

A Two-Phase Algorithm for Joint Task Assignment and Coordination in Hierarchical Multi-Vehicle Systems

Yoonjae Lee*, Ahmed Khalil†, and Efstathios Bakolas‡
The University of Texas at Austin, Austin, TX, 78712

Gregory M. Gremillion §
U.S. Army DEVCOM Army Research Laboratory, Adelphi, MD, 20783

This paper considers a variant of the classical location-allocation problem in which a fleet of mobile airships (clustered agents), each carrying multiple delivery drones (subagents), seeks to jointly determine the optimal placement of airships and allocation of drones to customers to minimize the total transportation cost. We propose a two-phase, gradient descent-based algorithm that iteratively updates airship locations and drone-customer allocation until convergence. Furthermore, to handle scenarios where airships are subject to communication constraints, we introduce a cluster-based distributed drone-customer allocation subroutine using the consensus alternating direction method of multipliers (C-ADMM). Numerical simulations are provided to demonstrate the effectiveness of the proposed algorithms.

I. Nomenclature

N	=	Number of airships
M_i	=	Number of drones in airship i
K	=	Number of customers
$\mathcal{A}$	=	Index set of airships, where $\mathcal{A} = \{1, \dots, N\}$
$\mathcal{D}_i$	=	Index set of drones in airship i , where $\mathcal{D}_i = \{1, \dots, M_i\}$
$\mathcal{T}$	=	Index set of customers, where $\mathcal{T} = \{1, \dots, K\}$
Π	=	Set of airship-drone pairs, where $\Pi = \{(i, j) : i \in \mathcal{A}, j \in \mathcal{D}_i\}$
$\mathcal{G}$	=	Drone-customer bipartite graph, where $\mathcal{G} = (\Pi, \mathcal{T}, \mathcal{E})$ with $\mathcal{E} = \Pi \times \mathcal{T}$
$\mathcal{M}$	=	Matching in $\mathcal{G}$
$\mathbb{M}$	=	Set of perfect matchings in $\mathcal{G}$
p_i	=	Position of airship i
θ_i	=	Heading angle of airship i
v_i	=	Linear velocity of airship i
ω_i	=	Angular velocity of airship i
$\mathbf{p}$	=	Joint airship position, where $\mathbf{p} = [p_1^\top, \dots, p_N^\top]^\top$
z_k	=	Location of customer k
$J_{\mathcal{M}}$	=	Transportation cost function
w_{ijk}	=	Square unit transportation cost of serving customer k for drone j in airship i
μ_i	=	Weighted centroid of cluster of customers assigned to drones in airship i
$\mathcal{C}$	=	Airship communication graph
$ \cdot $	=	Cardinality of a set
$\ \cdot\ $	=	Euclidean norm of a vector
$\mathbb{S}^1$	=	Unit circle

*PhD Candidate, Department of Aerospace Engineering and Engineering Mechanics, Email: yol033@utexas.edu

†Graduate Student, Department of Aerospace Engineering and Engineering Mechanics, Email: akhalil@utexas.edu

‡Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, Associate Fellow AIAA, Email: bakolas@austin.utexas.edu

§Researcher, Humans in Complex Systems Division, Email: gregory.m.gremillion.civ@army.mil

II. Introduction

Last-mile delivery plays a crucial role in the supply chain, representing the phase where goods move from a distribution hub to the consumer's doorstep. Two primary global trends, urbanization and e-commerce, significantly fuel the growing need for last-mile delivery services. Predictions suggest that by 2050, about 70% of the global population, or roughly 6.3 billion individuals, will reside in large urban areas with populations exceeding 10 million [1]. Additionally, in 2018, the global e-commerce sector experienced a growth rate of 23.3% [2]. This shift towards more people living in densely populated areas, and a rise in per capita online shopping is leading to a continuous surge in parcel delivery volumes. Despite its importance, last-mile delivery is often the least efficient and most expensive part of the shipping process due to the costs of navigating diverse routes to deliver individual orders. Conventional home delivery methods using delivery vans are notably expensive. For example, a simulation study utilizing real-world Finnish data revealed that traditional van delivery costs range from 2 to 6 Euros, influenced by the density of customers [3]. Significant factors contributing to these elevated costs, particularly in terms of personnel, include traffic congestion, the lack of parking in densely packed areas, and instances where customers are unavailable to receive their packages.

Alongside the rapid advancements in technology for autonomous systems, such as unmanned aerial vehicles (UAVs), a wealth of innovative last-mile delivery methods have gained attention in recent years. These approaches vary widely from currently utilized alternatives (e.g., cargo bikes) to tested prototypes (e.g., drone-based parcel delivery [4]), and even extend to futuristic concepts, including Amazon's patent for the flying warehouse delivery system called the airborne fulfillment centers [5]. These drone carrier airships would hover over urban centers, deploying drones to deliver goods directly from the sky. A significant benefit of this high-altitude launch is that it can not only save the expenditure of energy and fuel but also extend the operational range of drones. Efficient operation of this hierarchical UAV platform, however, requires substantial research efforts due to the complexity involved in coordinating different layers of vehicles.

An architecture for cooperative missions using multiple drones that accounts for information exchanges and physical couplings required for joint load transportation was developed in [6]. Another approach concurrently addresses a combinatorial task assignment problem while generating collision-free trajectories for a team of robots [7]. The authors of [8] propose a robotic rebalancing solution for a mobility-on-demand transportation system by utilizing a fluid model and linear programming to ensure vehicle availability across stations while minimizing trips. A unifying cost function for multi-robot deployment was proposed in [9], where it was shown that geometric, probabilistic, and potential field-based strategies can be interconnected through a single parameter. Potential game-based methods for decentralized control and task assignment in multi-agent systems were proposed in [10]. Recent work introduces an algorithmic framework to efficiently route a large fleet of drones for urban package delivery by leveraging public transit to extend drone range and minimize delivery times [11].

Central to many practical locational optimization problems, the classical facility location problem is concerned with determining optimal placement of facilities to serve a set of demand points [12]. When the facilities are mobile agents, this class of problems branch out into coverage control problems, in which the aim is to strategically position the agents based on their demand, with the importance of different subregions typically determined by a density function defined over the spatial domain [13]. As for this class of problems, the seminal work [14] introduced algorithmic frameworks for distributed control of multi-agent networks for area coverage, emphasizing Voronoi partitions and gradient descent methods to optimize (time-invariant) spatial distributions. Modified algorithms for spatially non-uniform and time-varying distributions and clusters of points have been proposed in [15–17]. A recent approach presents a novel framework for controlling the distribution of multi-agent networks across a specific area to match a target coverage density, using a two-level approach combining macroscopic and microscopic perspectives [18]. Other previous works concerning coverage control of multi-agent networks include [19–22], which by no means is a comprehensive list.

Inspired by the abovementioned Amazon's concept of airborne fulfillment centers, this paper addresses the problem of serving a set of customers with a group of mobile airships (clustered agents), where each airship is equipped with multiple delivery drones (subagents). This problem is formulated as a variant of the classical location-allocation problem [23], the objective of which is to minimize the total cost of serving customers by concurrently finding the optimal placement of airships and allocation of customers to the drones carried by each airship. A two-phase, gradient descent-based algorithm with guaranteed convergence is proposed to achieve this goal. In addition, a cluster-based distributed drone-customer allocation subroutine based on the consensus alternating direction method of multipliers (C-ADMM) [24] is introduced to handle the case where airships are subject to communication constraints. Lastly, numerical simulation results highlight the effectiveness of the proposed solution methods.

