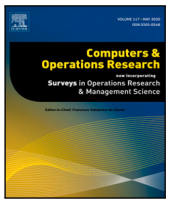
Computers & Operations Research 174 (2025) 106890

Contents lists available at ScienceDirect

Computers and Operations Research

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



Optimizing task assignment and routing operations with a heterogeneous fleet of unmanned aerial vehicles for emergency healthcare services

Ziru Lin [a,b], Xiaofeng Xu [a], Emrah Demir [b,*], Gilbert Laporte [c,d]

[a] School of Economics and Management, China University of Petroleum, Qingdao, 266580, China
[b] Logistics and Operations Management, Cardiff Business School, Cardiff University, Cardiff, CF10 3EU, UK
[c] Department of Decision Sciences, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada
[d] School of Management, University of Bath, Bath, BA2 7AY, United Kingdom

ARTICLE INFO

Keywords:
Healthcare logistics
Unmanned aerial vehicles
Team orienteering problem
Q-learning
Adaptive large neighborhood search

ABSTRACT

This paper studies the optimization of task assignment and pickup and delivery operations using a heterogeneous fleet of unmanned aerial vehicles (UAVs). We specifically address the distribution of emergency medical supplies, including medications, vaccines, and essential medical aid, as well as the collection of biological blood samples for testing and analysis. Unique challenges, such as supply shortages, time windows, and geographical considerations, are explicitly taken into account. The problem is first formulated as a mixed-integer linear programming model aimed at maximizing the total profit derived from the execution of a set of emergency healthcare pickup and delivery tasks. An enhanced Q-learning-based adaptive large neighborhood search (QALNS) is proposed for large-scale benchmark instances. QALNS exhibits a superior performance on benchmark instances. It also improves the quality of the solutions on average by 5.49% and 6.86% compared to the Gurobi solver and a state-of-the-art adaptive large neighborhood search algorithm, respectively. Sensitivity analyses are performed on critical factors contributing to the performance of the QALNS algorithm, such as the learning rate and the discount indicator. Finally, we provide managerial insights on the use of the fleet of UAVs and the design of the network.

1. Introduction

The increasing occurrence of natural disasters causes significant disruptions in global logistics and supply chain networks. Following major natural disasters, traditional land transport systems may be disrupted (Jeong et al., 2019). Disaster-stricken areas often have an urgent need to distribute emergency supplies such as medications, vaccines, and other essentials, along with the collection of biological blood samples. These emergency challenges prompt us to reconsider and explore new delivery methods with the help of promotion of last-mile innovations (Demir et al., 2022).

With the rapid development of e-commerce and intelligent autonomous technologies in various industries, unmanned aerial vehicles (UAVs) are gradually finding extensive applications, such as emergency rescue, communication relay, and disaster detection (see e.g., Saxena et al., 2019; Liu et al., 2022). UAVs offer distinct advantages such as mobility and movement flexibility in a limited traffic infrastructure, and they can already be used for transportation in emergency healthcare (Dukkanci et al., 2023). For example, UAVLatam is partnering with Wingcopter to use UAVs to deliver medicines, vaccines, laboratory samples, and specific treatments to remote villages in the Peruvian Andes (Howe, 2022). Zipline operates on four continents (Africa, Asia, Europe and North America) and in eight countries: Côte D'Ivoire, Ghana, Japan, Kenya, Nigeria, Rwanda, the United Kingdom and the United States. It supports the medical, health, and retail sectors and serves more than 4,000 health centers and over 45 million people (Zipline, 2024). At the same time, heterogeneous UAVs have different flight advantages and provide diverse solutions for handling the ever-changing healthcare logistics tasks in emergency services. Exploiting the advantages of UAVs in emergency healthcare to flexibly and quickly achieve maximum use of limited resources is highly important for enhancing the flexibility and stability of emergency medical logistics systems.

Compared to last-mile logistics, emergency healthcare delivery and pickup services typically involve more complicated tasks, tighter time windows, and limited available supplies (Fragkos et al., 2022). First, distinct characteristics for pickup and delivery tasks arise from customer needs, as indicated by Enayati et al. (2023). This requires the matching of different tasks and heterogeneous UAVs in the first stage of

* Corresponding author.
E-mail addresses: linzr_1018@163.com (Z. Lin), xuxiaofeng@upc.edu.cn (X. Xu), DemirE@cardiff.ac.uk (E. Demir), gilbert.laporte@cirrelt.net (G. Laporte).

https://doi.org/10.1016/j.cor.2024.106890
Received 26 June 2024; Received in revised form 27 October 2024; Accepted 28 October 2024
Available online 6 November 2024

optimization. Second, the unpredictability of emergency events makes it impossible to maintain a large reserve of resources, resulting in the lack of emergency resources after a disaster (Gao et al., 2020). Meanwhile, customer tolerance for the timely delivery service of emergency medical items is lower than that of standard parcel delivery. Existing studies on emergency logistics have focused mainly on route planning problem, along with some side constraints, such as time windows, limited capacity and endurance, and landing platforms (Li et al., 2020; Ghelichi et al., 2021). However, the heterogeneity of tasks and the capability of a wide range of UAV types are rarely considered simultaneously. The scheduling of a heterogeneous UAV fleet, based on an optimal utilization of UAVs, has yet to be thoroughly researched. This gives rise to the task assignment and pickup and delivery operations using a heterogeneous fleet of unmanned aerial vehicles (HUTA-PDP).

Taking into account the effectiveness of task assignment, supply shortages, and tighter service windows, the HUTA-PDP in emergency healthcare services can be regarded as a variant of the team orienteering problem (TOP) (Hanafi et al., 2020). However, given that different urgency priorities of healthcare demand leads to varying profits for task completion, as well as considering factors such as pickup and delivery and time windows, the HUTA-PDP shows a correlation with both the price-collecting vehicle routing problem (PCVRP) (Stavropoulou et al., 2019) and the pickup and delivery problem with time windows (PDPTW) (Sartori and Buriol, 2020). Since both TOP and PCVRP are NP-hard, classical exact solution algorithms often struggle to solve large-scale instances in a timely and efficient manner (Xu et al., 2022). In contrast, metaheuristics can effectively solve (Moosavi Heris et al., 2022). For example, adaptive large neighborhood search algorithm (ALNS) is particularly effective due to its adaptability and its ability to explore extensive neighborhoods of a feasible solution (Hammami et al., 2020). Given the complex and large-scale nature of the HUTA-PDP, we also introduce Q-learning, a reinforcement learning technique, to enhance the operator selection and performance of on ALNS algorithm.

Considering the significance of emergency healthcare logistics, the main contributions of our research are threefold. (*i*) An integrated optimization model of task assignment and route planning of pickup and delivery services with a heterogeneous UAV fleet is studied. We note that emergency healthcare services are considered, characterized by time sensitivity and resource scarcity. (*ii*) A new metaheuristic variant, namely Q-learning ALNS (QALNS), is developed. We incorporate Q-learning into multiple parts of the algorithm (i.e., operator selection and execution) to improve its performance. Finally (*iii*) extensive numerical results demonstrate the strong performance of the proposed QALNS algorithm. It significantly improves both convergence speed and the quality of a solution when compared to an exact method and to a standard ALNS algorithm. This makes it an ideal approach for solving emergency healthcare logistics problems.

The remainder of this paper is organized as follows. Section 2 reviews related previous works in healthcare logistics, as well as UAV route planning and Q-learning. Section 3 provides a description of the HUTA-PDP and introduces a mixed-integer linear programming (MILP) formulation. The overall framework of QALNS algorithm, along with detailed explanations of Q-learning method are described in Section 4. A numerical study is reported in Section 5. Finally, Section 6 concludes and discusses future research directions.

2. Literature review

This section presents a brief literature review related to three research areas: (*i*) healthcare pickup and delivery (PD) operations, (*ii*) UAV (drone) routing problems, and (*iii*) Q-learning.

The literature on healthcare logistics focuses on the management of post-disaster operations to efficiently and effectively distribute medications, vaccines, and other medical supplies to people in need, and to collect biological samples as well (Lu et al., 2022). It prioritizes task completion time and total cost, while being constrained by various factors such as the relationship of supply and demand, time windows and resource types (see e.g., Elluru et al., 2019; Do C. Martins et al., 2021). In recent years, the focus of healthcare logistics has rapidly expanded, incorporating objectives such as lifesaving utility, humanitarian fairness and delay costs (see e.g., Huang et al., 2015; Seraji et al., 2022). Considering the suddenness of emergency events often leading to short-term shortages of healthcare supplies, maximizing task satisfaction rates with limited resources became one of the key objectives (Cinar et al., 2021). In this situation, the TOP, which was introduced by Chao et al. (1996), provides potential solutions for the pickup and delivery with limited healthcare supplies (Dasdemir et al., 2022). For example, Martin et al. (2021) provided MILP formulations for a pickup and delivery TOP for various business cases with competing operators. Angelelli et al. (2021) considered an online version of the TOP with pickup and delivery, with the objective of maximizing the sum of expected profits for each acceptance and rejection decision. Although the literature on the TOP with time windows (TOPTW) is quite rich (Yu et al., 2019; Wang et al., 2019), there is limited research in the area of healthcare logistics. Yadav and Tanksale (2022) introduced a general model for the joint problem of routing and scheduling home health care workers to maximize profits. In another study, Saeedvand et al. (2020) explored the TOPTW for a disaster rescue scenario with the aim of optimizing multiple objectives, including task rewards, completion time, energy consumption, and missed deadline penalties. To the best of our knowledge, the pickup and delivery, time windows, task priorities, and limited supplies for diverse healthcare tasks within the TOP framework, has not yet been examined.

In the context of healthcare logistics, it is challenging to provide pickup and delivery services due to complex terrain-related factors (Xiong et al., 2023; Zhen et al., 2024). The efficient and timely delivery of UAVs can significantly enhance the operations of healthcare network operations (Gao et al., 2023). However, many studies reveal that such services also entail new challenges, such as limited flight range and payload capacity (Wang et al., 2023). Differences in UAV types also result in performance variations, including the number of drops that they can perform, takeoff and landing methods, flight height, etc. (see e.g., Gonzalez-R et al., 2020; She and Ouyang, 2021).

Given the diversity of UAV types, there is limited research on heterogeneous UAV fleets (Zhou et al., 2018; Pasha et al., 2022). In a recent study, Liu et al. (2022) presented an optimal task assignment and path planning method for multiple disaster relief UAVs, including delivering medicine, collecting images and relaying communications. Bartolini et al. (2020) applied multiple fleets of cooperative UAVs to monitor critical scenarios, detecting anomalies early and intervening as necessary. In another study, Wen and Wu (2022) examined a logistics delivery issue involving a heterogeneous multi-drone system, in which a large drone transports multiple small-size drones to distribution areas. Despite the existence of some research that focuses on heterogeneous tasks within logistics networks (Kang and Lee, 2021; Wang et al., 2023), few studies match specific task requirements based on UAV fleet characteristics, such as utilizing fixed-wing UAVs for high-altitude delivery tasks, multi-drop UAVs for geographically dense tasks and multi-pickup, and high-speed UAVs for urgent demands.

Metaheuristics are computationally intelligent algorithms that are widely used for solving complex optimization problems, particularly NP-hard problems (Osman and Laporte, 1996). ALNS, equipped with selection processes, was proposed by Ropke and Pisinger (2006). It enables the exploration of multiple neighborhoods within the same search procedure in an adaptive fashion (Boualamia et al., 2023), and has proved to be successful in solving a wide range of VRPs (Aksen et al., 2014; Gu et al., 2019). In the last decade, there has been a rising interest in combining reinforcement learning methods with metaheuristics to solve different variations of VRPs (Qin et al., 2021; Zhang et al., 2023). Some studies argue that since the data stored in the Q-table should not be excessively large, they opt to utilize

neural networks to convert Q-learning into deep reinforcement learning methods (Chen et al., 2022; Kallestad et al., 2023). However, for discrete data processing problems such as the HUTA-PDP, deep learning methods can be quite time-consuming, whereas Q-learning enables faster computations (Zhu and Fang, 2021). By integrating Q-learning into a multi-purpose metaheuristic, Chen and Wang (2024) proposed a new learning-based ruin-and-recreate heuristics framework, which proved to be one of the most advanced heuristics for solving last-mile delivery problems. Boualamia et al. (2023) replaced the roulette wheel selection in standard ALNS by Q-learning to obtain a balance between exploration and exploitation for the capacitated VRP. Additionally, Q-learning is also used for fitness evaluation, initialization, and parameter setting (Ji et al., 2021; Xu and Wei, 2023). It is known that the updating of a solution in most metaheuristic algorithms is random, resulting in a heavy computational burden at each iteration. However, there is a lack of research investigating the utilization of Q-learning to overcome this limitation. Hence, in our approach, we employ Q-learning to determine the update rate for altering solutions in QALNS, another modification being the replacement of the roulette wheel selection mechanism.

