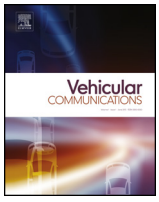
Vehicular Communications 42 (2023) 100631

Contents lists available at ScienceDirect

Vehicular Communications

journal homepage: www.elsevier.com/locate/vehcom



Cooperative aerial and ground vehicle trajectory planning for post-disaster communications: An attention-based learning approach ☆

Yuchao Zhu, Shaowei Wang *

School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China

ARTICLE INFO

Article history:
Received 25 October 2022
Received in revised form 26 March 2023
Accepted 9 June 2023
Available online 16 June 2023

Keywords:
Deep reinforcement learning
Emergency communications
Trajectory planning
Unmanned aerial vehicles

ABSTRACT

The ground infrastructures are highly susceptible to disruption after disasters, which causes the paralysis of communication. In this case, solutions besides original architecture are needed to meet the requirements of communication. Since unmanned aerial vehicle (UAV) can be quickly sent to disaster events to provide temporary connection due to its agility and mobility, it is suitable for performing disaster relief. Nevertheless, the limited onboard energy restricts the UAVs from fulfilling such persistent tasks. To this end, we introduce a ground vehicle carrying backup batteries to handle the energy issue of the UAV. Considering a time-constrained disaster-affected area, we propose a cooperative trajectory planning scheme to provide emergency communications swiftly and timely. The goal of our optimization task is to minimize the total cost of the mission, which consists of the operation cost for mission completion and the penalty cost for latency. We show that the task can be regarded as an extension of traveling salesman problem with soft time window constraints, which is NP-hard in general, and we propose a novel attention-based deep reinforcement learning with a sequential model strategy to learn the policy for the UAV's visiting order, based on which the trajectories of the UAV and ground vehicle are jointly designed. Numerical results show that our proposed attention-based trajectory planning scheme is effective and efficient, providing a guideline for the system design of post-disaster communications.

1. Introduction

Natural or man-made disasters, such as floods, hurricanes, fires, electrical outages, have been bringing enormous casualties and heavy losses of property yearly. The disaster management cycle mainly consists of three categories, i.e., pre-disaster, response and recovery [2]. Although some measures and precautions can be taken in the pre-disaster stage, disasters are generally unforeseeable and hard to be prevented. Response is the stage where resources are utilized to reach affected areas to save lives and assets, which is of paramount importance for minimizing further loss when a disaster strikes. Establishing a first contact and maintaining real-time communication with the affected victims are critical for obtaining post-disaster situational awareness, which can improve the efficiency of the subsequent rescue mission. Unfortunately, large-scale disasters are often accompanied by the destruction of basic infrastructures in the afflicted area, which results in the collapse of the function of ground communication devices. Accordingly, emergency communication with rapid response is crucial for search and rescue in the event of disasters [3].

Specific technical solutions have been proposed as potential candidates for supporting data transmission in disaster-affected regions, such as satellite communications and device-to-device (D2D) networks. Satellite communication technologies can provide large coverage regardless of the availability of terrestrial infrastructures, which is suitable for the post-disaster communication. But the low data rates, high end-to-end latency and limited dedicated equipment available to public users are still obstacles needed to be solved [4]. In another attempt, D2D networks, in which packets are transmitted between mobile users in proximity without traversing the base stations (BSs) or the core network, can potentially improve energy efficiency, throughput and delay [5]. Nevertheless, D2D connections are unreliable since the users may quit in the relay and the messages are susceptible to malicious interception or tampering.

Thanks to the fast-paced progress in design and production, unmanned aerial vehicles (UAVs), acting as BSs [6], relay nodes [7],

☆ This work was partially supported by the National Natural Science Foundation of China (61931023 and U1936202). Part of this work was presented at the IEEE GLOBECOM 2022 [1], Rio de Janeiro, Brazil, in December 2022.

* Corresponding author.
E-mail addresses: dz20230030@smail.nju.edu.cn (Y. Zhu), wangsw@nju.edu.cn (S. Wang).

https://doi.org/10.1016/j.vehcom.2023.100631
2214-2096/

and mobile anchors [8], have developed the application in various domains in recent years. Given the high penetration rate of mobile devices in our society, it is reasonably assumed that victims are equipped with smart devices that can be detected by the UAV networks. UAVs mounting BSs have emerged as a cost-efficient scheme to address emergency scenarios for multiple reasons [9,10]:

- Maneuverability and flexibility. UAVs can dynamically change their locations as well as approach difficult-to-reach places, which is suitable for devastated areas.
- On-demand service. UAVs can be rapidly deployed and reconfigured to form a standalone mobile network for disaster areas.
- Line-of-sight (LoS) channel. UAVs can adjust their flying altitudes to improve the probability of establishing LoS communication links with the ground users, which can provide high-quality data transmission.

The UAV deployment problem is considered in [11], a constrained clustering method and a joint resource allocation algorithm are proposed to maximize the energy efficiency of the network while meeting the real-time service and stringent quality of service constraints in disaster emergency communications. In [12], the UAV is used to perform the pseudo-trilateration technique to localize victims with an accuracy of tens of meters, which is practical for search and rescue. In [13], an integrated and dynamic deployment of aerial and ground BSs is proposed to provide swift and stable area coverage.

In particular, timeliness (e.g., the "Golden 72 hours" for life savings) is critical in the post-disaster rescue. Trapped individuals require prompt communication with relief personnel to report their situation, prioritizing in-time contact over access to long-term high data rate services. Moreover, the resources in the aftermath of disasters are definitely not abundant, including the availability of UAVs. Based on these characteristics of the post-disaster scenarios, we believe that it is more suitable for the UAVs to scan and scout the impacted region to provide on-demand service. It is also noteworthy that the UAV's flight time is still a bottleneck due to its limited battery capacity, which causes a severe impact on its performance in practice [14]. Therefore, how to scan the entire region as soon as possible under the constraint of scarce available resources remains a challenging problem.

Motivated by the aforementioned points, in this paper, we focus on the aerial-ground hybrid trajectory planning task in the time-constrained post-disaster scenario with one UAV (as mobile aerial BS) and one GV (as mobile recharging platform), where different parts of the affected region are assumed to have service deadlines reflecting their urgency (e.g., the region with collapsed buildings may have a tighter deadline than the open region with blocked roads) [15]. To summarize, the key contributions of this paper are listed as follows.

- To serve impacted regions swiftly and timely, we consider both the operation cost for the UAV and ground vehicle scanning the entire area, as well as the penalty cost generated by service latency. We aim to minimize the overall cost, which is a weighted sum of the operation cost and penalty cost, while satisfying the UAV's energy constraints. We show that the optimization task can be reconsidered as an extended traveling salesman problem with soft time windows (TSPSTW), which is NP-hard.
- We decompose the non-trivial optimization task into three tractable subproblems, and show that the crux of the problem is to sequentially select the visiting locations for the UAV. By viewing the trajectory planning problem as a sequence-to-sequence model [16], we propose a practical deep reinforcement learning-based collaborative route planning approach to address the time-constrained response mission.
- Extensive numerical results demonstrate that our proposal is effective and efficient, and the trained model by our proposed attention-based scheme shows good abilities of scalability and generalization, which offers an appealing balance between performance and complexity as compared to baseline algorithms.

We organized the rest of this article as follows. In Section 2, we provide a review of related works. In Section 3 we specify the system model and formulate the optimization task. In Section 4, we describe our proposed learning-based collaborative trajectory planning approach in detail. In Section 5 we present numerical results and performance evaluation. Finally, we conclude the paper in Section 6.

2. Related works

Recently, extensive studies have been done to address various challenges in UAV-assisted emergency communication networks [17–19], among which designing an optimal trajectory for the UAV remains a significant research challenge. In [20], the flight path is optimized to maximize the system throughput under limited UAV battery capacity. The optimization task is formulated as a multi-armed bandit problem and is solved by distance-aware upper confidence bound algorithm and ϵ-exploration algorithm. Taking limited user equipment energy and air obstacles into account, a restricted trajectory optimization problem to balance the system uplink throughput and energy efficiency is studied in [21], where the task is transformed into a constrained Markov decision-making process and tackled by a deep Q network based algorithm. In [7], the number of served devices is maximized by jointly optimizing bandwidth, power allocation, and the UAV trajectory while satisfying devices' latency requirements and the UAV's limited storage capacity. The emergency communication network with multiple UAVs is studied in [22], where an approximation algorithm is proposed to minimize the maximum mission time among a fleet of UAVs for disaster area surveillance. Wu et al. [23] present the air-ground cooperative emergency networks to reconstruct the communication in the disaster area, where the trajectory of ground vehicles (GVs) and UAVs are jointly optimized to maximize the average spectrum efficiency by multi-agent deep reinforcement learning methods.

The limited flight duration of the UAV, however, may cause intermittent communication. With the development in supporting infrastructures, such as automatic battery replacement systems, recent works have extended the limits of UAV applications by designing revolutionary UAV platforms [24]. A multi-UAV assisted data dissemination mission with fixed charging stations is studied in [25], the UAV's trajectory, along with user scheduling and association as well as their power assignment is optimized by a block coordinate descent-based algorithm. In [26], the optimal deployment of UAV recharging station is investigated, where the flight duration of the UAVs and the number of charging stations are jointly minimized by a heuristic algorithm based on the ant colony optimization. Adopting a GV as a mobile recharging station, Rucco et al. [27] present an optimal control approach while taking UAV and GV dynamics and kinematics into account. Zhu and Wang [28] consider the UAV-GV-aided data collection problem in a large-scale wireless sensor network. The cooperative trajectory planning of the UAV and GV is formulated as an extension of the traveling salesman problem for the purpose of minimizing mission completion time. Considering multiple UAVs and finite candidate locations of charging stations after a disaster, the path planning of UAVs and mobile recharging stations is regarded as a vehicle routing prob-

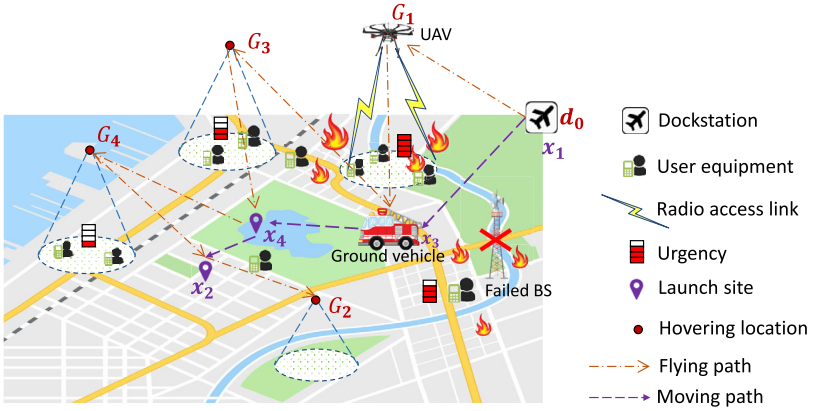


Fig. 1. UAV and GV assisted emergency communication networks.

lem with synchronized networks in [29] and solved by a genetic algorithm-based heuristic method.

