Transportmetrica A: Transport Science

ISSN: 2324-9935 (Print) 2324-9943 (Online) Journal homepage: www.tandfonline.com/journals/ttra21

Heterogeneous vehicle scheduling with precedence constraints

Ruiyou Zhang, Zhujun Liu & Ilkyeong Moon

To cite this article: Ruiyou Zhang, Zhujun Liu & Ilkyeong Moon (2026) Heterogeneous vehicle scheduling with precedence constraints, Transportmetrica A: Transport Science, 22:1, 1, DOI: 10.1080/23249935.2024.2338249

To link to this article: https://doi.org/10.1080/23249935.2024.2338249

Published online: 08 Apr 2024.

TRANSPORTMETRICA A: TRANSPORT SCIENCE
2026, VOL. 22, NO. 1, 2338249
https://doi.org/10.1080/23249935.2024.2338249

Heterogeneous vehicle scheduling with precedence constraints

Ruiyou Zhang[a], Zhujun Liu[a,b] and Ilkyeong Moon[c]

[a]College of Information Science and Engineering, Northeastern University, Shenyang, People's Republic of China; [b]Technology Testing Center of Shengli Oilfield Branch, China Petrochemical Co., Ltd., Dongying, People's Republic of China; [c]Department of Industrial Engineering and Institute of Engineering Research, Seoul National University, Seoul, Republic of Korea

ABSTRACT

The problem of heterogeneous vehicle scheduling with precedence constraints is inspired by the transportation service in tourism areas. Tourists must take the shuttle vehicles provided by the areas because of the long distances between the scenic spots. The scheduling of vehicles in tourism areas is complicated because the transportation requests of tourists are precedence-constrained temporally and spatially. The problem optimises both the cost of using vehicles and the waiting time of tourists. A mixed-integer linear programming model is formulated according to the description of a graph. An adaptive large neighbourhood search algorithm with several specialised operators is designed to solve the problem. Experiments based on randomly generated instances validate the mathematical model and the algorithm. A real-size instance based on Qiandao Lake in China is also analysed. The results indicate that the algorithm outperforms the model. The sensitivities of key parameters are analysed with managerial insights presented.

ARTICLE HISTORY
Received 29 April 2023
Accepted 27 March 2024

KEYWORDS
Transportation; heterogeneous vehicle scheduling; precedence constraint; adaptive large neighbourhood search; tourism

1. Introduction

Efficient operation management in the tourism industry is a recent hot topic in both academic and industrial fields. The management and operation of a tourism area influence both the revenue of the area and the satisfaction degree of customers directly. Many tourism areas provide shuttle transportation services because of the long distances among the scenic spots. For instance, shuttle buses are provided in both the Kanas scenic area located in Xinjiang, China, and the Grand Canyon located in Arizona, USA. In the Aegean Sea located on the Greek peninsula and the Qiandao Lake scenic area located in Zhejiang, China, visitors take steamers or boats to travel among the islands in areas. Such areas are responsible for the management of shuttle transports. They usually prohibit the use of private transport, given concerns over public order and driving safety.

Vehicle (or some other transportation modes) scheduling is significant for improving the operational efficiency of tourism areas. However, the shuttle vehicles in many areas

CONTACT Ilkyeong Moon ikmoon@snu.ac.kr Department of Industrial Engineering and Institute of Engineering Research, Seoul National University, Seoul 08826, Republic of Korea

are usually operated traditionally with some weaknesses (Dou, Meng, and Liu 2021; Guo et al. 2024). For example, a typical mode, in which one group of people charters one vehicle (Duan et al. 2020), is common in many areas. The fares for chartering a vehicle are usually high and the idle time of vehicles is usually long in this mode. One main reason for this is that a vehicle in the typical mode has to stop at a scenic spot to wait for visitors. Furthermore, many tourism areas provide the same transportation services using homogeneous vehicles, but the transportation demands are usually diverse.

Flexible operation modes of vehicles are necessary and significant in tourism areas. One of the reasons for this is that the complexity of vehicle scheduling has increased with both the growth of the number of visitors and the increase of personalised requirements. The flexible operation mode of vehicles proposed in the research of Zhang, Liu, and Feng (2021) is typical in tourism areas. A vehicle in such a flexible mode need not wait for visitors at any location. It can be scheduled to pick up or deliver some other tasks after it has finished a delivery task. Furthermore, the transportation requirements of a visitor to different locations could be satisfied by different vehicles in the mode. As a result, the flexible mode is the potential to improve the utilisation rate of vehicles, especially when compared to the common modes (Chen et al. 2017; Guo et al. 2019).

The problem of heterogeneous vehicle scheduling with precedence constraints in tourism areas considers the flexible operation mode of vehicles. Specifically, the transportation requests in the problem are precedence-constrained temporally and spatially. An example (Example 1) is given to explain the problem, in which a group of tourists arrives at a scenic area and leaves the area after visiting two scenic spots in sequence. A group of tourists consists of several persons who travel together in a scenic area. As shown in Figure 1, for any two neighbouring transportation requests (Requests 1 and 2, or Requests 2 and 3) of Example 1, the destination of the former request is the same as the origin of the latter request. These are the spatial precedence constraints (Melis and Sörensen 2021; Mo et al. 2020). Moreover, the time of a pickup task at a scenic spot must be later than the time of the delivery task at the spot, and the time gap is at least the visit duration. All the starting times of transportation requests are interrelated and precedence-constrained in the problem. However, in general, vehicle scheduling problems, the starting times of transportation requests are usually independent of each other and are only constrained by given time windows or planning horizons.

Furthermore, the problem of heterogeneous vehicle scheduling with precedence constraints is an extension of the research of Liu, Moon, and Zhang (2024). The main extensions are as follows: (1) Heterogeneous vehicles with different fixed costs, driving costs, and loading capacities are used to serve tourists. A vehicle can pick up or deliver several groups on every trip within its loading capacity. Therefore, the actual ride time of a transportation request may be longer than the direct driving time between the departure location and the destination of this request. (2) The longest waiting time of tourists in a transportation request is optimised as a component of the objective function, which is related to the satisfaction degree of tourists. (3) The decision of the starting time of a pickup service is precedence-constrained not only by the time related to the tourist but also by the time related to the vehicle. Moreover, the starting times of pickup services would influence tourists' waiting time in the objective function.

Such characteristics and extensions bring great challenges to formulate and to solve the problem. First, the vehicle routes, the vehicle type of each route, and the starting time

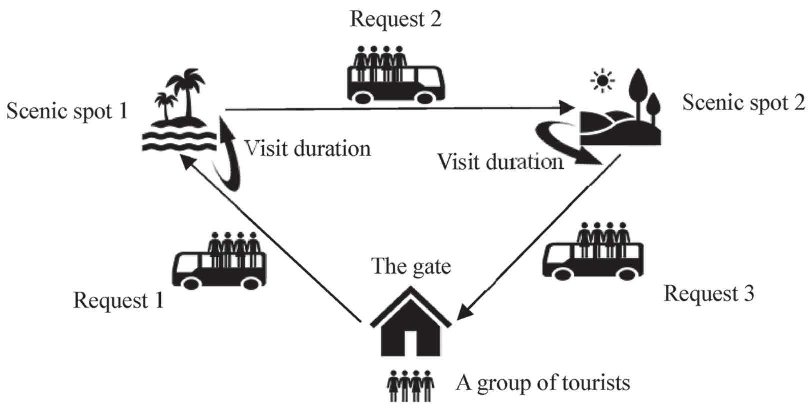

Figure 1. Example 1 illustrating the problem.

of each pickup or delivery service should be determined simultaneously. In particular, the starting times of services are hard to decide because they are precedence-constrained and are related to the longest waiting time of tourists in the optimisation objective. Second, matching different transportation requests to each trip of vehicles is limited by the loading capacities of different types of vehicles. The matching would affect both the utilisation rate of vehicles and the waiting time of tourists. More importantly, the vehicle routes in a solution are interdependent of each other because the precedence-constrained transportation requests may be handled by different vehicles.

The main contributions made in this study can be summarised as follows. A problem of heterogeneous vehicle scheduling with precedence constraints (HVS-P) inside tourism areas is formally presented and efficiently solved. The problem considers a flexible operation mode of vehicles and minimises both the using cost of vehicles and the cost of the longest waiting time of tourists. Furthermore, a graph-based method is presented to formulate the problem based on the description of task nodes and feasible arcs. Precedence constraints are clearly described based on the graph-based method. An Adaptive Large Neighborhood Search (ALNS) algorithm is also designed to handle the problem efficiently, in which some simplified mathematical models are employed to decode the staring times of services and to check the feasibility of solutions. Importantly, the model and the algorithm are validated using a number of numerical instances.

The following sections of this paper are organised as follows. Section 2 summarises the literature regarding the heterogeneous vehicle scheduling problems, vehicle scheduling with precedence constraints, and the ALNS algorithm. Section 3 describes the HVS-P problem in tourism areas. Section 4 presents the graphical description and the mixed-integer linear programming model of the problem. Section 5 proposes an ALNS algorithm to solve the problem. Several experiments are implemented in Section 6. Finally, the conclusions are given in Section 7.

2. Literature review

This section reviews the related literature. Specifically, Section 2.1 summarises the heterogeneous vehicle scheduling problems. Section 2.2 reviews the vehicle scheduling with precedence constraints in various fields. Finally, the ALNS algorithm for vehicle scheduling problems is briefly reviewed in Section 2.3.

2.1. Heterogeneous vehicle scheduling problems

The heterogeneous vehicle scheduling (HVS) problem, as a main variant of the vehicle routeing problem, has been around for more than thirty years (Koç et al. 2016). In the HVS problem, heterogeneous fleets with different loading capacities, fixed costs of using a vehicle and the travelling costs of a unit distance, are considered to provide transportation services for customers (Ren, Jin, and Wu 2023). The operational cost of vehicles in many industries can be 50% to 60% of the total cost. Some researchers pointed out that the use of heterogeneous fleets plays a crucial role in saving transportation costs and improving the flexibility of transportation services (Chaowasakoo et al. 2017).

The research of the HVS problem and its variants is still important and popular nowadays because of its application in many fields (Messaoud 2022; Moon, Salhi, and Feng 2020; Ren, Jin, and Wu 2023). For example, a green HVS problem was researched to optimise carbon emissions in the transportation industry, in which the experiments indicated that the use of heterogeneous vehicles can reduce carbon emissions in most scenarios (Sun, Yu, and Wang 2019). A similar green logistics problem was also researched by Wang et al. (2020a). Furthermore, the use of heterogeneous transport is significant for the development of a logistics system. For instance, a collaborative two-echelon multicenter vehicle routeing optimisation setup was researched to increase operation efficiencies and to reduce the transportation emissions of service providers (Wang et al. 2020b). A truck drone hybrid routeing problem was studied by Wang, Moon, and Zhang (2022) to facilitate a sustainable urban logistics distribution system.

A public transit vehicle scheduling problem with multiple vehicle types (Lai et al. 2022; Shang et al. 2023) is similar to the problem in this research except for the operation mode of vehicles. Public vehicles are usually operated with fixed routes and schedules. Consequently, the research on public vehicle scheduling mainly focuses on creating efficient schedules or optimising timetables (Lin et al. 2023; Tian, Lin, and Wang 2023). In recent years, multiple types of electric vehicles have usually been considered in the public transit scheduling problem because there is the potential to improve the satisfaction of passengers and the income of bus companies (Yao et al. 2020; Zhang et al. 2022b). The problem of routeing scheduling for technicians is also a type of HVS problem. Heterogeneous technicians with different skills are usually scheduled to satisfy the requirements of customers (Cappanera, Requejo, and Scutellà 2020). For example, Hanafi, Mansini, and Zanotti (2020) solved a route scheduling problem of workers that was motivated by the assembly of kitchen equipment, in which each assembly requirement needed to be accomplished orderly by several workers with different skills.

Heuristic algorithms are usually designed to solve the HVS problem in literature. One of the main reasons behind this is that the HVS problem is more complex than general vehicle routeing problems (Tayachi and Jendoubi 2023) and it is usually applied in many industry fields with a limited calculation time (Rattanamanee and Nanthavanij 2022). For instance, a math-heuristic based on kernel search was designed to solve a flexible periodic vehicle routeing problem with heterogeneous fleets (Huerta-Muñoz et al. 2022). Furthermore, many heuristic algorithms based on the local search (Máximo, Cordeau,

Table 1. Summary of heterogeneous routeing problems.

Problems	Articles	Solution methods
Transportation and logistics	Ren, Jin, and Wu (2023)	Fast elitist non-dominated sorting genetic algorithm
	Chaowasakoo et al. (2017)	Heuristic truck dispatching methods
	Moon, Salhi, and Feng (2020)	Heuristic and a hybrid genetic algorithm
	Messaoud (2022)	Hybrid genetic algorithm
	Sun, Yu, and Wang (2019)	Exact algorithm
	Wang et al. (2020a)	Gaussian mixture clustering algorithm
	Wang et al. (2020b)	Tree-component solution framework
	Wang, Moon, and Zhang (2022)	Iterative local search heuristic
Public transit vehicle scheduling	Shang et al. (2023)	Capacity restraint incremental assignment
	Lai et al. (2022)	Heuristic and mathematical model
	Lin et al. (2023)	Bilevel model
	Tian, Lin, and Wang (2023)	Two-stepped heuristic
	Yao et al. (2020)	Heuristic procedure
	Zhang et al. (2022b)	Adaptive large neighbourhood search
	Xue et al. (2022)	Adaptive large neighbourhood search
Routeing scheduling of technicians	Cappanera, Requejo, and Scutellà (2020)	Mathematical model and lower bounding techniques
	Hanafi, Mansini, and Zanotti (2020)	Kernel search; Branch-and-cut
Vehicle routeing	Huerta-Muñoz et al. (2022)	Math-heuristic
	Máximo, Cordeau, and Nascimento (2022)	Adaptive large neighbourhood search
	Meliani et al. (2022)	Tabu search
Transportation of tourists	**This paper**	Adaptive large neighbourhood search

and Nascimento 2022), such as the tabu search (Meliani et al. 2022) and the adaptive large neighbourhood search (Xue et al. 2022), are typically designed for solving the HVS problems.

