



Article

Quarantine Vehicle Scheduling for Transferring High-Risk Individuals in Epidemic Areas

Min-Xia Zhang ¹, Hong-Fan Yan ¹, Jia-Yu Wu ¹ and Yu-Jun Zheng ^{2,*}

¹ College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China; minxia_zhang@yeah.net (M.-X.Z.); hongfangyan@compintell.cn (H.-F.Y.); jiayuwu@compintell.cn (J.-Y.W.)

² School of Information Science and Engineering, Hangzhou Normal University, Hangzhou 311121, China

* Correspondence: yujun.zheng@computer.org

Received: 09 March 2020; Accepted: 23 March 2020; Published: 27 March 2020



Abstract: In a large-scale epidemic outbreak, there can be many high-risk individuals to be transferred for medical isolation in epidemic areas. Typically, the individuals are scattered across different locations, and available quarantine vehicles are limited. Therefore, it is challenging to efficiently schedule the vehicles to transfer the individuals to isolated regions to control the spread of the epidemic. In this paper, we formulate such a quarantine vehicle scheduling problem for high-risk individual transfer, which is more difficult than most well-known vehicle routing problems. To efficiently solve this problem, we propose a hybrid algorithm based on the water wave optimization (WWO) metaheuristic and neighborhood search. The metaheuristic uses a small population to rapidly explore the solution space, and the neighborhood search uses a gradual strategy to improve the solution accuracy. Computational results demonstrate that the proposed algorithm significantly outperforms several existing algorithms and obtains high-quality solutions on real-world problem instances for high-risk individual transfer in Hangzhou, China, during the peak period of the novel coronavirus pneumonia (COVID-19).

Keywords: public health emergencies; epidemics; medical isolation; vehicle scheduling; optimization; water wave optimization (WWO)

1. Introduction

In an outbreak of a severe epidemic, a large number of suspected cases and close contacts of patients, which we call *high-risk individuals*, need to be isolated for medical observation as soon as possible to prevent the spread of the virus carried by them. Another benefit of isolation is to reduce the mortality, that is, whenever a high-risk individual is diagnosed, he/she can get prompt treatment. However, high-risk individuals are often scattered in a city, and the number of available quarantine vehicles that are eligible to transfer high-risk individuals is limited. Therefore, we must efficiently schedule the vehicles, i.e., assign high-risk individuals to the vehicles and determine the route of each vehicle (as illustrated by Figure 1), to transfer the individuals to isolated regions as quickly as possible to control the spread of the epidemic.

The considered problem can be regarded as a variant of the well-known vehicle routing problem (VRP) [1], which has been widely studied in the fields of computer science and operations research. Nevertheless, This problem differs from most well-known VRPs in the following aspects:

- The number of high-risk individuals in a location or small area may exceed the capacity of a vehicle. In such a case, it may require a vehicle to go to the area more than once, or require two or more vehicles to load the individuals in the area.
- After loading high-risk individuals in an area, if a vehicle still has room, it can either go to another area to load more individuals or directly go back to the isolated region (some individuals in serious

condition should be immediately sent to the hospital and are excluded from the individuals to be transferred).

- The objective of the problem is to reduce the risk of epidemic spread as much as possible, and hence the objective function should be defined based on the exposure durations of high-risk individuals.

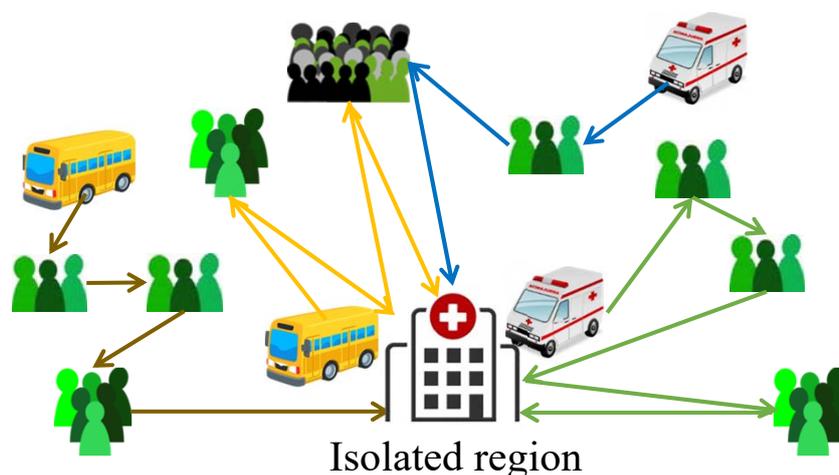


Figure 1. An illustration of the vehicle routing problem for transferring high-risk individuals in epidemics.

