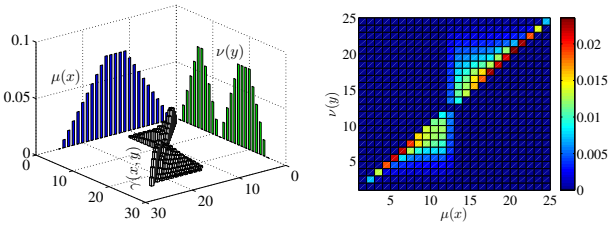
Saptarshi Bandyopadhyay *Student Member, IEEE*, Soon-Jo Chung *Senior Member, IEEE*, and Fred Y. Hadaegh *Fellow, IEEE*

***Abstract*—Probabilistic swarm guidance enables autonomous agents to generate their individual trajectories independently so that the entire swarm converges to the desired distribution shape. In contrast with previous homogeneous or inhomogeneous Markov chain based approaches [1], this paper presents an optimal transport based approach which guarantees faster convergence, minimizes a given cost function, and reduces the number of transitions for achieving the desired formation. Each agent first estimates the current swarm distribution by communicating with neighboring agents and using a consensus algorithm and then solves the optimal transport problem, which is recast as a linear program, to determine its transition probabilities. We discuss methods for handling motion constraints and also demonstrate the superior performance of the proposed algorithm by numerically comparing it with existing Markov chain based strategies.**

I. Introduction

Swarms of autonomous robots are widely studied because they can be controlled to collectively exhibit useful emergent behavior [2]–[5]. Similarly, swarms of hundreds to thousands of femtosatellites (100-gram-class satellites) are being developed for challenging formation flying missions like synthetic aperture radar, interferometry, etc [6]. In this paper, we introduce an optimal transport based approach to develop distributed probabilistic guidance algorithms for swarms of autonomous robots or spacecraft. Since deterministic algorithms of controlling individual agents scale poorly, we probabilistically control the swarm density distribution of a large number of index-free agents.

Probabilistic swarm guidance is centered on the idea of controlling the swarm density distribution in a distributed manner so that each autonomous agent or robot determines its own trajectory in an independent manner while the entire swarm converges to the desired distribution. Each agent transitions using a homogeneous Markov chain synthesized to achieve the desired formation as its stationary distribution [7]–[10]. However, our prior work [11] pointed out that the homogeneous Markov chains approach has the drawback of driving agents to other bins even after the overall swarm has sufficiently converged to the desired formation. This issue is solved in [1] by synthesizing an inhomogeneous (time-varying) Markov chain so that the agents settle down when the swarm converges to the desired formation. Moreover, these Markovian approaches are robust to external disturbances or damages to the formation.



(a) (b)

Figure 1. (a) The initial probability mass function (pmf) $\mu(x)$ (in blue) is transported to the desired pmf $\nu(y)$ (in green) while minimizing the cost function $c(x, y) = \|x - y\|_2$. The optimum transference plan $\gamma(x, y)$ is shown in gray. (b) Another view of the optimum transference plan $\gamma(x, y)$ from $\mu(x)$ to $\nu(y)$, where the colorbar represents the transported mass.

The main limitation of using Markov chains is that multiple iterations are necessary before the swarm finally converges to the desired formation. This results in significant wastage of control effort (e.g. fuel) for achieving the desired formation. Moreover, the Markov chain based approach does not give the option of optimizing trajectories for a given cost function. On the other hand, optimal transport is used to find the optimum transference plan $\gamma(x, y)$ from an initial distribution $\mu(x)$ to the desired distribution $\nu(y)$ with respect to the cost function $c(x, y)$ as shown in Fig. 1. In this paper, we develop the new probabilistic swarm guidance algorithm using optimal transport (PSG–OT) to find optimal trajectories for each agent in a distributed manner so that the given cost function is minimized. This algorithm also guarantees faster convergence and reduces the number of transitions required to attain or maintain the formation. This is achieved by each agent first estimating the current swarm distribution and then solving the optimal transport problem (OTP) to determine its transition probabilities. Since we control the swarm density distribution in a distributed manner, this algorithm is also robust to external disturbances or damages to the formation.

Optimal transport is used for transportation, resource allocation, assignment, etc [12]–[14]. The discrete OTP is solved using linear programming (LP) [15], [16]. In [17]–[19], LP is used, in a centralized manner, for formation flying applications and path planning of robots. In contrast, in this

S. Bandyopadhyay and S.-J. Chung are with the Department of Aerospace Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801, USA. (email: bandyop2@illinois.edu and sjchung@illinois.edu). F. Y. Hadaegh is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA (e-mail: fred.y.hadaegh@jpl.nasa.gov).

This research was supported in part by AFOSR grant FA95501210193 and the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

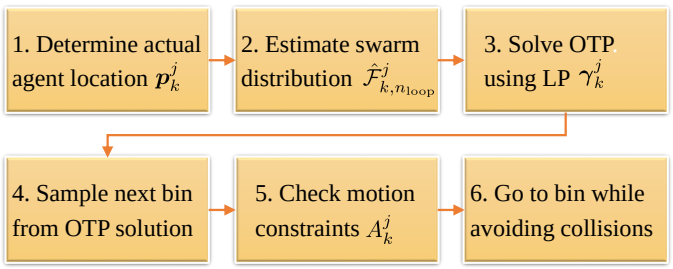


Figure 2. Flowchart of PSG–OT describing the key steps for a single agent in a single time step.

paper we solve, in a distributed manner, the discrete OTP of probabilistic guidance of swarms using LP.

The solution of the probabilistic guidance algorithm establishes the target bin for each agent. Collision avoidance algorithms are necessary for generating collision-free guidance trajectories from the present location to the target bin. A recent survey paper [20] compares a number of existing collision avoidance algorithms based on model predictive control, boundary following algorithms, artificial potential fields, etc. The guidance trajectories obtained from PSG–OT can be implemented using sequential convex programming and model predictive control [21], [22]. In this paper, we present a novel distributed collision avoidance algorithm for multiagent path planning by modifying the coverage control algorithm using Lloyd's descent [23]. The integrated collision avoidance algorithm, working in conjunction with the bin-based solution of the probabilistic guidance algorithm, results in a complete collision-free solution for the probabilistic swarm guidance problem.

A. Overview of PSG–OT and Contributions

The key idea of the proposed PSG–OT is to independently determine the optimal trajectory of each agent using an optimal transport based approach so that the swarm density distribution converges to the desired formation. The flowchart for the algorithm is shown in Fig. 2 and its pseudo code is given in **Algorithm 1**. In step 1, each agent determines its present location, as discussed in Section II-A. During step 2, each agent estimates the current swarm distribution by first making a localized guess and then reaching an agreement across the network using the consensus algorithm [24]. Step 2 is elucidated in Section II-B.

