DDL: Empowering Delivery Drones With Large-Scale Urban Sensing Capability

Xuecheng Chen, Haoyang Wang, *Graduate Student Member, IEEE*, Yuhan Cheng, Haohao Fu, Yuxuan Liu, Fan Dang, *Member, IEEE*, Yunhao Liu, *Fellow, IEEE*, Jinqiang Cui, and Xinlei Chen, *Member, IEEE*

***Abstract*—Delivery drones provide a promising sensing platform for smart cities thanks to their city-wide infrastructure and large-scale deployment. However, due to limited battery lifetime and available resources, it is challenging to schedule delivery drones to derive both high sensing and delivery performance, which is a highly complicated optimization problem with several coupled decision variables. Meanwhile, this complex optimization problem involves multiple interconnected decision variables, making it even more complex. In this paper, we first propose a delivery drone-based sensing system and formulate a mixed-integer non-linear programming problem (MINLP) that jointly optimizes the sensing utility and delivery time, considering practical factors including energy capacity and available delivery drones. Then we provide an efficient solution that integrates the strength of deep reinforcement learning (DRL) and heuristic, which decouples the highly complicated optimization search process and replaces the heavy computation with a rapid approximation. Evaluation results compared with the state-of-the-art baselines show that *DDL* improves the scheduling quality by at least 46% on average. More importantly, our proposed method could effectively improve the computational efficiency, which is up to 98 times higher than the best baseline.**

***Index Terms*—Cyber-physical systems, deep reinforcement learning, drone swarm, smart cities.**

Manuscript received 11 September 2023; revised 15 April 2024; accepted 18 June 2024. Date of publication 19 July 2024; date of current version 2 September 2024. This work was supported in part by Yunnan Forestry and Grassland Science and Technology Innovation Joint Special Project under Grant 202404CB090017, in part by the Natural Science Foundation of China under Grant 62371269, in part by Guangdong Innovative and Entrepreneurial Research Team under Grant 2021ZT09L197, in part by Shenzhen 2022 Stabilization Support under Grant WDZC20220811103500001, and in part by the Tsinghua Shenzhen International Graduate School Cross-disciplinary Research and Innovation Fund Research under Grant JC20220011. The guest editor coordinating the review of this article and approving it for publication was Dr. Wenbo Ding. *(Xuecheng Chen and Haoyang Wang contributed equally to this work.) (Corresponding author: Xinlei Chen.)*

Xuecheng Chen and Yuxuan Liu are with the Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen 518071, China (e-mail: chenxc21@mails.tsinghua.edu.cn; yuxuan-l21@mails.tsinghua.edu.cn).

Haoyang Wang, Yuhan Cheng, and Haohao Fu are with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518071, China (e-mail: haoyang-22@mails.tsinghua.edu.cn; cyh22@mails.tsinghua.edu.cn; fhh22@mails.tsinghua.edu.cn).

Fan Dang is with Global Innovation Exchange, Tsinghua University, Beijing 100190, China (e-mail: dangfan@tsinghua.edu.cn).

Yunhao Liu is with Global Innovation Exchange, Tsinghua University, Beijing 100190, China, and also with the Department of Automation, Tsinghua University, Beijing 100190, China (e-mail: yunhao@tsinghua.edu.cn).

Jinqiang Cui is with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: cuijq@pcl.ac.cn).

Xinlei Chen is with Shenzhen International Graduate School, Tsinghua University, Shenzhen 518071, China, also with Peng Cheng Laboratory, Shenzhen 518055, China, and also with RISC-V International Open Source Laboratory, Shenzhen 518056, China (e-mail: chen.xinlei@sz.tsinghua.edu.cn).

Digital Object Identifier 10.1109/JSTSP.2024.3427371

I. Introduction

DRONES (aka. unmanned aerial vehicles) are becoming a pivotal sensing platform for large-scale urban sensing. Due to their three-dimensional mobility, autonomy, and sensing capacity, drones can gather data with high efficiency and perform in more flexible urban sensing circumstances compared to ground sensor networks [1], [2]. In particular, when multiple drones are organized into a cohesive system, they are able to handle complex tasks, such as searching for available parking spots, detecting traffic congestion, and monitoring air pollution [3], [4].

Researchers have proposed different methods for scheduling drones to gather sensor data, most of them have primarily concentrated on dedicated drone scheduling [5], [6], [7]. However, when conducting large-scale urban sensing, it requires a high cost to build a large-scale drone swarm and a lot of human effort for maintenance. Specifically, commercially available drones like the DJI Inspire 3 [8] is more than $18,000, so that a swarm with 50 drones may require around one million dollars. Besides, since the battery life of commercial drones is usually less than half an hour, it requires human labor to maintain charging stations and recharge the drones during long operations.

Fortunately, due to the popularity of instant delivery and the online shopping market [9], various delivery drones from giant companies have emerged, which provide opportunities for large-scale urban sensing in a cost-efficient way. Especially, Meituan [10], one of China's top internet companies, has delivered 170,000 meals by drones in the last two years in Shenzhen, China. The proliferation of delivery services brings about the city-wide distribution of delivery stations in high density, which indicates delivery drones can cover a large part of the city.

In light of this, we advocate conducting urban sensing with delivery drones. As shown in Fig. 1, a group of delivery drones equipped with efficient sensors starts from the starting station, visits the sensing task locations sequentially, and finally lands at the terminal station of delivery. Typically, the sensing task can be the inspection and mapping of important structural objects in the city such as buildings, bridges, and power lines [11], [12], [13], promoting the following operation of multiple robots. In this way, delivery drones provide low-altitude and high-accuracy sensing with little extra deployment effort [14]. Furthermore, by leveraging the communication, and ubiquitous computing services of delivery companies, the extra purchase and maintenance costs can be greatly reduced.

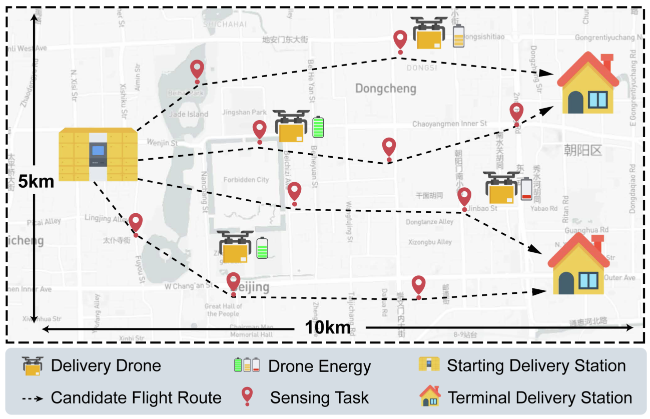


Fig. 1. Illustration of hiring delivery drones to perform urban sensing.

Our work aims to schedule the delivery drones intelligently so that they satisfy both primary delivery demands and accomplish high-quality sensing tasks. Albeit inspiring, translating this intuition into a practical delivery drones-based sensing system is non-trivial and faces significant challenges:

- *Inconsistent goals of delivery and sensing:* To summarize, our objective involves two parts: visiting all scattered delivery destinations and allocating time at particular stops to collect sensing data. The former requires efficiency as customers would want their delivery to take as little time as possible, while the latter demands ample time for data collection tasks. Besides, delivery tasks require that our drone arrive at a particular destination, which is oftentimes different from the places where sensing tasks take place. This discrepancy in preferences poses a challenge in devising a mechanism that achieves high sensing performance without compromising delivery performance. Therefore, we are to design a solution that captures the sweet spot that finds a balance between these two goals. This adds another level of complexity to the average optimization problem, which leads to the second major challenge below [15].
- *Complex mixed integer optimization problem with two intertwined decision variables:* Combining two inconsistent goals leads to a series of new problems, including (i) assigning sensing tasks to drones, (ii) determining the visiting order for each drone, and (iii) allocating sensing time for each task. Optimizing these issues is akin to solving a variant of the NP-hard capacitated vehicle routing problem (CVRP) [16], but it's more complex because the time consumption of each task node is uncertain and needs to be decided by our algorithm. Moreover, the involved decision variables including discrete elements (task locations) and continuous factors (sensing time) jointly impact drones' energy consumption, leading to a complex mixed-integer programming problem. Consequently, exhaustive search or conventional optimization methods are infeasible due to their computational demands and time-consuming nature.

To tackle the above challenges, we present *DDL*, a deep reinforcement learning (DRL)-enhanced city-scale smart sensing system based on the delivery drone swarm. *DDL* optimizes both the sensing utility and delivery time with a limited energy reserve.

To address the first challenge, We come up with a delivery scheduling method that optimizes the task assignment, route planning, and sensing time allocation for delivery drones to ensure that sensing utility is maximized while delivery time is minimized with reasonable constraints on energy capacity and the number of delivery drones. The intuition behind combining these two seemingly conflicting goals is the hidden similarity between patterns in sensing and delivery requests. Frankly speaking, when choosing between different paths to get to the delivery destination, there are room for us to alternate so that we could traverse stops for sensing tasks without sacrificing significant delivery time. We believe that by smartly assigning drones and geographical pairing between sensing and delivery tasks, we can stitch sensing stops into the delivery track so that we can catch two birds with one stone, and our results confirm this idea. We formalize the delivery drone scheduling problem as a dual objectives mixed-integer non-linear programming problem (MINLP), where an adjustable trade-off coefficient is used to balance the importance of the two objectives.