On the basis of the above discussion, the research gaps found in the literature include: (*i*) a lack of joint optimization of task assignment and route planning problem for healthcare pickup and delivery logistics; (*ii*) a lack of consideration for matching the functional characteristics of heterogeneous UAVs with logistical task requirements; and (*iii*) the potential for exploration of the improvement role of Q-learning in metaheuristics. Therefore, our research defines the HUTA-PDP in the context of healthcare logistics setting and proposes a corresponding MILP model to address the first point (*i*). For the second point (*ii*), we consider the "Tasks-UAVs-Centers" combinations by matching characteristics of UAVs, such as payload, fly range, height, and speed with potential tasks; and we finally design a hybrid metaheuristics to explore the new role of Q-learning in ALNS for the third point (*iii*).

3. Problem description and mathematical formulation

This section defines HUTA-PDP for emergency healthcare scenarios. We first present a MILP model with the objective of maximizing the total profits derived from the execution of a set of PD tasks. The objective of this problem is to optimize both task assignment and PD routing operations for a fleet of UAVs. The HUTA-PDP framework includes a pre-operational phase and two optimization stages as shown in Fig. 1.

In the pre-operational phase, we categorize UAVs by assessing their performance metrics, including flight range, payload capacity, height capabilities, and their speed. At the same time, we analyze the characteristics of emergency logistics missions related to healthcare, such as demand types, priority levels, geographic locations, and other pertinent factors. Taking into account these two sets of criteria, we create a comprehensive framework to match UAVs and locations with the diverse requirements of PD missions, resulting in the generation of viable "Tasks-UAVs-Centers" combinations.

In the task assignment stage, we select specific combinations from the pre-established options for each task. The aggregated set of demands for each group of customers is treated as a single task and is exclusively assigned to a single healthcare center and an UAV within a designated time window. This tactical assignment process yields a streamlined list of available "Tasks-UAVs-Centers" combinations.

Moving to the route planning stage, we formulate an optimal UAV PD routing plan based on the task assignment results. The goal is the maximization of the total profit, with consideration of various constraints, such as time windows, payload restrictions, and the flying ranges required in emergency situations. This two-stage approach ensures systematic orchestration of the entire logistics process, effectively balancing the challenges of task assignment with the development of efficient UAV routes for emergency healthcare operations.

It should be noted that considering the limited healthcare resources in emergency situations, we must account for the possibility of not being able to meet all demands within the specified time. Hence some orders may be unserved. Moreover, given the urgency of completing demand in emergency situations, we have defined distinct time windows for orders of different emergency priority levels. For higher-priority demands, the time windows are relatively tight, while regular medical orders may be delivered at any time. The objective of maximizing total profits can help completing more tasks with higher emergency priorities. With the remaining inventory, unserved orders with lower profits can still be served via different and relatively slower services. Tasks that cannot be served do not earn profit and can also be left for next planning cycle with a higher priority.

The HUTA-PDP can be formulated as follows. The emergency healthcare logistics network $G = (V, A)$ is composed of $|M|$ healthcare centers and $|N|$ groups with multiple number of patients, where the vertex set $V = M \cup N$, and A is an arc set. Within the network, there are $|R|$ different categories of healthcare resources for delivery and a R_0th category for pickup, i.e., $R = \{1, 2, \ldots, |R|, R_0\}$. A fleet of $|U|$ types of heterogeneous UAVs, indexed by k, each with c^k compartments with a payload capacity of b^k, flight range w^k, flight height h^k and flight speed as v^k, is located in healthcare centers. The set of UAVs of type u in center m can be represented as K_m^u, and $K = K_1^1 \cup K_2^1 \cup \cdots \cup K_{|M|}^{|U|}$. For each group $j \in N$, the altitude is denoted as g_j, the profit of the unit delivery and pickup task of the rth type of healthcare item is denoted as p_{jr}, and the service end time as T_{jr}. The available stock of the rth type of healthcare items in the center m is s_{mr}, and the quantity ordered of the rth item by the customer group j is d_{jr}.

At the task assignment stage, we collect and identify features of the healthcare center set M, UAV set K and customer group set N, specifying the responsible center and UAV for each task. For example, this can be selecting UAVs with higher speeds for emergency tasks, and using UAVs capable of flying at high altitudes for customers located in hilly areas. Next, in the second stage, UAV k departs from center m and transports different types of resources. It travels a distance of l_{ij} with a speed of v^k, arrives at the group j at time t_j^k, and spends a period of time t_0 to deliver or pick up packages. If the group j has a request for the pickup task, UAV k collects the weighing package d_{jR_0}. However, if there is no pickup request, UAV k continues to deliver the remaining healthcare items until all delivery tasks are completed or returns to the center m before running out of battery energy.

To achieve optimal scheduling of limited resources in emergency healthcare, the HUTA-PDA model is formulated to coordinate heterogeneous UAVs within a limited emergency time window to deliver various medical items and collect blood samples. Table 1 lists the sets, parameters and decision variables used in the model.

The MILP model of the HUTA-PDP is given as follows.

$$\text{maximize} \sum_{k\in K}\sum_{j\in N}\sum_{r\in R} p_{jr} d_{jr} y_{jr}^k \tag{1}$$

$$\sum_{i\in V} x_{iv}^k = \sum_{j\in V} x_{vj}^k = 1, k \in K, v \in V \tag{2}$$

$$x_{ij}^k + x_{ji}^k \le 1, k \in K, i, j \in M \text{ or } i, j \in N \tag{3}$$

$$\sum_{k\in K} y_{jr}^k \le 1, j \in N, r \in R \tag{4}$$

$$\sum_{r\in R} y_{jr}^k \le M \sum_{i\in V} x_{ij}^k, k \in K, j \in N \tag{5}$$

$$z_i^k - \sum_{r=1}^{|R|} y_{jr}^k + y_{jR_0}^k - M(1 - x_{ij}^k) \le z_j^k, k \in K, i \in V, j \in N \tag{6}$$

$$\sum_{j\in N}\sum_{r=1}^{|R|} y_{jr}^k \le \sum_{i\in M} z_i^k, k \in K \tag{7}$$

$$t_i^k + l_{ij}/v^k + t_0 - M(1 - x_{ij}^k) \le t_j^k, k \in K, i, j \in V \tag{8}$$

$$t_j^k \le T_{jr} + M(1 - x_{ij}^k), k \in K, j \in N, r \in R \tag{9}$$

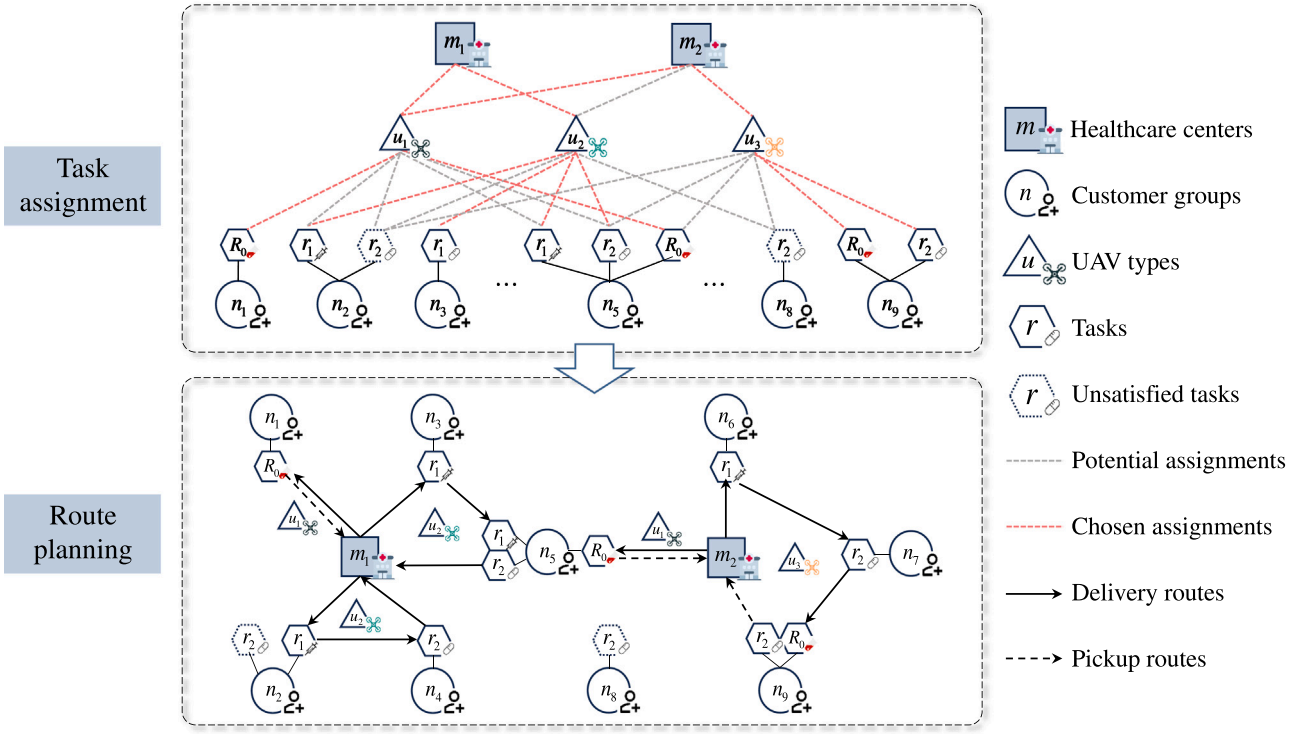


Fig. 1. HUTA-PDP framework in emergency healthcare.

Table 1
Sets, parameters, and decision variables for the HUTA-PDP.