The remainder of the paper is structured as follows. In Section III, the problem formulation is provided. In Sections IV, we present our main algorithms and their proofs of convergence. Numerical simulation results are discussed in Section V. Lastly, Section VI concludes the paper with remarks and suggestions for future research directions.

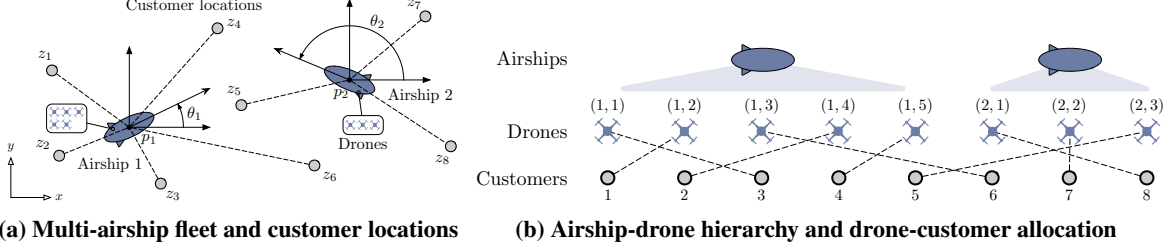


Fig. 1 Illustration example of the considered problem with 2 airships, 8 ($= 5 + 3$) drones, and 8 customers.

III. Problem Formulation

Consider a fleet of N airships (clustered agents) with index set $\mathcal{A} = \{1, \dots, N\}$. The kinematics of each airship is described as

$$\dot{x}_i = v_i \cos \theta_i, \quad (1a)$$

$$\dot{y}_i = v_i \sin \theta_i, \quad (1b)$$

$$\dot{\theta}_i = \omega_i, \quad (1c)$$

where $p_i = [x_i, y_i]^\top \in \mathbb{R}^2$, $\theta_i \in \mathbb{S}^1$, $v_i \in \mathbb{R}$, and $\omega_i \in \mathbb{R}$ denote the position, heading angle, linear velocity, and angular velocity of airship $i \in \mathcal{A}$, respectively. The control input of airship i is defined as $u_i = [v_i, \omega_i]^\top$. Consider also K customers with index set $\mathcal{T} = \{1, \dots, K\}$, where the position of customer $k \in \mathcal{T}$ is denoted as $z_k \in \mathbb{R}^2$. Each airship i carries $M_i \geq 2$ number of delivery drones (subagents) that can be deployed to deliver goods to customers. Let $\mathcal{D}_i = \{1, \dots, M_i\}$ denote the index set of drones in airship i . The total number of customers is assumed to be equal to that of drones, i.e., $K = \sum_{i=1}^N M_i$. Figure 1 illustrates the problem setting described above.

Let $\Pi = \{(i, j) : i \in \mathcal{A}, j \in \mathcal{D}_i\}$ be the set of airship-drone pairs, $\mathcal{E} = \Pi \times \mathcal{T}$ be the set of feasible drone-customer pairs, and $\mathcal{G} = (\Pi, \mathcal{T}, \mathcal{E})$ be the (complete and balanced) drone-customer bipartite graph. Additionally, let $w_{ijk} > 0$ represent the square unit transportation cost of serving customer k for drone j in airship i (which depends on, for instance, the customer's demand as well as the drone's specifications such as weight, payload capacity, and battery level). The objective is to jointly steer airships to a joint position $\mathbf{p} = [p_1^\top, \dots, p_N^\top]^\top \in \mathbb{R}^{2N}$ and determine a drone-customer matching $\mathcal{M} \in 2^{\mathcal{E}}$ such that the total cost of serving customers,

$$J_{\mathcal{M}}(\mathbf{p}) = \sum_{((i,j),k) \in \mathcal{M}} w_{ijk} \|p_i - z_k\|^2, \quad (2)$$

is minimized, subject to the kinematic constraint in (1) and one-to-one assignment constraint $\mathcal{M} \in \mathbb{M}$, where $\mathbb{M}$ denotes the set of perfect matchings (i.e., a matching that covers all vertices) in graph $\mathcal{G}$.

IV. Algorithms

The problem of jointly optimizing $\mathbf{p}$ and $\mathcal{M}$ to minimize the cost function $J_{\mathcal{M}}$ is computationally intractable in general. This is because, when $w_{ijk} = 1$ for all i, j , and k , the problem reduces to the capacitated location-allocation (or multi-source Weber) problem [25], which is known to be NP-hard even if all customers are located on a straight line [26]. Consequently, we focus on developing a heuristic method that can effectively find locally optimal solutions.

A. Matching-Based Centroidal Configurations

For any fixed matching $\mathcal{M} \in \mathbb{M}$, the gradient of the convex quadratic function $J_{\mathcal{M}}$ at $\mathbf{p} \in \mathbb{R}^{2N}$ can be calculated by

$$\nabla J_{\mathcal{M}}(\mathbf{p}) = 2 \begin{bmatrix} \sum_{(j,k) \in \mathcal{M}_1} w_{1jk} (p_1 - z_k) \\ \vdots \\ \sum_{(j,k) \in \mathcal{M}_N} w_{Njk} (p_N - z_k) \end{bmatrix} = 2 \begin{bmatrix} \left(\sum_{(j,k) \in \mathcal{M}_1} w_{1jk} \right) (p_1 - \mu_1) \\ \vdots \\ \left(\sum_{(j,k) \in \mathcal{M}_N} w_{Njk} \right) (p_N - \mu_N) \end{bmatrix}, \quad (3)$$

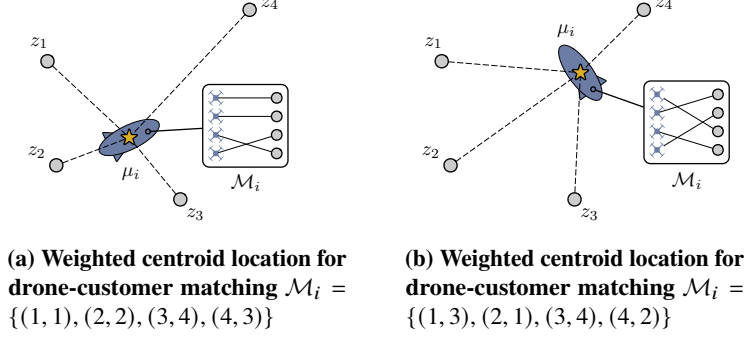


Fig. 2 Weighted centroid locations for one airship under two different drone-customer matchings.

where $\mathcal{M}_i = \{(j, k) \in \mathcal{D}_i \times \mathcal{T} : ((i, j), k) \in \mathcal{M}\}$ denotes the index set of drone-customer pairs associated with airship i , while μ_i represents the weighted centroid of the cluster of customers assigned to drones in airship i (see Figure 2 for illustrative examples):

$$\mu_i = \frac{\sum_{(j,k) \in \mathcal{M}_i} w_{ijk} z_k}{\sum_{(j,k) \in \mathcal{M}_i} w_{ijk}}. \quad (4)$$

Let $\boldsymbol{\mu} = [\mu_1^\top, \dots, \mu_N^\top]^\top$ denote the joint weighted centroid vector. Since all w_{ijk} are assumed to be positive, we have that $\nabla J_{\mathcal{M}}(\mathbf{p}) = \mathbf{0}$ if and only if $\mathbf{p} = \boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}$ is the (unique) global minimum of $J_{\mathcal{M}}$.