In this paper, different from the aforementioned works, we are interested in studying the time-constrained aerial and ground vehicle assisted emergency communications, where the trajectories of the UAV and GV are designed while considering the urgency of different impacted regions. Particularly, we propose a deep reinforcement learning-based scheme to find the solution of NP-hard optimization task.

3. System model and problem formulation

3.1. Network description

In a disaster response application, without loss of generality, we consider a square region $\mathcal{A} \in \mathbb{R}^2$ with side length L as depicted in Fig. 1, where the existing ground network infrastructures are wiped out. In this case, the UAV is used as a mobile aerial BS to provide temporary communication for the trapped people. The UAV operates in hovering to connect with the ground user equipments (UEs) and the UAV's coverage radius is limited, thus the entire mission region is divided into N regions of interests (ROIs) for the purpose of serving each ROI through only one taking-off and landing. The population distribution which is generated by the living behavior can be regarded as regular from a statistical view, hence the trapped people are assumed to follow a given known distribution. We also assume that the communication time is proportional to the number of UEs in each ROI. Through the remote sensing images, we can obtain the severity of the damage caused by a disaster, consequently the service urgency differs by the damage severity of ROI. Taking the battery endurance of the UAV into consideration, an emergency GV is dispatched to move along with the UAV so that the UAV can fly back to the moving GV to replace its battery timely. The set of ROIs and UAV's 3D hovering positions are represented by $\mathcal{R} = \{R_1, R_2, ..., R_N\}$ and $\mathcal{G} = \{G_1, G_2, ..., G_N\}$, respectively. Each ROI R_i is assumed to have a service deadline τ_i reflecting its urgency. The UAV takes off from the dockstation $\boldsymbol{d}_0$ to traverses the ROIs along its trajectory, when the UAV hovers at $G_i = (g_i^x, g_i^y, g_i^z)$ with a duration T_i, it establishes communication links with potential UEs in ROI R_i. The UAV returns to the GV for battery replacement before it visits the next ROI, the positions at which they rendezvous with each other is denoted by 'launch sites' $\mathcal{X} = \{\boldsymbol{x}_i = (x_i^x, x_i^y, 0), 1 \le i \le N\}$. Let $\mathcal{U}_N \ni \sigma(\cdot)$ denote the set of permutations of $\{1, 2, ..., N\}$ representing all the possible sequences of visiting ROIs, where we set $\sigma(N+1) = \sigma(1)$ to simplify the notation (e.g. $\sigma = \{1, 3, 4, 2, 1\}$ in Fig. 1). The trajectories of the ground and aerial vehicle can be represented as $[\boldsymbol{x}_{\sigma(1)}, \boldsymbol{x}_{\sigma(2)}, ..., \boldsymbol{x}_{\sigma(N+1)}]$ and $[\boldsymbol{x}_{\sigma(1)}, G_{\sigma(1)}, \boldsymbol{x}_{\sigma(2)}, ..., G_{\sigma(N)}, \boldsymbol{x}_{\sigma(N+1)}]$, respectively, where the dockstation is the start/end point (i.e., $\boldsymbol{d}_0 = \boldsymbol{x}_{\sigma(1)} = \boldsymbol{x}_{\sigma(N+1)}$).

3.2. Channel model

The average statistics of channel state rather than the instantaneous ones are taken into consideration since the operation time is relatively long as compared to the channel coherence time. Hence, we only consider the large-scale path loss effect in the channel gain expression. We adopt the probabilistic air-to-ground channel model as given in [30], where the mean path loss of line-of-sight (LoS) links and non-line-of-sight (NLoS) links are as follows:

$$\begin{aligned} L_{LoS} &= 20\log d + 20\log f_c + 20\log(4\pi/c) + \mu_{LoS}, \\ L_{NLoS} &= 20\log d + 20\log f_c + 20\log(4\pi/c) + \mu_{NLoS}, \end{aligned} \tag{1}$$

where d is the distance between the UAV and the UEs, c is the speed of light, f_c represents the carrier frequency. $20\log d + 20\log f_c + 20\log(4\pi/c)$ is the free space propagation loss between the UAV and the UE. μ_{LoS} and μ_{NLoS} are the mean values of the additional loss in LoS and NLoS links, respectively.

The probability of LoS is closely approximated to a modified sigmoid function of the following form:

$$P_{LoS}(\psi) = \frac{1}{1 + a\exp\left[-b\left(\psi - a\right)\right]}, \tag{2}$$

where a, b are environment constants. Denote H as the altitude of the UAV and r as the horizontal distance between the UEs and UAV, the elevation angle between the UEs and the UAV can be written as $\psi = \frac{180^\circ}{\pi}\arctan(H/r)$. The probability of NLoS links is $P_{NLoS}(\psi) = 1 - P_{LoS}(\psi)$, then the average path loss can be expressed as:

$$\begin{aligned} P_{loss}(H, r) &= P_{LoS}(\psi)L_{LoS} + P_{NLoS}(\psi)L_{NLoS} \\ &= 20\log\sqrt{H^2 + r^2} + \frac{A}{1 + a\exp\left[-b\left(\arctan(\frac{H}{r}) - a\right)\right]} + B, \end{aligned} \tag{3}$$

where $A = \mu_{LoS} - \mu_{NLoS}$ and $B = 20\log f_c + 20\log(\frac{4\pi}{c}) + \mu_{NLoS}$ are constants under a given environment.

The average communication data rate between the UEs and the UAV is defined as:

$$\eta_t = W\log_2\left(1 + \frac{P_t}{P_{loss}N_0}\right), \tag{4}$$

where W is the communication bandwidth. N_0 and P_t are the noise power and transmission power, respectively.

Table 1
Notations and Physical meanings of variables in power consumption model.

Notation	Physical meaning
δ_p	Profile drag coefficient
κ	Incremental correction factor
Ω	Blade angular velocity (in radians/second)
ρ	Air density (in kg/m^3)
A_r	Rotor disc area (in m^2)
d_0	Fuselage drag ratio
P_0	Blade profile power in hovering status (in watt)
P_i	Induced power in hovering status (in watt)
R	Rotor radius (in meter)
s	Rotor solidity (in m^3)
v_{i0}	Mean rotor induced velocity in hover (in m/s)
W_A	Aircraft weight (in Newton)

3.3. UAV's energy model

Typically, the UAV's energy consumption consists of the communication-related energy and the propulsion energy. The former includes that for circuitry, signal reception and processing, etc., which can be regarded as a constant P_c for the sake of simplicity. The latter is required for the UAV to fly and keep aloft, which depends on the moving speed and the acceleration/deceleration of the UAV. In our considered large-scale disaster scenario, the duration of acceleration/deceleration phase is negligible when compared with the constant speed phase, thus we ignore the energy consumption caused by the acceleration/deceleration of UAV. Composed of blade profile, induced, and parasite power, the propulsion power consumption of the UAV as a function of its flight velocity can be expressed as follows [31]:

$$P(V) = P_0\left(1+\frac{3V^2}{\Omega^2R^2}\right) + P_i\left(\sqrt{1+\frac{V^4}{4v_{i0}^4}}-\frac{V^2}{2v_{i0}^2}\right) + \frac{1}{2}d_0\rho s A_r V^3, \tag{5}$$

where V is the velocity of the UAV. The notations and the corresponding physical meanings of the variables in (5) are clarified in Table 1. Also note that, $P_0 = \frac{\delta_p}{8}\rho s A_r \Omega^3 R^3$ and $P_i = (1+\kappa)\frac{W_A^{3/2}}{\sqrt{2\rho A_r}}$ are two constants depending on the weight of the UAV, air density, and rotor disc area, etc., which represent the blade profile power and induced power in hovering status, respectively.

3.4. Problem formulation

Our goal is to scan the entire region rapidly as well as trying not to exceed the ROIs' deadline, thus the total cost of the mission is defined as a weighted sum of the operation cost C_o and the penalty cost C_p. The former is the completion time of serving all the ROIs reflecting the swiftness of recovery while the latter is the penalty of service latency aiming at avoiding out of time.

Let $t_{GV}^{\sigma(i)}$ represent the amount of time for the GV to move from the i-th launch site to the next, and denote $t_{UAV}^{\sigma(i)}$ as the time for the UAV to leave one launch site, reach the i-th hovering location and provide communications, then return to rendezvous with the GV at the next launch site for battery replenishment, we have:

$$t_{GV}^{\sigma(i)} = \frac{\|\boldsymbol{x}_{\sigma(i)} - \boldsymbol{x}_{\sigma(i+1)}\|}{V_0}, \quad t_{UAV}^{\sigma(i)} = \frac{\|\boldsymbol{x}_{\sigma(i)} - G_{\sigma(i)}\| + \|G_{\sigma(i)} - \boldsymbol{x}_{\sigma(i+1)}\|}{V_1} + T_{\sigma(i)}, \tag{6}$$

where $V_0, V_1 > 0$ denote the speed of the GV and the UAV, respectively, with $V_0 < V_1$. Then the arrival time at the k-th ROI can be expressed as:

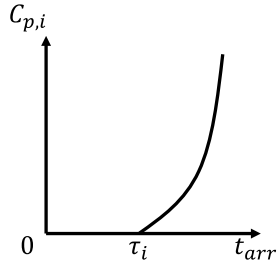


Fig. 2. The penalty function for the post-disaster scenario.

$$t_{arr}^{\sigma(k)} = \sum_{j=1}^{k-1}\max\left\{t_{GV}^{\sigma(j)}, t_{UAV}^{\sigma(j)}\right\} + \frac{1}{V_1}\|\boldsymbol{x}_{\sigma(k)} - G_{\sigma(k)}\|. \tag{7}$$

