DeliverSense: Efficient Delivery Drone Scheduling for Crowdsensing with Deep Reinforcement Learning

Xuecheng Chen
Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
Shenzhen, China
chenxc21@mails.tsinghua.edu.cn

Haoyang Wang
Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
haoyang-22@mails.tsinghua.edu.cn

Zuxin Li
Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
Shenzhen, China
lizx21@mails.tsinghua.edu.cn

Wenbo Ding
Shenzhen International Graduate School, Tsinghua University
RISC-V International Open Source Laboratory
Shenzhen, China
ding.wenbo@sz.tsinghua.edu.cn

Fan Dang
Global Innovation Exchange, Tsinghua University
Beijing, China
dangfan@tsinghua.edu.cn

Chenye Wu
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
chenyewu@year.net

Xinlei Chen*
Shenzhen International Graduate School, Tsinghua University
Peng Cheng Laboratory
Shenzhen, China
chen.xinlei@sz.tsinghua.edu.cn

ABSTRACT

Delivery drones provide a promising sensing platform for Mobile Crowdsensing (MCS) due to their high mobility and large-scale deployment. However, due to limited battery lifetime and available resources, it is challenging to schedule large-scale delivery drones to derive both high crowdsensing and delivery performance, which is a highly complicated optimization problem with several coupled decision variables. In this paper, we first formalize the delivery drones scheduling problem as a mixed-integer nonlinear programming problem with both sensing and delivery utilities as dual objectives. Then we propose a novel framework *DeliverSense* with a reinforcement learning-based efficient solution, which decouples the highly complicated optimization search process and replaces the heavy computation via fast approximation. Evaluation results compared with state-of-the-art baseline show that *DeliverSense* improves the total utility by 13% and 23% on average under various energy budgets and numbers of selected routes, respectively. More importantly, our proposed method achieves much lower computational complexity which is nearly 3 times lower than the baseline.

*Xinlei Chen is the corresponding author.

UbiComp/ISWC '22 Adjunct, September 11–15, 2022, Cambridge, United Kingdom

ACM ISBN 978-1-4503-9423-9/22/09…$15.00
https://doi.org/10.1145/3544793.3560412

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Computing methodologies → Multi-agent planning**.

KEYWORDS

Mobile Crowdsensing; Large-Scale Scheduling; Reinforcement Learning

ACM Reference Format:
Xuecheng Chen, Haoyang Wang, Zuxin Li, Wenbo Ding, Fan Dang, Chenye Wu, and Xinlei Chen. 2022. DeliverSense: Efficient Delivery Drone Scheduling for Crowdsensing with Deep Reinforcement Learning. In *Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp/ISWC '22 Adjunct), September 11–15, 2022, Cambridge, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3544793.3560412

1 INTRODUCTION

In the past decade, with the rapid growth of 5G and the popularization of mobile devices (e.g., smartphones and wearable devices), Mobile Crowdsensing (MCS) has been an attractive paradigm to collectively perceive, collect and exchange city-wide data from the surrounding environments [7, 15, 20, 35]. By fusing and analyzing the collected data [39, 40], MCS applications are able to infer phenomena of common interest, such as air pollution and traffic congestion [8, 17, 31, 33, 34].

Delivery drones provide a novel and promising sensing platform for crowdsensing [4]. Drone delivery services have attracted considerable attention due to their cost-effectiveness and timeliness [29]. Giant companies such as Amazon [1], Walmart [19], UPS [27]

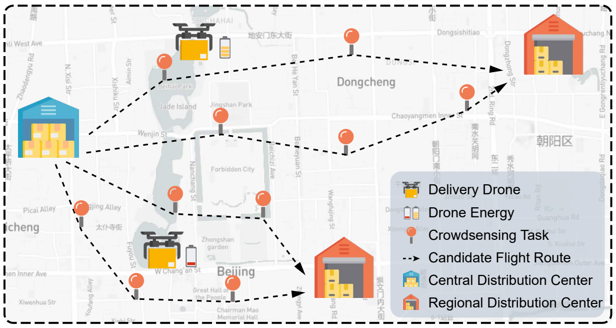


Figure 1: Illustration of hiring delivery drones to perform crowdsensing and delivery simultaneously.

and DHL [12] start widely deploying delivery drones to deliver city-wide packages (e.g., medical, food). As illustrated in Fig. 1, when delivery drones are hired to perform crowdsensing and delivery tasks simultaneously, we benefit from the following advantages of delivery drones [16]: (1) **High Mobility**: The three-dimensional mobility enables delivery drones to quickly deploy without considering congestion [3, 5]. (2) **Large-Scale**: Delivery service stations are distributed densely in the city [23], providing opportunities for delivery drones to cover almost anywhere. (3) **Low Cost**: Compared to other dedicated sensing platforms [9, 30], delivery drones collect urban data while delivering packages with little extra purchase and maintenance cost.