As shown in Table 1, the application of heterogeneous vehicle scheduling is significant. However, it has not been widely researched for the transportation of tourists in scenic areas. In fact, the usage of heterogeneous vehicles in tourism areas has the potential to save the operational cost of vehicles, satisfy the diversified transportation requirements of tourists, and improve the degree of tourists' satisfaction.

2.2. Vehicle scheduling with precedence constraints

Vehicle scheduling with precedence constraints (VS-P) exists in various fields, such as freight transportation and package delivery in the logistics industry (Baykasoglu et al. 2019; Esztergar-Kiss, Gentile, and Nuzzolo 2020; Huang et al. 2023). The precedence constraint in such problems is that some customers (or services) need to be served before others (Roohnavazfar and Pasandideh 2022). For example, Bai et al. (2020) investigated a package delivery problem minimising the total time of serving all customers, in which some customers need to be visited before others.

The precedence constraint in the vehicle scheduling problem for tourists in tourism areas is different from the constraints in literature because two types of precedence constraints exist simultaneously in the tourist transportation problem. One type of precedence is between the pickup and the delivery activities in a single transportation request; the other is between the transportation requests of the same tourist. Consequently, the precedence constraints among the requests of tourists are considered not only in time but also in space.

Table 2. Comparison between this study and the relevant previous studies.

Problems	Articles	Precedence constraints in time	Precedence constraints in space	Capacity constraints	Heterogeneous vehicles	Time constraints	Solution methods
Package delivery	Bai et al. (2020)	✓		✓	✓	PH*[1]	Heuristic algorithm
Home healthcare scheduling	Cinar et al. (in press)	✓			✓	TW*[2], PH	Exact algorithm
Assembly problem	Ali, Côté, and Coelho (2021)	✓			✓	TW, PH	ALNS*[4]
Container drayage	Zhang, Wang, and Wang (2021)	✓				TW, PH	Heuristic algorithm
	Wang, Moon, and Zhang (2022)	✓				TW, PH	ALNS
Vessel routeing	Lianes et al. (2021)	✓		✓	✓	TW, PH	ALNS
Transportation of tourists	Zhang, Liu, and Feng (2021)	✓				WT*[3], PH	Two-staged algorithm
	Liu, Moon, and Zhang (2024)	✓	✓	✓		PH	Math-heuristic
	This research	✓	✓	✓	✓	WT, PH	ALNS

PH*[1]: Planning Horizon; TW*[2]: Time Window; WT*[3]: Wait Time; ALNS*[4]: Adaptive Large Neighborhood Search

However, as shown in Table 2, only one type of precedence constraint is considered in literature usually. For example, in the home healthcare scheduling problem, patients enjoyed the care provided by multiple medical staff with synchronisation or precedence constraints at their homes (Cinar et al. in press), which is the precedence in time. The research on the precedence constraint that is similar to the home healthcare problem can also be found in the assembly problem (Ali, Côté, and Coelho 2021), the container drayage problem (Wang, Moon, and Zhang 2022; Zhang, Wang, and Wang 2021), as well as the vessel routeing problem arising in the aquaculture service industry (Lianes et al. 2021).

Both Zhang, Liu, and Feng (2021) and Liu, Moon, and Zhang (2024) studied VS-P problems for tourists in scenic areas. However, Zhang, Liu, and Feng (2021) only considered the precedence between the delivery and the pickup tasks of the same tourist at the same location. Furthermore, Zhang, Liu, and Feng (2021) assumed that the capacity of a vehicle was exactly one group of tourists.

Liu, Moon, and Zhang (2024) considered the same precedence constraints but with the following different components as the HVS-P problem. Firstly, Liu, Moon, and Zhang () studied the scheduling of homogeneous vehicles. Furthermore, the waiting time that involved the satisfaction degree of tourists' was not considered. Finally, the decision of the starting time of each pickup task was only constrained by the time of tourists. Liu, Moon, and Zhang () assumed that a tourist group would be picked up without any waiting time when the tourists finished visiting a scenic spot. However, the HVS-P problem in this research relaxes the assumption because the decision of starting times would influence the waiting time of tourists.

2.3. ALNS for vehicle scheduling problems

Very few articles focus on exact algorithms to handle some similar complicated problems, including the problem of VS-P (Dohn, Rasmussen, and Larsen 2011), although exact algorithms have been researched to solve many vehicle scheduling problems. For example, Rasmussen et al. (2012) handled the precedence constraints using the branching procedure of a branch-and-price algorithm according to the preprocessing of time windows. Similar handling of the precedence constraints in the exact algorithm can also be found in the research of Li et al. (2020).

Heuristic algorithms have attracted much attention for solving vehicle scheduling problems in academic fields because of their advantages in solving large-scale instances (Chargui et al. 2023; Messaoud 2022). For example, a hybrid variable neighbourhood metaheuristic was designed to solve a heterogeneous vehicle routeing problem, where a tabu search was used to explore the neighbourhood of the metaheuristic (Molina et al. 2020).

The Adaptive Large Neighborhood Search (ALNS), as an important heuristic algorithm, is popular for solving vehicle scheduling problems in various fields (Huang and Zhang 2023; Ji et al. 2020; Zhang, Guo, and Wang 2022a). One of the main reasons behind this is that the removal and repair operators in ALNS can be designed according to the characteristics of the researched problem. For instance, a heterogeneous electric vehicle scheduling problem was solved by an ALNS algorithm, which showed that the algorithm outperformed the common solvers for different scales of instances (Yu, Jodiawan, and Gunawan 2021).

The ALNS algorithm is also famous for solving vehicle scheduling with synchronisation or precedence constraints (Ali, Côté, and Coelho 2021; Lianes et al. 2021). For instance, Sarasola and Doerner (2020) designed an ALNS algorithm to solve a vehicle routeing problem with synchronisation constraints in urban area freight transportation, where two personalised operators were designed according to the time windows of customers. An ALNS algorithm was proposed to handle a vehicle routeing problem with time windows and synchronisation constraints, where the repair operators employed several linear programming models to evaluate the insertions (Hà et al. 2020). Wang, Moon, and Zhang (2022) gave an ALNS algorithm to solve a container drayage problem, in which the algorithm employed a mathematical model to decode a solution.

In summary, the HVS-P problem is motivated by the transportation services for tourists in scenic areas, in which the scheduling of heterogeneous vehicles is considered. An ALNS algorithm is designed to solve the problem efficiently. Therefore, this research contributes to the literature from both the viewpoint of problem description and the viewpoint of algorithm design.

3. Problem description

A tourism area comprises a set of scenic spots O and a gate e, where the gate is both the entrance and the exit. A fleet of heterogeneous vehicles with a set of vehicle types K is provided in the area. The number of each type of vehicle is supposed to be large enough. For a vehicle with type $k \in K$, the loading capacity of tourists is $Q_k(\geq 0)$; the fixed cost of using a vehicle on a working day is $f_k(\geq 0)$; the using cost of a vehicle for a unit of driving time is $d_k(\geq 0)$. Specifically, we have $Q_k < Q_{k'}, f_k < f_{k'}$, and $d_k < d_{k'}$ for two vehicle types $k < k'$. The maximum loading capacity among all types of vehicles is $Q_{max}(\geq 0)$.

Both the initial departure and the final destination of a vehicle are the gates. Tourists must take the aforementioned vehicles to travel between locations in the tourism area because private transports are forbidden in consideration of the public order. The driving speed of vehicles with or without tourists is assumed to be the same. The driving time of a vehicle between locations $i \in O \cup \{e\}$ and $j \in O \cup \{e\}$ is $\tau(i, j)$. Furthermore, we have $\tau(i, j) \geq 0$, $\tau(i,j) = \tau(j,i)$, and $\tau(i,i) = 0$ for any two locations i and j.

The tourists in the scenic area travel in groups, with the set of groups given as C. The area is opened at time T^A and is closed at time $T^B(T^A \leq T^B)$ in every working day. A tourist group $c \in C$ is composed of q_c tourists with an arrival time t_c, where t_c is the time when all tourists in the group arrive. The groups with a larger number of tourists than the maximum loading capacity of vehicles are divided in advance such that the number of tourists of any group is not larger than Q_{max}. The selected scenic spots of a tourist group c is O_c. Both the first and the last locations in O_c are the gate and the spots in O_c are visited in a predefined sequence. Furthermore, the visit duration of a group $c \in C$ at a spot $o \in O_c$ is $T_{co}(\geq 0)$.

When a tourist group has finished visiting a scenic spot, a vehicle should pick up the tourists at the spot and deliver the tourists to the next location. The coupling of a pickup task and its corresponding delivery task is a transportation request of a tourist group. The waiting time threshold of tourists in a transportation request is given as $L(\geq 0)$ in consideration of the satisfaction degree of tourists. The time for tourists to get on and get off a vehicle is ignored.

The HVS-P problem determines the vehicle routes and the vehicle type of each route simultaneously to satisfy all the precedence-constrained transportation requests of tourists, considering the loading capacity of vehicles and the waiting time threshold of tourists. The objective function of the problem is to minimise the total cost of the use and driving of all involved vehicles, as well as the waiting cost of tourists. The cost of a unit waiting time for tourists is $\rho(\geq 0)$.

4. Graphical formulation and mathematical model

A directed graph was introduced before the HVS-P problem was formulated as a mixed-integer linear programming model. One main reason is that even a similar but simpler vehicle scheduling problem in tourism areas is difficult to mathematically model directly (Zhang, Liu, and Feng 2021).

4.1. Graphical formulation

Pickup and delivery services are the basic components of the transportation requests of tourists. Specifically, the delivery of a tourist group to a location is called a delivery task. The pickup of a tourist group at a location is called a pickup task.

An example (Example 2) with one tourist group is presented in Figure 2 to illustrate the transportation requests of groups. The visit route of the tourist group is (e, o_1, o_3, o_2, e). That is, the tourist group should be picked up at the gate and delivered to scenic spot o_1, picked up at o_1 and delivered to o_3, picked up at o_3 and delivered to o_2, and finally picked up at o_2 and delivered to e. As shown in Figure 2, the group has a delivery task and a pickup task at each location. The delivery and the pickup tasks at the same location are involved in two different transportation requests with precedence constraints. The four transportation

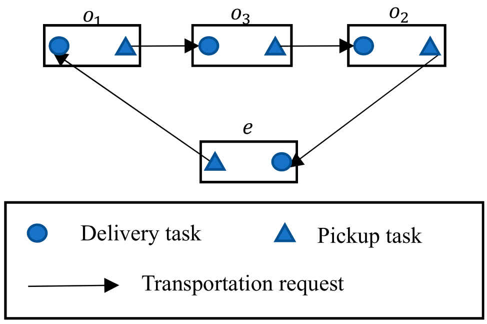

Figure 2. The transportation requests of Example 2.

requests of the group are precedence-constrained by the visit route and could be served by different vehicles.

The HVS-P problem can be described on a directed graph $G = (V, A)$. Specifically,

$$V = P \cup D \cup \{e\} \tag{1}$$

is the node set, where $P = \{1, \cdots, n\}$ is the pickup node set, $D = \{n+1, \cdots, 2n\}$ is the delivery node set. Node i in P and $i+n$ in D refer to the same group. Let $e = 0$ and Node 0 is a virtual node. As a comparison, $N = P \cup D$ represents the task node set. $n = \sum_{c \in C} (|O_c| - 1)$ is the total number of transportation requests, where $|\cdot|$ is the number of elements of a given set throughout this paper.

For a task node $i \in N$, the *group* attribute, say $\alpha(i)$, the *location* attribute, say $\beta(i)$, and the *group size* attribute, say $\gamma(i)$, are defined as the corresponding tourist group, location, and the number of tourists of task i, respectively. Furthermore, the delivery node that has the same group and location attributes as a pickup node $i \in P$ is recorded as $\theta(i)$.

In Graph G,

$$A = A_1 \cup A_2 \cup A_3 \cup A_4 \tag{2}$$

is the set of arcs, where

$$A_1 = \{(i,j) | i = 0, j \in P; \text{ or } i \in D, j = 0\}, \tag{3}$$

$$A_2 = \{(i,j) | i \in P, j \in P, \ \alpha(i) \neq \alpha(j); \text{ or } i \in D, j \in D, \alpha(i) \neq \alpha(j)\}, \tag{4}$$

$$A_3 = \{(i,j) | i \in P, j \in D, \ \alpha(i) \neq \alpha(j); \text{ or } i \in P, j = (n+i) \in D, \alpha(i) = \alpha(j)\}, \tag{5}$$

and

$$A_4 = \{(i,j) | i \in D, j \in P, \ \alpha(i) \neq \alpha(j); \text{ or } i \in D, j \in P, \alpha(i) = \alpha(j), \beta(j) \text{ locates later than } \quad (1)$$
$$\beta(i) \text{ in } O_{\alpha(i)}\} \tag{6}$$

In particular, set A_1 includes the arcs that leave or enter Node 0. Note that the arc (0, 0) does not exist. Set A_2 involves the arcs that transfer either from a pickup node to another pickup node, or from a delivery node to another delivery node. Such arcs that transfer either between the pickup nodes or between the delivery nodes for the same group are infeasible and hence discarded. Set A_3 contains the arcs that leave a pickup node and enter a delivery node. Specifically, for the same group $\alpha(i) = \alpha(j)$, the delivery node corresponding to a pickup node i can be only $n + i$. The other arcs are infeasible and hence removed. A_4 is the set of such arcs that transfer from a delivery node to a pickup node. Specifically, for the same group $\alpha(i) = \alpha(j)$, the transferring from a delivery node i to such a pickup node j that its corresponding spot $\beta(j)$ locates in front of spot $\beta(i)$ according to the order $O_{\alpha(i)}$ is infeasible and hence is removed. The travel time of an arc $(i, j) \in A$ is $t_{ij} = \tau(\beta(i), \beta(j))$.

