A multi-stage heuristic algorithm based on task grouping for vehicle routing problem with energy constraint in disasters

Names of authors:
Lei Jiao[a,b], Zhihong Peng[a,b], Lele Xi[a,b], Miao Guo[a,b], Shuxin Ding[c], Yue Wei[d]

Affiliations and addresses of authors:
[a] School of Automation, Beijing Institute of Technology, Beijing 100081, China
[b] Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China
[c] Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China
[d] Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen 518055, China

Email addresses of authors:
jiaolei_1029@163.com
peng@bit.edu.cn
freexile@163.com
guomiao618@163.com
dingshuxin@rails.cn
weiy@pcl.ac.cn

Corresponding Author:
Zhihong Peng
School of Automation, Beijing Institute of Technology, Beijing 100081, China.
Tel: +86-13910437979
Email: peng@bit.edu.cn

Abstract

In recent years, serious disasters have happened frequently, causing significant loss of lives and substantial economic consequences. Considering that rescue vehicles are mainly powered by electricity, they cannot compete with the conventional ones in terms of the cruising range. Therefore, it is of great significance to study how rescue vehicles can execute multiple rescue tasks quickly and efficiently with energy constraint. In this paper, a multi-stage vehicle routing algorithm based on task grouping (MSVR-TG) is proposed for the rescue vehicle routing problem with energy constraint (VRPEC) in disasters. In the task grouping stage, a novel K-means algorithm based on angular density (K-means-ad) is suggested to address the instability of the result of K-means caused by the random selection of initial cluster centers. It can balance the travel distance of each vehicle as much as possible, which is critical when vehicles are energy constrained. In the sequence planning stage, a problem-specific genetic algorithm (PSGA) with multiple population initialization rules and the route improvement strategy is introduced to balance the convergence speed and population diversity, achieving a promising planning result. In the route adjusting stage, various removal and insertion heuristics in the large neighborhood search (LNS) are designed to adjust the routes, thereby further improving the planning result. The routes participating in adjustment are obtained via the developed route selection rule. The validity of each stage in MSVR-TG has been verified in a series of scenarios. A comparison about the proposed algorithm against several advanced algorithms is constructed, and results reveal that MSVR-TG is superior over compared algorithms for VRPEC.

Keywords: emergency rescue, energy constraint, vehicle routing problem, task grouping

1. Introduction

The occurrence of natural and man-made disasters usually leads to significant social and economic disruption, as well as high numbers of casualties (Wan et al., 2021; Sun et al., 2021). Upon the occurrence of a disaster, emergency rescue should be carried out at disaster areas with different severities in

the shortest time, so as to effectively reduce post-disaster losses and improve the survival probability of victims (Liu et al., 2020; Yu et al., 2020; Chen et al., 2022). Emergency rescue operations mainly involve research on roadway repair problem, routing problem, facility location problem, etc.(Zheng et al., 2015). We focus on the rescue vehicle routing problem in disasters, which has the following characteristics (Jiao et al., 2022; Song et al., 2022; Zhang et al., 2022a):

- **Tasks with multiple attributes are heterogeneous in terms of urgency, importance and execution time, etc.** Taking the nuclear leakage scenario as an example, office buildings and fire points are the areas where search and rescue tasks should be carried out urgently. What's more, compared with open environments, the execution time for casualty searching in complex indoor environments is longer.
- **Rescue vehicles are electrically driven with the limited max-duration.** Rescue vehicles should return to the transportation platforms for energy supply or to be sent to other areas before energy depletion.
- **The situation is complex and changeable in disasters, so the requirement for information validity is high.** Especially in nuclear leakage scenarios, wireless communication equipment may fail due to the nuclear radiation, resulting in rescue vehicles need to return to the transportation platform within a specified time for information transmission.

Vehicle routing problems (VRPs) in emergency logistics are characterized as scheduling limited vehicles to accomplish various tasks in rigidly limited times and under harsh environment conditions. In recent years, many scholars have carried out research on rescue vehicle routing problem in disasters. Wan et al. (2021) proposed a hybrid ant colony optimization algorithm (HACO) to solve multi-objective multi-constraint emergency material scheduling problem. Sun et al. (2021) developed a robust optimization model for combined facility location and casualty transportation under uncertainty in the number of casualties. Xu et al. (2021) presented an efficient rescue route planning scheme operating within a high performance emergent rescue management system for vehicles based on the mobile cloud computing

paradigm. However, above studies focus on the rescue vehicle routing problem with sufficient resources, and rarely consider the opposite situation. In practical applications, the number of electric-driven rescue vehicles is limited and the max-durations of them exist, so the number and scale of tasks they can execute during the duration are limited.

In view of the limited energy of rescue vehicles, route planning of rescue vehicles in disasters can be modeled as the vehicle routing problem under the energy constraint (VRPEC), where transportation platforms provide assistance to a certain number of tasks with different types of rescue demand through rescue vehicles. There are two main characteristics of VRPEC, namely, the multiple attributes of tasks and the limited energy of the rescue vehicles, which lead to the following two problems to be solved.

(1) Trade-off among tasks with multiple attributes.

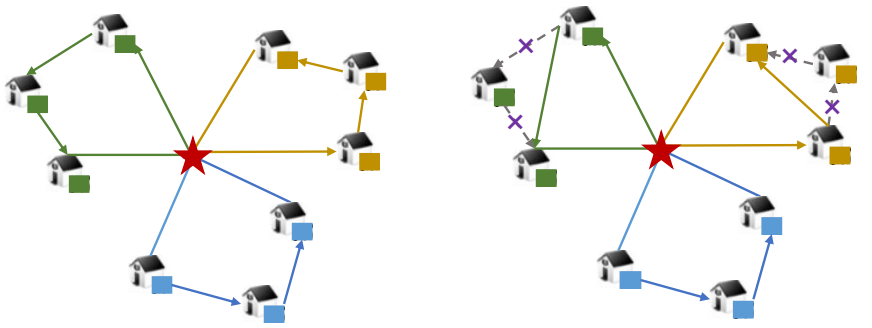
As can be seen in Table 1, VRPEC introduces the task importance and the execution time into the capability vehicle routing problem (CVRP), and expresses capability constraint as the max-duration, resulting in the situation where tasks are not fully executed within a single duration. Fig. 1 shows the route planning results in both CVRP and VRPRC. The crosses in Fig. 1(b) represent the route segments deleted due to the energy constraint, and the related tasks will be executed during the following durations.

Table 1: Comparison between CVRP and VRPEC.

Problem	**Objectives**	**Capability constraint**	**Importance of task(customer)**	**Execution time of task(customer)**	**Trade-off among tasks(customers)**
CVRP	Cost, time	Load			
VRPEC	The total weight of the tasks executed	Energy	✓	✓	✓

As a variant of VRP (Dantzig & Ramser, 1959), CVRP is NP-hard (Xiao et al., 2021), so the VRPEC is NP-hard as well. It is difficult or even impossible to obtain the optimal solution to the above problems in an acceptable time. Therefore, in recent years, scholars have designed a variety of metaheuristics with excellent performance for CVRP and its variants, such as (Jia et al., 2021; Altabeeb et al., 2021; Fg et al., 2022).

Existing CVRPs consider the cases of sufficient resources, that is, nodes can be fully served. As mentioned above, in emergency rescue situations, the number and the energy of rescue vehicles are limited, so the selected tasks



(a) Planning result without energy constraint in CVRP

(b) Planning result with energy constraint in VRPEC

Figure 1: Planning results with and without energy constraint.

and the execution routes need to be reasonably optimized by combining the attributes of tasks. Given that tasks have multiple attributes, the reasonable choice of the tasks to be executed is a complex problem.

(2) Trade-off among energy consumptions of vehicles with limited energy.

Electric vehicles (EVs) are promising to reduce both transportation costs and pollution effects in comparison to fossil-fuel-based engines. However, the cruising range of electric vehicles is not comparable to that of the fossil-fuel-based ones. Researches on the routing problem of energy-constrained vehicles has received increasing attention, leading to a new variant of VRP, EVRP (Kucukoglu et al., 2021). Sai et al. (2018) implemented a hybrid genetic algorithm (HGA), which is a combination of GA and a local search strategy to minimize the sum of the vehicle fixed cost, travel cost, and charging cost. Jia et al. (2021) proposed a novel bilevel ant colony optimization algorithm (BACO) to optimize both the service order of customers and the recharging schedules of EVs.

Compared with EVRP, VRPEC in emergency rescue scenarios has the following characteristics: (i) Tasks have more attributes, and the requirements of them are expressed in terms of execution time, importance, etc. Under the condition of energy constraint, tasks with higher urgency and importance need to be preferentially executed. (ii) Since there are no recharging stations except for transportation platforms during task execution, rescue vehicles with limited energy need to reasonably select the tasks to be executed and

optimize the execution sequence within the duration. (iii) Energy consumption is utilized as a constraint rather than the optimization goal. Therefore, under the condition of insufficient resources, it is necessary to balance the energy consumption of each vehicle as much as possible to improve the rescue efficiency.

To sum up, existing algorithms have limitations in solving VRPEC, due to the fact that the case of insufficient resources and tasks with multiple attributes are rarely considered. In VRPEC, tasks to be executed need to be reasonably selected, and the energy of each vehicle needs to be fully utilized. For this reason, our motivation is to design an effective heuristic algorithm according to the characteristics of VRPEC, which can achieve promising solutions in bounded time.

In emergency rescue situations, tasks are assigned weights to indicate the importance, and ones with higher values represent more important (Li et al., 2020; Wan et al., 2021). Accordingly, we define the objective as maximizing the total weight of the executed tasks within the duration of each vehicle. To solve VRPEC efficiently, a multi-stage vehicle routing algorithm based on task grouping (MSVR-TG) is proposed. The innovations and contributions of this paper are listed below.

1. A novel multi-stage heuristic algorithm structure is proposed, including the task grouping stage, the sequence planning stage and the route adjusting stage, which can effectively solve VRPEC.

    - In the task grouping stage, according to the distribution of the tasks and the depot, an improved K-means based on angular density (K-means-ad) is suggested to address the instability of the result of K-means caused by the random selection of initial cluster centers. Due to the introduction of problem characteristics, K-means-ad achieves high-quality grouping result which balances the travel distance of each vehicle as much as possible.
    - In the sequence planning stage, a variety of population initialization rules and the route improvement strategy are designed in the problem-specific genetic algorithm (PSGA), which can effectively balance the convergence and diversity of the population, so as to obtain a promising result.
    - In the route adjusting stage, the large neighborhood search (LNS) with various removal and insertion heuristics is introduced to ad-

just the sequence of the tasks in the selected routes, which can effectively improve the total weight of the executed tasks. Combined with the total weight and energy consumption of each route, a route selection rule is developed to improve the directionality of the neighborhood search.

2. The validity of each stage in MSVR-TG has been verified in a series of scenarios. Simulations demonstrate that MSVR-TG is superior over state-of-art algorithms for VRPEC and has an advantage in the running time.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive related literature review. Section 3 introduces the problem definition and the mathematical model for VRPEC. Section 4 elaborates MSVR-TG in detail. In Section 5, MSVR-TG is compared with state-of-art algorithms in a variety of test cases, and the results are analyzed. Section 6 makes concluding remarks and future works.

2. Literature review

Throughout the research history of VRP, it is easy to find that sufficient research on the fundamental CVRP can benefit the other variants a lot. Following the same path, CVRP is studied in this section as the basic VRPEC model to contribute to the research of VRPEC.

In recent years, CVRP has attracted extensive attention. Since it is NP-hard, scholars have proposed a variety of heuristic algorithms with excellent performance for it and its variants, as shown in Table 2. Algorithms involved consist of global-search-based heuristics and divide-and-conquer-based heuristics according to whether the original problem is decomposed. The findings obtained from the literatures are presented.

(1) Global-search-based heuristic algorithms.

As shown in Table 2, many excellent optimization algorithms and their combinations are used to address CVRP and its variants, such as genetic optimization (Shahab et al., 2016; Sai et al., 2018; Karakatič, 2021), ant colony optimization (Jia et al., 2021, 2022), particle swarm optimization (Islam et al., 2021), neighborhood search based algorithms (Akpinar, 2016; Xu & Cai, 2018; Lu et al., 2020), firefly algorithms (Altabeeb et al., 2019,

Table 2: Comparison among literatures.

Literature	Energy constraint	Insufficient resources[1]	Decomposition method	Algorithm
(Fang et al., 2013)			Customer-based	Voronoi spatial neighborhood-based search heuristic
(Mei et al., 2014)			Route-based	RDG-MAENS[2]
(Akpinar, 2016)			–[11]	LNS–ACO[3]
(Praveen et al., 2016)			Customer-based	Enhanced K-means clustering algorithm
(Shahab et al., 2016)			Customer-based	Two-step genetic algorithm
(Zhang, 2017)			Customer-based	Novel two-phase heuristic[4]
(Sai et al., 2018)	✓		–	Hybrid genetic algorithm[5]
(Kancharla & Ramadurai, 2018)	✓		–	Adaptive large neighborhood search with special operators
(Chen et al., 2019)			–	Highest effective time ratio first algorithm
(Altabeeb et al., 2019)			–	CVRP-FA[6]
(Lu et al., 2020)	✓		–	Iterated variable neighborhood search algorithm
(Li et al., 2020)	✓	✓	–	Weighted targets sweep coverage algorithm
(Xiao et al., 2021)			Route-based	EMRG-HA[7]
(Xu et al., 2021)			Task-based	(OT-K-means++) and GSOCI[8]
(Jia et al., 2021)	✓		–	Bilevel ant colony optimization
(Islam et al., 2021)			–	Hybrid particle swarm optimization
(Lan et al., 2022)	✓		–	Region-focused memetic algorithm with heuristic-assisted solution initialisation
(Altabeeb et al., 2021)			–	Cooperative hybrid firefly algorithm
(Zhang et al., 2022b)			Route-based	Route clustering and search heuristic
(Jia et al., 2022)	✓		–	Confidence-based bi-level ant colony optimization algorithm
(Shang et al., 2022)			Customer-based	Memetic search with evolutionary multitasking
(Liu et al., 2022)	✓		–	Hybrid genetic algorithm[9]
(Öztaş & Tuş, 2022)			–	ILS-RVND-TA[10]
This study	✓	✓	Task-based	Multi-stage vehicle routing algorithm based on task grouping (MSVR-TG)

[1] Insufficient resources means vehicles cannot complete all tasks or serve all customers in a single trip.
[2] RDG-MAENS: the combination of route distance grouping (RDG) and the memetic algorithm with extended neighborhood search (MAENS).
[3] LNS–ACO: the combination of the large neighbourhood search (LNS) and the ant colony optimization (ACO).
[4] Novel two-phase heuristic: the combination of an improved density-based clustering algorithm and the max-min ant system.
[5] Hybrid genetic algorithm (HGA): the combination of GA and a local search strategy.
[6] CVRP-FA: a firefly algorithm with two types of local search and genetic operators.
[7] EMRG-HA: an evolutionary multi-objective route grouping-based heuristic algorithm.
[8] GSOCI: a glow-worm swarm optimisation algorithm based on chaotic initialization.
[9] Hybrid genetic algorithm (HGA): the combination of GA and 2-opt algorithm.
[10] ILS-RVND-TA: the combination of iterated local search (ILS), variable neighborhood descent (RVND) and threshold accepting (TA).
[11] – indicates the problem decomposition is not involved in the literature.