Therefore, this problem is significantly more complex than most existing VRPs that are known to be NP-hard. For large instances of the problem, exact algorithms (such as branch-and-bound) are often impractical, and traditional metaheuristic algorithms (such as genetic algorithms) also converge slowly [2]. To handle this difficult problem, we propose a hybrid water wave optimization (WVO) [3] and neighborhood search algorithm, which adapts the WVO metaheuristic to efficiently explore the solution space of the problem and utilizes neighborhood search to improve the solution accuracy. Compared to most existing metaheuristics that require many generations to evolve a large population to explore the solution space, the proposed WVO algorithm evolves a small population of solutions to rapidly locate an optimal or near-optimal solution. The neighborhood search method uses a gradual strategy to further improve the solution accuracy. We demonstrate the efficiency of the proposed algorithm on real-world instances for transferring thousands of high-risk individuals during the peak period of the novel coronavirus pneumonia (COVID-19) in Hangzhou city, China. The main contributions of this paper can be summarized as follows:

- We present a new VRP variant based on the requirements of high-risk individual isolation in large-scale epidemics.
- We provide a set of rules for scheduling multiple vehicles for high-risk individual transfer.
- We propose an efficient algorithm for the problem, which was used to solve real-world instances in COVID-19 within a short response time.

The remainder of this paper is structured as follows. Section 2 introduces related work on basic and emergency VRPs, Section 3 formulates the quarantine VRP for transferring high-risk individuals in epidemics, which also provides the basic scheduling rules that are useful in practice. Section 4 proposes the hybrid algorithm, Section 5 presents the computational results, and Section 6 concludes with a discussion.

2. Related Work

First proposed by Dantzig and Ramse [4], VRPs are well-known NP-hard problems that have been widely studied in the fields of computer science and operations research [1]. Traditional exact

methods are only applicable to small problem instances [5]. In recent years, considerable efforts have been devoted to evolutionary algorithms [6–8], including genetic algorithms (GA) [9–13], particle swarm optimization [14–18], ant colony optimization (ACO) [19–23], artificial bee colony [24–26], biogeography-based optimization [27,28], etc., for VRPs. Compared to exact algorithms and construction heuristics, evolutionary algorithms are more capable of jumping out of local optima and obtaining optimal or near-optimal solutions within an acceptable time by evolving a population of candidate solutions to simultaneously explore multiple regions of the search space [29].

Emergency VRPs are more challenging because they are needed to be solved within a limited response time [30]. Daskin and Haghani [31] studied an emergency VRP that extends the basic VRP to an emergency scene by allowing edge travel times to be normally distributed and the path travel times of different vehicles to be correlated. In [32] Haghani et al. evaluated different response strategies including first-called first-served, nearest-origin assignment, and flexible assignment for assigning response vehicles and guiding those vehicles through non-congested routes. Vargas-Villamil [33] used mixed-integer nonlinear programming to solve a vehicle routing and dispatching problem for emergency personnel evacuation from off-shore oil platforms. Wang and Deng [34] studied an emergency VRP in post-disaster, where customer demands are described as the fuzzy variables; they proposed a two-stage approach, where the first stage optimizes the vehicle routes to minimize transportation time, and the second stage uses fuzzy linear programming to minimize transportation cost. Wohlgemuth et al. [35] considered a dynamic VRP with varying travel times and unknown orders in the specific environment of emergencies; they used a multi-stage mixed integer programming method to find solutions that decrease delays and increase equipment utilization. Chini et al. [36] studied a VRP in both deterministic and stochastic scenarios; they proposed a routing algorithm for graph-represented mission spaces and a switching algorithm to dynamically change the vehicle behavior according to the time-variable configuration of both vehicles and targets. Ye et al. [37] studied an emergency evacuation path planning problem in catastrophic events, which was modeled using an improved cellular automaton based on ACO. Considering fire evacuation scenarios, Shahparvari and Abbasi [38] proposed a greedy heuristic method to solve a VRP under uncertainties in evacuee population, time windows, and bushfire propagation; they applied their approach to a real case study of the 2009 Black Saturday bushfires in Victoria, Australia.