Notations	Definitions
Sets	
G	Healthcare logistics network, $G = (V, A)$.
V	Set of all vertices, $V = M \cup N$, $i, j \in V$.
A	Set of arcs, indicated by (i, j), $i, j \in V$.
M	Set of all healthcare centers, $M = \{1, 2, \ldots, \lvert M\rvert\}$, $m \in M$.
N	Set of all customer groups, $N = \{1, 2, \ldots, \lvert N\rvert\}$, $n \in N$.
R	Set of categories of resources, $R = \{1, 2, \ldots, \lvert R\rvert, R_0\}$, where $\{1, 2, \ldots \lvert R\rvert\}$ for delivery and R_0 for pickup, $r \in R$.
K	Set of all UAVs, $K = \{1, 2, \ldots, \lvert K\rvert\}$, $k \in K$.
U	Set of UAV types, $U = \{1, 2, \ldots, \lvert U\rvert\}$, $u \in U$.
K_m^u	Set of UAVs of type u in center m, $K_m^u = \{1, 2, \ldots, \lvert K_m^u\rvert\}$, $k_m^u \in K_m^u$, and $K = K_1^1 \cup K_2^1 \cup \cdots \cup K_{\lvert M\rvert}^{\lvert U\rvert}$.
Parameters	
p_{jr}	Unit profit of satisfying the r^{th} type of items in group j ($j \in N, r \in R$).
s_{mr}	Available stock of the r^{th} type of items in the center m ($m \in M, r \in R$).
d_{jr}	Quantity ordered of the r^{th} type of items by group j ($j \in N, r \in R$).
l_{ij}	Travel distance of arc (i, j) ($i, j \in V$).
g_j	Altitude of vertices j ($j \in V$).
t_0	Service time at each group.
c^k	Available compartments of UAV k ($k \in K$).
b^k	Maximum payload capacity of UAV k ($k \in K$).
w^k	Maximum fly range of UAV k ($k \in K$).
h^k	Maximum fly height of UAV k ($k \in K$).
v^k	Maximum fly speed of UAV k ($k \in K$).
T_{jr}	Available service end time of the r^{th} type of items in group j ($j \in N, r \in R$).
Decision variables	
x_{ij}^k	1 if arc (i, j) is traveled by UAV k, 0 otherwise ($k \in K, i, j \in V$).
y_{jr}^k	1 if the r^{th} type of items in the group j is served by the UAV k, 0 otherwise ($k \in K, j \in N, r \in R$).
z_j^k	The number of loaded parcels of UAV k after leaving vertex j ($k \in K, j \in V$).
t_j^k	The arrival time of UAV k at vertex j ($k \in K, j \in V$).

$$\sum_{k\in K_m}\sum_{j\in N} d_{jr}y_{jr}^k \le s_{mr}, m \in M, r \in \{1,2,\dots,|R|\} \quad (10)$$

$$\sum_{k\in K_m^u}\sum_{j\in N} x_{ij}^k \le |K_i^u|, i \in M \quad (11)$$

$$g_j x_{ij}^k \le h^k, k \in K, i,j \in V \quad (12)$$

$$\sum_{j\in N}\sum_{r\in R} d_{jr}y_{jr}^k \le b^k, k \in K \quad (13)$$

$$z_j^k \le c^k, k \in K, j \in V \quad (14)$$

$$\sum_{i,j\in V} l_{ij}x_{ij}^k \le w^k, k \in K \quad (15)$$

$$x_{ij}^k = \{0,1\}, k \in K, i,j \in V \quad (16)$$

$$y_{jr}^k = \{0,1\}, k \in K, j \in N, r \in R \quad (17)$$

$$z_j^k \ge 0, k \in K, j \in V \quad (18)$$

$$t_j^k \ge 0, k \in K, j \in V, r \in R. \quad (19)$$

The objective function (1) maximizes the total profit, where the p_{jr} is the unit profit associated with each task. Constraints (2) ensure the flow balance of a route. Constraints (3) prevent UAVs from forming loops between every adjacent vertices. Constraints (4) guarantee that each task is performed at most once. Constraints (5) impose restrictions that maintain the relation between the number of UAVs attending customer group n and the number of tasks completed by a single UAV while undertaking multiple tasks for that particular group. Constraints (6) calculate the number of packages loaded onto UAV k after completing its pickup or delivery operation for group n. Constraints (7) mean that the assigned task for each UAV cannot exceed the total number of packages it can carry. Constraints (8) indicate the temporal relationships of UAVs arriving at adjacent vertices. Constraints (9) ensure that UAVs are not allowed to provide services after the time window is closed. Constraints (10) guarantee that the total quantity of delivered items from each healthcare center does not exceed the available stock. Constraints (11) ensure that the total available number of UAVs is not exceeded. Constraints (12) guarantee that UAVs cannot fly above their maximum height. Constraints (13) ensure that the weight limit of the payload does not exceed the maximum payload of UAV compartment. Constraints (14) limit the number of packages that are loaded. Constraints (15) limit the travel range of UAV k. Constraints (16)–(19) define the domains of the decision variables.

The HUTA-PDP is an extension of the TOP and is therefore NP-hard. Gurobi solver has been successfully applied and verified to obtain an optimal solution of the small-scale MILP. To solve large-scale HUTA-PDP instances, we have designed a metaheuristic.

4. A Q-learning ALNS algorithm

We have developed a hybrid metaheuristic for an ALNS heuristic in which Q-learning is integrated. We describe the framework of the metaheuristic and its main components, including the initial solution generation, destroy and repair operators, improved Q-learning-based selection mechanism, and acceptance criteria for solutions.

The framework of the proposed QALNS algorithm is presented in Algorithm 1. The algorithm starts with an initialization stage to generate a feasible solution as the current solution. Then, the destruction and repair mechanism of the feasible solution is continuously performed to explore better solutions within the limit of the number of iterations and of the maximum allowed computation time. In this process, the Q-learning component serves two roles: (*i*) replacing the roulette method for choosing the destroy and repair operators, and (*ii*) adjusting the destroy–repair rate, as shown in Fig. 2. The following subsections provide a detailed description of these components.

Algorithm 1 The QALNS procedure

Input: Initial solution ς; initial temperature τ_0, minimum temperature τ_{min} and cooling rate ρ; initial Q-table 1 and Q-table 2; programming end conditions $Iter_{max}$, $Time_{max}$
Output: Best solution ς^*
Let current best-found solution $\varsigma^* \leftarrow \varsigma$, current temperature $\tau \leftarrow \tau_0$, $Iter \leftarrow 0$
while $Iter \le Iter_{max}$ and $Time_{now} \le Time_{max}$ **do**
  while $\tau \ge \tau_{min}$ **do**
    Select a destroy-repair action based on Q-table 1
    Determine a destroy-repair rate based on Q-table 2
    Perform destroy and repair operations to get a new solution ς'
    if $Obj(\varsigma') \ge Obj(\varsigma)$ **then**
      $\varsigma \leftarrow \varsigma'$
      if $Obj(\varsigma') \ge Obj(\varsigma^*)$ **then**
        $\varsigma^* \leftarrow \varsigma'$
      end if
    else
      if simulated annealing acceptance criteria is met **then**
        $\varsigma \leftarrow \varsigma'$
      end if
    end if
    Update the current state, reward, Q-table 1 and Q-table 2
    $\tau \leftarrow \rho * \tau_0$
  end while
  $Iter \leftarrow Iter + 1$, $\tau \leftarrow \tau_0$
end while

4.1. Initialization

We generate an initial feasible solution by performing several steps. First, the sequence of tasks is randomly generated, taking into account constraints such as available "Tasks-UAVs-Centers" combinations, UAV feature capacity, available number of healthcare resources and UAVs, etc. Then, the corresponding sequence of vertices is generated, and the tasks of the same vertex are made adjacent to each other, avoiding subloops. Next, we remove the selected "Tasks-UAVs-Centers" combinations from the task set and calculate the remaining inventory. Finally, the arrival time of a solution is obtained based on the generated vertex sequence, geographic information, and UAV flight parameters. The complete solution structure for the HUTA-PDP is shown in Fig. 3.

Although the initialization procedure of the proposed QALNS algorithm is random, the construction of the initial solution exhibits considerable efficacy due to the strict rules. On the other hand, the ALNS algorithm can easily recover from a poor initial solution and does not require the initial solution to be of excellent quality (Demir et al., 2012).

4.2. Destroy–repair operators

The destroy and repair operators are used as an effective technique to update a solution. The destroy operators disrupt the current solution by removing several elements, while the repair operators add elements to generate a new feasible solution. Although many problem-specific destruction and repair operators, including sequence-based operators, priorities, and historical information, have been proposed (Voigt, 2024), they do not align well with the characteristics of the HUTA-PDP. In this study, we designed three destroy operators and three repair operators to solve the HUTA-PDP. Beyond basic random operators, we introduce a new feature where some pickup and delivery tasks may occur within the same group. To efficiently optimize UAV operations, we propose group-based operators. Furthermore, given the profit-oriented nature of the HUTA-PDP model, we also introduce

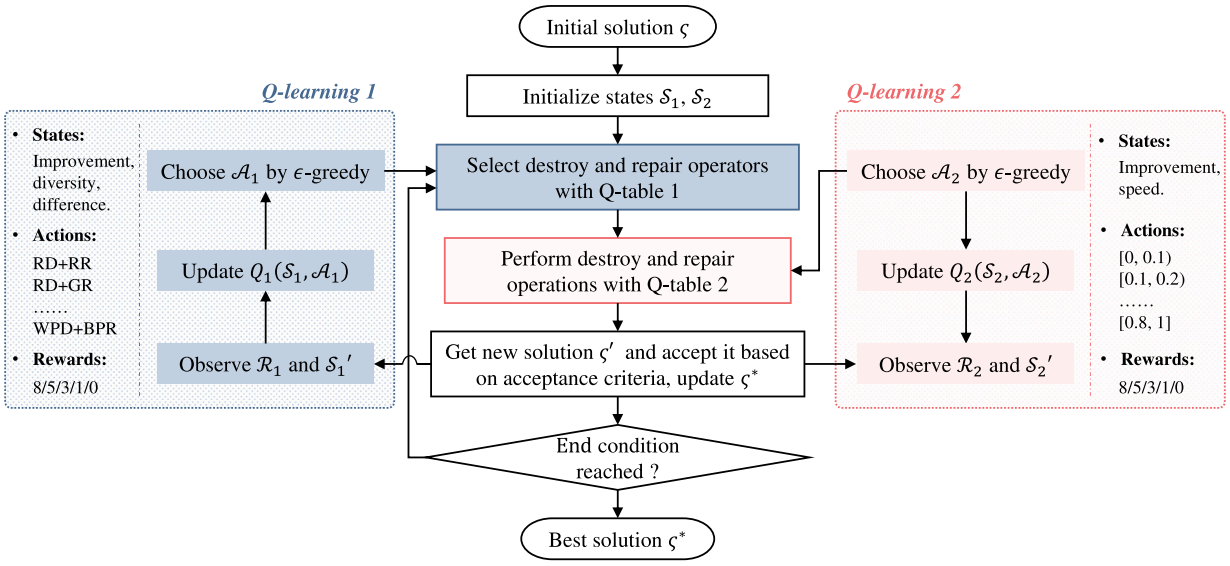


Fig. 2. Framework of the QALNS algorithm.

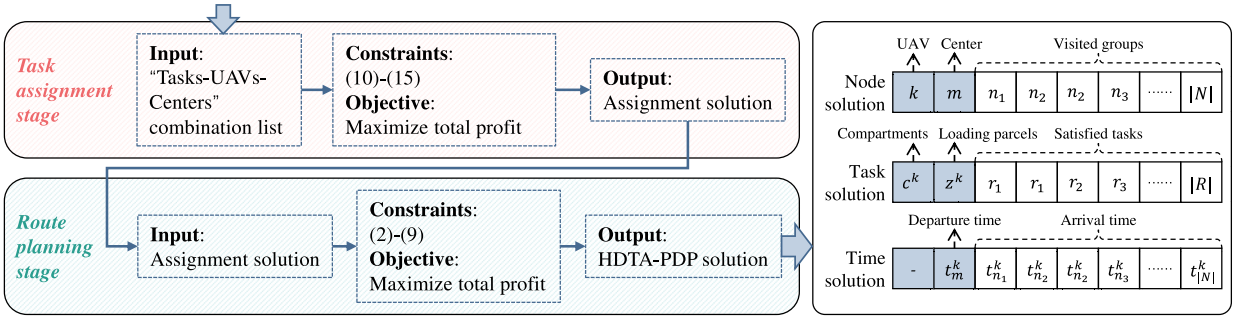


Fig. 3. Solution generation.

profit-based operators. All operators are performed in task sequences. The corresponding vertex sequences and time sequences are generated similarly to the initialization process.

4.2.1. Destroy operators

We now describe the destroy operators used in the QALNS algorithm.

Random Destroy (RD): This operator randomly selects multiple tasks for removal from the task list for each UAV distribution. Subsequently, the serial numbers of these deleted tasks are incorporated into the residual task list, denoted as $Task_{remain}$, with their associated resource quantities concurrently appended to the existing inventory, Inv_{remain}. This process is followed by an update of both the vertex and the time lists. The RD operator helps improve the diversity of updated solutions.

Group-Based Destroy (GD): This operator focuses on groups for task removal. It involves the selection of one or more groups accessed by the UAV k, followed by the removal of all serial numbers associated with the selected groups from the task list. For example, if the group vertex n_1 is chosen, all tasks associated with n_1 are consequently deleted, whilst tasks pertaining to other tasks remain unaffected.