2021), and so on. Kancharla & Ramadurai (2018) presented an adaptive large neighborhood search (ALNS) with special operators to deal with the electric vehicle routing problem. Islam et al. (2021) combined the particle swarm optimization (PSO) and variable neighborhood search (VNS) to solve the clustered vehicle routing problem. Altabeeb et al. (2021) introduced a cooperative hybrid firefly algorithm (CVRP-CHFA) with multiple firefly algorithm (FA) populations to handle the capacitated vehicle routing problem. Liu et al. (2022) formulated the electric vehicle routing problem with time windows as a mixed integer programming model and developed a hybrid genetic algorithm (HGA). Jia et al. (2022) proposed a confidence-based bi-level ant colony optimization algorithm to solve the capacitated electric vehicle routing problem.

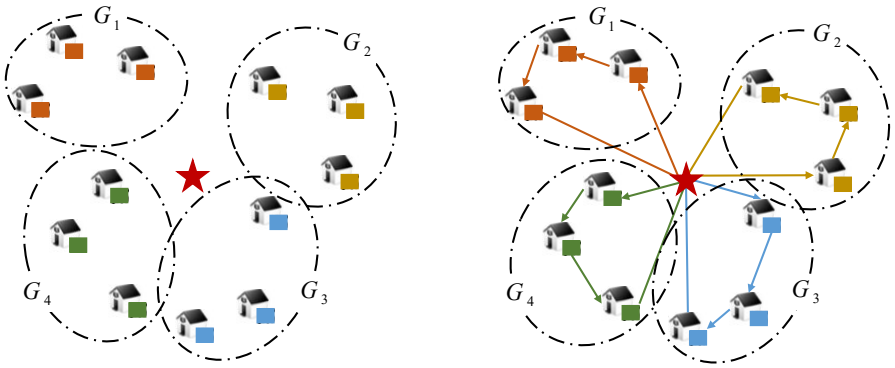
The above studies mostly consider the case of sufficient resources. Even for electric vehicles, recharging stations are introduced to achieve energy replenishment during task execution. However, in emergency rescue, there are no pre-set recharging stations on the way of task execution. Rescue vehicles need to return to the transportation platform for energy supply. Task selection and route optimization under the condition of limited energy are challenging when the attributes of tasks are diverse.

(2) Divide-and-conquer-based heuristic algorithms.

Rescue missions have high requirements for timeliness, rapid and efficient route planning for rescue vehicles will greatly reduce economic losses and casualties. Thus, divide-and-conquer-based heuristic algorithms are adopted to decompose the original problem into several independent sub-problems (Xiao et al., 2021), which mainly includes task-grouping-based (customer-grouping-based) algorithms (Zhang, 2017; Xu et al., 2021; Shang et al., 2022) and route-grouping-based algorithms (Xiao et al., 2021; Zhang et al., 2022b).

For task-grouping-based algorithms, tasks (customers) are grouped based on their locations and the depot, as shown in Fig. 2. Shahab et al. (2016) proposed a two-step genetic algorithm (TSGA) to decompose CVRP into regions (sub-problems) and found the shortest route for each region. Zhang (2017) determined sets of cost-effective feasible groups through the improved density-based clustering algorithm. Each cluster was assigned to one vehicle, and the Max-min ant system was adopted to optimize the route. However, the effectiveness of the above algorithms depends largely on the adopted grouping method. Considering that tasks are different in types, needs, execution times and priorities in disasters, a single-shot grouping based only on

the locations of tasks may not achieve optimal solution. How to effectively group tasks with multiple attributes is a difficult problem (Xu et al., 2021).

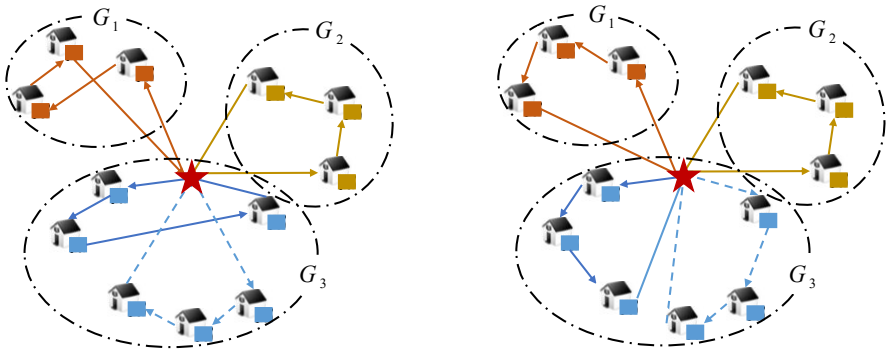


(a) Task grouping result (b) Route planning

Figure 2: The process of task-grouping-based algorithms.

The route-grouping-based algorithms treat routes as the clustering units, as shown in Fig. 3. Mei et al. (2014) developed an effective decomposition scheme called the route distance grouping (RDG) to decompose the problem, which was able to obtain high-quality decompositions and focused the search on the promising regions of the vast solution space. Xiao et al. (2021) introduced an evolutionary multi-objective route grouping-based heuristic algorithm (EMRG-HA) to decompose large-scale CVRPs into multiple subproblems, and adopted a local search method to improve the quality of the routes in the selected subproblem. Similar to task-grouping-based algorithms, the effectiveness of such algorithms is affected by the grouping method. What's more, the applicability of route-grouping-based algorithms is limited under the condition of insufficient resources (Li et al., 2020).

Due to the long cruising range of vehicles involved in traditional CVRPs, the max-duration does not need to be considered during delivery. However, in emergency rescue, taking into account the energy constraint of electric-driven rescue vehicles, the cruising range of them cannot match those of the conventional ones. To effectively improve the rescue efficiency, vehicles need to execute tasks with high importance within their durations, and the remaining ones will be completed during the subsequent durations. What's more, because of the energy constraint, the task scale, that is, the execution time of the task, has to be considered in the task selection process. These



(a) Route grouping result (b) Route adjustment

Figure 3: The process of route-grouping-based algorithms.

problems make it much harder to manage a fleet of rescue vehicles than conventional ones.

In summary, to effectively deal with the routing planning problem in disasters, the following problems need to be considered in VRPEC.

- **Efficient grouping of heterogeneous tasks.** Existing studies mainly cluster tasks based on their locations, and ignore other attributes, resulting in poor quality of problem decomposition. However, in emergency rescue, tasks with various attributes are randomly distributed, so it is challenging to group them effectively.

- **Trade-off among tasks under energy constraint.** Currently, the planning problem with sufficient resources is of great concern. However, in emergency rescue, the electric-driven rescue vehicles have max-durations, and the number of them is limited. When resources are insufficient and tasks have various attributes, the choice of tasks to be executed is a difficult problem.

- **Trade-off among energy consumptions of vehicles.** When the objective is to minimize the total distance, it may cause the energy consumption of each vehicle imbalance. However, failure to fully utilize resources will reduce the overall performance of the system under the condition of insufficient resources.

3. Problem formulation

3.1. Basic assumption

To facilitate the description and understanding of the problem, the following conditions are assumed.

- Each vehicle starts and ends its trip at the depot.
- Each task can be executed by at most one rescue vehicle.
- Since we focus on a homogeneous fleet of rescue vehicles, the inherent loads of them are the same.
- The rescue tasks involved mainly include radiation concentration detection, hazard source search, etc., so the load of each vehicle does not change during the task execution.
- The energy consumption rates of rescue vehicles are consistent and do not change throughout the process.

3.2. Problem statement

VRPEC can be modeled as an undirected graph $G = (A, E)$, where $A = A_T \cup A_0$ is a set of nodes including tasks $A_T = \{1, 2, \cdots, m\}$ and the transportation platform A_0 labeled as 0. We call A_0 as depot in the following content. E denotes the arc set, and each element of it is the Euclidean distance between two nodes. A homogeneous fleet of rescue vehicles is situated at the depot, and K is the set of them. VRPEC can be formulated as follows.

$$\max \sum_{k=1}^{n} \sum_{i=0}^{m} \sum_{j=0}^{m} \omega_j x_{ij}^k \tag{1}$$

$$\text{s.t.} \quad \sum_{i=0}^{m} x_{ih}^k = \sum_{j=0}^{m} x_{hj}^k \quad \forall h \in A_T, \forall k \in K \tag{2}$$

$$\sum_{k=1}^{n} \sum_{i=0}^{m} x_{ij}^k \leq 1 \quad \forall j \in A_T \tag{3}$$

$$\sum_{j=1}^{m} x_{0j}^k = 1 \quad \forall k \in K \tag{4}$$

$$\sum_{i=1}^{m} x_{i0}^{k} = 1 \quad \forall k \in K \tag{5}$$

$$\sum_{k=1}^{n} \sum_{j=1}^{m} x_{0j}^{k} = n \tag{6}$$

$$\sum_{k=1}^{n} \sum_{i=1}^{m} x_{i0}^{k} = n \tag{7}$$

$$\sum_{i=0}^{m} \sum_{j=0}^{m} (\frac{d_{ij}}{v} + at_j) \cdot x_{ij}^{k} \leq E_{\text{th}} \quad \forall k \in K \tag{8}$$

$$\mu_i - \mu_j + 1 - m \cdot (1 - x_{ij}^{k}) \leq 0 \quad \forall i, j \in A_T, i \neq j, \forall k \in K \tag{9}$$

$$x_{ij}^{k} \in \{0, 1\} \quad \forall i, j \in A, \forall k \in K \tag{10}$$

The goal of maximizing the total weight of the executed tasks under the condition of limited energy is determined by Eq. (1). Constraint (2) ensures the arriving and the departing vehicle is the same for a given task. Constraint (3) indicates each task can be executed at most once by the vehicle. Constraints (4)-(5) denote that each vehicle needs to start and end its trip at the depot. Constraints (6)-(7) assure the number of the routes. Constraint (8) ensures the energy consumption of each vehicle is no more than the energy threshold. In this paper, energy consumption is expressed as the total time spent by the vehicle for one trip, including the travel time between tasks and the time to execute tasks, and we call it makespan in the following content. The energy threshold is expressed as the max-duration of the vehicle. Constraint (9) is the sub-tour elimination constraint. Constraint (10) indicate the range of the decision variable x_{ij}^{k}. For convenience, abbreviations and notations involved are summarized and explained in Tables 3 and 4, respectively.

4. Multi-Stage Vehicle Routing Algorithm based on Task Grouping

In this paper, a multi-stage planning algorithm is proposed to solve VRPEC in disasters. The overall framework of the proposed algorithm is shown in Fig. 4. It is easy to get VRPEC is divided into three stages at the problem

Table 3: Abbreviations and interpretations.

Algorithm	Abbreviation	Interpretation
Problem layer	VRPEC	a vehicle routing problem with energy constraint utilized in the post-disaster emergency rescue
Algorithm layer	MSVR-TG	the proposed algorithm, multi-stage vehicle routing algorithm based on task grouping
	MSVR-TG-random	a variant of MSVR-TG where the routes involved in LNS are selected randomly
	K-means-ad	a novel K-means algorithm based on angular density
	PSGA	a problem-specific genetic algorithm
	IGOs	the improved genetic operators
	TGOs	the traditional genetic operators
	RI	random initialization
	HII	hybrid incremental initialization
	HHI	hybrid heuristic initialization
	RR	random removal
	AR	adjacent removal
	TGR	time greedy removal
	TWGR	time-weight ratio greedy removal
	RIs	random insertion
	TGIs	time greedy insertion
	TWGIs	time-weight radio greedy insertion
	HTWGIs	hybrid time-weight radio greedy insertion

level, including the task grouping stage, the sequence planning stage and the route adjusting stage. The role of each stage and the algorithm modules involved are introduced as follows.

4.1. Task grouping

The task grouping stage aims to group heterogeneous tasks according to their locations and the depot, thereby balancing the moving distance of each vehicle as much as possible.

It is well known that K-means dynamically adjusts cluster centers and aggregates the points with similar characteristics (Ahmad & Khan, 2021). However, traditional K-means considers the Euclidean distance between tasks, which may cause some clusters to be far away from the depot. As shown in Fig. 5(a), circles represent tasks, the star represents the depot, and sectors

Table 4: Nomenclature.

	Notation	Definition
Sets	A_T	the set of tasks
	A_0	the transportation platform, named as depot
	K	the set of rescue vehicles
	$\mathcal{W}_{\text{exe}}$	the weight set of the executed tasks
	$\mathcal{W}$	the weight set of all tasks
	AT	the set of task execution times
	$\mathcal{U}$	the set of cluster centers
	$\mathcal{P}$	the location set of tasks
	$\bar{\mathcal{P}}$	the location set of tasks unassigned
	ξ_k	the set of the tasks in group k
	ς_k	the set of the selected tasks on route k
Parameters	n	the number of rescue vehicles
	m	the number of tasks
	d_{ij}	the Euclidean distance between task i and task j
	ω_j	the weight of task j
	at_j	the execution time of task j
	v	the linear velocity of each rescue vehicle
	p_0	the location of the depot
	E_{th}	the energy threshold of each rescue vehicle
	θ_i	the angle between the depot and the task connection line and the positive direction of the x-axis
	θ_i^*	the θ_i corresponding to the task with the largest angular density
	Δ_{angle}	the preset threshold in K-means-ad
	δ	the threshold of the angular density
	$G_{\max}$	the number of iterations in PSGA
	N_p	the population size in PSGA
	p_c	the crossover probability in PSGA
	p_m	the mutation probability in PSGA
	γ	the random factor in HII
	λ	the task deletion ratio in HHI
	η	the weight coefficient of makespan in the objective funtion of PSGA
	σ	the random factor in insertion heuristic
	α_1, α_2	random factors in route selection rule
	β	the task removal ratio in removal heuristic
	T	the number of iterations in LNS
Decision variables	x_{ij}^k	a decision variable. If rescue vehicle k travels from task i to task j, $x_{ij}^k = 1$. Otherwise, $x_{ij}^k = 0$
	μ_i	the access order of task i

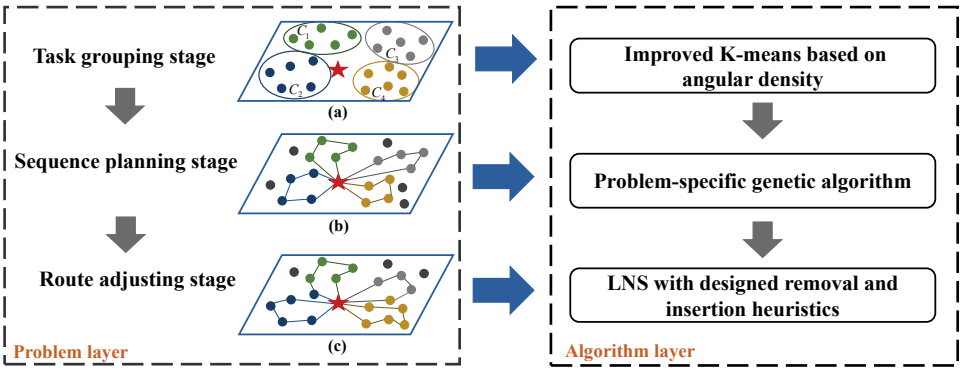
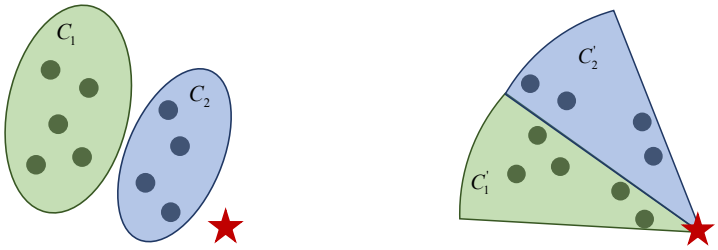


Figure 4: The framework of MSVR-TG.

represent clusters. It is easy to get that the vehicle assigned to cluster C_1 consumes more energy on the way, which greatly reduces the number of the tasks executed under the condition of energy constraint. And the vehicle assigned to cluster C_2 may have excess energy due to the fact that tasks involved are closed to the depot.