Researchers have proposed different methods to schedule drones as new crowdsensing platforms. Rashid *et al.* [22] designed a sensing framework that integrates social media and drones for reliable disaster response. Yang *et al.* [37] proposed a vision guided aerial-ground sensing system for air quality monitoring and forecasting. However, the majority of research works focused on scheduling dedicated drones [18, 26], instead of taking advantage of the delivery drones. Though Xiang *et al.* [32] started to use delivery drones for crowdsensing, their manually designed heuristic solution may suffer from heavy computation and unsatisfactory algorithmic decisions when scaled to more variables. Another related study [25] applied a reinforcement learning algorithm to schedule delivery drones for crowdsensing, but they simplify the drones' energy consumption model. We argue that a more efficient solution is required, and more practical factors including energy budget and delivery weight need to be considered simultaneously.

The research question this paper tries to answer is: **how to schedule large-scale delivery drones to derive high crowdsensing performance while maintaining high delivery performance?** The major challenges are two folds. *First, crowdsensing usually has conflicting goals against package delivery.* Package delivery requires drones to follow the shortest routes to achieve high efficiency, while crowdsensing prefers drones to collect sufficient information scattered in the city. Such inconsistency of goals makes it difficult to derive high sensing performance without sacrificing the delivery performance. The integration of key practical factors, energy budget and delivery weight further complicates the problem. *Second, it is hard to achieve optimality and efficiency simultaneously.* The large-scale delivery drone scheduling problem involves a large number of drones, candidate routes and sensing tasks. It is impossible to directly use exhaustive search or other traditional optimization methods to achieve the optimal solution in practical time.

This paper proposes **DeliverSense**, an efficient delivery drones scheduling framework, aiming at jointly optimizing the delivery route selection and crowdsensing time allocation with consideration of practical factors. To address the first challenge, we formalize the delivery drones scheduling problem as a mixed-integer non-linear programming problem (MINLP), with sensing utility and delivery utility as dual objectives. In the formalization, we use an adjustable trade-off coefficient to balance the importance of two objectives, considering energy budget, energy consumption, and available drones as constraints. To address the second challenge, we exploit the structural characteristics of the problem. Specifically, we design the *Coordinator* model to decouple the complicated optimization search process into two parts: *RouteSelector* selects delivery routes, *Timeallocator* allocates sensing time. Then we utilize reinforcement learning to iteratively improve the scheduling policy toward the optimum.

The main contributions of this paper are listed as follows:

- Design a framework that jointly optimizes the delivery utility and sensing utility, considering several practical factors simultaneously.
- Propose a reinforcement learning-based efficient solution, which learns the structural characteristics of the delivery drone scheduling problem and replaces the heavy computation.
- Evaluate our solution with comprehensive experiments based on data collected from a real-world implemented system.

The remainder of the paper is organized as follows: §2 presents problem definition. §3 introduces our framework overview, key components and corresponding algorithms. §4 demonstrates experimental results. §5 concludes the paper.

2 PROBLEM DEFINITION

2.1 Background and Definitions

Fig. 1 illustrates the architecture of a typical two-level delivery system with a tree structure [14], composed of Central Distribution Center (CDC) and Regional Distribution Centers (RDC). When drones deliver packages from CDCs to RDCs, they can slightly modify their flying trajectories to conduct crowdsensing tasks. We define other key concepts as follows.

Participant: A participant is a delivery drone equipped with multiple sensors, which is denoted as p and belongs to a specific Delivery Team (DT) i. Each participant flies between CDC and RDC and collects data along the given flight route.

Delivery Team (DT): We define the drones for the same delivery pair between CDC and RDC as a Delivery Team (DT). Assume we have I DT and each has up to n_i drones. Let J denote the set of candidate routes and each route j is assigned to a DT i. The direct flying distance from the CDC to the RDC of DT i is d_{i0}, while the flying distance with crowdsensing on route j is d_{ij}. Each delivery drone (i.e., participant) of DT i on route j conducts crowdsensing tasks with the maximum energy E_{ij}.

Delivery Utility: Delivery utility is a measurement of the delivery performance. The delivery weight on the route j of DT i is represented as a discrete variable w_{ij} in kilogram. Generally, the delivery profit depends on the weight of the packages delivered by the drone [2, 38]. Therefore, the delivery utility is set as βw_{ij}, where parameter β is the unit utility of delivery weight.