As for the post-disaster communication, intuitively, that the longer the service latency is, the higher the risk will be. For the purpose of avoiding extreme latency as well as guaranteeing the fairness of each ROI, we adopt a quadratic function depicted in Fig. 2 as the penalty function, and the penalty cost generated at ROI $R_{\sigma(k)}$ can be written as follows:

$$C_{p,\sigma_{(k)}} = \left(\max\left\{0, t_{arr}^{\sigma(k)} - \tau_{\sigma(k)}\right\}\right)^2. \tag{8}$$

Define α as the penalty coefficient that adjusts the sensitivity of latency, the total cost can be mathematically written as:

$$C_{total} = C_o + \alpha C_p = C_o + \alpha\sum_{k=1}^{N} C_{p,\sigma_{(k)}} = \sum_{i=1}^{N}\max\left\{t_{GV}^{\sigma(i)}, t_{UAV}^{\sigma(i)}\right\} + \alpha\sum_{k=1}^{N}\left(\max\left\{0, t_{arr}^{\sigma(k)} - \tau_{\sigma(k)}\right\}\right)^2. \tag{9}$$

The considered problem thus can be formulated as follows:

$$\underset{\sigma\in\mathcal{U}_N,\ \boldsymbol{x}_{\sigma(\cdot)}}{\text{minimize}} \quad C_{total} \qquad \text{s.t.} \quad P_f t_{UAV}^{\sigma(i)} + P_h T_{\sigma(i)} \le E_{max},\ 1\le i\le N, \tag{10}$$

where P_f, P_h are the power consumption of flying and hovering, respectively, and the UAV's battery capacity is denoted as E_{max}. We assume that the UAV only establishes connection with UEs when it hovers above corresponding ROIs, so it can be derived that $P_f = P(V_1)$ and $P_h = P(0) + P_c$, respectively. We can show that problem (10) is NP-hard, which is stated by the following theorem.

Definition 1. TSPSTW
Given a depot D and a set V of customers, each customer $v_i \in V$ is associated with a specified soft time window $[e_i, l_i]$ and a positive service time s_i. Soft time windows enable to serve customers outside their time windows, but some penalty costs must be incurred for early or late servicing [32]. The TSPSTW is to find a tour U such that
(a) the tour U starts from and ends at the depot D,
(b) U visits all of the customers in V, and
(c) the weighted cost of U (including travel, service and penalty costs) is minimized.

Theorem 1. *The problem (10) is NP-hard.*

Proof. We show that the cost minimization problem with battery constraints is NP-hard, by a reduction to a well-known NP-hard problem, i.e., the traveling salesman problem (TSP).

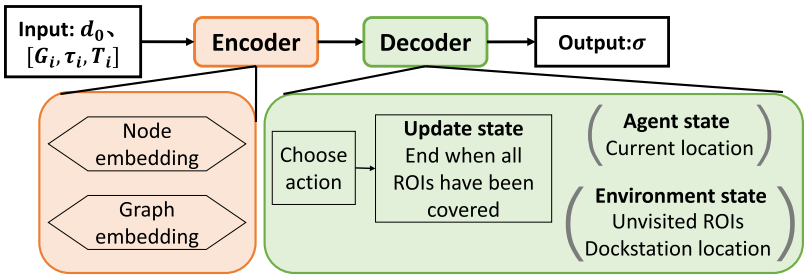


Fig. 3. The encoder-decoder framework.

When we set the battery constraints aside (i.e., the UAV does not need to fly back to the GV for battery replacement), we can rewrite the optimization task (10) as follows:

$$\underset{\sigma \in \mathcal{U}_N}{\text{minimize}}\ C(\sigma) = t_a^{\sigma(N+1)} + \alpha \left(\sum_{k=1}^{N} \left(\max \left\{ 0, t_a^{\sigma(k)} - \tau_{\sigma(k)} \right\} \right)^2 \right), \tag{11}$$

where $t_a^{\sigma(k)} = \sum_{j=0}^{k-1} \left(\frac{\|G_{\sigma(j)} - G_{\sigma(j+1)}\|}{V_1} + T_{\sigma(j)} \right)$ and we set $T_{\sigma(0)} = 0$. $G_{\sigma(0)} = G_{\sigma(N+1)} = \boldsymbol{d}_0$ is the start/end point.

According to Definition 1, problem (11) can be regarded as a task to find the TSPSTW tour, which is an extension of TSP that requires considering the service time window of the targets. The TSP is NP-hard problem [33], so is the TSPSTW. Our optimization task (10) considers the positions of launch sites and battery life of the UAV based on the TSPSTW, thus it is also NP-hard. □

Notes: Our formulation can be easily extended to the scenario with multiple UAVs and GVs by combining with a load-balancing region partitioning scheme. To be more specific, we can divide the entire region $\mathbf{R}_E$ into n disjoint subregions $\mathbf{R}_s^i$ in a way that balances the workload (i.e., the total cost of the mission) of each subregion. As the total cost relies on the hovering time, flying time and the penalty cost, which correspond to three measurable functions, i.e., the population distribution $g(\cdot)$, the region area $Area(\cdot)$ and the urgency $Urgency(\cdot)$[1] of the subregion, we can find an asymptotically load-balancing partition when satisfying $Area(\mathbf{R}_s^i)Urgency(\mathbf{R}_s^i) = \frac{Area(\mathbf{R}_E)Urgency(\mathbf{R}_E)}{n}$ and $\iint_{\mathbf{R}_s^i} g(x)dA = \frac{1}{n} \iint_{\mathbf{R}_E} g(x)dA$. The feasibility is given in Lemma 1, and we can apply ham sandwich cuts to find the equitable partition [34,35].

Lemma 1. *Given a simple polygon $\mathbb{S}$ with two measurable functions $f(\cdot)$ and $h(\cdot)$ defined on $\mathbb{S}$, there exists a partition of $\mathbb{S}$ into n relatively convex subregions $\{\mathbb{S}_1, ..., \mathbb{S}_n\}$ with disjoint interiors, while satisfying $\iint_{\mathbb{S}_i} f(x)dA = \frac{1}{n} \iint_{\mathbb{S}} f(x)dA$ and $\iint_{\mathbb{S}_i} h(x)dA = \frac{1}{n} \iint_{\mathbb{S}} h(x)dA$.*

4. Our proposed approach

Since the optimization task is a canonical example of combinatorial optimization and is challenging to solve, we decompose the task into three subproblems, i.e., selection of hovering positions, finding the near-optimal visiting sequence under deadline constraints, and determining the locations of launch sites. Observing that the formulated problem (10) over variables $\boldsymbol{x}_i$ will be convex and easy to solve when the permutation σ is fixed, then the key point of the task lies in the second subproblem. We propose a novel deep reinforcement learning-based scheme to deal with the cooperative trajectory optimization problem.

4.1. Selection of hovering positions

For preliminary preparation, we clarify the selection of hovering positions for the purpose of guaranteeing the communication links between the UAV and the UEs. Assume that the minimum requirement of transmission rate is η_t^{min}, i.e., the communication link is considered successful if $\eta_t \geq \eta_t^{min}$, we can correspondingly obtain a maximum allowable path loss P_{loss}^{max} according to (4). The UEs are supposed to connect to the UAV if satisfies $P_{loss}(H, r) \leq P_{loss}^{max}$, accordingly the maximum coverage radius can be written as:

$$r_{max} = \left\{ r | P_{loss}(H, r) = P_{loss}^{max} \right\}. \tag{12}$$

Since neither H nor r can be written as explicit function of each other, (12) is implicit. Note that the coverage radius rises first and then descends as the UAV altitude increases, we can search the value H satisfying $\partial r_{max}/\partial H = 0$ to get the optimal altitude H_{opt} that yields the widest coverage. Without loss of generality, the entire region is divided into multiple ROIs based on the maximum coverage radius to minimize the number of taking off and landing, i.e., the ROIs are small squares with side length $\frac{L}{\sqrt{2}r_{max}}$. The center of the ROIs is the horizontal part of the hovering positions, the flying altitude of the UAV is set as H_{opt}.

4.2. Attention-based framework for UAV's visiting order

Once the hovering positions are given, we can reconsider the optimization task as a TSPSTW with a given start point as shown in (11) by setting the battery constraint aside.

Problem (11) is still NP-hard and difficult to solve by exact methods. Note that the task can be viewed as a sequence decision problem by a policy, we propose a sequential model-based deep neural network to tackle the trajectory optimization problem in an unsupervised manner. As depicted in Fig. 3, one network encodes the input start node and all hovering nodes, and then another network converts the encoded information to a visiting order as its output.

Our attention-based encoder-decoder model defines a stochastic policy $p(\sigma|\boldsymbol{s})$ for selecting a solution σ to a problem instance $\boldsymbol{s}$, i.e., the probability that the UAV follows the corresponding trajectory can be decomposed using the following chain rule factorized and parameterized by $\boldsymbol{\theta}$ [36]:

$$p_{\boldsymbol{\theta}}(\sigma|\boldsymbol{s}) = \prod_{t=1}^{N} p_{\boldsymbol{\theta}}\left(\sigma(t)|\boldsymbol{s}, \sigma(1), ..., \sigma(t-1)\right), \tag{13}$$

where t is time step, $p_{\boldsymbol{\theta}}(\sigma(t)|\cdot)$ is the probability of the ROI being visited at the t-th time step based on $\boldsymbol{s}$ and the ROIs that have been visited at previous time steps.

[1] As the subregion will further be divided into N ROIs attaching with different service deadline τ_j, $j \in \{1, ..., N\}$ as described in 3.1, the urgency of each subregion can be quantified as $Urgency(\mathbf{R}_s^i) = 1/(\sum_{j=1}^{N} \tau_j)$.