Exact algorithms are not practical for solving large instances of emergency VRPs. Moreover, traditional evolutionary algorithms also converge slowly in large search spaces, and therefore studies on adapting evolutionary algorithms for emergency VRPs are relatively few. Sun et al. [39] proposed an improved ACO algorithm to solve an emergency VRP, where GA is utilized to optimize the parameters of ACO. Chen [40] utilized a GA to solve an emergency VRP with strict timeliness requirements. Du and Yi [41] also tested the performance of GAs for a multi-objective emergency VRP. Nevertheless, the results showed that classical GAs only perform well on small or medium instances. To solve emergency transportation that combines cumulative VRP and multi-depot VRP, Wang et al. [42] proposed an improved ACO algorithm with a smart design of the ant tabus to speed up solution construction. Recently, Fallahi and Sefrioui [43] studied a problem of scheduling ambulances to bring the maximal number of alive victims to hospitals; they proposed a memetic algorithm that employs variable neighborhood search to improve solution accuracy. However, to the best of our knowledge, there have been no reports of VRPs for transferring high-risk individuals in epidemic areas, mainly due to the specific characteristics of the problem in the epidemic situation.

3. Problem Description

3.1. Problem Inputs

We consider the problem of routing multiple quarantine vehicles to transfer high-risk individuals (potential propagators) to an isolated region in epidemic situations. Let $A = \{a_1, a_2, \dots, a_n\}$ be the set of n locations or small areas with high-risk individuals, and b_j be the number of high-risk individuals

in a_j ($1 \leq j \leq n$). We have a set $V = \{v_1, v_2, \dots, v_m\}$ of vehicles, and the capacity of v_i (i.e., the maximum number of individuals that can be carried by v_i) is c_i ($1 \leq i \leq m$). Based on the distances between the areas and the velocities of the vehicles, we can obtain the travel time $t_{i,j}$ of each vehicle v_i from its original location to each area a_j , the time $t_{i,j,j'}$ for v_i to travel from a_j to $a_{j'}$, and the time $t'_{i,j}$ for v_i to travel from a_j to the isolated region ($1 \leq i \leq m; 1 \leq j, j' \leq n$). We also have the average time interval $\Delta t_{i,j}$ between loading two individuals in a_j by v_i , which depends on the dispersion of individuals in the area (for example, if individuals are gathered in a community health center, the interval can be several seconds; if individuals are in different units of a housing estate, the interval can be several minutes). The inputs indicate that we can use vehicles with different capacities and velocities, and thus the problem is a heterogeneous VRP. Table 1 lists the variables used in the problem formulation.

Table 1. Mathematical variables used in the problem formulation.

Symbol	Description
A	Set of areas with high-risk individuals
n	Size of A
a_j	j th area in A ($1 \leq j \leq n$)
b_j	Number of individuals in a_j
V	Set of vehicles
m	Size of V
v_i	i th vehicle in V ($1 \leq i \leq m$)
c_i	Capacity of v_i
c_{\min}	Minimum capacity among all vehicles
$t_{i,j}$	Travel time of v_i from its original location to a_j
$t_{i,j,j'}$	Travel time of v_i from a_j to $a_{j'}$
$t'_{i,j}$	Travel time of v_i from a_j to the isolated region
$\Delta t_{i,j}$	Average time interval between loading two individuals in a_j by v_i
\mathbf{x}_i	Sequence of areas assigned to v_i
$x_{i,j}$	j th area in \mathbf{x}_i
\mathbf{y}_i	Sequence of intermediate decisions of v_i
$y_{i,j}$	Boolean variable denoting whether v_i goes back to the isolated region after loading individuals in $x_{i,j}$
X	Vector of sequences $\{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_m\}$
Y	Vector of sequences $\{\mathbf{y}_1, \dots, \mathbf{y}_i, \dots, \mathbf{y}_m\}$
$\tau_{i,j}$	First arrival time of v_i at x_{ij}
$\tau'_{i,j}$	Final leave time of v_i at x_{ij}
$c_{i,j}$	Remaining capacity of v_i when it arrives at x_{ij}
$k_{i,j}$	Number of times that v_i goes back and forth between a_j and isolated region
A^+	Set of areas, each of which has been assigned to two or more vehicles
$V(a)$	Set of vehicles to which the area a is assigned
T_j	Total exposure time of individuals in a_j