The HVS-P problem described in Graph G can be summarised as follows: A set of vehicle routes as well as the vehicle type of each route with the minimal cost are to be determined simultaneously to satisfy all the tasks. Each task node in set N must be served exactly once. Each route starts from Node 0, travels along the feasible arcs in set A, and finally returns to Node 0. A transportation request of a tourist group on Graph G is recorded as $(i, n + i)$, which means picking up tourist group $\alpha(i)$ at the departure place $\beta(i)$ and delivering it to the destination $\beta(n + i)$. The precedence constraints of the starting times of task nodes, the waiting time threshold of transportation requests, and the loading capacity of vehicles are considered at the same time.

4.2. The mixed-integer linear programming model

The notations are as follows:

e	The gate of the tourism area
K	The set of vehicle types of heterogeneous vehicles
Q_k	The loading capacity of tourists of a vehicle with type $k \in K$
Q_{max}	The maximum loading capacity among all types of vehicles
f_k	The fixed cost of using a vehicle with type $k \in K$ on a working day
d_k	The cost of a vehicle with type $k \in K$ for a unit of driving time
T^A	The opening time of the tourist area on every working day
T^B	The closing time of the tourist area on every working day
t_c	The time when all tourists of the group $c \in C$ arrive at the scenic area
T_{co}	The visit duration of a group $c \in C$ at a spot $o \in O_c$
L	The waiting time threshold of tourists in a transportation request
ρ	The cost of a unit waiting time for tourists
G	The graph that is used to describe the problem
A	The arc set of Graph G
V	The node set of Graph G
N	The task node set
0	The virtual node located at the gate
P	The pickup task node set
D	The delivery task node set
n	The total number of transportation requests
$\alpha(i)$	The group attribute of a task node $i \in N$
$\beta(i)$	The location attribute of a task node $i \in N$
$\gamma(i)$	The group size attribute of a task node $i \in N$
$\theta(i)$	The delivery node that has the same group and the same location attributes as a pickup node $i \in P$

t_{ij} The travel time of an arc $(i,j) \in A$ on Graph G
$\boldsymbol{N}^*$ The set of positive integers

Two decision variables were introduced as

$$x_{ijk} = \begin{cases} 1, & \text{if a vehicle of type } k \in K \text{ serves an arc } (i,j) \in A, \\ 0, & \text{otherwise,} \end{cases}$$

y_i: the time when a task node $i \in N$ is served.

Four intermediate variables were introduced as

z_i: the number of tourists in a route after serving a node $i \in N$,

v_i: the index of a route that serves a node $i \in N$,

h: the longest waiting time for tourists,

y_i^{min}: the time when the tourist group $\alpha(i)$ finishes visiting the spot $\beta(i)$, where $i \in P$.

We formulate the HVS-P problem as the following mixed-integer linear programming (**MILP**) model:

$$\min \sum_{k\in K}\sum_{j\in P} f_k x_{0jk} + \sum_{k\in K}\sum_{(i,j)\in A} d_k t_{ij} x_{ijk} + \rho h, \tag{7}$$

subject to

$$\sum_{j\in P} x_{0jk} = \sum_{j\in D} x_{j0k}, \forall k \in K, \tag{8}$$

$$\sum_{k\in K}\sum_{j\in V \text{ and } (i,j)\in A} x_{ijk} = 1, \forall i \in N, \tag{9}$$

$$\sum_{j\in V \text{ and } (i,j)\in A} x_{ijk} - \sum_{j\in V \text{ and } (j,i)\in A} x_{jik} = 0, \forall i \in N, k \in K, \tag{10}$$

$$y_i + t_{ij} - y_j \le (T^B + t_{ij})(1 - x_{ijk}), \forall (i,j) \in A \setminus A_1, k \in K, \tag{11}$$

$$y_i^{min} \le y_i \le y_i^{min} + h, \forall i \in P, \tag{12}$$

$$y_i + t_{i,n+i} \le y_{n+i} \le y_i^{min} + t_{i,n+i} + h, \forall i \in P, \tag{13}$$

$$y_i^{min} = t_{\alpha(i)}, \forall\, i \in \{i | i \in P \text{ and } \beta(i) = e\}, \tag{14}$$

$$y_i^{min} = y_{\theta(i)} + T_{\alpha(i)\beta(i)}, \forall\, i \in \{i | i \in P \text{ and } \beta(i) \ne e\}, \tag{15}$$

$$y_i \le T^B,\ \forall\, i \in \{i | i \in D \text{ and } \beta(i) = e\}, \tag{16}$$

$$y_i + t_{i0} - T^B \le t_{i0}(1 - x_{i0k}), \forall i \in \{i | i \in D \text{ and } \beta(i) \ne e\}, k \in K, \tag{17}$$

$$y_i^{min} \ge T^A, \forall i \in P, \tag{18}$$

$$0 \le h \le L, \tag{19}$$

$$v_i = v_{n+i}, \forall i \in P, \tag{20}$$

$$j x_{0jk} \le v_j \le j x_{0jk} - |P|(x_{0jk} - 1), \forall\, j \in P, k \in K, \tag{21}$$

$$v_i + |P|(x_{ijk} - 1) \le v_j \le v_i + |P|(1 - x_{ijk}), \forall (i,j) \in A \setminus A_1, k \in K, \tag{22}$$

$$v_i \in \boldsymbol{N}^*, \forall i \in N, \tag{23}$$

$$-Q_{max}(1-x_{0jk}) \leq \gamma(j) - z_j \leq Q_{max}(1-x_{0jk}), \forall j \in P, k \in K, \quad (24)$$

$$-2Q_{max}(1-x_{ijk}) \leq z_i + \gamma(j) - z_j \leq 2Q_{max}(1-x_{ijk}), \forall (i,j) \in A \setminus A_1, k \in K, \quad (25)$$

$$0 \leq z_i \leq Q_k + (Q_{max} - Q_k)(1-x_{ijk}), \forall (i,j) \in A \setminus A_1, k \in K, \quad (26)$$

$$z_i \leq Q_{max}(1-x_{i0k}), \forall\, i \in D, k \in K, \quad (27)$$

$$x_{ijk} \in \{0,1\},\ \forall (i,j) \in A, k \in K. \quad (28)$$

In Model MILP, the objective function (7) minimises the total serving cost, including the fixed-use cost of vehicles and the variable driving cost of vehicles, and the cost of the longest waiting time of tourists. Constraints (8) guarantee that, for each type k, the number of vehicles initially starting from the gate is equal to the number of vehicles finally returning to the gate. Constraints (9) indicate that only one arc flows out from every task node. Constraints (10) imply that the number of incoming arcs is the same as the number of outgoing arcs of every task node. Constraints (9) and (10) illustrate that any task node must be carried out by a vehicle of a type just once.

Constraints (11)–(19) are the formulations of the starting times. In particular, Constraints (11) define the relationship of the starting times of any two neighbouring task nodes in a route, removing the sub-tours among task nodes simultaneously. If $x_{ijk} = 0$, Constraints (11) are relaxed automatically. If $x_{ijk} = 1$, Constraints (11) become

$$y_i + t_{ij} \leq y_j, \forall (i,j) \in A \backslash A_1.$$

Constraints (12) ensure that the starting time of a pickup node i is later than or equal to the time when group $\alpha(i)$ finishes visiting spot $\beta(i)$, but it should be earlier than the longest waiting time of tourists after group $\alpha(i)$ finishes visiting spot $\beta(i)$. Constraints (13) illustrate that the starting time of a delivery node $n+i$ is later than or equal to the starting time of its corresponding pickup node i plus the travelling time of the transportation request $(i, n+i)$. Meanwhile, y_{n+i} is earlier than the sum of the time when group $\alpha(i)$ finishes visiting spot $\beta(i)$, the travelling time of transportation request $(i, n+i)$, and the longest waiting time of tourists. Constraints (14) and (15) denote y_i^{min} of a pickup node i when picking up tourists at the gate and at a scenic spot, respectively. In particular, $t_{\alpha(i)}$ is the arrival time of group $\alpha(i)$ at the tourism area. The item $y_{\theta(i)} + T_{\alpha(i)\beta(i)}$ is the time when group $\alpha(i)$ finishes visiting spot $\beta(i)$. Constraints (16) ensure that the time of delivering a tourist group to the gate is earlier than the ending time of the area. Constraints (17) limit that the finish time of a route is earlier than the ending time of the area. If $x_{i0k} = 0$, Constraints (17) are relaxed automatically because they become

$$y_i \leq T^B, \forall i \in D.$$

If $x_{i0k} = 1$, they become

$$y_i + t_{i0} \leq T^B, \forall i \in D.$$

Constraints (18) and (19) denote the types of decision variables y_i^{min} and h, respectively.

Constraints (20)–(23) guarantee that the pickup and delivery tasks in each transportation task $(i, i+n)$ must be carried out by the same route. The introduction of a decision variable v_i in Constraints (20)–(23) is significant for solving the mathematical model, which has been verified in the research of Liu, Moon, and Zhang (2024). Specifically, Constraints (20) imply that the route index of node i is the same as that of node $(i+n)$. Constraints (21) guarantee that the index of a route is the index of the first task node in the route. If $x_{0jk}=0$, Constraints (21) are relaxed. If $x_{0jk}=1$, Constraints (21) become $v_j=j, \forall j \in P$. Constraints (22) denote that the indexes of all tasks in a route are same. If $x_{ijk}=0$, Constraints (22) are relaxed. If $x_{ijk}=1$, Constraints (22) become $v_j=v_i, \forall i \in N, j \in N$. Constraints (23) show the type of decision variable v_i, where $\boldsymbol{N}^*$ is the set of positive integers.

Constraints (24)–(27) limit the loading capacities of vehicles. Constraints (24)–(25) update the number of tourists of a vehicle after serving an arc. If $x_{0jk}=0$, Constraints (24) are relaxed. If $x_{0jk}=1$, they become

$$z_j=\gamma(j),\ \forall j \in P.$$

Similarly, if $x_{ijk}=0$, Constraints (25) are relaxed. If $x_{ijk}=1$, they become

$$z_i+\gamma(j)=z_j,\ \forall (i,j) \in A \setminus A_1.$$

The loading capacity of any type of vehicle is ensured by Constraints (26). If $x_{ijk}=0$, Constraints (26) are relaxed. If $x_{ijk}=1$, they become

$$0 \le z_i \le Q_k, \forall (i,j) \in A \setminus A_1, k \in K.$$

Constraints (27) limit that every route returns to the gate without loading any tourists. If $x_{i0k}=0$, Constraints (27) are relaxed. If $x_{i0k}=1$, Constraints (27) become

$$z_i=0, \forall i \in D.$$

Finally, Constraints (28) show the type of decision variable x_{ijk}.

The MILP model is mixed-integer, linear, and deterministic. However, the decision variable y_i makes the model harder to solve because it is not only constrained by the time related to vehicles in Constraints (11) but also precedence-constrained by the time related to tourists in Constraints (12)–(15). Particularly, the decision of the variable y_i is critical because it would also influence the value of the intermediate variable h in the objective function.

4.3. Property of the problem

Theorem 1: *The problem of HVS-P is NP-hard.*

Proof: The HVS-P problem degenerates to the dial-a-ride problem (Cordeau and Laporte 2007) if the type of vehicles is reduced to one and the precedence constraints among transportation requests of tourists are relaxed. Furthermore, the dial-a-ride problem reduces to a vehicle routeing problem with pickup and delivery (Berbeglia et al. 2007) if the constraints and the optimisation objective from the human perspective is ignored. The vehicle routeing problem with pickup and delivery is NP-hard because the vehicle routeing problem is NP-hard. As a result, the HVS-P problem is NP-hard. ∎

5. The ALNS algorithm

An Adaptive Large Neighborhood Search (ALNS) algorithm is designed for the HVS-P problem. The main reasons are as follows: Firstly, the problem is NP-hard and the MILP model is difficult to solve. The design of a meta-heuristic to solve the problem is a better choice. Secondly, the ALNS algorithm is more flexible than some other meta-heuristics because particular operators can be designed for different types of problems. Furthermore, the ALNS algorithm has been popular and successful in solving various scheduling problems (Huang and Zhang 2023; Zhang, Guo, and Wang 2022a). The HVS-P problem is distinct from the existing problems such that it cannot be handled directly by the algorithms in literature. Consequently, an ALNS algorithm that is particularly designed for the HVS-P problem is meaningful.

5.1. Framework of the ALNS algorithm

In the framework of the ALNS algorithm presented in Figure 3, an initial solution S^{Ini} is generated first. Both the best-so-far solution S^{Opt} and the current solution S^{Cur} are initialised as solution S^{Ini}. Two removal and two repair operators were introduced. The score and the weight of each operator are initialised as zero and one, respectively.

The algorithm is performed in a loop after the initialisation. In each iteration, both a removal operator and a repair operator are selected according to their historical performances using the roulette wheel selection. A solution S^{Cur_DR} is generated by destroying and reconstructing the current solution using the selected removal and repair operators, respectively. The acceptance mechanism of simulated annealing is used to update solutions S^{Opt} and S^{Cur}. If S^{Cur_DR} is better than S^{Opt}, both S^{Opt} and S^{Cur} are updated as S^{Cur_DR}; if S^{Cur_DR} is only better than S^{Cur}, S^{Cur} is updated as S^{Cur_DR}; otherwise, S^{Cur_DR} is accepted to update S^{Cur} with a probability. Furthermore, both the score η^{Cur} and the number of calls n^{Call} of a selected operator are updated in the iteration.