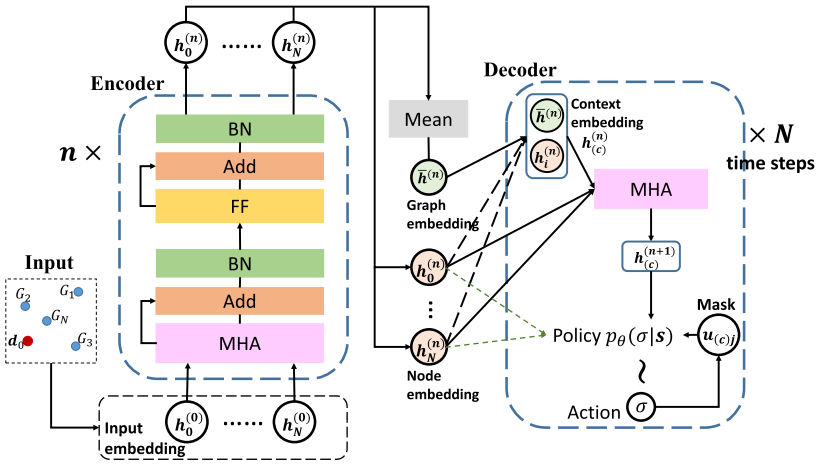


Fig. 4. The attention-based encoder-decoder structure.

4.2.1. Encoder

Following the Transformer architecture [37] but without positional encoding,[2] the encoder reads and maps the low-dimensional input features into several high D_h-dimensional vectors by a series of operations as shown in Fig. 4. Given the dockstation and the target hovering positions, one of the input features of the encoder is the coordinates of all the nodes. Additionally, we provide the deadline τ_i and required service time T_i of each ROI as input features of the hovering positions. To allow the model to distinguish the start node from the hovering nodes, separate parameters W_0^x and b_0^x are used to compute the initial embedding of the dockstation (start point). The initial embeddings are computed through a learned linear projection:

$$\boldsymbol{h}_i^{(0)} = \begin{cases} W_0^x \boldsymbol{d}_0 + \boldsymbol{b}_0^x, & i = 0 \\ W^x [G_i, \tau_i, T_i] + \boldsymbol{b}^x, & i = 1, ..., N, \end{cases} \tag{14}$$

where W_0^x, W^x, $\boldsymbol{b}_0^x$ and $\boldsymbol{b}^x$ are learnable parameters with sizes $D_h \times 2$, $D_h \times 4$, D_h and D_h, respectively.

The embeddings are then updated by n attention layers, each consisting of a multi-head attention (MHA) layer and a feed-forward (FF) layer. The attention mechanism passes weighted messages between the nodes in a graph, the attention weights ω_{ij} of which can be mathematically expressed as:

$$u_{ij} = \begin{cases} \frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{D_k}}, & \text{if } i \text{ adjacent to } j \\ -\infty, & \text{otherwise,} \end{cases} \tag{15}$$

$$w_{ij} = \text{softmax}(u_{ij}) = \frac{e^{u_{ij}}}{\sum_{j'} e^{u_{ij'}}}, \tag{16}$$

where $\boldsymbol{k}_i = W^K \boldsymbol{h}_i^{(\cdot)}$, $\boldsymbol{q}_i = W^Q \boldsymbol{h}_i^{(\cdot)}$ are the key and query for each node, respectively. u_{ij} calculates the compatibility of the query $\boldsymbol{q}_i$ of node i with the key $\boldsymbol{k}_j$ of node j as the scaled dot-product, and we compute the attention weights $w_{ij} \in [0, 1]$ using a softmax as shown in (16). Then, the vector $\boldsymbol{h}_i'$ received by node i is the combination of message $\boldsymbol{v}_j = W^V \boldsymbol{h}_j^{(\cdot)}$:

$$\boldsymbol{h}_i' = \sum_j w_{ij} \boldsymbol{v}_j, \tag{17}$$

[2] Note that we do not use positional encoding here since the resulting embeddings are invariant to the input order.

Instead of using a single head, a multi-head attention with head size M is used for feature augmentation, which allows nodes to receive different types of messages from neighbors. The final multi-head attention value for node i can be written as follows:

$$\text{MHA}_i(\boldsymbol{h}_1, ..., \boldsymbol{h}_N) = \sum_{m=1}^{M} W_m^O \boldsymbol{h}_{im}', \tag{18}$$

where W_m^O is a parameter matrix with size $d_h \times d_v$ used for projection. The output of the MHA sublayer along with skip connection is passed through batch normalization (BN) layer [38] and then a fully connected FF layer with ReLU activation function (the FF sublayer also adds a skip-connection and BN), the operations are expressed as:

$$\begin{aligned} \hat{\boldsymbol{h}}_i &= \text{BN}^l \left(\boldsymbol{h}_i^{(l-1)} + \text{MHA}_i^l \left(\boldsymbol{h}_1^{(l-1)}, ..., \boldsymbol{h}_n^{(l-1)} \right) \right), \\ \boldsymbol{h}_i^{(l)} &= \text{BN}^l \left(\hat{\boldsymbol{h}}_i + \text{FF}(\hat{\boldsymbol{h}}_i) \right), \end{aligned} \tag{19}$$

where l is the number of the layer, $\text{FF}(\hat{\boldsymbol{h}}_i) = W^{ff,1} \cdot \text{ReLU}\left(W^{ff,0} \hat{\boldsymbol{h}}_i + \boldsymbol{b}^{ff,0} + \boldsymbol{b}^{ff,1} \right)$.

Similar to [39], we compute an aggregated embedding $\bar{\boldsymbol{h}}^{(n)}$ of the input graph as the mean of final node embeddings: $\boldsymbol{h}_i^{(n)}$: $\bar{\boldsymbol{h}}^{(n)} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{h}_i^{(n)}$. Both the node embeddings $\boldsymbol{h}_i^{(n)}$ and the graph embedding $\bar{\boldsymbol{h}}^{(n)}$ are used as the input to the decoder.

4.2.2. Decoder

Decoding happens sequentially, the decoder outputs the selected node $\sigma(t)$ at time step $t \in \{1, ..., N\}$ based on the current state. We design the state and action space in an explicit manner:

- State: The state of the problem at the t-th time step includes the agent state and the environment state, the former is composed of the features of the UAV's current location, and the latter consists of the information of the dockstation and all the unvisited hovering positions. We design a special context node representing the decoding context to utilize the information of these states [39]:

$$\boldsymbol{h}_{(c)}^{(n)} = \begin{cases} \left[\bar{\boldsymbol{h}}^{(n)}, \boldsymbol{h}_{\sigma(t-1)}^{(n)} \right], & t > 1, \\ \left[\bar{\boldsymbol{h}}^{(n)}, \boldsymbol{h}_0^{(n)} \right], & t = 1. \end{cases} \tag{20}$$

Here $[\cdot, \cdot]$ is the horizontal concatenation operator.

- Action: The action of the UAV at times step t is the selection of the next target hovering position to be visited. The action takes affects on the environment and, consequently, changes the state in which the agent is.

As shown in Fig. 4, first, we compute a new context node embedding $\boldsymbol{h}_{(c)}^{(n+1)}$ using multi-head attention mechanism again to augment the exchange and fusion of information. Note two facts that the start point can not be visited if not yet all nodes have been visited, and the ROIs should not be visited twice, mask (set $u_{(c)j} = -\infty$) nodes are introduced. Thus, the compatibility of the query with all nodes is given as:

$$u_{(c)j} = \begin{cases} -\infty, & j = 0, \text{ and } t < N, \\ -\infty, & j \neq 0, \text{ and } \exists t' < t : \sigma(t') = j, \\ \frac{\boldsymbol{q}_{(c)}^T \boldsymbol{k}_j}{\sqrt{D_k}}, & \text{otherwise}, \end{cases} \tag{21}$$

where the keys $\boldsymbol{k}_i = W^K \boldsymbol{h}_i^{(n)}$ and values $\boldsymbol{v}_i = W^V \boldsymbol{h}_i^{(n)}$ come from the node embeddings, and the query $\boldsymbol{q}_{(c)} = W^Q \boldsymbol{h}_{(c)}^{(n)}$ is from the context node. Then, by applying the same multi-head self attention mechanism as described in (16)-(18), we get the result $\boldsymbol{h}_{(c)}^{(n+1)}$. With query from $\boldsymbol{h}_{(c)}^{(n+1)}$, we compute the compatibilities by (21) using a single attention head, and clip the result within $[-C, C]$: $u'_{(c)j} = C \cdot \tanh\left(u_{(c)j}\right)$ [40]. These compatibilities are regarded as unnormalized log-probabilities, and we compute the final probability p of choosing node i at time step t using a softmax function:

$$p_i = p_\theta\left(\sigma(t) = i | \boldsymbol{s}, \sigma(1), ..., \sigma(t-1)\right) = \frac{e^{u'_{(c)i}}}{\sum_j e^{u'_{(c)j}}}. \tag{22}$$

The decoder outputs the selected node based on p_i and the process will be end when all ROIs have been visited.

4.2.3. Training method

In reinforcement learning, an agent optimizes its behavior by interacting with the environment, which is treated as a black box. The presented attention-based neural network must be trained through exploring actions and receiving feedback in a form of rewards. We define the training objective function as follows:

$$\mathcal{C}(\boldsymbol{\theta}|\boldsymbol{s}) = \mathbb{E}_{p_\theta(\sigma|\boldsymbol{s})}\left[C(\sigma)\right], \tag{23}$$

which is the expectation of the total cost $C(\sigma)$ shown in (11).

We optimize $\mathcal{C}$ by gradient descent, using REINFORCE [41] gradient estimator with baseline $\boldsymbol{B}(s)$:

$$\mathcal{C}(\boldsymbol{\theta}|\boldsymbol{s}) = \mathbb{E}_{p_\theta(\sigma|\boldsymbol{s})}\left[\left(C(\sigma) - \boldsymbol{B}(s)\right) \nabla \log p_\theta(\sigma|\boldsymbol{s})\right]. \tag{24}$$

Algorithm 1 REINFORCE with baseline algorithm.

