symmetry

MDPI

Article

A Knowledge-Guided Multi-Objective Shuffled Frog Leaping Algorithm for Dynamic Multi-Depot Multi-Trip Vehicle Routing Problem

Yun Zhao [1], Xiaoning Shen [2,3,4,5,*] and Zhongpei Ge [2]

[1] School of Electrical Engineering, Nanjing Vocational University of Industry Technology, Nanjing 210023, China; 2013100792@niit.edu.cn
[2] School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China; 001987@nuist.edu.cn
[3] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing 210044, China
[4] Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China
[5] Jiangsu Engineering Research Center on Meteorological Energy Using and Control (C-MEIC), Nanjing 210044, China
* Correspondence: sxnystsyt@sina.com; Tel.: +86-25-13813835843

Abstract: Optimization algorithms have a wide range of applications in symmetry problems, such as graphs, networks, and pattern recognition. In this paper, a dynamic periodic multi-depot multi-trip vehicle routing model for scheduling test samples is constructed, which considers the differences in testing unit price and testing capacity of various agencies and introduces a cross-depot collaborative transport method. Both the cost and the testing time are minimized by determining the optimal sampling routes and testing agencies, subjecting to the constraints of vehicle capacity, number of vehicles, and delivery time. To solve the model, a knowledge-guided multi-objective shuffled frog leaping algorithm (KMOSFLA) is proposed. KMOSFLA adopts a convertible encoding mechanism to realize the diversified search in different search spaces. Three novel strategies are designed: the population initialization with historical information reuse, the leaping rule based on the greedy crossover and genetic recombination, and the objective-driven enhanced search. Systematic experimental studies are implemented. First, feasibility analyses of the model are carried out, where effectiveness of the cross-depot collaborative transport is validated and sensitivity analyses on two parameters (vehicle capacity and proportion of the third-party testing agencies) are performed. Then, the proposed algorithm KMOSFLA is compared with five state-of-the-art algorithms. Experimental results indicate that KMOSFLA can provide a set of non-dominated schedules with lower cost and shorter testing time in each scheduling period, which provides a reference for the dispatcher to make a final decision.

Keywords: vehicle routing problem; cross-depot collaborative transport; evolutionary multi-objective optimization; knowledge-guided; convertible encoding

check for updates

Citation: Zhao, Y.; Shen, X.; Ge, Z. A Knowledge-Guided Multi-Objective Shuffled Frog Leaping Algorithm for Dynamic Multi-Depot Multi-Trip Vehicle Routing Problem. *Symmetry* **2024**, *16*, 697. https://doi.org/10.3390/sym16060697

Academic Editor: Christos Volos

Received: 5 May 2024
Revised: 27 May 2024
Accepted: 29 May 2024
Published: 5 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of economic globalization, a growing number of people are pouring into cities. Large-scale utilities and convenient transportation make public health emergencies spread at an unprecedented speed. Regular sample testing is essential to prevent the spread of pandemics, which can locate the source of infection and cut off the transmission in time. As a crucial part of sample testing, a complete and rational transportation and delivery schedule can guarantee the safety of test samples and improve the testing efficiency. In recent years, more and more scholars have focused on resource scheduling in the background of public health emergencies, such as vaccine distribution [1], vaccination projects involving factory-in-a-box manufacturing [2], medical waste collection [3], and the

transfer of isolated persons [4]. Most of the existing work formulated these problems as search-based optimization problems [5] and adopted metaheuristic algorithms [6] to find the optimal solutions.

In the vehicle routing problem of test samples (TSVRP), multiple vehicles depart from different testing agencies (the departure depots). Both the cost and testing time are minimized by determining the optimal collection order of sampling sites (clients) and testing agencies (the returned depots) for each vehicle, meeting the constraints of vehicle capacity, number of available vehicles, and delivery time. TSVRP has three characteristics: (1) Periodic scheduling: Due to the large number of samples, the long time-span of transport and the strict requirements for delivery time [7], it is difficult to transfer all samples once, and multiple scheduling periods are needed to dispatch samples to the appropriate testing agencies in batches. (2) Multiple trips and multiple heterogeneous depots: To improve resource utilization, each vehicle can undertake more than one transfer task in a period, i.e., one vehicle is allowed to have multiple trips. Meanwhile, at each trip, each vehicle departs from the testing agency where it arrived in the last trip, and then delivers the samples collected in the current trip to an appropriate testing agency. Such agencies (depots) are heterogeneous for their different testing capabilities and unit prices. (3) Dynamic changes in the environment of the sample scheduling network: Influenced by the number of samples sent in the previous period, the number of testing agencies, the speed of testing, and the staff efficiency, each testing agency will have a backlog of samples for the next period to varying degrees. Therefore, the detectable capacity of each agency in different periods for additional samples is changed. In addition, the number of samples sampled by the same sampling sites varies from period to period. In summary, TSVRP is a class of the dynamic periodic multi-depot multi-trip vehicle routing problem.

Currently, it is common for the dispatcher to manually develop a fixed transfer route for each vehicle. However, fixed routes are difficult to adapt to the dynamically changing environment and cause the situation that the workloads of some testing agencies are over-saturated while others are over-empty, affecting the testing efficiency. When there is a surge in sample collection, the dispatcher has to adjust the original routes manually again. Therefore, it is urgent to study an efficient and automatic method for solving TSVRP.

The multi-depot vehicle routing problem (MDVRP) has been proved to be an NP-hard problem [8]. TSVRP is an extension to MDVRP, where multiple trips of each vehicle need to be further optimized and the environmental changes are considered. Thus, TSVRP is also an NP-hard problem. Exact algorithms such as branch-and-bound [9] and integer programming [10] can provide an exact solution to the problem, but the runtime increases exponentially with the problem size [11] and they are only applicable to small-scale problems. Metaheuristic algorithms [12] are search-based methods inspired by the social behavior of a biological population or the laws of natural phenomena. They search in a large scope of the decision space based on population and find the approximate global optimal solution in a short time, which is more suitable for solving NP-hard problems.

The shuffled frog leaping algorithm (SFLA) is a metaheuristic algorithm proposed by Eusuff et al. [13] in 2003. It simulates the foraging process of frogs in which they alter positions by exchanging information to leap to the most abundant food area. SFLA performs the local search by updating the worst individual in each memeplex and realizes the global information sharing by shuffling all memeplex individuals. Compared with traditional metaheuristic algorithms, SFLA has fewer parameters, a simpler structure, and a stronger global search ability, which shows more superiority in solving high-dimensional combinatorial optimization problems with multiple local optima [14]. SFLA has been widely applied to flow shop scheduling [15], currency exchange rate forecasting [16], water resources scheduling [17], etc. However, SFLA still has some problems with complex constrained multi-objective optimization problems. First, the learning object lacks diversity, which makes the population tend to assimilate. Second, the basic multi-objective SFLA only accepts the new individual that dominates the worst one of the memeplex during the local search, which might lose a lot of non-dominated solutions with good genes. To this

end, a knowledge-guided multi-objective shuffled frog leaping algorithm (KMOSFLA) is proposed as a dynamic scheduling method for solving TSVRP, which generates a schedule adapted to the current environment at each scheduling period.

Three difficulties exist when solving TSVRP: (a) The first is how to construct a rigorous mathematical model for TSVRP according to its unique characteristics. (b) TSVRP contains three strongly coupled subproblems: multi-trip determination (identifying the number of trips for each vehicle and the sampling sites served in each trip), delivery agency selection (choosing an appropriate destination for each trip, i.e., the testing agency to which the samples are delivered), and vehicle routing (optimizing the collection order of sampling sites in each trip). It is a challenge to optimize the three subproblems simultaneously. (c) The final difficulty is how to make a real-time response to the dynamically changing environment and provide an adapted schedule. To address the above difficulties, three contributions are made in this work: (i) By using a cross-depot collaborative transport method, a dynamic periodic multi-depot multi-trip vehicle routing model is constructed for scheduling test samples. (ii) For the strongly coupled decision space, a convertible form of individual encoding is designed. In this way, our algorithm can adopt distinct search mechanisms at different stages (a leaping rule for the vehicle routes, and an objective-driven enhanced search for selecting testing agencies). (iii) A periodic rescheduling method is used to react to environmental changes. A dynamic response mechanism is introduced to construct the initial population when facing the new environment, which provides a good starting point for the searching.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 establishes the mathematical model of TSVRP. Section 4 describes the proposed algorithm KMOSFLA for solving the constructed model. Section 5 details the experimental studies. Section 6 concludes the work and provides an outlook on future work.

2. Related Work

In this section, we review the existing literature for resource scheduling under public health emergencies and the multi-depot vehicle routing problem and summarize the existing work on the shuffled frog leaping algorithm.

2.1. Resource Scheduling Model under Public Health Emergencies

The outbreak of public health emergencies like the COVID-19 epidemic has brought severe challenges to healthcare systems worldwide. With the surge in the number of infections, medical agencies are experiencing a shortage of medical supplies, which affected the routine treatment of hospitals, and the social order in various countries has suffered an unprecedented impact [18]. To reduce the effects of public health emergencies on society, many scholars have studied resource scheduling under emergencies. In terms of vaccine distribution, Zheng et al. established a two-echelon distribution model considering the uncertainty of the demand for the COVID-19 vaccines [6]. The demand for vaccines transported to satellites is predicted in advance by using a fuzzy deep-learning model. Then, the route of vaccine delivery to the inoculation spots is optimized according to the actual needs at the inoculation spots. Manupati et al. established a multi-echelon cold chain model by dividing the distribution process into three phases, which considered the priority and the delivery cost of the COVID-19 vaccines [1]. In terms of medical supplies distribution, Goodarzian et al. designed a sustainable healthcare supply chain network based on the periods of production and the perishability of medicines [19]. Affected by the scarcity of mask resources, Pacheco and Laguna established a delivery model in which raw materials for mask production were provided to manufacturers through civil society organizations and newly produced masks were delivered to the medical facilities [20]. In terms of medical waste transportation, Eren and Tuzkaya introduced medical waste management (MWM) safety scores and developed a medical waste transportation model with the optimization objectives of transport distance and transport safety [3]. Tirkolaee et al. developed a model

to formulate the sustainable multi-trip location-routing problem with time windows for medical waste management in the COVID-19 pandemic [21].

The vehicle routing problem for test samples is studied in this work.

2.2. Method for Solving Multi-Depot Vehicle Routing Problem

The TSVRP studied in this work can be summarized as a class of MDVRP [22]. The study of MDVRP can be classified into two categories: static and dynamic scheduling. Static MDVRP assumes that all information about road conditions, customer locations, and their demands is known in advance and does not change as the delivery progresses. Research on static MDVRP has yielded some results. Jin et al. proposed a two-step procedure to address the problem of MDVRP [23]. In step one, the nearest assignment and average distance method are used to transform the MDVRP into multiple VRPs with a single depot. In step two, a genetic algorithm is adopted to solve the VRPs. Yesodha and Amudha solved MDVRP with time windows using an improved firefly algorithm [24], which assigned customers to the depots based on the inter depot method in the assignment task phase and applied the Clarke and Wright algorithm and the local search in the routing task phase. Wang et al. addressed the issue of collaboration across multiple time periods by formulating a two-echelon collaborative multi-depot multi-period vehicle routing problem, and they proposed a hybrid heuristic algorithm to solve the collaborative transportation problem across multiple periods [25]. Zhen et al. developed a hybrid particle swarm optimization algorithm which considered the last mile distribution characteristics of dispersive customer positions, large order volumes, small batches, and multiple repeated routes [26]. Cen et al. formulated a relaxation mixed integer linear programming model for solving a vehicle routing problem with cross-docking and three-dimensional loading constraints and designed a hybrid heuristic algorithm to solve the model [27].

Dynamic MDVRP can only obtain some of the information in advance. The original routes will be adjusted according to new information or environmental changes, such as new customer requests or demand changes. The introduction of dynamic characteristics makes MDVRP more practical. There are few studies on dynamic MDVRP. In terms of the uncertainty of customer demand, Xu et al. proposed a clustering approach based on the nearest distance to allocate all customers to the depots and optimized vehicle routes by using a hybrid ant colony optimization with mutation operation and local interchange [28]. Meanwhile, a real-time addition and optimization approach is designed to handle the new customer requests. Yu et al. used the nearest addition method to process the new customers occurring on the time slice to meet the dynamic real-time requirements and found the new route with the smallest distance increase compared with the original route [29]. In terms of distribution network uncertainty, Liu et al. introduced a planning strategy for planning pools to maximize the number of visiting customers considering extended service times and traffic congestion [30]. Seyyedhasani et al. proposed a method that allowed the dynamic replanning of routes involving dynamic environments in agricultural work and verified the update of vehicle routes in agricultural operation possibilities [31].

The constructed model TSVRP in this work differs from the existing MDVRP in the following ways: (i) It has the characteristics of dynamic periodic scheduling, taking into account the factors of multiple depots and multiple trips. (ii) It adopts a cross-depot collaborative transport method. The departure agency (the departure depot) and delivery agency (the returned depot) of each vehicle can be different, and the vehicle can flexibly choose the delivery agency according to the real-time testing capability of the testing agency. (iii) The attributes (testing capability and testing unit price) of the testing agency (depot) are considered. The depot in TSVRP is not only the location of vehicle departure and delivery but is also capable of testing samples (i.e., consuming the delivered samples). Samples beyond the current testing capacity of the testing agency will be backlogged for testing in the next period.

2.3. Improvement of Shuffled Frog Leaping Algorithm

The existing work improves SFLA in three aspects: population initialization, grouping, and local search. Regarding population initialization, Fang et al. proposed a multi-objective differential evolution chaos shuffled frog leaping algorithm for the shortcomings of easily falling into local minima and population premature convergence [32]. The algorithm initialized the population by chaos and replaced the local update of SFLA with the stochastic search strategy of the differential evolution algorithm. Yang et al. proposed a discrete SFLA based on a chaotic cloud model and used it to solve a nonlinear mixed-integer planning problem for the short-term economic operation of a power plant [33]. The algorithm introduced chaotic population initialization, set three subpopulations of leading frogs, following frogs, and mutating frogs, and designed a heuristic frog activation mechanism for an elite frog set based on the cloud model. Regarding grouping, He et al. divided the population into three memeplexes [14]. He used a global updating strategy, a local updating strategy, and a random updating strategy to perform local depth search for the three memeplexes, respectively. The information is exchanged periodically among these memeplexes, which further improves the convergence accuracy of the algorithm. Considering that the basic grouping method tends to make the algorithm fall into the local optimum, Lei and Guo applied binary tournament selection for population division and introduced the neighborhood search to update the individuals [34]. Regarding local search, Liu et al. improved the quality of individuals by introducing a specular reflection learning mechanism and a simulated annealing mechanism based on chaos and levy flight strategies, which allowed the population to search the entire search space more evenly [35,36]. Tang et al. used levy flight formulas to control the leaping step in the local search of SFLA and added an interaction learning rule in the global shuffling to improve the exploration ability and information exchange [37]. Sharma and Pant utilized the idea of opposition-based learning to generate solutions corresponding to the opposite positions of individuals and then selected elite individuals to participate in the search to improve the ability of the algorithm to jump out of the local optimum [38].

Currently, there are fewer studies on the multi-objective shuffled frog leaping algorithm. In multi-objective function optimization, Luo et al. divided the population by grid-based memeplex clustering analyses and used the relaxed dominated mechanism with the improved crowding distance to select the good individual [39]. Jie and Wei adopted the ε indicator to compare the merits among non-dominated individuals in the SFLA population and selected the global optimal individual [40]. In multi-objective combinatorial optimization problems, most existing work introduces the multi-objective processing framework of NSGA2 in SFLA, and the improvement strategies focus on how to design the update strategy of individuals for the characteristics of the applied problem. Yang et al. used the individuals in the elite solutions set and the optimal individuals to guide the updating of the worst individuals, and they updated the elite individual set using an evolutionary strategy based on the cloud model theory [33]. Lakshmi and Rao introduced a customized neighborhood search algorithm [41], an adaptive search factor, and a crossover operator, and they adaptively selected the search strategy based on the feedback on the convergence of the population. In order to avoid falling into local optima, Taher et al. designed a mutation-based SFLA to improve the diversity of frog population distribution by introducing the trial mutated vectors in the individual update process [42].