Crowdsensing Task & Sensing Utility: We define there are K tasks distributed in different locations. Each task S_k is allocated to a specific delivery route j of DT i with sensing time t_{ij}^k. Though many application-specific physical factors(e.g. sensing distance, unpredictable wind) affect the sensing utility, a longer sensing time to execute a sensing task can bring about a better sensing effect in general [10, 11, 37]. Without loss of generality, we define b_k as the benefit that a delivery drone spends unit sensing time on the task S_k, and B_k as the utility upper bound. Then, the sensing utility model can be expressed as:

$$T(t_{ij}^k) = \min(b_k t_{ij}^k, B_k). \tag{1}$$

Energy Consumption: We adopt an energy consumption model which considers the impact factors in detail, including the delivery weight of the drone, the flying time, and the hovering time [32]. The mathematical form of the energy consumption c_{ij} for a delivery drone is defined as follows:

$$c_{ij} = P_h\left(w_{ij}\right) \sum_{k\in K} t_{ij}^k + P_f\left(w_{ij}\right) \frac{d_{ij}}{v}, \tag{2}$$

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

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

where $P_h(w_{ij})$ and $P_f(w_{ij})$ denote the power of hovering and flying with the delivery weight w_{ij}, respectively. ρ_0^{h}, ρ_1^{h}, ρ_0^{f} and ρ_1^{f} are the environment-dependent model parameters. The flying speed v is set as a constant consistent with existing works [21, 26].

2.2 Optimized Objectives and Constrains

Motivated by the requirements of crowdsensing and delivery companies, our problem considers two kinds of decision variables. The binary decision variable $x_{ij} \in \{0, 1\}$ represents whether the route j of DT i is selected, while the continuous decision variable t_{ij}^k denotes the sensing time allocated to task S_k.

Goal-1: Maximize the delivery utility. Our first goal is to select delivery routes to maximize the total delivery utility, which is formulated as:

$$D\left(w_{ij}\right) = \max_{\mathbf{x}} \sum_{i\in I}\sum_{j\in J} x_{ij}\beta w_{ij}. \tag{5}$$

Goal-2: Maximize the sensing utility. Our second goal of the total sensing utility optimization can be mathematically expressed as:

$$Q\left(t_{ij}^k\right) = \max_{\mathbf{x},\mathbf{t}} \sum_{i\in I}\sum_{j\in J} x_{ij} T(t_{ij}^k). \tag{6}$$

Constraints: The constraints of our problem lie in two aspects. First, the available resources of each DT are limited, including the number of drones and their energy (i.e., battery capacity). Second, the delivery drone company usually determines a total energy budget ρ for drones, which refers to the total extra energy cost due to performing crowdsensing tasks, compared with delivering the package directly.

2.3 Problem Formulation

To optimize both the delivery utility and sensing utility with limited resources, we give the mathematical formulation of the delivery drone scheduling problem as:

$$\max_{\mathbf{x},\mathbf{t}} \sum_{i\in I}\sum_{j\in J} x_{ij}\left(\alpha \sum_{k\in K} Q\left(t_{ij}^k\right) + (1-\alpha) D\left(w_{ij}\right)\right), \tag{7}$$

$$\text{s.t. } x_{ij} \in \{0,1\}, t_{ij}^k \geq 0, \forall i \in I, \forall j \in J, \forall k \in K, \tag{8}$$

$$\sum_{j\in J} x_{ij} \leq n_i, \forall i \in I, \tag{9}$$

$$\sum_{i\in I} x_{ij} \leq 1, \forall j \in J, \tag{10}$$

$$c_{ij} \leq E_{ij}, \forall i \in I, \forall j \in J, \tag{11}$$

$$\Delta e_{ij}^f = P_f\left(w_{ij}\right)\left(d_{ij} - d_{i0}\right), \forall i \in I, \forall j \in J, \tag{12}$$

$$\Delta e_{ij}^h = P_h\left(w_{ij}\right) \sum_{k\in K} t_{ij}^k, \forall i \in I, \forall j \in J, \tag{13}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ij}(\Delta e_{ij}^f + \Delta e_{ij}^h) \leq \rho, \tag{14}$$

where α is a trade-off coefficient defined according to the specific requirements. Constraint (9) ensures the number of participants in each DT to perform tasks does not exceed the maximum; Constraint (10) enforces that each route can only be assigned to at most one DT; Constraint (11) states the available energy of each drone is finite due to the limited battery; Constraints (12)-(14) ensure that the total extra energy cost due to crowdsensing would not exceed the energy budget ρ. Decision variables involved in this problem, including both discrete and continuous types, are highly coupled, making the joint optimization problem a NP-hard Mixed Integer Non-Linear Programming Problem (MINLP).