Input: number of epochs E, steps per epoch S, batch size B, training dataset $\mathcal{S} = \{\boldsymbol{s}_1, ...\boldsymbol{s}_{S\times B}\}$, significance δ.
Initialization: $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{bl} \leftarrow \boldsymbol{\theta}$
for epoch $= 1, ..., E$ **do**
  for step $= 1, ..., S$ **do**
    Randomly choose training data $\boldsymbol{s}_k (\forall k \in \{1, ..., B\})$ from $\mathcal{S}$;
    Find routes $\sigma_k (\forall k \in \{1, ..., B\})$ by sampling;
    Find routes $\sigma_k^{bl} (\forall k \in \{1, ..., B\})$ by greedy decoding;
    $\nabla \mathcal{C} \leftarrow \sum_{k=1}^{B} \left(C(\sigma_k) - C(\sigma_k^{bl})\right) \nabla_\theta \log p_\theta(\sigma_k)$;
    $\boldsymbol{\theta} \leftarrow \text{Adam}(\boldsymbol{\theta}, \nabla \mathcal{C})$;
  end for
  if OneSidedPairedTTest$(p_\theta, p_{\theta^{bl}}) < \delta$ **then**
    $\boldsymbol{\theta}^{bl} \leftarrow \boldsymbol{\theta}$
  end if
end for

The training procedure is shown in Algorithm 1. Here, greedy decoding[3] and sampling decoding[4] are employed for baseline policy and current policy, respectively. With the greedy decoding based baseline $\boldsymbol{B}(s)$, the function $C(\sigma) - \boldsymbol{B}(s)$ will be negative when the sampling decoding based solution is better than the greedy based one, causing actions to be reinforced, and vice versa. We compare the current model with the baseline model at the end of each epoch, and replace the baseline parameters $\boldsymbol{\theta}^{bl}$ only if the improvement is significant in terms of a paired t-test. If the baseline policy is updated, new evaluation instances are sampled to prevent over-fitting. The optimizer used to train the parameters is Adam [42].

4.3. Cooperative route planning

Algorithm 2 Cooperative route planning.

Initialization: region information: $\mathcal{A}$, $\boldsymbol{d}_0$, $\mathcal{G}$, T_i, τ_i $(1 \le i \le N)$, parameters of aerial and ground vehicle: V_0, V_1, P_f, P_h, E_{max}, late penalty: α.
Using attention-based neural network to solve the TSPSTW tour of $\mathcal{G}$ to find the visiting sequence;
Return the solution σ;
Using CVX to solve (25) with σ to find the optimal coordinated routes;
return σ, $\boldsymbol{x}$, c_{N+1}.

Given one permutation σ to visit all the ROIs, the origin problem can be rewritten as follows $(\forall i \in \{2, ..., N+1\})$:

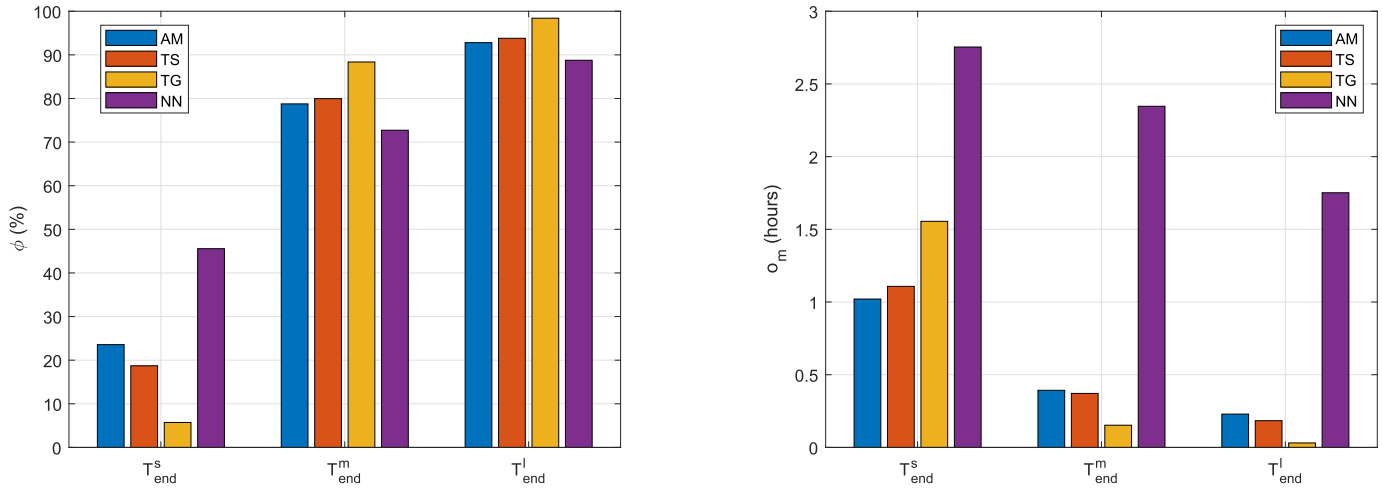
$$\begin{aligned} &\underset{\boldsymbol{x}_{(\cdot)}, t_{(\cdot)}, c_{(\cdot)}}{\text{minimize}} \quad c_{N+1} \\ \text{s.t. } C_1:\ & t_i \ge t_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - \boldsymbol{x}_i\|}{V_0}, \\ C_2:\ & t_i \ge t_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - G_{i-1}\| + \|G_{i-1} - \boldsymbol{x}_i\|}{V_1} + T_{i-1}, \\ C_3:\ & c_i \ge c_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - \boldsymbol{x}_i\|}{V_0}, \\ C_4:\ & c_i \ge c_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - G_{i-1}\| + \|G_{i-1} - \boldsymbol{x}_i\|}{V_1} + T_{i-1}, \\ C_5:\ & c_i \ge c_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - \boldsymbol{x}_i\|}{V_0} + \alpha\Big(t_{i-1} \\ & \quad + \frac{\|\boldsymbol{x}_{i-1} - G_{i-1}\|}{V_1} - \tau_{i-1}\Big), \\ C_6:\ & c_i \ge c_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - G_{i-1}\| + \|G_{i-1} - \boldsymbol{x}_i\|}{V_1} \\ & \quad + T_{i-1} + \alpha\left(t_{i-1} + \frac{\|\boldsymbol{x}_{i-1} - G_{i-1}\|}{V_1} - \tau_{i-1}\right), \\ C_7:\ & \|\boldsymbol{x}_{i-1} - G_{i-1}\| + \|G_{i-1} - \boldsymbol{x}_i\| \le V_1 \frac{E_{max} - P_h T_{i-1}}{P_f}, \\ C_8:\ & c_1 = 0,\ t_1 = 0,\ \boldsymbol{x}_1 = \boldsymbol{x}_{N+1}, \end{aligned} \tag{25}$$

where $t_{(\cdot)}$ and $c_{(\cdot)}$ represent the accumulative operation time and total cost at each ordered node, respectively. $\boldsymbol{x}_1 = \boldsymbol{x}_{N+1} = \boldsymbol{d}_0$ is the fixed dockstation. Therefore, based on the visiting sequence of the TSPSTW tour of $\mathcal{G}$, we can solve this convex optimization problem by using standard techniques, then the coordinated routes under ordered visiting assignment can be found.

5. Numerical results

We consider a disaster-affected geographical region in urban with a size of $L \times L$ km^2, where the dockstation is located at

[3] Select the best action with the maximum probability at each time step.
[4] Sample several solutions and report the best.



(a) The deadline hit rate. (b) The maximum latency.

Fig. 5. The deadline hit rate and the maximum latency under different ranges of deadline, $L = 12$ km, $\alpha = 3.57$.

$\boldsymbol{d}_0 = [0\text{m}, 0\text{m}]$. For the probabilistic air-to-ground channel, the urban environment parameters for $f_c = 2$ GHz are $a = 9.61$, $b = 0.16$, $\mu_{LoS} = 1$, $\mu_{NLoS} = 20$, respectively [30]. The communication bandwidth is $W = 1$ MHz, the transmission power and the noise power are $P_t = 20$ dBm and $N_0 = -110$ dBm, respectively [43]. Assume that the required transmission rate is 4.45 Mbps, thus the flying altitude is chosen as $H_{opt} = 1800$ m to achieve the maximum coverage radius $r_{max} = 2000$ m. The number of divided ROIs is set as $N = \left(\lceil \frac{L}{\sqrt{2} r_{max}} \rceil\right)^2$. The UAV's propulsion energy parameters are $\Omega = 300$ radians/s, $\rho = 1.225$ kg/m^3, $A_r = 0.503$ m^2, $d_0 = 0.3$, $R = 0.4$ m, $s = 0.05$, $v_{i0} = 4.03$ m/s, $P_0 = 79.86$ W, $P_i = 88.63$ W, respectively [31]. And the battery capacity of the UAV is 70 Wh. The hovering time above each ROI in range $(0, 9]$ minutes signifies the distribution of the UEs. The speeds of GV and UAV are set as $V_0 = 20$ km/h[5] and $V_1 = 80$ km/h, respectively. The time required for replacing battery can be ignored. Unless otherwise specified, the coefficient for late penalty is set as the mean value of the estimated operation time considering that T_o and T_p are of equal importance.

We train the model for 100 epochs with randomly generated data[6] under the learning rate of 10^{-4}. In every epoch, 2500 batches of 512 instances are processed. Each element in any problem instance is embedded into a vector of size 128 by the encoder network with 3 layers and 8 attention heads. To thoroughly evaluate the performance of our proposed attention model (AM) based cooperative route planning method, we compare it with the following common baseline methods, which adopt different strategies for the selection of visiting sequence:

(a) End time greedy (TG): TG is an intuitive greedy algorithm that only considers deadlines, in which the UAV designs its route based on the ROIs' urgency, i.e., the visiting sequence is arranged in an ascending order of deadlines.
(b) Nearest neighbor (NN): NN is another greedy-based method that only considers distance, where the UAV selects the one closest to its current position among all ROIs to be served.
(c) Tabu search (TS): The UAV's visiting sequence is designed by tabu search, a well-known meta-heuristic for vehicle routing problems [44], which employs a local search procedure to move from one potential solution to an improved solution in the neighborhood until some stopping criterion has been satisfied. Since tabu search is sensitive to the initial solution, here we initialize the route by TG. The tabu length is 30, and the maximum number of iterations is set as 500.

We generate results on CPU with Intel Core i7-9700 CPU @ 3.00 GHz speed, 16 GB memory ram and 64-bit windows operating system.

In addition to the total cost C_{total}, other meaningful evaluation indexes are also employed to measure the performance. The deadline hit rate (i.e., the percentage of on-time served ROIs) denoted as ϕ and the maximum latency denoted as o_m are used to reflect the timeliness and fairness of the service, respectively. We also denote T_t (in hours) as the time spent for traveling to measure the efficiency of the mission. All results are averaged over 100 Monte Carlo simulations.