In this work, KMOSFLA is designed as a periodic rescheduling method, which differs from the existing work in the following ways: (i) A convertible encoding mechanism is designed. Through encoding conversion, the algorithm can track the valuable information of elite individuals for large-scale exploration, while implementing fine mining of the vehicle routes using objective knowledge, which improves the convergence accuracy of the algorithm. In contrast, the encoding of the existing MDVRP work is fixed, and it is difficult for the algorithm to carry out diversified search patterns. (ii) The criterion for SFLA to receive new individuals in multi-objective optimization is relaxed when updating the worst frog of a memeplex, with the aim of retaining the solution that is not dominated

by the worst frog. In contrast, the existing work only receives the new frog that dominates the worst frog of the memeplex, and the population in the later stage is difficult to update and easily falls into the local optimum. (iii) The individual updating method applicable to TSVRP is designed. A leaping rule based on greedy crossover and genetic recombination is proposed to update individuals, which extracts the effective knowledge of both optimal and poor individuals. In addition, an objective-driven enhanced search is designed to realize extreme value mining on different objectives of the algorithm.

3. Mathematical Modeling for TSVRP

In this section, a scheduling network for test samples is introduced, and a mathematical model for TSVRP is constructed.

3.1. The Scheduling Network for Test Samples

Due to the large population base, it is difficult for public testing agencies to undertake massive sample testing tasks in a short period of time. Therefore, a mode of cooperative testing between public and third-party testing agencies has been developed.

The scheduling network for test samples consists of discrete sampling sites and testing agencies. The number of samples waiting to be collected at each sampling site varies, as does the testing unit price and testing capacity of each testing agency. A complete sample scheduling process is shown in Figure 1, including sample collection, sample transport, and sample testing.

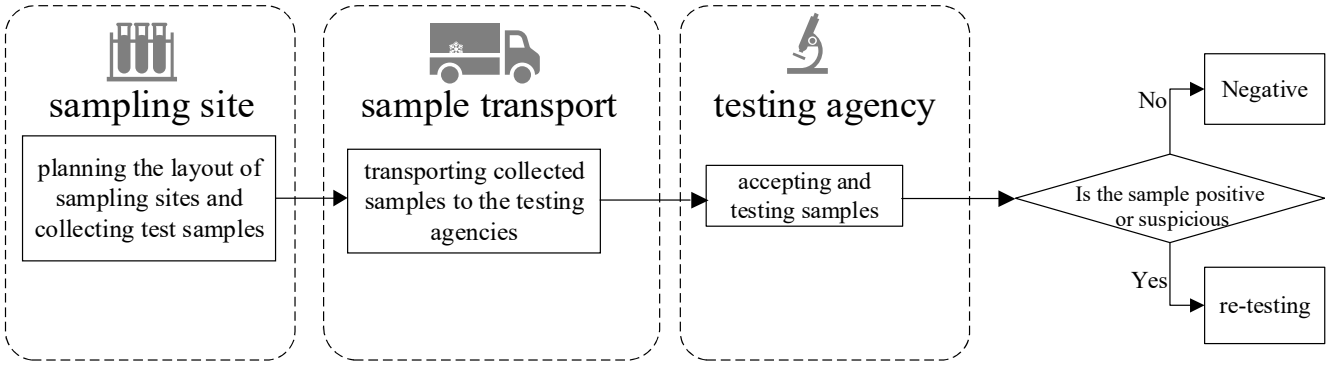


Figure 1. The flowchart of scheduling the test samples.

In this work, a cross-depot collaborative transport method is introduced, which allows for different departure agencies (the departure depot) and delivery agencies (the returned depot) for each vehicle. The transfer routes in Figure 2 illustrate the advantages of this method, where s1~s10 are the sampling sites, d1~d3 represent the testing agencies, $Trip_k^w$ denotes the wth trip of the kth vehicle, and the number above the routes represents the distance (km) between two nodes. Figure 2a shows the basic transportation method, where the departure agencies of each vehicle are consistent with the delivery agencies. Figure 2b expresses the cross-depot collaborative transport method. For example, after completing the sample collection at s4 and s2, the samples will be delivered to the testing agencies d2 and d3 instead of the departure agencies d1 and d2, respectively. Compared with the basic transport method, vehicle 1 and vehicle 2 travel a total of 13.8 km less, which significantly reduces the vehicle travel distance. In addition to saving on travel distance, the cross-depot collaborative transport method also allows for a more flexible allocation of collected samples to delivery agencies, taking the agencies' real-time testing capacity and testing unit price into account to balance the transportation cost and the testing time of the testing agencies dynamically. Ultimately, the government settles the testing and transportation costs of each agency after the dispatching is completed.

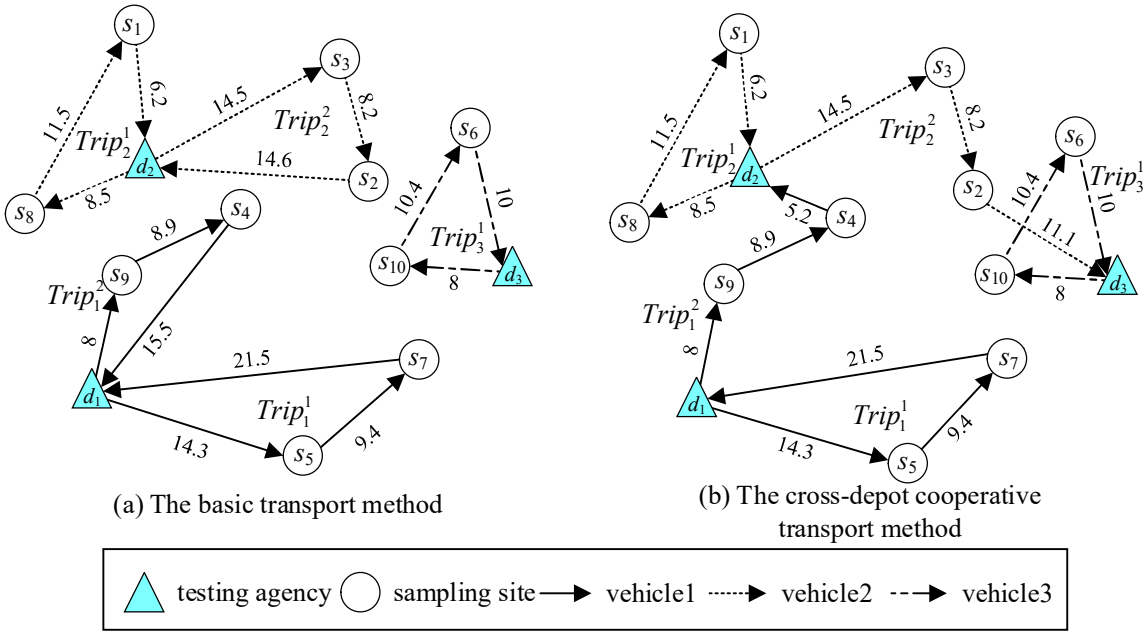


Figure 2. An illustration of the cross-depot collaborative transport.

3.2. Mathematical Modeling

In this work, considering the differences in testing capabilities and unit prices of testing agencies in the sample testing, we construct a dynamic periodic multi-depot multi-trip vehicle routing model for the TSVRP, which is an extension to the classical MDVRP [22].

3.2.1. Problem Statement

TSVRP considers the sample transport in one day, which can be described as follows. Let $G = (V, E)$ denote a complete directed graph where $V = \{S \cup D\}$ is the total nodes set and is composed of two subsets, $S = \{s_1, s_2, \ldots, s_n\}$ and $D = \{d_1, d_2, \ldots, d_m\}$, which comprise n sampling sites and m testing agencies separately. $E = \{(p_i, p_j) | p_i, p_j \in V, i \neq j\}$ represents the set of edges. Let t_l (l = 1, 2, …, $|TL|$) denote the scheduling point for periodic scheduling and TL be the set of scheduling points t_l (e.g., if the period is 2 h, the test time is from 8:00 to 16:00, and the sample transfer time is from 10:00 to 16:00, then TL = {10, 12, 14, 16}). TSVRP requires that at each scheduling point t_l, on the premise of meeting the latest sample collection moment and vehicle capacity constraints, several vehicles are selected, and appropriate transport routes are determined for them, so as to minimize the scheduling cost and the maximum testing time of the testing agency during the current scheduling period. Vehicles are allowed to deliver samples to any testing agency, and each vehicle can perform more than one trip in a scheduling period. It is assumed that each vehicle has the same capacity, average velocity, and fuel price and that each testing agency works 24 h per day. The notations are defined in Table 1.

Table 1. Notations of the TSVRP model.

Indexes and Sets			
u	Index of the sampling site.	K	The set of vehicles.
v	Index of the testing agency.	K_v (t_l)	The set of available vehicles at the testing agency d_v for the lth scheduling period (vacant vehicles in d_v and vehicles that deliver samples to d_v on the last trip of the l-1th scheduling period).
k	Index of the vehicle.		
w	Index of the trip.		
S	The set of sampling sites s_u.		
D	The set of testing agencies d_v.	$Trip_k$ (t_l)	The routes of vehicle k at the lth scheduling period, $Trip_k(t_l) = \{Trip_k^w(t_l) \mid w = 1, 2, \ldots, L_k(t_l)\}$, $Trip_k^w(t_l)$ denotes the wth trip of vehicle k, $L_k(t_l)$ denotes the number of trips for vehicle k. If vehicle k is not used, then $Trip_k(t_l) = \varnothing$.
p_i, p_j	The nodes visited by vehicles, $p_i, p_i \in S \cup D$.		
t_l	The lth scheduling point.		
TL	The set of scheduling points, $t_l \in TL$.		

Table 1. *Cont.*

Indexes and Sets			
Parameters			
T	Scheduling period.	$q_u\ (t_l)$	The number of samples to be transferred at sampling site s_u in scheduling point t_l (tubes).
Q	Vehicle capacity (tube).	$rm_v\ (t_l)$	The number of backlogged samples not tested at the testing agency d_v before scheduling point t_l (tubes).
V	Vehicle average velocity.	dis_{ij}	The distance from node p_i to node p_j (km), $p_i, p_j \in S \cup D$.
CV	Fixed cost for one vehicle used.	TC_i	The handover time spent at node p_i, $p_i \in S \cup D$.
CF	Fuel cost of vehicles (RMB/km).	$AT_k\ (t_l)$	When $l \geq 2$, the moment when the last trip of vehicle k completes the sample delivery based on the schedule of the l-1th scheduling period. $AT_k\ (t_l) > t_l$ is allowed. Let $AT_k\ (t_1) = t_1$.
CD_v	The testing unit price of the testing agency d_v (RMB/tube, 10-in-1 test).		
VD_v	The testing capacity of the testing agency d_v (tube/hour).		
Decision Variables			
$z_k\ (t_l)$	1 denotes that the vehicle k is used in the lth scheduling period, 0 means not.	$x_{ijkw}\ (t_l)$	1 denotes that the wth trip of vehicle k travels from p_i to p_j in the lth scheduling period, 0 means not. p_i and p_j can be the sampling site or testing agency, but not both testing agencies, i.e., $x_{ijkw}(t_l) = 0$, if $p_i \in D \& p_j \in D$.
$y_{uk}\ (t_l)$	1 denotes that the samples collected from s_u are delivered by the vehicle k in the lth scheduling period, 0 means not.		
$e_{uv}\ (t_l)$	1 denotes that the samples collected from s_u are delivered to the testing agency d_v in the lth scheduling period, 0 means not.		

3.2.2. The Scheduling Cost and the Testing Time

Considering the difference in testing unit prices and testing capacities of testing agencies, the cross-depot collaborative transport method is adopted, and the testing cost $Cost_1$, transportation cost $Cost_2$, and fixed cost $Cost_3$ are formulated as follows:

$$Cost_1(t_l) = \sum_{s_u \in S} \sum_{d_v \in D} (rm_v(t_l) + e_{uv}(t_l) q_u(t_l)) CD_v \tag{1}$$

$$Cost_2(t_l) = \sum_{k \in K} \sum_{Trip_k^w(t_l) \in Trip_k(t_l)} \sum_{p_i \in S \cup D} \sum_{p_j \in S \cup D} x_{ijkw}(t_l) dis_{ij} CF \tag{2}$$

$$Cost_3(t_l) = \sum_{k \in K} z_k(t_l) CV \tag{3}$$

The testing time of the testing agency d_v is as follows:

$$CTime_v = \sum_{s_u \in S} \frac{rm_v(t_l) + e_{uv}(t_l) q_u(t_l)}{VD_v}, d_v \in D \tag{4}$$

The correlation matrix between the decision variables in $Cost_1$, $Cost_2$, and $Cost_3$ is shown as follows:

$$\begin{bmatrix} & z_k(t_l) & y_{uk}(t_l) & e_{uv}(t_l) & x_{ijkw}(t_l) \\ z_k(t_l) & 1 & 1 & 0 & 1 \\ y_{uk}(t_l) & 1 & 1 & 1 & 1 \\ e_{uv}(t_l) & 0 & 1 & 1 & 0 \\ x_{ijkw}(t_l) & 1 & 1 & 0 & 1 \end{bmatrix} \tag{5}$$

3.2.3. A Constrained Dynamic Multi-Objective Optimization Model for TSVRP

For any scheduling point $t_l \in TL$, a constrained dynamic multi-objective optimization model for TSVRP is as follows:

$$\min f_1(t_l) = Cost_1(t_l) + Cost_2(t_l) + Cost_3(t_l) \tag{6}$$

$$\min f_2(t_l) = \max_{d_v \in D} \{CTime_v\} \tag{7}$$

s.t.

constraints on the testing agencies:

$$\sum_{dv\in D}\sum_{s_u\in S} x_{vukw}(t_l) = \sum_{s_u\in S}\sum_{dv\in D} x_{uvkw}(t_l), \forall k \in K, \forall Trip_k^w(t_l) \in Trip_k(t_l) \tag{8}$$

$$\sum_{k\in K}\sum_{s_u\in S} x_{vuk1}(t_l) \le |K_v(t_l)|, \forall d_v \in D \tag{9}$$

$$\sum_{t_l\in TL}\sum_{s_u\in S} e_{uv}(t_l) q_u(t_l) \le VD_v \cdot 24, \forall d_v \in D \tag{10}$$

constraints on the sampling sites:

$$\sum_{Trip_k^w(t_l)\in Trip_k(t_l)}\sum_{p_j\in S\cup D} x_{jukw}(t_l) = y_{uk}(t_l), \forall s_u \in S, \forall k \in K \tag{11}$$

$$\sum_{Trip_k^w(t_l)\in Trip_k(t_l)}\sum_{p_j\in S\cup D} x_{ujkw}(t_l) = y_{uk}(t_l), \forall s_u \in S, \forall k \in K \tag{12}$$

$$\sum_{k\in K} y_{uk}(t_l) = 1, \forall s_u \in S \tag{13}$$

constraints on the transportation:

$$\sum_{p_j\in S\cup D} x_{jukw}(t_l) = \sum_{p_j\in S\cup D} x_{ujkw}(t_l), \forall s_u \in S,\ \forall k \in K,\ \forall Trip_k^w(t_l) \in Trip_k(t_l) \tag{14}$$

$$\begin{gathered}\sum_{p_i\in R_k^w(t_l)}\sum_{p_j\in R_k^w(t_l)} x_{ijkw}(t_l) \le \left|R_k^w(t_l)\right| - 1,\ \forall k \in K,\ \forall Trip_k^w(t_l) \in Trip_k(t_l), \\ R_k^w(t_l) = \left\{s_u \middle| s_u \in Trip_k^w(t_l)\ \&\ s_u \in S\right\}\end{gathered} \tag{15}$$

$$\sum_{p_j\in S\cup D}\sum_{s_u\in S} x_{jukw}(t_l) q_u(t_l) \le Q, \forall k \in K, \forall Trip_k^w(t_l) \in Trip_k(t_l) \tag{16}$$

$$\begin{aligned}&\max\{AT_k(t_l),\ t_l\} + \sum_{Trip_k^w(t_l)\in Trip_k(t_l)\setminus Trip_k^{L_k}(t_l)}\sum_{p_i\in S\cup D}\sum_{p_j\in S\cup D}\left(x_{ijkw}(t_l)\left(\frac{dis_{ij}}{V} + TC_j\right)\right) \\ &+ \sum_{p_i\in S\cup D}\sum_{s_u\in S}\left(x_{iukL_k}(t_l)\left(\frac{dis_{iu}}{V} + TC_u\right)\right) \le t_{l+1}, \forall k \in K\end{aligned} \tag{17}$$

where Equations (6) and (7) represent the two optimization objectives. Equation (6) minimizes the total scheduling cost, including the cost of sample testing, the cost of vehicle fuel consumption, and the fixed cost of the vehicles. Equation (7) minimizes the maximum testing time of all testing agencies. Equations (8)–(17) are constraints. Equation (8) assures that all trips depart from and return to the testing agency, while the departure and delivery agencies can be different. Equation (9) guarantees that in the first scheduling period, the number of vehicles departing from the testing agency d_v in the first trip cannot exceed the number of available vehicles in d_v. Equation (10) defines that the total number of samples sent to the testing agency d_v in one day does not exceed its maximum daily testing capacity. Equations (11) and (12) require that a vehicle must leave the sampling site after arriving at it and that the vehicle can arrive and leave the sampling site at most once in one scheduling period. Equation (13) guarantees that a sampling site is served by only one vehicle in a scheduling period. Equation (14) represents the flow-conservation constraint for each trip of each vehicle. Equation (15) is the sub-tour elimination constraint, which ensures that there are no sub-tours in each trip of each vehicle, where $R_k^w(t_l)$ is the set of sampling sites in the wth trip of vehicle k. Equation (16) limits that the quantity of samples transported by the vehicle per trip does not exceed the vehicle capacity. Equation (17) is the latest time constraint for the sample collection, which ensures that the collection of samples must be completed within the current scheduling period, i.e., the travel time to the sampling sites for the last trip $Trip_k^{L_k}(t_l)$ of vehicle k is before t_{l+1}.