Now, consider the pointwise minimum function $J^*(\cdot) = \min_{\mathcal{M} \in \mathbb{M}} J_{\mathcal{M}}(\cdot)$. Although this function is generally nonsmooth and neither concave nor convex, any global minimum $\mathbf{p}^*$ of J^* and the corresponding optimal matching $\mathcal{M}^* \in \arg \min_{\mathcal{M} \in \mathbb{M}} J_{\mathcal{M}}(\mathbf{p}^*)$ minimize the original cost function together. In addition, for any $\mathcal{M} \in \mathbb{M}$, the global minimum of $J_{\mathcal{M}}$ is a local minimum of J^* if J^* is differentiable thereat. In the sequel, we say that the airships have reached a *matching-based centroidal configuration* if there exists a matching $\tilde{\mathcal{M}} \in \arg \min_{\mathcal{M} \in \mathbb{M}} J_{\mathcal{M}}(\mathbf{p})$ such that $\mathbf{p} = \tilde{\boldsymbol{\mu}}$, where $\tilde{\boldsymbol{\mu}}$ is the joint weighted centroid associated with $\tilde{\mathcal{M}}$.

B. Main Algorithm

The main solution method we propose is a variant of the well-known Lloyd's algorithm [27], which is commonly used for data clustering and multi-sensor coverage. Similar to Lloyd's algorithm, our algorithm consists of two alternating phases: drone-customer allocation and airship steering. These phases are repeated iteratively until convergence, at which point the locations of airships form a matching-based centroidal configuration. The pseudocode for the algorithm is presented in Algorithm 1, with detailed explanations provided below.

Allocation Phase (Line 3) Given the current joint airship location $\mathbf{p}(t)$, solve the following linear-sum assignment problem (LSAP) (using, for instance, the Hungarian algorithm [28]):

$$\underset{\{\mathbf{x}_{ij}\}}{\text{minimize}} \quad \sum_{(i,j) \in \Pi} c_{ij}(\mathbf{p}(t))^\top \mathbf{x}_{ij} \quad (5a)$$

$$\text{subject to} \quad \sum_{(i,j) \in \Pi} \mathbf{x}_{ij} = \mathbf{1}, \quad (5b)$$

$$\mathbf{1}^\top \mathbf{x}_{ij} = 1, \quad \forall (i, j) \in \Pi, \quad (5c)$$

$$\mathbf{x}_{ij} \in \{0, 1\}^K, \quad \forall (i, j) \in \Pi, \quad (5d)$$

where $\mathbf{x}_{ij}$ denotes the decision variables of drone j of airship i , i.e., $\mathbf{x}_{ij} = [x_{ij1}, \dots, x_{ijK}]^\top$, and $c_{ij}(\mathbf{p})$ is the vector of weighting coefficients given by $c_{ij}(\mathbf{p}) = [w_{ij1} \|p_i - z_1\|^2, \dots, w_{ijK} \|p_i - z_K\|^2]^\top$. Note that the constraints in (5b) and (5c) ensure that each drone serves exactly one customer and each customer is served by one drone, while the constraint in (5d) ensures that x_{ijk} is a binary variable which is equal to one if drone j in airship i is assigned to customer k and zero otherwise. Convert the obtained optimal solution $\{\bar{\mathbf{x}}_{ij}\}$ for problem (5) into the corresponding drone-customer matching $\tilde{\mathcal{M}} \in \mathbb{M}$ by using the relation $\bar{x}_{ijk} = 1 \iff ((i, j), k) \in \tilde{\mathcal{M}}$.

Algorithm 1: Joint Airship Coordination and Drone-Customer Allocation

```
1  $t \leftarrow 0, \bar{\mu} = [\bar{\mu}_1^\top, \dots, \bar{\mu}_N^\top]^\top \leftarrow [\infty, \dots, \infty]^\top$  // Initialization
2 while  $\|\bar{\mu} - \mathbf{p}(t)\| \geq \epsilon$  do
3    $\bar{\mathcal{M}} \leftarrow \text{Solve (5)}$  // Find an optimal drone-customer matching
4   for  $i \in \mathcal{A}$  do in parallel
5      $\bar{\mu}_i \leftarrow \text{Use (4)}$  // Compute weighted centroids
6      $\bar{u}_i = (\bar{v}_i, \bar{\omega}_i) \leftarrow \text{Use (6)}$  // Construct airship control laws
7      $p_i(t + \Delta t), \theta_i(t + \Delta t) \leftarrow \text{Update using } \bar{u}_i \text{ during time interval } [t, t + \Delta t)$ 
8    $t \leftarrow t + \Delta t$ 
```

Steering Phase (Lines 5-7) Compute the weighted centroids $\{\bar{\mu}_i\}$ associated with the matching $\bar{\mathcal{M}}$ obtained above using (4). During time interval $[t, t + \Delta t)$, where $\Delta t > 0$, let each airship i update its position and heading angle using the state-feedback control law $\bar{u}_i = (\bar{v}_i, \bar{\omega}_i) : \mathbb{R}^2 \times \mathbb{S}^1 \rightarrow \mathbb{R} \times \mathbb{R}$ given as follows [14, 29]:

$$\bar{v}_i(p_i, \theta_i) = -\kappa \left\langle \begin{bmatrix} \cos \theta_i & \sin \theta_i \end{bmatrix}^\top, p_i - \bar{\mu}_i \right\rangle, \quad (6a)$$

$$\bar{\omega}_i(p_i, \theta_i) = 2\kappa \arctan \left(\frac{\left\langle \begin{bmatrix} -\sin \theta_i & \cos \theta_i \end{bmatrix}^\top, p_i - \bar{\mu}_i \right\rangle}{\left\langle \begin{bmatrix} \cos \theta_i & \sin \theta_i \end{bmatrix}^\top, p_i - \bar{\mu}_i \right\rangle} \right), \quad (6b)$$

where $\kappa > 0$ is a controller gain.

Theorem 1 (Convergence of Algorithm 1). Suppose that for every drone-customer matching $\mathcal{M} \in \mathbb{M}$, the cost function J^* is differentiable at the associated joint weighted centroid μ . Then, for any given initial joint airship position $\mathbf{p}_0 \in \mathbb{R}^{2N}$, $\mathbf{p}(t)$ converges to a local minimum of J^* as $t \rightarrow \infty$.

Proof. See Appendix. □

C. Cluster-Based Distributed Drone-Customer Allocation Subroutine

In practice, a multi-airship fleet may encounter communication delays or failures due to adverse weather conditions. To model these scenarios, we assume that each airship can only communicate with its neighboring airships, as determined by the topology of a time-varying undirected graph $\mathcal{C}(t) = (\mathcal{A}, \mathcal{L}(t))$. Here, $\mathcal{L}(t)$ denotes the set of communication links, comprising unordered pairs of airships that can exchange information at time t .