First, for validating our proposed algorithm and illustrating the impact of deadlines on the trajectories, we present the performance of our proposal when given different ranges of deadline T_{end} (in hours) under a fixed region size. τ_i for each ROI is sampled from the uniform distribution $[0, T_{end}]$. Here we consider three representative deadlines for a region with 12×12 km^2, i.e., $T^s_{end} = 3.1$, $T^m_{end} = 6.2$, $T^l_{end} = 12.4$, reflecting the degree of urgency (see Appendix 7.2).

As can be seen from Fig. 5, the deadline hit rate increases when broadening the range of deadline and the maximum latency follows an opposite trend, which are expected since the probability of out-of-time decreases when extending the deadline. Our proposal achieves similar performance as compared to TS, with a slight advantage when the deadline is tight. It is worth noting that TG shows the best performance on timeliness and fairness when the deadline is relatively loose (e.g., T^m_{end} and T^l_{end}), while when the deadline becomes tighter (i.e., T^s_{end}), its performance deteriorates sharply. This is reasonable because the UAV would have abundant time for traveling to ROIs that are far away from each other due to the less strict deadlines, which is advantageous to the TG since it always reaches the most urgent ROI regardless of its location. NN seemingly outperforms the other three methods when it comes to on-time service under a tight deadline, but it causes extremely large latency, which is unacceptable.

Taking two instances in the scenarios of deadline T^s_{end} and T^l_{end} as examples, we give the trajectory planning results in Fig. 6. This

[5] A relatively lower speed than normal GV speed is set to compensate for the non-Euclidean distance of the route under real road condition, which would not influence the solution since we focus on the time cost.
[6] The method of dataset generation is given in Appendix 7.1.

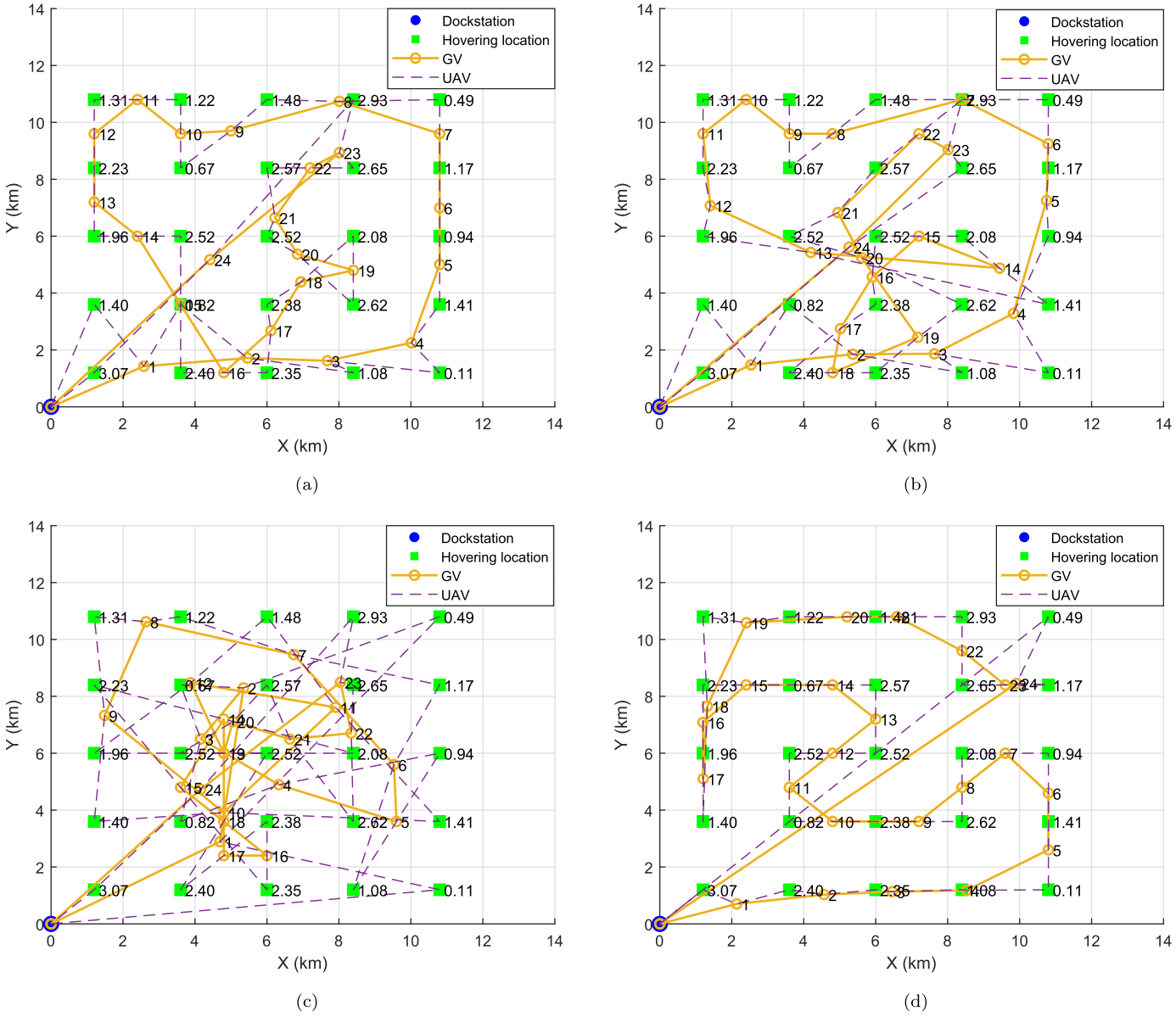


Fig. 6. The trajectory planning results of different methods, (a)-(d) correspond to the trajectory produced by AM, TS, TG, NN under T^s_{end}, respectively. (e)-(h) correspond to the trajectory produced by AM, TS, TG, NN under T^l_{end}, respectively. The number attached to the green squares are the service deadlines of the ROIs.

shows that the time windows of ROIs restrict the UAV and GV trajectories, which results in the intersections. For our proposal and TS, the intersections would be fewer as the deadline becomes looser. The average traveling time obtained from the randomly created 100 samples is depicted in Fig. 7. Generally speaking, our proposal searches for the trajectories more comprehensively and is able to find near-optimal routes from a global view, which tends to choose a path as short as possible on the premise of not causing significant latency. As can be seen from Fig. 8, our proposal always achieves the lowest total cost, especially when the deadline is tight. For T^s_{end}, the gain is 12.19%, 61.25% and 69.69% as compared to TS, TG and NN, respectively. Results indicate that our proposal can achieve a good trade-off among the swiftness, the timeliness as well as the fairness.

Then, we investigate the impact of the penalty coefficient. Here we also consider three representative penalty coefficients to reflect the tolerance of latency: α_m is a benchmark representing moderate delay tolerance, which is taken as the mean value of the estimated operation time, we set $\alpha_s = 0.1\alpha_m$ and $\alpha_l = 10\alpha_m$ to represent the delay-tolerant and delay-sensitive scenarios, respectively. Given a region with 12×12 km^2, the penalty coefficients are $\alpha_s = 0.357$, $\alpha_m = 3.57$ and $\alpha_l = 35.7$, respectively. Fig. 9 and Table 2 indicate that our proposal and TS would adjust the trajectory to balance the operation cost and penalty cost, which prefer to refrain from large maximum latency at a slight cost of traveling time as the significance of latency increases. Table 3 validates that our proposal can achieve the best performance on the total cost. AM improves about 8.66%, 56.21% and 64.01% on average in the total cost as compared with TS, TG and NN, respectively.

Table 2
Traveling time of different methods.

Penalty coefficient	α_s	α_m	α_l
AM	1.73	1.95	2.05
TS	1.90	2.07	2.14
TG	2.60	2.68	2.73
NN	1.72	1.92	1.98

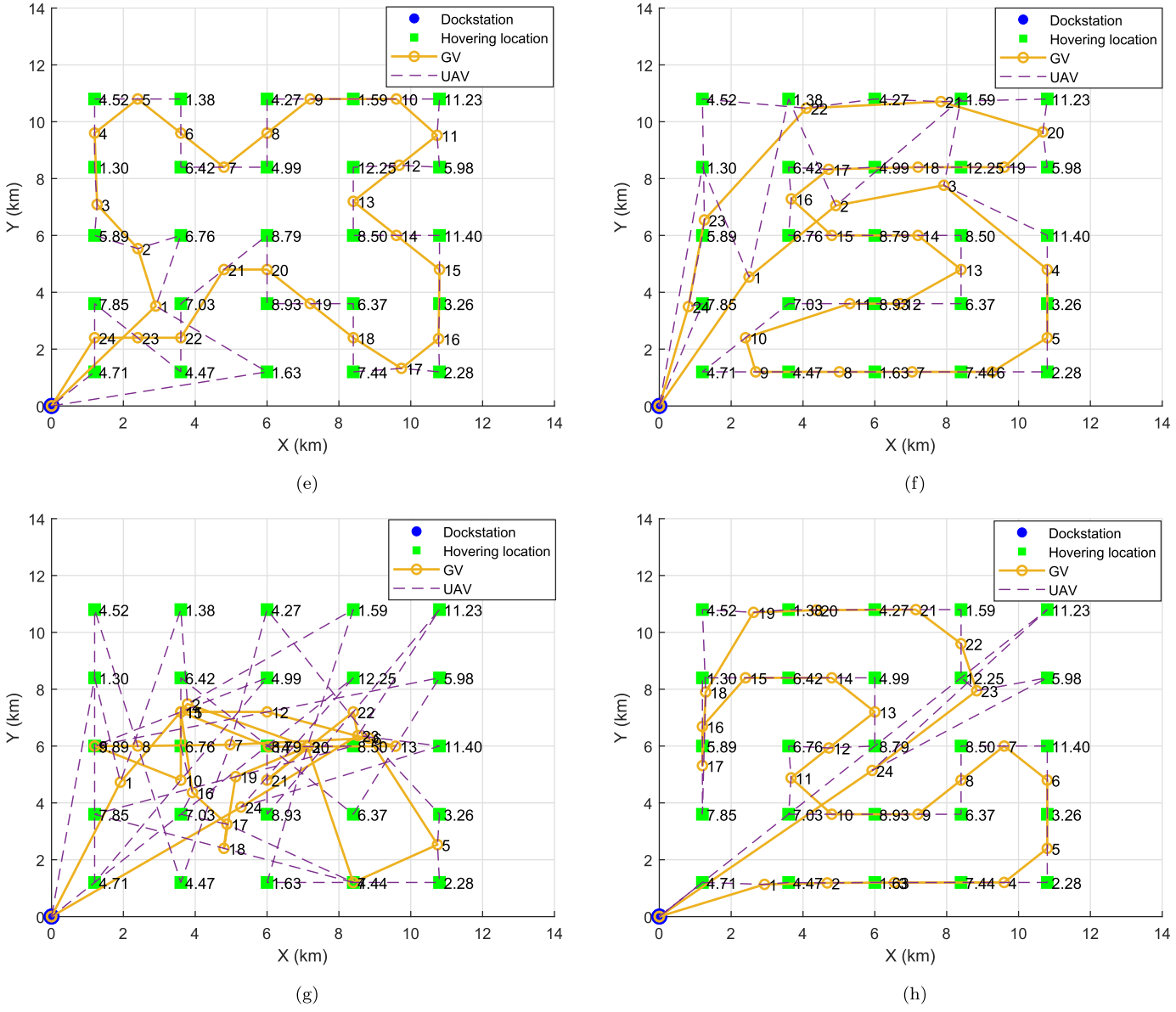


Fig. 6. (*continued*)

Table 3
Total cost of different methods.