To avoid the above problem as much as possible, a novel clustering algorithm based on task angular density (K-means-ad) is proposed, and the grouping result is shown in Fig. 5(b). By comparison, it can be found that the grouping result obtained by K-means-ad makes each vehicle more balanced in terms of the moving distance. In the following content, K-means-ad is introduced in detail.



(a) Grouping result obtained via K-means

(b) Grouping result obtained via K-means-ad

Figure 5: Grouping results obtained by different clustering algorithms.

4.1.1. Cluster center initialization

Although K-means is one of the most powerful clustering algorithms, its result may be affected by the initial centers (Ahmad & Khan, 2021). For incorrect initial centers, effective clustering may not be obtained. Therefore, a center initialization rule based on angular density is proposed. Here, the cluster center is represented as a ray starting from the depot and having an angle with the positive horizontal direction in the range of $[0°, 360°)$.

Definition 1 (Angular density). *Define* $\rho_i = \sum_{j \in A_T, j \neq i} I(|\theta_j - \theta_i| \leq \delta/2)$ [1] *as the angular density of node* i.

The schematic diagram of θ_i is shown in Fig. 6, and δ is the pre-set threshold. Taking tasks i, j and k as examples, the angular density is illustrated. As shown in Fig. 7, the included angle of each colored region is δ. In the blue area, except for task i, the remaining nodes are the neighbors of task i, and ρ_i is the number of neighbors. It is easy to get, the number of neighbors of task i, j and k are 5, 3 and 2 respectively, i.e. $\rho_i = 5, \rho_j = 3, \rho_k = 2$.

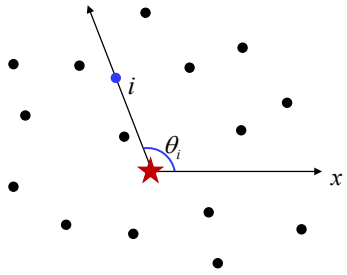


Figure 6: The angle of task i.

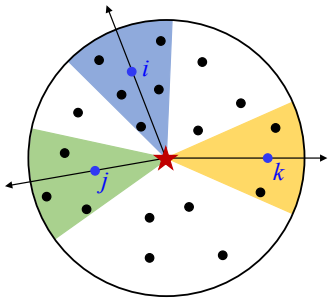


Figure 7: The angular density of each task is calculated based on the number of tasks in the corresponding colored area.

Algorithm 1 is the pseudocode of the center initialization rule based on angular density. Lines 3-5 calculate the angular density of each task. Lines 6-7 select the task with the maximum angular density, and delete it and its neighbors from the alternative set. The above process is repeated until $|\mathcal{U}| = n$.

[1] I represents the indicator function, which means that when the input is true, the output is 1. Otherwise, the output is 0.

Algorithm 1 Cluster center initialization

Input: n, $\mathcal{P}$, p_0, δ;
Output: Set of cluster centers $\mathcal{U}$;

```
1: Initialize U = φ;
2: while |U| < n do
3:     for i = 1 to |P| do
4:         Calculate the angular density ρ_i of task i;
5:     end for
6:     obtain i* with the maximum ρ_i;
7:     Update P by removing i* and its neighbor nodes;
8:     U = U ∪ θ_{i*};
9: end while
```

4.1.2. Task grouping based on angular density

Similar to K-means, K-means-ad iteratively adjusts the grouping result by calculating the distance between each task and the cluster center, namely the included angle. Algorithm 2 is the pseudocode of K-means-ad. Lines 3-5 are the adjustment process of both the cluster center and the task grouping result. Firstly, the center nearest to each task is labeled according to Eq. (11). Then, based on Eq. (12), the average angle of the tasks in each cluster is calculated to update the cluster center. Finally, the change of the current centers and the previous ones is calculated as Eq. (13).

$$c^{(k)} = \arg\min_{k} |\theta_i - \chi_k| \tag{11}$$

$$\chi_k{}' = \frac{\sum_{i=1}^{m} I(c^{(i)} = k) \cdot \theta_i}{\sum_{i=1}^{m} I(c^{(i)} = k)} \tag{12}$$

$$J(\mathcal{U}, \mathcal{U}') = \sum_{k=1}^{n} |\chi_k{}' - \chi_k| \tag{13}$$

where χ_k and $\chi_k{}'$ denote the previous and the current center of cluster k, respectively.

Algorithm 2 K-means-ad

Input: n, $\mathcal{P}$, p_0, δ, Δ_{angle};
Output: Grouping result $\mathcal{C}$;
1: Initialize the cluster center set $\mathcal{U}$ (Algorithm 1);
2: **while** $J(\mathcal{U},\mathcal{U}') > \Delta_{\text{angle}}$ **do**
3: Categorize tasks based on (12);
4: Update the cluster centers in $\mathcal{U}'$ (13);
5: Calculate the difference between two iterations $J(\mathcal{U},\mathcal{U}')$ (14);
6: **end while**

4.2. Sequence planning

The purpose of the sequence planning stage is to optimize the task execution sequence based on the grouping result. Considering that the introduction of the problem specific knowledge can improve the performance of a metaheuristic (Nucamendi-Guillén et al., 2021), PSGA is proposed to generate route of each vehicle satisfying energy constraint. Algorithm 3 is the pseudocode of PSGA, which is described in detail as follows.

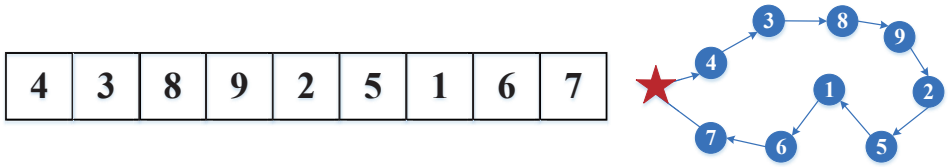
Algorithm 3 Problem-specific genetic algorithm

Input: $\mathcal{P}, p_0, \xi_k, v, AT, \mathcal{W}, E_{\text{th}}, N_p, G_{max}, p_c, p_m, \gamma, \eta, \lambda, \sigma$;
Output: Planning result $\mathcal{R}$;
1: Population initialization (Section 4.2.2);
2: Fitness calculation (Section 4.2.3);
3: **while** maximum iteration G_{max} is not reached **do**
4: Crossover with route improvement strategy (Section 4.2.4);
5: Mutation with route improvement strategy (Section 4.2.5);
6: Fitness calculation (Section 4.2.3);
7: Population update (Section 4.2.6);
8: Restart demand judgment, if necessary, execute restart strategy (Section 4.2.7);
9: **end while**

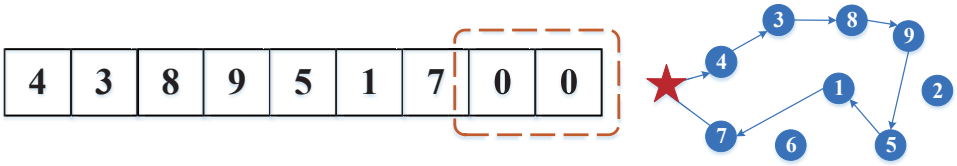
4.2.1. Solution representation

The solution is represented by decimal encoding, with length $|\xi_k|$. Fig. 8(a) shows the encoding for the full execution of tasks. And the opposite situation is shown in Fig. 8(b), where tasks 2 and 6 are not executed due

to the energy constraint. The star and the circles represent the depot and tasks, respectively. The number in each circle represents the index of the task. The population size is denoted by N_p.



(a) Tasks are all executed



(b) Part of the tasks are unexecuted

Figure 8: Solution representation.

4.2.2. Population initialization

In evolutionary algorithms, a promising initial solution can speed up the exploration of the solution space, thus accelerating the convergence of the algorithm (Guo et al., 2020). Therefore, the problem knowledge is utilized to initialize the population and three initialization rules are introduced, including the random initialization (RI), hybrid incremental initialization (HII), and hybrid heuristic initialization (HHI).

- Random initialization (RI)

RI randomly selects tasks from the unassigned task set in turn under the energy constraint and places them at the tail of the route, which can ensure the diversity of the population.

- Hybrid incremental initialization (HII)

Although RI can increase the diversity of the population, it leads to blindness in exploration due to the lack of problem knowledge (Guo et al., 2020). Considering that the construction heuristic rule has advantages in utilizing problem knowledge (Azad et al., 2017; Chang et al., 2018), it is

applied to initialize the population. To increase the population diversity, a random factor $\gamma \in (0,1)$ is introduced. In detail, for each individual, if $rand < \gamma$ ($rand \in (0,1)$), one of the unassigned tasks is randomly selected on the premise of satisfying the energy constraint. Otherwise, the one with the minimum time-weight ratio is selected. The expression of the time-weight ratio ψ_{pq} when task q is inserted after task p is expressed as Eq. (14). The selection rule is described as Eq. 15. ς_k represents the selected tasks on route k.

$$\psi_{pq} = \frac{(d_{pq} + d_{q0} - d_{p0})/v + at_q}{\omega_q} \quad \forall p \in \varsigma_k, q \in \xi_k/\varsigma_k \tag{14}$$

$$q^* = \arg\min_q \psi_{pq} \tag{15}$$

- Hybrid heuristic initialization (HHI)

Although HII introduces problem knowledge in the process of individual generation, it does not consider the overall quality of the route, and there may be a large deviation from the optimal solution. In HHI, population initialization is divided into two steps. Firstly, γ in HII is set to 0 to generate the first individual *pop*. Next, based on *pop*, the remaining individuals are obtained through a series of operations. In detail, after deleting a certain number of tasks randomly in *pop*, part of the tasks in the alternative set are selected based on the proposed **route improvement strategy**. $\lambda \in (0,1)$ represents the proportion of deletion. Algorithm 4 is the pseudocode of HHI.

Algorithm 4 Hybrid heuristic initialization (HHI)

Input: $\mathcal{P}$, p_0, ξ_k, N_p, v, AT, $\mathcal{W}$, E_{th}, γ, λ;
Output: The initial population P_{ini};

```
Construct pop based on the time-weight ratio greedy rule in HII;
P_ini = pop;
for j = 2 to N_p do
    pop' = delete(pop, ξ_k, λ);
    pop'' = RouteImprovementStrategy(pop', ξ_k, P, p_0, v, AT, W, E_th);
    P_ini = P_ini ∪ pop'';
end for
```

The details of the **route improvement strategy** is shown in Fig. 9. The orange area indicates the tasks that can be selected, and the triangles denote the locations where tasks can be inserted. The blue data in each yellow box represents the weight of the task, and the purple rectangles represent the execution time increment of inserting each task at the corresponding location. When selecting tasks for insertion, the makespan increment of inserting each task at each location is calculated. The makespan increment ΔT_q^k for inserting task q between node i and node j is expressed as Eq. (16). It can be seen that inserting task 4 in the third location can obtain the minimum time-weight ratio and satisfy the energy constraint. The expression of time-weight ratio $\Delta\psi_q^k$ for inserting task q between node i and node j is shown in Eq. (17). After inserting task 4, the insertable locations, the optional tasks, and the makespan increment need to be updated. The above task insertion operation is called the route improvement strategy, which is also involved in both crossover and mutation operators.

$$\Delta T_q^k = \frac{d_{iq} + d_{qj} - d_{ij}}{v} + at_q \quad \forall i, j \in \{0, \varsigma_k\}, q \in \xi_k/\varsigma_k, k \in K \tag{16}$$

$$\Delta\psi_q^k = \Delta T_q^k/\omega_q \tag{17}$$

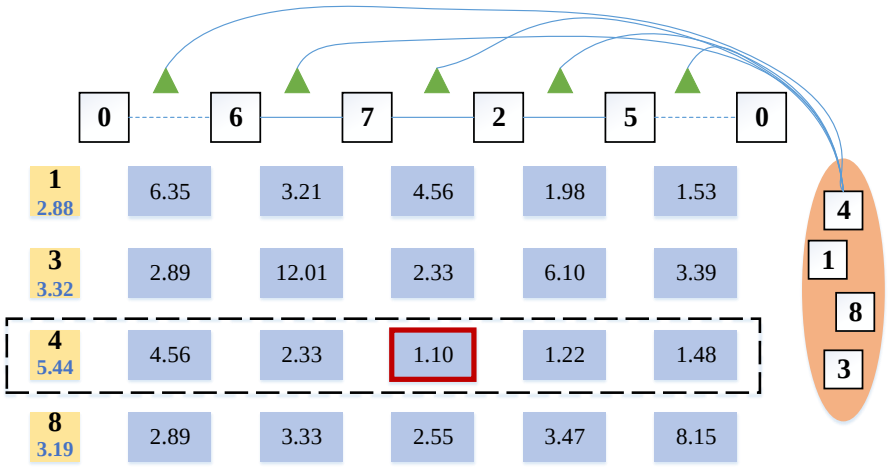


Figure 9: Route improvement strategy.

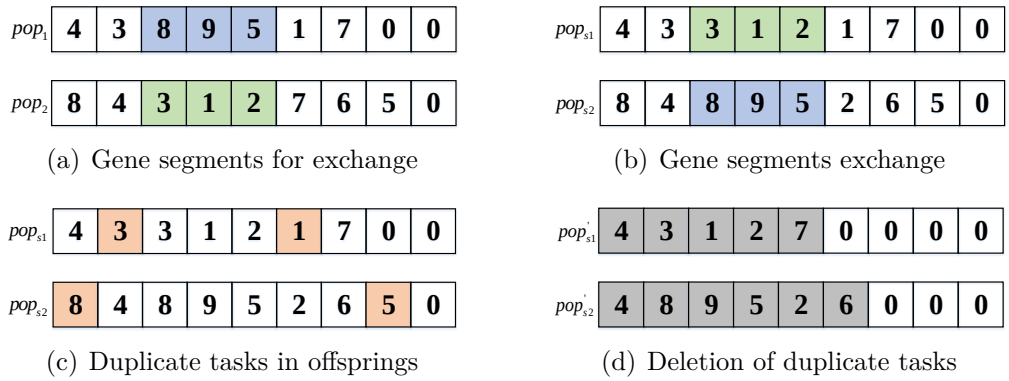
4.2.3. Objective funtion

It can be seen from Eq. (1) that the goal is to maximize the total weight of the executed tasks under the energy constraint. When the total weights of routes are the same, the smaller the makespan, the more likely it is to obtain a high total weight in the route adjusting stage (Lan et al., 2022). Therefore, to provide a promising planning result, the makespan of vehicle k is taken as a part of the objective function f_k, which is expressed as Eq. (18), and $\eta \in (0, 1)$.

$$f_k = \sum_{i \in \{0, \xi_k\}} \sum_{j \in \{0, \xi_k\}} (\omega_j + \eta \cdot (\frac{d_{ij}}{v} + at_j)) \cdot x_{ij}^k \quad \forall k \in K \tag{18}$$

4.2.4. Crossover

The crossover operator plays an important role in the population exploration process (Daglayan & Karakaya, 2016). Combined with the characteristics of the problem, the partially matched crossover (PMX) (Umbarkar & Sheth, 2017) is improved to enhance the quality of the offspring individuals and accelerate the convergence, as shown in Fig. 10.



(a) Gene segments for exchange

(b) Gene segments exchange

(c) Duplicate tasks in offsprings

(d) Deletion of duplicate tasks

Figure 10: Improved crossover operator.