Worst-Profit Destroy (WPD): This operator iteratively removes low-profit tasks. It randomly generates the number of tasks to be deleted, and then deletes multiple tasks with the lowest profits in descending order of each task, from lowest to highest. After deleting the task, the unsatisfied tasks $Task_{remain}$ and the remaining inventory Inv_{remain} are recomputed. The list of groups visited by this UAV and the arrival times are generated again following the operational steps in the initialization stage.

4.2.2. Repair operators

We now introduce three repair operators.

Random Repair (RR): This operator initially computes the additional number of delivery and pickup tasks that can be accommodated, considering the capacity to transport the UAV task k and the already scheduled tasks. It then randomly selects tasks from the remaining pool of tasks $Task_{remain}$, ensuring that the total task conform to the UAV k's parcel capacity constraint. Additionally, the remaining inventory Inv_{remain} in each healthcare center is taken into account. Finally, an updated and extended task list, as well as its corresponding group vertex list and time list are generated.

Group-Based Repair (GR): This operator inserts tasks based on the group vertices already matched by the current UAV k. For a non-empty task list, the operator randomly selects groups traversed by UAV k and picks additional task serial numbers associated with chosen groups from the remaining task pool $Task_{remain}$ to insert into k's task list. For example, if n_2 is the selected customer group vertex, then the tasks inserted are all associated with n_2. For an empty task list, the operator can randomly select several groups and insert tasks related to these chosen groups, and the time list is subsequently generated afterwards.

Best-Profit Repair (BPR): The operator sorts tasks from the remaining pool of tasks $Task_{remain}$ in descending order of profits and selects a random number of top tasks to insert into the list of tasks of this UAV k. At the same time, reordering is performed to ensure that tasks from the same customer group are adjacent to avoid duplicate visits. It then checks whether the inserted tasks comply with the k's capability constraints and the center's inventory constraints. Based on this, the vertex solution and the arrival time solution of k are generated.

4.3. Q-learning mechanism

In the standard ALNS framework, the roulette wheel method selects the destroy or repair operator based on the weights obtained in the previous iterations. This may reduce diversity, leading to a fall into a local optimum (Boualamia et al., 2023). However, particularly for larger-scale problems, traversing all elements for destroy and repair operations greatly diminishes convergence speed and solution quality (Pan et al., 2021). Therefore, we apply Q-learning in two ways: operator selection and operation rate determination. The steps are shown in Algorithm 2.

Algorithm 2 The Q-learning mechanism

Input: Initial $Q(S, \mathcal{A})$, learning rate α, discount factor γ, initial exploration rate ϵ_0, minimum exploration rate ϵ_{min}, decay rate λ
Output: Policy $\pi^*(S) = \arg\max_{\mathcal{A}} Q(S, \mathcal{A})$
for each episode **do**
  Initialize S
  while S is not terminal **do**
    if $random\_number > \epsilon$ or $Q(S, \mathcal{A}) = 0, \forall \mathcal{A}$ **then**
      Randomly choose an action $\mathcal{A}$ from S
    else
      $\mathcal{A} \leftarrow \arg\max Q(S, \mathcal{A})$
    end if
    Take action $\mathcal{A}$, observe reward $\mathcal{R}$ and new state S'
    $Q(S, \mathcal{A}) \leftarrow Q(S, \mathcal{A}) + \alpha[\mathcal{R} + \gamma \max_{\mathcal{A}} Q(S', \mathcal{A}) - Q(S, \mathcal{A})]$
    $S \leftarrow S'$
    $\epsilon \leftarrow \max(\epsilon_{min}, \epsilon_0 \times \lambda)$
  end while
end for

4.3.1. Operator selection

Q-learning is first deployed to dynamically adjust the weights of the operators with feedback information and enhance the evolution process of ALNS. This operator replaces the roulette wheel selection process in the original ALNS algorithm, where the weights of the destroy and repair operators were updated using the weight update factor β.

QL1-State: We define a set of states to guide the algorithm in selecting the most appropriate destroy and repair operators in the search, as shown in Table 2. The first five are adapted or inspired by Kallestad et al. (2023), whereas the last two are new. All states are characterized by three types of indicators: improvement indicators, diversity indicators, and difference indicators. The improvement indicators includes *Best_improved*, *Best_equal*, *Previous_improved*, *Current_accept*, which are used to measure the improvement of the objective values of the new solution compared to the current best solution and the previous solution. Diversity metrics include *Be_changed* and *First_generated*, used respectively to indicate whether the new solution has changed compared to the previous generation and whether the new solution has never appeared before. The difference indicator, i.e., *Difference_best*, serves to calculate the distance between the current solution and the best-found solution, whose value is a decimal within the range of $[0, 1]$. Considering that solutions closer to the best-found solution should be treated with more importance in terms of their state and action, we discretize its possible values into five different-sized intervals: $[-\infty, 0)$, [0, 0.1), [0.1, 0.3), [0.3, 0.6), [0.6, 1]. These three categories of metrics are combined to generate a total of over 60 states. For example, $S(Best\_improved, Be\_Changed, [-\infty, 0))$, $S(Current\_accept, First\_generated, [0.6, 1])$, etc. Taking into account that some of these metrics are mutually exclusive, such as *Best_improved* and non-negative intervals, or *Current_accept* and negative intervals, after removing these, the final number of states is reduced to 30.

QL1-Action: The actions in our QALNS are the same as the destroy–repair operations introduced in Section 4.2. Three destroy operators and three repair operators are combined with each other, resulting in a total of nine possible actions. For a single solution, we perform only one type of destroy operation and one type of repair operation.

QL1-Reward: The reward function reflects the immediate feedback received by the agent after taking a specific destroy–repair action and transitioning to a new state. In our study, we propose a reward piecewise function, which is mainly determined according to the promotion index, that is

$$\mathcal{R}_{QL1} = \begin{cases} 8 & \text{if } Obj(\varsigma') > Obj(\varsigma^*) \\ 5 & \text{if } Obj(\varsigma') = Obj(\varsigma^*) \\ 3 & \text{if } Obj(\varsigma) < Obj(\varsigma') < Obf(\varsigma^*) \\ 1 & \text{if } \varsigma' \text{ is accepted} \\ 0 & \text{if } \varsigma' \text{ is not accepted.} \end{cases} \tag{20}$$

This reward function encourages the agent to find a better solution, as it gives a higher reward. However, even if the new solution only meets the acceptance criteria without improvement, we still provide a small reward to enhance the diversity of solutions.

4.3.2. Operation rate determination

In the initialization phase, we assume that each solution is composed of multiple UAV lists. It is believed that updating all UAV lists to be highly time-consuming (Demir et al., 2012). However, for different environments and solution sizes, we cannot manually specify a best execution interval. Applying Q-learning to the return optimal interval policy based on the current environment provides a better solution.

QL2-State: Different operation rates have an impact on the optimality of solutions. It is self-evident that this proportion also affects the computational duration of generating new solutions, as noted by $P(Iter)$. Based on this, we define two indicators for setting the states in the second Q-learning mechanism: objective improvement and computational speed. For improvement, it is labeled as *Best* if $Obj(\varsigma') > Obj(\varsigma^*)$, as *Better* if $Obj(\varsigma) < Obj(\varsigma') \leq Obf(\varsigma^*)$. Otherwise, it is labeled as *Worse*. Similarly, the speed is labeled as *Faster* if $P(Iter') < P(Iter)$ and as *Slower* if $P(Iter') \geq P(Iter)$.

QL2-Action: We set the operation intervals as [0, 0.1), [0.1, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8) and [0.8, 1]. Thus, we obtain a set of six actions. Before each destroy operation, we randomly generate a proportion from the selected action interval, and perform the operation on the UAV lists corresponding to this proportion. Within each iteration, the operation interval remains constant, but after each update of temperature τ in simulated annealing, the specific proportion is randomly generated.

QL2-Reward: The reward function are set according to the states defined above, that is

$$\mathcal{R}_{QL2} = \begin{cases} 8 & \text{if } Obj(\varsigma') > Obj(\varsigma^*) \text{ and } P(Iter') < P(Iter) \\ 5 & \text{if } Obj(\varsigma') > Obj(\varsigma^*) \text{ and } P(Iter') \geq P(Iter) \\ 3 & \text{if } Obj(s) < Obj(\varsigma') \leq Obf(\varsigma^*) \text{ and } P(Iter') < P(Iter) \\ 1 & \text{if } Obj(s) < Obj(\varsigma') \leq Obf(\varsigma^*) \text{ and } P(Iter') \geq P(Iter) \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

4.4. Acceptance criteria

The QALNS algorithm employs acceptance criteria to maximize the objective of the HUTA-PDP. This is based on the disparity in objective values between the new solution, denoted as ς', and the preceding solution, denoted as ς. The corresponding formula is determined as follows.

$$P\_accept = \begin{cases} 1 & \text{if } Obj(\varsigma') > Obj(\varsigma) \\ e^{[Obj(\varsigma') - Obj(\varsigma)]/\tau} & \text{if } Obj(\varsigma') \leq Obj(\varsigma) \end{cases} \tag{22}$$

If the new solution ς' is better than the previous solution ς, then it will always be accepted. The worse solution ς' still has a chance of being accepted with a probability of $e^{[Obj(\varsigma') - Obj(\varsigma)]/\tau}$, where the τ is current temperature.

Table 2
State description for QL1-table.

Indicator	Description
Best_improved	1 if $Obj(\varsigma') > Obj(\varsigma^*)$, 0 otherwise.
Best_equal	1 if $Obj(\varsigma') = Obj(\varsigma^*)$, 0 otherwise.
Previous_improved	1 if $Obj(\varsigma) < Obj(\varsigma') < Obf(\varsigma^*)$, 0 otherwise.
Current_accept	1 if ς' was accepted but $Obj(\varsigma') \leq Obf(\varsigma)$, 0 otherwise.
Be_changed	1 if ς' was changed from the previous solution but has been generated, 0 otherwise.
First_generated	1 if ς' has never been generated before, 0 otherwise.
Difference_best	The difference between the values of ς' and ς^*.

Table 3
Parameters used in ALNS and QALNS.

Parameters	Benchmark experiments		HUTA-PDP experiments	
	ALNS	QALNS	ALNS	QALNS
Termination condition	10,000 iterations	10,000 iterations	1,000 iterations or 900 s	1,000 iterations or 900 s
SA initial temperature τ_0	1,000	1,000	100	100
SA minimum temperature τ_{min}	0.01	0.01	0.01	0.01
SA cooling rate ρ	0.99	0.99	0.99	0.99
Weight update factor β	0.5	–	0.5	–
Learning rate α	–	0.3	–	0.3
Discount factor γ	–	0.9	–	0.9
Initial exploration rate ϵ_0	–	0.99	–	0.99
Minimum exploration rate ϵ_{min}	–	0.1	–	0.1
Decay rate λ	–	0.99	–	0.99
Number of states S_{QL1}	–	9×3	–	3×3
Number of states S_{QL2}	–	6	–	6
Rewards $\mathcal{R}_{QL1}$	–	8/5/3/1/0	–	8/5/3/1/0
Rewards $\mathcal{R}_{QL2}$	–	8/5/3/1/0	–	8/5/3/1/0

5. Computational experiments

We now present the detailed results of various experiments on the performance of the QALNS algorithm. In Section 5.1, we set the number of search iterations to 10,000 for the comparison of the QALNS with the best known solutions on the benchmark instances of Li and Lim (2008). In Section 5.2, we solve the MILP model, original ALNS and QALNS on four sets of different scale HUTA-PDP instances, and we analyze the performance of the QALNS algorithm and the sensitivity of the parameters. All computations were carried out on a 3.2 GHz computer with 16.0 GB of RAM. Where applicable, the MILP was solved using Gurobi 10.0.3, and the main algorithm and comparison heuristics were programmed in Python 3.11. The parameter settings of the original ALNS and of our QALNS algorithm are shown in Table 3.