Penalty coefficient	α_s	α_m	α_l
AM	6.61	30.99	288.27
TS	7.10	34.99	310.52
TG	12.21	79.28	748.97
NN	13.73	101.37	976.26

Table 4
Total cost of different methods.

L (km)	6	9	12	15	18	21
α	1.24	2.23	3.57	5.20	7.40	9.98
AM	1.26	2.31	4.27	8.03	24.03	368.79
TS	1.31	2.34	4.10	8.37	27.10	414.92
TG	1.39	2.69	4.71	8.91	43.48	4514.01
NN	1.39	4.79	32.45	167.01	884.05	3063.81

Finally, we investigate the bearing capacity of the system, i.e., the maximum region that could be covered potentially with the required deadline hit rate by one UAV and one GV. Given the range of service deadline $T_{end} = 12$ hours, Fig. 10 shows that the maximum manageable region is 18×18 km^2, with about 80% ROIs served on time and maximum latency less than one hour by our proposal. Although TG shows slight advantages on timeliness and fairness when the region is relatively small, it suffers severe delay under wide regions due to the rough-and-tumble routes. It is also worth noting that the performance of NN decreases linearly, and it would presumably achieve the highest timeliness as compared to the other three methods when the region is large, albeit at the cost of extremely large latency. The corresponding total cost of the four methods is given in Table 4, our proposal and TS achieve similar performance when the region is relatively small (i.e., $L \leq 12$ km), but the performance gap between our proposal and others becomes larger as the region extends since our proposed attention-based scheme can exploit the global information.

Furthermore, Table 5 gives the running times (in seconds) of different algorithms on 100 instances. As the region becomes larger (i.e., the number of ROIs increases), the computation time of all four methods increases. The computation time of our proposal is slightly higher than that of two greedy-based algorithms

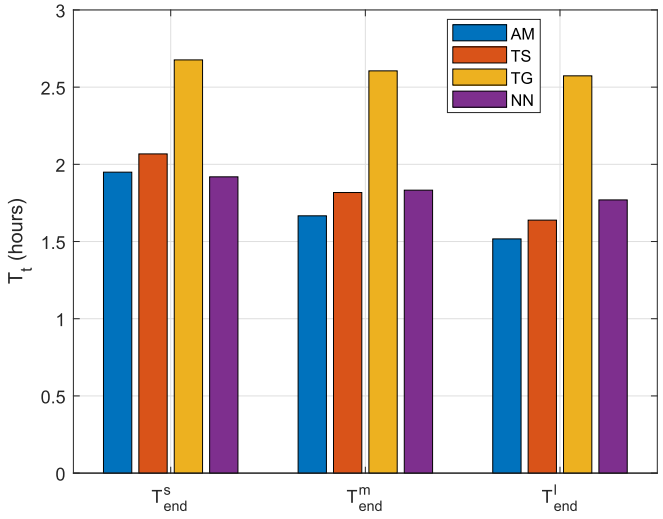


Fig. 7. The time spent for traveling under different ranges of deadline, $L = 12$ km, $\alpha = 3.57$.

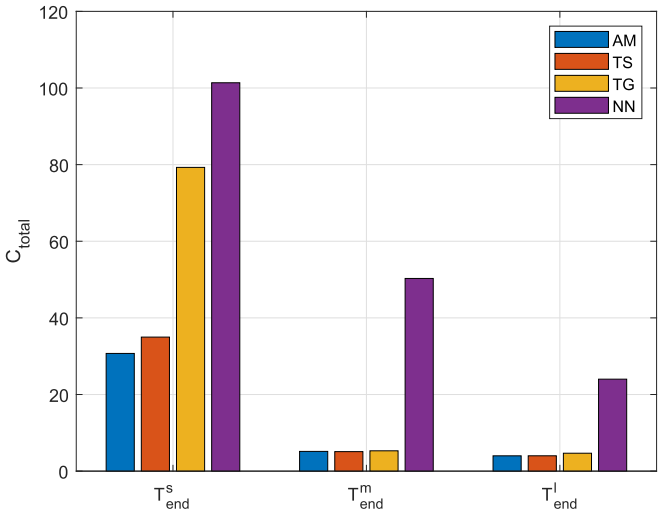


Fig. 8. The total cost under different ranges of deadline, $L = 12$ km, $\alpha = 3.57$.

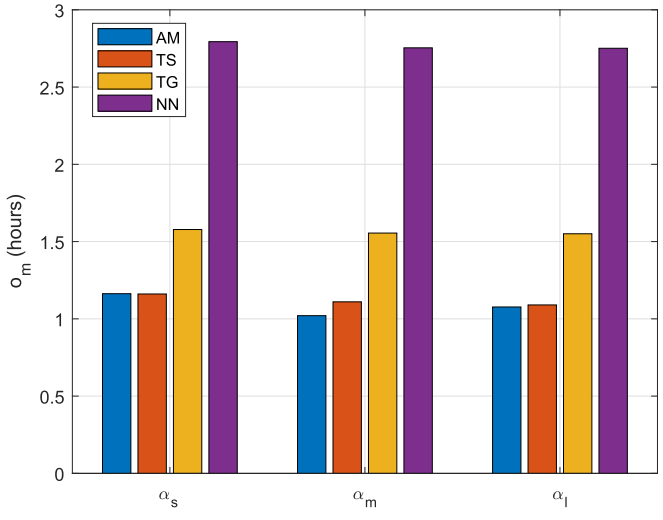


Fig. 9. The maximum latency under different ranges of penalty coefficient, $L = 12$ km, $T_{end} = 3.1$ hours.

Table 5
Running time comparison.

L (km)	6	9	12	15	18	21
AM	1.81	3.07	4.49	6.23	8.52	11.31
TS	1.92	3.47	5.68	9.59	17.49	34.10
TG	1.82	3.04	4.49	6.18	8.44	11.26
NN	1.80	3.04	4.45	6.18	8.40	11.25

(i.e., TG and NN), and significantly less than that of the TS method. We can also observe that the computational complexity of our proposal follows a linear growth with the size of the networks while that of TS grows exponentially. Our proposal thus is able to scale to large networks.

6. Conclusion

In this paper, we studied the collaborative trajectory planning problem targeting on the optimization of the overall cost of the mission in a time-constrained post-disaster area while applying both UAV and ground vehicles. Upon the formulation, we discovered that the optimization task is an extension of the traveling salesman problem with soft time windows, which is NP-hard. Inspired by the promising development of deep reinforcement learning, we proposed an innovative learning-based scheme to solve the non-trivial cooperative path planning task. The optimization task was decomposed into three tractable proportions: First, the hovering positions were selected based on the best coverage radius; second, a Seq2Seq neural network adopting attention mechanism was used to learn the policy of the trajectory planning with deadline constraints; third, a cooperative route planning algorithm was introduced to work out the optimal rendezvous for the aerial and ground vehicle. Numerical results indicated that our proposed learning-based scheme generally outperforms the compared methods, and achieves an appealing balance among swiftness, timeliness and fairness. In addition, our proposal has the advantage of scalability, which is a novel approach for large-scale disaster response that can open the door to a new field of UAV implementation and usage.

7. Appendices

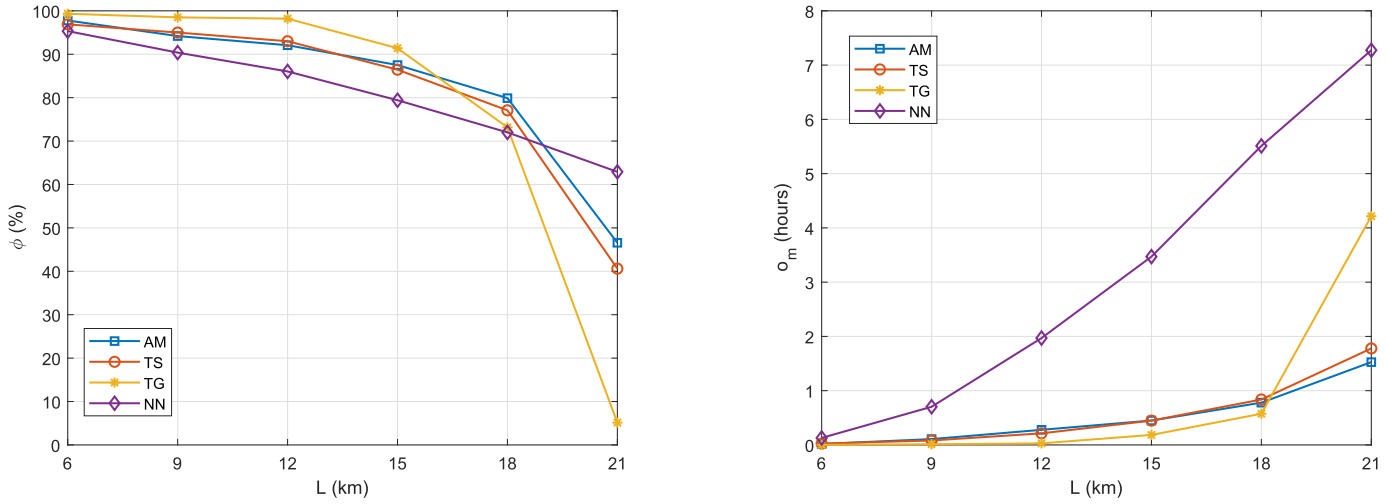
7.1. Dataset generation

The location of dockstation is fixed at $[0\text{m}, 0\text{m}]$. The hovering positions, deadlines and required hovering time are randomly generated from a uniform distribution. Specifically, the coordinates of N hovering positions are randomly generated in the square $L \times L$ km^2. The deadlines of ROIs are sampled $\tau_i \sim \text{Uniform}[0, T_{end}]$ hours. The hovering time above each ROI is sampled from the uniform distribution $(0, 9]$ minutes.