Step 3 involves solving the OTP using LP to obtain the transition probabilities to other bins for the current time step. The existence and properties of the OTP solution are presented in Section III. Random sampling of the optimal transport solution generates the next bin for the agent, as shown in step 4. Step 5 involves checking if the transition determined using by the OTP solution satisfies motion constraints. Methods for handling motion constraints are presented in Section IV. Finally, in step 6, the agent goes to the next bin while avoiding collision, as discussed in Section V.

The first contribution of this paper is to solve the discrete OTP in a distributed manner. A centralized node could solve an OTP using the exact location of each agent and allocate the final positions to each agent. Instead, in PSG–OT, each agent solves an equivalent OTP using its best estimate of the current swarm distribution and then determines its trajectory in an independent manner. Note that the equivalent OTP solved by each agent has n_{cell}^2 variables compared to the centralized OTP with m^2 variables, where n_{cell} is the number of bins, m is the number of agents and $m \gg n_{\text{cell}}$.

The second contribution of this paper is to use optimal transport for the problem of probabilistic swarm guidance. We show that this approach not only minimizes a given cost function, but also guarantees faster convergence and reduces the number of transitions for achieving and maintaining the formation, while obeying motion constraints.

The third contribution of this paper is a novel distributed collision avoidance algorithm, which is obtained by modifying the Voronoi partitioning based Lloyd's descent algorithm [23], which is suitable for the bin-based architecture of this paper. This is discussed in Section V. Hence this paper presents a complete collision-free solution for probabilistic swarm guidance problem, by integrating the collision avoidance algorithm with the OTP solution. Strategies for implementing PSG–OT and comparison with other existing Markov chain based approaches are discussed in Section VI.

B. Notation

The *time index* is denoted by a right subscript and the *agent index* is denoted by a lower-case right superscript. Let $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}^{m\times n}$ be the sets of natural numbers (positive integers), real numbers, positive real numbers and m by n matrices respectively. Let $\mathbf{1} = [1, 1, \ldots, 1]^T$, $\mathbf{I}$, and $\emptyset$ be the ones (column) vector, the identity matrix of appropriate size, and the empty set respectively. The symbols $|\cdot|$, $\lceil\cdot\rceil$, and $\|\cdot\|_p$ denote the absolute value or the cardinality of a set, ceiling function, and the ℓ_p vector norm respectively. The set $\mathcal{L}_p(\mu)$ denotes the set of all functions $f(\boldsymbol{x}) : \mathbb{R}^{n_x} \to \mathbb{R}$ with the bounded integral $\left(\int_{\mathcal{X}} |f(\boldsymbol{x})|^p d\mu(\boldsymbol{x})\right)^{1/p}$, where μ is a measure. For $f, g : \mathbb{N} \to \mathbb{R}$, we say that $f \in O(g)$ if there exists $n_0 \in \mathbb{N}$ and $k \in \mathbb{R}^+$ such that $|f(n)| \leq k|g(n)|$ for all $n \geq n_0$.

II. Problem Formulation and Preliminaries

In this section, we first present the problem statement and then present a distributed algorithm for estimating the current swarm distribution.

A. Problem Statement

The n_x-dimensional compact physical space over which the swarm is distributed is denoted by $\mathcal{R} \subset \mathbb{R}^{n_x}$ and partitioned into n_{cell} disjoint convex bins represented by $R[i]$, $i = 1, \ldots, n_{\text{cell}}$. Let $m \in \mathbb{N}$ agents belong to the swarm, where $m \gg n_{\text{cell}}$. We assume that each agent can determine its actual location in $\mathcal{R}$, which is denoted by $\boldsymbol{p}_k^j \in \mathbb{R}^{n_x}$. The row vector $\boldsymbol{r}_k^j$ represents the actual bin in which the j^{th} agent is present at the k^{th} time instant, i.e., if the j^{th} agent is in bin $R[i]$ (i.e., $\boldsymbol{p}_k^j \in R[i]$) then $\boldsymbol{r}_k^j[i] = 1$. The current swarm distribution ($\mathcal{F}_k^\star$), which is a row vector,

is given by the ensemble mean of actual agent positions, i.e., $\mathcal{F}_k^\star := \frac{1}{m}\sum_{j=1}^m \boldsymbol{r}_k^j$. Each agent in a particular bin can only transition to some bins, due the dynamics or physical constraints, which are captured by the motion constraints matrix A_k^j. Let us now define the desired formation and the cost function.

Definition 1. *(Desired Formation $\boldsymbol{\pi}$)* The desired formation shape is represented by a probability (row) vector $\boldsymbol{\pi} \in \mathbb{R}^{n_{\text{cell}}}$ over the bins in $\mathcal{R}$, i.e., $\boldsymbol{\pi}\mathbf{1} = 1$. Note that $\boldsymbol{\pi}$ can be any arbitrary probability vector, but it is the same for all agents within the swarm. In the presence of motion constraints, $\boldsymbol{\pi}$ needs to satisfy Assumption 1 on page 5. For example, $\boldsymbol{\pi}$ is the pmf $\nu(y)$ in Fig. 1. □

Definition 2. *(Cost Function $\boldsymbol{c}_k$)* The (possibly time-varying) cost of transitioning between bins is represented by the matrix $\boldsymbol{c}_k \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}$, where each element $\boldsymbol{c}_k[i,\ell]$ is the cost of transporting an agent from bin $R[i]$ to bin $R[\ell]$. For the existence of optimal transference plans, $\boldsymbol{c}_k$ should satisfy Theorem 1 on page 3. For example, $\boldsymbol{c}(x,y) = \|x-y\|_2$ in Fig. 1. □

The objectives of probabilistic swarm guidance using optimal transport (PSG–OT) running onboard each agent are as follows:

1) Determine the optimal trajectory of each agent with respect to given cost function ($\boldsymbol{c}_k$), which obeys motion constraints (A_k^j), so that the overall swarm converges to a desired formation ($\boldsymbol{\pi}$).
2) Maintain the swarm distribution and automatically detect and repair damages to the formation.

B. Estimation of Current Swarm Distribution using Consensus

In our prior work [1], the consensus algorithm [24]–[27] is used to estimate the current swarm distribution ($\mathcal{F}_k^\star$) in a distributed manner. We follow the same approach in this paper. Let the row vector $\hat{\mathcal{F}}_{k,\nu}^j$ represent the j^{th} agent's estimate of the current swarm distribution during the ν^{th} consensus loop at the k^{th} time instant. Each agent first locally estimates the swarm distribution, i.e., $\hat{\mathcal{F}}_{k,1}^j = \boldsymbol{r}_k^j$. Then the agents recursively combine their local estimates with their neighboring agents using the Linear Opinion Pool (LinOP) of probability measures [28], [29]:

$$\hat{\mathcal{F}}_{k,\nu+1}^j = \sum_{\ell \in \mathcal{J}_k^j} a_k^{\ell j} \hat{\mathcal{F}}_{k,\nu}^j,\ \forall j,\ell \in \{1,\dots,m\}, \forall \nu \in \mathbb{N}, \quad (1)$$

where $\mathcal{J}_k^j$ is the set of inclusive neighbors of the j^{th} agent and $\sum_{\ell\in\mathcal{J}_k^j} a_k^{\ell j} = 1$. The matrix P_k, with entries $P_k[\ell, j] = a_k^{\ell j}$, conforms with the time-varying communication network topology of the multi-agent system $\mathcal{G}_k$. Since $m \gg n_{\text{cell}}$ and multiple agents are within the same bin, it is guaranteed almost surely using Erdös–Rényi random graphs or random nearest–neighbor graphs that $\mathcal{G}_k$ is strongly connected (SC) [30]–[32]. Moreover, distributed algorithms exist for the agents to generate SC balanced graphs [33]–[35].