5.1. Experiments on PDPTW benchmark instances

Taking into account the similarity of HUTA-PDP with PDPTW, our algorithm was run with a fixed configuration on PDPTW benchmark instances from Li and Lim (2008) to demonstrate its robustness across different sets of instances. Since PDPTW aims to minimize the total distance traveled while satisfying all demands, we modify the objective function and constraints of the designed QALNS to solve PDPTW. Table 4 reports the results obtained on the PDPTW benchmark dataset of 100, 200, 400, 600 and 800 tasks. For each instance, we provide the total traveled distance (Dis) and the number of used vehicles (#Veh) for the best-known solutions, average results of five runs and the best results obtained by QALNS. We also provide the CPU time and iteration numbers (#Iter) performed. The last two columns of the table show the percentage improvement (Imp) of the QALNS algorithm from those of the best-known solutions, where $\Delta^{Dis}_{Avg} = (Dis_{Avg} - Dis_{Best-known})/Dis_{Best-known}$ and $\Delta^{Dis}_{Best} = (Dis_{Best} - Dis_{Best-known})/Dis_{Best-known}$.

As shown in Table 4, QALNS obtains best-known solutions for a majority of the benchmark instances, showing a robust average performance. For some 200-task, 400-task, 600-task and 800-task instances, QALNS even found new best-known solutions, which are highlighted in bold. The average improvement for best-known solutions obtained by QALNS is 1.26% for the 200-task instances, 6.50% for the 400-task instances, 4.69% for the 600-task instances, and 3.43% for the 800-task instances. Regarding the average performance of QALNS among five runs, the improvement is 0.04% for the 200-task instances, 0.35% for the 400-task instances, and 0.15% for the 600-task instances. The termination condition limits the ability to find the best-known solutions for all large-scale instances, but the solution values of the 800-task instances are on average only 0.28% below the best-know values. In terms of CPU time, all 100-task instances are solved quickly within one minute. The 200-task instances need about two or three minutes to solve in most cases. The 400-task instances can still be solved within 10 min. For the larger scale instances, the variation in CPU time is much larger, which is around 15 min for 600-task instances and 25 min for 800-task instances. Although we have incorporated two Q-learning method frameworks into the original ALNS structure, the actual computational efficiency has not been weakened. The search policy based on Q-learning makes the algorithm more viable for different practical-scale problems. Given that the problem defined in our paper is a variant of the PDPTW, we believe that our QALNS algorithm should also exhibit a good performance on the HUTA-PDP model.

5.2. Experiments on HUTA-PDP instances

We consider a set of randomly generated instances of four sets of different sizes based on real geographic data from a county in Sichuan Province, China, as shown in Fig. 4. The geodesic distance matrix based on the real coordinates between two locations, the altitude of vertices was obtained using a DEM elevation map. Each instance, named CH_M_N_K, involves M centers, N customer groups, and K UAVs of each type per center. For instances of the same set, the numbers of centers and customers are fixed, but the number of UAVs increases. Each group contains multiple individual customers with their randomly generated tasks. This results in five types of PD tasks with different priorities in each group. Each completion of the four emergency delivery tasks is assigned a unit profit of 5, 3, 2, and 1 based on their urgency priority, while the profit for every single pick up task is set at 10. We set all emergency task orders to occur between 6:00 h and 8:00 h on one day, with varying service durations ranging from two to

Table 4
Results of the QALNS on PDPTW benchmark instances.

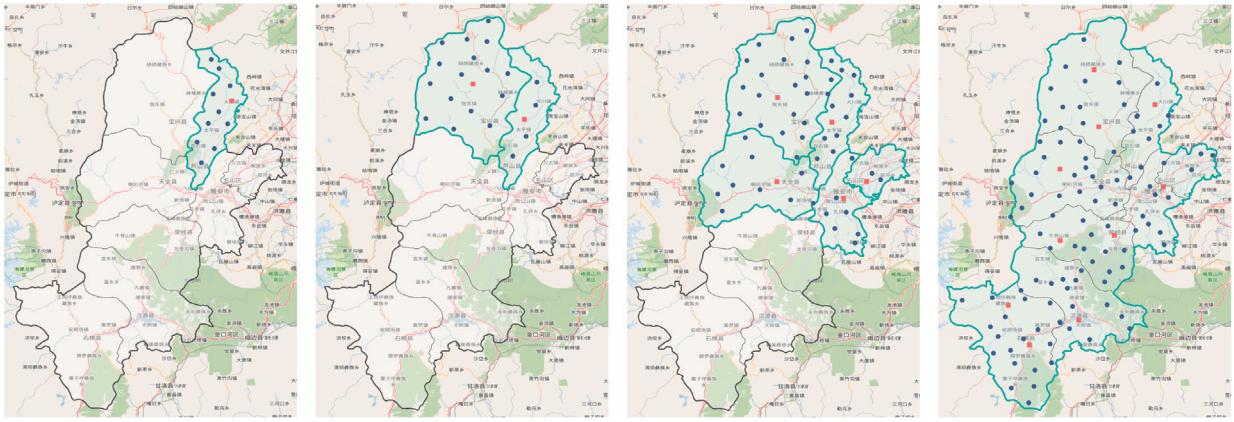
Instances	Best-known[a]		Average results by QALNS				Best solutions by QALNS				Imp (%)	
	Dis	#Veh	Dis	#Veh	Time (s)	#Iter	Dis	#Veh	Time (s)	#Iter	Δ^{Dis}_{Avg}	Δ^{Dis}_{Best}
100 tasks												
LR1_0_1	1,650.80	19	1,650.80	19	3.59	235	1,650.80	19	2.84	188	0.00	0.00
LR1_0_2	1,487.57	17	1,487.57	17	14.82	1,307	1,487.57	17	13.62	1,287	0.00	0.00
LR1_0_3	1,292.68	13	1,292.68	13	16.14	1,474	1,292.68	13	15.47	1,326	0.00	0.00
LR1_0_4	1,013.39	9	1,013.39	9	57.97	4,319	1,013.39	9	56.55	3,878	0.00	0.00
LR1_0_5	1,377.11	14	1,377.11	14	18.80	2,146	1,377.11	14	16.94	1,864	0.00	0.00
LR1_0_6	1,252.62	12	1,252.62	12	26.08	2,549	1,252.62	12	5.48	1,141	0.00	0.00
LR1_0_7	1,111.31	10	1,111.31	10	37.33	2,490	1,111.31	10	17.29	2,472	0.00	0.00
LR1_0_8	968.97	9	968.97	9	36.09	2,073	968.97	9	33.97	2,033	0.00	0.00
LR1_0_9	1,208.96	11	1,208.96	11	62.57	5,109	1,208.96	11	54.43	4,711	0.00	0.00
LR1_1_0	1,159.35	10	1,159.35	10	49.32	2,790	1,159.35	10	43.76	2,540	0.00	0.00
Average	1,252.28	12	1,252.28	12	32.27	2,449	1,252.28	12	26.03	2,144	0.00	0.00
200 tasks												
LC1_2_1	2,704.57	20	2,704.57	20	52.58	3,075	2,704.57	20	43.11	2,432	0.00	0.00
LC1_2_2	2,764.56	19	2,764.56	19	163.53	4,648	2,764.56	19	123.89	3,491	0.00	0.00
LC1_2_3	2,772.20[b]	–	2,802.05	18	368.00	7,785	2,786.70	18	134.28	3,793	−1.08	−0.52
LC1_2_4	2,693.41	17	2,693.93	18	426.28	6,013	**2,678.70**[d]	**18**	**401.65**	**6,389**	−0.02	**0.55**
LC1_2_5	2,702.05	20	2,702.05	20	184.57	4,264	2,702.05	20	137.09	3,927	0.00	0.00
LC1_2_6	2,701.04	20	2,701.04	20	56.40	1,643	2,701.04	20	50.60	1,547	0.00	0.00
LC1_2_7	2,701.04	20	2,701.04	20	98.15	2,902	2,701.04	20	93.83	2,582	0.00	0.00
LC1_2_8	2,689.83[b]	–	2,689.83	20	542.32	6,997	2,689.83	20	453.67	6,055	0.00	0.00
LC1_2_9	2,724.24	18	2,724.24	18	623.53	7,543	2,724.24	18	301.52	5,585	0.00	0.00
LC1_2_10	2,942.13	17	**2,897.76**	**19**	**743.45**	**7,824**	**2,884.24**[d]	**19**	**729.48**	**8,109**	**1.51**	**1.97**
Average	2,739.51	19	2,738.11	19	325.88	5,269	2,733.70	19	246.91	4,391	0.04	0.20
400 tasks												
LC1_4_1	7,152.06	40	7,152.06	40	403.54	4,706	7,152.06	40	257.22	3,035	0.00	0.00
LC1_4_2	8,007.79	38	**7,851.78**	**40**	**754.30**	**6,075**	**7,495.13**[d]	**41**	**456.33**	**3,908**	**1.95**	**6.40**
LC1_4_3	8,678.23	32	**8,476.27**	**40**	**1643.04**	**8,398**	**8,231.91**[d]	**40**	**1,113.83**	**6,635**	**2.33**	**5.14**
LC1_4_4	6,451.68	30	6,451.68	30	593.67	4,327	6,451.68	30	546.94	3,942	0.00	0.00
LC1_4_5	7,150.00	40	7,150.00	40	402.84	3,966	7,150.00	40	295.06	3,587	0.00	0.00
LC1_4_6	7,154.02	40	7,157.35	40	743.32	6,320	7,154.02	40	311.12	3,106	−0.05	0.00
LC1_4_7	7,149.43	40	7,149.43	40	632.79	7,524	7,149.43	40	424.42	5,025	0.00	0.00
LC1_4_8	8,305.42[c]	38	**7,893.06**	**40**	**603.56**	**4,566**	**7,645.41**[d]	**41**	**691.33**	**4,201**	**4.96**	**7.95**
LC1_4_9	7,451.20	36	7,451.20	36	362.80	3,591	7,451.20	36	358.76	3,341	0.00	0.00
LC1_4_10	7,850.22	34	8,294.99	40	501.17	4,021	8,207.76	39	332.68	2,816	−5.67	−4.55
Average	7,535.01	37	7,502.78	39	664.10	5,349	7,408.86	39	478.77	3,960	0.35	1.49
600 tasks												
LRC1_6_1	18,288.90	52	18,288.90	52	1,165.09	6,710	18,288.90	52	831.52	5,280	0.00	0.00
LRC1_6_2	16,515.41	43	16,515.41	43	1,292.44	7,426	16,515.41	43	751.48	5,569	0.00	0.00
LRC1_6_3	13,975.98	36	14,011.69	36	841.06	7,087	13,975.98	36	848.55	7,893	−0.26	0.00
LRC1_6_4	10,800.44	25	10,847.32	25	646.98	5,497	10,847.32	25	651.34	5,077	−0.43	−0.43
LRC1_6_5	17,463.94	45	17,624.89	52	1,081.18	7,099	17,604.69	52	838.73	5,420	−0.92	−0.81
LRC1_6_6	19,025.36	41	**18,235.28**	**48**	**881.17**	**6,494**	**18,133.30**[d]	**51**	**685.07**	**4,645**	**4.15**	**4.69**
LRC1_6_7	15,914.85	37	15,914.85	37	760.58	5,458	15,914.85	37	584.64	4,351	0.00	0.00
LRC1_6_8	15,317.63	33	15,317.63	33	969.05	7,034	15,317.63	33	1234.33	7,747	0.00	0.00
LRC1_6_9	15,344.64	33	15,344.64	33	713.81	5,168	15,344.64	33	1,050.03	6,622	0.00	0.00
LRC1_6_10	13,963.66	29	14,106.59	29	742.16	6,582	14,040.83	29	564.73	4,675	−1.02	−0.55
Average	15,661.08	37	15,620.72	39	909.35	6,455	15,598.36	39	804.04	5,728	0.15	0.29
800 tasks												
LRC1_8_1	32,252.28	66	32,252.28	66	1,505.87	8,193	32,252.28	66	1,252.92	7,397	0.00	0.00
LRC1_8_2	27,878.89	56	27,878.89	56	1,710.79	9,292	27,878.89	56	1,385.04	7,458	0.00	0.00
LRC1_8_3	24,371.95	48	24,371.95	48	968.44	6,123	24,371.95	48	1,025.20	7,038	0.00	0.00
LRC1_8_4	18,208.51	34	18,208.51	34	1,590.71	8,556	18,208.51	34	1,219.48	7,985	0.00	0.00
LRC1_8_5	31,169.16	58	31,439.42	60	1,877.95	9,467	31,270.41	58	1,501.58	7,891	−0.87	−0.32
LRC1_8_6	28,961.66	54	29,215.99	65	1,676.97	8,630	29,103.63	56	1,817.35	9,157	−0.88	−0.49
LRC1_8_7	28,768.40	50	29,435.05	61	1,390.09	7,860	29,010.82	62	1,607.64	9,095	−2.32	−0.84
LRC1_8_8	26,902.93	44	27,364.45	47	1,506.01	8,752	27,134.39	46	1,511.67	8,026	−1.72	−0.86
LRC1_8_9	24,854.96	44	24,854.96	44	1,302.11	7,313	24,854.96	44	1,142.52	6,789	0.00	0.00
LRC1_8_10	24,622.59	39	**23,882.62**	**40**	**1,498.80**	**8,134**	**23,776.90**[d]	**40**	**1,402.19**	**7,246**	**3.01**	**3.43**
Average	26,799.13	49	26,890.41	52	1,502.78	8,232	26,786.27	51	1,386.56	7,808	−0.28	0.09

[a] Provided by Li and Lim (2008).
[b] Provided by Ropke and Cordeau (2009).
[c] Provided by Christiaens and Vanden Berghe (2020).
[d] New best-known solutions.

six hours. The service time required for each task execution is 1,200 s. The inventories of healthcare centers may not always be sufficient, ranging approximately between 60% and 120% of total demand. There are five types of UAVs in our instances, and their parameters are shown in Table 5.