7.2. Selection of deadline

If the deadlines are too large, we can always visit all nodes, making the time constraint obsolete; if the deadlines are too small, the penalty will dominate the total cost, making trajectory planning useless. Therefore, we choose the deadline reflecting the urgency base on the estimation of the traveling time to avoid improper setting. The following classical theorem, known as BHH theorem [33], relates the length of a traveling salesman tour of a sequence of points with the distribution from which they were sampled:

Theorem 2. *Suppose that $X = \{X_1, X_2, ...\}$ is a sequence of random points independent and identically distributed according to an absolutely continuous probability density function f defined on a compact*



(a) The deadline hit rate. (b) The maximum latency

Fig. 10. The deadline hit rate and the maximum latency as functions of the region size with $T_{end} = 12$ hours.

planar region $\mathcal{D}$. Then with probability one, the length $TSP(X)$ of the optimal traveling salesman tour through X satisfies

$$\lim_{N\to\infty} \frac{TSP(X)}{\sqrt{N}} = \beta \iint_{\mathcal{D}} \sqrt{f(x)}dx, \tag{26}$$

where $0.6250 \leq \beta \leq 0.9204$ is a constant [45].

Taking hovering time t_{hover} and additional travel cost for the UAV returning the GV into consideration, we assume that the minimum required time to complete the mission of serving all the nodes X in a region $\mathcal{D}$ is

$$\mu_T = N\mathbb{E}(t_{hover}) + 1.5TSP(X)/V_1, \tag{27}$$

where $\mathbb{E}(t_{hover})$ is the mean value of the time spent for hovering above each ROI. Thus, we set the deadline $T^m_{end} = 2\mu_T$ as a benchmark, and set $T^s_{end} = 1/2T^m_{end}$, $T^l_{end} = 2T^m_{end}$ as the tight and loose deadlines, respectively.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

The authors would like to thank the editors and the anonymous reviewers, whose invaluable comments helped improve the presentation of this paper substantially.

References

[1] Y. Zhu, S. Wang, Learning-based cooperative aerial and ground vehicle routing for emergency communications, in: IEEE GLOBECOM'22, Rio de Janeiro, Brazil, 2022.
[2] B.E. Oruc, B.Y. Kara, Post-disaster assessment routing problem, Transp. Res., Part B, Methodol. 116 (2018) 76–102.
[3] D.C. Deepak, et al., An overview of post-disaster emergency communication systems in the future networks, IEEE Wirel. Commun. 26 (6) (2019) 132–139.
[4] T. Pecorella, et al., Emergency satellite communications: research and standardization activities, IEEE Commun. Mag. 53 (5) (2015) 170–177.
[5] A. Asadi, Q. Wang, V. Mancuso, A survey on device-to-device communication in cellular networks, IEEE Commun. Surv. Tutor. 16 (4) (2014) 1801–1819.
[6] N. Parvaresh, et al., A tutorial on AI-powered 3D deployment of drone base stations: state of the art, applications and challenges, Veh. Commun. 36 (2022) 100474.
[7] D.H. Tran, et al., UAV relay-assisted emergency communications in IoT networks: resource allocation and trajectory optimization, IEEE Trans. Wirel. Commun. 21 (3) (2022) 1621–1637.
[8] Y. Zhu, S. Wang, Aerial data collection with coordinated UAV and truck route planning in wireless sensor network, in: Proc. IEEE GLOBECOM'21, Madrid, Spain, 2021.
[9] K. Namuduri, Flying cell towers to the rescue, IEEE Spectr. 54 (9) (2017) 38–43.
[10] M. Erdelj, et al., Help from the sky: leveraging UAVs for disaster management, IEEE Pervasive Comput. 16 (1) (2017) 24–32.
[11] T. Do-Duy, et al., Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications, IEEE J. Sel. Areas Commun. 39 (11) (2021) 3411–3424.
[12] A. Albanese, V. Sciancalepore, X. Costa-Perez, SARDO: an automated search-and-rescue drone-based solution for victims localization, IEEE Trans. Mob. Comput. 21 (9) (2022) 3312–3325.
[13] X. Xu, Y. Zeng, Time-weighted coverage of integrated aerial and ground networks for post-disaster communications, in: Proc. IEEE WCNCW'20, Seoul, Korea (South), 2020.
[14] B. Li, Z. Fei, Y. Zhang, UAV communications for 5G and beyond: recent advances and future trends, IEEE Int. Things J. 6 (2) (2019) 2241–2263.
[15] M.T. Rashid, D.Y. Zhang, D. Wang, SocialDrone: an integrated social media and drone sensing system for reliable disaster response, in: Proc. IEEE INFOCOM'20, Toronto, ON, Canada, 2020.
[16] Y. Bengio, A. Lodi, A. Prouvost, Machine learning for combinatorial optimization: a methodological tour d'horizon, Eur. J. Oper. Res. 290 (2) (2021) 405–421.
[17] N. Zhao, et al., UAV-assisted emergency networks in disasters, IEEE Wirel. Commun. 26 (Feb. 2019) 45–51.
[18] L. Zhang, et al., Privacy-aware laser wireless power transfer for aerial multi-access edge computing: a colonel blotto game approach, IEEE Int. Things J. 10 (7) (2023) 5923–5939.
[19] L. Zhang, et al., Q-learning aided intelligent routing with maximum utility in cognitive UAV swarm for emergency communications, IEEE Trans. Veh. Technol. 72 (3) (2023) 3707–3723.
[20] Y. Lin, T. Wang, S. Wang, UAV-assisted emergency communications: an extended multi-armed bandit perspective, IEEE Commun. Lett. 23 (5) (2019) 938–941.
[21] T. Zhang, et al., Trajectory optimization for UAV emergency communication with limited user equipment energy: a safe-DQN approach, IEEE Trans. Green Commun. Netw. 5 (3) (2021) 1236–1247.
[22] Q. Guo, et al., Minimizing the longest tour time among a fleet of UAVs for disaster area surveillance, IEEE Trans. Mob. Comput. 21 (7) (2022) 2451–2465.
[23] S. Wu, et al., Distributed federated deep reinforcement learning based trajectory optimization for air-ground cooperative emergency networks, IEEE Trans. Veh. Technol. 71 (8) (2022) 9107–9112.
[24] Y. Wang, et al., Mobile wireless rechargeable UAV networks: challenges and solutions, IEEE Commun. Mag. 60 (3) (2022) 33–39.

[25] H. Yang, et al., Aiding a disaster spot via multi-UAV-based IoT networks: energy and mission completion time-aware trajectory optimization, IEEE Int. Things J. 9 (8) (2022) 5853–5867.
[26] M. Won, UBAT: on jointly optimizing UAV trajectories and placement of battery swap stations, in: Proc. IEEE ICRA'20, Paris, France, 2020.
[27] A. Rucco, et al., Optimal rendezvous trajectory for unmanned aerial-ground vehicles, IEEE Trans. Aerosp. Electron. Syst. 54 (2) (2018) 834–847.
[28] Y. Zhu, S. Wang, Efficient aerial data collection with cooperative trajectory planning for large-scale wireless sensor networks, IEEE Trans. Commun. 70 (1) (2022) 433–444.
[29] R.G. Ribeiro, et al., Unmanned-aerial-vehicle routing problem with mobile charging stations for assisting search and rescue missions in postdisaster scenarios, IEEE Trans. Syst. Man Cybern. Syst. 52 (11) (2022) 6682–6696.
[30] A. Al-Hourani, S. Kandeepan, S. Lardner, Optimal LAP altitude for maximum coverage, IEEE Wirel. Commun. Lett. 3 (6) (2014) 569–572.
[31] Y. Cai, et al., Joint trajectory and resource allocation design for energy-efficient secure UAV communication systems, IEEE Trans. Commun. 68 (7) (2020) 4536–4553.
[32] Y. Dumas, et al., An optimal algorithm for the traveling salesman problem with time windows, Oper. Res. 43 (2) (1995) 367–371.
[33] J. Beardwood, J.H. Halton, J.M. Hammersley, The shortest path through many points, in: Math. Proc. Cambridge Philos. Soc., vol. 55, Cambridge Univ. Press, 1959, pp. 299–327.
[34] S. Bespamyatnikh, D. Kirkpatrick, J. Snoeyink, Generalizing ham sandwich cuts to equitable subdivisions, Discrete Comput. Geom. 24 (4) (2000) 605–622.
[35] J.G. Carlsson, Dividing a territory among several vehicles, INFORMS J. Comput. 24 (4) (2012) 565–577.
[36] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, in: Proc. NeurIPS'15, Montreal, Quebec, Canada, 2015.
[37] A. Vaswani, et al., Attention is all you need, in: Proc. NeurIPS'17, Long Beach, California, USA, 2017.
[38] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proc. ICML'15, Lille, France, 2015.
[39] W. Kool, H. van Hoof, M. Welling, Attention, learn to solve routing problems!, in: Proc. ICLR'19, New Orleans, LA, USA, 2019.
[40] I. Bello, et al., Neural combinatorial optimization with reinforcement learning, https://doi.org/10.48550/ARXIV.1611.09940, 2016.
[41] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (3) (1992) 229–256.
[42] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, https://doi.org/10.48550/ARXIV.1412.6980, 2014.
[43] Y. Zeng, R. Zhang, Energy-efficient UAV communication with trajectory optimization, IEEE Trans. Wirel. Commun. 16 (6) (2017) 3747–3760.
[44] M. Zachariasen, M. Dam, Tabu Search on the Geometric Traveling Salesman Problem, Springer US, Boston, MA, 1996, pp. 571–587.
[45] D.L. Applegate, et al., The Traveling Salesman Problem: A Computational Study, Princeton Univ. Press, 2006.