4. KMOSFLA for Solving TSVRP

A knowledge-guided multi-objective shuffled frog leaping algorithm KMOSFLA is proposed for TSVRP.

4.1. The Dynamic Periodic Scheduling Framework for Transporting the Test Samples

TSVRP is solved in a periodic rescheduling manner. The framework is divided into three steps.

Step 1: $l = 1$, the initial population is randomly generated based on the state of sampling sites and testing agencies at the first scheduling point t_1, and the optimal schedule for testing samples in the first schedule period is determined by the proposed algorithm KMOSFLA.

Step 2: $l = l + 1$, at the scheduling point t_l, and the sample backlog and the number of available vehicles at each testing agency are calculated according to the optimal schedule of the l-1th scheduling period. The initial population is generated based on the new state and the optimal schedule for the lth period is determined using KMOSFLA.

Step 3: Step 2 is repeated till the final scheduling period.

4.2. The Flowchart of KMOSFLA

The general flowchart of the proposed KMOSFLA is presented in Figure 3. The main differences from the basic SFLA include (1) the population initialization phase, (2) the individual updating phase, and (3) the objective enhancement phase. A greedy crossover and genetic recombination strategy-based discrete leaping rule is proposed to replace the frog leaping rule of the basic shuffled frog leaping algorithm, and the objective-driven enhanced search is designed based on the characteristics of the optimization objective.

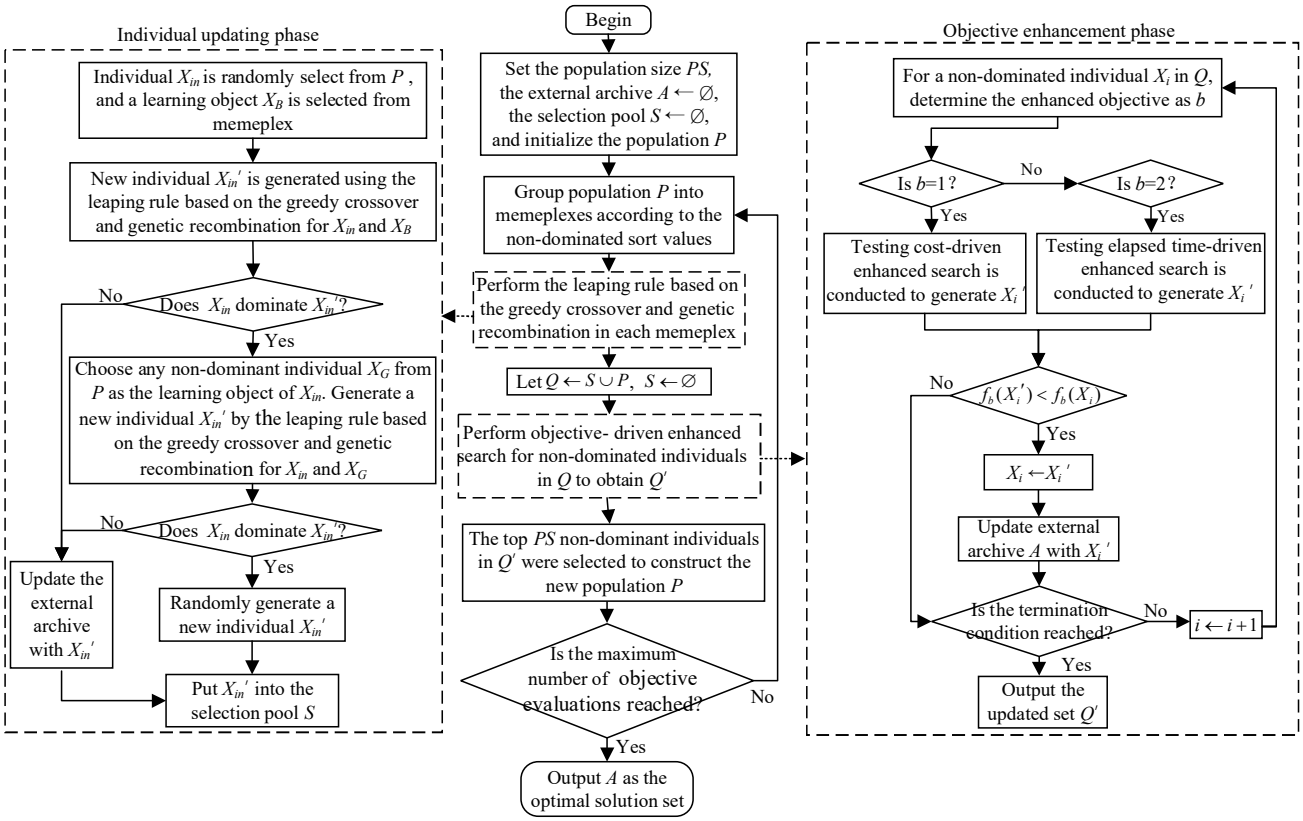


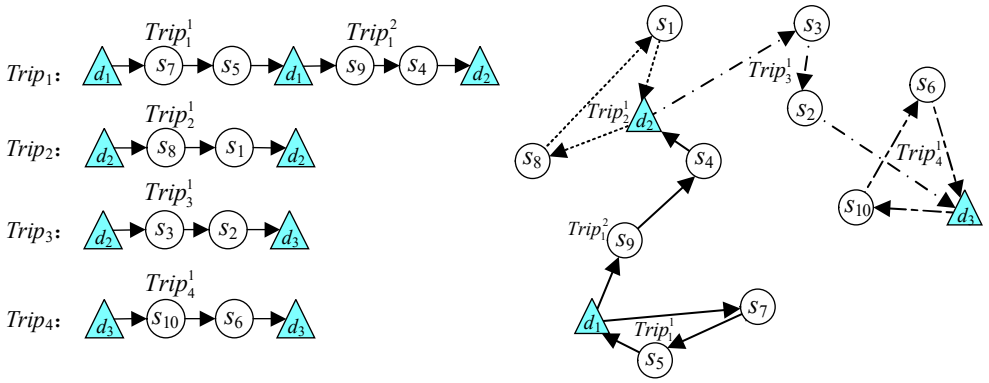
Figure 3. Flowchart of the proposed algorithm KMOSFLA.

4.3. The Convertible Encoding

We design a convertible encoding in which individuals can be represented in two forms: (1) the basic encoding and (2) the three-layer encoding. In the local updating phase, individuals are converted from the basic encoding into the three-layer encoding, where the encoding information is utilized for crossover to generate new individuals, thus achieving a coarse-grained search of the algorithm. In the objective enhancement phase, individuals are converted from the three-layer encoding into the basic form. At this time, individuals are represented by a set of vehicle routes, where some specific routes can be optimized without affecting the rest of them, realizing the fine-grained search of the algorithm.

4.3.1. The Basic Encoding

In the basic encoding form, we use a set of vehicle routes $X = \{Trip_k \mid k \in K\}$ to represent an individual X. Figure 4 illustrates the basic encoding form of individuals, where $D = \{d_1, d_2, d_3\}$ represents the set of three testing agencies, and $S = \{s_1, s_2, \ldots, s_{10}\}$ denotes the set of ten sampling sites. The basic encoding form of the individual and the corresponding routes are shown in Figure 4a,b, respectively, and the individual can be represented as $X = \{Trip_4, Trip_4, Trip_4, Trip_4\}$.

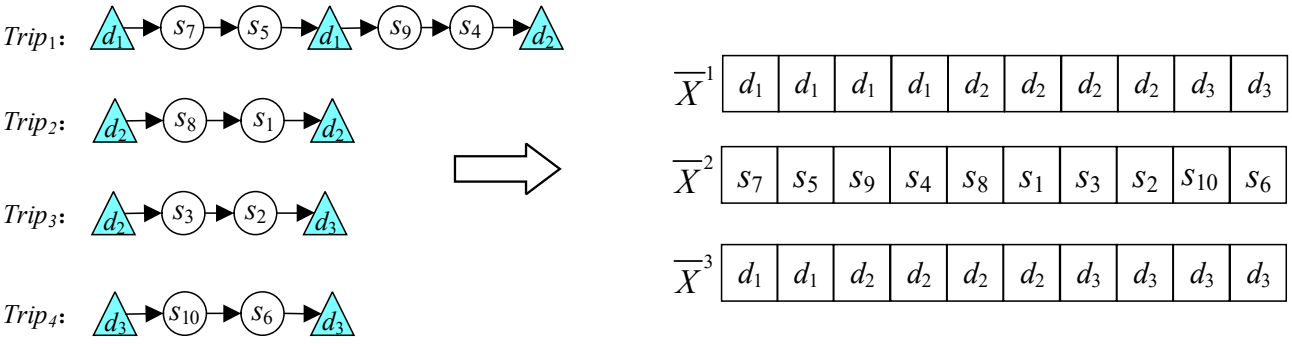


(a) The basic encoding form of the individual X (b) The transfer routes of each vehicle

Figure 4. An illustration of the basic individual encoding.

4.3.2. Conversion from the Basic Encoding to the Three-Layer Encoding

Figure 5 gives an example of the conversion from the basic encoding form to the three-layer encoding form. First, the sampling sites of each trip visited are put into $\overline{X}^2$ according to the order of vehicle 1 to $|K|$. After that, let the agencies in $\overline{X}^1$ and $\overline{X}^3$ represent the departure and delivery agencies of the trip in which the corresponding sampling sites are located, respectively.



(a) The basic encoding form X (b) The three-layer encoding form $\overline{X}$

Figure 5. Conversion from the basic encoding to the three-layer encoding.

4.3.3. Decoding of the Three-Layer Encoding

Figure 6 gives an example of individual decoding in the three-layer encoding form. Assume that there are three testing agencies and eight sampling sites, the scheduling point t_l and the latest collection moment are 10:00 and 12:00, respectively, and the vehicle capacity is 500 tubes. Figure 6a shows the three-layer encoding form of the individual. $\overline{X}^1$ and $\overline{X}^3$ represent the testing agencies that the vehicles departed from and delivered to, respectively, $\overline{X}^2$ indicates the collection order of sampling sites, and the value above the sampling sites represents the number of samples that need to be collected. The sampling site and testing agencies in each column of the three-layer encoding correspond to each other. The decoding process is as follows. First, one of the available vehicles from the first testing agency d_3 from the left of $\overline{X}^1$ is randomly selected to start its first trip, visiting the sampling sites of $\overline{X}^2$ in turn. After collecting the samples from s_2 and s_4, the vehicle is loaded with 450 tubes, leaving insufficient capacity to load samples from s_5. At this point, the vehicle sends the samples to the delivery agency d_2, which is in the same column as s_4, completing the first trip, as shown in Figure 6b. Then, the vehicle departs from d_2 to collect samples from the remaining sampling sites in turn after completing the handover at d_2, as in Figure 6c. After completing sample collection at s_5 and s_3, the vehicle has driven a total of 0.2 + 0.3 + 0.5 + 0.1 + 0.1 + 0.7 = 1.9 h, and the current moment is 11:54. If the vehicle continues to visit s_1, the latest collection moment of 12:00 will be exceeded. Therefore, after collecting the samples at s_3, the vehicle delivers them to the testing agency d_1, which is in the same column as s_3, completes its second trip, and ends the transfer task for the current scheduling period. Subsequently, the routes of the second vehicle are planned. At the moment t_1 = 10, an available vehicle is randomly selected from the departure agency d_2 corresponding to sampling site s1. If no vehicle is available at d_2 at this time, the vehicle that arrives at d_2 at the earliest time is selected. As shown in Figure 6d, the vehicle can perform the collection task for the current scheduling period starting from 10:10. The remaining sampling sites are collected the same way as above until all samples are collected. In the end, we obtain the vehicle routes $Trip_1$, $Trip_2$, and $Trip_3$ as $d_3 \rightarrow s_2 \rightarrow s_4 \rightarrow d_2 \rightarrow s_5 \rightarrow s_3 \rightarrow d_1$, $d_2 \rightarrow s_1 \rightarrow s_7 \rightarrow d_1$, and $d_3 \rightarrow s_6 \rightarrow s_7 \rightarrow d_1$, respectively. The basic encoding form of the individual is X = { $Trip_1$, $Trip_2$, $Trip_3$ }.

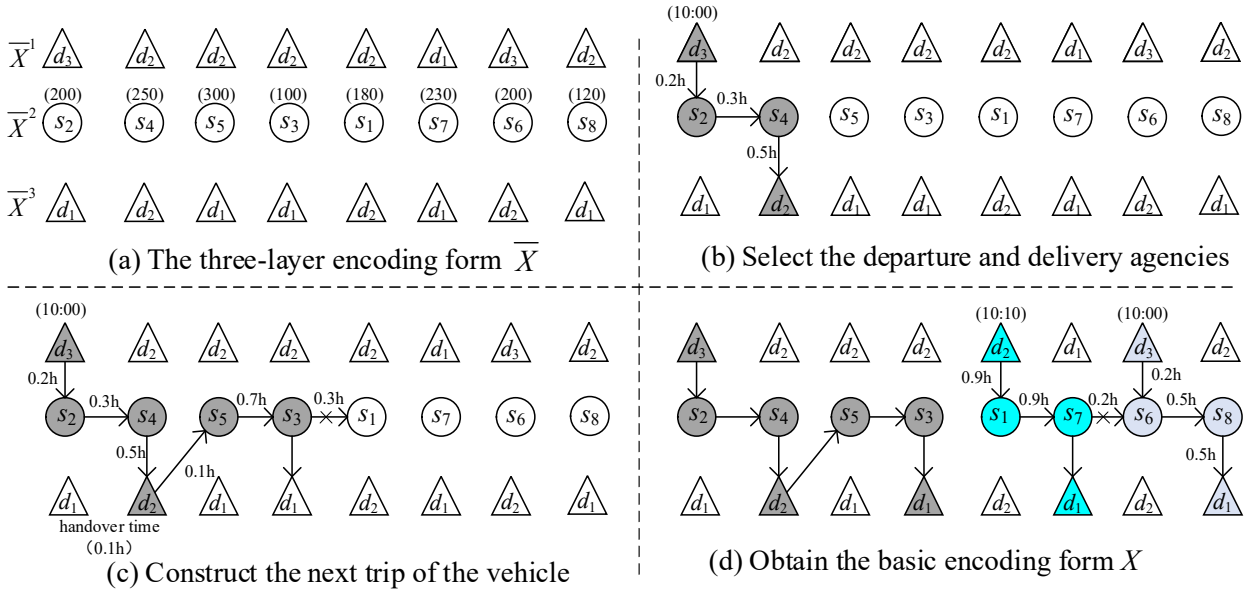


Figure 6. Decoding from the three-layer form to the basic form.

4.4. Population Initialization

At the initial scheduling point t_1, the population is initialized randomly. At the scheduling point t_l ($l > 1$), the population is constructed by the method of 50% of historical information reuse and 50% of random individuals to improve the quality of the initial population while increasing the diversity.

(1) Random initialization. First, a three-layer encoding form $\overline{X} = [\overline{X}^1; \overline{X}^2; \overline{X}^3]$ of the individual X is randomly generated, where $\overline{X}^1$, $\overline{X}^2$, and $\overline{X}^3$ are row vectors of length n (number of sampling sites). Each node in $\overline{X}^1$ and $\overline{X}^3$ is randomly selected from the set of testing agencies D, and each node in $\overline{X}^2$ is a random non-repeating sampling site chosen from the set of sampling sites S. Subsequently, $\overline{X}$ is decoded into the basic encoding form X.