To address the second challenge, we propose a novel *DRL-enhanced hierarchy scheduling* approach that decomposes the original problem into a two-stage optimization process, which contributes to fast approximating the optimal solution. Specifically, We design an upper-level learning-based method to optimize the task assignment and route planning strategies, and a lower-level heuristic algorithm determines the sensing time allocation for each task to maximize the sensing utility. During each optimization iteration, we enhance the allocation of sensing time for each flight route based on the current task assignment and route planning results. This updated allocation is then used to update the task assignment and route planning. This approach combines the advantage of DRL and heuristic algorithm, building a neural network that can efficiently explore the large search space without heavy computation. In summary, the main contributions of this paper are listed as follows:

- Propose a delivery drone-based sensing system *DDL*, and formulate a mixed-integer non-linear programming problem (MINLP) to jointly optimize sensing and delivery performance, considering practical factors.
- Provide a *DRL-enhanced hierarchy scheduling* approach that decomposes the original problem into a bi-level optimization process. We adopt neural networks to replace the heavy computation in the upper-level optimization and solve the simpler lower-level optimization problem with the heuristic method.
- Evaluate our solution with extensive experiments based on real-world delivery station data. Results show that our proposed algorithm improves the scheduling quality and computation efficiency by at least 46% and 98% compared with the state-of-the-art baselines, respectively.

This paper presents an extended version of our previous work [17]. The major modifications include problem definition, method, and evaluation. The remainder of the paper is organized as follows: Section II summarizes the related work. Section III presents the system and problem definition. Section V introduces system overview, key components, and algorithms. Section VI demonstrates extensive experimental results based on a real-world dataset. Section VII concludes the paper.

II. Related Work

A. Drone Scheduling for Urban Sensing

Drone-based sensing systems have drawn much attention in both industry [18], [19] and research communities [20], [21]. Based on the characteristics of drones, many approaches have been proposed to dispatch drones for various tasks, such as monitoring [22], [23], surveillance [7], [24], and safety inspection [25], [26]. In such drone-based sensing systems, the sensing task is typically collecting sensing data for the specific application, and the utility of sensing refers to a combination of the sensing quality and sensing quantity [27]. The sensing quality is influenced by many task-specific factors, such as the sensing angle [28], sensing distance [24], and environmental situation [25], [29]. A longer duration for executing a sensing task can result in the collection of more sensing data, thereby enhancing the overall sensing utility [30], [31]. For example, when a drone equipped with a camera inspects disasters, the distance between drone and object has a significant impact on the quality of the image [32], [33], a longer sensing time allows the drone to gather more images [24], [34].

Several studies have explored the scheduling of delivery drones for sensing tasks. For example, Xiang et al. [35] are the first to study using delivery drones for crowdsensing. However, their design made efforts to manually derive an explicit heuristic algorithm for the optimization problem, which is quite time-consuming when scaled to more variables. Although Tao et al. [36] started to apply a reinforcement learning algorithm to address the trajectory design problem of delivery drones, they simplify the drones' energy consumption model without considering the delivery weights. More recently, Chen et al. [17] also aims at scheduling delivery drones, but their optimization objective does not take the delivery time into account, which is not feasible in practical instant delivery scenarios.

B. DRL for Combinatorial Optimization

Combinatorial optimization (CO) problems are commonly used in various fields [37], [38]. The most significant challenge is the search space explosion (i.e., the probability grows rapidly with the size of the problem, resulting in no polynomial-time solutions) [39]. Traditionally, CO problems are solved via manually designed heuristics that sequentially construct a solution [40]. However, these hand-crafted methods highly rely on expert knowledge about the optimization algorithm, which may not be sufficient [41]. Besides, these approaches usually suffer from heavy computation when dealing with large-scale problems.

Recently, reinforcement learning (RL) has been adopted to solve CO problems with better and faster results. RL enables the algorithm to learn and evolve, via interaction with environments or knowledge induction with look-ahead search [42], [43]. Therefore, it is a natural tool to make algorithmic decisions in a more principled and optimized way through supervised or self-supervised training. RL has shown effectiveness in solving various CO problems, including traveling salesman problem (TSP) [44], vehicle routing problem (VRP) [45], and maximum cut (MC) [46]. The majority of these works follow an end-to-end manner, which is directly constructing a solution from scratch [47]. More recently, researchers proposed hybrid approaches that combine machine learning and traditional solver [47], [48]. However, these methods cannot be directly applied to our problem, since we are dealing with a complex dual-objective scenario. Our goal is to design an efficient DRL-enhanced solution to enable the delivery drone-based sensing system.

TABLE I
Important Notations

Notation	Description
$k, \mathcal{K}$	Delivery drone k, set of delivery drones for a Delivery Team
$v, \mathcal{V}$	Sensing task, set of sensing tasks
$r, \mathcal{R}$	Flight route r, set of flight routes
E_k	Total energy consumption for the drone k
E_h, E_f	Hovering energy cost, flying energy cost
P_f, P_h	Energy power of flying and hovering for drones
σ_k	Battery capacity for the drone k
d_0	Flying distance wo/ sensing
d_k	Flying distance w/ sensing
d_{ij}	Distance between task i and task j
u_v	Utility of task v in unit time (sensing utility weight)
U_v	Utility upper bound of task v
x_{ij}^k	Indicator of whether the drone k takes the route r_{ij}
t_v^k	Allocated sensing time for task v on route k
w_k	Delivery weight of delivery drone k

III. Problem Definition

In this section, we first provide definitions and background in Section III-A. Subsequently, we discuss the objective of package delivery and urban sensing in SectionIII-B. Finally, we formulate the problem of delivery drone scheduling in Section III-C. For ease of reference, the key notations for the system design are shown in Table I.

A. Background and Definitions

Fig. 2 illustrates the architecture of the delivery drone-based urban sensing system, which consists of four components. The *Drone Delivery Company* provides a large number of delivery drones equipped with different sensors according to the requirement of the *Application System* [49]. The delivery task information, including the depots and destinations of the delivery tasks, as well as the available drone number and energy limitation are informed to the *Scheduling System*. The *Scheduling System* then schedules the delivery drone to collect the sensing data based on the delivery task information and sensing task information (e.g., tasks' locations and utilities) [50]. The *Data Request End* collects and analyzes the sensing data [51], [52] to infer the phenomena of common interest, which can be used by the *Application System*.

A typical delivery system follows a tree structure [53], composed of starting delivery stations and terminal delivery stations. The daily products are dispatched from the starting delivery stations to the terminal delivery stations. If a delivery drone is recruited, it will be equipped with various sensors according to practical application, such as PM2.5 sensors for air quality

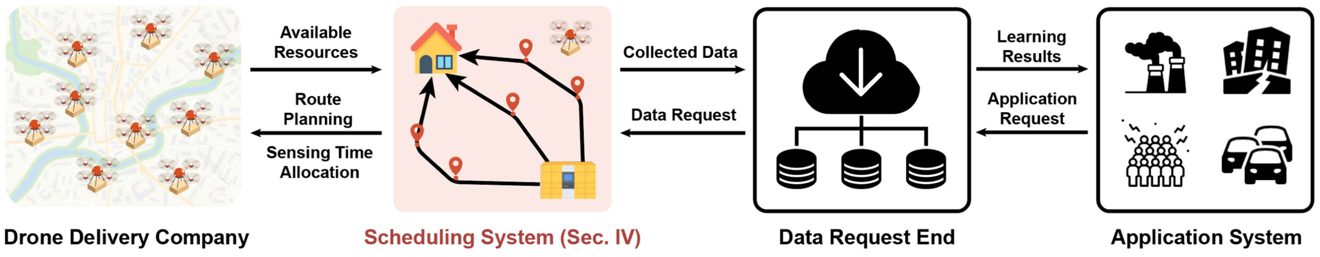


Fig. 2. Illustration of *DDL*'s architecture, which is composed of four subsystems.

measurement [54]. Later, given the delivery task information and sensing task information, the *Scheduling System* decides (1) *Sensing Task Assignment*, which selects part of sensing task to sense for each drone [55]; (2) *Sensing Time Allocation*, which allocate time of each selected sensing tasks to collected sensory data; (3) *Drone Scheduling*, which schedules drones to visit and sense the selected sensing tasks one by one while delivering packages from predefined depots to destinations. Note that the Scheduling System is not dependent on the particular applications and can be used for any type of high-level data collection task. The design details will be discussed in Section V. We define other key concepts as follows.

Delivery Team (DT): We define the drones for the same delivery pair between starting delivery station and terminal delivery station as a Delivery Team (DT). Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote a set of delivery drones in a DT, which has up to K delivery drones. The direct flying distance from the starting delivery station to the terminal delivery station is d_0, while the flying distance with sensing for drone k is d_k. The maximum available energy of the drone k is σ_k.

Sensing Task: In the urban area, there is a set of V sensing tasks, denoted as $\mathcal{V} = \{1, 2, \ldots, V\}$, which are distributed. Each task $v \in V$ is assigned to a specific delivery drone k, and the drone takes a certain amount of time t_v^k to complete the task. The quality of the sensing is influenced by various physical factors specific to each application, such as sensing distance and unpredictable wind. Generally, a longer sensing time leads to better results [30], [31]. Therefore, we define u_v as the benefit obtained by the drone per unit of time spent on task v, and U_v as the maximum utility that can be achieved. If the total utility obtained from sensing exceeds the upper bound, it does not result in any additional benefit. Then, the *sensing utility* model can be expressed as:

$$U\left(h_v^k, t_v^k\right) = \min\left\{u_v t_v^k h_v^k, U_v\right\}, \tag{1}$$

where the binary variable h_v^k denotes whether task v is selected by drone k. Thus, the drone k can allocate sensing time to execute task v only when it has planned to visit this task (i.e., $h_v^k = 1$).

Flight Route: A flight route is composed of a set of sequentially visited task nodes and denoted as:

$$r_k = \{v_c, \ldots, v_r\}, \tag{2}$$

where v_c is a starting delivery station (i.e., depot) and v_r is a terminal delivery station (i.e., destination). For example, a route $[v_c, 3, 2, 5, v_r]$ is a traveling plan that starts at the depot v_c, visits task nodes 3,2, and 5 sequentially, and finally flies to the destination v_r.