Assumption 1 (Communication Graph). For all $t \geq 0$, the airship communication graph $\mathcal{C}(t)$ is connected.

To address this scenario, we propose a decentralized solution method for the LSAP in (5) (see Algorithm 2) based on the consensus alternating direction method of multipliers (C-ADMM) [24, 30]. C-ADMM is a distributed optimization technique well-suited for certain classes of separable optimization problems, including LSAPs [31, 32]. For LSAPs specifically, C-ADMM-based algorithms can achieve optimal solutions, in contrast to popular auction-based heuristics [33], and even outperform the distributed Hungarian-based methods [34] in terms of computation time [32]. Unlike the general C-ADMM approaches in the references cited above, however, Algorithm 2 is slightly modified to address the unique characteristics of the considered problem in which the decision makers are clustered agents.

Algorithm 2 is derived as follows. Observe first that problem (9) is in a form that describes the drone-customer allocation problem at the level of drones, not airships. To restructure the problem, we define the decision variable X_i of each airship $i \in \mathcal{A}$ as the aggregated decision variables of all its subagents:

$$X_i = \begin{bmatrix} \mathbf{x}_{i1} & \dots & \mathbf{x}_{iM_i} \end{bmatrix}. \quad (7)$$

Algorithm 2: Cluster-Based C-ADMM Algorithm

```

1  $\mathbf{X}_0^i \in \mathbb{R}^{K \times K}$  (for all  $i \in \mathcal{A}$ ),  $\Lambda_0^i \leftarrow \mathbf{0}$  (for all  $i \in \mathcal{A}$ ),  $k \leftarrow 0$  // Initialization
2 while not converged do
3   for  $i \in \mathcal{A}$  do in parallel
4     Send  $\mathbf{X}_k^i$  to neighbors in  $\mathcal{N}_i(t)$  // Transmit information
5      $\Lambda_{k+1}^i \leftarrow$  Use (11) // Dual update
6      $\mathbf{X}_{k+1}^i \leftarrow$  Use (12) // Primal update
7    $k \leftarrow k + 1$ 

```

Subsequently, we rewrite the problem in the form

$$\underset{X_1, \dots, X_N}{\text{minimize}} \quad \sum_{i \in \mathcal{A}} \text{tr}(C_i(\mathbf{p}(t))X_i) \quad (8a)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{A}} X_i \mathbf{1} = \mathbf{1}, \quad (8b)$$

$$X_i^\top \mathbf{1} = \mathbf{1}, \quad \forall i \in \mathcal{A}, \quad (8c)$$

$$X_i \in \{0, 1\}^{K \times M_i}, \quad \forall i \in \mathcal{A}, \quad (8d)$$

where $C_i(\mathbf{p})$ denotes the vertical concatenation of row vectors $c_{ij}(\mathbf{p})^\top$ for all $j \in \mathcal{D}_i$. This problem can be relaxed into a linear program by replacing the integer constraints in (8d) with box constraints $X_i \in [0, 1]^{K \times M_i}$:

$$\underset{X_1, \dots, X_N}{\text{minimize}} \quad \sum_{i \in \mathcal{A}} \text{tr}(C_i(\mathbf{p}(t))X_i) \quad (9a)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{A}} X_i \mathbf{1} = \mathbf{1}, \quad (9b)$$

$$X_i^\top \mathbf{1} = \mathbf{1}, \quad \forall i \in \mathcal{A}, \quad (9c)$$

$$X_i \in [0, 1]^{K \times M_i}, \quad \forall i \in \mathcal{A}, \quad (9d)$$

Given that the original problem (8) is feasible and has a linear objective, any optimal solution to the relaxed problem (9) is attained at a vertex of the feasible set and corresponds also to an optimal solution of problem (5). Now, letting each airship i maintain a local copy $\mathbf{X}^i = [X_1^i, \dots, X_N^i]$ of $\mathbf{X} = [X_1, \dots, X_N]$, the distributed form of the above problem can be described as

$$\underset{\mathbf{X}^1, \dots, \mathbf{X}^N}{\text{minimize}} \quad \sum_{i \in \mathcal{A}} \text{tr}(C_i(\mathbf{p}(t))X_i^i) \quad (10a)$$

$$\text{subject to} \quad \sum_{l \in \mathcal{A}} X_l^i \mathbf{1} = \mathbf{1}, \quad \forall i \in \mathcal{A}, \quad (10b)$$

$$X_i^{i^\top} \mathbf{1} = \mathbf{1}, \quad \forall i \in \mathcal{A}, \quad (10c)$$

$$X_l^i \in [0, 1]^{K \times M_i}, \quad \forall i \in \mathcal{A}, \quad (10d)$$

$$\mathbf{X}^i = \mathbf{X}^l, \quad \forall i \in \mathcal{A}, \forall l \in \mathcal{N}_i(t), \quad (10e)$$

where $\mathcal{N}_i(t)$ denotes the set of neighbors of airship i at time t . Under Assumption 1, problem (10) can be shown to be equivalent to problem (8) (see, e.g., [30, 32, 35]). Lastly, under the assumption that $\Lambda_0^i = \mathbf{0}$ for all $i \in \mathcal{A}$, C-ADMM leads to the update rules below (details can be found in [31, 32]).

Dual Update Procedure (Line 5) Each airship i updates the matrix Λ^i according to

$$\Lambda_{k+1}^i = \Lambda_k^i + \rho \sum_{l \in \mathcal{N}_i(t)} (\mathbf{X}_k^i - \mathbf{X}_k^l), \quad (11)$$

where $\rho > 0$ is a penalty parameter.

Primal Update Procedure (Line 6) Each airship i updates its local variable $\mathbf{X}^i$ by solving the following (convex) quadratic program:

$$\underset{\mathbf{X}^i}{\text{minimize}} \quad \text{tr} (C_i(\mathbf{p}(t))X_i^i) + \mathbf{1}^\top (\Lambda_k^i \circ \mathbf{X}^i) \mathbf{1} + \rho \sum_{l \in \mathcal{N}_i(t)} \left\| \mathbf{X}^i - \frac{\mathbf{X}_k^i + \mathbf{X}_k^l}{2} \right\|_F^2 \quad (12a)$$

$$\text{subject to} \quad \sum_{l \in \mathcal{A}} X_l^i \mathbf{1} = \mathbf{1}, \quad (12b)$$

$$X_i^{i\top} \mathbf{1} = \mathbf{1}, \quad (12c)$$

$$X_l^i \in [0, 1]^{K \times M_i}, \quad \forall l \in \mathcal{A}, \quad (12d)$$

where $\circ$ denotes the Hadamard product (i.e., element-wise matrix multiplication) and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

Similar to Algorithm 1, Algorithm 2 alternates between the two procedures described above. As shown in [32, Theorem 1], the sequence of local variables $\{\mathbf{X}_k^i\}_{k=0,1,\dots}$ (for all $i \in \mathcal{A}$) converges to an optimal solution of problem (8), provided that Assumption 1 holds. Therefore, when Algorithm 2 is used as a subroutine in line 3 of Algorithm 1, the overall algorithm retains the same convergence guarantees as in the case of no communication constraints.