If $\mathcal{G}_k$ is SC and balanced, then each agents local estimate $\hat{\mathcal{F}}_{k,\nu}^j$ converges globally exponentially converges to the global estimate of the current swarm distribution $\mathcal{F}_k^\star = \sum_{i=1}^m \frac{1}{m}\hat{\mathcal{F}}_{k,1}^i$ pointwise with a rate faster or equal to the second largest singular value of P_k, i.e., $\sigma_{m-1}(P_k)$ [1]. For some $\varepsilon_{\text{cons}} > 0$, if the number of consensus loops within each consensus stage $n_{\text{loop}} \geq \left\lceil \frac{\ln(\varepsilon_{\text{cons}}/(2\sqrt{m}))}{\ln \sigma_{m-1}(P_k)} \right\rceil$; then each agent has a good estimate of the current swarm distribution ($\hat{\mathcal{F}}_{k,n_{\text{loop}}}^j$), i.e., $\|\boldsymbol{\theta}_{k,n_{\text{loop}}}\|_2 \leq \varepsilon_{\text{cons}}$, where $\boldsymbol{\theta}_{k,n_{\text{loop}}} = \left[\theta_{k,n_{\text{loop}}}^1, \dots, \theta_{k,n_{\text{loop}}}^m\right]$ and $\theta_{k,n_{\text{loop}}}^j = \sum_{R[i]\in\mathcal{R}} |\hat{\mathcal{F}}_{k,n_{\text{loop}}}^j[i] - \mathcal{F}_k^\star[i]|$. In this paper, we assume $\varepsilon_{\text{cons}} \leq \frac{1}{m}$ and n_{loop} is computed accordingly.

III. Optimal Transport Problem

The key concept of PSG–OT is that each agent can independently determine its optimal trajectory by solving an OTP using LP so that the swarm converges to the desired formation. In the absence of motion constraints, the objective of this OTP is to transport the swarm from its estimated current distribution ($\hat{\mathcal{F}}_{k,n_{\text{loop}}}^j$) to the desired formation ($\boldsymbol{\pi}$) while minimizing the cost function ($\boldsymbol{c}_k$).

Let $\boldsymbol{\kappa}[i] \in \mathbb{R}^{n_x}$ represent the centroid of the convex bin $R[i]$. The Dirac measure on $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the measure $\delta_{\boldsymbol{x}}(\boldsymbol{\kappa}[i]) = 1$ if $\boldsymbol{x} = \boldsymbol{\kappa}[i]$ and 0 otherwise. Then the measure induced by the j^{th} agent's estimated current swarm distribution ($\hat{\mathcal{F}}_{k,n_{\text{loop}}}^j$) on $\mathcal{R}$ is given by:

$$\mu_k^j(\boldsymbol{x}) = \sum_{i=1}^{n_{\text{cell}}} \hat{\mathcal{F}}_{k,n_{\text{loop}}}^j[i]\delta_{\boldsymbol{x}}(\boldsymbol{\kappa}[i])\,. \quad (2)$$

Similarly, for $\boldsymbol{y} \in \mathbb{R}^{n_x}$, the measure induced by the desired formation ($\boldsymbol{\pi}$) on $\mathcal{R}$ is given by:

$$\nu_k(\boldsymbol{y}) = \sum_{i=1}^{n_{\text{cell}}} \boldsymbol{\pi}[i]\delta_{\boldsymbol{y}}(\boldsymbol{\kappa}[i])\,. \quad (3)$$

The OTP is represented as the Monge–Kantorovich minimization problem [13]:

$$\boldsymbol{\gamma}_k^j(\boldsymbol{x},\boldsymbol{y}) = \arg\inf_{\varpi(\boldsymbol{x},\boldsymbol{y})} \int_{\mathcal{R}\times\mathcal{R}} \boldsymbol{c}_k(\boldsymbol{x},\boldsymbol{y}) d\varpi(\boldsymbol{x},\boldsymbol{y})\,, \quad (4)$$

where the infimum runs over all joint probability measures $\varpi(\boldsymbol{x},\boldsymbol{y})$ on $\mathcal{R}\times\mathcal{R}$ with marginals $\mu_k^j(\boldsymbol{x})$ and $\nu_k(\boldsymbol{y})$. The joint measures or couplings that achieve the optimum are called optimum transference plans ($\boldsymbol{\gamma}_k^j(\boldsymbol{x},\boldsymbol{y})$). The following theorem gives conditions for the existence of optimum transference plans.

Theorem 1. *[13, Theorem 4.1, pp. 43] (Existence of an optimal coupling) Let $\boldsymbol{a}(\boldsymbol{x}) \in L_1(\mu_k^j)$, $\boldsymbol{b}(\boldsymbol{y}) \in L_1(\nu_k)$ be two upper semi-continuous functions and $\boldsymbol{c}_k(\boldsymbol{x},\boldsymbol{y})$ be a lower semi-continuous cost function, such that $\boldsymbol{c}_k(\boldsymbol{x},\boldsymbol{y}) \geq \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{y}), \forall \boldsymbol{x},\boldsymbol{y}$. Then there exists an optimal coupling $\boldsymbol{\gamma}_k^j(\boldsymbol{x},\boldsymbol{y})$ which minimizes the total cost $\int_{\mathcal{R}\times\mathcal{R}} \boldsymbol{c}_k(\boldsymbol{x},\boldsymbol{y})d\varpi(\boldsymbol{x},\boldsymbol{y})$ among all possible couplings $\varpi(\boldsymbol{x},\boldsymbol{y})$ on $\mathcal{R}\times\mathcal{R}$ with marginals $\mu_k^j(\boldsymbol{x})$ and $\nu_k(\boldsymbol{y})$.*

In this paper, we use the ℓ_2 distance between the centroids of bins as the cost function, because it satisfies Theorem 1:

$$\boldsymbol{c}_k[i,\ell] = \left\|\boldsymbol{\kappa}[i] - \boldsymbol{\kappa}[\ell]\right\|_2, \forall i,\ell \in \{1,\ldots,n_{\text{cell}}\}, \forall k \in \mathbb{N}. \tag{5}$$

Moreover, the optimum transference plan $\boldsymbol{\gamma}_k^j$ has the following important properties [13]:

- **Cyclical monotonicity:** It is not possible to improve the incurred cost by rerouting masses along some cycle.
- **Duality:** The problem of minimizing cost of transportation has a dual Kantorovich problem where the objective is to maximize profits.
- **Restriction:** Given an optimum transference plan, any induced sub-plan (transferring part of the initial mass to part of the final mass) is optimal too.
- **Shortening principle:** Two distinct optimal transference trajectories do not intersect, except at endpoints.