Energy Consumption: Typically, the energy consumption of a drone can be categorized into two primary components: the energy expenditure during flight and the energy consumption while hovering. Let $P_f(w_k)$ and $P_h(w_k)$ denote the powers of flying and hovering with package weight w_k, respectively. Similar to existing work [35], we consider the impact of delivery weight on the powers as follows:

$$P_h\left(w_k\right) = \rho_0^{\mathrm{h}} + \rho_1^{\mathrm{h}} w_k, \tag{3}$$

$$P_f\left(w_k\right) = \rho_0^{\mathrm{f}} + \rho_1^{\mathrm{f}} w_k, \tag{4}$$

where ρ_0^{h}, ρ_1^{h}, ρ_0^{f} and ρ_1^{f} are the environment-dependent model parameters. Hence, the mathematical expression representing the cumulative energy consumption of a given delivery drone k can be formulated as:

$$E_k = P_f(w_k)\frac{d_k}{s} + P_h(w_k)\sum_{v\in\mathcal{V}} t_v^k \tag{5}$$

$$= P_f(w_k)\sum_{i,j\in\mathcal{V}}\frac{x_{ij}^k d_{ij}}{s} + P_h(w_k)\sum_{v\in\mathcal{V}} t_v^k, \tag{6}$$

where d_{ij} is the flying distance from task i to task j, and d_k is the total flying distance of drone k. s denotes the flying speed and x_{ij}^k indicates whether the drone k has taken the route from task i to task j. Note that the communication energy of drones only accounts for a little part of the consumption, thus it can be neglected to some extent.

B. Optimized Objectives

Driven by the demands of urban sensing and delivery enterprises, our study embraces two distinct types of decision variables. The binary decision variable $x_{ij}^k \in \{0, 1\}$ signifies whether drone k has opted for the route connecting task i and task j, whereas the continuous decision variable t_v^k symbolizes the duration of sensing time dedicated to task v by drone k. For simplicity, we denote the decision variables as $\mathbf{x} := \{x_{ij}^k\}, \mathbf{t} := \{t_v^k\}$. Given the available resources and initial states, our target is to optimize the following two goals:

Goal 1. Maximize the sensing utility: Our primary objective is to optimize the scheduling of the delivery drone in order to maximize the overall sensing utility. As definite in (1), we want to collect sensing data as much as possible without exceeding the utility upper bound. Since we use the decision variable x_{ij}^k to denote whether drone k has been allocated the route connecting task i and task j, we have $h_v^k = \max\{x_{iv}^k\}, i \in \mathcal{V}$. This means that $h_v^k = 1$ only when the routes allocated to drone k involve task v. Therefore, our first goal can be mathematically expressed as:

$$\begin{aligned}\max U(\mathbf{x}, \mathbf{t}) &= \max \sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}} U(h_v^k, t_v^k) \\ &= \max \sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}} \min\left\{u_v t_v^k h_v^k, U_v\right\} \\ &= \max \sum_{k\in\mathcal{K}} \sum_{i,v\in\mathcal{V}} \min\left\{u_v t_v^k \max\left\{x_{iv}^k\right\}, U_v\right\}.\end{aligned} \tag{7}$$

Goal 2. Minimize the delivery time: Typically, delivery companies aim to minimize delivery times, as customers prefer their packages to arrive as quickly as possible [56]. The total delivery time is composed of the flying time $T_f(\mathbf{x}, \mathbf{t})$ and the hovering time (sensing time) $T_h(\mathbf{x}, \mathbf{t})$. Therefore, our second goal of the total delivery time optimization can be given by:

$$\begin{aligned}\min T(\mathbf{x}, \mathbf{t}) &= \min\left(T_f(\mathbf{x}, \mathbf{t}) + T_h(\mathbf{x}, \mathbf{t})\right) \\ &= \min \sum_{k\in\mathcal{K}} \left(\sum_{i,j\in\mathcal{V}} \frac{x_{ij}^k d_{ij}}{s} + \sum_{v\in\mathcal{V}} t_v^k\right).\end{aligned} \tag{8}$$

C. Problem Formulation

Our objective is to maximize *scheduling quality* (SQ), which is defined as a weighted combination of sensing utility and delivery time. The mathematical formulation of delivery drone scheduling problem is given as:

$$\max_{\mathbf{x},\mathbf{t}} SQ = \alpha U'(\mathbf{x}, \mathbf{t}) - (1-\alpha) T'(\mathbf{x}, \mathbf{t}), \tag{9}$$

$$\text{s.t. } x_{ij}^k \in \{0,1\}, t_v^k \geq 0, \quad \forall i, j, v \in \mathcal{V}, \forall k \in \mathcal{K}, \tag{10}$$

$$\sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{V}, i\neq j} x_{ij}^k = 1 \quad \forall j \in \mathcal{V}, \tag{11}$$

$$\sum_{j\in\mathcal{V}} x_{v_c j}^k = 1 \quad \forall k \in \mathcal{K}, \tag{12}$$

$$\sum_{j\in\mathcal{V}} x_{j v_r}^k = 1 \quad \forall k \in \mathcal{K}, \tag{13}$$

$$\sum_{i\in\mathcal{V}, i\neq j} x_{ij}^k - \sum_{i\in\mathcal{V}} x_{ji}^k = 0 \quad \forall j \in \mathcal{V}, \forall k \in \mathcal{K}, \tag{14}$$

$$E_k \leq \sigma_k \quad \forall k \in \mathcal{K}, \tag{15}$$

where α is a trade-off coefficient to adjust the priority of delivery and sensing, according to the specific requirement. $U'(\mathbf{x}, \mathbf{t}) \in [0,1]$ and $T'(\mathbf{x}, \mathbf{t}) \in [0,1]$ are the normalized values of $U(\mathbf{x}, \mathbf{t})$ and $T(\mathbf{x}, \mathbf{t})$, respectively. The normalization aims to make two parts range in a similar order of magnitude. Constraint (11) guarantees that each task is exclusively assigned to a single delivery drone. Constraint (12) and Constraint (13) state that each delivery drone can leave the depot and land at the destination only once, respectively; Constraint (14) enforces that the numbers of drones coming in and out of a task's location are the same; Constraint (15) declares that the energy available to each drone is limited and cannot exceed its battery capacity σ_k. The formulation indicates that, to maximize the objective of the system, we need both higher sensing delivery utility and shorter delivery time, without exceeding the limitation of energy and the number of available drones. Nevertheless, the decision variables within this problem, encompassing discrete and continuous types, are intricately intertwined, ultimately transforming the joint optimization problem into a Mixed Integer Non-Linear Programming Problem (MINLP).

IV. DDL Architecture

In this section, we first show the problem has high computational complexity. Then, we propose the surrogate function construction to decompose the original problem. Finally, we provide an overview of our algorithm, which integrates a reinforcement learning-based method and a learning-free heuristic method.

A. Complexity Analysis

Given the formulation in Section III-C, we can see that the problem includes two types of optimization. The first one involves the sensing task selection and tour scheduling while the second one involves sensing time allocation for each selected task.

Theorem 1: The formulated delivery drone scheduling problem is NP-hard.

Proof: We prove this theorem by reducing our delivery drone scheduling problem to a Capacitated Vehicle Routing Problem (CVRP) [16], which is known to be NP-hard. In CVRP, the goal is to efficiently distribute goods or services to a set of customers using a fleet of vehicles, while satisfying capacity constraints and minimizing transportation costs. Each vehicle can serve customers as long as their total demand doesn't exceed its capacity.

In our delivery drone scheduling problem, we equate the energy constraint to capacity limitations, with sensing tasks analogous to customers. Minimizing delivery time corresponds to minimizing transportation costs. If the energy consumption for each sensing task were known, our problem would be simplified to CVRP. However, our problem is more intricate due to uncertainties in energy consumption, arising from practical factors like flying time, delivery weight, and sensing time (one of the decision variables). Consequently, we demonstrate the NP-hardness of our problem. ■

Because of this NP-hardness, as the number of sensing tasks and delivery drones increases, finding the optimal solution becomes computationally infeasible. Therefore, our objective lies in the pursuit of a solution that is near-optimal within a pragmatic time, as opposed to the relentless pursuit of an exact solution.

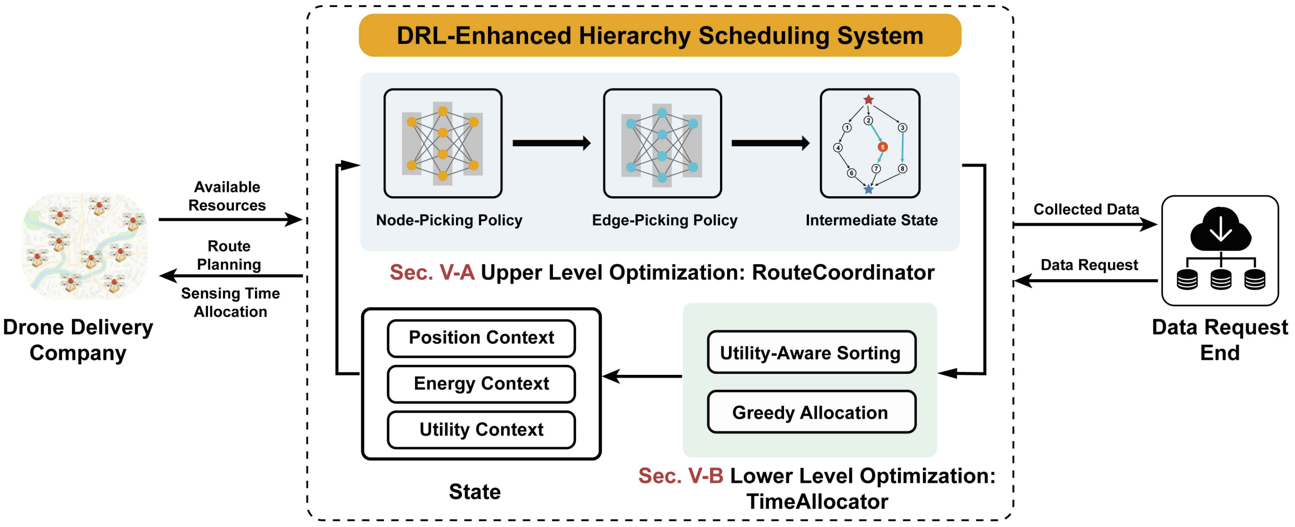


Fig. 3. An illustration of the algorithm. Given the current feasible solution, the algorithm iteratively updates the solution with an upper-level learning-based method and a lower-level heuristic method. After a certain number of steps, we choose the best one among all visited solutions.