(2) Reuse of the historical information. To improve the search efficiency, the superior scheduling solutions generated in the previous period are considered as historical information for reuse at each scheduling point. Since the number of samples and available vehicles at each sampling site in the current period are different from those in the previous period, the historically superior solution may become infeasible in the current period. Let the non-dominated solutions obtained by the proposed algorithm in the (l-1)th period be the historical optimal solution. The basic encoding forms of these solutions are repaired based on the constraint of available vehicles of each testing agency of Equation (8) and the constraint of vehicle capacity of Equation (13), so that they become feasible initial solutions in the lth period. The specific steps for generating initial solutions based on historical information reuse are shown in Appendix A.

4.5. The Individual Updating Phase

The matching leaping strategy is designed to select learning objects for the worst frogs in the memeplex. The leaping rule based on greedy crossover and genetic recombination is performed to update the worst frog for a large-scale coarse-grained search.

4.5.1. The Matching Leaping Strategy

The learning object of each memeplex in the basic SFLA lacks diversity, which might lead to the population assimilation and falling into the local optimum. For this reason, the matching leaping strategy is proposed. First, the population is divided into m_1 memeplexes according to fitness, with n_1 frogs in each memeplex (n_1 is an even number). Second, for each memeplex, the best individual in the memeplex is paired with the worst individual, the second-best individual in the memeplex is paired with the second-to-last individual, and so on.

After all individuals in each memeplex are paired, the inferior individual X_W leaps toward the superior individual Xb to update its position to obtain X_C. If X_W does not dominate X_C, then X_C is put into the selection pool; otherwise, frog X_W leaps to any non-dominated individual in the population, and if X_W does not dominate the newly updated individual X'_C, then it is put in the selection pool; otherwise, it randomly generates an individual and puts it in the selection pool.

4.5.2. The Leaping Rule Based on the Greedy Crossover and Genetic Recombination

In the individual updating phase, to fully use the information of the sampling sites, departure, and delivery agencies of the individual encoding form, it is necessary to convert individuals into the three-layer encoding form, so that poor individuals leap toward superior individuals.

(a) Greedy crossover

The greedy crossover is an individual updating method that continuously constructs a route from the starting node. Based on the idea of greedy selection, among the individuals

to be crossed, the nearest node adjacent to the current sampling site of the new individual is selected as the next sampling site of the new individual to retain the valuable information in the crossover individual. The greedy selection is repeatedly performed to generate the complete route. In this work, the greedy crossover is used to update the second layer of the three-layer encoding form of individuals.

Figure 7 gives an example of the greedy crossover. Let $\overline{X}_W$ and $\overline{X}_B$ denote the inferior and superior individuals represented in the three-layer encoding form. In Step 1, a sampling site (e.g., s_5) is randomly selected as the starting node of the second layer encoding for the new individual, and a direction (e.g., to the right) is determined randomly. In Step 2, the closest point to s_5 (i.e., s_3) is determined from the two sampling sites (s_1 and s_3) adjacent to the right side of the starting node s_5 in $\overline{X}_W^2$ and $\overline{X}_B^2$ as the next sampling site of $new\overline{X}^2$. In Step 3, the next sampling site is determined, i.e., s_1, the node nearest to s_3 among the nodes adjacent to the right side of $\overline{X}_W^2$ and $\overline{X}_B^2$. In Step 4, only the last two sampling sites s_2 and s_4 are unselected. Since s_2 is closer to s_1, the next sampling site is s_2, and s_4 is the last. The length of the new route $new\overline{X}^2$ is 4.4 km, which is 1.2 km and 1.9 km less compared to $\overline{X}_W^2$ and $\overline{X}_B^2$, respectively.

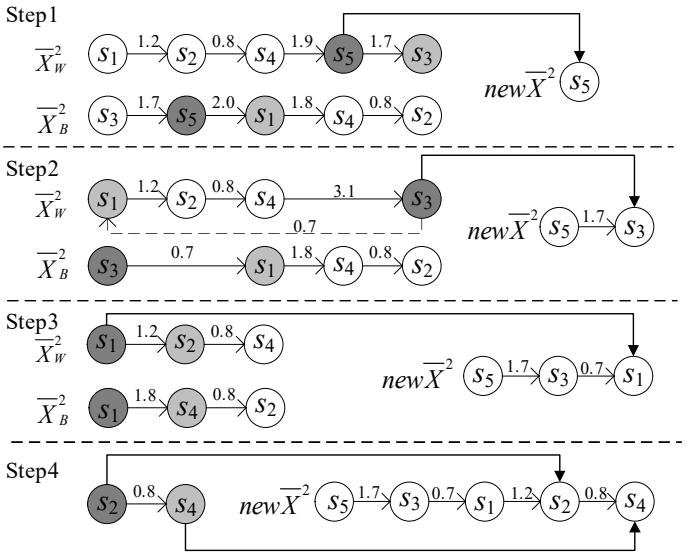


Figure 7. An illustration of the greedy crossover.

(b) Genetic recombination

We design a genetic recombination method to update the first-layer encoding (departure agencies) and the third-layer encoding (delivery agencies) of individuals.

An example of the genetic recombination method is shown in Figure 8. In Step 1, the second layer encoding the worst frog $\overline{X}_W$ is replaced with the $new\overline{X}^2$ obtained by the greedy crossover described in Section 4.5.2 (a). Then, the first and third layers of the worst frog are adjusted to obtain the frog $\overline{X}_W{}'$ accordingly, so that the departure and delivery agencies of the sampling sites are the same as the original three-layer encoding. In Step 2, the departure agencies are updated (the first layer encoding). The distance from the departure agency to the sampling site in each column is calculated, and the top $rand_1$ (a random number between 1 and the number of sampling sites n) farthest departure agencies from the sampling site are selected (e.g., the three agencies in solid gray). The gray solid departure agencies in $\overline{X}_W{}'$ are replaced with the departure agencies (shaded in gray) of the best frog $\overline{X}_B$ according to the corresponding sampling sites. In Step 3, the delivery agencies (the third layer encoding) are updated in the same way, while replacing the solid blue delivery agency selected in $\overline{X}_W{}'$ with the blue-shaded one in $\overline{X}_B$.

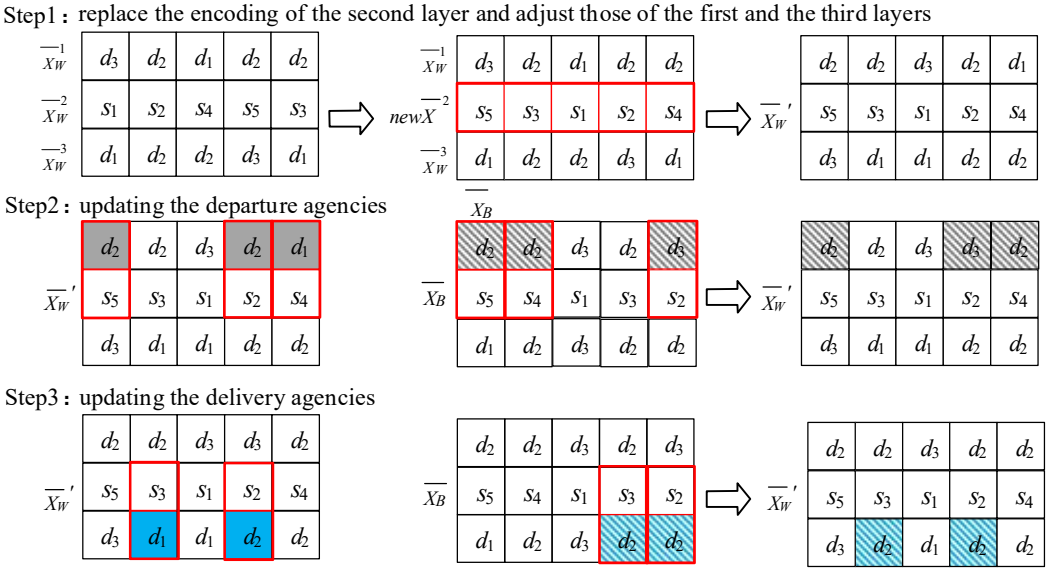


Figure 8. An illustration of the genetic recombination.

4.6. The Objective Enhancement Phase

In this section, we introduce problem-based heuristic information to optimize the specific route in the basic encoding form of the individuals to achieve local information reconstruction of routes and improve individuals' objective value. Since the individuals of the population are in the three-layer encoding form after the local updating phase in Section 4.5, the individuals need to be decoded into the basic encoding form using the method in Section 4.3.3. After that, the algorithm can directly optimize the specific vehicle route.

In the transport of testing samples, the travel distance of each vehicle, the testing amount of the testing agencies, and delivery agency selection of each trip jointly affect the scheduling cost and testing time. Therefore, the route optimization operator *RO*, the neighborhood search operators NC_1 and NB_1, and the delivery agency adjustment operators NC_2 and NB_2 are designed from these three perspectives, respectively.

(a) The route optimization operator *RO* (optimizing a single trip)

Figure 9 gives an example of the route optimization operator *RO*. Step 1: for individual X_i, the wth trip of vehicle k is randomly selected as $Trip_k^w$, then the sum distances from each sampling site to its two neighboring nodes in $Trip_k^w$ is calculated, and the sampling site with the largest value is selected (e.g., s_3). Step 2: the selected sampling site in $Trip_k^w$ is removed, and the sampling sites s_1 and s_2 adjacent to s_3 are connected. Step 3: the extra travel distance after inserting the sampling site s_3 into different positions of $Trip_k^w$ is calculated (except the position before the departure agency and after the delivery agency), and it is inserted into the least increased position (e.g., the position between s_2 and s_4).

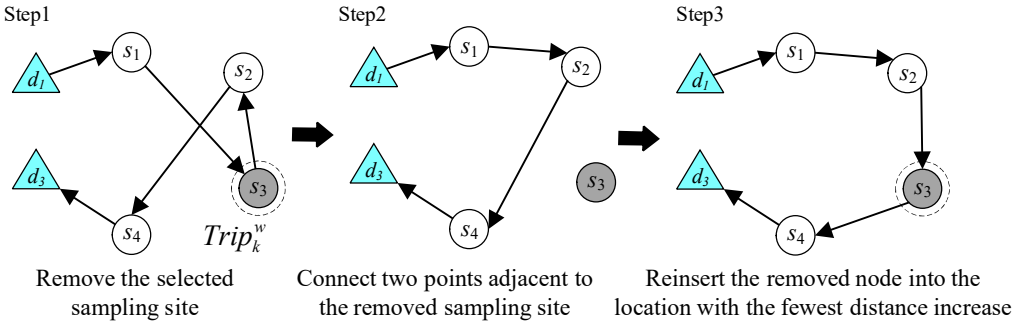


Figure 9. An illustration of the RO operator.

(b) The neighborhood search operators NC1 and NB1 (optimizing multiple trips)

The large-scale neighborhood search *LNS* consists of the removal and reinsertion phase [43]. In the removal phase, Nr selected points are removed from the individual X_i by the removal operator (Nr is a random integer within $[4, \xi \cdot n]$, $\xi = 0.4$, and the total number of points n is much larger than 4 in general). In the reinsertion phase, the insertion operator reinserts the removed points into X_i. In this work, the corresponding removal and reinsertion operators are designed for each objective using different problem-based heuristic knowledge to achieve the local mining of individuals on different objectives.

The removal operators for the sampling sites are described as follows:

(1) The removal operator R_1 for the route with the maximum cost: (a) Calculate the testing cost of each testing agency. Let d_{Cmax} denote the largest one, (b) Randomly select a trip that delivers to d_{Cmax}, calculate the decreased distance after removal of each sampling site in the trip, and remove the largest one. Since the testing cost of each agency will change after removing the sampling site, go to (a) and repeat the above steps after removing N_r sampling sites.

(2) The removal operator R_2 for the route with the maximum testing time: R_2 has the same procedure as R_1, except that the testing cost is replaced with the testing time.

The reinsertion operators are introduced as follows (note that the sampling site cannot be inserted before the departure agency or after the delivery agency):

(1) The minimum cost reinsertion operator I_1: for the sampling site s_u to be inserted, calculate the testing cost after inserting it into different positions of a given trip set, and choose the position with the lowest increase to insert.

(2) The minimum distance reinsertion operator I_2: I_2 has the same procedure as I_1, except that the testing cost is replaced with the travel distance.

The steps of the neighborhood search operators NC_1 and NB_1 are given below based on the above removal and reinsertion operators:

***NC*$_1$**: In the removal phase, the removal operator R_1 is used to remove the Nr sampling sites of X_i (e.g., s_1, s_2, s_7, and s_9 in Figure 10a). In the reinsertion phase, the procedure of sampling sites insertion is as follows: (i) Randomly select a removed sampling site s_u (e.g., s_2 in Figure 10b, which was planned to deliver to agency d_2). (ii) Determine the trip that can be inserted. Let $Dset$ denote the set of trips where the testing unit price does not exceed d_2 and $Trip_{s_u}$ denote the set of trips that deliver to agency d_v ($d_v \in Dset$). As shown in Figure 10b, the trips that are marked as red are those whose testing unit price of delivery agencies is less than or equal to RMB 3.3 for d_2. (iii) Determine the position where s_u can be inserted. The reinsertion operator I_1 is used to reinsert s_u into different trips in $Trip_{s_u}$ and calculate the testing cost that will be increased by the insertion. The position with the least increase and satisfying the capacity constraint and the delivery time constraint (Equations (13) and (14)) is selected to reinsert s_u, as shown in Figure 10c. If no position satisfies the above requirements, a new trip is created to collect the samples of s_u. The departure agency d'_v with available vehicles and which is closest to s_u is determined, and a vehicle is randomly selected. Then the vehicle departs from d'_v through s_u and finally returns to d'_v. The above insertion steps are repeated until all the removed sampling sites are inserted into X_i, as shown in Figure 10d.

***NB*$_1$**: In the removal phase, the removal operator R_2 is adopted to remove Nr sampling sites of individual X_i (e.g., s_2, s_4, s_6, and s_9 in Figure 11a). In the reinsertion phase, the sampling sites are reinserted in the following three steps: (I) Let d_{Tmin} denote the testing agency with the shortest testing time. Select the sampling site s_u with the shortest distance to d_{Tmin} from all the removed sample sites to implement the insertion operation (e.g., s_4 in Figure 11b). (II) Determine the trips that can be inserted. Let all trips with a delivery agency of d_{Tmin} form the set $Trip'_{s_u}$, as in the red trip in Figure 11b. (III) Determine the position that can be inserted. The reinsertion operator I_2 is used to reinsert the s_u into different trips and calculate the testing time that will be increased after insertion. The position with the least increase and satisfying the capacity constraint and delivery time constraint

(Equations (13) and (14)) is chosen to insert s_u, and the testing time of the delivery agency is updated after insertion, as in Figure 11c. If there is no feasible position, the same method of adding a new trip to the s_u as in step (III) of NC_1 is used. Repeat the above insertion steps until all removed sampling sites are inserted into the individual X_i, as in Figure 11d.

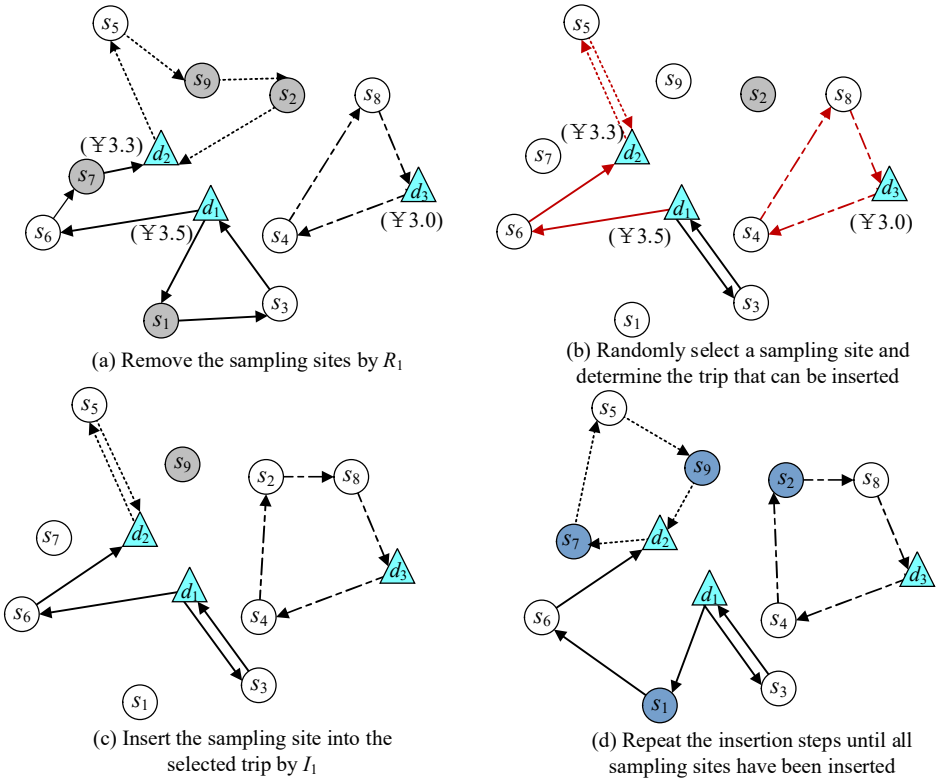


Figure 10. An illustration of the operator NC1.