V. Numerical Simulation

This section presents numerical simulation results that illustrate the performance of the proposed algorithms. We consider a fleet of five airships ($N = 5$), each carrying four drones ($M_i = 4, \forall i \in \mathcal{A}$). The initial airship locations are selected uniformly at random from the set $\{x \in \mathbb{R}^2 : \|x\| \leq 1\}$. There are total 20 customers ($K = 20$) divided into three groups, each containing 6, 7, and 7 customers, respectively. The locations of customers in each group are randomly sampled from normal distributions centered at $[2, 2]^\top$, $[-2, 2]^\top$, and $[0, -2]^\top$. See Figure 3 for the sampled initial airship and customer locations. The following two scenarios are considered:

- 1) **Scenario 1:** The airship communication graph $\mathcal{C}(t)$ is complete for all $t \geq 0$, and every drone has a unit transportation capability: $w_{ijk} = 1$ for all i, j , and k .
- 2) **Scenario 2:** The airship communication graph $\mathcal{C}(t)$ is determined by the Delaunay triangulation [36] of the airship locations $\{p_i(t)\}$, and each drone has varying transportation capabilities, with the values of w_{ijk} drawn uniformly at random from the interval $[1, 5]$.

The remaining parameters are as follows: $\kappa = 0.01$ (controller gain), $\epsilon = 1 \times 10^{-3}$ (convergence threshold for Algorithm 1), $\Delta t = 10$ (sampling time), and $\rho = 0.05$ (penalty parameter for Algorithm 2).

Scenario 1 In this scenario, the Hungarian algorithm [28] is used as a centralized subroutine in line 3 of Algorithm 1 to solve the LSAP in (5). Simulation snapshots at three different time instants are shown in Figures 4a, 4b, and 4c, with black lines indicating the drone-customer allocations, gray dashed lines indicating communication links, and colored

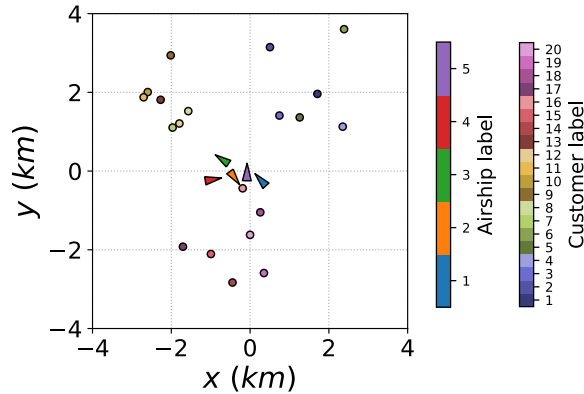


Fig. 3 Initial airship and customer locations, where triangles represent airships and circles represent customers.

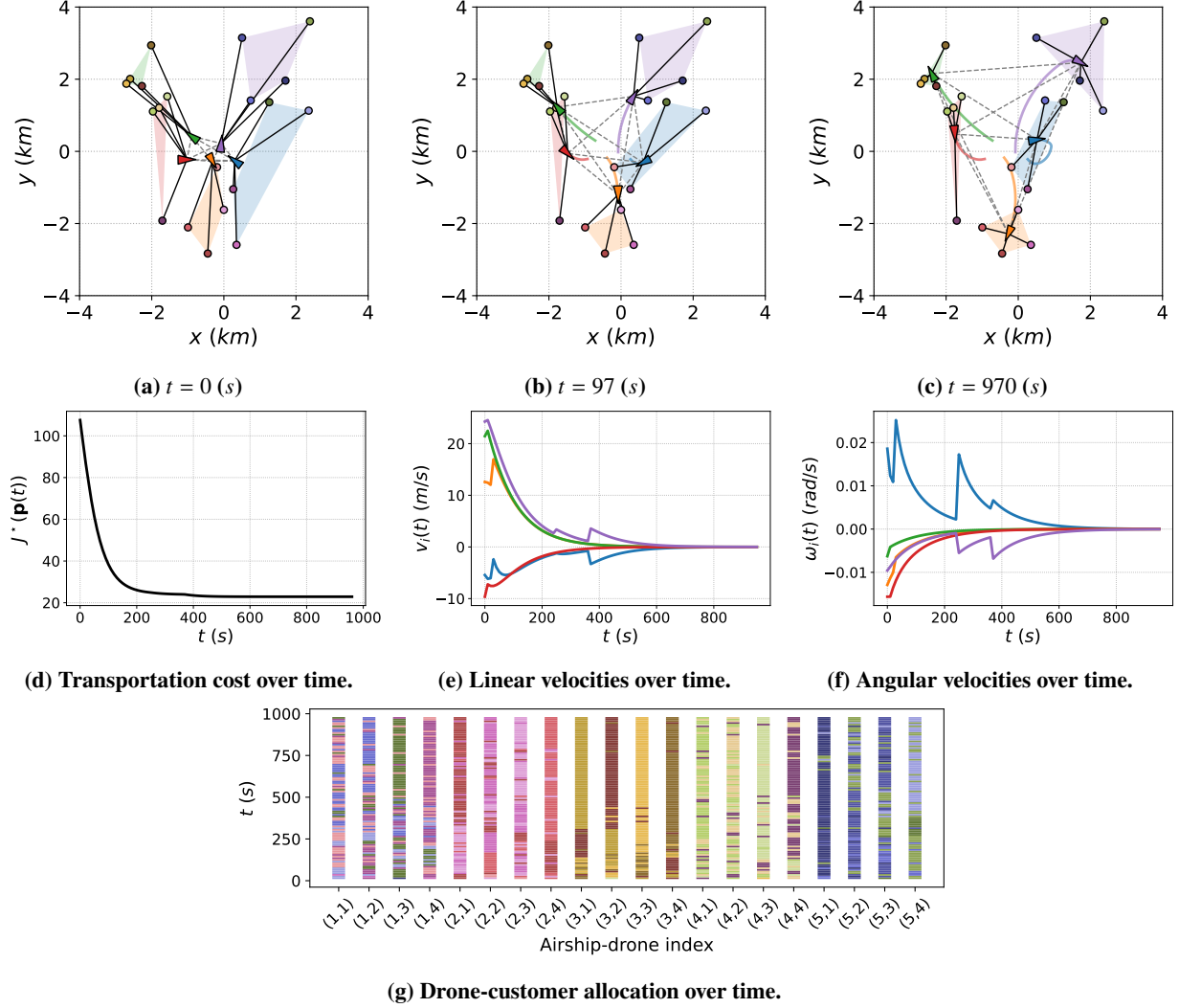


Fig. 4 Simulation snapshots and time evolutions of total transportation cost, control inputs, and drone-customer allocations in Scenario 1.

polygons indicating the convex hulls of customers allocated to the subagents of airship i . It can be immediately observed that airships tend to allocate to their drones the customers that are closest to their current locations; this is evidenced by the convex hulls of customers for different airships never intersecting at any time. Figure 4d illustrates the time evolution of the transportation cost, which clearly shows that the total transportation cost monotonically decreases over time. Figures 4e and 4f depict the evolution of the control inputs of each airship over time, all converging to zero. Figure 4g shows the change of drone-customer allocations over time; one notable observation here is that while airships successfully converge to a matching-based centroidal configuration, drone-customer allocations continue to chatter, failing to reach a consensus. The reason this occurs is that due to the uniform transportation capabilities of drones, the LSAP in (5) admits multiple solutions at the centroidal configuration to which airships converge. The chattering, however, is confined to allocations within each airship and does not prevent airships from converging to a centroidal configuration.