B. Bi-Level Reformulation of Original Problem

We consider the optimization problem with a graph $\mathcal{G}(\mathcal{V},\mathcal{R})$, where $\mathcal{V},\mathcal{R}$ are the set of candidate sensing task nodes and flight routes, respectively. Without loss of generality, the original problem in (9) can be abstracted as:

$$\max_{\mathbf{x},\mathbf{t}} f(\mathbf{x},\mathbf{t}|\mathcal{G}) \quad \text{s.t.} \quad h(\mathbf{x},\mathbf{t},\mathcal{G}) \leq 0, \tag{16}$$

where $f(\mathbf{x},\mathbf{t}|\mathcal{G})$ denotes the objective function given the input graph $\mathcal{G}$, $h(\mathbf{x},\mathbf{t},\mathcal{G}) \leq 0$ represents the set of constraints, and $\mathbf{x}$, $\mathbf{t}$ indicate the decision variable (i.e., solution).

To ease the challenge of high complexity, we exploit the concept of adapting the original problem to facilitate the problem-solving process [48]. We propose to gradually improve the solution quality of this complex problem by solving two levels of problems alternatively, where $f_u(\mathbf{x}',\mathcal{G}'|\mathcal{G})$, $f_l(\mathbf{t}'|\mathbf{x}',\mathcal{G}')$ serve as objectives for upper and lower level problems, respectively. Specifically, for the upper-level problem $f_u(\mathbf{x}',\mathcal{G}'|\mathcal{G})$, we target achieving the optimal sensing task assignments and tour scheduling solution $\mathbf{x}^*$, via transitional modifications from $\mathcal{G}$ to $\mathcal{G}^*$; For the lower-level problem $f_l(\mathbf{t}'|\mathbf{x}',\mathcal{G}')$, we aim to allocate the optimal sensing time t^* for each selected sensing task, through constructing a value mapping from $\mathbf{x}^*$ to the objective function in (9). With this bi-level reformulation, we start with a feasible solution and optimize these two levels of problems alternatively, resulting in a continuously refined solution. Building upon it, we are capable of optimizing the complex upper-level problem by an DRL agent, while the lower-level problem with less computation can be solved by a heuristic algorithm. Overall, the bi-level reformulation allows *DDL* to unite the strength of DRL and heuristic algorithms, leading to improved solution quality and computational efficiency. More details of the algorithm design are elaborated in Section V.

C. System Overview

Fig. 3 illustrates the hierarchy framework of our algorithm. The framework is composed of two parts, including a learning-based *RouteCoordinator* and a learning-free *TimeAllocator* for the upper level and lower level problems, respectively. RouteCoordinator is designed to assign the sensing task and plan the route simultaneously. TimeAllocator decides the amount of sensing time allocated to each sensing task, based on the results from RouteCoordinator.

At a higher level, the framework acts as an intelligent agent and learns the searching policies based on the Neural Rewriter architecture [57]. We first construct a feasible solution, then the algorithm improves the solution iteratively with RouteCoordinator and TimeAllocator. After a certain number of steps, we choose the visited solution with the best objective value as our final solution. During the update process, all practical constraints in Eqs. (11)–(15) remain satisfied. Therefore, with the guarantee of feasibility, the data-driven characteristic of reinforcement learning enables the algorithm to explore the large search space, and potentially learn the searching policies with the best performance out of data. On the other side, the neural network replaces some heavy computation via a fast approximation at the upper level, then the sensing time allocation problem can be efficiently solved by heuristic at the lower level. The details of the algorithm components are defined in the next section.

V. Algorithm Design

In this section, we describe the algorithm design of *DDL*. We first present the design of RouteCoordinator in Section V-B, followed by descriptions of TimeAllocator in Section V-C and training details in Section V-D.

Fig. 4 gives an illustration of the algorithm's decision process. Given the current state s_t (i.e., current feasible solution), we first select a task node using the *node-picking policy* $\pi_\omega(\omega_t|s_t)$,

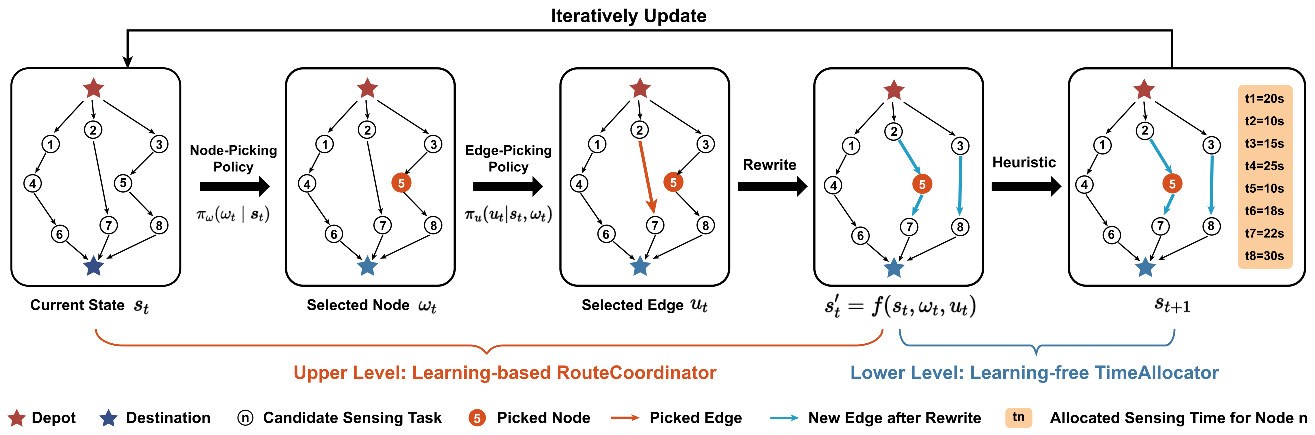


Fig. 4. Illustration of the structure of the proposed algorithm.

where ω_t is the selected node. We then pick an edge between two task nodes using the *edge-picking policy* $\pi_u(u_t \mid s_t, \omega_t)$, where u_t is the selected edge. After rewriting the constructed routes based on the selected node and selected edge, we obtain an intermediate state s'_t, which can be further optimized by the learning-free TimeAllocator. Algorithm 1 demonstrates the main steps when deploying the *DDL* framework.

1) State Space: In our setting, each solution to the optimization problem is a state. Considering the practical applications, we use the position, energy consumption, and utility as contextual information to construct features for each task node $v \in V$ in the solution. The corresponding features are encoded as a nine-dimension vector e_v and fed into the network. Specifically, the state features include:

- *position states:*$(x_v, y_v, x_v^p, y_v^p, Dis)$, where (x_v, y_v) denotes the node position, (x_v^p, y_v^p) represents the position of the node visited at the last time step, and Dis is the distance between these two nodes. Note that we incorporate Dis as an auxiliary feature since it provides explicit information to simplify model understanding and thus speed up the training process.
- *energy states:* $(\frac{\mu_v}{\sigma_k}, Cap_k)$, where μ_v is the energy consumed by flying from the previous node to the current node and σ_k is the battery capacity of the corresponding drone k. Cap_k is the remaining energy for the drone k.
- *utility states:* $(\frac{U_v}{u_v}, u_v)$, where U_v represents the utility upper bound of task v and u_v is the utility of task v in unit time. Thus the first term indicates the maximal amount of time to allocate.

A. Routecoordinator

1) Action Space: Actions can be classified into two types, node-picking and edge-picking. The node-picking policy $\pi_\omega(\omega_t \mid s_t)$ attempts to predict a score for each task node in the current solution. This score is a measurement of the benefit for rewriting each node, and a higher score indicates we could obtain a higher objective value by rewriting the corresponding node. Similarly, the edge-picking policy $\pi_u(u_t \mid s_t[\omega_t])$ aims to predict a score for each edge between the task nodes. As shown in Fig. 4, based on these two distributions of scores, we could sample a node (e.g., red circle) and an edge (e.g., red arrow) from the current solution, and then replace the selected edge with the connections to the selected node (e.g., blue arrow).

The cooperation of the node-picking policy and the edge-picking policy allows us to reduce the total flying distance by modifying the position of arbitrary task nodes intra-route or inter-route. The inter-route operation represents moving the task node among different routes, while the intra-route operation means moving the task node to a different position in an individual route. Therefore, both the task assignment and route planning for each drone can be continuously improved.

2) Reward: The reward function focus on the improvement of the objective function compared with the last state. We can compute the reward for each step as: $r(s_t, \omega_t, u_t) = SQ(s_{t+1}) - SQ(s_t)$, where $SQ(\cdot)$ is the objective function definite in (9). The reward function encourages the algorithm to move to a neighbor solution that has the largest objective improvement.

B. TimeAllocator

Assuming the RouteCoordinator has given a good task assignment and route planning solution s'_t without breaking the constraints, then the objective function is only related to the time allocation strategy. Therefore, our mission can be regarded as finding a mapping from the intermediate state s'_t to the sensing utility.

Given the task assignment result for each route, we rank the task first according to their utility in decreasing order. Then, we greedily allocate sensing time for each task, until it reaches the upper bound or there is no energy left. In other words, we prefer to allocate more time to the task with higher utility, since we could achieve a better objective value.