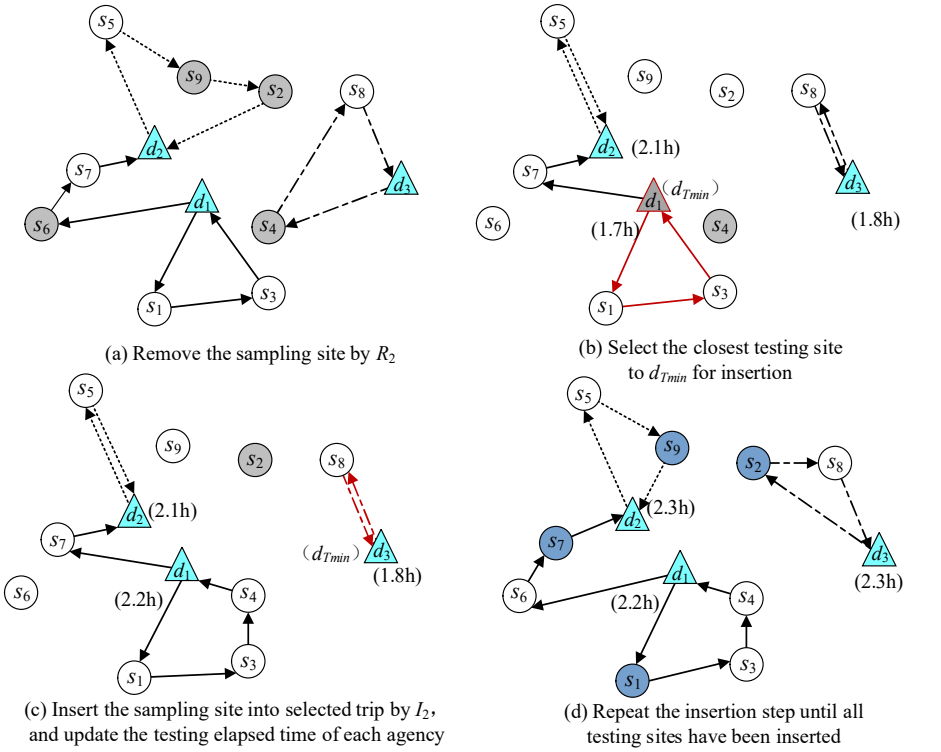


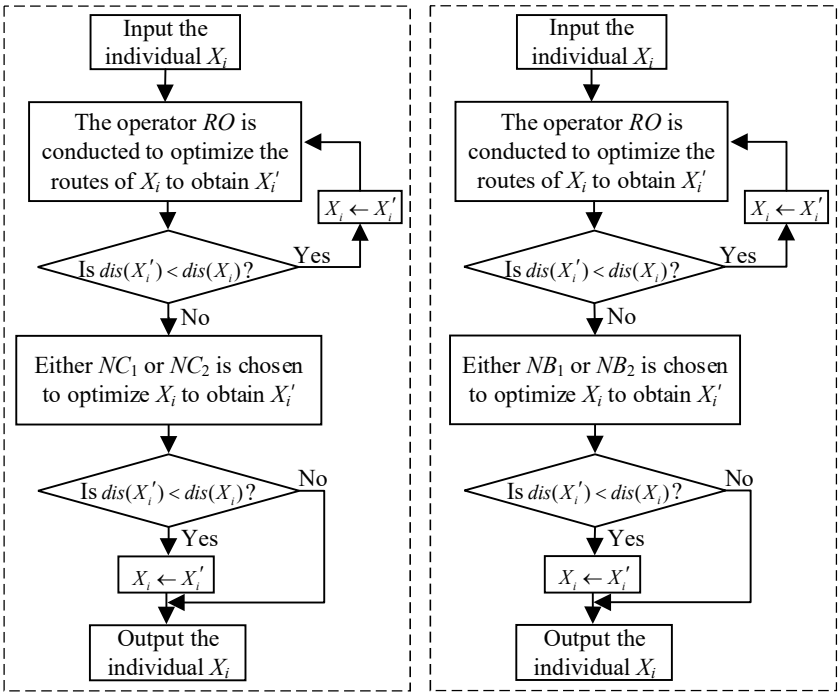
Figure 11. An illustration of the operator NB1.

(c) The delivery agency optimization operators NC_2 and NB_2

NC_2: Randomly select a trip $Trip_k^w$ from the trips where it delivers to the non-lowest unit price testing agency and assume that s_{u1} is its last collected sampling site. Select the closest agency to s_{u1} from the testing agency with a lower testing unit price than $Trip_k^w$ to replace its original delivery agency of $Trip_k^w$.

NB_2: Determine the testing agency that takes the longest testing time as d_{Tmax}. Randomly select a trip $Trip_k^w$ that delivers to testing agency d_{Tmax} and assume that s_{u1} is its last collected sampling site. Determine the closest agency to s_{u1} from the testing agency that is not d_{Tmax} to replace the original testing agency of $Trip_k^w$.

Based on the above operators, testing cost-driven and testing time-driven enhanced search are proposed, as in Figure 12. Since the trip operator *RO* only adjusts the position of a sampling site in its trip and does not change the elapsed testing time of agencies, we iteratively use *RO* to reduce the transportation cost as much as possible. The neighborhood search operators NC_1 and NB_1 and the delivery agency optimization operators NC_2 and NB_2 involve changes in multiple sampling sites. In order to avoid excessive damage to the original routes while saving computational resources, NC_1 or NC_2 (NB_1 or NB_2) is applied only once to optimize the individual.



(a) Testing cost-driven enhanced search (b) Testing time-driven enhanced search

Figure 12. Flowchart of the two objective-driven enhanced search operators.

With the aim of improving the convergence accuracy and distribution of the Pareto front, an objective-driven enhanced search is designed for each non-dominated individual in each generation, and one of the two strategies in Figure 12 is selected for optimization. The steps are described as follows.

Step 1: Sort the individuals in the current non-dominated set in ascending order according to f_1.

Step 2: Put the individuals whose ranking values are in the top 50% into the set Set_1 (f_1 is better) and the bottom 50% into the set Set_2 (f_2 is better).

Step 3: Perform the testing cost-driven search for individuals in Set_1, and perform the testing time-driven search for individuals in Set_2.

5. Experimental Studies

In this section, we perform computational experiments to evaluate the effectiveness of the model and algorithm. Three groups of experiments were carried out with the background of regular sampling testing in four subdistricts of Jiangbei New District, Nanjing: (1) validating the effectiveness of the proposed transportation method and sensitivity analysis of key parameters, (2) verifying the validity of the improvement strategies, and (3) proving the overall performance of the proposed algorithm. In this work, PyCharm 3.9 software was used. All the experiments were performed on a computer with Intel (R) core (TM) i5-7200u CPU @2.50GHz and 12GB running memory.

5.1. Instance and Parameter Settings

The real instance of the test sample transport is generated based on the data of regular sample testing conducted in May 2022 in Dachang subdistrict, Pancheng subdistrict, Getang subdistrict, and Yanjiang subdistrict of Jiangbei New District, Nanjing, China. A total of 78 sampling sites and 11 testing agencies (6 public and 5 third-party) are arranged in the four subdistricts with a total of about 408,000 people. The geographical distribution of sampling sites and testing agencies and the testing capacity and testing unit price of each testing agency are shown in Appendix B.

With two hours as a scheduling period, a total of four scheduling periods are set, and each testing agency is equipped with five vehicles at the initial scheduling point. For each vehicle, the maximum loading capacity is 400 tubes, and the fixed cost is 200 RMB each time in each period. The average vehicle velocity is 50 km/h, and the sample handover time for both the sampling sites and testing agencies is 0.1 h. For the proposed algorithm KMOSFLA, the population size N, the number of memeplexes m_1, the maximum number of objective function evaluations $FEmax$, and the parameters ξ in LNS are set as 100, 5, 20,000, and 0.4, respectively [44], which are commonly used in the existing work. The proposed algorithm and each comparison algorithm were run 20 times independently.

5.2. Performance Measures

In this work, two metrics of hypervolume ratio (*HVR*) [45] and the inverted generational distance (*IGD*) [46], which are commonly used in multi-objective optimization, are adopted to measure the performance of the algorithm. A larger *HVR* value indicates a better convergence and wider spread of the obtained Pareto front. Meanwhile, a smaller *IGD* value denotes that the generated solutions cover various parts of the whole reference Pareto front and are closer to the reference Pareto front. The worst value on each objective obtained by all the optimization runs forms the reference point required in *HVR*. Since the true Pareto front is unknown, the non-dominated solutions obtained by all algorithms in 20 runs are merged as the reference Pareto front of the problem.

5.3. Analyses of the Constructed Model TSVRP

This section verifies the feasibility of the proposed cross-depot collaborative transport method and conducts sensitivity analyses on key parameters in the model, such as vehicle capacity and the proportion of the third-party testing agencies.

5.3.1. Validating the Feasibility of the Transport Method

In order to verify the feasibility of the cross-depot collaborative transport method adopted in our model TSVRP, it is replaced by the transport method of Zhen et al. [26] to obtain the comparison model MDVRP. In MDVRP, each vehicle can only deliver the collected samples back to its departure agency. In each scheduling period, the proposed algorithm KMOSFLA runs 20 times on both models independently.

Figure 13 shows the relationship between the transportation mode and the average values of the testing cost (Equation (1)), transportation cost (Equation (2)), fixed cost (Equation (3)), and the maximum testing time (Equation (7)) in 20 runs for both models.

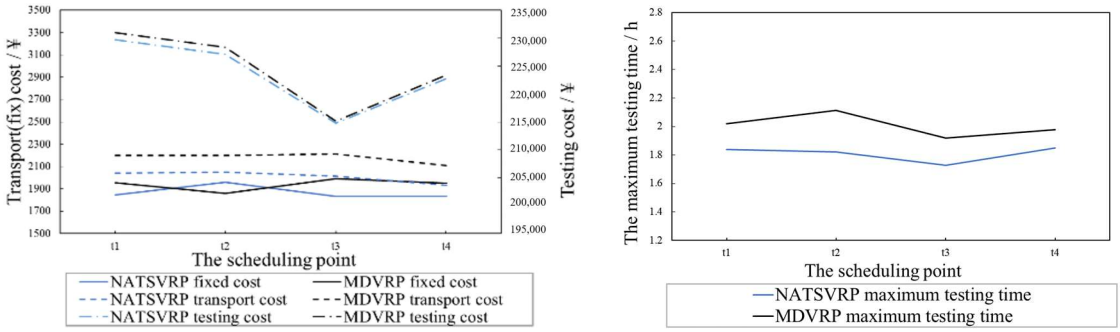


Figure 13. Costs and the maximum testing time obtained by using different transport methods.

As shown in Figure 13, the proposed model TSVRP outperforms MDVRP in terms of transportation cost, testing cost, and the maximum testing time for all scheduling periods, and the fixed cost is also better than the MDVRP in most scheduling periods. Since TSVRP allows vehicles to be transported across testing agencies, it reduces the extra distance the vehicle returns to its departure agency, especially when its last visited sampling site is too far from its departure agency. In addition, the flexible transport method allows vehicles to dynamically adjust their delivery agency according to the difference in sample backlog in the testing agencies, achieving a balance of sample testing volume among different agencies and reducing the maximum testing time and testing cost. In summary, the proposed cross-depot collaborative transport method is feasible and effective.

5.3.2. Sensitivity Analyses of the Vehicle Capacity

In regular test sample transport, taxis, buses, and refrigerated vehicles are usually chosen for sample transfer. The maximum loading capacity of samples varies depending on the vehicle type. In order to provide a guidance to the government on selecting appropriate vehicle types for different requirements, we investigate the relationship between the vehicle capacity Q and the fixed cost, transportation cost, and the maximum testing time (testing cost has no relation to Q). The values of Q are taken as follows: Q = 300, Q = 400, Q = 500, Q = 600, Q = 700, and Q = 800 tubes. The proposed algorithm KMOSFLA runs 20 times at each Q value, and the results in each scheduling period (t_1, t_2, t_3, and t_4) are shown in Figure 14.

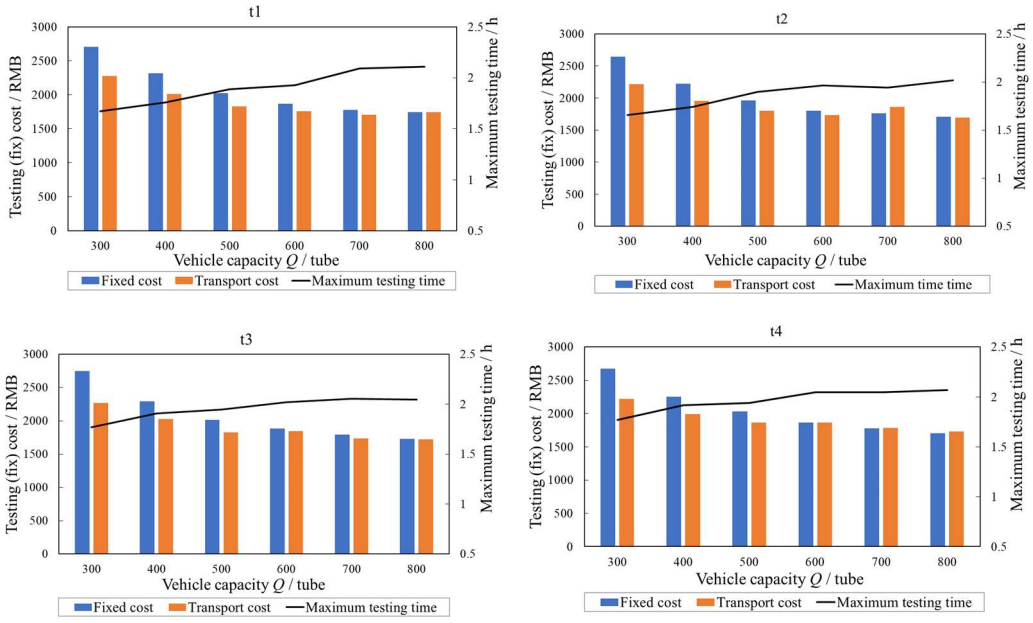


Figure 14. The impact of vehicle capacity on costs and the maximum testing time.

It can be found from Figure 14 that the fixed cost decreases as the vehicle capacity increases. The reasons are as follows: On the one hand, the increase in vehicle capacity allows each vehicle to visit more sampling sites, reducing the number of vehicle uses and thus lowering fixed costs. On the other hand, fewer vehicle trips reduce the distance traveled by the vehicle, thus reducing transportation costs. In addition, although a larger vehicle capacity can improve the number of samples conveyed in a single trip, it is easy to concentrate the samples in some testing agencies, increasing the maximum testing time. Therefore, using low-capacity vehicles allows for a more even distribution of collected samples to the different testing agencies.

5.3.3. Sensitivity Analyses of the Proportion of the Third-Party Testing Agencies

The surge in demand for regular sample testing has led to a boom in third-party testing agencies, which are often better equipped and staffed than public testing agencies and therefore have higher testing unit prices and more substantial testing capacity. To clarify the impact of the testing unit price and testing capacity on the total scheduling cost (Equation (6)) and the maximum testing time, we conducted experiments based on the parameter settings in Section 5.1 with different proportions of the public and the third-party testing agencies. The geographical locations of the 11 testing agencies in Section 5.1 were still used. The testing unit prices and testing capacities of NanjingJiangbei Hospital (20,000 tubes/day, 3 RMB/person) and Nanjing Meizilian Medical Testing Laboratory (60,000 tubes/day, 3.7 RMB/person) were adopted as benchmarks for the public and the third-party testing agencies, respectively. The third-party testing agencies were set at 0%, 27.2% (3/11), 54.5% (6/11), 81.8% (9/11), and 100% of the total number of testing agencies (11), respectively. At different scheduling periods, KMOSFLA was run 20 times with five different types of testing agency combinations. Figure 15 shows the average values of the total cost and the maximum testing time over 20 runs with the proportion of the third-party testing agencies.