The weight of an operator is updated according to its score and the number of calls for each N^{Wgt} number of iterations in the algorithm. Furthermore, if solution S^{Opt} has not been updated for a given number of iterations, solution S^{Cur} is set as solution S^{Opt}. The algorithm stops with solution S^{Opt} obtained if the maximum number of iterations, say N^{Ite}, is met.

5.2. Solution representation and evaluation

The fundamental building block for the solution representation of the problem is a *trip-segment*, which is defined as one transportation request, or a combination of several transportation requests with different group attributes. A vehicle is empty when it starts to serve a trip-segment, and it is also empty after finishing serving the segment.

A solution to the problem is represented as a set of vehicle routes. An order encoding scheme is used to represent each route in the ALNS algorithm. For example, a vehicle route can be encoded as $(k, e, r_0, r_1, ..., r_i, e)$. That is, the kth vehicle starts from the gate, serves a set of trip-segments in order, and returns to the gate finally.

The evaluation of a solution includes determining the starting time of each task, checking the feasibility, and calculating the objective value simultaneously. In particular, the feasibility of a solution is checked according to the precedence of tasks, the loading capacity of vehicles, and the waiting time threshold of transportation requests at the same time.

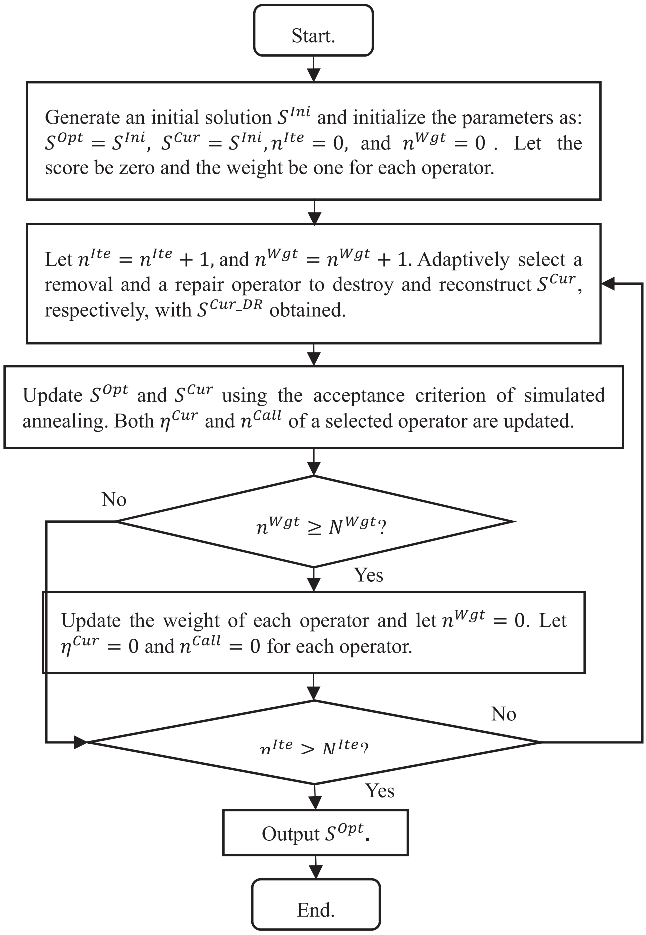

Figure 3. Framework of the ALNS algorithm.

5.3. *Initial solution*

An initial solution is generated based on the set of trip-segments, say $R^S = \{(i, n+i)|\forall i \in P\}$, in which a trip-segment is a transportation request of a tourist group, and the waiting time of each transportation request is set as zero. Therefore, the travelling time of a trip-segment $(i, n+i)$ is $t_{i,n+i}$. Furthermore, the starting time of each pickup or delivery task of the problem can be calculated according to the visit routes of tourist groups as:

$$y_i = t_{\alpha(i)}, \forall i \in \{i|i \in P \text{ and } \beta(i) = e\}, \tag{29}$$

$$y_i = y_{\theta(i)} + T_{\alpha(i)\beta(i)}, \forall i \in \{i|i \in P \text{ and } \beta(i) \neq e\}, \tag{30}$$

$$y_{n+i} = y_i + t_{i,n+i}, \forall i \in P. \tag{31}$$

As a result, the problem with the introduction of trip-segment set R^S can be described on a directed graph $G' = (V', A')$ when each trip-segment is regarded as a vertex, where $V' = \{0, 1, \cdots, |R^S|\}$ is the node set, and $A' = \{(u, v)|u \in V', v \in V', u \neq v\}$ is the arc set.

The following parameters were introduced for a vertex $u \in V'\backslash\{0\}$ on Graph G': the *vehicle-type* attribute $k_u \in K$, which is defined as the minimum vehicle type whose loading capacity of vehicles is feasible for serving vertex u; the original task r_u^O, the terminal task r_u^D, the starting time t_u^O, the ending time t_u^D, and the travelling time T_u^{RS}. For Node 0, the location of both the origin and the terminal is the gate. The travelling time between vertexes $u \in V'$ and $v \in V'$ is $t_{uv}^{RS} = \tau(\beta(r_u^D), \beta(r_v^O))$.

Only one decision variable was introduced as

$$x'_{uvk} = \begin{cases} 1, & \text{if a route of type k} \in K \text{ travels from vertex } u \in V' \text{ to vertex } v \in V', \\ 0, & \text{otherwise.} \end{cases}$$

The initial solution of the algorithm can be obtained easily using the following integer linear programming (**ILP-Init**) model:

$$\min \sum_{k \in K} \sum_{v \in V'\backslash\{0\}} f_k x'_{0vk} + \sum_{k \in K} \sum_{(u,v) \in A'} d_k (t_{uv}^{RS} + T_v^{RS}) x'_{uvk}, \tag{32}$$

subject to

$$\sum_{v \in V'\backslash\{0\}} x'_{0vk} = \sum_{v \in V'\backslash\{0\}} x'_{v0k}, \forall k \in K, \tag{33}$$

$$\sum_{k \in K} \sum_{v \in V'} x'_{uvk} = 1, \forall u \in V'\backslash\{0\}, \tag{34}$$

$$\sum_{v \in V'} x'_{uvk} - \sum_{v \in V'} x'_{vuk} = 0, \forall u \in V'\backslash\{0\}, k \in K, \tag{35}$$

$$t_u^D + t_{uv}^{RS} - t_v^O \le (T^B + t_{uv}^{RS}) \left(1 - \sum_{k \in K} x'_{uvk}\right), \forall u, v \in V'\backslash\{0\}, k \in K \tag{36}$$

$$\max\{k_u, k_v\} \le k - |K|(x'_{uvk} - 1), \forall u, v \in V'\backslash\{0\}, k \in K. \tag{37}$$

In the ILP-Init model, the objective function (32) minimises the total use and driving cost of all vehicles. Notably, the waiting cost of tourists has been supposed to be zero in set R^S. Constraints (33) denote that, for each type k, the number of vehicles initially starting from the gate is equal to the number of vehicles finally returning to the gate. Constraints (34) ensure that each vertex (trip-segment) $u \in V'\backslash\{0\}$ must be handled just once. Constraints (35) limit the flow conservation of each vertex. Constraints (36) indicate the relationship between the starting times of any two continuous vertexes in a route. Sub-tours among vertexes can also be removed by Constraints (36). If $\sum_{k \in K} x'_{uvk} = 0$, Constraints (36) are relaxed. If $\sum_{k \in K} x'_{uvk} = 1$, Constraints (36) become

$$t_u^D + t_{uv}^{RS} \le t_v^O, \forall u, v \in V'\backslash\{0\}.$$

Constraints (37) imply that the vehicle type of a route must be larger than or equal to the vehicle-type attribute of any trip-segment in the route. If $x'_{uvk} = 0$, Constraints (37) are relaxed. If $x'_{uvk} = 1$, Constraints (37) become

$$\max\{k_u, k_v\} \le k, \forall u, v \in V'\backslash\{0\}, k \in K.$$

5.4. Removal operators

A random removal operator and a relatedness-based removal operator were designed in the ALNS algorithm. A trip-segment is removed from the current solution as a unit in a removal operator. The number of the removed trip-segments is $m \in (0.1n, \lceil \pi n \rceil)$, where $\lfloor \cdot \rfloor$ is the downward integer function, $\lceil \cdot \rceil$ is the upward integer function, and π is a random parameter that belongs to (0, 1). The set of trip-segments of the current solution S^{Cur} is recorded as R^{cur}. Three parameters were given in a removal operator as the following: the set of the removed trip-segments R^{Rem}, the destroyed solution S^{Cur_Des}, and the set of trip-segments R^{Cur_Des} of solution S^{Cur_Des}.

5.4.1. Random removal operator

We select m trip-segments randomly and remove them from the solution. The details are shown as follows:

Step 1: Initialise solution S^{Cur_Des} as S^{Cur}, and set R^{Cur_Des} as R^{Cur}. Let set R^{Rem} be empty.

Step 2: Select a trip-segment r from set R^{Cur_Des} randomly. Put segment r into set R^{Rem}, and remove it from both set R^{Cur_Des} and solution S^{Cur_Des}.

Step 3: If $|R^{Rem}| < m$, go to Step 2; otherwise, the procedure stops with sets R^{Rem} and R^{Cur_Des}, and solution S^{Cur_Des} provided.

5.4.2. Relatedness-based removal operator

In the relatedness-based removal operator, the following parameters of a trip-segment i were introduced: the number of transportation requests n_i, the set of task nodes V_i, the starting time of the first task node t_i^O, and the starting time of the last task node t_i^D.

Definition 1: The *relatedness* of trip-segments i and j is

$$R(i,j) = \epsilon_n \frac{n_i + n_j}{2n} + \epsilon_d \frac{\sum_{a \in V_i} \sum_{b \in V_j} t_{ab}}{D} + \epsilon_t \varphi_{ij}, \tag{38}$$

where, ϵ_n, ϵ_d and ϵ_t are the weight coefficients. Specifically, the first component is about the total number of trip-segments. Segments i and j are more related to each other when the number of transportation requests is smaller. The second component is related to the total travel time of the two segments, where

$$D = \sum_{a \in N} \sum_{b \in N, b \neq a} t_{ab} \tag{39}$$

is the sum of Euclidean distance between any two tasks in set N. Segments i and j are more related when the total travel time is smaller. Furthermore, the last component is related to the overlap or the gap of time durations of the two segments, where

$$\varphi_{ij} = \begin{cases} \frac{\min\{t_i^D, t_j^D\} - \max\{t_i^O, t_j^O\}}{T^A - T^B}, & \text{if } t_j^O \le t_i^O \text{ and } t_j^D \ge t_i^O; \\ & \text{or } t_i^O < t_j^O \le t_i^D, \\ \frac{\min\{|t_i^O - t_j^D|, |t_j^O - t_i^D|\}}{T^B - T^A}, & \text{otherwise.} \end{cases} \tag{40}$$

Segments i and j are more related to each other when the overlap of the time durations is larger or the gap of the time durations is smaller. The denominator in

Function (40) is set as negative $(T^A - T^B)$ when the time durations are overlapped ($t_j^O \leq t_i^O$ and $t_j^D \geq t_i^O$, or $t_i^O < t_j^O \leq t_i^D$). Consequently, in consideration of the three components in the definition of relatedness, the smaller the value of $R(i,j)$ is, the more related trip-segments i and j would be.

The details of the relatedness-based removal operator are shown as the following:

Step 1: Initialise solution S^{Cur_Des} as S^{Cur}, and set R^{Cur_Des} as R^{Cur}. Let set R^{Rem} be empty.

Step 2: Select a trip-segment from set R^{Cur_Des} randomly. Put the selected segment into set R^{Rem}, and remove it from both set R^{Cur_Des} and solution S^{Cur_Des}.

Step 3: Randomly choose a trip-segment r from set R^{Rem}. Let r' be the segment with the minimum value of $R(r, r')$ in set R^{Cur_Des}. Put r' into set R^{Rem}, and remove it from both set R^{Cur_Des} and solution S^{Cur_Des}.

Step 4: If $|R^{Rem}| < m$, return to Step 3; otherwise, the procedure stops with sets R^{Rem} and R^{Cur_Des} and solution S^{Cur_Des} provided.

5.5. Repair operators

A two-stage greedy (**TG**) operator and a mathematical insertion (**MI**) operator were introduced to repair a destroyed solution in the ALNS algorithm. All tasks in the removed trip-segment set R^{Rem} are inserted into the destroyed solution S^{Cur_Des} using a repair operator, where both set R^{Rem} and solution S^{Cur_Des} are obtained from a removal operator. A solution S^{Cur_Rep} would be obtained using a repair operator finally.

5.5.1. TG repair operator

The TG repair operator consists of two stages. A repaired solution S^{Cur_Rep} is generated in the first stage and is evaluated by a mathematical model in the second stage. Specifically, in the first stage, all tasks in set R^{Rem} are greedily inserted into the destroyed solution with the loading capacity of vehicles guaranteed. The evaluation of a solution in the second stage comprises determining the starting time of each task, checking the precedence of tasks and the waiting time threshold of transportation requests, and calculating the objective value of the solution.

Three types of insertions are considered in the first stage and a transportation request is considered as an inserted unit. In a **Type 1** insertion, a request is inserted into a trip-segment of the destroyed solution. Its feasibility is checked by the maximum loading capacity of vehicles. In a **Type 2** insertion, a transportation request is inserted into a vehicle route of the destroyed solution as an independent trip-segment. Its feasibility is checked by the precedence of the tasks in the selected route. In a **Type 3** insertion, a request as a trip-segment is put into an empty route, and the route is added to the destroyed solution.