The discrete OTP can also be written in the following form, where $\boldsymbol{\gamma}_k^j \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}$ is the optimum transference plan:

$$\begin{aligned} &\min \sum_{i=1}^{n_{\text{cell}}} \sum_{\ell=1}^{n_{\text{cell}}} \boldsymbol{\gamma}_k^j[i,\ell]\boldsymbol{c}_k[i,\ell] \qquad (6)\\ \text{subject to } &\sum_{\ell=1}^{n_{\text{cell}}} \boldsymbol{\gamma}_k^j[i,\ell] = \hat{\mathcal{F}}_{k,n_{\text{loop}}}^j[i], \quad \forall i \in \{1,\ldots,n_{\text{cell}}\}\\ &\sum_{i=1}^{n_{\text{cell}}} \boldsymbol{\gamma}_k^j[i,\ell] = \boldsymbol{\pi}[\ell], \quad \forall \ell \in \{1,\ldots,n_{\text{cell}}\}\\ &\boldsymbol{\gamma}_k^j[i,\ell] \geq 0, \quad \forall i,\ell \in \{1,\ldots,n_{\text{cell}}\} \end{aligned}$$

This problem can be solved using LP in time $O(n_v^2 n_q + e^{O(\sqrt{n_v \log n_v})})$, where n_v and n_q are the number of variables and linear inequalities respectively [17].

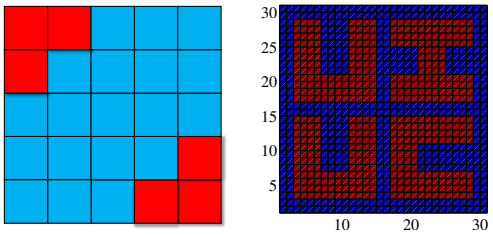
The optimal transference plan $\boldsymbol{\gamma}_k^j$, which is the solution of the discrete OTP, can also be interpreted in the Markovian sense. For a non-empty bin $R[i]$, the probability of the j^{th} agent transitioning from bin $R[i]$ to bin $R[q]$ during the k^{th} time instant is given by:

$$\mathbb{P}\left(\boldsymbol{r}_{k+1}^j[q] = 1 | \boldsymbol{r}_k^j[i] = 1\right) = \frac{\boldsymbol{\gamma}_k^j[i,q]}{\sum_{\ell=1}^{n_{\text{cell}}} \boldsymbol{\gamma}_k^j[i,\ell]}. \tag{7}$$

The Markov matrix, representing the transition probabilities of the j^{th} agent at the k^{th} time instant given by (7), does not have $\boldsymbol{\pi}$ as its stationary distribution. Hence it is dissimilar from the existing approaches [1], [10] utilizing the stationary distribution of the Markov chain to achieve the desired formation. But it is due to this Markovian interpretation that PSG–OT is also robust to external disturbances or damages to the formation.

IV. Motion Constraints

In this section, additional constraints on the motion of agents are considered and the cost function in PSG–OT is modified to handle them. The agents in a particular bin can only transition to some bins but cannot transition to other bins because of the dynamics or physical constraints. These (possibly time-varying) motion constraints are specified by the matrix A_k^j as follows [1]:



(a) (b)

Figure 3. (a) Let the red region denote the set Π. The motion constraints are such that the agents can only transition to the immediate neighboring bins. The two corner regions form the two subsets of Π, such that any agent cannot transition from one subset to another subset without exiting the set Π. (b) The letters UIUC (in red) is the desired formation $\boldsymbol{\pi}$ in Section VI. Although $\mathcal{R}$ is partitioned into 900 bins, there are about 448 non-empty bins.

$$A_k^j[i,\ell] = \begin{cases} 1 & \text{if the } j^{\text{th}}\text{agent can transition to } R[\ell] \\ 0 & \text{if the } j^{\text{th}}\text{agent cannot transition to } R[\ell] \end{cases},$$
$$\text{where } \boldsymbol{r}_k^j[i] = 1,\ \forall \ell \in \{1,\ldots,n_{\text{cell}}\}. \tag{8}$$

It is assumed that the graph conforming to the A_k^j matrix is strongly connected and the A_k^j matrix satisfies Assumption 1 on page 5.

In this paper, we capture the effect of these motion constraints by modifying the cost function (5). Let us choose a positive constant $c_{\max}$ such that $c_{\max} \gg \boldsymbol{c}_k[i,\ell], \forall i,\ell \in \{1,\ldots,n_{\text{cell}}\}, \forall k \in \mathbb{N}$. The modified cost function that captures the effect of the motion constraints is given by:

$$\tilde{\boldsymbol{c}}_k[i,\ell] = \begin{cases} \left\|\boldsymbol{\kappa}[i] - \boldsymbol{\kappa}[\ell]\right\|_2 & \text{if } A_k^j[i,\ell] = 1 \\ c_{\max} & \text{if } A_k^j[i,\ell] = 0 \end{cases},$$
$$\text{where } \boldsymbol{r}_k^j[i] = 1, \forall \ell \in \{1,\ldots,n_{\text{cell}}\}, \forall k \in \mathbb{N}. \tag{9}$$

In the presence of motion constraints, this modified cost function (9) is used in the discrete OTP (6). Note that the solution of the discrete OTP with the original cost function $\boldsymbol{c}_k$ (5) is a feasible solution for the discrete OTP with the modified cost function $\tilde{\boldsymbol{c}}_k$ (9). Hence, the existence of an optimal solution for the discrete OTP (6) with the modified cost function $\tilde{\boldsymbol{c}}_k$ (9) is guaranteed because the existence of at least one feasible solution guarantees existence of an optimal solution [15].

Let us define Π as the set of all bins that have nonzero probabilities in $\boldsymbol{\pi}$. For a bin $R[i]$, let us define $\mathcal{A}_k^j(R[i])$ as the set of all bins that the j^{th} agent can transition to at the k^{th} time instant. If the j^{th} agent is actually located in bin $R[i]$ and we observe that $\mathcal{A}_k^j(R[i]) \cap \Pi = \emptyset$ for all $k \in \mathbb{N}$, then the j^{th} agent is trapped in the bin $R[i]$ forever. Similar to [1], we avoid the trapping situation by enforcing a secondary condition that the j^{th} agent in bin $R[i]$ with $\mathcal{A}_k^j(R[i]) \cap \Pi = \emptyset$ will transition to the bin $\Psi_k^j(R[i])$ during the k^{th} time step, where the bin $\Psi_k^j(R[i]) \in \mathcal{A}_k^j(R[i])$ is closest to the bins in Π or does not have this trapping problem. In all other cases, if the solution to the OTP orders the agent to transition to a bin that is not allowed by the motion constraints, then the agent remains in its current bin.