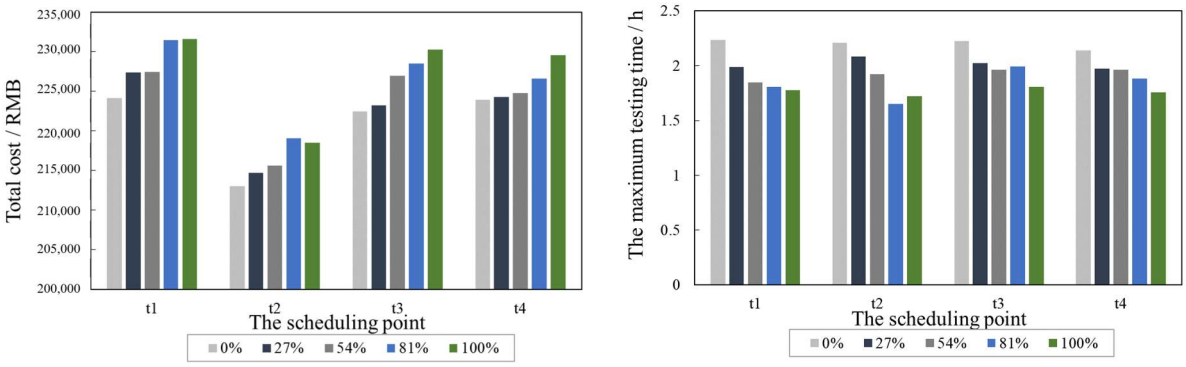


Figure 15. Variations in the total cost and the maximum testing time with the proportion of third-party testing agencies.

It can be observed from Figure 15 that when the testing agencies are all public, the total cost of scheduling is the lowest because of the lower testing unit price. However, the maximum testing time is the longest due to the insufficient overall testing capacity, exceeding the time of a scheduling period (2 h), resulting in an excessive backlog of samples. As the proportion of third-party testing agencies increases, the total testing time decreases rapidly, and all collected samples can gradually complete testing within one scheduling period due to the greater testing capabilities of third-party testing agencies than public agencies. In addition, the government regulates the unit testing price, so the total cost increase is minor. The total cost is the highest when the testing agencies are all third-party agencies because the testing unit price is relatively high. However, the overall testing capacity is the greatest, and the testing time is much less than a scheduling period. In summary, the proportion of third-party testing agencies is controlled at about 54% to

achieve a better balance between the total cost and the maximum testing time spent by testing agencies.

5.4. Performance Validation of the Proposed Algorithm

It is assumed that a TSVRP instance contains $|TL|$ scheduling periods. In each period, 13 comparison algorithms (the proposed algorithm KMOSFLA, 7 algorithms in Section 5.4.1, and 5 algorithms in Section 5.4.2) are adopted to solve the scheduling problem. Each algorithm terminates when the number of objective evaluations reaches 20,000. Since TSVRP has multiple scheduling periods, the overall performance evaluation of each algorithm on all scheduling periods is different from that of static scheduling (only a single scheduling period). The experimental procedures for dynamic scheduling and the overall performance evaluation method of the algorithm are shown in Appendix C. Furthermore, to significantly compare the KMOSFLA with other algorithms, the Wilcoxon rank sum test with a significance level of 0.05 was applied. Table 2 shows the average and best values of *HVR* and *IGD* and the statistical test results of each algorithm. "+" and "−" indicate that KMOSFLA is significantly better or significantly worse than the comparison algorithm, and "=" suggests that there is no significant difference between them. Distributions of the non-dominated solutions obtained by each algorithm in each scheduling period and the corresponding analyses are provided in Appendix D.

Table 2. The average and best values on *HVR* and *IGD* and the statistical test results of 13 algorithms.

Metrics	***HVR***			***IGD***		
Real Instance	**Avg.**	**Best**	**W**	**Avg.**	**Best**	**W**
KMOSFLA	0.9880	1		0.0085	0.0008	
KMOSFLA-RBX	0.8032	0.8996	+	0.0519	0.0360	+
KMOSFLA-CES	0.9265	0.9360	+	0.0118	0.0049	+
KMOSFLA-TES	0.9174	0.9200	+	0.0159	0.0075	+
KMOSFLA-RO	0.6829	0.9860	+	0.0219	0.0129	+
KMOSFLA-LNS	0.7445	0.9741	+	0.0249	0.0111	+
KMOSFLA-DS	0.9566	0.9699	+	0.0169	0.0117	+
KMOSFLA-RN	0.9608	0.9699	+	0.0123	0.0072	+
IACO	0.5021	0.7259	+	0.1255	0.0795	+
MOPSO	0.3094	0.7590	+	0.0776	0.0533	+
MOSFLA-GA	0.5152	0.8364	+	0.0820	0.0647	+
MOSFLA	0.8396	0.9203	+	0.0886	0.0572	+
TS-MOEA	0.8745	0.9389	+	0.1103	0.0494	+

5.4.1. Validating the Effectiveness of the Three Novel Strategies

To verify the effectiveness of the population initialization strategy in Section 4.4, KMOSFLA is compared with KMOSFLA-RN, which randomly generates all the initial individuals, while others remain unchanged. To verify our leaping rule in Section 4.5, KMOSFLA is compared to KMOSFLA-RBX, which replaces the leaping rule with the route-based crossover operator in [47]. The effectiveness of the objective-driven enhanced search in Section 4.6 is validated from two aspects. From the perspective of the enhanced search for objectives, the testing cost-driven search and the testing time-driven search are removed from KMOSFLA to obtain the algorithms KMOSFLA-CES and KMOSFLA-TES, respectively. From the perspective of operators, the route optimization operator (*RO*), the neighborhood search operators (NC_1 and NB_1), and the delivery agency optimization operators (NC_2 and NB_2) are removed to obtain KMOSFLA-RO, KMOSFLA-LNS, and KMOSFLA-DS, respectively. All the above seven algorithms use the same parameter settings as KMOSFLA.

It can be seen from Table 2 that KMOSFLA achieves better average and best values of *HVR* and *IGD* compared with the seven comparison algorithms. Statistical test results also show that KMOSFLA significantly outperforms the comparison algorithms, which indicates that the three novel strategies introduced in KMOSFLA are feasible and effective.

In terms of the population initialization, KMOSFLA repairs the individuals of the external archive and non-dominated individuals of the population at the previous scheduling period to reuse the historical information, and the repaired individuals are adopted as the initial scheduling route for the current scheduling period. Thus, a portion of the high-quality initial solution is obtained using a priori knowledge. Meanwhile, the rest of the initial solutions are generated by random initialization to improve the diversity of the population and avoid population assimilation. In contrast, the comparison algorithm KMOSFLA-RN generates the initial population by random initialization and re-optimizes the new transfer routes at each scheduling period, which is difficult to guarantee the initial population quality and has lower search efficiency and convergence accuracy.

In terms of the individual update strategy, the comparison algorithm KMOSFLA-RBX generates new individuals by route crossover between individuals and performs repair procedures for new individuals, including removing duplicate visited nodes and the reinsertion of unvisited nodes. Although the new individuals retain some excellent parts of the superior individuals, the heuristic information is not utilized, and the algorithm is prone to blind search. In addition, the quality of the repaired individuals cannot be guaranteed because the routes of the new individuals are easily destroyed during the repair procedure. In contrast, the leaping rule of KMOSFLA uses the effective information of the superior individual and the problem-inspiring information of "shorter paths" to guide the inferior individual to adjust its structure and gradually move towards a high-quality region in the objective space. At the same time, the conversion mechanism of the encoding can ensure that the generated individuals are feasible, thus avoiding the information loss caused by the repair procedure and improving the search efficiency.

In terms of the objective-driven enhanced search, KMOSFLA leads to worse performance after removing either type of search operator. There are several reasons for this: the route optimization operator RO reduces the travel distance of the trip; the neighborhood search operators and the delivery agency optimization operators both take the testing cost and testing time as the heuristic information to restructure the basic encoding form of the individuals (i.e., the route of each vehicle) and select a more appropriate delivery agency for the trip, respectively. Each of the three operators optimizes the routes from different perspectives, reducing the coupling between the search operators and jointly driving the improvement in the objective values. In addition, the proposed objective-driven enhanced search performs extreme value mining based on the superior objective of non-dominated individuals. The testing cost-driven search is adopted for the individuals with better f_1, while the testing time-driven search is conducted for the individuals with better f_2. The combined use of the two strategies can effectively locate the extreme value of individuals on both objectives and widen the distribution of the Pareto front, which provides a practical objective enhancement framework for the algorithm. In summary, KMOSFLA significantly outperforms KMOSFLA-CES and KMOSFLA-TES (using only a single objective-driven enhanced search).

5.4.2. Comparisons with the State-of-the-Art Algorithms

The overall performance of the proposed algorithm KMOSFLA is validated. Five state-of-the-art algorithms are selected as the scheduling methods to be compared, including MOPSO [47], TS-MOEA [44], IACO [15], MOSFLA-GA [28], and MOSFLA [48]. MOPSO, TS-MOEA, and IACO are multi-objective metaheuristic algorithms for solving MDVRP used in recent years. MOSFLA-GA and MOSFLA are two multi-objective shuffled frog leaping algorithms, which have been applied to the unmanned aerial vehicle task assignment and the encoding of proteins, respectively. MOPSO, MOSFLA-GA, and MOSFLA adopt the same encoding and decoding method as KMOSFLA where the individual encoding can

be converted between the basic and the three-layer forms, while those of TS-MOEA and IACO remain consistent with their original literature. MOPSO adopts the same individual updating method as its original literature. TS-MOEA employs the route cross-based operator and the mixed neighborhood structure to update individuals. Since multiple trips of each vehicle are not considered in the model of Wang et al. [44], the local search and neighborhood operators of TSMOEA are extended to the operators that fit the multi-trip problem when solving TSVRP. IACO uses an innovative pheromone updating method to optimize the vehicle routes and delivery agencies, allowing vehicles to have different departure and delivery agencies. Since the original applications of MOSFLA-GA and MOSFLA are different from this work, their individual updating strategies cannot be adapted to TSVRP. Therefore, we replace their original strategies with our leaping rule in Section 4.5. MOPSO and TS-MOEA employ their original population initialization methods, and other comparison algorithms use the same method as this work. All five comparison algorithms in this section use the same parameter settings as their original literature, and each algorithm terminates when the number of objective evaluations reaches 20,000.

It can be found from Table 2 that KMOSFLA achieves better overall average and best values of *HVR* and *IGD* compared with the five state-of-the-art algorithms, and the statistical test results also show that KMOSFLA significantly outperforms the comparison algorithms. The superior performance of KMOSFLA benefits from the following three aspects: (i) The convertible encoding mechanism which provides the basis for implementing the proposed strategies: By transforming the encoding forms, the algorithm maps individuals to the corresponding search space at different searching stages. In this way, it can conduct a comprehensive search on sampling sites, departure agencies, delivery agencies, vehicle routes, and trips. (ii) The maintenance of population diversity: The matching leaping strategy selects appropriate learning objects for individuals of different memeplexes, reducing the possibility of the population falling into assimilation. In addition, the individuals that the original individuals do not dominate are retained during the individual updating process to avoid the loss of good individuals. (iii) Comprehensive use of the three novel strategies: Population initialization that reuses historical information provides a good starting point for the search of the algorithm. The individual updating strategy promotes the global convergence of the algorithm in the large-scale search space. The objective-driven enhanced search extracts the characteristics of both objectives and performs extreme value mining to widen the distribution of the Pareto front. The above three strategies cooperate to maintain the balance between the "exploration" and "exploitation" of the algorithm.

6. Conclusions

In this work, we present a knowledge-guided multi-objective shuffled frog leaping algorithm (KMOSFLA) to solve the vehicle routing problem of test samples (TSVRP). Our main work is listed as follows.

First, considering the differences in testing capacity and testing unit price among testing agencies, we adopt a cross-depot collaborative transport method to transfer samples flexibly and develop a dynamic periodic multi-depot multi-trip vehicle routing problem model for test samples. The model contains three strongly coupled subproblems: multi-trip determination, delivery agency selection, and vehicle route optimization. The scheduling cost of government expenditures is minimized, and the sample testing efficiency is improved under the constraints of vehicle capacity, number of vehicles, and delivery time.

Second, a KMOSFLA based on the periodic rescheduling method is proposed to solve the TSVRP. The convertible encoding mechanism is introduced to map individuals to different search space forms at different stages, providing a transformable search environment for the algorithm. Three improvement strategies are designed: (1) constructing the initial population in the new scheduling period by reusing historical information and random initialization to provide a high-quality starting point for the dynamic scheduling algorithm, (2) designing frog leaping rule based on greedy crossover and genetic recombination strate-

gies, using the excellent information of learning objects and the "shorter distance" heuristic information to perform a large-scale search for individuals under the three-layer encoding form, and (3) extracting the characteristics of each objective and conducting the corresponding objective-driven enhanced search for the non-dominated individuals to achieve a small-scale local search for vehicle routes under the basic encoding form of individuals.

Third, three groups of experiments are conducted with the background of sample transport for regular testing in four subdistricts of Jiangbei New District, Nanjing, China. The first group of experiments analyzes the influence of key parameters in the model. Regarding the transportation mode, the cross-depot collaborative transport can effectively reduce scheduling costs and testing time. Regarding vehicle type selection, fixed costs and transportation costs are reduced as vehicle capacity increases, but this causes an increase in testing time; therefore, the choice of medium-sized vehicles can balance cost and testing efficiency. In addition, the proportion of third-party testing agencies is controlled at about 54%, which can better balance the cost and testing time. The second group of experiments verifies the effectiveness of the improvement strategy. The experimental results show that the population initialization of historical information reuse can improve the quality of the initial population. The leaping rule of KMOSFLA improves the convergence accuracy and diversity of the algorithm. The objective-driven enhanced search expands the breadth of the Pareto front distribution. Removing any of the strategies degrades the performance of KMOSFLA, and the strategies affect each other but are not redundant. The third group of experiments verifies the overall performance of the proposed algorithm KMOSFLA. The experimental results show that the *HVR* and *IGD* performance of KMOSFLA is significantly better than the five representative algorithms, indicating that KMOSFLA is able to search for a set of Pareto non-dominated solutions with higher convergence accuracy, more uniform distribution, and a wider width for the proposed model.

Although the model in this work captures the characteristics of dynamic periodic scheduling, multiple trips, and multiple depots of TSVRP, some other factors in the actual sample transfer process are still not considered. For example, vehicle velocity uncertainty might occur concerning road network distribution. Travel time, weather changes, and other factors might cause road congestion and the late delivery of vehicles, thus slowing down the progress of testing. In addition, test samples are infectious items. In order to avoid infections during sample transfer, a transport risk indicator, which concerns the quantity of transferred samples, the transportation duration, and the population density along the transport routes, needs to be defined to improve the safety of sample transportation.

Author Contributions: Conceptualization, Y.Z. and X.S.; methodology, Y.Z. and X.S.; software, Y.Z. and Z.G.; validation, X.S. and Z.G.; formal analysis, Y.Z.; investigation, X.S.; resources, Y.Z.; data curation, Z.G.; writing—original draft preparation, Z.G.; writing—review and editing, Y.Z. and X.S.; visualization, X.S.; Funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant No. 61502239 and the Natural Science Foundation of Jiangsu under Grant No. BK20150924.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Due to the nature of this research, participants in this study did not agree for their data to be shared publicly, thus supporting data are not available.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Steps for the Population Initialization That Reuses Historical Information

Step 1: Determine the initial number G of individuals that are generated by reusing historical information. For the external archive A obtained by the (l-1)th scheduling period, if $|A| \geq G$, G individuals are randomly selected from A to form the history set H. Otherwise, sort the individuals of the final population in the (l-1)th scheduling period based on ranking

values. Then select (G-$|A|$) top individuals that are not in A and merge them with A to form H. Let $g = 1$ and $v = 1$.

Step 2: The repair procedure is performed based on the number of available vehicles at each testing agency according to Equation (8) in the manuscript.