Figure 4 illustrates three types of insertions when inserting transportation request $(i, n+i)$ into an original vehicle route in the destroyed solution (Figure 4(a)). Particularly, Figure 4(b) shows an example of inserting request $(i, n+i)$ into a trip-segment of the solution. An integrated trip-segment $P_2 \rightarrow i \rightarrow D_2 \rightarrow n+i$ is obtained finally. Figure 4(c) is an example of inserting request $(i, n+i)$ into the original vehicle route directly. Figure 4(d) presents the situation of adding a vehicle route into the solution to serve request $(i, n+i)$.

The details of the first stage of the TG repair operator are shown as follows.

Step 1: Initialise solution S^{Cur_Rep} as S^{Cur_Des}.

$(0, P_1 \rightarrow D_1, P_2 \rightarrow D_2, P_3 \rightarrow D_3, 0)$

(a) An original route in the destroyed solution.

$(0, P_1 \rightarrow D_1, P_2 \rightarrow i \rightarrow D_2 \rightarrow n + i, P_3 \rightarrow D_3, 0)$

(b) An example of Type 1 insertion.

$(0, P_1 \rightarrow D_1, i \rightarrow n + i, P_2 \rightarrow D_2, P_3 \rightarrow D_3, 0)$

(c) An example of Type 2 insertion.

$(0, P_1 \rightarrow D_1, P_2 \rightarrow D_2, P_3 \rightarrow D_3, 0)$

$(0, i \rightarrow n + i, 0)$

(d) An example of Type 3 insertion.

Figure 4. Three types of insertions in the TG repair operator.

Step 2: If set R^{Rem} is empty, output solution S^{Cur_Rep} and the algorithm stops; otherwise, remove the first request $(i, n+i)$ in the first trip-segment r of set R^{Rem}. If segment r is empty, remove it from set R^{Rem}.

Step 3: Insert request $(i, n+i)$ into solution S^{Cur_Rep} using Type 1. The minimum increase of driving time W^{Best} is set as a sufficiently large constant. The integrated trip-segment $l_{u^*v^*}$ is initialised as request $(i, n+i)$. The selected segment is the first $(u = 1)$ trip-segment in the first $(v = 1)$ vehicle route of solution S^{Cur_Rep}.

Step 4: If it is feasible to insert request $(i, n+i)$ into the uth segment in the vth route, let the integrated segment be l_{uv} and let its increase of driving time be W_{uv}. If $W_{uv} < W^{Best}$, update the parameters as $W^{Best} = W_{uv}, u^* = u, v^* = v$, and $l_{u^*v^*} = l_{uv}$.

Step 5: If the uth trip-segment is the last segment of solution S^{Cur_Rep}, go to Step 6. If the uth trip-segment is the last segment in the vth route but the vth route is not the last one in solution S^{Cur_Rep}, let $v = v + 1, u = 1$, and return to Step 4; otherwise, let $u = u + 1$, and go to Step 4.

Step 6: If Type 1 insertion is feasible, the u^*th trip-segment in the v^*th route of solution S^{Cur_Rep} is updated as segment $l_{u^*v^*}$, and go to Step 2; otherwise, select the position (u^{T2}, v^{T2}) with the minimum increase of driving time when inserting request $(i, n+i)$ into solution S^{Cur_Rep} using Type 2 insertion.

Step 7: If Type 2 insertion is feasible, insert request $(i, n+i)$ into position (u^{T2}, v^{T2}) of solution S^{Cur_Rep}; otherwise, insert request $(i, n+i)$ into solution S^{Cur_Rep} using Type 3 insertion. Go to Step 2.

Given a repaired solution S^{Cur_Rep} generated from the first stage of the TG operator, the vehicle type, say k^R, of a route R is the maximum vehicle-type attribute among all trip-segments in route R. Furthermore, the set of arcs of route R is

$$A_R^{GF} = \{(i_u, i_{u+1}) | i_u \in R, u = 0, \cdots, |R| - 1\}. \tag{41}$$

Consequently, the set of arcs of solution S^{Cur_Rep} is

$$A^{GF} = \bigcup_{R \in S^{Cur\_Rep}} A_R^{GF}. \tag{42}$$

The summation of the fixed cost and the driving cost of all vehicles involved in solution S^{Cur_Rep} is

$$W\_ft = \sum_{R \in S^{Cur\_Rep}} \left(f_{k^R} + d_{k^R} \sum_{(u,v) \in A_R^{GF}} t_{uv} \right). \tag{43}$$

In the second stage of the TG operator, three variables about starting times introduced in the MILP model, y_i $(i \in N)$, h, and $y_i^{min}(i \in P)$ are still used. A linear programming (**LP-TG**) model based on the MILP model is formulated to evaluate a repaired solution as the following:

$$\min W\_ft + \rho h, \tag{44}$$

subject to

$$y_i + t_{ij} \leq y_j, \forall (i,j) \in A^{GF}, \forall i,j \in N, \tag{45}$$

and Constraints (12) – (19).

In the LP-TG model, the objective function (44) minimises the total cost, and only the cost of the longest waiting time of tourists should be determined. Constraints (45) indicate the relationship between the starting times of any two neighbourhood tasks in a route.

The LP-TG model performs quickly and efficiently when evaluating solution S^{Cur_Rep}, because the routes have been constructed and the decision variables only involve the starting times. If solution S^{Cur_Rep} is feasible, its objective value can be calculated directly.

5.5.2. MI repair operator

The trip-segments in set R^{Cur_Des} are set as the given segments in the repaired solution of the MI repair operator. The problem with a number of given trip-segments is a sub-problem of the HVS-P problem. It can also be solved using the MILP model. Let A^{MF} be the set of arcs of the given trip-segments. For each trip-segment r in set R^{Cur_Des}, arc (i_u, i_{u+1}) of segment r is put into set A^{MF}, where $i_u \in r$ and $u = 0, \cdots, |r| - 1$. We have

$$\sum_{k \in K} x_{ijk} = 1, \forall (i,j) \in A^{MF}. \tag{46}$$

Constraints (46) guarantee that each arc in A^{MF} must be served by a vehicle exactly once. Furthermore, let N_i^- (N_i^+) be the set of the former (latter) tasks of task i in the given visit route of group $\alpha(i)$. If the group attributes $\alpha(i)$ and $\alpha(j)$ are different for an arc (i,j)in A^{MF}, we have

$$\sum_{k \in K} x_{uvk} = 0, \forall u \in N_i^-, v \in N_j^+, (u,v) \in A, (i,j) \in A^{MF}, \alpha(i) \neq \alpha(j), \tag{47}$$

and

$$\sum_{k \in K} x_{uvk} = 0, \forall u \in N_i^+, v \in N_j^-, (u,v) \in A, (i,j) \in A^{MF}, \alpha(i) \neq \alpha(j). \tag{48}$$

Constraints (47)–(48) limit the decision variables involving infeasible arcs to zero, which indicates that a route cannot travel from a former (latter) task $u \in N_i^-$ (N_i^+) of task i to a latter (former) task $v \in N_j^+(N_j^-)$ of task j. One reason is that it violates the precedence constraints in the given visit routes of tourist groups $\alpha(i)$ and $\alpha(j)$ when arc $(i,j) \in A^{MF}$ has been guaranteed in Constraints (46).

An enhanced model with an introduction of Constraints (46)–(48) to the MILP model, called the **MILP-MI** model, is used in the MI operator to obtain the repaired solution. The MILP-MI model solves a restricted sub-problem of the HVS-P problem so that it would be more efficient than the MILP model in terms of the calculation time.

6. Validation and evaluation

This section evaluates the problem as well as the algorithm. Specifically, Section 6.1 gives information of equipment, software, and instances of the experiments. Section 6.2 tunes the main parameters of the algorithm. Section 6.3 validates the MILP model and the ALNS algorithm. Section 6.4 compares the ALNS algorithm to a math-heuristic algorithm. Section 6.5 explores the advantages of using heterogeneous vehicles. Section 6.6 analyses the sensitivity of the cost coefficient of waiting time and the degree of dispersion of the scenic spots in a tourism area. Furthermore, Section 6.7 gives a real-size instance study.

6.1. Setting of experiments

6.1.1. Hardware and software

A personal computer with 3.30 GHz CPU and 4.0 GB RAM was used to carry out all the experiments. All procedures are coded in C++. All mathematical models are solved by IBM ILOG CPLEX 12.6.1. The CPLEX software handles mathematical models using the branch-and-bound algorithm. The *callback* function in the solver provides the objective value, the best lower bound, and the gap between the objective value and the best lower bound at each search node of the branch-and-bound algorithm.

Each run of the solver for a model cannot be longer than 7, 200 s, except for the MILP-MI model. The longest calculation time of the MILP-MI model was set according to the solution information obtained at the root node of the branch-and-bound algorithm. If the solver cannot provide a feasible solution at the root node, the longest computation time of the MILP-MI model is 20 s; otherwise, it is set considering the value of the gap, say $g^{Root}(\geq 0)$, between the objective value and the best lower bound of the root node. If $g^{Root} \leq G_1^{Root}$, the longest computation time is 100 s; if $G_1^{Root} < g^{Root} \leq G_2^{Root}$, the longest computation time is 60 s; if $g^{Root} > G_2^{Root}$, the longest computation time is 40 s. Both G_1^{Root} and G_2^{Root} are the parameters in the range of (0,1), and $G_1^{Root} \leq G_2^{Root}$, which are tuned in Section 6.2.

6.1.2. Generation of instances

The existing instances for some other problems do not lend themselves easily to this research, even after minor modifications. One reason is that the precedence constraints in the HVS-P problem depend on the characteristics of tourists' transportation requests in the tourism area, which is quite distinct from the existing research. Consequently, small-size instances S1–S15, middle-size instances M1–M15, and large-size instances L1–L15, were randomly generated with the number of task nodes ranging from 6 to 120.

The opening time of a tourism area is set as [8 : 00, 19 : 00]. The locations of scenic spots and the distances among locations refer to the data of Qiandao Lake in China in the Baidu

Table 3. Parameters of the heterogeneous vehicles.

Vehicle type k	Q_k	f_k	d_k
1	22	60	1.0
2	35	62	1.1
3	49	64	1.2

map, where the number of scenic spots is seven. The set of vehicle types is $K = \{1, 2, 3\}$. The loading capacity, the fixed cost, and the driving cost for different types of vehicles are presented in Table 3. The cost coefficient of the longest waiting time, ρ, is two per minute unless explicitly stated.

For the arriving tourists, the number of scenic spots and the visit time of a group at a spot are generated randomly from 1 to 5, and from 50 to 120 min, respectively. Parameters t_c and q_c of tourist group c are generated randomly from 8:00 to 13:00, and from 3 to 40, respectively. The number of tourist groups is set in the range of 3–20. The waiting time threshold is 30 min. The given data of all instances are available in the Figshare repository (https://doi.org/10.6084/m9.figshare.24053652).

6.2. Parameter setting and tuning of the ALNS algorithm

The parameters ϵ_n, ϵ_d, and ϵ_t in the relatedness-based operator are 0.8, 1.6, and 1.4, respectively. The maximum number of iterations N^{Wgt} to update the weights of operators is 5. The parameter μ that is used to update the weight of an operator is 0.3. The maximum number of iterations without improving the best-so-far solution is 4.

Four key parameters of the algorithm, including the maximum number of iterations N^{Ite}, parameters G_1^{Root} and G_2^{Root} that determine the longest computation time of the MILP-MI model, and parameter π that is used to calculate the number of removed elements in the removal operators, are tuned using Instances S5, M5, and L5. When changing a parameter, the other parameters are fixed. The potential value of a parameter with the minimum objective value, on average, among the three instances is chosen and fixed. The details of the parameter tuning are presented in Table 4. In particular, parameter N^{Ite} is set considering the scales of instances.

6.3. Comparison between the MILP model and the ALNS

All instances were solved by both the MILP model and the ALNS algorithm. For each instance, the ALNS algorithm runs three times independently and the best solution is selected unless explicitly stated.

Table 4. Potential values and tuning results of the key parameters.

Parameters	Instances	Potential values	Tuning results
N^{Ite}	S1–S15	15; 25; 35	15
	M1–M15	50; 70; 100	70
	L1–L15	50; 70; 100	100
G_1^{Root}	S1–S15, M1–M15, L1–L15	0.15; 0.25; 0.35	0.25
G_2^{Root}	S1–S15, M1–M15, L1–L15	0.30; 0.45; 0.60	0.45
π	S1–S15, M1–M15, L1–L15	0.20; 0.30; 0.40	0.20

Table 5. Comparison results of Instances S1–S15.

Instance	CPLEX OBJ.	CPLEX CPU (s)	ALNS OBJ.	ALNS CPU (s)	OBJ. Gap (%)
S1	251.60	1.06	251.60	5.69	0
S2	252.15	0.65	252.15	5.27	0
S3	157.86	0.84	157.86	6.90	0
S4	217.79	0.64	217.79	6.04	0
S5	258.40	2.43	258.40	6.98	0
S6	221.40	0.64	221.40	6.01	0
S7	406.37	167.66	406.37	73.98	0
S8	317.15	26.54	317.15	12.60	0
S9	407.68	371.73	413.08	22.23	−1.32
S10	288.83	6.14	288.83	6.38	0
S11	348.93	12.55	348.93	8.32	0
S12	253.74	2.09	253.74	7.34	0
S13	321.75	3.69	321.75	8.31	0
S14	251.92	1.97	251.92	6.81	0
S15	296.88	1.42	296.88	5.73	0
Average		40.00		12.57	−0.09

As shown in Table 5, the optimum solutions for Instances S1–S15 can be obtained by the solver. The ALNS algorithm provides the optimum solutions for Instances S1–S8, S10–S15, and a feasible solution for Instance S9 with a gap of 1.32%. The algorithm takes a little longer calculation time for Instances S1–S6, S10, and S12–S15 when compared to the model. The main reason behind this is that both the random factor and the iteration operation exist in the algorithm. Furthermore, the calculation speed of the algorithm is faster than that of the model for Instances S7–S9 and S11. The experiment validates both the MILP model and the ALNS algorithm.