In some situations, Π can be decomposed into subsets, such that any agent cannot transition from one subset to another subset without exiting the set Π due to motion constraints (for example see Fig. 3(a)). In PSG–OT, the agents will only transition within the bins in Π once it has entered any bin in Π. Hence the agents in one such subset of Π will never transition to the other subsets of Π because it has to travel through bins which are not in Π. In order to avoid such situations, we need the following assumption on Π and A_k^j.

Assumption 1. [1] Each agent can transition from any bin in Π to any other bin in Π without exiting the set Π, while satisfying the motion constraints. □

The pseudo code of PSG–OT, with motion constraints and collision avoidance, is shown in **Algorithm 1**.

Algorithm 1 Probabilistic swarm guidance algorithm using optimal transport (PSG–OT)

(one cycle of j^{th} agent during k^{th} time instant)
Agent determines its present location, e.g., $\boldsymbol{p}_k^j$, $R[i]$
Set n_{loop}, $a_k^{j\ell}$, and $\tau_{\max}$
for $\nu = 1$ to n_{loop} } Estimate current swarm distribution
  if $\nu = 1$ **then** Set $\hat{\mathcal{F}}_{k,1}^j = \boldsymbol{r}_k^j$ **end if**
  Exchange pmfs $\hat{\mathcal{F}}_{k,\nu}^\ell, \forall \ell \in \mathcal{J}_k^j$
  Compute the new pmf $\hat{\mathcal{F}}_{k,\nu}^j$ using (1)
end for
Compute the cost function $\tilde{\boldsymbol{c}}_k$ } Eq. (9)
Solve discrete OTP to get $\boldsymbol{\gamma}_k^j$ } Eq. (6)
Generate a random number $z \in \text{unif}\left[0; \sum_{\ell=1}^{n_{\text{cell}}} \boldsymbol{\gamma}_k^j[i,\ell]\right]$ } Random sampling
Choose bin $R[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{\gamma}_k^j[i,\ell] \le z < \sum_{\ell=1}^{q} \boldsymbol{\gamma}_k^j[i,\ell]$
if $A_k^j[i,q] = 1$ **then** Go to bin $R[q]$ } Check motion constraints
else if $\mathcal{A}_k^j(R[i]) \cap \Pi = \emptyset$
  then Go to bin $\Psi_k^j(R[i])$
else Remain in present bin $R[i]$ **end if**
for $\tau = 1$ to $\tau_{\max}$ } Generate collision-free trajectory
  Exchange locations $\boldsymbol{p}_{k,\tau}^\ell, \forall \ell \in \mathcal{J}_k^j$
  Compute $V_{k,\tau}^j$, $\tilde{V}_{k,\tau}^j$, and $\mathcal{C}(\tilde{V}_{k,\tau}^j)$
  if $\tilde{V}_{k,\tau}^j = \emptyset$, **then** $\boldsymbol{p}_{k,\tau+1}^j = \boldsymbol{p}_{k,\tau}^j$
  else if $R[q] = R[i]$,
    then $\boldsymbol{p}_{k,\tau+1}^j = \mathcal{C}(\tilde{V}_{k,\tau}^j)$
  else set $\boldsymbol{p}_{k,\tau+1}^j$ from (13) **end if**
end for

V. Collision-free Trajectory Generation

In PSG–OT, collision avoidance algorithms are necessary for obtaining the guidance trajectory of each agent from its current location to the target bin. In this section, we present a new distributed collision avoidance algorithm, by modifying the coverage control algorithm, which is suitable for the bin-based architecture of this paper.

The coverage control algorithm, which uses the Voronoi partitioning based Lloyd's descent algorithm [23], is designed for optimally distributing multiple agents over a given region. This distribution is optimal with respect to a ℓ_2-norm based cost function. An interesting property of these Voronoi partitions is that, if the given region is a convex polytope in an n_v-dimensional Euclidean space, then the boundary of each Voronoi partition is the union of (n_v-1)-dimensional convex polytopes and each Voronoi partition is itself a n_v-dimensional convex polytope.

In PSG–OT, at every time step, the agents in a particular bin either remain in the same bin or transition to target bins. Let us call the agents that remain in the same bin as stationary agents and those that transition to target bins as transient agents. Let each agent construct a Voronoi partition in its bin. The key idea for collision-free trajectory generation is to direct each stationary agent to the centroid of its Voronoi partition. Each transient agent is directed to the position in its Voronoi partition that has the minimum distance from its target bin.

Let r_{col} denote the minimum collision avoidance distance to be maintained between agents. Let $\boldsymbol{p}_{k,\tau}^j$ represent the actual location of the j^{th} agent during the τ^{th} collision avoidance loop at the k^{th} time instant. At the beginning of the collision avoidance loop at the k^{th} time instant, the actual location of the j^{th} agent is the final location of the j^{th} agent at the end of the previous time instant, i.e., $\boldsymbol{p}_{k,1}^j = \boldsymbol{p}_{k-1,\tau_{\max}}^j$ where $\tau_{\max}$ is the number of collision avoidance loops in each time instant. The distributed collision avoidance algorithm for agents in bin $R[i]$ during the τ^{th} collision avoidance loop at the k^{th} time instant is given as follows:

1) Let agents $\{j_1, j_2 \ldots, j_n\}$ be the set of all agents present in the bin $R[i]$, i.e., each $\boldsymbol{p}_{k,\tau}^{j_1}, \boldsymbol{p}_{k,\tau}^{j_2}, \ldots, \boldsymbol{p}_{k,\tau}^{j_n} \in R[i]$. Let $P_{k,\tau}(R[i]) = \{\boldsymbol{p}_{k,\tau}^{j_1}, \boldsymbol{p}_{k,\tau}^{j_2}, \ldots, \boldsymbol{p}_{k,\tau}^{j_n}\}$ represent the actual locations of all the agents in $R[i]$.
2) Construct the Voronoi partition $\mathcal{V}(P_{k,\tau}(R[i])) = \{V_{k,\tau}^{j_1}, V_{k,\tau}^{j_2}, \ldots, V_{k,\tau}^{j_n}\}$ over the bin $R[i]$ corresponding to $P_{k,\tau}(R[i])$. For each agent $j \in \{j_1, j_2 \ldots, j_n\}$, its Voronoi set ($V_{k,\tau}^j$) is given by:
$$V_{k,\tau}^j = \Big\{ \boldsymbol{q} \in R[i] \,:\, \|\boldsymbol{q} - \boldsymbol{p}_{k,\tau}^j\|_2 \le \|\boldsymbol{q} - \boldsymbol{p}_{k,\tau}^\ell\|_2, \quad \forall \ell \in \{j_1, j_2 \ldots, j_n\} \text{ and } j \ne \ell \Big\} . \tag{10}$$
3) Let $\partial V_{k,\tau}^j$ denote the boundary of the Voronoi set $V_{k,\tau}^j$. Construct a modified Voronoi partition $\tilde{\mathcal{V}}(P_{k,\tau}(R[i])) = \{\tilde{V}_{k,\tau}^{j_1}, \tilde{V}_{k,\tau}^{j_2}, \ldots, \tilde{V}_{k,\tau}^{j_n}\}$ such that for all $j \in \{j_1, j_2 \ldots, j_n\}$:
$$\tilde{V}_{k,\tau}^j = \Big\{ \boldsymbol{q} \in V_{k,\tau}^j \,:\, \|\boldsymbol{q} - \partial V_{k,\tau}^j\|_2 \ge r_{\text{col}} \Big\} . \tag{11}$$
4) Compute the centroid $\mathcal{C}(\tilde{V}_{k,\tau}^j)$ of all the sets in the modified Voronoi partition $\tilde{\mathcal{V}}(P_{k,\tau}(R[i]))$, which is defined as:
$$\mathcal{C}(\tilde{V}_{k,\tau}^j) = \frac{\int_{\tilde{V}_{k,\tau}^j} \boldsymbol{q} d\boldsymbol{q}}{\int_{\tilde{V}_{k,\tau}^j} d\boldsymbol{q}} . \tag{12}$$
5) Depending on the type of the agent, set the new location of the agent as follows.
   a) If $\tilde{V}_{k,\tau}^j = \emptyset$, then we set $\boldsymbol{p}_{k,\tau+1}^j = \boldsymbol{p}_{k,\tau}^j$.