Like PMX, the improved crossover operator randomly selects gene segments of parents for exchange, as shown in Fig. 10(a)-(b), and obtains two offspring individuals pop_{s1} and pop_{s2}. However, considering the limitation of the energy, some tasks do not exist in parents. To enhance the exploration of the decision space, the duplicate tasks marked in Fig. 10(c) that are not in the swap positions are deleted, and the result is shown in Fig.

10(d). Since deleting some tasks may cause energy constraint unsaturated, the route improvement strategy in Section 4.2.2 is utilized to improve the offsprings.

4.2.5. Mutation

Mutation operator simulates the evolution process in the biological world. Although the occurrence probability is small, mutation is very important for the generation of new species. The exchange mutation operator is adopted to mutate individuals (Karakatič & Podgorelec, 2015), as shown in Fig. 11. The route improvement strategy in Section 4.2.2 is utilized to improve the offspring individual pop_s.

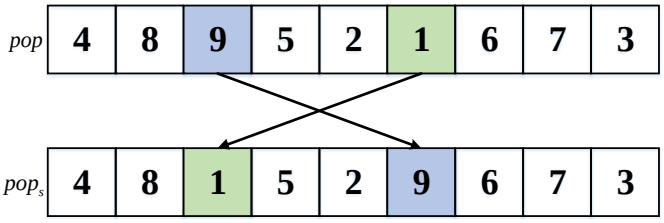


Figure 11: Exchange mutation operator.

4.2.6. Population update

In each iteration, due to the introduction of the genetic operators, the population size will be larger than N_p, and screening is required. Elite selection strategy is adopted to update the population (Karakatič & Podgorelec, 2015). All individuals in the parent population and the offspring population are sorted in descending order based on their objective values, and the first N_p individuals are selected to enter the next iteration.

4.2.7. Restart strategy

In the iterative process, the diversity of the population may gradually decrease. To avoid evolutionary stagnation due to the loss of diversity, the restart strategy is employed (Guo et al., 2020). After updating the population, the similarity SD_i between each individual pop_i and the optimal individual pop^* is calculated to determine whether a restart is required (Altabeeb et al., 2019). The expression of SD_i between pop^* and pop_i is shown as Eq. (19). Denote ϑ as the set of individuals whose $SD_i = 1$. When $|\vartheta|$ is

greater than $\sigma \cdot N_p$ ($\sigma \in (0,1)$ is a pre-set threshold), individuals in ϑ will be reinitialized by the initialization rules in Section 4.2.2.

$$SD_i = sn_i / \min(|\varsigma_{pop^*}|, |\varsigma_{pop_i}|) \tag{19}$$

where sn_i is the number of non-zero elements in the same position in both pop^* and pop_i, as can be seen in Fig. 12. $|\varsigma_{pop^*}|$ and $|\varsigma_{pop_i}|$ represent the numbers of non-zero elements in pop^* and pop_i, respectively.

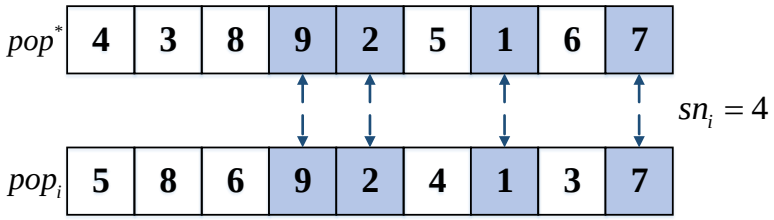


Figure 12: Example of calculating sn_i for pop_i.

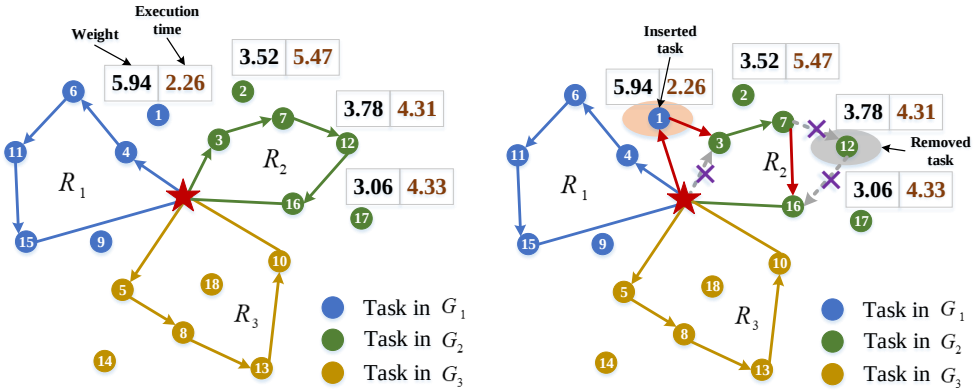
4.3. Route adjusting

Since tasks have multiple attributes, a single grouping of them can be biased, requiring fine-tuning of routes. With the purpose of effectively balancing the energy consumption of each vehicle and maximizing the total weight of the executed tasks, the route adjusting stage makes reasonable trade-off among tasks with different execution times and weights.

Various factors may affect the planning result, such as attributes of tasks and the durations of vehicles, so the initial grouping result may have deviation. In other words, when tasks belonging to the same group are executed collaboratively by different vehicles, it may bring greater total weight, as shown in Fig. 13. $R_k(k \in K)$ and $R'_k(k \in K)$ denote the routes of vehicle k before and after adjustment, respectively. $G_k(k \in K)$ represents the task set belonging to group k.

If the route of each vehicle is planned strictly according to the grouping result, the optimal planning result is shown as Fig. 13(a). However, since the weights of the tasks in G_2 are lower compared to those of the unassigned tasks in G_1, adjusting routes R_1 and R_2 results in a larger total weight, as shown in Fig. 13(b). The tasks removed and inserted after adjustment are indicated in gray and orange areas, respectively. The red lines represent the added elements, and the gray lines represent the deleted ones. In each

rectangle, the number on the left represents the weight of the task, and that on the right represents the execution time.



(a) Initial planning result (b) Planning result after adjustment

Figure 13: Route adjustment.

As mentioned above, the grouping result may be biased, so LNS is utilized to dynamically adjust the tasks on each route. LNS has two successive operators including removal heuristic and insertion heuristic (Shaw, 1998; Kancharla & Ramadurai, 2018; Akpunar & Akpinar, 2021). The former removes some components of the current solution, while the latter fixes the destroyed solution by inserting some removed ones via a specific rule. The newly generated solution is evaluated via an acceptance function to decide whether to accept it as the new current solution.

Algorithm 5 shows the pseudocode of the LNS algorithm utilized in the route adjusting stage of MSVR-TC. The route selection rule and the designed removal and insertion heuristics involved in LNS will be described in detail in the following content.

4.3.1. Route selection rule

In existing researches, adjustment is often applied to one or more routes selected randomly (Machado et al., 2021; Zachariadis et al., 2022). Although random selection can increase the exploration of the decision space, there may be many useless attempts due to its blindness. To reduce the number of invalid operations, the gravity center approach and the shannon entropy approach are introduced in (Polat et al., 2015) and (Öztaş & Tuş, 2022), respectively. The above approaches are combined with the roulette wheel

Algorithm 5 LNS

Input: $\mathcal{R}, \mathcal{P}, T, p_0, v, AT, \mathcal{W}, E_{\text{th}}, \alpha_1, \alpha_2, \beta$;
Output: The improved planning result $\mathcal{R}'$;
$\mathcal{R}' = \mathcal{R}$;
while maximum iteration T is not reached **do**
    I_s = RouteSelectionRule($\mathcal{R}', \mathcal{P}, p_0, v, AT, \alpha_1, \alpha_2$) (Algorithm 6);
    $[I'_s, \bar{\mathcal{P}}]$ = RemovalHeuristic($I_s, \mathcal{P}, p_0, v, AT, \mathcal{W}, \beta$);
    I''_s = InsertionHeuristic($I'_s, \bar{\mathcal{P}}, v, AT, \mathcal{W}, E_{\text{th}}$);
    if I''_s satisfies the aspiration criterion **then**
        Replace I_s with I''_s;
        Update $\mathcal{R}'$;
    end if
end while

technique for route selection, and achieve good results. Considering the characteristics of VRPEC, a new route selection rule is designed in MSVR-TG to effectively improve the search efficiency, shown as Algorithm 6. In each iteration, a neighborhood search is performed on two routes chosen via different rules. To enhance the exploration of the decision space, two random factors are introduced, including $\alpha_1 \in (0, 1)$ and $\alpha_2 \in (0, 1)$.

Algorithm 6 Route selection rule

Input: $\mathcal{R}, \mathcal{P}, p_0, v, AT, \alpha_1, \alpha_2$;
Output: Selected routes I_s;
$Time_{\text{exe}}$ = TimeCalculation($\mathcal{R}, \mathcal{P}, p_0, v, AT$);
if $rand > \alpha_1$ **then**
    Select the route with the minimum $Time_{\text{exe}}$;
else
    Select a route randomly;
end if
if $rand > \alpha_2$ **then**
    Select the one closest to the first route;
else
    Select a route that is different from the first one randomly;
end if

- For the first route, if $rand > \alpha_1$, the one with the minimum makespan

is selected. Otherwise, a route is selected randomly. This rule takes into account that routes with more remaining energy will have more room for adjustment (Lan et al., 2022).

- For the second route, if $rand > \alpha_2$, the one closest to the first route is selected. Otherwise, a route different from the first one is selected randomly. It is easy to get that routes with less spatial distance are more likely to have valid task exchanges (Lan et al., 2022). The expression of distance between routes R_u and R_v is shown in Eq. (20) (Xiao et al., 2021).

$$Dis(R_u, R_v) = \frac{\sum_{i \in R_u} \sum_{j \in R_u} d_{ij}}{N_{R_u} \times N_{R_u}} \quad \forall u, v \in K \tag{20}$$

where N_{R_u} and N_{R_v} denote the numbers of tasks in route R_u and R_v, respetively.

4.3.2. Removal heuristic

Combined with the problem knowledge, four removal heuristics are introduced, and the details of them are as follows. Here, β represents the proportion of tasks removed during the process of destroying the solution.

Random removal (RR): for route k, $[\beta \cdot |\varsigma_k|]$ tasks are randomly removed in turn, and the remaining ones are connected in sequence.

Adjacent removal (AR): for route k, the first task to be removed is randomly selected, based on which, $[\beta \cdot |\varsigma_k|] - 1$ tasks are removed one after another from route k. Specifically, the Euclidean distance between each task on the route and each one that has been removed is calculated, and the task on the route with the smallest distance value is removed in turn.

Time greedy removal (TGR): for route k, $[\beta \cdot |\varsigma_k|]$ tasks are removed with the maximum makespan reduction $\Delta\tilde{T}_q^k$ in turn. $\Delta\tilde{T}_q^k$ denotes the change in the makepan of removing task q, expressed as Eq. (21).

$$\Delta\tilde{T}_q^k = \begin{cases} \frac{d_{(q-1)q}}{v} + \frac{d_{q(q+1)}}{v} - \frac{d_{(q-1)(q+1)}}{v} + at_q & others \\ \frac{d_{(q-1)q}}{v} + \frac{d_{q0}}{v} - \frac{d_{(q-1)0}}{v} + at_q & q = |\varsigma_k| \end{cases} \tag{21}$$

Time-weight ratio greedy removal (TWGR): for route k, $[\beta \cdot |\varsigma_k|]$ tasks are removed with the maximum time-weight ratio $\Delta\tilde{\psi}_q^k$ in turn. $\Delta\tilde{\psi}_q^k$ denotes the time-weight ratio of removing task q, and its calculation is shown in Eq. (22).

$$\Delta\tilde{\psi}_q^k = \Delta\tilde{T}_q^k/\omega_q \quad \forall q \in \varsigma_k \tag{22}$$

4.3.3. Insertion heuristic

In combination with energy constraint and the attributes of tasks, four insertion heuristics are utilized, and the detailed introduction is as follows.

Random insertion (RIs): unassigned tasks are selected and inserted into route randomly under the condition of the energy constraint.

Time greedy insertion (TGIs): the unassigned task with the minimum makepan increment ΔT_q^k is selected and inserted into the corresponding location of route k in turn under the condition of the energy constraint. The expression of ΔT_q^k is shown as Eq. (16).

Time-weight radio greedy insertion (TWGIs): the unassigned task with the minimum time-weight ratio $\Delta\psi_q^k$ is selected and inserted into the corresponding location of route k in turn under the condition of the energy constraint. The expression of $\Delta\psi_q^k$ is shown as Eq. (17).

Hybrid time-weight radio greedy insertion (HTWGIs): there are two ways of task selection and insertion. Under the condition of the energy constraint, one is to randomly select and insert unassigned tasks with probability $\varepsilon \in (0, 1)$ and the other is to select the one with the minimum $\Delta\psi_q^k$.

4.4. Computational complexity

There are three main steps in MSVR-TC algorithm, and the computational complexity of each step is shown in Table 5.

- The complexity of K-means-ad is $O(nm^2)$.
- The complexity of PSGA is $O(\mathrm{G}_{\max} N_p \overline{m_\xi}^3)$.
- The complexity of LNS in MSVR-TC is $O(Tm^3)$, where T is the number of iterations of LNS.

To sum up, the worst-case computational complexity of MSVR-TG is $O(\mathrm{G}_{\max} N_p k \overline{m_\xi}^3 + Tm^3)$.

Table 5: Computational complexity analysis of MSVR-TG.

Algorithm	**Operation**		**Computational complexity**
K-means-ad	Center initialization		$O(nm^2)$
	Center adjustment		$O(nml)$
PSGA	Population Initialization	RI	$O(N_p\overline{m_\xi})$
		HII	$O(N_p\overline{m_\xi}^2)$
		HHI	$O(N_p\overline{m_\xi}^3))$
	Objective function		$O(\mathrm{G_{max}}N_p\overline{m_\xi})$
	Crossover		$O(\mathrm{G_{max}}N_p\overline{m_\xi}^3)$
	Mutation		$O(\mathrm{G_{max}}N_p\overline{m_\xi}^3)$
	Population update		$O(\mathrm{G_{max}}N_p\log(N_p))$
	Restart strategy		$O(\mathrm{G_{max}}N_p\overline{m_\xi}^2)$
LNS	Route selection	Route I	$O(nm)$
		Route II	$O(nm^2)$
	Removal heuristic	RR	$O(m)$
		AR	$O(m^3)$
		TGR	$O(m^2)$
		TWGR	$O(m^2)$
	Insertion heuristic	RIs	$O(m)$
		TGIs	$O(m^3)$
		TWGIs	$O(m^3)$
		HTWGIs	$O(m^3)$

[1] l: the number of iterations of K-means-ad.
[2] $\overline{m_\xi}$: the maximum number of tasks in each group.

4.5. Illustrative example

For a better understanding of MSVR-TG, we present a typical numerical example. The parameters of each task are shown in Table 6. The number of the rescue vehicles is 3, and the energy threshold is 30.

Table 6: The parameters of tasks in the typical numerical example.