3 FRAMEWORK DESIGN

3.1 Framework Architecture

As the objectives and constraints are highly coupled, it is computationally intractable to solve the problem in practical applications. Thus, we aim at searching for a near-optimal solution in practical time instead of pursuing the exact scheme. Assuming we have a selected routes set X which has satisfactory delivery utility and does not violate the constraints (9)-(10). Then, the sensing utility optimization could be transformed to find the best sensing time allocation decision given X. Though these two optimizations are coupled with each other, we can take turns optimizing them iteratively until convergence. Based on this idea, we decouple the searching process into two parts, and leverage reinforcement learning to replace the heavy computation via fast approximation. Fig. 2 shows the overall architecture of the solution.

The **Drone Delivery Company** provides candidate flight routes and a large number of delivery drones equipped with various sensors. Both the energy budget and available resources are determined by the drone delivery company.

The **Data Request End** requests and analyzes the sensing data [13, 28]. The data request end specifies the desired geographical granularity to discretize the map of a city, then it feeds the

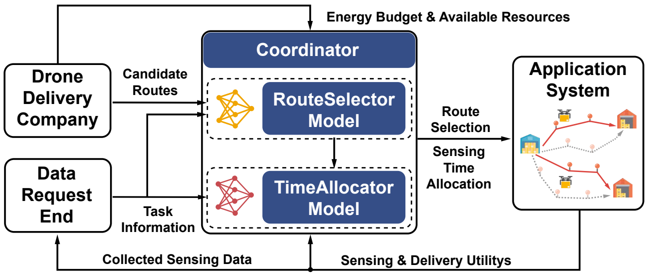


Figure 2: Illustration of architecture of DeliverSense.

application-specific sensing task information into the Coordinator for scheduling.

The **Coordinator** consists of RouteSelector and TimeAllocator. It receives the candidate routes and sensing tasks information, then schedules the delivery routes for the drones to achieve the largest total utility considering the practical limitations (described in §3.2).

The **Application System** actuates the delivery drones to conduct crowdsensing tasks when the scheduling decision is given. Then the application system return the collected data to the data request end after scheduling [36].

The key idea of the Coordinator design is to decompose the joint optimization problem and reduce the computational overhead. Specially, we decompose the architecture into two parts, including **RouteSelector** and **Timeallocator**. RouteSelector and TimeAllocator act as intelligent agents and make decisions for selecting routes and allocating time for each crowdsensing task, respectively.

Solution Overview: The Coordinator learns policies to rewrite the current solution and iteratively improve it until convergence, based on the Neural Rewriter [6]. We regard each candidate route as a *node* since utility is calculated on route level in the optimization. The Coordinator's observations are information from the data request end and drone delivery company. Each solution to the optimization problem and its contextual information is a *state*.

State Space: Fig. 3 illustrates two kinds of states for each step, including the state of the selected routes and the state of the corresponding sensing tasks. Considering the practical applications, we use the variables in the objective function and constraints as contextual information to construct features. Specifically,

Route state: Each route is specified as a four-dimension vector $(d_{ij} - d_{i0}, w_{ij}, \overline{b_{ij}}, n_k)$, where $\overline{b_{ij}}$ is the average benefit of all tasks along the route j of DT i and n_k is the total number of tasks along the route.

Task state: Each task is specified as a four-dimension vector $(t_{ij}^k, b_k, B_k, E_{ij})$ containing the task-specific information. Note that there are multiple sensing tasks on a delivery route.

3.2 Coordinator

Action Space: RouteSelector first implements feature extraction from the route states and task states of the selected nodes. Then these two kinds of features are concatenated as the state s_t of the current solution. The following action consists of three parts.

Region-picking policy: We first calculates a state-dependent *region set* $\Omega(s_t)$ with the integrated feature. Then $\pi_\omega(s_t)$ estimate each node's value in $\Omega(s_t)$, and obtain a subregion $\omega_t \in \Omega(s_t)$ from it with weighted sampling.

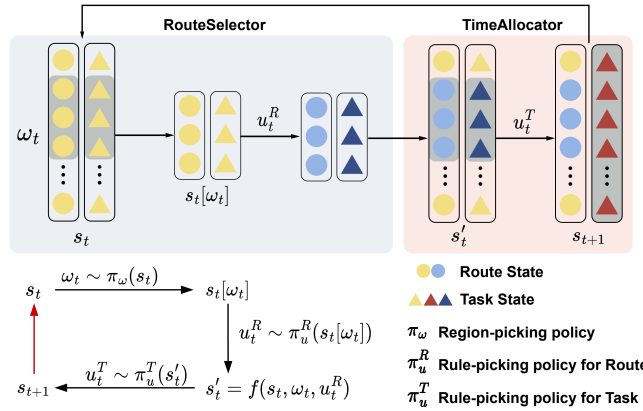


Figure 3: Illustration of Coordinator, including RouteSelector and TimeAllocator.