C. Training Details

In the training process, the model aims to maximize the expected return from each state s_t. The return is defined as a total sum of discounted rewards $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where $\gamma \in (0, 1]$ is a discount factor. We train the node-picking policy π_ω and the edge-picking policy π_u simultaneously.

Algorithm 1: *DDL* Algorithm.

```
Input: Task set $\mathcal{V}$, depot $v_c$, destination $v_r$, utility $\{u_v\}$,
  utility upper bound $\{U_v\}$
Output: Flight route $\{\mathcal{R}\}$, time allocation $\{t_v^k\}$
 1: for each decision period do
 2:   Update states: update the state based on the sensing
      and delivery tasks information.
 3:   Route scheduling: assign the task and decide the
      visiting order
 4:   for each route in the route list do
 5:     Sort tasks: sort the tasks by the utility descending
 6:     for each task in the current route do
 7:       Time allocation: greedily allocates sensing time
          until it reaches the upper bound or no energy left
 8:     end for
 9:   end for
10: end for
```

For the node-picking policy π_ω in RouteCoordinator, we use the following equation to select the node in the current solution with probability and then rewrite it:

$$\pi_\omega\left(\omega_t \mid s_t ; \theta\right)=\frac{\exp \left(Q\left(s_t, \omega_t ; \theta\right)\right)}{\sum_{\omega_t} \exp \left(Q\left(s_t, \omega_t ; \theta\right)\right)}, \tag{17}$$

where Q is an approximate action-value function with parameters θ. Similar to traditional value-based reinforcement learning methods, our action-value function is represented using a Multi-Layer Perceptron (MLP) neural network. The parameter θ of π_ω is updated by the following loss function:

$$L_\omega(\theta)=\frac{1}{T} \sum_{t=0}^{T-1}\left(\sum_{t^{\prime}=t}^{T-1} \gamma^{t^{\prime}-t} r\left(s_{t^{\prime}},\left(\omega_{t^{\prime}}, u_{t^{\prime}}\right)\right)-Q\left(s_t, \omega_t ; \theta\right)\right)^2, \tag{18}$$

where T is the length of the total rewriting steps, and γ is the discount factor.

For the edge-picking policy π_u, we use the advantage-actor-critic algorithm to update the parameters of the policy network, because the algorithm is capable of handling sample insufficiency and instability in training. At each iteration, the edge-picking policy updates the parameter using the following loss function:

$$L_u(\phi)=-\sum_{t=0}^{T-1} \Delta\left(s_t,\left(\omega_t, u_t\right)\right) \log \pi_u\left(u_t \mid s_t[\omega_t] ; \phi\right), \tag{19}$$

$$\Delta\left(s_t,\left(\omega_t, u_t\right)\right)=\sum_{t^{\prime}=t}^{T-1} \gamma^{t^{\prime}-t} r\left(s_{t^{\prime}},\left(\omega_{t^{\prime}}, u_{t^{\prime}}\right)\right)-Q\left(s_t, \omega_t ; \theta\right), \tag{20}$$

where $\Delta(s_t,(\omega_t,u_t))$ is the advantage function.

Finally, the total loss function is $L=L_u(\phi)+\lambda L_\omega(\theta)$, where λ is a coefficient.

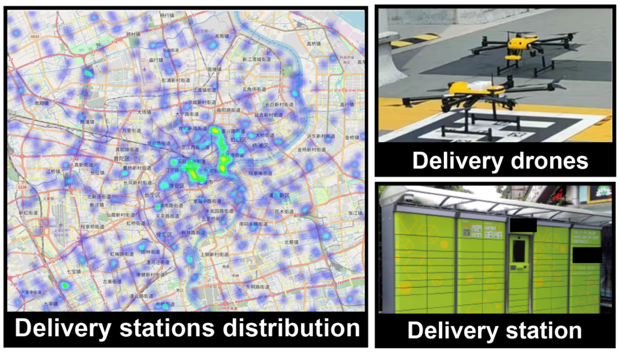


Fig. 5. Heat map of the distribution of delivery stations, as well as examples of delivery drones and delivery stations.

VI. Evaluation

In this section, we first introduce the evaluation setup for experiments in Section 5.1. Then, the details of the evaluation results and analysis are shown from Section 5.2 to Section 5.4. In the evaluation, we aim to demonstrate the performance of *DDL* in following aspects:

1) We evaluate the performance of our *DDL* system on the scheduling quality, scheduling efficiency, and the total number of drones after scheduling.
2) We validate the effectiveness of our system under the impact of different key factors.
3) We characterize the system computation cost in run time compared with the state-of-the-art algorithms under different system settings.

A. Experimental Methodology

Experiment Setup: To evaluate our system, we implemented a delivery drone simulation environment capable of supporting more than 300 drones for various experiments. This simulator incorporates delivery service stations based on a real-world dataset from 520 stations in Shanghai city. Among these stations, 58 serve as starting points, and 462 serve as terminal points. The area of interest for evaluation spans 80 $km \times 80$ km, with a partial representation shown in Fig. 5. The default number of sensing tasks is set to 20, generated randomly within this area. The sensing utility weight is uniformly distributed in the range of [2, 15], with a default upper bound of 2000. Furthermore, the dataset is divided into training and test sets in an 8:2 ratio.

Each drone starts with an initial energy of 57,000 KJ by default. The determination of delivery weight and other parameters in the energy consumption model is based on established research and prior references [58], [59]. Additionally, the discount factor for the reward is set to $\gamma = 0.8$, as determined through an ablation study.

Experiment Platform: We evaluate our *DDL* based on the same hardware configuration, Intel Core i7, 16 G memory, RTX2080Ti. All the algorithms are implemented in Python.

Metrics: To measure the experimental results, we mainly focus on the following four metrics:

- *Scheduling Quality:* Scheduling quality reflects the overall performances of the algorithm on the sensing objective and delivery objective. α is set as 0.5, which means equal importance between the sensing performance and the delivery performance. Therefore, the scheduling quality in the evaluation is calculated as:

$$SQ(V,K) = \max_{\mathbf{x},\mathbf{t}} \quad 0.5\,U'(V,K) - 0.5\,T'(V,K). \quad (21)$$

According to (9), the value of $SQ(V,K)$ lies in $[-0.5, 0.5]$.

- *Scheduling Efficiency:* Scheduling efficiency is defined as the ratio between sensing utility and total energy consumption. It reflects the algorithm's ability to reduce energy consumption and improve the sensing performance simultaneously. Considering the limited battery capacity of delivery drones, it is vital to improve the scheduling efficiency ρ, which is described as:

$$\rho = \frac{U(V,K)}{E_k}. \quad (22)$$

- *Run time:* We use the run time of all evaluated algorithms on the same hardware to compare the computation cost.

Baselines: To comprehensively evaluate the performance of *DDL*, five baselines are implemented for comparison. RC-Ratio is a cutting-edge approximation algorithm that solves an MINLP problem similar to ours with a theoretical guarantee [60]. Therefore, we implement four baselines derived from this algorithm, alongside a state-of-the-art reinforcement learning-based algorithm for comparison [61].

- *RC-Ratio (RC) [60]*: It greedily selects the task node with the largest *RC-ratio* (i.e., the ratio of sensing utility increase to flying distance increase) to construct the travel routes.
- *Sensing Utility Greedy (UG):* It greedily selects the task node with the maximal incremental sensing utility to construct the travel routes.
- *Scheduling Efficiency Greedy (EG):* It always greedily selects the routes with the maximal scheduling efficiency (i.e., the ratio of sensing utility increase to energy consumption increase) to construct the travel routes.
- *Delivery Time Greedy (TG):* It greedily selects the task node with the minimal incremental delivery time to construct the travel routes.
- *HEM [61]:* A state-of-the-art hierarchical sequence model for solving mixed-integer programs via reinforcement learning.

Note that the baselines RC, UG, EG, and TG adopt the greedy sensing time allocation strategy and construct the routes with *fast nearest neighbour rule* [62].

B. Evaluation on the Overall Performance of DDL

In order to illustrate the overall performance of our *DDL* and five baselines, we plot the results on three metrics under the default settings in Fig. 6. From the comparison, we can make the following observations.

First, in Fig. 6(a), we can find that the average scheduling quality of *DDL* is 0.153, which shows obvious improvement compared with all baselines. Specifically, the average scheduling quality is improved by 46% and 90%, compared with RC and TG, respectively. To illustrate the practical meaning of scheduling quality, we take an improvement of 0.01 in scheduling

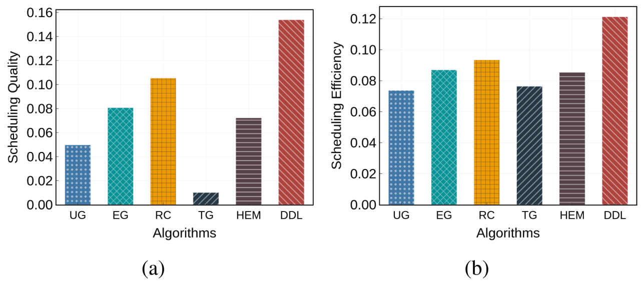


Fig. 6. Overall Performance of *DDL*. (a) Scheduling Quality. (b) Scheduling Efficiency.

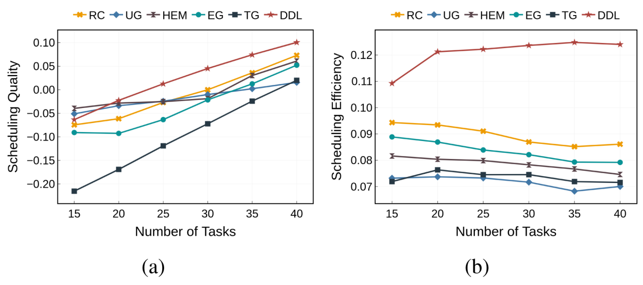


Fig. 7. Performance evaluations of *DDL* in terms of the number of tasks on two metrics. (a) Impact of the number of tasks on scheduling quality. (b) Impact of the number of tasks on scheduling efficiency.

quality as an example. Recall the definition of scheduling quality, which is a weighted sum of the normalized sensing utility and normalized delivery time. Therefore, an improvement of 0.01 in scheduling quality indicates an increase of 800 in sensing utility or a decrease of 224 seconds in delivery time. This validates our idea that combining the strength of DRL and heuristic can help gather more sensing information while spending less extra delivery time. The improvement comes from our algorithm's ability to assign sensing tasks for delivery drones and plan flight routes from a holistic perspective.