Index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
ω_i	5.74	3.52	3.92	5.52	4.78	5.32	5.38	4.36	4.99
at_i	3.26	4.47	3.07	2.83	1.89	2.19	3.15	2.45	3.48
Index	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
ω_i	5.71	4.12	5.51	4.52	3.16	5.17	4.99	3.94	4.37
at_i	3.06	2.19	3.36	3.47	4.22	2.55	2.25	4.27	4.12

As can be seen from Fig. 14, the task grouping result consists of three parts, including $(1, 4, 6, 9, 11, 15)$, $(2, 3, 7, 12, 16, 17)$ and $(5, 8, 10, 13, 14, 18)$. Based on the grouping result, the initial route of each vehicle obtains via PSGA, containing $(0, 4, 6, 11, 9, 0)$, $(0, 3, 2, 7, 16, 0)$ and $(0, 5, 8, 18, 10, 0)$. It shows that tasks within each group are not fully executed due to the energy constraint. Through evolutionary process, the initial solution is optimized as $(0, 4, 6, 11, 15, 0)$, $(0, 3, 7, 12, 16, 0)$ and $(0, 5, 8, 13, 10, 0)$ with a higher total weight. Since the unassigned tasks in G_1 have higher weights than tasks belonging to G_2 and G_3, the planning result is further optimized by performing the designed removal and insertion heuristics repeatedly until the termination condition is fulfilled. The route adjusting part shows the improved results where R_1 and R_2, and R_1 and R_3 are selected in sequence. The grey dots indicate the unassigned tasks participating in LNS. The final result shows that tasks belonging to the same group initially are performed by different vehicles, and the tasks involved are marked with the yellow oval area in Fig. 14.

5. Simulations and results

In this section, the performance of MSVR-TG is tested in a variety of scenarios and the results are analyzed in detail. Computations are performed using Matlab software version R2020b, on a personal computer with 64-bit operating system, Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz.

5.1. Test case generator

To our knowledge, there is no recognized benchmark for VRPEC. Therefore, a new set of benchmark cases that comprehensively consider the characteristics of the route planning problem in disasters is designed, following the approach that generates benchmarks for the CVRP (Uchoa et al., 2017). The test case generator is applied to generate cases from five dimensions including the number of vehicles, the locations of tasks, the size of the environment, the energy threshold and the location of the depot, as shown in Table 7.

Tasks are located randomly in a square, whose side length is l_{zone}. The weight and the execution time of each task are obtained in the range of $[5, 15]$ and $[1, 10]$, respectively. To make the benchmark representative and avoid chance, the locations, weights and execution times of tasks and the energy thresholds of vehicles are randomly generated within the corresponding interval. Considering the particularity of VRPEC, the number of vehicles in

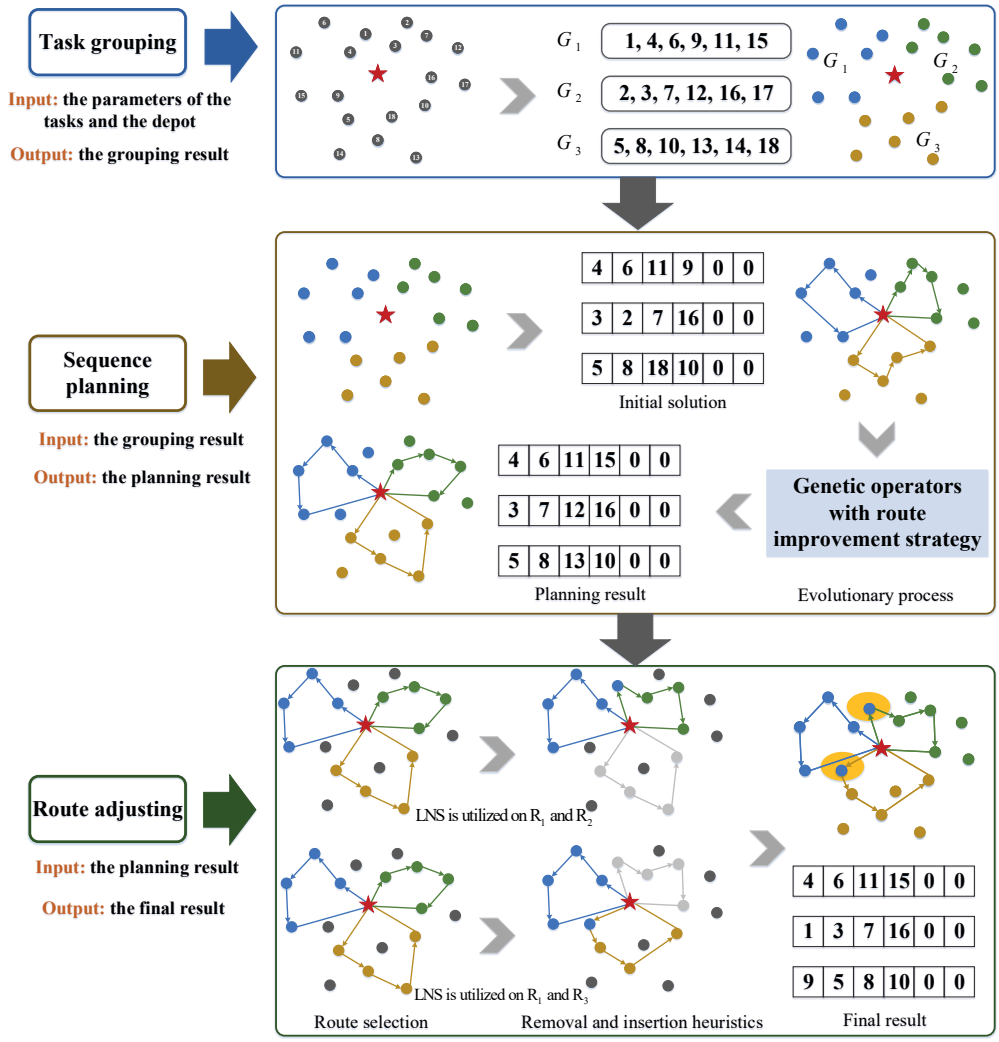


Figure 14: An example to illustrate the procedure of the proposed algorithm.

each case is fixed, ranging from $[2, 10]$. The difference between cases in set A and set B lies in the scale of the problem, such as the number of agents, the number of tasks, the scope of the environment, etc. Cases in set P-I and set P-II are utilized to test the influence of the depot location on the performance of the algorithm. In above cases, three different types of locations for the depot are considered, mainly including central, eccentric and random locations. The corresponding ranges are marked in Table 7.

Table 7: Test cases for VRPEC.

Set	**Case**	n	m	E_{th}	l_{zone}	p_0
A	A-1	2	20	90	20	(10, 10)
	A-2	2	25	100	20	(15, 10)
	A-3	2	20	80	15	(5, 8)
	A-4	2	15	60	10	(5, 5)
	A-5	2	18	65	10	(3, 2)
	A-6	2	25	80	15	(7, 8)
	A-7	2	25	80	20	(15, 15)
	A-8	2	28	100	18	(10, 10)
	A-9	2	25	100	20	(15, 15)
	A-10	3	20	55	15	(7, 8)
B	B-1	3	40	150	40	(20, 20)
	B-2	3	60	150	30	(15, 15)
	B-3	5	90	150	30	(15, 15)
	B-4	4	100	150	20	(10, 10)
	B-5	4	40	100	30	(15, 15)
	B-6	4	50	120	40	(20, 20)
	B-7	5	50	90	30	(15, 15)
	B-8	5	70	150	50	(25, 25)
	B-9	5	70	150	50	(10, 15)
	B-10	6	180	150	30	(15, 15)
	B-11	6	60	120	50	(25, 25)
	B-12	8	100	150	60	(30, 30)
	B-13	7	150	180	40	(20, 20)
	B-14	9	120	120	50	(25, 25)
	B-15	10	150	150	50	(25, 25)
P-I	P-I-v $v = 1, 2, ..., 10$	4	60	150	40	([0, 40], [0, 40])
P-II	P-II-v $v = 1, 2, ..., 10$	8	120	150	50	([0, 50], [0, 50])

([0, 40], [0, 40]) and ([0, 50], [0, 50]) specify the ranges of the abscissa and ordinate of the depot in each case group, and 10 different depot locations are obtained for P-I and P-II, respectively.

5.2. Parameter setting

Since the performance of the algorithm varies with the values of parameters, Taguchi method (Ding et al., 2018) is adopted to adjust some important ones, including $\delta, \alpha_1, \alpha_2, \beta, T$. The method is implemented with orthogonal arrays containing all the information about the factors that affect the performance of the algorithm. The factors involved are divided into two types: (1) controllable or signal factors; (2) noise factor. The Taguchi method seeks

to find the optimal combination of signal factor levels that minimizes the effects of noise factors in the response.

In this paper, the response value is the total weight of the executed tasks, shown in Eq. (1), and the larger-the-better type of response is adopted. Through preliminary experiments, we set four conditions for the value of each parameter, that is, $\delta \in \{5, 10, 15, 20\}$, $\alpha_1 \in \{0.1, 0.3, 0.5, 0.7\}$, $\alpha_2 \in \{0.1, 0.3, 0.5, 0.7\}$, $\beta \in \{0.1, 0.3, 0.5, 0.7\}$, $T \in \{10, 20, 30, 40\}$. Thus, $L_{16}(4^5)$ orthogonal table is utilized to adjust them. The experimental result and the setting of each parameter are shown in Fig. 15 and Table 8, respectively.

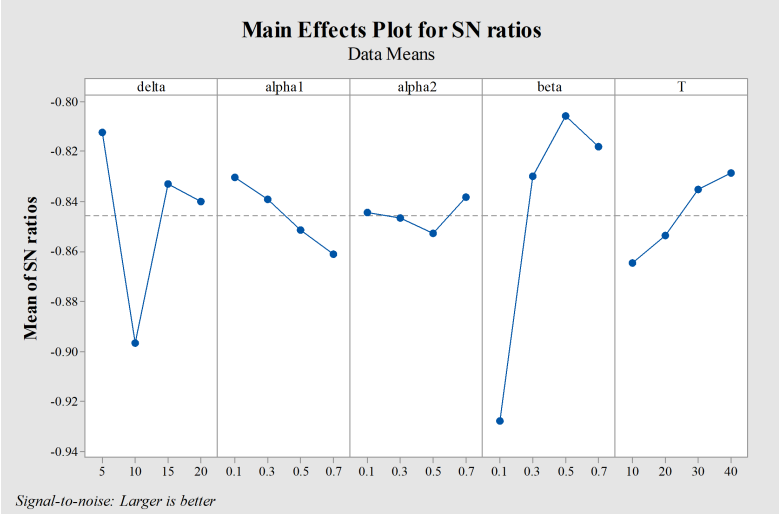


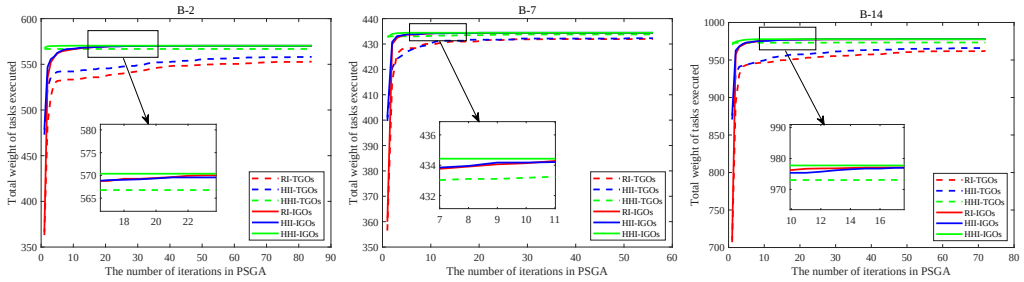
Figure 15: Parameter setting via Taguchi method.

5.3. Performance test of PSGA

Firstly, the performances of population initialization rules and genetic operators designed in PSGA are tested. The combination of crossover and mutation operators utilized in PSGA is expressed as the improved genetic operators (IGOs), where the route improvement strategy is adopted. The combination of crossover and mutation operators for comparison, that is, the combination of PMX and exchange mutation, is represented as the traditional genetic operators (TGOs), where the route improvement strategy is not adopted. Different combinations of initialization rules and genetic operators are treated as different algorithms. Fig. 16 shows the convergence speeds and the solutions of these above algorithms based on part of the cases in Table 7. The following conclusions can be drawn from the curves.

Table 8: Parameter setting for MSVR-TG.

Stage	Algorithm	Parameter	Setting
Task grouping stage	K-means-ad	δ	5°
Sequence planning stage	PSGA	N_p	2*$\lvert\xi_k\rvert$
		G_{max}	2*N_p
		p_c	0.85
		p_m	0.15
		γ	0.5
Route adjusting stage	LNS	α_1	0.1
		α_2	0.7
		β	0.5
		T	40



(a) (b) (c)

Figure 16: Evolutionary curves for different combinations of initialization rules and genetic operators.

1. **The quality of the initial population obtained by HHI is significantly better than those generated by the other two rules.** Although RI guarantees the diversity of the population, its randomness causes blindness in construction, so the optimal initial solution obtained by RI has the minimum value in each case. In HII, problem knowledge is introduced in the process of construction but lacks globality. While the random factor increases the diversity, it may also bring about the problem that high-quality individuals may not be obtained due to the inappropriate task selection. In HHI, the above problems are solved. Since HHI takes a more comprehensive consideration in task selection, it avoids the problem of overall quality decline due to the improper task selection to a certain extent.

2. **The introduction of the route improvement strategy effectively speeds up the convergence of the algorithm and improves the quality of the solution.** TGOs only guarantees the availability of the solution, but not its efficiency. On the basis of retaining the excellent genes of the parents, the route improvement strategy can adjust the improper sequences. After the exchange of gene segments, removing duplicate tasks in offsprings can effectively improve the exploration of the decision space in the process of performing the route improvement strategy. At the same time, the introduction of the route improvement strategy for the mutated route also enhances the quality of the solution.

To sum up, PSGA adopts the combination of hybrid heuristic initialization rule (HHI) and the improved genetic operators (IGOs) in the following simulations.

Next, the optimal initial solution obtained via HHI will be compared with two heuristic algorithms, which are briefly introduced as follows.

- HETRF (Chen et al., 2019)

Highest effective time ratio first algorithm (HETRF) is proposed to assign regions and obtain coverage orders for agents. The scanning time of regions and the flight time between regions are mainly considered in the planning process, and the effective time is maximized as much as possible.

- WTSC (Li et al., 2020)

Weighted targets sweep coverage algorithm (WTSC) is designed to solve a planning problem, where a set of agents are dispatched to efficiently patrol the tasks in the given area to achieve maximum coverage in the minimum time.

Considering that the total weight of each case varies greatly, the results are normalized for display. τ denotes the task execution rate, expressed as Eq. (23). Since the solution of PSGA utilizing HHI for initialization is related to the grouping result of K-means-ad, the initialization solution of PSGA is expressed as K-means-ad + HHI. As shown in Fig. 17, the average of the best initial solutions obtained by K-means-ad + HHI in 20 runs is better than those of the two heuristic algorithms in each case, which preliminarily illustrates the effectivenesses of K-means-ad and HHI.

$$\tau = \mathcal{W}_{\text{exe}}/\mathcal{W} \tag{23}$$

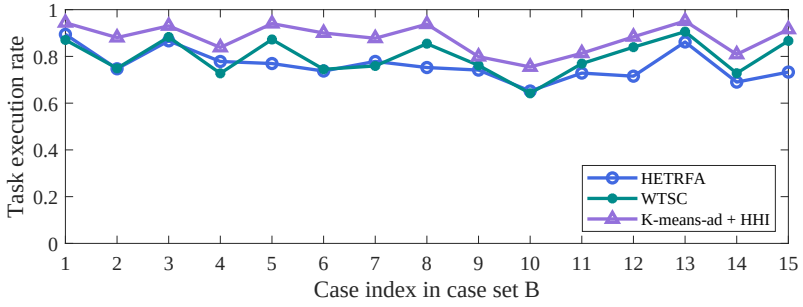


Figure 17: Comparison on the task execution rate obtained by K-means-ad+HHI with those generated by two heuristic algorithms.