Scenario 2 In this scenario, Algorithm 2 is used as a distributed subroutine in line 3 of Algorithm 1. To solve the convex quadratic program in the primal update procedure given in (12), OSQP [37] is used (see Appendix for details). Similar to Figure 4, Figure 5 presents key results from the simulation of Scenario 2, including simulation snapshots and time evolutions of total transportation cost, control inputs, and drone-customer allocations. For brevity, we remark only

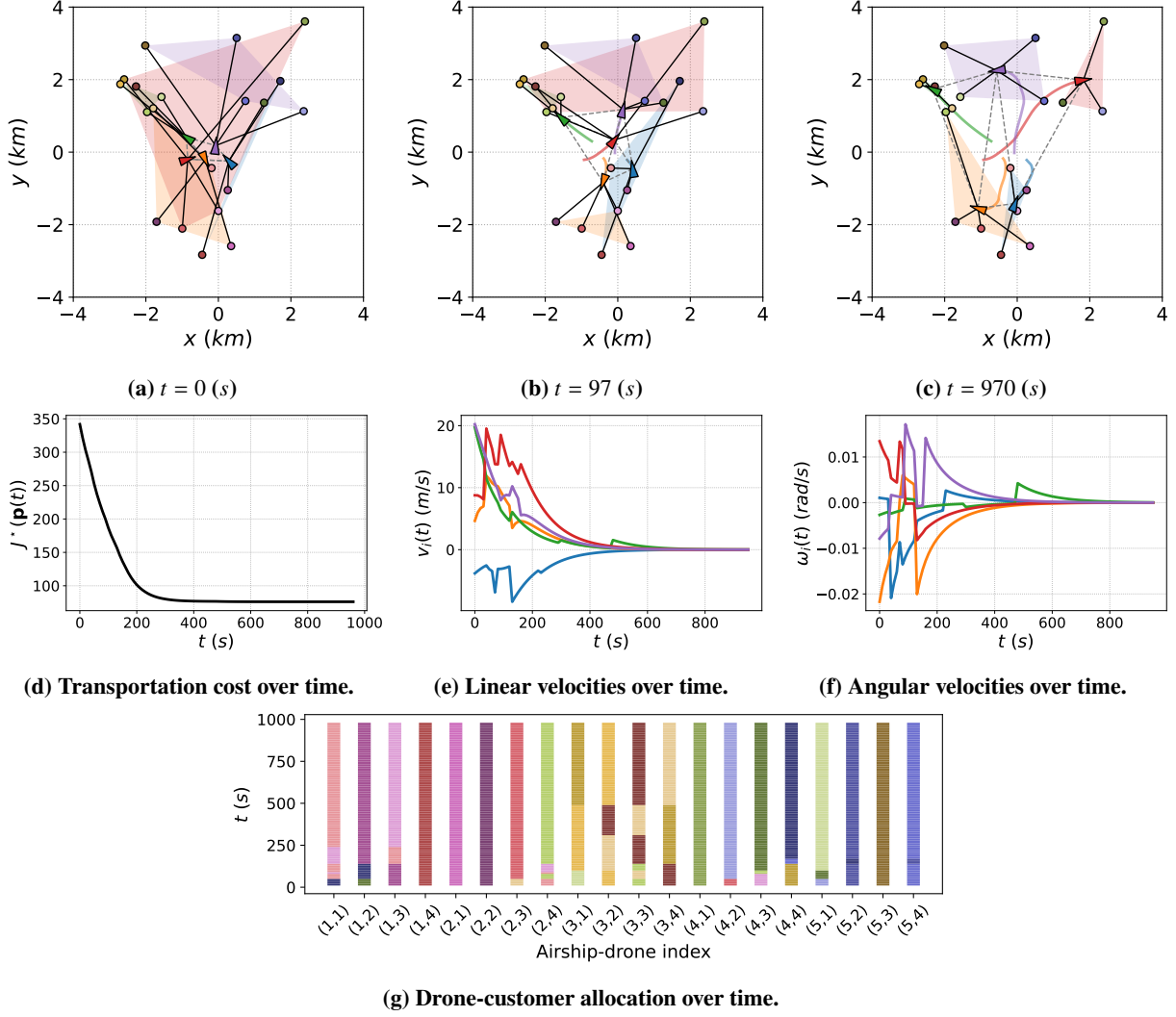


Fig. 5 Simulation snapshots and time evolutions of total transportation cost, control inputs, and drone-customer allocations in Scenario 2.

on notable differences from Scenario 1. Unlike the previous scenario, the convex hulls of customers often intersect (see airships 1 and 2 in Figure 5c), as the drone-customer allocations are heavily influenced by the values of w_{ijk} . Moreover, the allocations successfully reach a consensus as airships approach their centroidal configuration.

VI. Conclusion

This work addressed the problem of simultaneously locating airships (clustered agents) and allocating their delivery drones (subagents) to serve a set of customers. A two-phase, gradient descent-based algorithm was proposed to achieve efficient airship placement and drone-customer allocation. Additionally, a decentralized drone-customer allocation subroutine based on the consensus alternating direction method of multipliers was introduced to deal with communication constraints. Simulation results demonstrated the performance of the proposed algorithms. Future research directions include, for instance, modifying Algorithm 1 to accommodate nonlinear resource allocation models.

Appendix

Proof of Theorem 1 Let $F = (F_1, \dots, F_N) : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$ be an algorithm that maps a given joint airship position $\mathbf{p}$ to its future location after performing one iteration of the two phases of Algorithm 1 (i.e., after time Δt has elapsed). Let $\tilde{\mathcal{M}} \in \arg \min_{\mathcal{M} \in \mathbb{M}} J_{\mathcal{M}}(\mathbf{p})$ be the matching obtained by solving the LSAP in (5) and $\bar{\boldsymbol{\mu}}$ be the associated joint weighted centroid given by (4). By construction of the control law in (6), the following two statements are true:

- (i) For all $i \in \mathcal{A}$, $\|F_i(\mathbf{p}) - \bar{\boldsymbol{\mu}}_i\| \leq \|p_i - \bar{\boldsymbol{\mu}}_i\|$.
- (ii) If $\mathbf{p} \neq \bar{\boldsymbol{\mu}}$, then there exists an $i \in \mathcal{A}$ such that $\|F_i(\mathbf{p}) - \bar{\boldsymbol{\mu}}_i\| < \|p_i - \bar{\boldsymbol{\mu}}_i\|$.

The proof can be found in [29] (see also [14]). These properties imply that the set of fixed points of F (i.e., $\mathbf{p}$ such that $\mathbf{p} = F(\mathbf{p})$) coincides with the set of matching-based centroidal configurations.