Second, for the average scheduling efficiency, *DDL* significantly obtains the best overall performance, as shown in Fig. 6(b). Specifically, *DDL* improves the scheduling efficiency by 30%, 64%, 39%, 59%, and 42%, over RC, UG, EG, TG, and HEM, respectively. This shows that *DDL* has learned the ability to balance energy consumption and sensing utility, which attributes to the integration of reinforcement learning and heuristic method.

C. System Robustness Evaluation

1) Effect the Number of Tasks: We show the impact of the number of tasks on scheduling quality, scheduling efficiency, and route number in Fig. 7. We changed the number of tasks from 15 to 40 with a step size of 5, while leaving other parameters unchanged with their default values. From Fig. 7(a), we can see that *DDL* consistently outperforms all five baselines in terms of scheduling quality. For example, when the number of tasks is 40, *DDL* achieves a scheduling quality of 0.10, compared to RC given the best line, with a 37% improvement. Besides, for all approaches, the scheduling quality improves as the number of

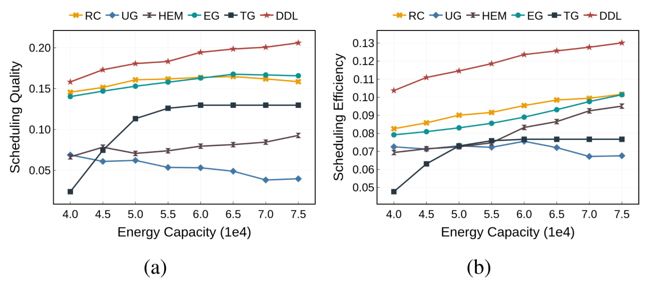


Fig. 8. Performance evaluations of *DDL* in terms of energy capacity on two metrics. (a) Impact of energy capacity on scheduling quality. (b) Impact of energy capacity on scheduling efficiency.

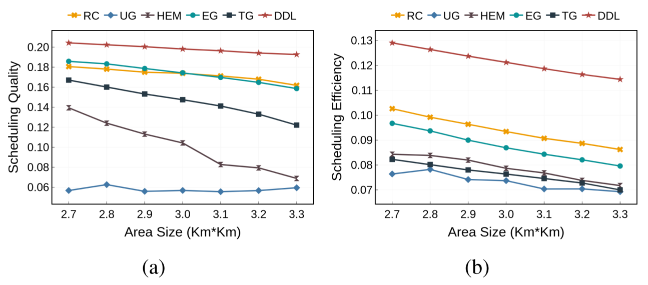


Fig. 9. Performance evaluations of *DDL* in terms of the area size on two metrics. (a) Impact of the area size on scheduling quality. (b) Impact of the area size on scheduling efficiency.

tasks increases. This is because the delivery drones are able to perform more sensing tasks, which enhances their total sensing utility. Increasing the number of sensing tasks reveals that there is a larger search space for scheduling the delivery routes, and selecting the tasks with manually designed rules for each delivery route tends to significantly increase the delivery time. Therefore, the baselines fail to balance the trade-off between delivery time and sensing utility. In contrast, the incorporation of the DRL successfully guides our algorithm to assign an adequate number of sensing tasks for each delivery drone, which helps the algorithm to obtain more sensing utility in scheduling.

The effects of task number on scheduling efficiency are also investigated as shown in Fig. 7(b). While the scheduling efficiency of *DDL* keeps improving with the increase of sensing tasks, this metric of other baselines decreases instead. Recall the definition of scheduling efficiency, which is the ratio of sensing utility and energy consumption. Since UG and TG focus more on a single metric, their performance on scheduling efficiency is not satisfactory enough. RC and EG perform slightly better than other baselines, because they rely on more indicators when it comes to decision of the search direction. Without effectively modeling the cooperation between different routes, these baselines schedule the delivery drones with a considerable increase in flying time, which lead to large energy consumption. When the extension in delivery time exceeds the improvement of sensing utility, the scheduling efficiency decreases significantly. It is noticed that our *DDL* outperforms all baselines with variant numbers of tasks, which shows the obvious advantage of our design.

2) Effect of the Energy Capacity: Fig. 8 shows the impact of the energy capacity for delivery drones. We vary the energy capacity from 40,000 KJ to 75,000 KJ with a step of 5,000 KJ. From Fig. 8(a)–(c), we can make the following observations.

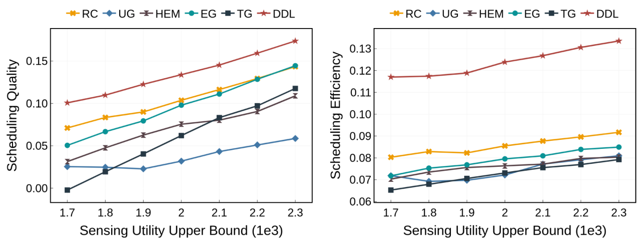
Fig. 8(a) illustrates that *DDL* outperforms all five baselines in terms of scheduling quality. For example, *DDL* reaches a scheduling quality of 0.21 when the energy capacity is 75,000 KJ, beating the best baseline EG by 24%, whose scheduling quality is just 0.16. On average, *DDL* outperforms RC, UG, EG, TG, and HEM methods by 17%, 270%, 18%, 127%, and 138% in terms of scheduling quality, respectively. Besides, a large energy capacity leads to high scheduling quality for most of the methods at the beginning, but when the energy capacity reaches more than 70,000 KJ, we discover that the impact on scheduling quality gradually diminishes. This may be explained by the fact that with the growth of energy capacity, the delivery drones have more remaining energy to hover and collect sensing data. However, when the energy capacity exceeds a threshold, all the sensing tasks are completed (i.e., the task utilities reach their upper bound), thus the scheduling quality plateaus.

From Fig. 8(b), we see that the scheduling efficiency of *DDL* increases monotonically as the energy capacity grows. *DDL* shows consistent advantage of scheduling efficiency with variant energy capacities of delivery drones. Especially, *DDL* achieves up to a scheduling efficiency of 0.12 when the energy capacity is $55,000KJ$, which is 29% higher than the best baseline RC. The comparison also indicates that *DDL* captures the general structure of this NP-hard combinatorial optimization problem, which enabled it to consistently assign the most appropriate number of drones to perform the task.

3) Effect of the Area Size: Next, we show the impact of area size in Fig. 9. We consider the situations that the service stations occupy a size of area from 2.7 $km \times$ 2.7 km to 3.3 $km \times$ 3.3 km. The primary observations are summarized as follows.

Fig. 9(a) reveals how the area size affects the scheduling quality of different methods. First, From Fig. 9(a), we can observe that *DDL* exceeds the performance of all competing baselines in the terms of scheduling quality. Especially, when the area size is 2.7 $km \times$ 2.7 km, *DDL* reaches the best scheduling quality. Second, it is noticed that all methods show a decline in scheduling quality with the growth of area size. The reason behind this phenomenon is that the larger area will lead to a longer flying distance, which indicates the delivery time increase. However, even in these cases, *DDL* outperforms all the baseline approaches by more than 14% percent in terms of scheduling quality. Since the sensing tasks are assigned to several delivery drones, a larger area size implies a smaller number of tasks in the unit area in average. Therefore, to improve the scheduling efficiency, it is significant to determine the number of tasks to execute with the consideration of the area size.

Fig. 9(b) depicts the scheduling efficiency of all methods when varying the area size. The scheduling efficiency trend of all methods is similar to the scheduling quality in Fig. 9(a). This is due to the same reason mentioned in the previous paragraph. By utilizing the coordination between the delivery drones to optimize the routing scheme, *DDL* beats the best baseline RC in terms of both scheduling quality and scheduling efficiency. This comparison result again validates the effectiveness of the combination of DRL and the heuristic method.



(a) (b)

Fig. 10. Performance evaluations of *DDL* in terms of the sensing utility upper bound on three metrics. (a) Impact of the sensing utility upper bound on scheduling quality. (b) Impact of the sensing utility upper bound on scheduling efficiency.

4) Effect of the Task Utility Upper Bound: We present the impact of the sensing utility upper bound in Fig. 10. The sensing utility upper bound is changed from 1,700 to 2,300. We can make the following observations from the experimental results.

The effects of sensing utility upper bound on scheduling quality are investigated as shown in Fig. 10(a). First, comparing the scheduling quality of different methods, we see that *DDL* consistently shows better performance than baselines. This proves that our algorithm effectively allocates reasonable sensing time for the tasks that bring higher sensing utility. Second, with a higher sensing utility upper bound, the scheduling quality of all methods increases. This is because the increase in sensing utility upper bound allows the drones to allocate more sensing time to collect sensor data, which results in total sensing utility improvement.

Fig. 10(b) displays that the scheduling efficiency of *DDL* and baselines rise with the increase in sensing utility upper bound. Each delivery drone aims to collect as much data as possible when the constraints remain satisfied. As a result, the total sensing utility rises and the scheduling efficiency naturally rises. We can see that *DDL* always has the highest scheduling efficiency while comparing different sensing utility upper bound. For example, when the sensing utility upper bound is 2000, *DDL*'s scheduling efficiency is 0.12, which is 29.7% and 39.4% higher than RC and EG, respectively. The reason behind this trend may be that *DDL* captures the relationship between sensing utility and energy consumption, thus it can complete tasks with less energy consumption.