To further prove the performance of the solution f_{PSGA} obtained by PSGA, CPLEX is adopted to calculate the route of each vehicle in the sequence planning stage. As shown in Table 9, PSGA has obtained the optimal solution in 9 cases, and the total number of optimal routes tn_{or} is 49, which is the total number of the vehicles in the above cases. In the other 6 cases, tn_{or} is 27, and the total number of routes without optimal solution is 8, whose gaps are within 0.1%. For the maximization problem, the expression of the gap is shown as Eq. (24). f^* and $\overline{f_{\text{PSGA}}}$ represent the optimal value of the problem and the average value in 20 runs obtained by PSGA, respectively. It is confirmed that the designed initialization rule and genetic operators can better balance the diversity and the convergence, and obtain a promising solution.

$$gap\% = \frac{f^* - \overline{f_{\text{PSGA}}}}{\overline{f_{\text{PSGA}}}} \times 100\% \tag{24}$$

5.4. Performance test of LNS

Firstly, the effects of different combinations of removal and insertion heuristics are compared. The variants of MSVR-TG with different combinations can be viewed as different algorithms. Based on case set B in Table 7, each algorithm runs independently for 20 times on each case. The Wilcoxon rank-sum test is utilized with a significance level of 0.05. If one algorithm is superior to another, the rank-sum test result is 1. Otherwise,

Table 9: Comparison on results of PSGA and CPLEX on case set B.

Case	**PSGA**		**CPLEX**		**gap**%
	f	Time/s	f	Time/s	
B-1	375.87 ± 0.21	4.58 ± 0.22	375.94	2.31	0.02
B-2	**570.37±0.00**	9.45 ± 0.23	570.37	11.60	0.00
B-3	**855.03±0.00**	14.53 ± 0.34	855.03	24.07	0.00
B-4	832.50 ± 0.17	20.43 ± 0.30	832.55	423.10	0.01
B-5	**387.81±0.00**	3.17 ± 0.11	387.81	2.78	0.00
B-6	**496.23±0.00**	5.03 ± 0.15	496.23	5.42	0.00
B-7	**434.43±0.00**	3.65 ± 0.14	434.43	4.85	0.00
B-8	638.00 ± 0.17	7.99 ± 0.19	638.10	15.22	0.02
B-9	**543.85±0.00**	7.93 ± 0.16	543.85	265.59	0.00
B-10	1459.00 ± 0.25	43.64 ± 1.11	1459.21*	−	0.01
B-11	**500.40±0.00**	4.49 ± 0.13	500.40	9.44	0.00
B-12	**890.29±0.00**	10.76 ± 0.28	890.29	25.88	0.00
B-13	1407.50 ± 1.36	29.41 ± 0.18	1408.96	215.87	0.10
B-14	**977.74±0.00**	11.24 ± 0.15	977.74	175.00	0.00
B-15	1391.50 ± 0.19	19.91 ± 0.20	1391.89	212.62	0.03

[1] Each data in the second column is the mean and standard deviation of the solutions obtained by PSGA in 20 runs, and that in the forth column is the (approximate) optimal solution obtained by CPLEX in sequence planning stage, where * represents the approximate optimal one within 2 hours.

[2] Each data in the third column represents the mean and standard deviation of running time obtained by PSGA in 20 runs, and that in the fifth column represents the running time of CPLEX, where − indicates that the running time is more than 2 hours.

[3] The optimal solutions obtained by PSGA are indicated in bold.

the rank-sum test result is 0. The result of the statistical analysis is shown in Table 10. Each data denotes the total score of each algorithm in comparison with others, which is the sum of the rank-sum test results in all cases.

According to Table 10, RR-TWGIs has the best performance among 16 variants of MSVR-TG, whose total score is 208. When the insertion heuristic is fixed, the variants of MSVR-TG with random removal (RR) and adjacent removal (AR) heuristics have higher scores in most cases, which implies that introducing randomness has an advantage in exploring the complex solution space. Conversely, when the removal heuristic is fixed, the variants of MSVR-TG with time-weight radio greedy insertion (TWGIs) and hybrid time-weight radio greedy insertion (HTWGIs) heuristics have higher scores, which implies

Table 10: Results of the comparison among the variants of MSVR-TG with different combinations of removal and insertion heuristics in LNS.

Algorithm	**RR-RIs**	**RR-TGIs**	**RR-TWGIs**	**RR-HTWGIs**
Score	0	166	**208**	191
Algorithm	**AR-RIs**	**AR-TGIs**	**AR-TWGIs**	**AR-HTWGIs**
Score	2	118	150	163
Algorithm	**TGR-RIs**	**TGR-TGIs**	**TGR-TWGIs**	**TGR-HTWGIs**
Score	23	22	36	38
Algorithm	**TWGR-RIs**	**TWGR-TGIs**	**TWGR-TWGIs**	**TWGR-HTWGIs**
Score	20	21	40	30

The highest score is indicated in bold.

that problem knowledge is conducive to the repair of the solution. In subsequent simulations, MSVR-TG adopts the combination of random removal heuristic (RR) and time-weight radio greedy insertion heuristic (TWGIs).

5.5. Performance test of the route selection rule

Next, the performance of the route selection rule is tested. The comparison algorithm named MSVR-TG-random is the same as MSVR-TG except for the route selection rule. In MSVR-TG-random, the routes involved in LNS are selected in a random way. That is, $\alpha_1 = 1, \alpha_2 = 1$ in Algorithm 6.

The performances of MSVR-TG and MSVR-TG-random are tested based on the cases in case set B. Each algorithm runs independently for 20 times, and the Wilcoxon rank-sum test is utilized with a significance level of 0.05. As show in Table 11, the proposed route selection rule can improve the solution of 20% cases. Cases with improved solutions are accompanied by a larger number of vehicles, including B-11, B-12, and B-15. To verify the conclusion, the above test is carried out based on the cases in case set P-II. It is found that the solutions are improved in 60% of cases, as shown in Table 11. It is proved that the proposed route selection rule has obvious advantages under the condition of a large number of vehicles. That is to say, when the number of the route combinations that can be selected is large, the proposed route selection rule can avoid blind exploration to a certain extent, and plays an important role in improving the solution.

5.6. Performance test of MSVR-TG

To verify the performance of MSVR-TG, it is compared with the other three algorithms, including EMRG-HA, CVRP-FA and HPSO. Each algo-

Table 11: Comparison between MSVR-TG-random and MSVR-TG.

Case set	MSVR-TG-random	MSVR-TG	Percentage of cases improved
B	0	3	20%
P-II	0	6	60%

rithm runs 20 times independently on each case.

Firstly, we give a brief description of the comparison algorithms. Parameters of each comparison algorithm are set according to the original text, as shown in Table 12.

Table 12: Parameter settings in comparison algorithms.

Algorithm	Parameter	Discription	Setting
EMRG-HA	$G_{\max}$	The maximum number of iterations	30
	δ	Ratio for controlling the number of groupings	0.1
	n_{group}	The maximum number of groups	k
	$G_{\text{max\_EMRG}}$	The maximum number of interations in EMRG	100
	$N_{\text{p\_EMRG}}$	The population size of EMRG	50
	$G_{\text{max\_TS\_LS}}$	The maximum number of interations in tabu search-based local search	$200\sqrt{NC}$
	t	Execution number of local search in one interation	$15 + random[0, 10]$
	l_{tube}	The tabu tenure	8
CVRP-FA	P-S	Population size	110
	MI	Maximum number of iteration	1000
	C-R	Crossover rate	0.95
	M-R	Mutation rate	0.1
HPSO	ω	Inertia coefficient	0.7
	c_1	Cognitive coefficient	2
	c_2	Social coefficient	2
	α_1, α_2	Independent random numbers	0.5
	K	Total number of the particles	$m/4$

[1] k is the number of routes in the solution adopted for route group.
[2] NC is the number of customers in the group to be optimized.
[3] m indicates the total number of customers.

- EMRG-HA (Xiao et al., 2021)

EMRG-HA groups routes through a multi-objective evolutionary algorithm, and the quality of routes in the selected group is improved via a local search method. Since the EMRG-HA does not consider the energy constraint, it is slightly modified during testing.

- CVRP-FA (Altabeeb et al., 2019)

The hybrid firefly algorithm for CVRPs (CVRP-FA) integrates the traditional firefly algorithm (FA) with two types of local search and genetic operators. Similar to EMRG-HA, energy constraint is introduced to it.

- HPSO (Islam et al., 2021)

The hybrid metaheuristic (HPSO) combines the particle swarm optimization (PSO) and variable neighborhood search (VNS) for the clustered vehicle routing problem. Similar to EMRG-HA, energy constraint is introduced to it.

5.6.1. Comparison of results on case set A

To verify the validity of the proposed algorithm, the results obtained by MSVR-TG are compared with those generated by the comparison algorithms on case set A, as shown in Table 7. The optimal solution of each case calculated by CPLEX is viewed as the benchmark.

As can be seen from Table 13, all the solutions obtained by MSVR-TG are superior to those from the comparison algorithms. MSVR-TG obtains the optimal solutions in 8 out of 10 cases, and the gaps of A-8 and A-10 are 0.07% and 0.26%, respectively. In terms of the running time, MSVR-TG also has an advantage over the comparison algorithms and CPLEX. Fig. 18 shows the average of the task execution rates of each case obtained by each algorithm in 20 runs.

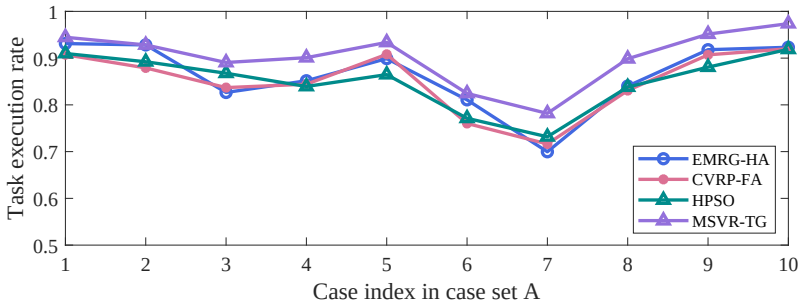


Figure 18: Comparison among task execution rates gained by algorithms for case set A.

5.6.2. Comparison of results on case set B

Next, the proposed algorithm is tested based on case set B in Table 7, and the results are shown in Table 14. It shows that the results obtained

Table 13: Comparison on results of MSVR-TG with those obtained via other algorithms for case set A.

Case	EMRG-HA		CVRP-FA		HPSO		MSVR-TG		CPLEX	
	f	Time/s	f	Time/s	f	Time/s	f	Time/s	f	Time/s
A-1	165.61±1.22	38.56±3.08	161.36±7.36	545.54±21.17	161.84±4.28	27.28±2.96	**168.01±0.00**	9.74±1.69	168.01	335.09
A-2	**236.53±0.00**	48.45±3.43	224.00±11.62	637.06±13.58	227.35±3.61	32.74±1.27	**236.53±0.00**	11.56±0.99	236.53	225.52
A-3	163.87±7.52	38.70±2.92	165.94±7.77	548.37±30.65	172.02±2.65	25.57±1.17	**176.58±0.00**	7.99±0.38	176.58	414.42
A-4	128.30±0.00	30.60±2.46	127.19±1.97	481.95±17.12	126.46±2.41	20.00±0.83	**135.74±0.00**	5.95±0.15	135.74	14.38
A-5	157.44±6.71	35.34±2.74	159.04±5.18	535.18±23.57	151.44±1.11	25.74±1.53	**163.51±0.00**	7.30±0.43	163.51	563.71
A-6	205.78±2.63	48.00±3.55	192.96±9.52	619.31±24.68	195.76±6.39	31.90±1.33	**209.27±0.00**	10.31±0.20	209.27	250.65
A-7	158.42±7.93	48.01±3.87	162.20±8.24	613.57±26.40	165.68±4.69	32.53±1.79	**176.99±0.00**	9.64±0.31	176.99	1068.80
A-8	215.07±5.63	53.59±3.85	212.64±6.52	642.40±27.20	214.53±4.50	37.91±1.77	230.03 ± 0.72	13.74±0.76	230.18*	7200.00
A-9	214.76±6.63	47.74±3.75	212.18±7.23	611.92±27.44	206.04±3.88	32.45±0.87	**222.56±0.00**	11.79±0.59	222.56	716.17
A-10	223.51±1.40	62.54±5.44	222.76±2.42	676.24±32.68	222.46±1.70	36.08±1.41	235.76 ± 1.76	8.03 ± 0.40	236.37*	7200.00
Score	9	–	3	–	2	–	**30**	–	–	–

[1] f represents the mean and standard deviation of solutions in 20 runs.
[2] Time/s represents the mean and standard deviation of running times in 20 runs.
[3] * denotes the approximate solution calculated by CPLEX within 2 hours.
[4] – indicates the results involved do not participate in the Wilcoxon rank-sum test.
[5] The optimal solutions obtained by MSVR-TG are indicated in bold.

via the proposed algorithm are better than those of CPLEX in 14 out of 15 cases, and the gap of case B-11 is 0.71%. Compared with the state-of-art algorithms, the proposed algorithm has the highest score, and the running time of it outperforms those of the comparison algorithms. Fig. 19 shows the average of the task execution rates of each case obtained by each algorithm in 20 runs.

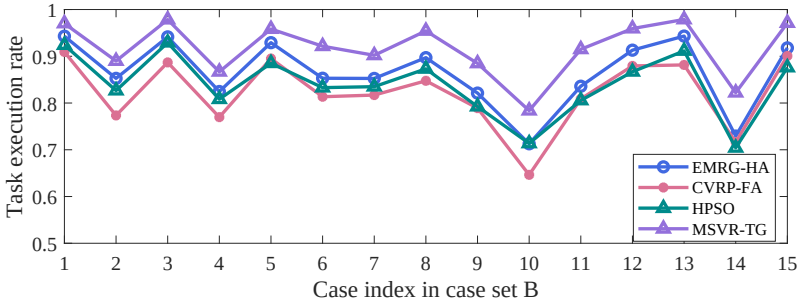


Figure 19: Comparison among task execution rates gained by algorithms for case set B.

5.6.3. Comparison of results on case sets P-I and P-II

Finally, the influence of the depot on the performance of the algorithm is tested on case sets P-I and P-II with different locations of the depot.

As can be seen in Table 15, the location of the depot has a great influence on the total weight of the executed tasks. The results obtained by the proposed algorithm are superior to those of the comparison algorithms. It indicates that K-mean-ad can obtain effective grouping under different depot location conditions. Except for P-I-3, the results of the proposed algorithm are better than those obtained by CPLEX, and the gap is 0.12% in P-I-3, which further proves the effectiveness of the proposed algorithm. Figs. 20-21 show the average of the task execution rates of each case in sets P-I and P-II obtained by each algorithm in 20 runs, respectively.

By comprehensively analyzing the results in Tables 13-15, the following conclusions can be drawn.

1. **Divide and conquer is beneficial to improve the efficiency of problem solving, and can obtain high-quality solutions.** Since global search is adopted in both CVRP-FA and HPSO, it is difficult to obtain promising solutions within a specified time for large-scale

Table 14: Comparison on results of MSVR-TG with those obtained via other algorithms for case set B.