The results of both the middle- and large-size instances are shown in Table 6. The algorithm outperforms the solver for middle- and large-size instances in terms of both the objective value and the calculation speed. Specifically, the improvements regarding the objective values for Instances M1–M15 and L1–L15 are 7.22% and 25.15% on average, respectively. The calculation time of the ALNS algorithm for Instances M1–M15 and L1–L15 are 434.56 s and 2,192.10 s on average, respectively. The ALNS algorithm provides a feasible solution for each instance within 3,600 s but the calculation time of the solver is 7,200 s for each instance. Consequently, the ALNS algorithm provides much better solutions than the MILP model, and within a shorter time.

Furthermore, the deviation of the objective values among three repeats of both the middle- and large-size instances are shown in the last column of Table 6. The maximum deviation is 5.50%, while the minimum deviation of that is 0. Even the deviation value of 5.50% should be acceptable in practice. As a result, the quality of solutions solved by the algorithm is stable.

6.4. *Comparison to a math-heuristic algorithm*

The ALNS algorithm is compared to a math-heuristic (MH) algorithm to test its advantage for solving the HVS-P problem. The MH is a variant of the algorithm in Liu, Moon, and Zhang (2024) and the modifications are as follows: First, the longest waiting time of tourists in a solution-seed is calculated as

$$h = \max_{i \in P}(y_{n+i} - y_i),$$

Table 6. Comparison results for Instances M1–M15 and L1–L15.

	CPLEX		ALNS			
Instance	OBJ.	CPU (s)	OBJ.	CPU (s)	OBJ. Gap (%)	The deviation of OBJ. (%)
M1	476.41	7200	420.50	642.80	11.74	1.30
M2	687.06	7200	514.62	529.50	25.10	2.28
M3	597.73	7200	534.40	265.91	10.60	0.00
M4	504.10	7200	447.11	1091.60	11.31	3.22
M5	660.81	7200	510.64	772.91	22.73	4.20
M6	461.15	7200	456.06	247.31	1.10	3.18
M7	473.16	7200	446.16	312.02	5.71	0.90
M8	478.40	7200	478.40	143.95	0.00	2.00
M9	443.80	7200	443.80	343.55	0.00	0.00
M10	491.98	7200	487.48	207.08	0.91	0.89
M11	481.40	7200	466.36	423.86	3.12	2.47
M12	507.50	7200	442.81	484.06	12.75	0.66
M13	574.18	7200	566.95	347.37	1.26	0.00
M14	420.23	7200	411.85	451.87	1.99	1.71
M15	439.84	7200	439.84	254.56	0.00	0.30
Average		7200		434.56	7.22	1.54
L1	953.29	7200	722.90	1815.98	24.17	3.92
L2	818.31	7200	640.60	1567.76	21.72	4.21
L3	881.86	7200	654.40	1876.61	25.79	3.22
L4	855.05	7200	686.03	1133.65	19.77	3.16
L5	1001.21	7200	771.13	2520.26	22.98	5.50
L6	1005.22	7200	751.63	2184.89	25.23	2.19
L7	1014.32	7200	768.60	1872.36	24.23	1.62
L8	1009.70	7200	725.33	2953.41	28.16	1.41
L9	821.86	7200	615.81	2562.11	25.07	0.44
L10	920.75	7200	678.52	2219.81	26.31	1.36
L11	1314.20	7200	967.84	2556.98	26.36	5.22
L12	1389.67	7200	1069.40	3016.61	23.05	4.02
L13	792.48	7200	601.80	1584.00	24.06	3.01
L14	1021.89	7200	696.08	2326.94	31.88	4.07
L15	1227.21	7200	877.00	2690.09	28.54	5.28
Average		7200		2192.10	25.15	3.24

where the starting time of each task has been determined when generating a solution-seed. The wait time of tourists for a vehicle to pick up them at a location is ignored. Second, the vehicle-type of each route-segment in a solution-seed is recorded. Third, the evaluation model of solution-seeds is modified as Model ILP-Init in Section 5.3, where the component of waiting cost of tourists (ρh) was added to the objective function (32).

Instances L1–L15 are selected to compare the two algorithms. The comparison results of the objective values are shown in Figure 5. The ALNS algorithm provides better solutions than the MH algorithm for all the large-size instances, with the average improvement being

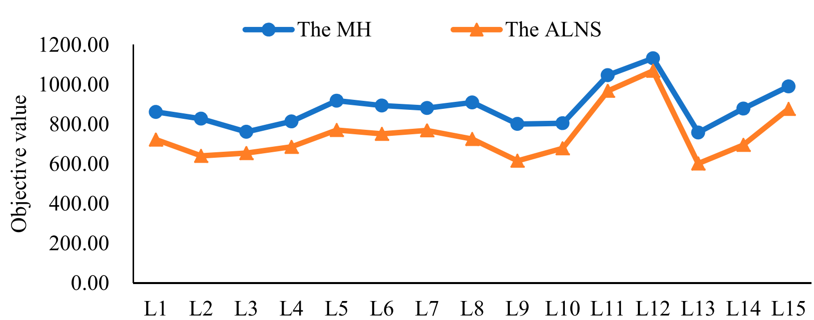

Figure 5. Comparison results between ALNS algorithm and the MH algorithm.

15.88%. The main behind reason is that the decision of both the vehicle type of each route and the waiting time of tourists have been particularly handled in the ALNS algorithm. The calculation time of the MH algorithm is shorter than the ALNS algorithm. However, the quality of solutions of the MH algorithm cannot be improved even given a longer calculation time, because that algorithm was designed based on the branch and construction but not based on the random and iteration. Consequently, the ALNS algorithm provides much better solutions for the HVS-P problem.

6.5. Comparison to the homogeneous vehicle scheduling with precedence constraints

The HVS-P problem is compared to the problem of homogeneous vehicle scheduling with precedence constraints (HoVS-P) to explore the advantage of the usage of heterogeneous vehicles. Instances M1–M15 are used in this experiment. The HoVS-P problem is a special case of the HVS-P problem, which considered only one vehicle type in set K. In consideration of the three types of vehicles of the HVS-P problem in Table 3, the vehicles of the first type are infeasible for solving Instances M1–M15 because of the limitation of the loading capacity. Consequently, the HoVS-P problem using the vehicles of the second type (called the HoVS-P-2), and the third type (called the HoVS-P-3) were verified separately.

Figure 6 shows the objective values of instances solved by the problems of HVS-P and HoVS-P. The average improvements on the objective value are 13.70% and 2.65% when the HVS-P problem is compared to the problems of HoVS-P-2 and HoVS-P-3, respectively. The results indicate that the solutions of Instances M1-M15 involve a greater number of the third type of vehicles compared to the number of the second type of vehicles. The improvements are sensitive to the setting of parameters of the heterogeneous vehicles. We changed the parameters of both the fixed costs and the travel costs of the heterogeneous vehicles, as shown in Table 7. Instances M1-M15 are resolved separately using the parameters noted in Table 7, and Figure 7 shows the objective values. The average improvements in the objective value are 9.51% and 15.56% when the HVS-P problem is compared to the problems of HoVS-P-2 and HoVS-P-3, respectively.

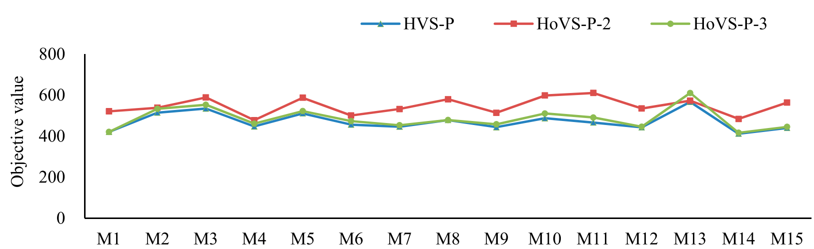

Figure 6. Comparison with the HoVS-P problem.

Table 7. The changed parameters of the heterogeneous vehicles.

Vehicle type k	Q_k	f_k	d_k
1	22	60	1.0
2	35	75	1.5
3	49	90	2.0

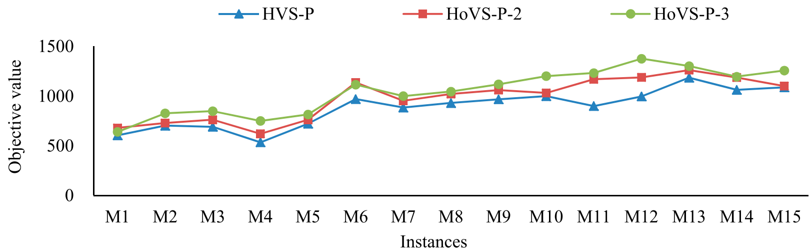

Figure 7. Comparison with the HoVS-P problem under the parameters noted in Table 7.

Consequently, it is always positive to save operational costs for tourism areas when using heterogeneous vehicles, although different settings of the parameters of heterogeneous vehicles are considered. One of the main reasons is that the usage of heterogeneous vehicles provides more scheduling schemes, such as using vehicles with lower fixed costs to serve some smaller-size groups.

6.6. Sensitivity analysis

6.6.1. The cost coefficient ρ

The sensitivity of the cost coefficient of the longest waiting time ρ is analysed using Instances M1–M15. We modified the value of ρ from 2 per minute to 5 per minute, and then to 10 per minute while the other parameters were fixed. Each of the instances was handled by the ALNS algorithm with every value of ρ. Two components of the optimisation objective of each instance, including the longest waiting time and the sum of the fixed cost and the driving cost of all the involved vehicles, are investigated individually.

Specifically, the value of h for each of the instances decreases as the value of ρ increases from 2 per minute to 5 per minute as shown in Figure 8. The value of W_ft for each of the instances increases as the value of ρ increases from 2 per minute to 5 per minute as presented in Figure 9. The value of h continues decreasing, and the value of W_ft continues increasing as ρ increases to 10 per minute for fourteen instances, except for Instance M3. The reason is that more vehicles are required as the value of ρ increases. The driving time of vehicles without loading tourists may be increased to reduce the longest waiting time of tourists. The exceptional result of Instance M3 might be caused by the random feature of the generation of data.

The sensitivity analysis of ρ illustrates that the tourists do not need to wait at any location in the HVS-P problem if the value of ρ is too large. However, the total use cost of

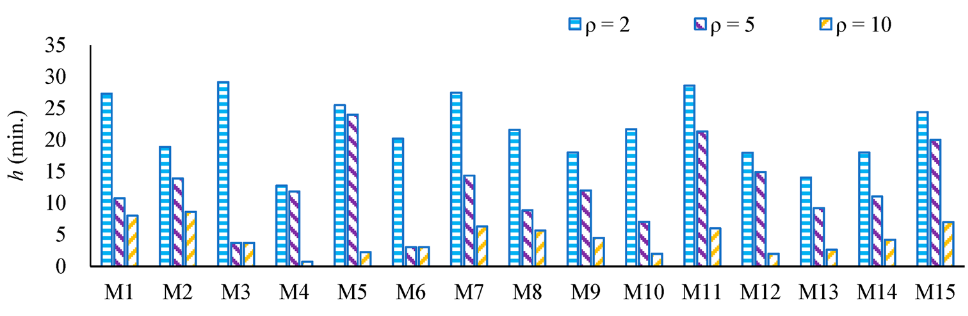

Figure 8. Values of h under different values of ρ.

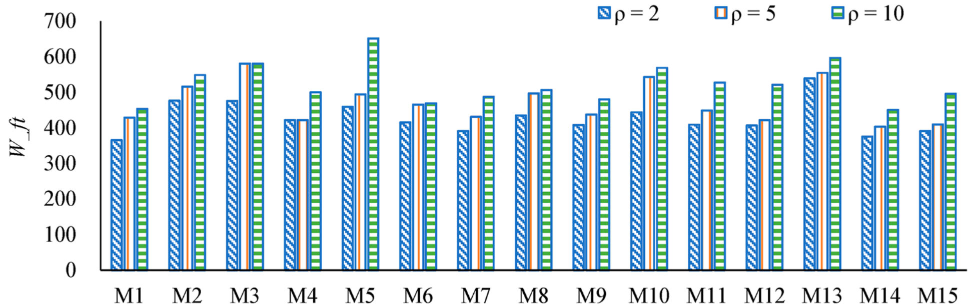

Figure 9. Values of W_ft with different values of ρ.

Table 8. Results of Instance L-D under different sizes of Euclidean plane.

Size of a Euclidean plane (min^2)	Num. of involved vehicles	OBJ.	CPU (s)
20	6	1431.61	2755
40	10	1973.72	3041
60	13	3315.97	3692

vehicles would be increased. A management insight is that the managers of the tourism area can adjust the value of ρ advisably to balance the operational cost of the area and the satisfaction degree of tourists.

6.6.2. The degree of dispersion for scenic spots

The sensitivity of the degree of dispersion for the scenic spots in a tourism area is tested using a large-size instance L-D, which is randomly generated according to Section 6.1.2. In Instance L-D, the number of groups is 25 and the number of tasks is 124. Specifically, the locations of scenic spots and the gate in a tourism area were randomly generated on a Euclidean plane, where the length (width) is modified from 20 min (by vehicle) to 40 min, and then to 60 min. Furthermore, in the three sizes of planes, the direct travelling time between any two locations is modified from 5 min to 15 min, and then to 25 min.