D. System Micro-Benchmark

1) Impact of Coefficients in the Objective Function: In order to check how the model coefficient α in (9) affect the performance, we plot α with variant values in Fig. 11. The value α is set from 0 to 1 with a step of 0.1. Note that since α indicates the trade-off between the sensing performance and delivery performance, it could be adjusted according to the practical requirements. As shown in Fig. 11(a), the weighted sensing utility consistently increases with the increase of the α value, while the weighted delivery time keeps decreasing. This is because that larger α value biases the model to focus more on the sensing utility, thereby the penalty for delivery time decreases. Meanwhile, we discovered an interesting phenomenon in Fig. 11(b) that as the α value rises from 0.5 to 1, the scheduling efficiency begins to degrade. This implies that a suitable model coefficient of *DDL* will contribute to the overall performance. When we assign equal importance to the dual objectives, the scheduling efficiency is more close to the optimum. In summary, the evaluation result shows that *DDL* could respond correctly to the parameter settings in the objective function.

2) Computational Efficiency: In order to assess the computational efficiency and benefits of implementing the reinforcement learning algorithm, we compare the average runtime (using a logarithmic scale) for the scheduling decision process depicted in Fig. 12. We mainly focus on the run time under different numbers of tasks, because the combination choices of decision variables grow exponentially with the number of tasks grows, which incurs a significant increase in computational complexity. Other parameters are set by default. Note that in adherence to preceding studies, the RC, UG, EG, and TG baselines have all been implemented on a CPU. Additionally, our DDL and the novel HEM baseline, designed employing deep learning techniques, have been executed on a GPU. We can see that when the number of tasks is less than 30, all baselines have a stable performance in terms of run time. However, as the number of tasks increases, the run time of EG and RC increases dramatically. Especially, when the task number is 60, the run time of EG reaches about 100 seconds, which is nearly 100 times longer than *DDL* on average. The reason behind is that both EG and RC rely on the *fast nearest neighbour rule* [62] to construct flight routes, and the computation complexity can be expressed as $\mathcal{O}(\mathcal{R} \cdot r(n)^2)$, where $\mathcal{R}$ is the number of drones and $r(n)$ is the number of tasks on route r, respectively. On the contrary, the computation complexity of constructing routes in *DDL* is $\mathcal{O}(T)$, where T is the maximal number of rewriting steps. Overall, *DDL* shows advantages in run time over all baselines when dealing with a large-scale problem.

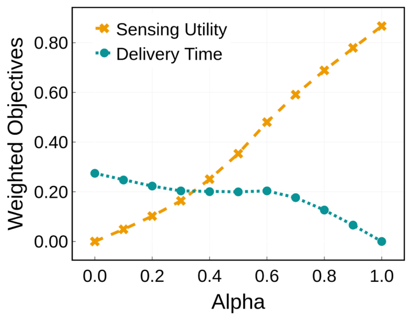
VII. Conclusion and Discussion

A. Conclusion

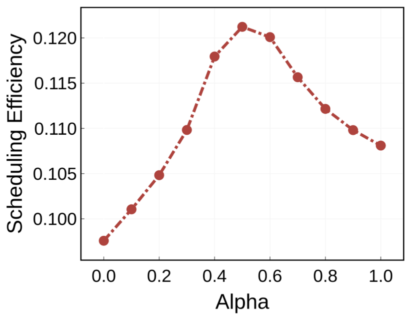
In this paper, we investigate how to perform urban sensing with delivery drones with the practical constraints. We first propose a delivery drone-based sensing system *DDL* and formulate a mixed-integer non-linear programming problem (MINLP) that jointly optimizes the sensing utility and delivery time. Then an efficient solution is presented which integrates: (1) a DRL model that guides the task assignment and route planning of delivery drones; (2) a heuristic model to allocate sensing time for sensing tasks. This approach disentangles the intricacies of the optimization search process and substitutes direct computation with rapid approximations. Evaluation results show that *DDL* improves the scheduling quality by at least 46% on average, compared with the state-of-the-art baselines. Furthermore, our proposed method could effectively improve the computational efficiency, which is up to 98 times higher than the best baseline.

B. Discussion

We discuss the potential improvements that can be explored as future research directions.



(a) Impact of α on delivery time/sensing utility.



(b) Impact of α on scheduling efficiency.

Fig. 11. Performance evaluations of *DDL* in terms of α.

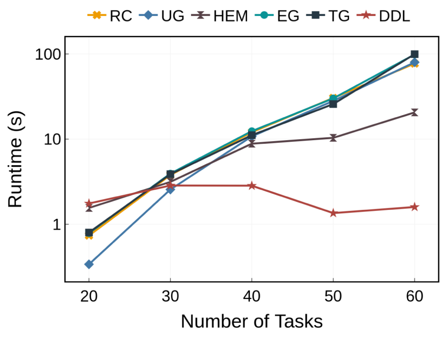


Fig. 12. Approximate run time of *DDL* and baselines.

Multiple Destinations: DDL currently does not account for multiple destinations in a delivery team. In a typical drone delivery system, multiple delivery teams (DTs) collaborate to cover an area. Each drone in a DT maintains a fixed destination throughout its operational period. Therefore, our assumption of a unique destination for each drone aligns with the design and operation of these delivery systems. If considering a solution for the entire delivery system with multiple destinations, we suggest applying our DDL algorithm separately for each individual DT.

Dynamic Environment Adaptation: Currently, our *DDL* system requires advance provision of sensing tasks due to government regulations regarding flying area management. In practice, multiple airlines share the aerial area, necessitating pre-approved drone flight routes according to strict regulations and protocols. Consequently, real-time online planning for drone delivery operations faces feasibility constraints. Nevertheless, our system can be easily modified to support online scheduling of delivery drones, akin to the Job Scheduling Problem (JSP). Existing research has explored DRL algorithm designs for JSP [63], which could potentially inform the development of an online algorithm for our delivery drone system.

Acknowledgment

The authors acknowledge the support from the Tsinghua Shenzhen International Graduate School-Shenzhen Pengrui Endowed Professorship Scheme of Shenzhen Pengrui Foundation. We acknowledge the support from Meituan company.

References

[1] H. Wang et al., "TransformLoc: Transforming MAVs into mobile localization infrastructures in heterogeneous swarms," in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 1101–1110, doi: 10.1109/INFOCOM52122.2024.10621375.

[2] J. Xu et al., "SwarmMap: Scaling up real-time collaborative visual slam at the edge," in *Proc. USENIX NSDI*, 2022, pp. 977–993.

[3] Z. Li, F. Man, X. Chen, B. Zhao, C. Wu, and X. Chen, "TRACT: Towards large-scale crowdsensing with high-efficiency swarm path planning," in *Proc. Adjunct Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, 2022, pp. 409–414.

[4] J. Xu et al., "Edge assisted mobile semantic visual SLAM," in *Proc. IEEE INFOCOM*, 2020, pp. 1828–1837.

[5] Y. Ding et al., "Online edge learning offloading and resource management for UAV-assisted MEC secure communications," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 54–65, Jan. 2023.

[6] W. Mei and R. Zhang, "Uplink cooperative NOMA for cellular-connected UAV," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 644–656, Jun. 2019.

[7] L. Ding, D. Zhao, M. Cao, and H. Ma, "When crowdsourcing meets unmanned vehicles: Toward cost-effective collaborative urban sensing via deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12150–12162, Aug. 2021.

[8] S.A Prophesee, "Dji inspire 3," 2024. [Online]. Available: https://www.dji.com/cn/inspire-3?site=brandsite&from=landing_page

[9] Y. Ding et al., "A city-wide crowdsourcing delivery system with reinforcement learning," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 3, pp. 1–22, 2021.

[10] R. Liao, "How meituan is redefining food delivery in China with drones," 2021. [Online]. Available: https://techcrunch.com/2021/12/29/meituan-food-drone-delivery-china

[11] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 606–617, Jun. 2011.

[12] A. Landstrom and M. J. Thurley, "Morphology-based crack detection for steel slabs," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 7, pp. 866–875, Nov. 2012.

[13] Z. Ning, H. Chen, X. Wang, S. Wang, and L. Guo, "Blockchain-enabled electrical fault inspection and secure transmission in 5G smart grids," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 1, pp. 82–96, Jan. 2022.

[14] X. Chen, A. Purohit, C. R. Dominguez, S. Carpin, and P. Zhang, "DrunkWalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms," in *Proc. 13th ACM Conf. Embedded Networked Sensor Syst.*, 2015, pp. 295–308.

[15] D. Zhao, M. Cao, L. Ding, Q. Han, Y. Xing, and H. Ma, "DroneSense: Leveraging drones for sustainable urban-scale sensing of open parking spaces," in *Proc. -IEEE Conf. Comput. Commun.*, 2022, pp. 1769–1778.

[16] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem," *Math. Program.*, vol. 94, pp. 343–359, 2003.

[17] X. Chen et al., "Deliversense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning," in *Adjunct Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, 2022, pp. 403–408.

[18] H. Wang, X. Chen, Y. Cheng, C. Wu, F. Dang, and X. Chen, "H-swarmloc: Efficient scheduling for localization of heterogeneous mav swarm with deep reinforcement learning," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, 2022, pp. 1148–1154.

[19] J. Ren, Y. Xu, Z. Li, C. Hong, X.-P. Zhang, and X. Chen, "Scheduling UAV swarm with attention-based graph reinforcement learning for ground-to-air heterogeneous data communication," in *Proc. Adjunct Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, 2023, pp. 670–675.

[20] X. Chen et al., "H-DrunkWalk: Collaborative and adaptive navigation for heterogeneous mav swarm," *ACM Trans. Sensor Netw.*, vol. 16, no. 2, pp. 1–27, 2020.

[21] C. Zhao et al., "SmoothLander: A quadrotor landing control system with smooth trajectory guarantee based on reinforcement learning," in *Proc. Adjunct Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, 2023, pp. 682–687.