Rule-picking policy for route: $\pi_u^R(s_t[\omega_t])$ gives the probability distribution of applying each node of the *complement set* ϕ_t to rewrite the existing node in $s_t[\omega_t]$, where the complement set is a node set of candidate nodes without selection.

Rule-picking policy for task: Given the intermediate state s_t', $\pi_u^T(s_t')$ first leverages the self-attention mechanism to compute scores for all the tasks on the selected routes and sorts the tasks in descending order. Then it allocates sensing time greedily while checking if it satisfies the constraints (11)-(14).

Reward: We design a reward function as $r\left(s_t, \left(\omega_t, u_t^R, u_t^T\right)\right) = O(s_{t+1}) - O(s_t)$, where $O(\cdot)$ is the total utility definite in Eq. (7). The reward function encourages the Coordinator to choose the best routes and allocate appropriate sensing time as the constraints are satisfied.

Training: The goal of the Coordinator is to maximize the expected return reward from each state s_t. We train RouteSelector and TimeAllocator simultaneously. For the region picking policy π_ω in RouteSelector, we use the following equation to select the node in the current solution to rewrite:

$$\pi_\omega(\omega_t \mid s_t; \theta) = \frac{\exp(Q(s_t, \omega_t; \theta))}{\sum_{\omega_t} \exp(Q(s_t, \omega_t; \theta))}, \tag{15}$$

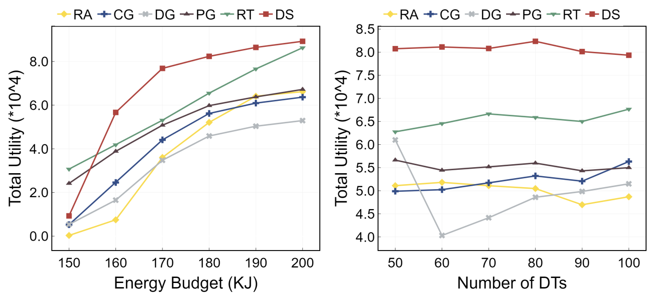
where Q is an approximate action-value function with parameters θ. Note that both the region-picking policy π_ω and the rule-picking for routes π_u^R are Multi-Layer Perceptron (MLP). The parameters of π_ω, π_u^R, and π_u^T are updated by the following loss function:

$$L_\omega(\theta) = \frac{1}{T}\sum_{t=0}^{T-1}\Big(\sum_{t'=t}^{T-1}\gamma^{t'-t} r(s_{t'}, (\omega_{t'}, u_{t'}^R, u_{t'}^T)) - Q(s_t, \omega_t; \theta)\Big)^2, \tag{16}$$

$$\Delta(s_t, (\omega_t, u_t^R, u_t^T)) = \sum_{t'=t}^{T-1}\gamma^{t'-t} r(s_{t'}, (\omega_{t'}, u_{t'}^R, u_{t'}^T)) - Q(s_t, \omega_t; \theta), \tag{17}$$

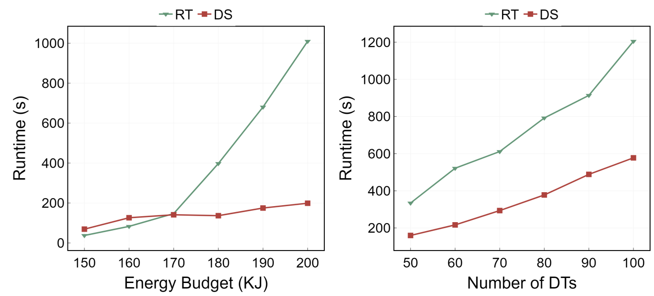
$$L_u(\phi_R, \phi_T) = -\sum_{t=0}^{T-1}\Delta(s_t, (\omega_t, u_t^R, u_t^T)) \\ (\log \pi_u^R(u_t^R \mid s_t[\omega_t]; \phi_R) + \log \pi_u^T(u_t^T \mid s_t'; \phi_T)), \tag{18}$$

where T is the length of the total rewriting steps, and γ is the decay factor. The total loss function is $L = L_u(\phi_R, \phi_T) + \lambda L_\omega(\theta)$, where λ is a coefficient.