Instance L-D with each setting of the degree of dispersion for locations was independently solved by the ALNS algorithm. The results are shown in Table 8, where all the objective value, the calculation time, and the number of involved vehicles increase as the degree of dispersion for locations increases. The main reason is that the travelling cost of detour or driving without tourist vehicles would increase as the distances between locations increase. Consequently, more vehicles would be used because the increase of the fixed cost of vehicles maybe lower than the increase in the travelling cost of vehicles by detour or driving without tourists. Furthermore, both a scheduling route and the distribution of scenic spots under the Euclidean plane with a length of 60 min are presented in Figure 10.

6.7. A real-size instance study

6.7.1. Data sources

A real-size instance is presented based on the data of Qiandao Lake in China on the day of Spring Festival in 2023. The number of arrival tourists on that day is 1,246. It is nearly a

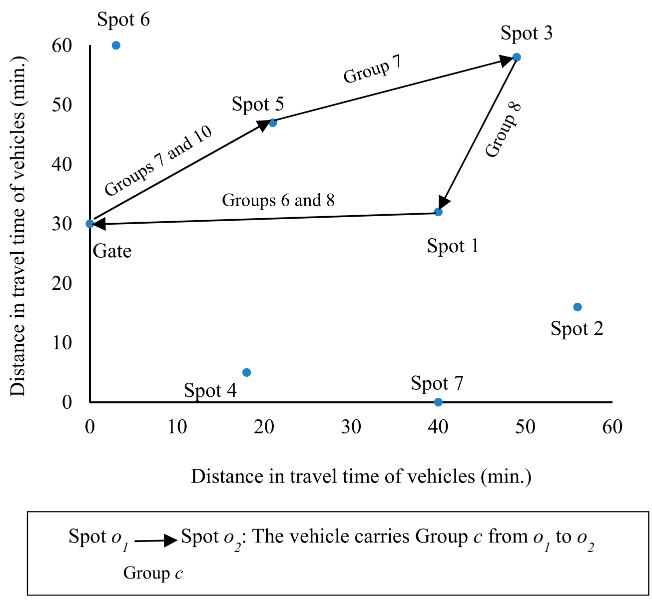

Figure 10. A route in the solution of Instance L-D.

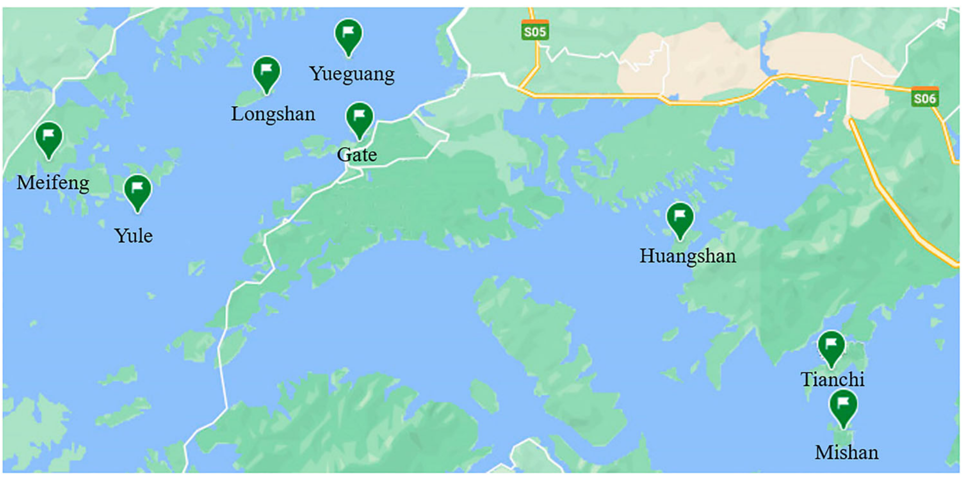

Figure 11. Locations of scenic spots in the Baidu map of Qiandao Lake.

30-fold increase compared with the number of the arrival tourists in Spring Festival in 2022 because of COVID-19.

Two main lakes are opened for tourists in Qiandao Lake, including the Central Lake and the Southeast Lake. The locations of scenic spots in the Baidu map are shown in Figure 11. The probability for tourists to choose each lake to visit is supposed to be the same. The visit information of tourists in each lack is shown in Table 9. Furthermore, the visit duration of tourists at each island is two hours. As shown in Table 10, the travelling time between any two islands in Qiandao Lake is tested based on Baidu Map. The arrival times of tourists are generated referred to Section 6.1.2.

Table 9. The visit information of tourists in the two Lake areas.

The time interval	The selected lakes	The visit scenic spots
8:00-12:00	Central Lake	Miefeng island, Yule island, and Yueguang island
	Southeast Lake	Mishan island, Tianchi island, and Huangshan island
12:00-14:00	Central Lake	Yueguang island and Longshan island
	Southeast Lake	Huangshan island and Tianchi island

Table 10. The travelling times between scenic spots of Qiandao Lake.

Travelling time (min.)	Gate	Meifeng	Yule	Yueguang	Longshan	Huangshan	Tianchi
Meifeng	9.3						
Yule	7.4	2.6					
Yueguang	2.2	9.1	7.7				
Longshan	3.6	6.2	4.9	3.0			
Huangshan	9.3	18.2	16.0	10.9	12.9		
Tianchi	14.1	22.5	20.1	15.9	17.7	5.1	
Mishan	15.9	23.7	21.4	17.6	19.2	6.9	1.8

6.7.2. *The analysis of the operation mode*

The real-size instance is solved under both the flexible operation mode and the charter operation mode. Specifically, the flexible operation mode is the mode researched in this problem, which has been described in the introduction section. The main difference between the two modes is the rules of vehicles for serving tourists. A vehicle in the charter operation mode only serves a tourist group until the group leaves the scenic area.

The objective of both the two problems is Function (7). Specifically, the problem under the charter mode is simple, and tourists have no waiting time. The CPLEX solver can provide the exact solution even for a real-size instance under the charter mode. The ALNS algorithm is used to solve the real-size instance under the flexible mode because CPLEX cannot provide a feasible solution for the real-size instance even in two hours. Furthermore, Sections 6.3 and 6.4 have verified the superiority of the ALNS algorithm for the problem under the flexible mode.

As shown in Table 11, the ALNS algorithm is able to solve the real-size instance within one hour. Consequently, the ALNS algorithm is effective for the HVS-P problem considering real-life scenarios, although it involves some integer linear programming models. One main reason is that the embedded models are simplified and are only used to solve some subproblems of the HVS-P problem. Consequently, the performance of the embedded models is acceptable even for a real-size instance of the HVS-P problem.

Furthermore, the comparative results on the real-size instance between the flexible operation mode and the charter operation mode are shown in Table 11. The flexible operation mode is more effective in saving operational costs compared with the charter mode. One main reason is that the flexible mode can improve the utilisation of vehicles and save the fixed cost of using vehicles. Vehicles in this mode would detour or travel some routes

Table 11. The solutions of the real-size instance under the flexible mode and charter mode.

Operation mode	OBJ.	CPU (s)	Fixed cost	Travel cost	Wait cost
Flexible mode	1796.27	2842	448.00	1291.20	57.07
Charter mode	2074.08	54.30	960.00	1114.08	0.00

Table 12. Parameters of the fixed cost of a vehicle with different types.

Vehicle type k	f_k^1	f_k^2	f_k^3
1	30	60	90
2	32	62	92
3	34	64	94

Table 13. The solutions of the real-size instance with different fixed costs.

Fixed cost of a vehicle	Number of involved vehicles	Total travel time of vehicles
f_k^1	15	813.1
f_k^2	7	1076.0
f_k^3	6	1128.5

without tourists to save the fixed cost although the travel cost and wait cost increase slightly.

6.7.3. The analysis of the cost of vehicles

The relationship between the fixed cost and the travel cost of using vehicles in the real-size instance is particularly analysed because the two components account for a large proportion of the total cost according to the results in Table 11. The fixed cost of a vehicle with different types is modified from f_k^1 to f_k^2 and then to f_k^3 while the other parameters are fixed (recall Table 3). The values of f_k^1, f_k^2 and f_k^3 for different vehicle types are shown in Table 12.

The real-size instance of the HVS-P problem described in Section 3 is handled by the ALNS algorithm with every setting of f_k. The number of involved vehicles decreases and the total travel time of vehicles increases as the fixed cost of a vehicle increases from f_k^1 to f_k^2 and then to f_k^3 as shown in Table 13. The main reason for this is that the operation efficiency of vehicles would be improved to save the vehicles when the fixed cost of a vehicle increases. However, the total travel time would increase at the same time, because the travel of vehicles without tourists would increase. Consequently, scenic areas could adjust the allocated number and travel time of vehicles by modifying the parameters of costs.

7. Conclusions and further research

A vehicle scheduling problem for tourists in a tourism area is researched, in which the visit requests with precedence constraints are served by heterogeneous vehicles. The objective function of the problem minimises the total cost including the fixed cost and the driving cost of all involved vehicles, as well as the waiting cost of tourists. A mixed-integer linear programming model is presented to formulate the problem. Meanwhile, an adaptive large neighbourhood search algorithm with four specialised operators is designed to solve the problem efficiently. Both the model and the algorithm are evaluated based on a number of random instances. Furthermore, a real-size instance is studied and analysed.

The experiments imply that both the mathematical model and the algorithm are effective for the problem. The algorithm is stable and it can provide better solutions within a shorter time compared to the mathematical model for large-size instances. It is possible to save operational cost for tourism areas when using heterogeneous vehicles with different types. Managers could balance the operational cost of the tourism area and the satisfaction degree of tourists by adjusting the cost coefficient of waiting time advisably. In addition, the number of involved vehicles would increase as the degree of dispersion of the locations in a tourism area increases.

The main limitation of this research is that the information of tourists is assumed to be given in advance. Actually, many elements of tourists should be dynamic or uncertain. Another limitation of this research is that the balance of task assignment has not been considered among the vehicle routes. Fair task assignment among vehicle routes would be significant for the drivers, in practice.

Several more elements could be considered in the future. For example, the break time of drivers could be introduced to the problem, because the scheduling of human resources is essential. Furthermore, tourists could be divided into different grades according to the prices of scenic tickets. Tourists with different grades might require different types of transportation services. Some other problems regarding the satisfaction degree of tourists, for instance, the clustering problem of tourists, would be considered simultaneously. The forecast of the information of tourists, including the number of tourists and the arrival times of tourists in one day, could also be considered in the future. The machine learning method might be used for the forecast of tourists' demands.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China under Grants 71971050 and 71831006. This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT) [Grant No. RS-2023-00218913].

ORCID

Ilkyeong Moon http://orcid.org/0000-0002-7072-1351

References

Ali, Ousmane, Jean-François Côté, and Leandro Callegari Coelho. 2021. "Models and Algorithms for the Delivery and Installation Routing Problem." *European Journal of Operational Research* 291 (1): 162–177. https://doi.org/10.1016/j.ejor.2020.09.011.

Bai, Xiaoshan, Ming Cao, Weisheng Yan, and Shuzhi Sam Ge. 2020. "Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles." *IEEE Transactions on Automation Science and Engineering* 17 (1): 248–260. https://doi.org/10.1109/TASE.2019.2914113.

Baykasoglu, Adil, Kemal Subulan, Ali Serdar Tasan, and Nurhan Dudakli. 2019. "A Review of Fleet Planning Problems in Single and Multimodal Transportation Systems." *Transportmetrica A: Transport Science* 15 (2): 631–697. https://doi.org/10.1080/23249935.2018.1523249.

Berbeglia, Gerardo, Jean-Francois Cordeau, Irina Gribkovskaia, and Gilbert Laporte. 2007. "Static Pickup and Delivery Problems: A Classification Scheme and Survey." *Top* 15 (1): 1–31. https://doi.org/10.1007/s11750-007-0009-0.

Cappanera, Paola, Cristina Requejo, and Maria Grazia Scutellà. 2020. "Temporal Constraints and Device Management for the Skill VRP: Mathematical Model and Lower Bounding Techniques." *Computers & Operations Research* 124: 105054. https://doi.org/10.1016/j.cor.2020.105054.

Chaowasakoo, Patarawan, Heikki Seppälä, Heikki Koivo, and Quan Zhou. 2017. "Improving Fleet Management in Mines: The Benefit of Heterogeneous Match Factor." *European Journal of Operational Research* 261 (3): 1052–1065. https://doi.org/10.1016/j.ejor.2017.02.039.

Chargui, Kaoutar, Tarik Zouadi, V. Raja Sreedharan, Abdellah El Fallahi, and Mohamed Reghioui. 2023. "A Novel Proactive-Reactive Scheduling Approach for the Quay Crane Scheduling Problem: A VUCA Perspective." *IEEE Transactions on Engineering Management* 70 (7): 2594–2607. https://doi.org/10.1109/tem.2022.3151842.

Chen, Jingxu, Shuaian Wang, Zhiyuan Liu, and Wei Wang. 2017. "Design of Suburban Bus Route for Airport Access." *Transportmetrica A: Transport Science* 13 (6): 568–589. https://doi.org/10.1080/23249935.2017.1306896.

Cinar, Ahmet, Fatma Sibel Salman, Ozgur Merih Araz, and Mert Parcaoglu. in press. "Managing Home Health-Care Services with Dynamic Arrivals During a Public Health Emergency." *IEEE Transactions on Engineering Management*. https://doi.org/10.1109/tem.2022.3209962.

Cordeau, Jean-François, and Gilbert Laporte. 2007. "The Dial-a-Ride Problem: Models and Algorithms." *Annals of Operations Research* 153 (1): 29–46. https://doi.org/10.1007/s10479-007-0170-8.

Dohn, Anders, Matias Sevel Rasmussen, and Jesper Larsen. 2011. "The Vehicle Routing Problem with Time Windows and Temporal Dependencies." *Networks* 58 (4): 273–289. https://doi.org/10.1002/net.20472.

Dou, Xueping, Qiang Meng, and Kai Liu. 2021. "Customized Bus Service Design for Uncertain Commuting Travel Demand." *Transportmetrica A: Transport Science* 17 (4): 1405–1430. https://doi.org/10.1080/23249935.2020.1864509.