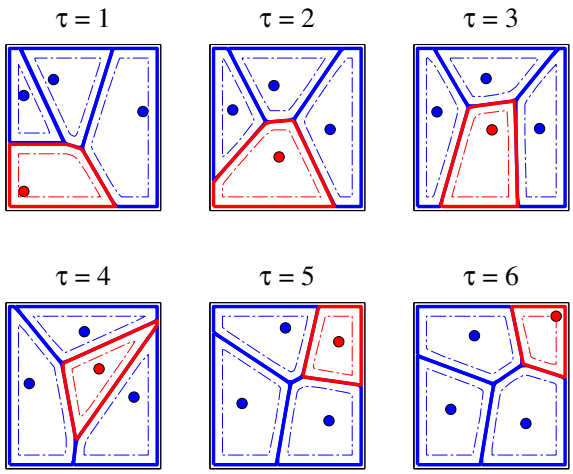


Figure 4. Time evolution of four agents in a bin, following the collision avoidance algorithm, where the transient agent (in red) goes from left-bottom to the right-top of the bin and the remaining stationary agents (in blue) give way to the transient agent. The Voronoi sets $(V^j_{k,\tau}, \tilde{V}^j_{k,\tau})$ of all the agents are also shown.

b) If the j^{th} agent is a stationary agent, then set the new location of the j^{th} agent to the centroid of its modified Voronoi set $\tilde{V}^j_{k,\tau}$, i.e., $\boldsymbol{p}^j_{k,\tau+1} = \mathcal{C}(\tilde{V}^j_{k,\tau})$.

c) If the j^{th} agent is a transient agent, then set the new location of the j^{th} agent as:

$$\boldsymbol{p}^j_{k,\tau+1} = \arg \min_{\boldsymbol{q} \in \tilde{V}^j_{k,\tau}} \|\boldsymbol{q} - \boldsymbol{\kappa}^j_k\|, \qquad (13)$$

where $\boldsymbol{\kappa}^j_k \in \mathcal{R}$ represents the centroid of its target bin.

6) If the transient agent is close to the boundary of the bin $R[i]$, then it also constructs Voronoi sets in the neighboring contiguous bins. This process allows it to cross the bin boundary and transition from bin $R[i]$ to a contiguous bin which is closer to its target bin. When the transient agent finally reaches its target bin, it becomes a stationary agent in that target bin.

These steps are illustrated in Fig. 4. The advantage of this novel distributed collision avoidance algorithm, using Voronoi partitions, is that it guarantees collision avoidance. Since each bin $R[i]$ is convex, the Voronoi set $(V^j_{k,\tau})$ and the modified set $(\tilde{V}^j_{k,\tau})$ are also convex. As each agent only moves within its Voronoi set, the agents are guaranteed to avoid collisions due to the buffer region of r_{col} along the boundary of each Voronoi set. Moreover, the process of constructing Voronoi sets is achieved in a distributed manner, hence we have a simple, efficient, distributed algorithm for collision avoidance using Voronoi partitions.

Although the process of generating Voronoi partitions is computationally expensive, it is less than the computational load for solving the discrete OTP using LP. Moreover, the communication load is the same as the consensus algorithm. Hence this algorithm can be executed using the existing computational and communication capabilities of agents performing PSG–OT.

VI. Numerical Simulation

In this section, we demonstrate distributed swarm guidance using PSG–OT and compare it with other existing probabilistic guidance techniques. Let us first introduce these Markov chain based probabilistic swarm guidance algorithms.

A. Markov Chain based Swarm Guidance Algorithms

In probabilistic swarm guidance algorithm using inhomogeneous Markov chains (**PSG–IMC**), each agent chooses the tuning parameter (ξ^j_k) based on the Hellinger distance (HD) between the estimated current swarm distribution $(\hat{\mathcal{F}}^j_{k,n_{\text{loop}}})$ and the desired formation $(\boldsymbol{\pi})$, using the following equation [1]:

$$\xi^j_k = D_H\left(\boldsymbol{\pi}, \hat{\mathcal{F}}^j_{k,n_{\text{loop}}}\right) := \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n_{\text{cell}}} \left(\sqrt{\boldsymbol{\pi}[i]} - \sqrt{\hat{\mathcal{F}}^j_{k,n_{\text{loop}}}[i]}\right)^2}. \qquad (14)$$

The HD is a symmetric measure of the difference between two probability distributions and it is upper bounded by 1 [36]. Each agent then transitions according to the following time-varying Markov matrix M^j_k, with $\boldsymbol{\pi}$ as its stationary distribution (i.e., $\boldsymbol{\pi} M^j_k = \boldsymbol{\pi}$) [1]:

$$M^j_k = \boldsymbol{\alpha}^j_k \frac{\xi^j_k}{\boldsymbol{\pi}\boldsymbol{\alpha}^j_k} \boldsymbol{\pi} \text{diag}(\boldsymbol{\alpha}^j_k) + \mathbf{I} - \xi^j_k \text{diag}(\boldsymbol{\alpha}^j_k), \qquad (15)$$

where $\boldsymbol{\pi}\boldsymbol{\alpha}^j_k \neq 0$ and $\sup_k \xi^j_k \|\boldsymbol{\alpha}^j_k\|_\infty \leq 1$. The probability that the j^{th} agent in bin $R[i]$ at the k^{th} time instant will transition to bin $R[\ell]$ at the $(k+1)^{\text{th}}$ time instant is given by $M^j_k[i, \ell]$. To ensure that the agents settle down after the desired formation is achieved, we see that $M^j_k \to \mathbf{I}$ if $\xi^j_k \to 0$ [1].

Meanwhile, probabilistic guidance algorithm (**PGA**) involves using a homogeneous Markov matrix M, with $\boldsymbol{\pi}$ as its stationary distribution, for all agents over all time steps [10]. We compare PSG–OT with PGA and PSG–IMC in the next section.