(a) (b)

Figure 4: (a) Impact of energy budgets on total utility. (b) Impact of number of DTs on total utility.



(a) (b)

Figure 5: (a) Impact of energy budgets on run time. (b) Impact of number of DTs on run time.

4 EVALUATION

In this section, we validate our framework based on the data collected from a real implemented system. We first evaluate the impact of different key factors on total utility. Then we compare the computation cost with the state-of-the-art algorithm in terms of run time. For simplicity, our *DeliverSense* is denoted as *DS* in the evaluation.

4.1 Evaluation Setup

Real-world dataset: The real-world dataset comes from the delivery services station in Shanghai city. The area to evaluate occupies a large-scale size of 80 $km \times 80$ km. We generate the candidate delivery routes according to the distribution of the delivery stations. We set the default number of DT as 50 and the default number of each DT's candidate routes as 4. 80% of the data is used for training and the rest 20% for testing. The default energy budget is set as 150KJ. The delivery weight, energy limitation and parameters in the energy consumption model are set according to real-world applications [24]. For the crowdsensing task, we generate a uniform distribution in the city with 10,000 tasks.

Metrics: Since our goal is to optimize the total utility in practical time, we mainly focus on two metrics: total utility and run time. Large total utility and short run time mean better scheduling performance.

Parameters: For the coefficients in objective function (7), we set α=0.5 by default in our implement. The decay coefficient γ in the loss function is set as 0.9.

Baselines:

- *Cost-greedy (CG):* CG always selects the routes with maximal cost efficiency, which is the ratio of the total utility to the extra energy cost.
- *utility-greedy (PG):* PG always selects the routes with maximal total utility.
- *Distance-greedy (DG):* DG only selects the routes with minimal incremental distance.
- *Random (RA):* RA selects routes randomly.
- *RT:* RT is a state-of-the-art Route-Time joint allocation algorithm that leverages the p-exchange local search strategy to iteratively achieve the solution [32].

4.2 Evaluation Results and Analysis

In general, *DeliverSense* outperforms five baselines substantially in terms of total utility. Moreover, *DeliverSense* can achieve much lower computation complexity since it decomposes the scheduling process and fast approximates toward the optimum.

Impact of energy budget: As shown in Fig. 4(a), the average total utility of *DeliverSense* with different energy budgets is 66,806. *DeliverySense* outperforms RT, RA, CG, DG, and PG by 13%, 77%, 57%, 94%, and 31%, respectively. Especially, when the energy budget is 160KJ, *DeliverSense* achieves a total utility of 56,677, compared to RT given the best line, with a 35% improvement. This is because *DeliverSense* always selects the most valuable routes and provides a better time allocation strategy.

Impact of number of DTs: On average, for the total utility, *DeliverSense* significantly improves 23%, 61%, 54%, 64%, 46% over RT, RA, CG, DG, and PG approaches, respectively. Especially, when the number of DTs is 100, five baseline approaches achieve similar results in terms of the total utility with a value less than 60,000, while the *DeliverSense* maintains a total utility of about 80,000. This is because when the problem scale increases, even the state-of-the-art method RT cannot maintain a good performance without capturing the general structure of this NP-hard combinatorial problem.

Run time comparision: To evaluate the computation efficiency and advantages of adopting the reinforcement learning algorithm, we plot the run time under different parameters in Fig. 5. Note that we aim to compare the run time with RT which has the performance most similar to *DeliverSense* instead of the heuristic baselines approaches. As shown in Fig. 5(a), when the energy budget is more than 170 KJ, the run time of RT increases significantly, which is nearly 3 times longer than *DeliverSense* on average. *DeliverSense* utilizes reinforcement learning to replace the heavy computation via fast approximation, thus keeping stable when the problem scale increases. Overall, the comparison results show that *DeliverSense* is more efficient than RT and can generalize well to different settings.

5 CONCLUSION

In this paper, we first formalize the delivery drones scheduling problem with both sensing utility and delivery utility as dual objectives considering the practical factors. Then we propose *DeliverSense* framework with a reinforcement learning-based efficient solution, which decouples the highly complicated optimization search process and avoids the heavy computation. *DeliverSense* learns from large-scale real-world data to make near-optimal schedule decisions, including delivery routes selection and sensing time allocation. Evaluations show that the proposed method achieves better efficiency and optimality.

REFERENCES

[1] Amazon. 2020. *Amazon Prime Air.* https://www.amazon.com/Amazon-PrimeAir/b?ie=UTF8&node=8037720011

[2] Nils Boysen, Stefan Fedtke, and Stefan Schwerdfeger. 2021. Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum* 43, 1 (2021), 1–58.