Case	EMRG-HA		CVRP-FA		HPSO		MSVR-TG		$f_{\text{timelimit}}$
	f	Time/s	f	Time/s	f	Time/s	f	Time/s	
B-1	375.08±1.31	107.77±9.36	361.68±8.44	891.10±36.43	367.54±7.21	65.66±3.22	**385.89±1.55**	15.32±0.46	381.57
B-2	548.98±13.11	170.11±16.50	497.96±19.32	1141.60±52.21	532.17±10.50	109.50±5.65	**572.93±0.83**	30.00±1.27	562.87
B-3	848.77±13.25	370.34±30.02	799.54±15.62	1591.90±121.92	837.64±8.46	233.52±11.85	**882.10±3.01**	36.16±1.80	838.68
B-4	799.63±12.95	415.80±35.81	746.48±12.62	1925.30±107.00	783.75±10.91	232.80±8.10	**840.57±1.15**	57.30±4.96	814.28
B-5	382.41±4.38	129.37±8.61	368.40±14.78	994.06±47.57	364.33±7.70	73.32±2.93	**423.17±0.27**	11.49±0.30	415.05
B-6	466.34±16.56	164.79±12.12	444.73±19.95	1107.60±56.33	455.37±10.24	101.80±3.64	**503.82±0.53**	16.08±0.79	498.72
B-7	420.68±7.74	176.12±13.15	403.12±13.80	1045.00±76.40	412.00±11.85	113.67±4.99	**445.09±1.29**	12.58±0.43	439.54
B-8	609.06±12.47	264.63±18.97	575.44±12.84	1438.90±58.09	592.80±13.11	169.39±7.83	**647.94±1.21**	22.23±0.92	627.43
B-9	557.58±12.60	268.75±22.23	535.74±17.65	1324.80±93.83	537.98±9.87	173.43±8.59	**600.72±3.09**	22.02±1.37	592.66
B-10	1354.60±31.47	1079.80±94.84	1228.30±41.16	3175.90±156.89	1356.10±19.59	680.18±32.01	**1489.30±4.24**	108.02±5.32	1454.49
B-11	513.04±19.23	237.14±19.98	496.38±17.85	1304.80±67.69	494.39±17.07	152.66±8.53	561.52±3.34	13.57±0.74	**565.50**
B-12	912.88±14.87	465.46±36.54	879.21±23.29	1899.60±155.29	866.66±22.37	356.62±9.98	**959.07±8.76**	26.19±1.07	896.02
B-13	1379.10±14.42	835.79±65.62	1288.90±32.00	2529.00±173.46	1332.80±14.49	572.76±24.73	**1431.10±3.78**	63.66±2.00	1347.56
B-14	877.67±23.60	597.14±47.36	862.01±23.59	2223.30±172.44	847.47±22.53	504.78±19.20	**989.44±3.05**	27.60±1.20	944.87
B-15	1388.80±14.12	870.24±71.53	1362.70±33.66	3053.20±141.66	1325.30±18.58	764.26±32.40	**1469.30±6.42**	41.99±1.95	1306.99
Score	29	–	1	–	8	–	**45**	–	–

[1] f represents the mean and standard deviation of solutions in 20 runs.
[2] Time/s represents the mean and standard deviation of running times in 20 runs.
[3] $f_{\text{timelimit}}$ is the best solution calculated by CPLEX in 2 hours.
[4] – indicates the results involved do not participate in the Wilcoxon rank-sum test.
[5] The best solutions are indicated in bold.

Table 15: Comparison on results of MSVR-TG with those obtained via other algorithms for case sets P-I and P-II.

Case	**EMRG-HA**		**CVRP-FA**		**HPSO**		**MSVR-TG**		$f_{\text{timelimit}}$
	f	Time/s	f	Time/s	f	Time/s	f	Time/s	
P-I-1	561.91±9.07	206.70±16.01	527.47±18.31	1194.30±57.02	541.74±12.24	122.26±5.90	**577.45±2.83**	21.45±0.54	565.10
P-I-2	531.11±12.79	207.53±15.86	509.08±14.99	1259.90±62.73	514.59±7.79	123.84±4.90	**563.73±2.31**	21.29±0.97	550.20
P-I-3	533.69±9.59	211.62±17.67	508.16±14.95	1258.90±51.59	523.42±12.02	128.02±6.15	559.47±2.35	20.03±0.49	**560.37**
P-I-4	513.69±13.18	207.55±14.65	498.00±16.33	1135.50±76.36	494.46±8.79	127.22±4.18	**544.14±2.49**	21.72±0.87	538.10
P-I-5	532.42±15.25	206.93±16.86	510.31±15.94	1239.80±51.79	522.32±12.10	126.25±5.37	**566.36±3.79**	22.79±0.63	558.22
P-I-6	537.42±15.02	209.67±17.52	516.63±14.24	1222.60±45.22	523.12±15.69	121.86±4.93	**563.75±2.92**	21.91±0.96	549.24
P-I-7	556.45±15.37	208.44±16.50	524.85±12.07	1184.10±64.42	545.92±8.89	125.44±6.37	**577.00±2.07**	21.71±0.76	565.10
P-I-8	553.82±8.44	211.04±16.17	521.42±19.13	1178.00±38.39	539.32±10.97	123.94±5.98	**573.59±2.06**	22.07±0.92	571.87
P-I-9	520.31±12.05	211.68±16.95	497.24±15.63	1178.60±48.91	505.92±11.59	125.80±5.96	**557.25±2.25**	22.46±0.91	556.96
P-I-10	510.29±10.59	210.45±18.96	487.58±16.36	1219.60±56.11	489.79±11.16	127.21±5.51	**545.88±1.40**	21.35±0.64	528.40
P-II-1	1114.60±15.17	637.93±59.52	1079.50±33.04	2136.60±162.29	1069.40±21.29	470.44±19.26	**1172.50±3.57**	36.16±1.51	1066.28
P-II-2	969.37±22.11	620.95±34.28	947.26±36.72	2253.90±185.73	931.14±15.77	472.95±20.92	**1076.90±6.98**	34.39±1.53	1004.79
P-II-3	1093.60±21.79	633.53±52.88	1055.70±31.69	2221.60±187.89	1043.50±17.48	467.83±26.35	**1159.40±5.92**	34.75±1.37	1118.07
P-II-4	941.23±22.98	634.08±31.43	932.49±31.26	2362.30±97.09	902.36±27.58	476.52±22.34	**1052.90±6.76**	36.44±1.61	966.99
P-II-5	1025.3±22.65	613.54±55.70	993.47±31.40	2249.10±200.07	973.48±27.75	474.36±23.34	**1117.40±5.62**	34.73±0.92	991.41
P-II-6	1010.10±24.93	622.19±52.61	979.95±22.45	2321.00±142.91	961.36±23.16	479.06±21.34	**1097.3±8.44**	33.56±1.00	1000.99
P-II-7	1038.60±22.13	612.78±24.03	1000.60±34.46	2704.50±123.08	990.36±25.64	468.75±21.87	**1113.40±8.07**	34.62±1.08	1037.93
P-II-8	981.39±20.55	639.86±19.99	955.47±27.31	2375.20±88.34	930.30±21.77	479.05±11.44	**1074.1±14.07**	34.56±1.15	1013.50
P-II-9	979.77±22.15	666.19±38.48	954.65±28.39	2261.00±191.35	931.02±16.95	477.82±23.62	**1068.9±6.38**	37.92±0.89	1013.01
P-II-10	995.27±26.43	639.82±50.98	981.47±21.99	2374.90±126.11	937.17±23.91	477.83±21.81	**1090.40±8.25**	36.60±2.02	1024.43
Score	38	–	6	–	5	–	**60**	–	–

[1] f represents the mean and standard deviation of solutions in 20 runs.
[2] Time/s represents the mean and standard deviation of running times in 20 runs.
[3] $f_{\text{timelimit}}$ is the best solution calculated by CPLEX in 2 hours.
[4] – indicates the results involved do not participate in the Wilcoxon rank-sum test.
[5] The best solutions are indicated in bold.

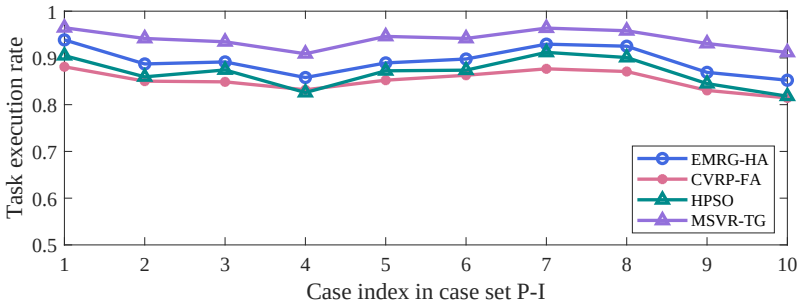


Figure 20: Comparison among task execution rates gained by algorithms for case set P-I.

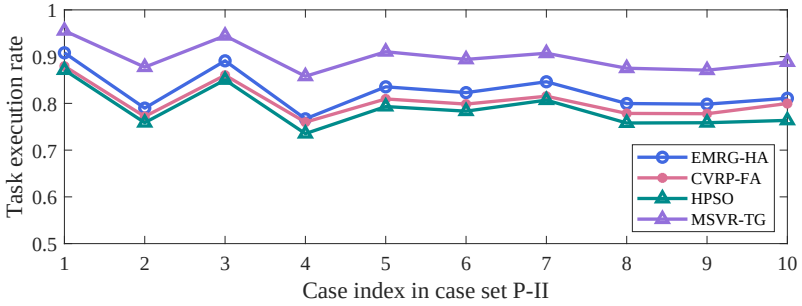


Figure 21: Comparison among task execution rates gained by algorithms for case set P-II.

complex problems. The results of EMRG-HA are superior to those of CVRP-FA and HPSO, mainly due to the introduction of route grouping and local search.

2. **Appropriate trade-off among multi-attribute tasks under the energy constraint is critical.** It is demonstrated that although EMRG-HA tries to balance the makespan of each vehicle as much as possible during route optimization, its performance is inferior to MSVR-TG due to the lack of the task trade-off strategy. Fig. 22 shows the route planning results of MSVR-TG for cases B-3, B-13, P-I-3 and P-II-5.

5.7. Performance test of the main modules in MSVR-TG

In this section, the validity of each main module in MSVR-TG is tested, including the functional modules in the three stages, and their combined

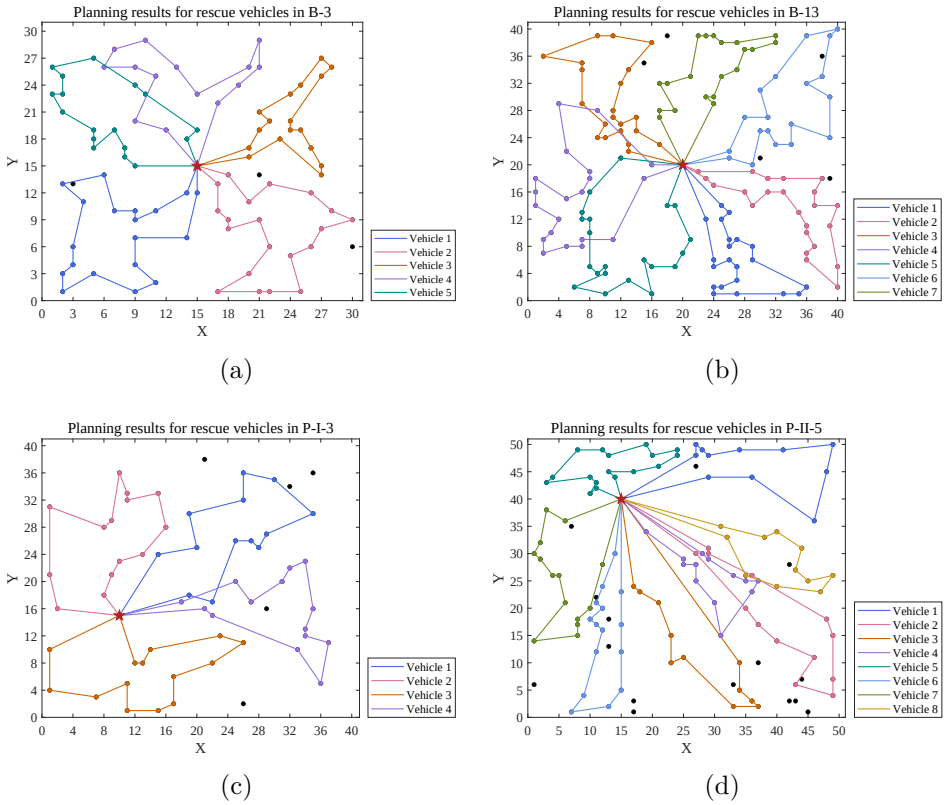


Figure 22: Route planning results of MSVR-TG.

modules. Firstly, the variants of MSVR-TG with module replacement are introduced, and the composition of each algorithm is shown in Table 16.

- Random Sampling (RS)

RS is chosen as a comparison algorithm for many problems, such as sensor-weapon-target assignment problem (Xin et al., 2019), multi-agent coalition formation problem (Guo et al., 2020). In this paper, a random planning rule, which randomly adds tasks to the route within energy constraint, is adopted in RS. The best solution is further improved through iteration.

- MSVR

Table 16: Functional modules of MSVR-TG and its variants in three stages.

Algorithm	Task grouping	Sequence planning	Route adjusting
RS	–	random planning	–
MSVR	–	random planning	LNS
MSVR-TG-I	K-means	PSGA	LNS
MSVR-TG-II	K-means-ad	random planning	LNS
MSVR-TG-III	K-means-ad	PSGA	–
MSVR-TG	K-means-ad	PSGA	LNS

– indicates the stage is not involved in the algorithm.

To verify the effectiveness of the combination of K-means-ad and PSGA in MSVR-TG, MSVR replaces it with the random sequence planning, in which the task execution sequence of each vehicle is optimized without problem decomposition. Routes satisfying the energy constraint are constructed by randomly selecting the task from the candidate set.

- MSVR-TG-I

To verify the effectiveness of K-means-ad in MSVR-TG, MSVR-TG-I replaces it with traditional K-means, which groups tasks according to the Euclidean distance.

- MSVR-TG-II

To verify the effectiveness of PSGA in MSVR-TG, MSVR-TG-II replaces PSGA with random planning in each group, which is similar to the solution construction in MSVR.

- MSVR-TG-III

To verify the effectiveness of LNS in MSVR-TG, only the first two functional modules of MSVR-TG are retained in MSVR-TG-III.

The results of MSVR-TG for cases in case set B are compared with those obtained via the above variants. The Wilcoxon rank-sum test with a significance level of 0.05 is utilized to analyze data statistically, as shown in Table 17.

Statistical results show that the functional module in each stage plays an important role in improving the performance of the proposed algorithm. The specific analysis is as follows.

Table 17: Performance test of the main modules in MSVR-TG.

Algorithm	**RS**	**MSVR**	**MSVR-TG-I**
Score	0	25	42
Algorithm	**MSVR-TG-II**	**MSVR-TG-III**	**MSVR-TG**
Score	26	17	**71**

1. Compared with the traditional K-means, K-mean-ad can effectively improve the quality of the grouping result, thus facilitating the exploration in the latter two stages. At the same time, the initial centers obtained based on the angular density can better deal with the instability problem of the result of traditional K-means caused by the random selection of initial cluster centers.

2. Comparing the scores of MSVR-TG with MSVR and MSVR-TG-II, it shows that a high-quality planning result is crucial for the route adjusting stage, which can effectively enhance the search efficiency and improve the quality of the solution. The population initialization rules and the genetic operators utilized in PSGA can effectively balance the exploration and exploitation, providing promising results and accelerating convergence. For experimental verification, please refer to Section 5.3.

3. Tasks within the same group may be executed by different vehicles due to the diversity of task attributes and the energy constraint. A single clustering may not result in optimal grouping, so it is important to adjust it based on the makespan and the total weight of each route.