B. Simulation Results

We now present the setup of this simulation example. The swarm containing $m = 5000$ agents is guided to form the desired formation $\boldsymbol{\pi}$ associated with the letters UIUC as shown in Fig. 3(b). As shown in Fig. 6, the simulation starts at $k = 1$ time instant with the agents uniformly distributed across $\mathcal{R} \subset \mathbb{R}^2$, which is partitioned into 30×30 bins. During the consensus stage, each agent is allowed to communicate with those agents which are at most 10 steps away. Let $\boldsymbol{\kappa}[i] = (x[i], y[i])$ denote the location of the bin $R[i]$ in the 30×30 grid and the communication network topology P_k is setup using Metropolis weights. Moreover, each agent is allowed to transition to only those bins which are at most 5 steps away. An agent undergoes a transition if it jumps from bin $R[i]$ to bin $R[\ell]$, $\ell \neq i$ during a given time step. It is desired that the swarm guidance algorithm minimizes the cost of transitioning during each time step. The modified cost function is given by (9).

Monte Carlo simulations were performed to compare the performance of PSG–OT, illustrated in **Algorithm 1**, with

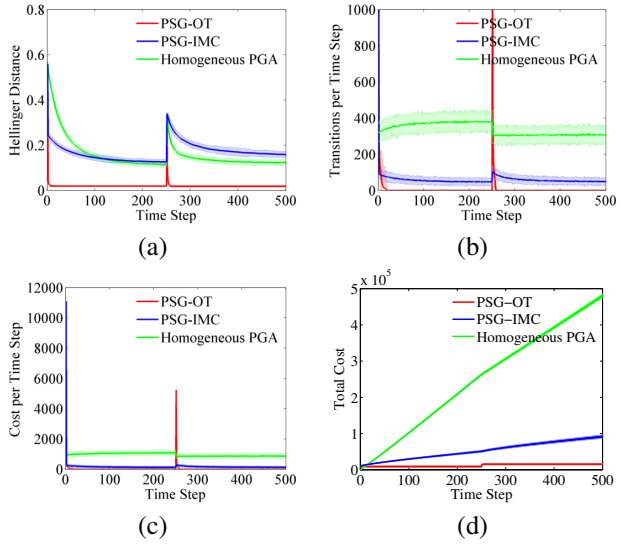


Figure 5. Monte Carlo simulations were performed to compare the performance of PSG–OT with PGA and PSG–IMC. The results from 50 simulation runs along with their $3-\sigma$ error-bars are presented for: (a) HD between the current swarm density distribution and the desired formation, (b) the number of transitions per time step, (c) the transition cost incurred per time step, and (d) the total cost incurred till the present time step. The spike or discontinuity after the 20^{th} time step is due to the external damage to the middle section of the swarm. Note that PSG–OT converges much faster and performs much better than PSG–IMC and PGA in all categories.

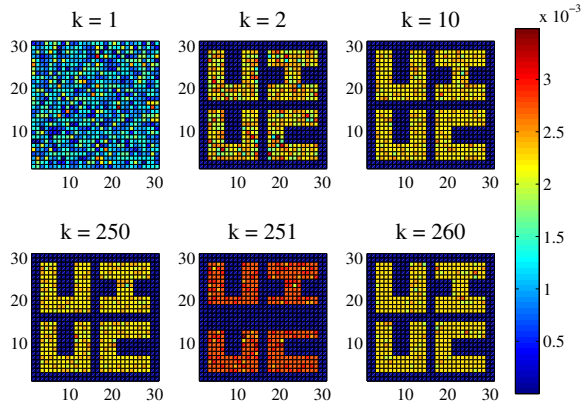


Figure 6. Histogram plots of the swarm distribution at different time instants for 5000 agents executing PSG–OT. The colorbar represents the pmf of the swarm distribution. Starting form an uniform distribution, the swarm converges to the desired formation within a couple of time steps. The middle section of the swarm is externally damaged and 1248 agents are removed after the 250^{th} time step, but the swarm autonomously recovers within another couple of time steps.

PGA and PSG–IMC. The cumulative results from 50 simulation runs are shown in Fig. 5. The histogram plots of the swarm distribution at different time instants, in a sample run of the Monte Carlo simulation where each agent executes PSG–OT, are shown in Fig. 6. The Hellinger distance (HD) metric between the current swarm distribution ($\mathcal{F}_k^\star$) and the desired formation ($\boldsymbol{\pi}$) (i.e. $D_H(\boldsymbol{\pi}, \mathcal{F}_k^\star)$ from (14)) is used to measure the convergence of the swarm. As shown in the HD plot in Fig. 5(a), the current swarm distribution rapidly converges to the desired formation within a couple of time step when the agents execute PSG–OT. On the other hand, the desired formation is almost achieved after approximately 200 time steps when the agents execute the PSG–IMC or PGA. Moreover, the HD between the swarm density distribution in steady state and the desired formation only reduces to ~ 0.12 for PSG–IMC and PGA cases as compared to ~ 0.02 in the case of PSG–OT. After the 250^{th} time step, the swarm is externally damaged by eliminating approximately 1000 ± 100 agents from the middle section of the formation. This is seen by comparing the images for the 250^{th} and 251^{st} time step in Fig. 6. Note that the swarm quickly recovers from this damage and the remaining agents attain the desired stationary distribution within another couple of time steps in the case of PSG–OT, while it takes approximately 200 time steps for PSG–IMC and PGA cases.

As shown in Fig. 5(b), the average number of transitions is per agent in 500 time steps is 1.34 ± 0.08 in the PSG–OT case compared to 6.2 ± 0.1 in the PSG–IMC case and 33.7 ± 0.1 in the case of PGA. Moreover, it is evident from Fig. 5(d) that the total cost incurred by PSG–OT ($(1.53 \pm 0.18) \times 10^4$) after 500 time steps is significantly lesser than that incurred by PSG–IMC ($(9.15 \pm 0.44) \times 10^4$) and PGA ($(4.80 \pm 0.06) \times 10^5$). It is also evident from the slope of the lines in Fig. 5(d) that the steady state cost of maintaining the formation is significantly higher for the PSG–IMC and PGA cases as compared to PSG–OT. Hence, we conclude that the performance of PSG–OT is the best in all these categories when compared to PSG–IMC and PGA.

Note that both PSG–OT and PSG–IMC use the consensus algorithm for estimating the current swarm distribution, but solving the discrete OTP using LP is significantly more computationally expensive than evaluating the Markov chain. Hence, solving the discrete OTP in a distributed manner using PSG–OT has the same communication load as the Markov chain based approaches, but has significantly more computational load. It is also evident from the cumulative results of the 50 simulation runs in Fig. 5 that PSG–OT works reliably well and achieves the desired objectives in all simulations runs.

During each time step, each agent executes the collision avoidance algorithm to reach its target bin. For visualization purposes a simpler problem, where 100 agents reach the desired formation of the letter "I", is shown in Fig. 7. Thus the collision avoidance algorithm helps the agents successfully implement PSG–OT.