[3] Xinlei Chen, Aveek Purohit, Carlos Ruiz Dominguez, Stefano Carpin, and Pei Zhang. 2015. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems.* 295–308.

[4] Xinlei Chen, Aveek Purohit, Shijia Pan, Carlos Ruiz, Jun Han, Zheng Sun, Frank Mokaya, Patric Tague, and Pei Zhang. 2017. Design experiences in minimalistic flying sensor node platform through sensorfly. *ACM Transactions on Sensor Networks (TOSN)* 13, 4 (2017), 1–37.

[5] Xinlei Chen, Carlos Ruiz, Sihan Zeng, Liyao Gao, Aveek Purohit, Stefano Carpin, and Pei Zhang. 2020. H-DrunkWalk: Collaborative and adaptive navigation for heterogeneous MAV swarm. *ACM Transactions on Sensor Networks (TOSN)* 16, 2 (2020), 1–27.

[6] Xinyun Chen and Yuandong Tian. 2019. Learning to Perform Local Rewriting for Combinatorial Optimization. *Advances in Neural Information Processing Systems* 32, NeurIPS (2019). arXiv:1810.00337

[7] Xinlei Chen, Susu Xu, Haohao Fu, Carlee Joe-Wong, Lin Zhang, Hae Young Noh, and Pei Zhang. 2019. ASC: Actuation system for city-wide crowdsensing with ride-sharing vehicular platform. In *Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering.* 19–24.

[8] Xinlei Chen, Susu Xu, Jun Han, Haohao Fu, Xidong Pi, Carlee Joe-Wong, Yong Li, Lin Zhang, Hae Young Noh, and Pei Zhang. 2020. Pas: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing. *IEEE Internet of Things Journal* 7, 5 (2020), 3719–3734.

[9] Xinlei Chen, Susu Xu, Xinyu Liu, Xiangxiang Xu, Hae Young Noh, Lin Zhang, and Pei Zhang. 2020. Adaptive hybrid model-enabled sensing system (HMSS) for mobile fine-grained air pollution estimation. *IEEE Transactions on Mobile Computing* (2020).

[10] Xinlei Chen, Xiangxiang Xu, Xinyu Liu, Hae Young Noh, Lin Zhang, and Pei Zhang. 2016. Hap: Fine-grained dynamic air pollution map reconstruction by hybrid adaptive particle filter. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM.* 336–337.

[11] Xinlei Chen, Xiangxiang Xu, Xinyu Liu, Shijia Pan, Jiayou He, Hae Young Noh, Lin Zhang, and Pei Zhang. 2018. Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers.* 1321–1330.

[12] DHL. 2019. *DHL Express Launches Its First Regular Fully-automated and intelligent Urban Drone Delivery Service.* https://www.dhl.com/twen/home/press/press-archive/2019/dhl-express-launches-its-first-regular-fully-automated-and-intelligent-urban-drone-delivery-service.html

[13] Ying Fu, Tao Zhang, Lizhi Wang, and Hua Huang. 2021. Coded Hyperspectral Image Reconstruction using Deep External and Internal Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[14] Gianpaolo Ghiani, Gilbert Laporte, and Roberto Musmanno. 2013. *Introduction to Logistics Systems Management* (second edition ed.). John Wiley & Sons, Ltd, Chichester, West Sussex, United Kingdom.

[15] Jinwei Liu, Haiying Shen, Husnu S Narman, Wingyan Chung, and Zongfang Lin. 2018. A survey of mobile crowdsensing techniques: A critical component for the internet of things. *ACM Transactions on Cyber-Physical Systems* 2, 3 (2018), 1–26.

[16] Huadong Ma, Dong Zhao, and Peiyan Yuan. 2014. Opportunities in mobile crowd sensing. *IEEE Communications Magazine* 52, 8 (2014), 29–35.

[17] Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. 2020. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies* 120 (2020), 102762.

[18] Naser Hossein Motlagh, Miloud Bagaa, and Tarik Taleb. 2017. UAV-based IoT platform: A crowd surveillance use case. *IEEE Communications Magazine* 55, 2 (2017), 128–134.

[19] Payments-Next. 2019. *Walmart drone delivery is up in the air.* https://paymentsnext.com/walmart-drone-delivery-is-up-in-the-air

[20] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Sensing as a service model for smart cities supported by internet of things. *Transactions on emerging telecommunications technologies* 25, 1 (2014), 81–93.