By the global convergence theorem (see [14] or [38]), F converges to the set of its fixed points if J^* is a descent function for F , i.e., $J^*(F(\mathbf{p})) \leq J^*(\mathbf{p})$, where the inequality is strict if $\mathbf{p} \neq \bar{\boldsymbol{\mu}}$. To prove this condition, note that

$$J^*(F(\mathbf{p})) - J^*(\mathbf{p}) = \min_{\mathcal{M} \in \mathbb{M}} \{J_{\mathcal{M}}(F(\mathbf{p}))\} - J_{\tilde{\mathcal{M}}}(\mathbf{p}) \leq J_{\tilde{\mathcal{M}}}(F(\mathbf{p})) - J_{\tilde{\mathcal{M}}}(\mathbf{p}).$$

Using (4) yields

$$\begin{aligned} J^*(F(\mathbf{p})) - J^*(\mathbf{p}) &\leq \sum_{i \in \mathcal{A}} \sum_{(j,k) \in \tilde{\mathcal{M}}_i} w_{ijk} \left(\|F_i(\mathbf{p}) - z_k\|^2 - \|p_i - z_k\|^2 \right) \\ &= \sum_{i \in \mathcal{A}} \left(\alpha_i \left(\|F_i(\mathbf{p})\|^2 - \|p_i\|^2 \right) - 2 (F_i(\mathbf{p}) - p_i)^\top \underbrace{\left(\sum_{(j,k) \in \tilde{\mathcal{M}}_i} w_{ijk} z_k \right)}_{= \alpha_i \bar{\boldsymbol{\mu}}_i} \right) \\ &= \sum_{i \in \mathcal{A}} \alpha_i \left(\|F_i(\mathbf{p}) - \bar{\boldsymbol{\mu}}_i\|^2 - \|p_i - \bar{\boldsymbol{\mu}}_i\|^2 \right), \end{aligned}$$

where $\alpha_i = \sum_{(j,k) \in \tilde{\mathcal{M}}_i} w_{ijk} > 0$. In view of conditions (i) and (ii) above, the desired result immediately follows.

Finally, the set of matching-based centroidal configurations is nonempty and finite, as $|\mathbb{M}| = K!$. Given all the conditions above, it follows from [14, Proposition 1.4] that F converges to one of its fixed points, namely a matching-based centroidal configuration. As mentioned in Section IV.A, any matching-based centroidal configuration is a local minimum of J^* , provided that this function is differentiable thereat. This completes the proof.

Quadratic Program for OSQP To formulate problem (12) into the form required by OSQP,

$$\underset{x}{\text{minimize}} \quad \frac{1}{2} x^\top P x + q^\top x \tag{13}$$

$$\text{subject to} \quad l \leq A x \leq u, \tag{14}$$

we define the following variables:

$$P = 2\rho |\mathcal{N}_i(t)| I, \quad q = \text{vec} \left[C_i(\mathbf{p}(t)) + \Lambda_k^i - \rho \sum_{l \in \mathcal{N}_i(t)} (\mathbf{X}_k^i + \mathbf{X}_k^l) \right], \quad A = \begin{bmatrix} \hat{P} \\ \hat{Q}_i \\ I \end{bmatrix}, \quad l = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \quad u = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \tag{15}$$

where $\hat{P} \in \mathbb{R}^{K \times K^2}$ is a horizontal block matrix with K identity matrices concatenated horizontally, and $\hat{Q}_i \in \mathbb{R}^{M_i \times K^2}$ represents a block matrix of all zeros where the j -th horizontal block component of the k -th row of $\hat{Q}_i$ is set to $\mathbf{1}$ for all j and k .

Acknowledgments

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-22-2-0091. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies; either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

- [1] Bretzke, W.-R., “Global urbanization: a major challenge for logistics,” *Logistics Research*, Vol. 6, No. 2, 2013, pp. 57–62. <https://doi.org/10.1007/s12159-013-0101-9>, URL <https://doi.org/10.1007/s12159-013-0101-9>.
- [2] Boysen, N., Fedtke, S., and Schwerdfeger, S., “Last-mile delivery concepts: a survey from an operational research perspective,” *OR Spectrum*, Vol. 43, No. 1, 2021, pp. 1–58. <https://doi.org/10.1007/s00291-020-00607-8>, URL <https://doi.org/10.1007/s00291-020-00607-8>.
- [3] Punakivi, M., Yrjölä, H., and Holmström, J., “Solving the last mile issue: reception box or delivery box?” *International Journal of Physical Distribution & Logistics Management*, Vol. 31, No. 6, 2001, pp. 427–439. <https://doi.org/10.1108/09600030110399423>, URL <https://doi.org/10.1108/09600030110399423>.
- [4] Otto, A., Agatz, N., Campbell, J., Golden, B., and Pesch, E., “Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey,” *Networks*, Vol. 72, No. 4, 2018, pp. 411–458. <https://doi.org/https://doi.org/10.1002/net.21818>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21818>.
- [5] Berg, P. W., Isaacs, S., and Blodgett, K. L., “Airborne fulfillment center utilizing unmanned aerial vehicles for item delivery,” , Apr. 5 2016. US Patent 9305280.
- [6] Maza, I., Kondak, K., Bernard, M., and Ollero, A., *Multi-UAV Cooperation and Control for Load Transportation and Deployment*, Springer Netherlands, Dordrecht, 2010, pp. 417–449. https://doi.org/10.1007/978-90-481-8764-5_22, URL https://doi.org/10.1007/978-90-481-8764-5_22.
- [7] Turpin, M., Michael, N., and Kumar, V., “Capt: Concurrent assignment and planning of trajectories for multiple robots,” *The International Journal of Robotics Research*, Vol. 33, No. 1, 2014, pp. 98–112. <https://doi.org/10.1177/0278364913515307>, URL <https://doi.org/10.1177/0278364913515307>.
- [8] Pavone, M., Smith, S. L., Frazzoli, E., and Rus, D., “Robotic load balancing for mobility-on-demand systems,” *The International Journal of Robotics Research*, Vol. 31, No. 7, 2012, pp. 839–854. <https://doi.org/10.1177/0278364912444766>, URL <https://doi.org/10.1177/0278364912444766>.
- [9] Schwager, M., Rus, D., and Slotine, J.-J., “Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment,” *The International Journal of Robotics Research*, Vol. 30, No. 3, 2011, pp. 371–383. <https://doi.org/10.1177/0278364910383444>, URL <https://doi.org/10.1177/0278364910383444>.
- [10] Bakolas, E., and Lee, Y., “Decentralized Game-Theoretic Control for Dynamic Task Allocation Problems for Multi-Agent Systems,” *2021 American Control Conference (ACC)*, 2021, pp. 3228–3233. <https://doi.org/10.23919/ACC50511.2021.9483030>.
- [11] Choudhury, S., Solovey, K., Kochenderfer, M. J., and Pavone, M., “Efficient Large-Scale Multi-Drone Delivery using Transit Networks,” *Journal of Artificial Intelligence Research*, Vol. 70, 2021, p. 757–788. <https://doi.org/10.1613/jair.1.12450>, URL <https://doi.org/10.1613/jair.1.12450>.
- [12] Farahani, R. Z., and Hekmatfar, M. (eds.), *Facility Location: Concepts, Models, Algorithms and Case Studies*, Contributions to Management Science, Physica Heidelberg, 2009. <https://doi.org/10.1007/978-3-7908-2151-2>.
- [13] Bullo, F., Cortés, J., and Martínez, S., *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*, Princeton Series in Applied Mathematics, Princeton University Press, 2009.
- [14] Cortes, J., Martinez, S., Karatas, T., and Bullo, F., “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 2, 2004, pp. 243–255. <https://doi.org/10.1109/TRA.2004.824698>.
- [15] Lee, S. G., Diaz-Mercado, Y., and Egerstedt, M., “Multirobot Control Using Time-Varying Density Functions,” *IEEE Transactions on Robotics*, Vol. 31, No. 2, 2015, pp. 489–493. <https://doi.org/10.1109/TRO.2015.2397771>.
- [16] Miah, S., Panah, A. Y., Fallah, M. M. H., and Spinello, D., “Generalized non-autonomous metric optimization for area coverage problems with mobile autonomous agents,” *Automatica*, Vol. 80, 2017, pp. 295–299. <https://doi.org/https://doi.org/10.1016/j.automatica.2017.02.044>, URL <https://www.sciencedirect.com/science/article/pii/S0005109817301188>.
- [17] Yu, D., Xu, H., Chen, C. L. P., Bai, W., and Wang, Z., “Dynamic Coverage Control Based on K-Means,” *IEEE Transactions on Industrial Electronics*, Vol. 69, No. 5, 2022, pp. 5333–5341. <https://doi.org/10.1109/TIE.2021.3080205>.
- [18] Abdulghafoor, A. Z., and Bakolas, E., “Two-Level Control of Multiagent Networks for Dynamic Coverage Problems,” *IEEE Transactions on Cybernetics*, Vol. 53, No. 7, 2023, pp. 4067–4078. <https://doi.org/10.1109/TCYB.2021.3131292>.