5.2.1. Results of QALNS on HUTA-PDP instances sets

Table 6 reports the results of instances for 10, 20, 50 and 100 customer groups. For Gurobi solver, we limit the solution time to one hour. The CPU time taken for these solutions in seconds and their upper bound (UB) of LP relaxation solution are also provided. For



(a) CH_1_10_K (b) CH_2_20_K (c) CH_5_50_K (d) CH_10_100_K

Fig. 4. Four sets of instances from China.

Table 5
UAV types and key parameters.

UAV type	Range w^k	Speed v^k	Height h^k	Payload b^k	Compartment c^k
Skycart Nimbus	160 km	36 m/s	1,000 m	15 kg	4
Wingcopter 198	110 km	40 m/s	5,000 m	5 kg	3
A2Z RDSX	50 km	24 m/s	4,800 m	3 kg	2
Freely ALTA X	60 km	20 m/s	4,500 m	16 kg	1
DJI Flycart 30	16 km	20 m/s	3,000 m	30 kg	1

two metaheuristic algorithms, we present their objective values, CPU times, iteration numbers (#Iter) performed, as well as the number of dispatched UAVs (#UAV) and the total duration (DU) in hours required to complete tasks. We run the QALNS algorithm five times with a limit of either 15 min or 1,000 iterations, depending on which criterion is met first. Other parameters are listed in Table 5. The columns Δ_G^{Obj} and Δ_A^{Obj} represent the percentage difference between the best-found objective values obtained by QALNS and the other comparison method. In detail, $\Delta_G^{Obj} = (Obj_{QALNS} - Obj_{Gurobi})/Obj_{Gurobi}$ and $\Delta_A^{Obj} = (Obj_{QALNS} - Obj_{ALNS})/Obj_{ALNS}$.

The table indicates that our QALNS algorithm produces high-quality solutions across all instances. Gurobi and QALNS can solve all the CH_1_10_K instances to optimality, while ALNS can only identify seven optimal solutions. However, the average CPU time of QALNS is 0.88 s, which is less than that of Gurobi (1.94 s) and ALNS (3.38 s), while also achieving an 70% reduction in the iteration number compared to the standard ALNS. For the CH_2_20_K instances, Gurobi experiences a significant increase in the number of nodes to explore, with 131,105 nodes after presolving in CH_2_20_6, and even more in the following instances. This results in a sharp increase in computational time, making it challenging to find an optimal solution within the one-hour limit. The QALNS algorithm consistently achieves superior solutions, with a 0.52% improvement over Gurobi and a 2.37% improvement over ALNS, while maintaining an average computation time of just 5.06 s. With the increase in instance size, the advantage of QALNS becomes even more pronounced. For CH_5_50_K instances, QALNS improves the solution quality by 9.75% compared to Gurobi, and by 6.80% to ALNS on average. Furthermore, QALNS typically requires only 102.91 s to achieve the best-found solution, saving 43.44% and 97.14% CPU time compared with ALNS and Gurobi, respectively. For the largest CH_10_100_K instances, Gurobi can solve only two instances within the given time limit. QALNS can still run quickly with an average CPU time of 341.19 s, resulting a 8.98% improvement over ALNS best solutions and saving 24.15% time on average.

It is also observed that within the same instance set, the quantity of available UAVs influences the algorithm's solution quality, particularly when comparing QALNS and ALNS. In scenarios where the number of UAVs is relatively limited compared to the volume of the task, QALNS demonstrates the ability to quickly identify higher-quality solutions. In contrast, although ALNS returns solutions within a modest number of iterations, they often converge to local optima. However, when there are a larger number of UAVs, the best-found solution is never unique. In such cases, the performance gap in solution quality between heuristics narrows, but QALNS still outperforms the alternative algorithms in convergence speed. Although some results in the table show that the number of UAVs used in the QALNS scheme is greater than that of Gurobi or ALNS, this study argues that as long as it is within the allowable range, more UAV deployments are worthwhile because they can generate higher profits in emergency healthcare situations.

5.2.2. Sensitivity analyses for the parameters of the QALNS algorithm

This section presents sensitivity analyses for the key parameters used in the QALNS algorithm, observing their impact on Q-learning convergence, the objective value, and CPU time. We give all solutions obtained among 10 runs with a termination condition of 1,000 iterations.

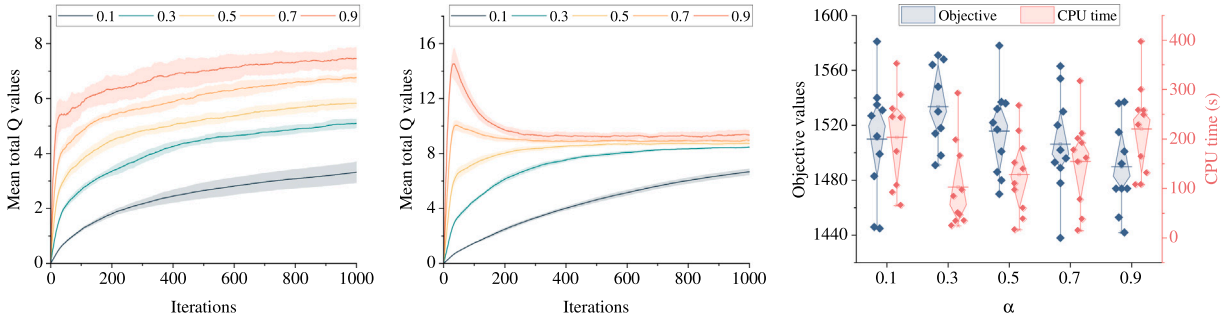
Learning rate α: Fig. 5 illustrates the convergence of average Q values at varying learning rates α within two Q-learning mechanisms, along with comparisons of objective values and CPU times. It is observed that the increase in α, which prioritizes past experiences, notably speeds up initial learning and convergence of the average Q values, but at the cost of reduced stability and pronounced Q-value oscillations. Fig. 5(b) shows the cases of premature convergence due to excessive initial rewards, which failed to enhance the objective values and instead led to entrapment in local optima, as shown in Fig. 5(c). With $\alpha = 0.3$, the average objective achieves optimality, surpassing the solutions at other learning rates by 1.75% on average. Moreover, it achieves a 39.01% reduction in CPU time compared to the average, despite not being the lowest.

Discount factor γ: The results shown in Fig. 6 indicate performance variations of QALNS with varying discount factor γ. A lower discount factor implies that the algorithm prioritizes immediate rewards over long-term gains, leading to earlier convergence. In contrast, a higher discount factor facilitates the exploration of future possibilities, also resulting in greater oscillations. With $\gamma = 0.1$, the mean Q values stabilize around 100 generations, while at 0.9, it requires over 800 generations. Regarding the quality of the solution, a higher discount factor yields better objective function values. For every 0.2 increase in γ, the objective value improves by an average of 0.53%. At $\gamma = 0.9$, there is an average enhancement of 1.37% compared to smaller values. Also, there is a clear trend in the CPU time needed to reach a best-found solution. For every 0.2 increase in γ, the CPU time decreases by 12.52%

Table 6
Comparison between Gurobi, ALNS and QALNS on newly generated instances.