Step 2.1: If $v \leq m$ (number of the testing agencies), determine the number of available vehicles for testing agency d_v in the lth scheduling period. Otherwise, go to Step 3.

Step 2.2: For the individual $X_g \in H$, in its basic encoding form, the number of vehicles N_v whose first trip departs from the testing agency d_v is calculated (e.g., N_2 corresponding to d_2 is 2 in Figure 4a of the main text, assume that $K_2(t_l) = 1$). If $N_v > K_v(t_l)$, randomly select a vehicle trip that delivers to d_v to perform the repair procedure (e.g., choose either $Trip_2^1$ or $Trip_3^1$ in Figure 4a of the main text). Let $s_{u\prime}$ denote the first sampling site visited by $Trip_k^1$, determine the nearest testing agency with an available vehicle that has a delivery agency that is not d_v, and replace the testing agency from d_v with $d_{v\prime}$, $N_v = N_v - 1$. Repeat the repair procedure until $N_v \leq K_v(t_l)$. Let $v = v + 1$ and go to Step 2.1.

Step 3: The repair procedure is performed based on the vehicle capacity constraint of Equation (13).

Step 3.1: For each vehicle trip in the individual $X_g \in H$, determine whether the capacity constraint is satisfied at the lth scheduling period. If it is violated, the last sampling site of that trip is removed repeatedly until the capacity constraint is satisfied.

Step 3.2: The last trip of each vehicle in the individual X_g is merged to form the set LT. For each sampling site $s_{u''}$ removed in Step 3.1, determine the nearest trip from $s_{u''}$ to the last sampling site of each trip in LT. Calculate the distance increase after inserting $s_{u''}$ into the left and right side of the last sampling site of the trip, and determine whether the capacity constraint is satisfied after the insertion on the side with the lesser increase. If satisfied, then insert it. Otherwise, find the next closest trip in LT and re-execute the above process. If no trip in LT meets the requirement, add a new trip to visit $s_{u''}$. Determine the nearest testing agency to $s_{u''}$ with available vehicles and use it as the departure and delivery agency for the additional trip. This step reinserts all the sampling sites removed in Step 3.1 into individual X_g.

Step 4: Add individual X_g to the initial population. If $g < G$, $g = g + 1$, $v = 1$, go to Step 2. Otherwise, output the initial population for the lth scheduling period.

Appendix B. Information on Sampling Sites and Testing Agencies for Nucleic Acid Testing in Four Subdistricts of Jiangbei New District, Nanjing, China

Table A1 shows the testing capacity and testing unit price of the testing agencies, while Figure A1 shows the distribution of sampling sites and testing agencies.

Table A1. The testing capacity and testing unit price of the testing agencies.

Testing Agency	Testing Capacity	Testing Unit Price (10-in-1 Test)
The Fourth Affiliated Hospital of Nanjing Medical University (Public)	10,000 tubes/day	3 RMB/person
Jiangbei New District Public Health Service Center (Public)	10,000 tubes/day	3.1 RMB/person
Zhongda Hospital Southeast University (Public)	20,000 tubes/day	3 RMB/person
Nanjing Jinyu Medical Testing Agency (Third-party)	100,000 tubes/day	4 RMB/person
Nanjing Shihe Testing Agency (Third-party)	30,000 tubes/day	3.8 RMB/person
Nanjing Meizhilian Testing Agency (Third-party)	60,000 tubes/day	3.7 RMB/person
Nanjing Weilan Testing Agency (Third-party)	30,000 tubes/day	3.5 RMB/person
Nanjing Shijianbanxue Testing Agency (Third-party)	20,000 tubes/day	3.6 RMB/person
Nanjing Diyang Testing Agency (Third-party)	20,000 tubes/day	3.4 RMB/person
NanjingJiangbei Hospital (Public)	20,000 tubes/day	3 RMB/person
Nanjing Jianghehuashen testing agency (Third-party)	20,000 tubes/day	3.5 RMB/person

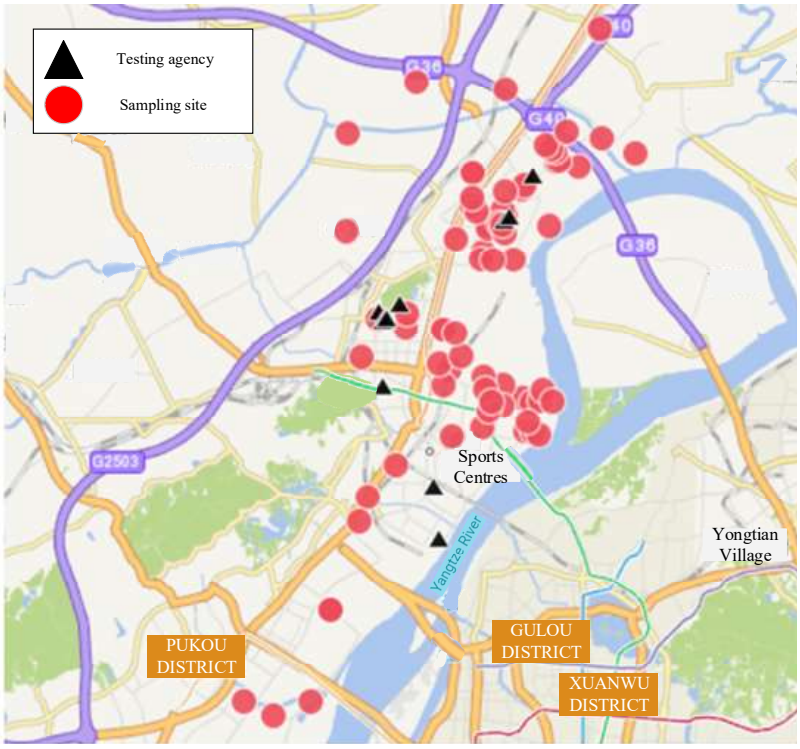


Figure A1. Distribution of sampling sites and testing agencies.

Appendix C. Procedure of the Overall Performance Comparisons of the Algorithms in Dynamic Environment

Step 1: At each scheduling moment t_l, run each of the 13 algorithms 20 times, with the maximum number of objective evaluations of 20,000 for each run.

Step 2: At each scheduling moment t_l, perform the following substeps:

Substep 2.1: All the non-dominated sets obtained by the 13 algorithms in 20 runs are merged, and the new non-dominated solutions are determined from them to form the reference Pareto front at the moment t_l.

Substep 2.2: For each algorithm in each of the 20 runs, the performance metrics (*HVR*, *IGD*) are calculated using its generated solution set. As shown in Figure A2, at the scheduling moment t_l, the 20 values are $metric_j^{k,i}(t_l)$, where $metric_j^{k,i}(t_l)$ denotes the kth performance metric value of the ith algorithm in the jth run at t_l, $k = 1, 2$, $i = 1, 2, \ldots, 13$, $j = 1, 2, \ldots, 20$. *HVR* and *IGD* are regarded as the 1st and 2nd metrics.

Substep 2.3: One solution is selected from the reference Pareto front as the new schedule at scheduling moment t_l to be implemented in the transferring of the nucleic acid test samples based on the Analytic Hierarchy Process (AHP). In this way, it can be guaranteed that at each scheduling point, the 13 methods are compared in the same project environment.

Step 3: Perform nucleic acid sample transfer tasks according to the schedule selected in Substep 2.3.

Step 4: If the maximum scheduling period has not been met yet, then move to the next scheduling period and go to Step 2. Otherwise, go to Step 5.

Step 5: To significantly compare the 13 algorithms in terms of the overall performance across different scheduling moments and runs, for the jth ($j = 1, 2, \ldots, 20$) run of the ith ($i = 1, 2, \ldots, 13$) algorithm, the kth ($k = 1, 2$) performance metrics are averaged over all the scheduling moments ($t_1, t_2, \ldots, t_{|TL|}$) as $mean_j^{k,i}$ shown in Figure A2. The 20 mean values $mean_j^{k,i}$ ($j = 1, 2, \ldots, 20$) form the vector $Vec^{k,i}$, and record the optimal and average values in $Vec^{k,i}$ as $best^{k,i}$ and $mean^{k,i}$, respectively. Finally, the vectors of the proposed algorithm

are tested statistically against each of the other 11 algorithms using the Wilcoxon rank sum test at a significance level of 0.05.

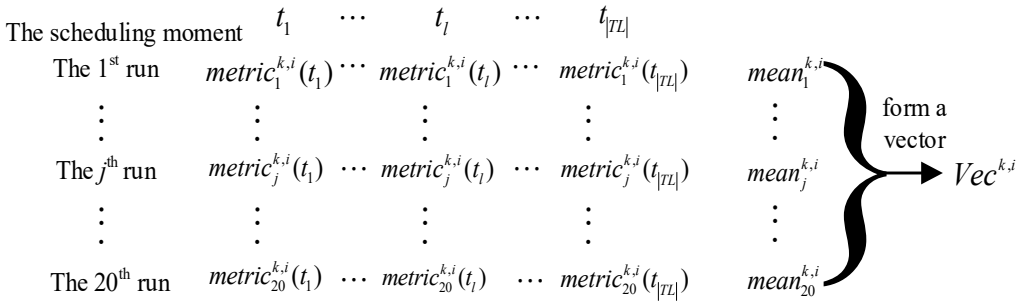


Figure A2. An illustration of the overall performance comparison of 13 algorithms in the NATSVRP instance.

Appendix D. Analyses of the Effectiveness of the Proposed Strategies

The Pareto fronts obtained by the proposed algorithm KMOSFLA and the four comparison algorithms adopted in validating the effectiveness of the three novel strategies are shown in Figure A3. It can be found that the Pareto front of KMOSFLA at different scheduling moments lies below the rest of the algorithm, i.e., it can search for a set of non-dominated solutions with higher accuracy, more uniform distribution, and a larger span on the two objectives. Due to the removal of the testing cost-driven search and retention of the testing time-driven search, KMOSFLA-CES searches for solutions with higher accuracy (better values of f_2) in the lower right of the Pareto front. However, there is no solution distribution or the solutions obtained with lower precision (worse values of f_1) are in the upper left. The results of KMOSFLA-TES are the opposite of KMOSFLA-CES. As KMOSFLA-RBX requires frequent individual repair in the process of individual updating, the algorithm obtains a small number of solutions, most of its non-dominated solutions are concentrated in the middle region of the objective space, and the solution accuracy is low. KMOSFLA-RN generates the initial population by random initialization, while KMOSFLA retains the high-quality individuals from the previous scheduling period, which improves the starting point of the algorithm and searches for a set of solutions with higher accuracy.

To verify the effectiveness of the route operator, the neighborhood search operators, and the delivery agency optimization operators, Figure A4 gives a comparison of the proposed algorithm KMOSFLA with the Pareto front after removing the three operators separately. It can be found that the convergence accuracy of KMOSFLA-RO and KMOSLFA_DS is slightly worse than that of KMOSFLA. However, the distribution of the solutions is comparable to that of KMOSFLA, indicating that the two operators can effectively improve the convergence of the algorithm. At the same time, the convergence accuracy and distributivity of KMOSFLA-LNS are lower than those of KMOSFLA, indicating that the neighborhood search operators have a significant role in both improving the convergence accuracy of the algorithm and expanding the distribution of solutions.

Figure A5 illustrates the Pareto fronts obtained by the proposed algorithm KMOSFLA and five state-of-the-art algorithms. It can be found that the non-dominated solutions obtained by the proposed KMOSFLA dominate that of the five comparison algorithms. TS-MOEA outperforms KMOSFLA in the distribution breadth of the Pareto front since TS-MOEA decomposes the multi-objective problem into multiple sub-problems, and multiple directions of the objective space are explored based on different weight vectors. In the second stage, the Pareto frontier is extended based on the populations obtained in the first stage to find the widely distributed non-dominated solutions. This result indicates that the extreme value mining capability of the proposed algorithm can be further improved.

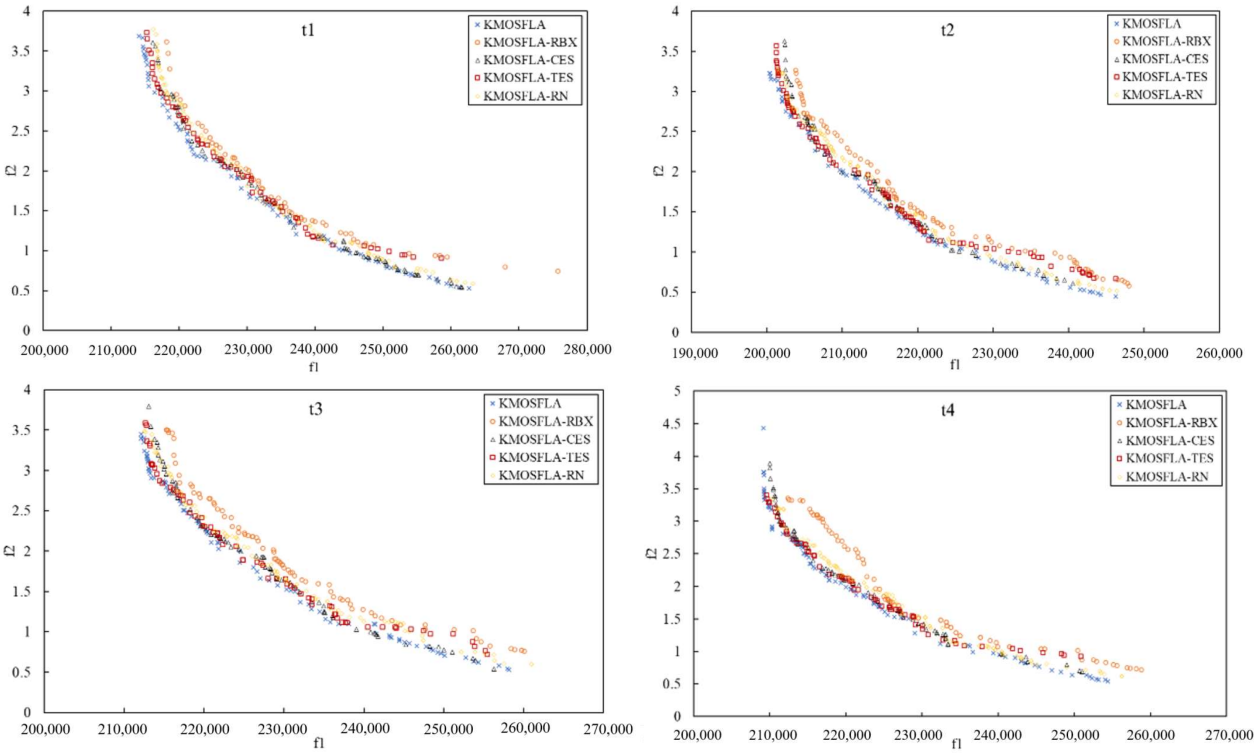


Figure A3. Pareto fronts obtained by the proposed algorithm and the four comparison algorithms when validating the effectiveness of the three novel strategies (results obtained from the first run).

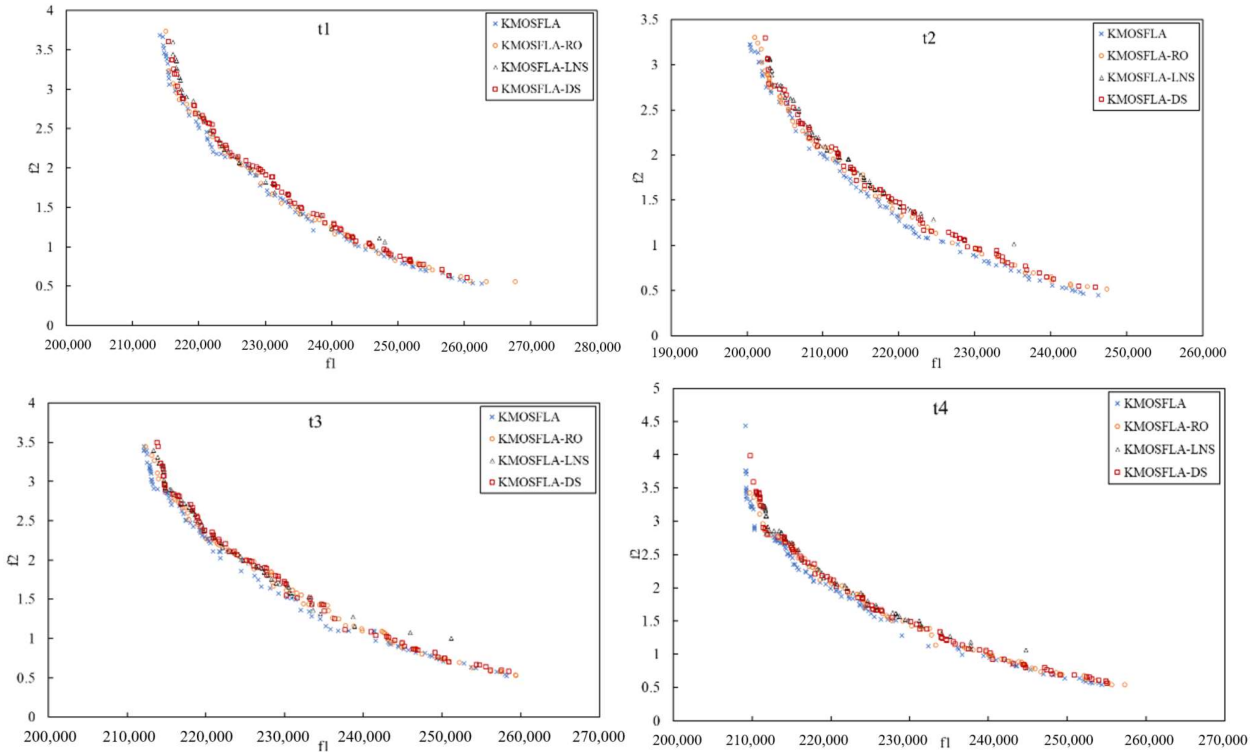


Figure A4. Pareto fronts obtained by the proposed algorithm and the three comparison algorithms when validating the effectiveness of the three operators (results obtained from the first run).