- [19] Schwager, M., McLurkin, J., and Rus, D., “Distributed Coverage Control with Sensory Feedback for Networked Robots.” *Robotics: Science and Systems*, 2006, pp. 49–56.
- [20] Bakolas, E., “Distributed Partitioning Algorithms for Locational Optimization of Multiagent Networks in SE(2),” *IEEE Transactions on Automatic Control*, Vol. 63, No. 1, 2018, pp. 101–116. <https://doi.org/10.1109/TAC.2017.2707602>.
- [21] Santos, M., Diaz-Mercado, Y., and Egerstedt, M., “Coverage Control for Multirobot Teams With Heterogeneous Sensing Capabilities,” *IEEE Robotics and Automation Letters*, Vol. 3, No. 2, 2018, pp. 919–925. <https://doi.org/10.1109/LRA.2018.2792698>.
- [22] Arslan, Ö., “Statistical Coverage Control of Mobile Sensor Networks,” *IEEE Transactions on Robotics*, Vol. 35, No. 4, 2019, pp. 889–908. <https://doi.org/10.1109/TRO.2019.2906474>.
- [23] Cooper, L., “Location-Allocation Problems,” *Operations Research*, Vol. 11, No. 3, 1963, pp. 331–343. <https://doi.org/10.1287/opre.11.3.331>, URL <https://doi.org/10.1287/opre.11.3.331>.
- [24] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends® in Machine Learning*, Vol. 3, No. 1, 2011, pp. 1–122. <https://doi.org/10.1561/22000000016>, URL <http://dx.doi.org/10.1561/22000000016>.
- [25] Cooper, L., “The Transportation-Location Problem,” *Operations Research*, Vol. 20, No. 1, 1972, pp. 94–108. <https://doi.org/10.1287/opre.20.1.94>, URL <https://doi.org/10.1287/opre.20.1.94>.
- [26] Sherali, H. D., and Nordai, F. L., “NP-Hard, Capacitated, Balanced p-Median Problems on a Chain Graph with a Continuum of Link Demands,” *Mathematics of Operations Research*, Vol. 13, No. 1, 1988, pp. 32–49. <https://doi.org/10.1287/moor.13.1.32>, URL <https://doi.org/10.1287/moor.13.1.32>.
- [27] Lloyd, S., “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, Vol. 28, No. 2, 1982, pp. 129–137. <https://doi.org/10.1109/TIT.1982.1056489>.
- [28] Kuhn, H. W., “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, Vol. 2, No. 1-2, 1955, pp. 83–97. <https://doi.org/https://doi.org/10.1002/nav.3800020109>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109>.
- [29] Astolfi, A., “Exponential Stabilization of a Wheeled Mobile Robot Via Discontinuous Control,” *Journal of Dynamic Systems, Measurement, and Control*, Vol. 121, No. 1, 1999, pp. 121–126. <https://doi.org/10.1115/1.2802429>, URL <https://doi.org/10.1115/1.2802429>.
- [30] Chang, T.-H., Hong, M., and Wang, X., “Multi-Agent Distributed Optimization via Inexact Consensus ADMM,” *IEEE Transactions on Signal Processing*, Vol. 63, No. 2, 2015, pp. 482–497. <https://doi.org/10.1109/TSP.2014.2367458>.
- [31] Haksar, R. N., Shorinwa, O., Washington, P., and Schwager, M., “Consensus-Based ADMM for Task Assignment in Multi-robot Teams,” *Robotics Research*, edited by T. Asfour, E. Yoshida, J. Park, H. Christensen, and O. Khatib, Springer International Publishing, Cham, 2022, pp. 35–51.
- [32] Shorinwa, O., Haksar, R. N., Washington, P., and Schwager, M., “Distributed Multirobot Task Assignment via Consensus ADMM,” *IEEE Transactions on Robotics*, Vol. 39, No. 3, 2023, pp. 1781–1800. <https://doi.org/10.1109/TRO.2022.3228132>.
- [33] Choi, H.-L., Brunet, L., and How, J. P., “Consensus-Based Decentralized Auctions for Robust Task Allocation,” *IEEE Transactions on Robotics*, Vol. 25, No. 4, 2009, pp. 912–926. <https://doi.org/10.1109/TRO.2009.2022423>.
- [34] Chopra, S., Notarstefano, G., Rice, M., and Egerstedt, M., “A Distributed Version of the Hungarian Method for Multirobot Assignment,” *IEEE Transactions on Robotics*, Vol. 33, No. 4, 2017, pp. 932–947. <https://doi.org/10.1109/TRO.2017.2693377>.
- [35] Mateos, G., Bazerque, J. A., and Giannakis, G. B., “Distributed Sparse Linear Regression,” *IEEE Transactions on Signal Processing*, Vol. 58, No. 10, 2010, pp. 5262–5276. <https://doi.org/10.1109/TSP.2010.2055862>.
- [36] Cortés, Jorge, Martínez, Sonia, and Bullo, Francesco, “Spatially-distributed coverage optimization and control with limited-range interactions,” *ESAIM: COCV*, Vol. 11, No. 4, 2005, pp. 691–719. <https://doi.org/10.1051/cocv:2005024>, URL <https://doi.org/10.1051/cocv:2005024>.
- [37] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S., “OSQP: An operator splitting solver for quadratic programs,” *Mathematical Programming Computation*, Vol. 12, No. 4, 2020, pp. 637–672.
- [38] David G. Luenberger, Y. Y., *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science, Springer Cham, 2021.