[21] Sixu Piao, Zhongjie Ba, Lu Su, Dimitrios Koutsonikolas, Shi Li, and Kui Ren. 2019. Automating csi measurement with uavs: from problem formulation to energy-optimal solution. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications.* IEEE, 2404–2412.

[22] Md Tahmid Rashid, Daniel Yue Zhang, and Dong Wang. 2020. Socialdrone: An integrated social media and drone sensing system for reliable disaster response. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications.* IEEE, 218–227.

[23] Valuates Reports. 2021. *Global Drone Package Delivery Market Size, Status and Forecast 2021-2027.* https://reports.valuates.com/reports/QYRE-Othe-1K491/drone-package-delivery

[24] Feng Shan, Junzhou Luo, Runqun Xiong, Wenjia Wu, and Jiashuo Li. 2020. Looking before crossing: An optimal algorithm to minimize UAV energy by speed scheduling with a practical flight energy model. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications.* IEEE, 1758–1767.

[25] Xi Tao and Abdelhakim Senhaji Hafid. 2021. Trajectory design in uav-aided mobile crowdsensing: A deep reinforcement learning approach. In *ICC 2021-IEEE International Conference on Communications.* IEEE, 1–6.

[26] Angelo Trotta, Fabio D Andreagiovanni, Marco Di Felice, Enrico Natalizio, and Kaushik Roy Chowdhury. 2018. When UAVs ride a bus: Towards energy-efficient city-scale video surveillance. In *Ieee infocom 2018-ieee conference on computer communications.* IEEE, 1043–1051.

[27] UPS. 2020. *UPS Flight Forward is changing the world of drone delivery.* https://www.ups.com/us/en/services/shipping-services/fight-forward-drones.page

[28] Kaixuan Wei, Ying Fu, Yinqiang Zheng, and Jiaolong Yang. 2021. Physics-based Noise Modeling for Extreme Low-light Photography. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3103114

[29] Chuxiong Wu, Xiaopeng Li, Lannan Luo, and Qiang Zeng. 2022. G2Auth: secure mutual authentication for drone delivery without special user-side hardware. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services.* 84–98.

[30] Zhengwei Wu, Xiaoxi Zhang, Susu Xu, Xinlei Chen, Pei Zhang, Hae Young Noh, and Carlee Joe-Wong. 2020. A generative simulation platform for multi-agent systems with incentives. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers.* 580–587.

[31] Jun Xia, Kai Wang, and Shuaian Wang. 2019. Drone scheduling to monitor vessels in emission control areas. *Transportation Research Part B: Methodological* 119 (2019), 174–196.

[32] Chaocan Xiang, Yanlin Zhou, Haipeng Dai, Yuben Qu, Suining He, Chao Chen, and Panlong Yang. 2021. Reusing Delivery Drones for Urban Crowdsensing. *IEEE Transactions on Mobile Computing* (2021).

[33] Susu Xu, Xinlei Chen, Xidong Pi, Carlee Joe-Wong, Pei Zhang, and Hae Young Noh. 2019. ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing. *IEEE Transactions on Mobile Computing* 19, 8 (2019), 1831–1847.

[34] Susu Xu, Xinlei Chen, Xidong Pi, Carlee Joe-Wong, Pei Zhang, and Hae Young Noh. 2019. Incentivizing vehicular crowdsensing system for large scale smart city applications. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019*, Vol. 10970. SPIE, 402–408.

[35] Susu Xu, Xinlei Chen, Xidong Pi, Carlee Joe-Wong, Pei Zhang, and Hae Young Noh. 2019. Vehicle dispatching for sensing coverage optimization in mobile crowdsensing systems. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* IEEE, 311–312.

[36] Xiangxiang Xu, Xinlei Chen, Xinyu Liu, Hae Young Noh, Pei Zhang, and Lin Zhang. 2016. Gotcha ii: Deployment of a vehicle-based environmental sensing system. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM.* 376–377.

[37] Yuzhe Yang, Zhiwen Hu, Kaigui Bian, and Lingyang Song. 2019. ImgSensingNet: UAV vision guided aerial-ground air quality sensing system. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications.* IEEE, 1207–1215.

[38] Junfang Yu and Yuanyuan Dong. 2013. Maximizing profit for vehicle routing under time and weight constraints. *International Journal of Production Economics* 145, 2 (2013), 573–583.

[39] Yunhao Zou and Ying Fu. 2022. Estimating Fine-Grained Noise Model via Contrastive Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 12682–12691.

[40] Yunhao Zou, Yinqiang Zheng, Tsuyoshi Takatani, and Ying Fu. 2021. Learning To Reconstruct High Speed and High Dynamic Range Videos From Events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2024–2033.