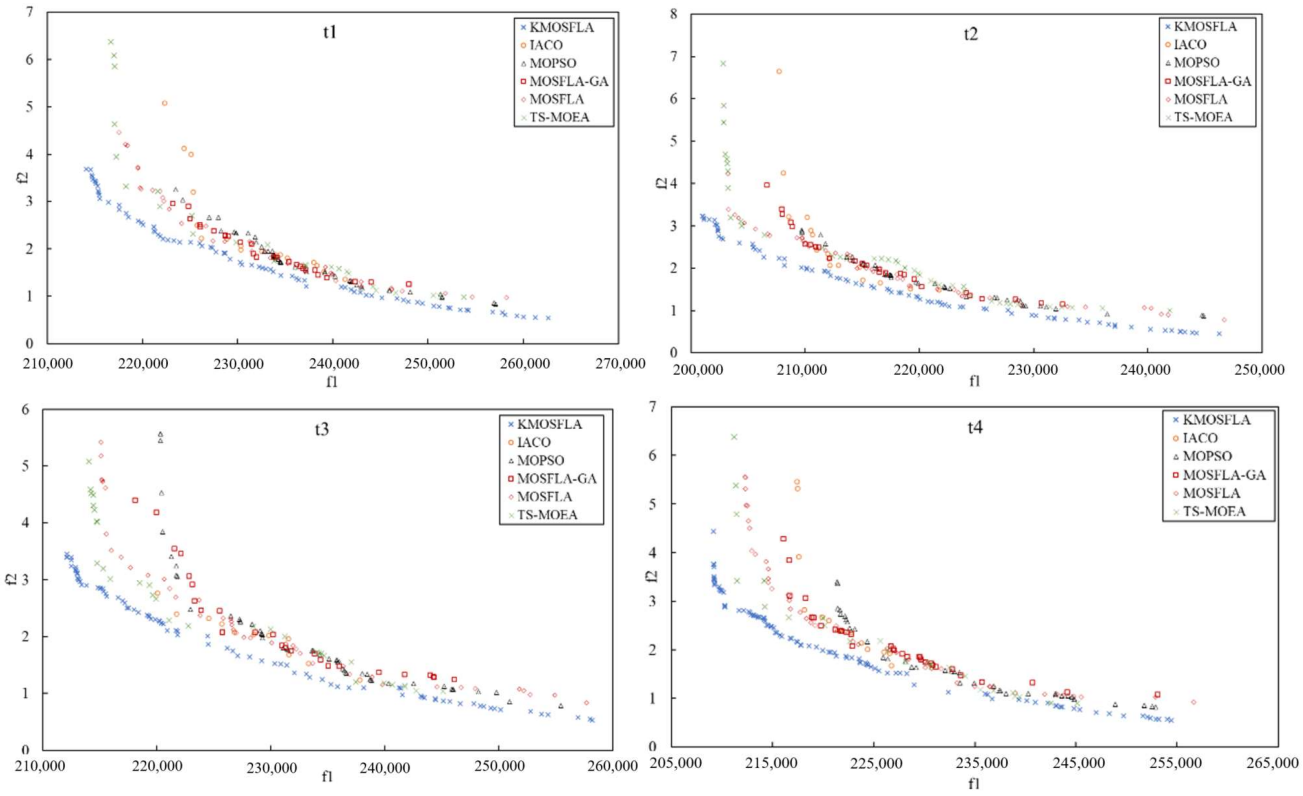


Figure A5. Pareto fronts obtained by the proposed algorithm and the four state-of-the-art comparison algorithms (results obtained from the first run).

References

1. Manupati, V.K.; Schoenherr, T.; Subramanian, N.; Ramkumar, M.; Soni, B.; Panigrahi, S. A multi-echelon dynamic cold chain for managing vaccine distribution. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *156*, 102542. [CrossRef] [PubMed]
2. Pasha, J.; Nwodu, A.L.; Fathollahi-Fard, A.M.; Tian, G.; Li, Z.; Wang, H.; Dulebenets, M.A. Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings. *Adv. Eng. Inform.* **2022**, *52*, 101623. [CrossRef]
3. Eren, E.; Tuzkaya, U.R. Safe Distance-Based Vehicle Routing Problem: Medical Waste Collection Case Study in COVID-19 Pandemic. *Comput. Ind. Eng.* **2021**, *157*, 107328. [CrossRef] [PubMed]
4. Zhang, M.-X.; Yan, H.-F.; Wu, J.-Y.; Zheng, Y.-J. Quarantine vehicle scheduling for transferring high-risk individuals in epidemic areas. *Int. J. Environ. Res. Public Health* **2020**, *17*, 2275. [CrossRef]
5. Osaba, E.; Yang, X.-S.; Fister, I.; Del Ser, J.; Lopez-Garcia, P.; Vazquez-Pardavila, A.J. A discrete and improved bat algorithm for solving a medical goods distribution problem with pharmacological waste collection. *Swarm Evol. Comput.* **2018**, *44*, S2210650217307630. [CrossRef]
6. Zheng, Y.-J.; Chen, X.; Song, Q.; Yang, J.; Wang, L. Evolutionary Optimization of COVID-19 Vaccine Distribution with Evolutionary Demands. *IEEE Trans. Evol. Comput.* **2022**, *27*, 141–154. [CrossRef]
7. Tlili, T.; Masri, H.; Krichen, S. Towards an efficient collection and transport of COVID-19 diagnostic specimens using genetic-based algorithms. *Appl. Soft Comput.* **2022**, *116*, 108264. [CrossRef] [PubMed]
8. Luo, J.; Chen, M.R. Multi-phase modified shuffled frog leaping algorithm with extremal optimization for the MDVRP and the MDVRPTW. *Comput. Ind. Eng.* **2014**, *72*, 84–97. [CrossRef]
9. Desaulniers, G.; Pecin, D.; Contardo, C. Selective pricing in branch-price-and-cut algorithms for vehicle routing. *EURO J. Transp. Logist.* **2019**, *8*, 147–168. [CrossRef]
10. Neira, D.A.; Aguayo, M.M.; De la Fuente, R.; Klapp, M.A. New compact integer programming formulations for the multi-trip vehicle routing problem with time windows. *Comput. Ind. Eng.* **2020**, *144*, 106399. [CrossRef]
11. Reddy, A.; Narayana, K. Meta-heuristics optimization in electric vehicles—An extensive review. *Renew. Sustain. Energy Rev.* **2022**, *160*, 112285. [CrossRef]
12. Ji, J.; Guo, Y.; Gong, D.; Shen, X. Evolutionary multi-task allocation for mobile crowdsensing with limited resource. *Swarm Evol. Comput.* **2021**, *63*, 100872. [CrossRef]
13. Eusuff, M.; Lansey, K.; Pasha, F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Eng. Optim.* **2006**, *38*, 129–154. [CrossRef]

14. He, Z.; Xiao, L.; Wang, X. Minimization for ternary fixed polarity Reed–Muller expressions based on ternary quantum shuffled frog leaping algorithm. *Appl. Soft Comput.* **2021**, *110*, 107647. [CrossRef]
15. Lei, D.; Wang, T. Solving distributed two-stage hybrid flowshop scheduling using a shuffled frog-leaping algorithm with memeplex grouping. *Eng. Optim.* **2020**, *52*, 1461–1474. [CrossRef]
16. Dash, R. An improved shuffled frog leaping algorithm based evolutionary framework for currency exchange rate prediction. *Phys. A Stat. Mech. Its Appl.* **2017**, *486*, 782–796. [CrossRef]
17. Guo, Y.; Tian, X.; Fang, G.; Xu, Y.-P. Many-objective optimization with improved shuffled frog leaping algorithm for inter-basin water transfers. *Adv. Water Resour.* **2020**, *138*, 103531. [CrossRef]
18. Kenan, N.; Diabat, A. The supply chain of blood products in the wake of the COVID-19 pandemic: Appointment scheduling and other restrictions. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *159*, 102576. [CrossRef] [PubMed]
19. Goodarzian, F.; Taleizadeh, A.A.; Ghasemi, P.; Abraham, A. An integrated sustainable medical supply chain network during COVID-19. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104188. [CrossRef]
20. Pacheco, J.; Laguna, M. Vehicle routing for the urgent delivery of face shields during the COVID-19 pandemic. *J. Heuristics* **2020**, *26*, 619–635. [CrossRef]
21. Tirkolaee, E.B.; Abbasian, P.; Weber, G.W. Sustainable fuzzy multi-trip location-routing problem for medical waste management during the COVID-19 outbreak. *Sci. Total Environ.* **2021**, *756*, 143607. [CrossRef] [PubMed]
22. Ramos, T.R.P.; Gomes, M.I.; Póvoa, A.P.B. Multi-depot vehicle routing problem: A comparative study of alternative formulations. *Int. J. Logist. Res. Appl.* **2020**, *23*, 103–120. [CrossRef]
23. Qin, J.; Ye, Y.; Cheng, B.-R.; Zhao, X.; Ni, L. The emergency vehicle routing problem with uncertain demand under sustainability environments. *Sustainability* **2017**, *9*, 288. [CrossRef]
24. Yesodha, R.; Amudha, T. A bio-inspired approach: Firefly algorithm for Multi-Depot Vehicle Routing Problem with Time Windows. *Comput. Commun.* **2022**, *190*, 48–56. [CrossRef]
25. Wang, Y.; Li, Q.; Guan, X.; Xu, M.; Liu, Y.; Wang, H. Two-echelon collaborative multi-depot multi-period vehicle routing problem. *Expert Syst. Appl.* **2021**, *167*, 114201. [CrossRef]
26. Zhen, L.; Ma, C.; Wang, K.; Xiao, L.; Zhang, W. Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, *135*, 101866. [CrossRef]
27. Cen, X.; Zhou, G.; Ji, B.; Yu, S.S.; Zhang, Z.; Fang, X. Modelling and heuristically solving three-dimensional loading constrained vehicle routing problem with cross-docking. *Adv. Eng. Inform.* **2023**, *57*, 102029. [CrossRef]
28. Xu, H.; Pu, P.; Duan, F. A hybrid ant colony optimization for dynamic multidepot vehicle routing problem. *Discret. Dyn. Nat. Soc.* **2018**, *2018*, 3624728. [CrossRef]
29. Yu, B.; Ma, N.; Cai, W.; Li, T.; Yuan, X.; Yao, B. Improved ant colony optimisation for the dynamic multi-depot vehicle routing problem. *Int. J. Logist. Res. Appl.* **2013**, *16*, 144–157. [CrossRef]
30. Liu, Y.; Khalifa, I.H.; El Kamel, A. The multi-period and multi-depot dynamic vehicle routing problem with time windows. In Proceedings of the 2016 3rd International Conference on Logistics Operations Management (GOL), Fez, Morocco, 23–25 May 2016; pp. 1–6. [CrossRef]
31. Seyyedhasani, H.; Dvorak, J.S. Dynamic rerouting of a fleet of vehicles in agricultural operations through a Dynamic Multiple Depot Vehicle Routing Problem representation. *Biosyst. Eng.* **2018**, *171*, 63–77. [CrossRef]
32. Fang, G.; Guo, Y.; Wen, X.; Fu, X.; Lei, X.; Tian, Y.; Wang, T. Multi-objective differential evolution-chaos shuffled frog leaping algorithm for water resources system optimization. *Water Resour. Manag.* **2018**, *32*, 3835–3852. [CrossRef]
33. Yang, Z.; Yang, K.; Su, L.; Hu, H. The multi-objective operation for cascade reservoirs using MMOSFLA with emphasis on power generation and ecological benefit. *J. Hydroinformatics* **2019**, *21*, 257–278. [CrossRef]
34. Lei, D.; Guo, X. A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents. *Expert Syst. Appl.* **2015**, *42*, 9333–9339. [CrossRef]
35. Liu, Y.; Heidari, A.A.; Cai, Z.; Liang, G.; Chen, H.; Pan, Z.; Alsufyani, A.; Bourouis, S. Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. *Neurocomputing* **2022**, *503*, 325–362. [CrossRef]
36. Chegini, S.N.; Bagheri, A.; Najafi, F. PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. *Appl. Soft Comput.* **2018**, *73*, 697–726. [CrossRef]
37. Tang, D.; Yang, J.; Dong, S.; Liu, Z. A lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems. *Appl. Soft Comput.* **2016**, *49*, 641–662. [CrossRef]
38. Sharma, T.K.; Pant, M. Opposition-based learning embedded shuffled frog-leaping algorithm. In *Soft computing: Theories and Applications*; Springer: Singapore, 2018; pp. 853–861. [CrossRef]
39. Luo, J.; Yang, Y.; Liu, Q.; Li, X.; Chen, M.; Gao, K. A new hybrid memetic multi-objective optimization algorithm for multi-objective optimization. *Inf. Sci.* **2018**, *448*, 164–186. [CrossRef]
40. Jie, Z.; Wei, N. Novel multi-objective optimization algorithm. *J. Syst. Eng. Electron.* **2014**, *25*, 697–710. [CrossRef]
41. Lakshmi, K.; Rao, A.R.M. Hybrid shuffled frog leaping optimisation algorithm for multi-objective optimal design of laminate composites. *Comput. Struct.* **2013**, *125*, 200–216. [CrossRef]
42. Azizipanah-Abarghooee, R.; Narimani, M.R.; Bahmani-Firouzi, B.; Niknam, T. Modified shuffled frog leaping algorithm for multi-objective optimal power flow with FACTS devices. *J. Intell. Fuzzy Syst. Appl. Eng. Technol.* **2014**, *26*, 681–692. [CrossRef]

43. Dumez, D.; Lehuédé, F.; Péton, O. A large neighborhood search approach to the vehicle routing problem with delivery options. *Transp. Res. Part B Methodol.* **2021**, *144*, 103–132. [CrossRef]
44. Wang, J.; Weng, T.; Zhang, Q. A Two-Stage Multi-objective Evolutionary Algorithm for Multiobjective Multidepot Vehicle Routing Problem With Time Windows. *IEEE Trans. Cybern.* **2018**, *49*, 2467–2478. [CrossRef] [PubMed]
45. Hui, L.; Zhang, Q. Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2009**, *13*, 284–302. [CrossRef]
46. Veldhuizen, D.; Lamont, G.B. Multi-objective evolutionary algorithm test suites. In Proceedings of the 1999 ACM symposium on Applied computing, San Antonio, TX, USA, 28 February–2 March 1999; pp. 351–357. [CrossRef]
47. Wang, Y.; Assogba, K.; Fan, J.; Xu, M.; Liu, Y.; Wang, H. Multi-depot green vehicle routing problem with shared transportation resource: Integration of time-dependent speed and piecewise penalty cost. *J. Clean. Prod.* **2019**, *232*, 12–29. [CrossRef]
48. Gonzalez-Sanchez, B.; Vega-Rodriguez, M.A.; Santander-Jimenez, S. Multi-objective Memetic Meta-heuristic Algorithm for Encoding the Same Protein with Multiple Genes. *Expert Syst. Appl.* **2019**, *136*, 83–93. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.