6. Conclusion and discussion

VRPEC is a complex combinatorial optimization problem, for which MSVR-TG is proposed. In the task grouping stage, it is demonstrated that the suggested K-means-ad can effectively handle task grouping problems in different situations. It can also address the instability of the result of K-means caused by the random selection of initial cluster centers. In the sequence planning stage, the PSGA is introduced combined with the problem knowledge. The population initialization rules and the route improvement strategy designed in PSGA can effectively improve the exploration

and exploitation capabilities of the algorithm, thereby providing a promising planning result. In the route adjusting stage, the performances of different combinations of removal heuristic and insertion heuristic are compared. Results show that the combination of random removal heuristic (RR) and time-weight radio greedy insertion heuristic (TWGIs) performs best. The effectiveness of the developed route selection strategy has also been verified. What's more, the validity of the functional module utilized in each stage is proved via simulations.

Through extensive cases, it proves that MSVR-TG is effective to solve VRPEC with a satisfactory performance, and significantly superior over the comparison algorithms in terms of the total weight of the executed tasks and the running time. In 8 out of 10 cases in case set A, MSVR-TG obtains the optimal solutions, and has an absolute advantage in running time compared to CPLEX. In 33 out of 35 cases in case sets B, P-I and P-II, the solutions obtained by MSVR-TG are better than those optimized by CPLEX within 2 hours. Compared with the comparison algorithms, the solutions obtained by MSVR-TG are all superior.

In future work, we will conduct an in-depth study on the following contents to effectively deal with the vehicle routing problem in the emergency rescue. 1) Route planning with sequential logic. 2) The impact of vehicle load on energy consumption. 3) Route planning under the condition of tasks with split servise. 4) Resource scheduling and route adjustment on the occurance of new tasks.

Declaration of Competing Interest

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

Credit authorship contribution statement

Lei Jiao: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Zhihong Peng:** Funding acquisition, Project administration, Supervision, Writing - review & editing. **Lele Xi:** Software, Supervision, Writing - review & editing. **Miao Guo:** Conceptualization, Formal analysis, Methodology. **Shuxin Ding:** Conceptualization, Formal analysis, Methodology, Writing -

review & editing. **Yue Wei:** Investigation, Supervision, Writing - review & editing.

Acknowledgements

We thank the editors and anonymous reviewers for their helpful comments and suggestions on improving the presentation of this paper. This work was supported in part by the National Natural Science Foundation of China under Grants U2013602, U1613225, 62088101, 62025301 and 61720106011, and the Shanghai Municipal Commission of Science and Technology Project (19511132101).

References

Ahmad, A., & Khan, S. S. (2021). initkmix-a novel initial partition generation algorithm for clustering mixed data using k-means-based clustering. *Expert Systems with Applications*, *167*, 114149. doi:https://doi.org/10.1016/j.eswa.2020.114149.

Akpinar, S. (2016). Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Systems with Applications*, *61*, 28–38. doi:https://doi.org/10.1016/j.eswa.2016.05.023.

Akpunar, Ö. Ş., & Akpinar, Ş. (2021). A hybrid adaptive large neighbourhood search algorithm for the capacitated location routing problem. *Expert Systems with Applications*, *168*, 114304. doi:https://doi.org/10.1016/j.eswa.2020.114304.

Altabeeb, A. M., Mohsen, A. M., Abualigah, L., & Ghallab, A. (2021). Solving capacitated vehicle routing problem using cooperative firefly algorithm. *Applied Soft Computing*, *108*, 107403. doi:https://doi.org/10.1016/j.asoc.2021.107403.

Altabeeb, A. M., Mohsen, A. M., & Ghallab, A. (2019). An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Applied Soft Computing*, *84*, 105728. doi:https://doi.org/10.1016/j.asoc.2019.105728.

Azad, A. S., Islam, M., & Chakraborty, S. (2017). A heuristic initialized stochastic memetic algorithm for mdpvrp with interdependent depot operations. *IEEE Transactions on Cybernetics*, *47*, 4302–4315. doi:https://doi.org/10.1109/TCYB.2016.2607220.

Chang, T., Kong, D., Hao, N., Xu, K., & Yang, G. (2018). Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization. *Applied Soft Computing*, *70*, 845–863. doi:https://doi.org/10.1016/j.asoc.2018.06.014.

Chen, J., Du, C., Lu, X., & Chen, K. (2019). Multi-region coverage path planning for heterogeneous unmanned aerial vehicles systems. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)* (pp. 356–3565). IEEE. doi:https://doi.org/10.1109/SOSE.2019.00060.

Chen, J., Sun, J., & Wang, G. (2022). From unmanned systems to autonomous intelligent systems. *Engineering*, *12*, 16–19. doi:https://doi.org/10.1016/j.eng.2021.10.007.

Daglayan, H., & Karakaya, M. (2016). The impact of crossover and mutation operators on a ga solution for the capacitated vehicle routing problem. *Universal Journal of Engineering Science*, *4*, 39–44. doi:https://doi.org/10.13189/ujes.2016.040301.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, *6*, 80–91. doi:https://doi.org/10.1287/mnsc.6.1.80.

Ding, S., Chen, C., Xin, B., & Pardalos, P. M. (2018). A bi-objective load balancing model in a distributed simulation system using nsga-ii and mopso approaches. *Applied soft computing*, *63*, 249–267. doi:https://doi.org/10.1016/j.asoc.2017.09.012.

Fang, Z., Tu, W., Li, Q., Shaw, S.-L., Chen, S., & Chen, B. Y. (2013). A voronoi neighborhood-based search heuristic for distance/capacity constrained very large vehicle routing problems. *International Journal of Geographical Information Science*, *27*, 741–764. doi:https://doi.org/10.1080/13658816.2012.707319.

Fg, A., Gdcs, B., Jmn, A., Cldc, A., & Apl, A. (2022). A multi-agent system for solving the dynamic capacitated vehicle routing problem with stochastic customers using trajectory data mining. *Expert Systems with Applications*, *195*, 116602. doi:https://doi.org/10.1016/j.eswa.2022.116602.

Guo, M., Xin, B., Chen, J., & Wang, Y. (2020). Multi-agent coalition formation by an efficient genetic algorithm with heuristic initialization and repair strategy. *Swarm and Evolutionary Computation*, *55*, 100686. doi:https://doi.org/10.1016/j.swevo.2020.100686.

Islam, M. A., Gajpal, Y., & ElMekkawy, T. Y. (2021). Hybrid particle swarm optimization algorithm for solving the clustered vehicle routing problem. *Applied Soft Computing*, *110*, 107655. doi:https://doi.org/10.1016/j.asoc.2021.107655.

Jia, Y.-H., Mei, Y., & Zhang, M. (2021). A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem. *IEEE Transactions on Cybernetics*, (pp. 1–14). doi:https://doi.org/10.1109/TCYB.2021.3069942.

Jia, Y.-H., Mei, Y., & Zhang, M. (2022). Confidence-based ant colony optimization for capacitated electric vehicle routing problem with comparison of different encoding schemes. *IEEE Transactions on Evolutionary Computation*, (pp. 1–15). doi:https://doi.org/10.1109/TEVC.2022.3144142.

Jiao, L., Peng, Z., Xi, L., Ding, S., & Cui, J. (2022). Multi-agent coverage path planning via proximity interaction and cooperation. *IEEE Sensors Journal*, *22*, 6196–6207. doi:https://doi.org/10.1109/JSEN.2022.3150098.

Kancharla, S. R., & Ramadurai, G. (2018). An adaptive large neighborhood search approach for electric vehicle routing with load-dependent energy consumption. *Transportation in Developing Economies*, *4*, 1–9.

Karakatič, S. (2021). Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. *Expert Systems with Applications*, *164*, 114039. doi:https://doi.org/10.1016/j.eswa.2020.114039.

Karakatič, S., & Podgorelec, V. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, *27*, 519–532. doi:https://doi.org/10.1016/j.asoc.2014.11.005.

Kucukoglu, I., Dewil, R., & Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, *161*, 107650. doi:https://doi.org/10.1016/j.cie.2021.107650.

Lan, W., Ye, Z., Ruan, P., Liu, J., Yang, P., & Yao, X. (2022). Region-focused memetic algorithms with smart initialisation for real-world large-scale waste collection problems. *IEEE Transactions on Evolutionary Computation*, *26*, 704–718. doi:https://doi.org/10.1109/TEVC.2021.3123960.

Li, J., Xiong, Y., She, J., & Wu, M. (2020). A path planning method for sweep coverage with multiple uavs. *IEEE Internet of Things Journal*, *7*, 8967–8978. doi:https://doi.org/10.1109/JIOT.2020.2999083.

Liu, B., Sheu, J.-B., Zhao, X., Chen, Y., & Zhang, W. (2020). Decision making on post-disaster rescue routing problems from the rescue efficiency perspective. *European Journal of Operational Research*, *286*, 321–335. doi:https://doi.org/10.1016/j.ejor.2020.03.017.

Liu, Q., Xu, P., Wu, Y., & Shen, T. (2022). A hybrid genetic algorithm for the electric vehicle routing problem with time windows. *Control Theory and Technology*, (pp. 279–286). doi:https://doi.org/10.1007/s11768-022-00091-1.

Lu, J., Chen, Y., Hao, J.-K., & He, R. (2020). The time-dependent electric vehicle routing problem: Model and solution. *Expert Systems with Applications*, *161*, 113593. doi:https://doi.org/10.1016/j.eswa.2020.113593.

Machado, A. M., Mauri, G. R., Boeres, M., & Rosa, R. (2021). A new hybrid matheuristic of grasp and vns based on constructive heuristics, set-covering and set-partitioning formulations applied to the capacitated vehicle routing problem. *Expert Systems with Applications*, *184*, 115556. doi:https://doi.org/10.1016/j.eswa.2021.115556.

Mei, Y., Li, X., & Yao, X. (2014). Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, *18*, 435–449. doi:https://doi.org/10.1109/TEVC.2013.2281503.

Nucamendi-Guillén, S., Padilla, A. G., Olivares-Benitez, E., & Moreno-Vega, J. M. (2021). The multi-depot open location routing problem with a heterogeneous fixed fleet. *Expert Systems with Applications*, *165*, 113846. doi:https://doi.org/10.1016/j.eswa.2020.113846.

Öztaş, T., & Tuş, A. (2022). A hybrid metaheuristic algorithm based on iterated local search for vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, *202*, 117401. doi:https://doi.org/10.1016/j.eswa.2022.117401.

Polat, O., Kalayci, C. B., Kulak, O., & Günther, H.-O. (2015). A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. *European Journal of Operational Research*, *242*, 369–382. doi:https://doi.org/10.1016/j.ejor.2014.10.010.

Praveen, V., Kousalya, K., & Kumar, K. P. (2016). A nearest centroid classifier based clustering algorithm for solving vehicle routing problem. In *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)* (pp. 414–419). IEEE. doi:https://doi.org/10.1109/AEEICB.2016.7538321.

Sai, Shao, Wei, Guan, Jun, & Bi (2018). Electric vehicle-routing problem with charging demands and energy consumption. *IET Intelligent Transport Systems*, *12*, 202–212. doi:https://doi.org/10.1049/iet-its.2017.0008.

Shahab, M. L., Utomo, D. B., & Irawan, M. I. (2016). Decomposing and solving capacitated vehicle routing problem (cvrp) using two-step genetic algorithm (tsga). *Journal of Theoretical & Applied Information Technology*, *87*.

Shang, Q., Huang, Y., Wang, Y., Li, M., & Feng, L. (2022). Solving vehicle routing problem by memetic search with evolutionary multitask-

ing. *Memetic Computing*, (p. 31–44). doi:https://doi.org/10.1007/s12293-021-00352-7.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming* (pp. 417–431). Springer. doi:https://doi.org/10.1007/3-540-49481-2_30.

Song, B. D., Park, H., & Park, K. (2022). Toward flexible and persistent uav service: Multi-period and multi-objective system design with task assignment for disaster management. *Expert Systems with Applications*, *206*, 117855. doi:https://doi.org/10.1016/j.eswa.2022.117855.

Sun, H., Wang, Y., Zhang, J., & Cao, W. (2021). A robust optimization model for location-transportation problem of disaster casualties with triage and uncertainty. *Expert Systems with Applications*, *175*, 114867. doi:https://doi.org/10.1016/j.eswa.2021.114867.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., & Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, *257*, 845–858. doi:https://doi.org/10.1016/j.ejor.2016.08.012.

Umbarkar, A. J., & Sheth, P. D. (2017). Crossover operators in genetic algorithms: a review. *ICTACT journal on soft computing*, *162*, 34–36. doi:https://doi.org/10.5120/ijca2017913370.

Wan, F., Guo, H., Li, J., Gu, M., Pan, W., & Ying, Y. (2021). A scheduling and planning method for geological disasters. *Applied Soft Computing*, *111*, 107712. doi:https://doi.org/10.1016/j.asoc.2021.107712.

Xiao, J., Zhang, T., Du, J., & Zhang, X. (2021). An evolutionary multiobjective route grouping-based heuristic algorithm for large-scale capacitated vehicle routing problems. *IEEE transactions on cybernetics*, *51*, 4173–4186. doi:https://doi.org/10.1109/TCYB.2019.2950626.

Xin, B., Wang, Y., & Chen, J. (2019). An efficient marginal-return-based constructive heuristic to solve the sensor–weapon–target assignment problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*, 2536–2547. doi:https://doi.org/10.1109/TSMC.2017.2784187.

Xu, X., Zhang, L., Trovati, M., Palmieri, F., Asimakopoulou, E., Johnny, O., & Bessis, N. (2021). Perms: An efficient rescue route planning system in disasters. *Applied Soft Computing*, *111*, 107667. doi:https://doi.org/10.1016/j.asoc.2021.107667.

Xu, Z., & Cai, Y. (2018). Variable neighborhood search for consistent vehicle routing problem. *Expert Systems with Applications*, *113*, 66–76. doi:https://doi.org/10.1016/j.eswa.2018.07.007.

Yu, X., Li, C., Zhao, W.-X., & Chen, H. (2020). A novel case adaptation method based on differential evolution algorithm for disaster emergency. *Applied Soft Computing*, *92*, 106306. doi:https://doi.org/10.1016/j.asoc.2020.106306.

Zachariadis, E. E., Nikolopoulou, A. I., Manousakis, E. G., Repoussis, P. P., & Tarantilis, C. D. (2022). The vehicle routing problem with capacitated cross-docking. *Expert Systems with Applications*, *196*, 116620. doi:https://doi.org/10.1016/j.eswa.2022.116620.

Zhang, J. (2017). An efficient density-based clustering algorithm for the capacitated vehicle routing problem. In *2017 international conference on computer network, electronic and automation (ICCNEA)* (pp. 465–469). IEEE. doi:https://doi.org/10.1109/ICCNEA.2017.96.

Zhang, J., Li, Y., & Yu, G. (2022a). Emergency relief network design under ambiguous demands: A distributionally robust optimization approach. *Expert Systems with Applications*, *208*, 118139. doi:https://doi.org/10.1016/j.eswa.2022.118139.

Zhang, Y., Mei, Y., Huang, S., Zheng, X., & Zhang, C. (2022b). A route clustering and search heuristic for large-scale multidepot-capacitated arc routing problem. *IEEE Transactions on Cybernetics*, *52*, 8286–8299. doi:https://doi.org/10.1109/TCYB.2020.3043265.

Zheng, Y.-J., Chen, S.-Y., & Ling, H.-F. (2015). Evolutionary optimization for disaster relief operations: A survey. *Applied Soft Computing*, *27*, 553–566. doi:https://doi.org/10.1016/j.asoc.2014.09.041.