Duan, Leyi, Yuguang Wei, Jinchuan Zhang, and Yang Xia. 2020. "Centralized and Decentralized Autonomous Dispatching Strategy for Dynamic Autonomous Taxi Operation in Hybrid Request Mode." *Transportation Research Part C: Emerging Technologies* 111: 397–420. https://doi.org/10.1016/j.trc.2019.12.020.

Esztergar-Kiss, Domokos, Gudio Gentile, and Agostino Nuzzolo. 2020. "Special Issue on Dynamic Models Serving Real-Time Urban Transport Operations." *Transportmetrica A: Transport Science* 16 (2): 161–163. https://doi.org/10.1080/23249935.2019.1692964.

Guo, Rongge, Wei Guan, Wenyi Zhang, Fanting Meng, and Zixian Zhang. 2019. "Customized Bus Routing Problem with Time Window Restrictions: Model and Case Study." *Transportmetrica A: Transport Science* 15 (2): 1804–1824. https://doi.org/10.1080/23249935.2019.1644566.

Guo, Haonan, Yun Wang, Pan Shang, Xuedong Yan, and Yunlin Guan. 2024. "Customised Bus Route Design with Passenger-to-Station Assignment Optimisation." *Transportmetrica A: Transport Science* 20 (3): 2214631. https://doi.org/10.1080/23249935.2023.2214631.

Hà, Minh Hoàng, Tat Dat Nguyen, Thinh Nguyen Duy, Hoang Giang Pham, Thuy Do, and Louis-Martin Rousseau. 2020. "A New Constraint Programming Model and a Linear Programming-Based Adaptive Large Neighborhood Search for the Vehicle Routing Problem with Synchronization Constraints." *Computers & Operations Research* 124: 105085. https://doi.org/10.1016/j.cor.2020.105085.

Hanafi, Saïd, Renata Mansini, and Roberto Zanotti. 2020. "The Multi-Visit Team Orienteering Problem with Precedence Constraints." *European Journal of Operational Research* 282 (2): 515–529. https://doi.org/10.1016/j.ejor.2019.09.045.

Huang, Di, Yiran Wang, Shuai Jia, Zhiyuan Liu, and Shuaian Wang. 2023. "A Lagrangian Relaxation Approach for the Electric Bus Charging Scheduling Optimisation Problem." *Transportmetrica A: Transport Science* 19 (2): 2023690. https://doi.org/10.1080/23249935.2021.2023690.

Huang, Chao, and Ruiyou Zhang. 2023. "Container Drayage Transportation Scheduling with Foldable and Standard Containers." *IEEE Transactions on Engineering Management* 70 (10): 3497–3511. https://doi.org/10.1109/tem.2021.3094994.

Huerta-Muñoz, Diana L., Claudia Archetti, Elena Fernández, and Federico Perea. 2022. "The Heterogeneous Flexible Periodic Vehicle Routing Problem: Mathematical Formulations and Solution Algorithms." *Computers & Operations Research* 141: 105662. https://doi.org/10.1016/j.cor.2021.105662.

Ji, Bin, Xiaohui Yuan, Yanbin Yuan, Xiaohui Lei, and Herbert Ho Ching Lu. 2020. "An Adaptive Large Neighborhood Search for Solving Generalized Lock Scheduling Problem: Comparative Study with Exact Methods." *IEEE Transactions on Intelligent Transportation Systems* 21 (8): 3344–3356. https://doi.org/10.1109/TITS.2019.2926405.

Koç, Çağrı, Tolga Bektaş, Ola Jabali, and Gilbert Laporte. 2016. "Thirty Years of Heterogeneous Vehicle Routing." *European Journal of Operational Research* 249 (1): 1–21. https://doi.org/10.1016/j.ejor.2015.07.020.

Lai, Yongxuan, Fan Yang, Ge Meng, and Wei Lu. 2022. "Data-Driven Flexible Vehicle Scheduling and Route Optimization." *IEEE Transactions on Intelligent Transportation Systems* 23 (12): 23099–23113. https://doi.org/10.1109/TITS.2022.3204644.

Li, Jiliu, Hu Qin, Roberto Baldacci, and Wenbin Zhu. 2020. "Branch-and-price-and-cut for the Synchronized Vehicle Routing Problem with Split Delivery, Proportional Service Time and Multiple Time Windows." *Transportation Research Part E: Logistics and Transportation Review* 140: 101955. https://doi.org/10.1016/j.tre.2020.101955.

Lianes, Ingeborg Margrete, Maren Theisen Noreng, Kjetil Fagerholt, Hans Tobias Slette, and Frank Meisel. 2021. "The Aquaculture Service Vessel Routing Problem with Time Dependent Travel Times and Synchronization Constraints." *Computers & Operations Research* 134: 105316. https://doi.org/10.1016/j.cor.2021.105316.

Lin, Peiqun, Chenxing He, Lingshu Zhong, Mingyang Pei, Chuhao Zhou, and Yang Liu. 2023. "Bus Timetable Optimization Model in Response to the Diverse and Uncertain Requirements of Passengers for Travel Comfort." *Electronic Research Archive* 31 (4): 2315–2336. https://doi.org/10.3934/era.2023118.

Liu, Zhujun, Ilkyeong Moon, and Ruiyou Zhang. 2024. "Flexible Vehicle Scheduling with Precedence Constraints for Tourists." *International Transactions in Operational Research* 31 (4): 2309–2337. https://doi.org/10.1111/itor.13256.

Máximo, Vinícius R., Jean-François Cordeau, and Mariá Cristina Vasconcelos Nascimento. 2022. "An Adaptive Iterated Local Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem." *Computers & Operations Research* 148: 105954. https://doi.org/10.1016/j.cor.2022.105954.

Meliani, Youssef, Yasmina Hani, Sâad Lissane Elhaq, and Abderrahman El Mhamedi. 2022. "A Tabu Search Based Approach for the Heterogeneous Fleet Vehicle Routing Problem with Three-Dimensional Loading Constraints." *Applied Soft Computing* 126: 109239. https://doi.org/10.1016/j.asoc.2022.109239.

Melis, Lissa, and Kenneth Sörensen. 2021. "The Static on-Demand Bus Routing Problem: Large Neighborhood Search for a Dial-a-Ride Problem with Bus Station Assignment." *International Transactions in Operational Research* 29 (3): 1417–1453. https://doi.org/10.1111/itor.13058.

Messaoud, Elhassania. 2022. "Solving a Stochastic Programming with Recourse Model for the Stochastic Electric Capacitated Vehicle Routing Problem Using a Hybrid Genetic Algorithm." *European Journal of Industrial Engineering* 16 (1): 71–90. https://doi.org/10.1504/EJIE.2022.119367.

Mo, Daniel Y., Yue Wang, Y. C. E. Lee, and Mitchell M. Tseng. 2020. "Mass Customizing Paratransit Services with a Ridesharing Option." *IEEE Transactions on Engineering Management* 67 (1): 234–245. https://doi.org/10.1109/TEM.2018.2873832.

Molina, Jose Carlos, Jose Luis Salmeron, Ignacio Eguia, and Jesus Racero. 2020. "The Heterogeneous Vehicle Routing Problem with Time Windows and a Limited Number of Resources." *Engineering Applications of Artificial Intelligence* 94: 103745. https://doi.org/10.1016/j.engappai.2020.103745.

Moon, Ilkyeong, Said Salhi, and Xuehao Feng. 2020. "The Location-Routing Problem with Multi-Compartment and Multi-Trip: Formulation and Heuristic Approaches." *Transportmetrica A: Transport Science* 16 (3): 501–528. https://doi.org/10.1080/23249935.2020.1720036.

Rasmussen, Matias Sevel, Tor Justesen, Anders Dohn, and Jesper Larsen. 2012. "The Home Care Crew Scheduling Problem: Preference-Based Visit Clustering and Temporal Dependencies." *European Journal of Operational Research* 219 (3): 598–610. https://doi.org/10.1016/j.ejor.2011.10.048.

Rattanamanee, Tarit, and Suebsak Nanthavanij. 2022. "Heuristic Procedure for Bi-Capacitated Multiple-Trip Vehicle Routing Problem." *European Journal of Industrial Engineering* 16 (3): 294–316. https://doi.org/10.1504/EJIE.2022.121904.

Ren, Jingxuan, Wenzhou Jin, and Weitiao Wu. 2023. "Multi-Objective Optimization for Multi-Depot Heterogeneous First-Mile Transportation System Considering Requests' Preference Ranks for Pick-Up Stops." *Transportmetrica A: Transport Science* 19 (3): 2103205. https://doi.org/10.1080/23249935.2022.2103205.

Roohnavazfar, Mina, and Seyed Hamid Reza Pasandideh. 2022. "Decomposition Algorithm for the Multi-Trip Single Vehicle Routing Problem with AND-Type Precedence Constraints." *Operational Research* 22: 4253–4285. https://doi.org/10.1007/s12351-021-00663-0.

Sarasola, Briseida, and Karl F. Doerner. 2020. "Adaptive Large Neighborhood Search for the Vehicle Routing Problem with Synchronization Constraints at the Delivery Location." *Networks* 75 (1): 64–85. https://doi.org/10.1002/net.21905.

Shang, Huayan, Yanping Liu, Wenxiang Wu, and Fangxia Zhao. 2023. "Multi-Depot Vehicle Scheduling with Multiple Vehicle Types on Overlapped bus Routes." *Expert Systems with Applications* 228: 120352. https://doi.org/10.1016/j.eswa.2023.120352.

Sun, Wei, Yang Yu, and Junwei Wang. 2019. "Heterogeneous Vehicle Pickup and Delivery Problems: Formulation and Exact Solution." *Transportation Research Part E: Logistics and Transportation Review* 125: 181–202. https://doi.org/10.1016/j.tre.2019.03.012.

Tayachi, Dalila, and Cheima Jendoubi. 2023. "Optimising Green Vehicle Routing Problem-a Real Case Study." *European Journal of Industrial Engineering* 17 (4): 570–596. https://doi.org/10.1504/EJIE.2023.131732.

Tian, Qingyun, Yunhui Lin, and David Zhi-Wei Wang. 2023. "Joint Scheduling and Formation Design for Modular-Vehicle Transit Service with Time-Dependent Demand." *Transportation Research Part C: Emerging Technologies* 147: 103986. https://doi.org/10.1016/j.trc.2022.103986.

Wang, Decheng, Ilkyeong Moon, and Ruiyou Zhang. 2022. "Multi-Trip Multi-Trailer Drop-and-Pull Container Drayage Problem." *IEEE Transactions on Intelligent Transportation Systems* 23 (10): 19088–19104. https://doi.org/10.1109/tits.2022.3156547.

Wang, Yong, Shouguo Peng, Xuesong Zhou, Monirehalsadat Mahmoudi, and Lu Zhen. 2020a. "Green Logistics Location-Routing Problem with Eco-Packages." *Transportation Research Part E: Logistics and Transportation Review* 143: 102118. https://doi.org/10.1016/j.tre.2020.102118.

Wang, Yong, Yingying Yuan, Xiangyang Guan, Maozeng Xu, Li Wang, Haizhong Wang, and Yong Liu. 2020b. "Collaborative Two-Echelon Multicenter Vehicle Routing Optimization Based on State-Space-Time Network Representation." *Journal of Cleaner Production* 258: 120590. https://doi.org/10.1016/j.jclepro.2020.120590.

Xue, Shouqiang, Rui Song, Shiwei He, Jiuyu An, Youmiao Wang, and Yajie Zou. 2022. "An Improved Adaptive Large Neighborhood Search Algorithm for the Heterogeneous Customized bus Service with Multiple Pickup and Delivery Candidate Locations." *Journal of Advanced Transportation* 2022: 1679469. https://doi.org/10.1155/2022/1679469.

Yao, Enjian, Tong Liu, Tianwei Lu, and Yang Yang. 2020. "Optimization of Electric Vehicle Scheduling with Multiple Vehicle Types in Public Transport." *Sustainable Cities and Society* 52: 101862. https://doi.org/10.1016/j.scs.2019.101862.

Yu, Vincent F., Panca Jodiawan, and Aldy Gunawan. 2021. "An Adaptive Large Neighborhood Search for the Green Mixed Fleet Vehicle Routing Problem with Realistic Energy Consumption and Partial Recharges." *Applied Soft Computing* 105: 107251. https://doi.org/10.1016/j.asoc.2021.107251.

Zhang, Ruiyou, Jingmei Guo, and Junwei Wang. 2022a. "A Time-Dependent Electric Vehicle Routing Problem with Congestion Tolls." *IEEE Transactions on Engineering Management* 69 (4): 861–873. https://doi.org/10.1109/TEM.2019.2959701.

Zhang, Aijia, Tiezhu Li, Yue Zheng, Xuefeng Li, Muhammad Ghazanfar Abdullah, and Changyin Dong. 2022b. "Mixed Electric Bus Fleet Scheduling Problem with Partial Mixed-Route and Partial Recharging." *International Journal of Sustainable Transportation* 16 (1): 73–83. https://doi.org/10.1080/15568318.2021.1914791.

Zhang, Ruiyou, Zhujun Liu, and Xuehao Feng. 2021. "A Novel Flexible Shuttle Vehicle Scheduling Problem in Scenic Areas: Task-Divided Graph-Based Formulation and ALGORITHM." *Computers & Industrial Engineering* 156: 107295. https://doi.org/10.1016/j.cie.2021.107295.

Zhang, Ruiyou, Decheng Wang, and Junwei Wang. 2021. "Multi-Trailer Drop-and-Pull Container Drayage Problem." *IEEE Transactions on Intelligent Transportation Systems* 22 (9): 5708–5720. https://doi.org/10.1109/TITS.2020.2991098.