[22] M. Torabbeigi, G. J. Lim, and S. J. Kim, "Drone delivery schedule optimization considering the reliability of drones," in *Proc. IEEE IEEE Int. Conf. Unmanned Aircr. Syst.*, 2018, pp. 1048–1053.

[23] K. Wei et al., "High-performance UAV crowdsensing: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18487–18499, Oct. 2022.

[24] M. T. Rashid, D. Y. Zhang, and D. Wang, "SocialDrone: An integrated social media and drone sensing system for reliable disaster response," in *Proc. -IEEE Conf. Comput. Commun.*, 2020, pp. 218–227.

[25] Y. Yang, Z. Hu, K. Bian, and L. Song, "ImgSensingNet: UAV vision guided aerial-ground air quality sensing system," in *Proc. -IEEE Conf. Comput. Commun.*, 2019, pp. 1207–1215.

[26] X. Chen et al., "Design experiences in minimalistic flying sensor node platform through sensorfly," *ACM Trans. Sensor Netw.*, vol. 13, no. 4, pp. 1–37, 2017.

[27] X. Chen, X. Xu, X. Liu, H. Y. Noh, L. Zhang, and P. Zhang, "Hap: Fine-grained dynamic air pollution map reconstruction by hybrid adaptive particle filter," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. CD-ROM*, 2016, pp. 336–337.

[28] Y. Wang and G. Cao, "Achieving full-view coverage in camera sensor networks," *ACM Trans. Sensor Netw.*, vol. 10, no. 1, pp. 1–31, 2013.

[29] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[30] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *IEEE Trans. Parallel Distrib. Syst*, vol. 25, no. 12, pp. 3190–3200, Dec. 2014.

[31] K. Han, C. Zhang, J. Luo, M. Hu, and B. Veeravalli, "Truthful scheduling mechanisms for powering mobile crowdsensing," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 294–307, Jan. 2016.

[32] Z. Lai, Y. Fu, and J. Zhang, "Hyperspectral image super resolution with real unaligned RGB guidance," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 18, 2024, doi: 10.1109/TNNLS.2023.3340561.

[33] M. Li, Y. Fu, T. Zhang, and G. Wen, "Supervise-assisted self-supervised deep-learning method for hyperspectral image restoration," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 09, 2024, doi: 10.1109/TNNLS.2024.3386809.

[34] J. Xu et al., "Taming event cameras with bio-inpired architecture and algorithm: A case for drone obstacle avoidance," in *Proc. ACM MobiCom*, 2023, pp. 977–993.

[35] C. Xiang et al., "Reusing delivery drones for urban crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2972–2988, May 2023.

[36] X. Tao and A. S. Hafid, "Trajectory design in UAV-aided mobile crowdsensing: A deep reinforcement learning approach," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[37] M. N. Kritikos and P. Z. Lappas, "Computational intelligence and combinatorial optimization problems in transportation science," in *Proc. Adv. Core Comput. Sci.-Based Technol.*, pp. 325–367, 2021.

[38] Y. Hou, Z. Cao, and S. Yang, "Cloud intelligent logistics service selection based on combinatorial optimization algorithm," *J. Européen des Systèmes Automatisés*, vol. 52, no. 1, pp. 73–78, 2019.

[39] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. North Chelmsford, MA, USA: Courier Corporation, 1998.

[40] R. Zhang, A. Prokhorchuk, and J. Dauwels, "Deep reinforcement learning for traveling salesman problem with time windows and rejections," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[41] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, 2021.

[42] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[43] D. Silver et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[44] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.-Volume 2*, 2015, pp. 2692–2700.

[45] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác, "Reinforcement learning for solving the vehicle routing problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 9861–9871.

[46] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6351–6361.

[47] H. Lu, X. Zhang, and S. Yang, "A learning-based iterative method for solving vehicle routing problems," in *Proc. Int. Conf. Learn. Representations*, 2019.

[48] R. Wang et al., "A bi-level framework for learning to solve combinatorial optimization on graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 21453–21466.

[49] T. Zhang, Y. Fu, and C. Li, "Hyperspectral image denoising with realistic data," in *Proc. Brit. Mach. Vis. Conf.*, 2021, pp. 2248–2257.

[50] X. Chen et al., "ASC: Actuation system for city-wide crowdsensing with ride-sharing vehicular platform," in *Proc. 4th Workshop Int. Sci. Smart City Operations Platforms Eng.*, 2019, pp. 19–24.

[51] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, "iLOCuS: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1831–1847, Aug. 2020.

[52] X. Chen et al., "PAS: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3719–3734, May 2020.

[53] G. Ghiani, G. Laporte, and R. Musmanno, *Introduction to Logistics Systems Management* (Wiley Series in Operations Research and Management Science Series). Chichester, U.K.: Wiley, 2013.

[54] X. Chen et al., "PGA: Physics guided and adaptive approach for mobile fine-grained air pollution estimation," in *Proc. ACM Int. Joint Conf. Int. Symp. Pervasive Ubiquitous Comput. Wearable Comput.*, 2018, pp. 1321–1330.

[55] Y. Liu, X. Liu, F. Man, C. Wu, and X. Chen, "Fine-grained air pollution data enables smart living and efficient management," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, 2022, pp. 768–769.

[56] A. Sanjab, W. Saad, and T. Başar, "Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[57] X. Chen and Y. Tian, "Learning to perform local rewriting for combinatorial optimization," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 6281–6292.

[58] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[59] F. Shan, J. Luo, R. Xiong, W. Wu, and J. Li, "Looking before crossing: An optimal algorithm to minimize UAV energy by speed scheduling with a practical flight energy model," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1758–1767.

[60] T. Wu et al., "Joint sensor selection and energy allocation for tasks-driven mobile charging in wireless rechargeable sensor networks," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11505–11523, Dec. 2020.

[61] Z. Wang et al., "Learning cut selection for mixed-integer linear programming via hierarchical sequence model," in *Proc. 11th Int. Conf. Learn. Representations*, 2023, [Online]. Available: https://openreview.net/forum?id=Zob4P9bRNcK

[62] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis II, "An analysis of several heuristics for the traveling salesman problem," *SIAM J. Comput.*, vol. 6, no. 3, pp. 563–581, 1977.

[63] C. Zhang, W. Song, Z. Cao, J. Zhang, P. S. Tan, and X. Chi, "Learning to dispatch for job shop scheduling via deep reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1621–1632.

Xuecheng Chen received the B.E. degree from the Department of Intelligence Engineering, Sun Yat-sen University, Guangzhou, China, in 2021. He is currently working toward the Ph.D. degree with the Department of Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, China. His research interests include mobile computing, multirobot systems, and cyber-physical systems.

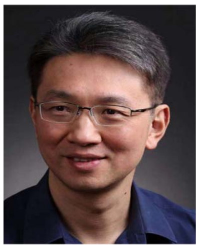
Haoyang Wang (Graduate Student Member, IEEE) received the B.E. degree from the School of Computer Science and Engineering, Central South University, Hunan, China, in 2022. He is currently working toward the Ph.D. degree with Tsinghua Shenzhen International Graduate School, Tsinghua University, Beijing, China. His research interests include AIoT and mobile computing.

Yuhan Cheng received the B.E. degree from the School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, China, in 2022. He is currently working toward the M.S. degree with Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. His research interests encompass cyber-physical systems and mobile sensing.

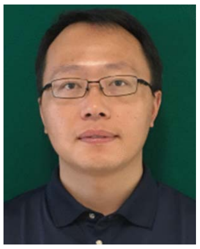
Haohao Fu received the B.A. degree from the School of L&S, University of California at Berkeley, Berkeley, CA, USA, in 2022. He is currently working toward the Ph.D. degree with the Department of Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, China. His research interests include information theory, multimodal learning, and transfer learning.

Yuxuan Liu received the B.S. degree from the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, in 2021. He is currently working toward the M.S. degree with Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, China. His research interests include mobile sensing, environmental sensing, and cyber-physical systems.

Fan Dang (Member, IEEE) received the B.E. and Ph.D. degrees in software engineering from Tsinghua University, Beijing, China, in 2013 and 2018, respectively. He is currently a Research Assistant Professor with the Global Innovation Exchange, Tsinghua University, Beijing. His research interests include the industrial Internet, edge computing, and mobile security. He is a member of CCF and ACM.



Yunhao Liu (Fellow, IEEE) received the B.E. degree from the Department of Automation, Tsinghua University, Beijing, China, in 1995, and the M.A. degree from Beijing Foreign Studies University, Beijing, in 1997, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, East Lansing, MI, USA, in 2003 and 2004, respectively. He is currently a Professor with the Department of Automation and the Dean of the Global Innovation Exchange, Tsinghua University, Beijing. His research interests include Internet of Things, wireless sensor networks, indoor localization, the Industrial Internet, and cloud computing. He is a Fellow of CCF and ACM.



Jinqiang Cui received the B.S. and M.S. degrees in mechatronic engineering from Northwestern Polytechnical University, Xi'an, China, in 2005 and 2008, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore (NUS), Singapore, in 2015. His Ph.D. research focuses on the navigation of unmanned aerial vehicles in GPS-denied environments, especially forest. From 2015 to 2016, he was a Research Scientist with the Control Science Group, Temasek Laboratories, NUS. He is currently the CTO of a startup company specializing in the development of UAV navigation systems.

Xinlei Chen (Member, IEEE) received the B.E. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2009 and 2012, respectively, and the Ph.D. degree in electrical engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2018. In 2021, he is currently an Assistant Professor with the Tsinghua Shenzhen International Graduate School, Shenzhen, China. From 2018 to 2020, he was a Postdoctoral Research Associate with the Department of Electrical Engineering, Carnegie Mellon University. His research interests include AIoT, pervasive computing, and cyber physical system.