Instances	Gurobi			ALNS					QALNS					Imp (%)	
	Obj	Time (s)	UB	Obj	Time (s)	#Iter	#UAV	DU (h)	Obj	Time (s)	#Iter	#UAV	DU (h)	Δ_G^{Obj}	Δ_A^{Obj}
1 center, 10 groups, 50 tasks															
CH_1_10_1	419	0.37	419	415	9.10	283	5	2.71	419	0.72	22	5	2.41	0.00	0.96
CH_1_10_2	628	0.88	628	621	3.11	91	10	2.43	628	1.83	64	9	2.41	0.00	1.13
CH_1_10_3	656	0.78	659	644	7.68	178	12	2.41	656	3.05	82	12	2.45	0.00	1.86
CH_1_10_4	723	0.89	727	723	5.17	77	11	2.56	723	1.56	36	13	2.67	0.00	0.00
CH_1_10_5	663	0.53	663	663	2.76	43	11	2.62	663	0.23	2	12	2.50	0.00	0.00
CH_1_10_6	665	1.60	665	665	0.78	10	13	2.71	665	0.15	1	13	2.75	0.00	0.00
CH_1_10_7	766	1.10	766	766	1.17	14	13	2.35	766	0.59	9	12	2.65	0.00	0.00
CH_1_10_8	688	3.65	688	688	3.37	39	12	2.13	688	0.26	3	12	2.60	0.00	0.00
CH_1_10_9	672	1.87	672	672	0.14	1	13	2.40	672	0.12	1	14	2.37	0.00	0.00
CH_1_10_10	543	7.68	543	543	0.49	4	9	2.46	543	0.24	2	10	2.54	0.00	0.00
Average	642	1.94	643	640	3.38	74	11	2.48	642	0.88	22	11	2.53	0.00	0.40
2 centers, 20 groups, 100 tasks															
CH_2_20_1	683	6.21	697	645	15.83	257	10	2.65	683	6.37	155	10	2.56	0.00	5.89
CH_2_20_2	1,210	18.79	1,215	1,118	28.43	287	17	2.46	1,210	11.02	151	19	2.66	0.00	8.23
CH_2_20_3	1,359	613.63	1,368	1,285	29.20	245	20	2.73	1,359	5.63	59	23	2.60	0.00	5.76
CH_2_20_4	1,484	1,219.58	1,520	1,451	26.45	180	24	2.74	1,494	6.74	59	25	2.46	0.67	2.96
CH_2_20_5	1,297	691.00	1,300	1,296	17.43	95	20	2.66	1,297	2.94	19	21	2.67	0.00	0.08
CH_2_20_6	1,325	3,600	1,378	1,321	17.49	86	27	2.76	1,330	5.59	34	27	2.55	0.38	0.68
CH_2_20_7	1,302	3,600	1,358	1,330	9.61	32	25	2.65	1,330	0.61	2	26	2.63	2.15	0.00
CH_2_20_8	1,342	3,600	1,414	1,362	4.71	16	25	2.71	1,363	1.94	7	24	2.74	1.56	0.07
CH_2_20_9	1,416	3,600	1,426	1,422	8.40	28	26	2.77	1,422	4.41	19	28	2.68	0.42	0.00
CH_2_20_10	1,313	3,600	1,362	1,314	9.46	30	29	2.71	1,314	5.38	17	29	2.52	0.08	0.00
Average	1,273	2,054.92	1,304	1,254	16.70	126	22	2.68	1,280	5.06	52	23	2.61	0.53	2.37
5 centers, 50 groups, 250 tasks															
CH_5_50_1	1,506	3,600	1,584	1,373	99.07	360	20	2.70	1,559	65.61	250	22	2.76	3.52	13.55
CH_5_50_2	2,104	3,600	2,371	2,061	98.11	93	40	2.74	2,307	65.63	61	41	2.76	9.65	11.94
CH_5_50_3	2,903	3,600	3,225	2,867	147.14	77	48	2.76	3,176	118.73	116	56	2.76	9.40	10.78
CH_5_50_4	2,894	3,600	3,366	3,027	128.95	104	56	2.77	3,183	38.01	37	55	2.77	9.99	5.15
CH_5_50_5	3,367	3,600	3,883	3,570	188.12	83	60	2.74	3,665	99.47	82	62	2.62	8.85	2.66
CH_5_50_6	3,243	3,600	3,586	3,306	239.85	138	64	2.72	3,477	124.94	77	66	2.77	7.22	5.17
CH_5_50_7	3,231	3,600	3,583	3,251	130.66	56	67	2.72	3,472	115.59	38	67	2.74	7.46	6.80
CH_5_50_8	2,850	3,600	3,241	2,948	285.19	71	59	2.77	3,182	146.27	41	73	2.72	11.65	7.94
CH_5_50_9	2,854	3,600	3,189	3,285	225.91	78	70	2.73	3,346	174.02	70	77	2.73	17.24	1.86
CH_5_50_10	2,665	3,600	3,060	2,937	276.50	55	75	2.74	3,000	80.86	28	69	2.66	12.57	2.15
Average	2,762	3,600	3,109	2,863	181.95	112	56	2.74	3,037	102.91	80	59	2.73	9.75	6.80
10 centers, 100 groups, 500 tasks															
CH_10_100_1	1,637	3,600	2,318	1,972	494.81	208	33	2.66	2,268	418.08	222	40	2.77	38.55	15.01
CH_10_100_2	2,988	3,600	4,465	3,449	611.65	155	59	2.70	4,022	457.24	102	58	2.75	34.61	16.61
CH_10_100_3	-[a]	–	–	4,142	471.46	83	69	2.76	4,729	407.09	88	67	2.77	–	14.17
CH_10_100_4	–	–	–	4,770	335.86	48	62	2.70	5,228	240.50	37	74	2.73	–	9.60
CH_10_100_5	–	–	–	5,233	365.42	53	70	2.76	5,692	300.52	39	70	2.77	–	8.77
CH_10_100_6	–	–	–	5,123	366.52	38	70	2.78	5,615	254.46	24	72	2.76	–	9.60
CH_10_100_7	–	–	–	5,291	629.68	61	87	2.76	5,669	480.98	48	79	2.74	–	7.14
CH_10_100_8	–	–	–	5,179	435.73	27	84	2.77	5,416	324.04	29	91	2.77	–	4.58
CH_10_100_9	–	–	–	5,250	379.37	26	87	2.77	5,349	294.48	23	77	2.76	–	1.89
CH_10_100_10	–	–	–	5,561	407.93	19	101	2.78	5,693	234.56	15	89	2.76	–	2.37
Average	2,313	3,600	3,392	4,597	449.84	72	72	2.74	4,968	341.19	63	72	2.76	36.58	8.98

[a] No feasible solution is found before memory runs out.

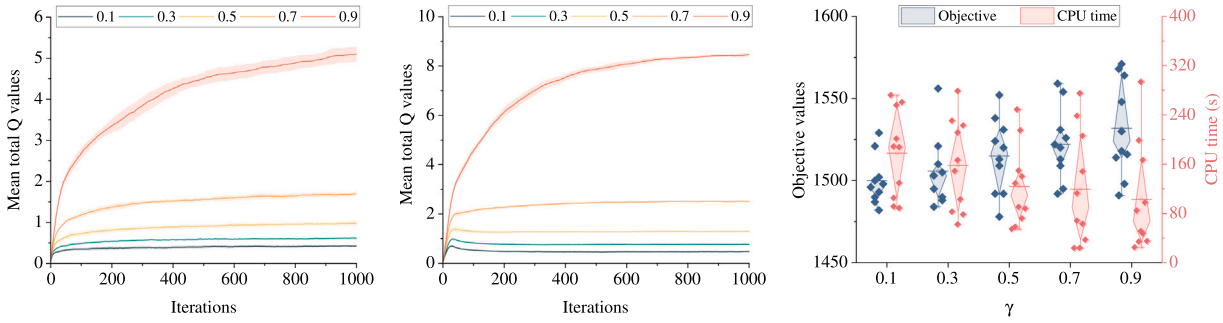


(a) Mean and standard deviation of Q values in QL1

(b) Mean and standard deviation of Q values in QL2

(c) Mean objective and CPU time (s)

Fig. 5. Performance of QALNS with varying learning rate ($\gamma = 0.9$).



(a) Mean and standard deviation of Q values in QL1 (b) Mean and standard deviation of Q values in QL2 (c) Mean objective and CPU time (s)

Fig. 6. Performance of QALNS with varying discount factor ($\alpha = 0.3$).

Table 7
Solutions for CH_2_20_K and CH_5_50_K instances with varying number of UAVs.

#UAV per type per center	#UAV in use						Obj	Unmet tasks	Remaining inventory
	Total	u_1	u_2	u_3	u_4	u_5			
CH_2_20_K instances									
1	9	2	2	2	2	1	635	40	273
2	19	4	4	3	4	4	1,109	21	134
3	23	6	5	2	6	4	1,317	11	46
4	24	8	5	1	8	2	1,382	9	23
5	22	9	4	2	9	1	1,387	10	22
6	27	9	4	3	9	2	1,383	9	22
7	27	9	6	1	9	2	1,397	9	20
8	26	10	4	1	10	1	1,405	10	21
CH_5_50_K instances									
1	22	5	5	3	5	4	1,486	100	809
2	39	10	9	4	10	6	2,610	53	486
3	53	15	9	6	15	8	3,265	27	283
4	62	20	12	5	17	8	3,634	21	160
5	66	24	11	5	20	6	3,660	19	146
6	68	30	8	3	19	8	3,698	19	124
7	75	30	8	5	23	9	3,755	16	107
8	76	33	8	3	22	10	3,784	15	104

and is optimal at $\gamma = 0.9$ with an improvement of 28.89% over the average of other values.

We observe that changes in the learning rate α and discount indicator γ indeed affect the performance of QALNS. Specifically, the impact of α on performance is non-monotonic. In our experiments, setting $\alpha = 0.3$ yielded superior average objective values and CPU time. At this setting, the standard deviation of the mean Q values is minimized, facilitating stable solutions across different runs. An increase in γ notably enhances computational effectiveness, improving both solution quality and efficiency. Overall, CPU time is more sensitive to both parameters than the objective value. Moreover, changes in the α values have a more pronounced effect on algorithm performance than changes in γ.

5.2.3. Sensitivity analyses for the parameters of the HUTA-PDP

This section presents sensitivity analyses on the results for the available number of UAVs and the distribution of healthcare centers with CH_2_20_K and CH_5_50_K instances.

Number of UAVs: In Section 5.2, we match the available number of UAVs for randomly generated instances of different sets. Here, we keep all other data fixed and only change the number of UAVs per type per center, observing the preferences of heterogeneous UAVs when performing tasks in Table 7.

From Table 7, it is evident that with increasing initial size of the UAV fleet, the number of tasks fulfilled increases significantly, regardless of whether it is for the CH_2_20_K and CH_5_50_K instances. When each center is equipped with no more than five UAVs of each type, for each additional UAV added, the total revenue increases by 28.19% and 24.68% for the CH_2_20_K and CH_5_50_K instances respectively, while the unmet task quantity decreases by 31.95% and 25.55% respectively. A stability in the objective values is observed once the number of UAVs results in a saturation of the tasks. We notice that, due to the performance differences among heterogeneous UAVs, there is a significant contrast in the quantity of UAVs selected to execute tasks, especially in the scenarios with sufficient UAVs. Skycart (u_1), Wingcopter (u_2) and Freely (u_4) perform better due to their better flight distance and payload capacity. Although DJI (u_5) can only fly up to 16 km, its exceptional payload capacity makes it superior to that of other types. Meanwhile, moderately performing UAVs are prone to being left idle when there is an abundance of alternative types available.

Number of centers: We test the sensitivity of the results for CH_10_100_K instances with varying numbers of centers, maintaining constant total demand and inventory while incrementally increasing centers, as shown in Table 8.

Initially, increasing the number of centers significantly increase the objective value, peaking at 4,297 profit with four centers and the fewest unmet tasks. Subsequently, the objective value slightly decreases, and there is a slight increase in the number of drones used and unmet tasks. Our data show that contrary to expectations, having more centers increases the profit, particularly with ample stocks and idle drones. This can be explained by the division of tasks among multiple centers. Given the limited fly range and payload of UAVs when the remaining

Table 8
Solutions for CH_10_100 instances with varying number of centers.

#Centers	Obj	#UAV in use	Unmet tasks	Remaining inventory
1	3,030	62	130	819
2	3,813	72	91	642
3	3,992	61	96	657
4	4,297	70	88	623
5	4,115	71	98	605
6	3,895	70	97	711
7	4,005	76	101	664
8	4,069	74	103	716
9	3,910	69	101	715
10	4,083	73	107	662

inventory of a single center is not sufficient to perform a task, the system automatically abandons it rather than collect resources from multiple centers. This setup results in a larger number of centers, sometimes reducing the fulfillment rate of orders.

6. Conclusions

This research have presented a joint optimization problem of task assignment and PD routing operations with heterogeneous UAVs in emergency healthcare scenarios. This problem is especially important for the delivery of medications, vaccines, and other medical items, and for the collection of biological samples. The objective is to maximize the total profit generated by performing tasks with different priorities. In the optimization process, the capacities of heterogeneous UAVs, including flying range, speed, height, and payload, as well as inventory constraints, are considered.

We first presented the MILP model for the HUTA-PDP, and then solved a small-scale instances with Gurobi. Next, we introduced the QALNS algorithm, a hybrid metaheuristic that incorporates the Q-learning method. This integration aims to enhance the selection of the destroy–repair operators and improve their efficiency. To fully evaluate the effectiveness of the proposed approaches, we designed a series of computational experiments on the well-known PDPTW benchmark instances and newly generated HUTA-PDP instances. The results showed that the HUTA-PDP can effectively distribute limited healthcare supplies in emergency scenarios by optimizing the routing of heterogeneous UAVs. Furthermore, the QALNS algorithm exhibited a high performance in solving different sets of both the benchmark instances and the generated HUTA-PDP instances. The performance gap between QALNS and other method, including Gurobi and the original ALNS, is shown to increase for larger-scale HUTA-PDP instances, making QALNS a suitable option for larger and practical problem instances. Sensitivity experiments demonstrate the impact of key parameters in the metaheuristic, including learning rate and discount indicator, on both the efficiency and effectiveness of the algorithm.

Additionally, management insights were derived from sensitivity analyses: (*i*) considering task requirements like terrain, payload, flying range, and number of drops when purchasing heterogeneous, UAVs can enhance utilization rates and prevent idle losses; (*ii*) matching the quantity of UAVs to the number of tasks and inventory suffices, particularly in emergency situations, since an excess of UAVs does not substantially improve the overall profit; and (*iii*) an excessive number of centers, given a fixed total inventory, may lead to resource over-dispersion, whereas an appropriate number of centers can enhance task completion rates, particularly with UAVs characterized by limited flying ranges and capacities.

Potential avenues for future research could involve addressing uncertainties within the UAV operations network. For instance, in emergency scenarios marked by challenges like communication disruptions and path obstacles, it is crucial to develop dynamic and robust UAV logistics solutions that adapt to limited information. Another prospective research area can be to explore diverse objective functions, including the minimization of the overall cost, operations time, and the number of UAVs in use.