VII. Conclusions

In this paper, we present a new approach to probabilistic guidance of swarms of autonomous agents using distributed optimal transport. In the proposed PSG–OT, each agent first estimates the current swarm distribution in a distributed manner using a consensus algorithm and then solves the OTP to obtain its guidance trajectory while avoiding collisions. Each agent thus independently determines its own trajectory so that the overall swarm quickly converges to the desired formation, and the algorithm is robust to external disturbances or damages. Compared to prior work on probabilistic swarm guidance using homogeneous and inhomogeneous Markov chains, the proposed algorithm guarantees faster convergence, reduces the number of transition to achieve the formation

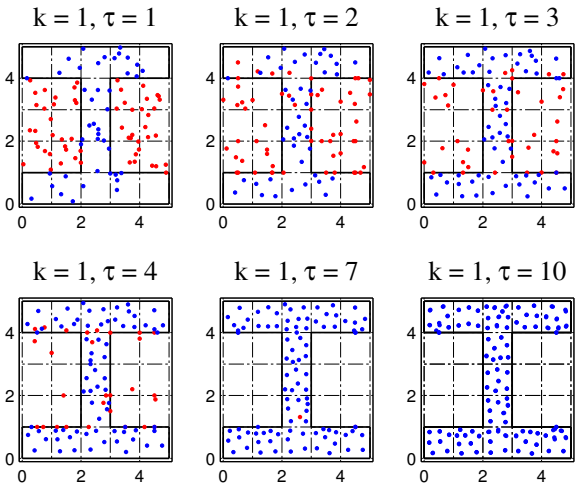


Figure 7. Time evolution of the actual location of 100 agents during multiple collision avoidance loops is shown. The agents reach the desired formation of the letter "I" within one PSG–OT time step. The transient agents are in red and the remaining stationary agents are in blue.

and minimizes a given cost function, while satisfying motion constraints. We also present a novel distributed collision avoidance algorithm for generating collision-free guidance trajectories for each agent. Results from multiple simulation runs demonstrate the properties of self-repair capability, reliability, convergence, collision avoidance, and cost effectiveness of PSG–OT. Hence PSG–OT is a complete collision-free solution for probabilistic swarm guidance problem. Future work will focus on finding appropriate cost functions that depend on the dynamics of the agents.

REFERENCES

[1] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using inhomogeneous Markov chains," *IEEE Trans. Control Network Syst.*, 2014. under review (http://arxiv.org/abs/1403.4134).

[2] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[3] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton University Press, 2010.

[4] J. S. Shamma, *Cooperative Control of Distributed Multi–Agent Systems*. Chichester, West Sussex, England: John Wiley & Sons, 2007.

[5] A. Cornejo, A. J. Lynch, E. Fudge, S. Bilstein, M. Khabbazian, and J. McLurkin, "Scale-free coordinates for multi-robot systems with bearing-only sensors," *Int. J. Robotics Research*, vol. 32, no. 12, pp. 1459–1474, 2013.

[6] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, "On development of 100-gram-class spacecraft for swarm applications," *IEEE Syst. J.*, 2014. to appear.

[7] A. R. Mesquita, J. P. Hespanha, and K. Åström, "Optimotaxis: A stochastic multi-agent on site optimization procedure," in *Proc. Hybrid Systems: Computation and Control*, 2008.

[8] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[9] I. Chattopadhyay and A. Ray, "Supervised self-organization of homogeneous swarms using ergodic projections of Markov chains," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1505–1515, 2009.

[10] B. Açikmeşe and D. S. Bayard, "A Markov chain approach to probabilistic swarm guidance," in *Amer. Control Conf.*, pp. 6300–6307, June 2012.

[11] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Inhomogeneous Markov chain approach to probabilistic swarm guidance algorithm," in *5th Int. Conf. Spacecraft Formation Flying Missions and Technologies*, (Munich, Germany), 2013.

[12] L. Kantorovitch, "On the translocation of masses," *Management Science*, vol. 5, no. 1, pp. 1–4, 1958.

[13] C. Villani, *Optimal transport: Old and New*. Springer Verlag, 2008.

[14] S. T. Rachev and L. Ruschendorf, *Mass Transportation Problems. Vol. I: Theorey*. Springer, 1998.

[15] Y. Zemel, "Optimal transportation: Continuous and discrete," Master's thesis, École polytechnique fédérale de Lausanne, 2012.

[16] S. Daneri and G. Savaré, "Lecture notes on gradient flows and optimal transport," in *Seminaires et Congres SMF*, pp. 1–27, 2010.

[17] G. Notarstefano and F. Bullo, "Network abstract linear programming with application to minimum-time formation control," in *Proc. 46th IEEE Conf. Decesion and Control*, (New Orleans, USA), pp. 927–932, Dec. 2007.

[18] M. Movafaghpour and E. Masehian, "A linear programming approach for probabilistic robot path planning with missing information of outcomes," in *IEEE Int. Conf. Automation Science and Engineering*, (Trieste, Italy), pp. 126–132, Aug 2011.

[19] L. Yang, J. Qi, and J. Han, "Path planning methods for mobile robots with linear programming," in *Proc. Int. Conf. Modelling, Identification and Control*, (Wuhan, China), June 2012.

[20] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, pp. 1–35, March 2014.

[21] D. Morgan, G. P. Subramanian, S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic guidance of distributed systems using sequential convex programming," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, (Chicago, US), Sept. 2014. accepted.

[22] D. Morgan, S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Guidance and control of swarms of spacecraft using optimal transport and model predictive control," in *AIAA Guidance Navigation and Control Conf.*, (Kissimee, FL), 2015. accepted.

[23] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robotics and Automation*, vol. 20, pp. 243–255, April 2004.

[24] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using Bayesian consensus filtering," in *Amer. Control Conf.*, (Portland, OR), pp. 634–641, June 2014.

[25] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803 – 812, 1986.

[26] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520 – 1533, 2004.

[27] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988 – 1001, 2003.

[28] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statistical Assoc.*, vol. 69, no. 345, pp. 688 – 704, 1960.

[29] C. Genest and J. V. Zidek, "Combining probability distributions: A critique and an annotated bibliography," *Statistical Sci.*, vol. 1, no. 1, pp. 114 – 135, 1986.

[30] M. Jackson, *Social and economic networks*. Princeton, NJ: Princeton University Press, 2008.

[31] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Trans. Autom. Control*, 2014. to appear.

[32] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, pp. 169–181, Mar. 2004.

[33] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65 – 78, 2004.

[34] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European J. Control*, vol. 18, no. 6, pp. 539–557, 2012.

[35] A. Rikos, T. Charalambous, and C. N. Hadjicostis, "Distributed weight balancing over digraphs," *IEEE Trans. Control Network Syst.*, 2014. to appear.

[36] E. Torgerson, *Comparison of Statistical Experiments*. Cambridge University Press, 1991.