CRediT authorship contribution statement

Ziru Lin: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Xiaofeng Xu:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Emrah Demir:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Gilbert Laporte:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization.

Acknowledgments

The work was partly supported by the China Scholarship Council program (No. 202306450084) and Innovation Fund Project for Graduate Students of China University of Petroleum (East China) (No. 23CX04024A). Thanks are due to the editors and the reviewers for their valuable comments.

Data availability

Data will be made available on request.

References

Aksen, D., Kaya, O., Salman, F.S., Tüncel, Ö., 2014. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. European J. Oper. Res. 239 (2), 413–426.
Angelelli, E., Archetti, C., Filippi, C., Vindigni, M., 2021. A dynamic and probabilistic orienteering problem. Comput. Oper. Res. 136, 105454.
Bartolini, N., Coletta, A., Maselli, G., Khalifeh, A., 2020. A multi-trip task assignment for early target inspection in squads of aerial drones. IEEE Trans. Mob. Comput. 20 (11), 3099–3116.
Boualamia, H., Metrane, A., Hafidi, I., Mellouli, O., 2023. A new adaptation mechanism of the ALNS algorithm using reinforcement learning. In: Aboutabit, N., Lazaar, M., Hafidi, I. (Eds.), Advances in Machine Intelligence and Computer Science Applications. Springer, Cham, pp. 3–14.
Chao, I.-M., Golden, B.L., Wasil, E.A., 1996. The team orienteering problem. European J. Oper. Res. 88 (3), 464–474.
Chen, X., Ulmer, M.W., Thomas, B.W., 2022. Deep Q-learning for same-day delivery with vehicles and drones. European J. Oper. Res. 298 (3), 939–952.
Chen, P., Wang, Q., 2024. Learning for multiple purposes: A Q-learning enhanced hybrid metaheuristic for parallel drone scheduling Traveling Salesman Problem. Comput. Ind. Eng. 187, 109851.
Christiaens, J., Vanden Berghe, G., 2020. Slack induction by string removals for vehicle routing problems. Transp. Sci. 54 (2), 417–433.
Cinar, A., Salman, F.S., Bozkaya, B., 2021. Prioritized single nurse routing and scheduling for home healthcare services. European J. Oper. Res. 289 (3), 867–878.
Dasdemir, E., Batta, R., Köksalan, M., Tezcaner Öztürk, D., 2022. UAV routing for reconnaissance mission: A multi-objective orienteering problem with time-dependent prizes and multiple connections. Comput. Oper. Res. 145, 105882.
Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. European J. Oper. Res. 223 (2), 346–359.
Demir, E., Syntetos, A., Van Woensel, T., 2022. Last mile logistics: Research trends and needs. IMA J. Manag. Math. 33 (4), 549–561.
Do C. Martins, L., Hirsch, P., Juan, A.A., 2021. Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. Int. Trans. Oper. Res. 28 (1), 201–221.
Dukkanci, O., Koberstein, A., Kara, B.Y., 2023. Drones for relief logistics under uncertainty after an earthquake. European J. Oper. Res. 310 (1), 117–132.
Elluru, S., Gupta, H., Kaur, H., Singh, S.P., 2019. Proactive and reactive models for disaster resilient supply chain. Ann. Oper. Res. 283 (1), 199–224.
Enayati, S., Li, H., Campbell, J.F., Pan, D., 2023. Multimodal vaccine distribution network design with drones. Transp. Sci. 57 (4), 1069–1095.
Fragkos, M.E., Zeimpekis, V., Koutras, V., Minis, I., 2022. Supply planning for shelters and emergency management crews. Oper. Res. 22 (1), 741–777.
Gao, W., Luo, J., Zhang, W., Yuan, W., Liao, Z., 2020. Commanding cooperative UGV-UAV with nested vehicle routing for emergency resource delivery. IEEE Access 8, 2169–3536.

Gao, J., Zhen, L., Laporte, G., He, X., 2023. Scheduling trucks and drones for cooperative deliveries. Transp. Res. Part E: Logist. Transp. Rev. 178, 103267.
Ghelichi, Z., Gentili, M., Mirchandani, P.B., 2021. Logistics for a fleet of drones for medical item delivery: A case study for Louisville, KY. Comput. Oper. Res. 135, 105443.
Gonzalez-R, P.L., Canca, D., Andrade-Pineda, J.L., Calle, M., Leon-Blanco, J.M., 2020. Truck-drone team logistics: A heuristic approach to multi-drop route planning. Transp. Res. C 114, 657–680.
Gu, W., Cattaruzza, D., Ogier, M., Semet, F., 2019. Adaptive large neighborhood search for the commodity constrained split delivery VRP. Comput. Oper. Res. 112, 104761.
Hammami, F., Rekik, M., Coelho, L.C., 2020. A hybrid adaptive large neighborhood search heuristic for the team orienteering problem. Comput. Oper. Res. 123, 105034.
Hanafi, S., Mansini, R., Zanotti, R., 2020. The multi-visit team orienteering problem with precedence constraints. European J. Oper. Res. 282 (2), 515–529.
Howe, S., 2022. Wingcopter teams with UAV LATAM for drone delivery operations in Peru. Available from: https://www.commercialuavnews.com/drone-delivery/wingcopter-teams-with-uav-latam-for-drone-delivery-operations-in-peru. (Accessed 25 Oct 2024).
Huang, K., Jiang, Y., Yuan, Y., Zhao, L., 2015. Modeling multiple humanitarian objectives in emergency response to large-scale disasters. Transp. Res. Part E: Logist. Transp. Rev. 75, 1–17.
Jeong, H.Y., Song, B.D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. Int. J. Prod. Econ. 214, 220–233.
Ji, J., Guo, Y., Gao, X., Gong, D., Wang, Y., 2021. Q-learning-based hyperheuristic evolutionary algorithm for dynamic task allocation of crowdsensing. IEEE Trans. Cybern. 53 (4), 2211–2224.
Kallestad, J., Hasibi, R., Hemmati, A., Sörensen, K., 2023. A general deep reinforcement learning hyperheuristic framework for solving combinatorial optimization problems. European J. Oper. Res. 309 (1), 446–468.
Kang, M., Lee, C., 2021. An exact algorithm for heterogeneous drone-truck routing problem. Transp. Sci. 55 (5), 1088–1112.
Li, H., Lim, A., 2008. Li & Lim's PDPTW benchmark. Available from: https://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark/. (Accessed 25 Oct 2024).
Li, Y., Zhang, J., Yu, G., 2020. A scenario-based hybrid robust and stochastic approach for joint planning of relief logistics and casualty distribution considering secondary disasters. Transp. Res. Part E: Logist. Transp. Rev. 141, 102029.
Liu, H., Ge, J., Wang, Y., Li, J., Ding, K., Zhang, Z., Guo, Z., Li, W., Lan, J., 2022. Multi-UAV optimal mission assignment and path planning for disaster rescue using adaptive genetic algorithm and improved artificial bee colony method. Actuators 11 (1), 4.
Lu, Y., Yang, C., Yang, J., 2022. A multi-objective humanitarian pickup and delivery vehicle routing problem with drones. Ann. Oper. Res. 319 (1), 291–353.
Martin, L., Minner, S., Poças, D., Schulz, A.S., 2021. The competitive pickup and delivery orienteering problem for balancing car-sharing systems. Transp. Sci. 55 (6), 1232–1259.
Moosavi Heris, F.S., Ghannadpour, S.F., Bagheri, M., Zandieh, F., 2022. A new accessibility based team orienteering approach for urban tourism routes optimization (a real life case). Comput. Oper. Res. 138, 105620.
Osman, I.H., Laporte, G., 1996. Metaheuristics: A bibliography. Ann. Oper. Res. 63, 511–623.
Pan, B., Zhang, Z., Lim, A., 2021. A hybrid algorithm for time-dependent vehicle routing problem with time windows. Comput. Oper. Res. 128, 105193.
Pasha, J., Elmi, Z., Purkayastha, S., Fathollahi-Fard, A.M., Ge, Y., Lau, Y., Dulebenets, M.A., 2022. The drone scheduling problem: A systematic state-of-the-art review. IEEE Trans. Intell. Transp. Syst. 23 (9), 14224–14247.
Qin, W., Zhuang, Z., Huang, Z., Huang, H., 2021. A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. Comput. Ind. Eng. 156, 107252.
Ropke, S., Cordeau, J.-F., 2009. Branch and cut and price for the pickup and delivery problem with time windows. Transp. Sci. 43 (3), 267–286.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.
Saeedvand, S., Aghdasi, H.S., Baltes, J., 2020. Novel hybrid algorithm for team orienteering problem with time windows for rescue applications. Appl. Soft Comput. 96, 106700.
Sartori, C.S., Buriol, L.S., 2020. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. Comput. Oper. Res. 124, 105065.
Saxena, V., Jaldén, J., Klessig, H., 2019. Optimal UAV base station trajectories using flow-level models for reinforcement learning. IEEE Trans. Cogn. Commun. Netw. 5 (4), 1101–1112.
Seraji, H., Tavakkoli-Moghaddam, R., Asian, S., Kaur, H., 2022. An integrative location-allocation model for humanitarian logistics with distributive injustice and dissatisfaction under uncertainty. Ann. Oper. Res. 319 (1), 211–257.
She, R., Ouyang, Y., 2021. Efficiency of UAV-based last-mile delivery under congestion in low-altitude air. Transp. Res. C 122, 102878.
Stavropoulou, F., Repoussis, P.P., Tarantilis, C.D., 2019. The vehicle routing problem with profits and consistency constraints. European J. Oper. Res. 274 (1), 340–356.
Voigt, S., 2024. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. European J. Oper. Res. Advance online publication.
Wang, J., Guo, J., Chen, J., Tian, S., Gu, T., 2019. Uncertain team orienteering problem with time windows based on uncertainty theory. IEEE Access 7, 63403–63414.
Wang, X., Liu, Z., Li, X., 2023. Optimal delivery route planning for a fleet of heterogeneous drones: A rescheduling-based genetic algorithm approach. Comput. Ind. Eng. 179, 109179.
Wen, X., Wu, G., 2022. Heterogeneous multi-drone routing problem for parcel delivery. Transp. Res. C 141, 103763.
Xiong, T., Liu, F., Liu, H., Ge, J., Li, H., Ding, K., Li, Q., 2023. Multi-drone optimal mission assignment and 3D path planning for disaster rescue. Drones 7 (6), 394.
Xu, X., Lin, Z., Li, X., Shang, C., Shen, Q., 2022. Multi-objective robust optimisation model for MDVRPLS in refined oil distribution. Int. J. Prod. Res. 60 (22), 6772–6792.
Xu, X., Wei, Z., 2023. Dynamic pickup and delivery problem with transshipments and LIFO constraints. Comput. Ind. Eng. 175, 108835.
Yadav, N., Tanksale, A., 2022. An integrated routing and scheduling problem for home healthcare delivery with limited person-to-person contact. European J. Oper. Res. 303 (3), 1100–1125.
Yu, V.F., Jewpanya, P., Lin, S.-W., Redi, A.A.N.P., 2019. Team orienteering problem with time windows and time-dependent scores. Comput. Ind. Eng. 127, 213–224.
Zhang, Z., Wu, Z., Zhang, H., Wang, J., 2023. Meta-learning-based deep reinforcement learning for multiobjective optimization problems. IEEE Trans. Neural Netw. Learn. Syst. 34 (10), 7978–7991.
Zhen, L., Yang, Z., Laporte, G., Yi, W., Fan, T., 2024. Unmanned aerial vehicle inspection routing and scheduling for engineering management. Engineering.
Zhou, Z., Feng, J., Gu, B., Ai, B., Mumtaz, S., Rodriguez, J., Guizani, M., 2018. When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning. IEEE Trans. Commun. 66 (11), 5526–5538.
Zhu, P., Fang, X., 2021. Multi-UAV cooperative task assignment based on half random Q-learning. Symmetry 13 (12), 2417.
Zipline, 2024. Zipline fact sheet. Available from: https://www.flyzipline.com/about/zipline-fact-sheet. (Accessed 25 Oct 2024).