3.2. Decision Variables and Scheduling Rules

The problem needs to first determine the sequence $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n_i}\}$ of areas to be visited by each vehicle v_i , where $x_{i,j}$ is the index of v_i 's j th visiting area in A , and n_i is the number of areas assigned to v_i . Moreover, as aforementioned, after visiting an area, a vehicle may either go back to the isolated region or go to the next area, so we also need to determine the sequence $\mathbf{y}_i = \{y_{i,1}, y_{i,2}, \dots, y_{i,n_i-1}\}$ of intermediate decisions of v_i , where $y_{i,j} = 1$ indicates that v_i will go back to the isolated region after visiting the j th area of its sequence and $y_{i,j} = 0$ otherwise. Consequently, a solution to the problem consists of both the set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ and set $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m\}$.

Nevertheless, the decision variables are not completely free. Based on experiences from the emergency management departments and related studies, we have a set of specific rules for cases where high-risk individuals in an area cannot be transferred by a vehicle in one round. Let $c_{\min} = \min_{1 \leq i \leq m} c_i$, i.e., the minimum capacity among all vehicles. The first rule is as follows:

R1. If $b_j \leq c_{\min}$, then the area a_j can only be assigned to one vehicle.

Note that the vehicle may still go to a_j more than once, in the case when the vehicle goes to a_j for the first time, it has already carried some individuals from other areas and its remaining capacity is less than b_j . In such a case, we have the second rule:

R2. If the area a_j is only assigned to the vehicle v_i , then v_i cannot go to the next area until all individuals in a_j have been loaded.

In other words, if v_i cannot transfer all individuals in a_j in one round and there is no other vehicle assigned for a_j , then v_i should go back and forth between a_j and the isolated region until there is no individual left in a_j .

The third rule limits the number of vehicles that can be utilized to transfer high-risk individuals in an area:

R3. Let k_j be a positive integer satisfying $(k_j - 1)c_{\min} < b_j \leq k_j c_{\min}$, then the area a_j can be assigned to at most k_j vehicles.

When there are two or more vehicles scheduled to transfer high-risk individuals in an area, we have the fourth rule:

R4. If an area a_j is assigned to two or more vehicles, then the vehicle first arrives at a_j should load individuals in a_j as much as possible in the first round.

That is if the available capacity of the vehicle first arrives at a_j is not smaller than b_j , it should load all b_j individuals; otherwise, should use up its capacity in the first round.

Finally, the fifth rule generalizes the application scope of the above four rules:

R5. Whenever a vehicle has been scheduled, the rules R1–R4 still apply to the remaining subproblem.

In general, we first determine the schedule of the vehicle with the minimum capacity without violating R1–R4, and then recalculate c_{\min} of the remaining vehicles; this process recursively continues until all vehicles have been scheduled. For the convenience of applying these rules, without loss of generality, we assume that all vehicles in V have been sorted in nondecreasing order of c_i .

3.3. Solution Evaluation

As our objective is to reduce the transmission risk associated with the individuals during their exposure, we need to calculate the exposure duration of each high-risk individual, i.e., the time period before he/she is loaded into a quarantine vehicle. We first consider a simplified version where each area is only assigned to one vehicle. For each vehicle v_i , its arrival time at its first area $a_{x_{i,1}}$ is:

$$\tau_{i,1} = t_{i,x_{i,1}} \tag{1}$$

The capacity of v_i when it arrives at $a_{x_{i,1}}$ is:

$$c_{i,1} = c_i \tag{2}$$

If $b_{x_{i,1}} \leq c_{i,1}$, then v_i can load all $b_{x_{i,1}}$ individuals at a time and then leave the area; otherwise, according to the rule R2, v_i should go back and forth between the area and isolated region for $k_{i,1}$ times, where

$$k_{i,1} = \lfloor \frac{b_{x_{i,1}}}{c_i} \rfloor \tag{3}$$

Here, $\lfloor x \rfloor$ takes the largest integer smaller than or equal to x , and $\lceil x \rceil$ takes the largest integer larger than or equal to x .

Consequently, the time at which v_i finally leaves $x_{i,1}$ is

$$\tau'_{i,1} = \begin{cases} \tau_{i,1} + (b_{x_{i,1}} - 1)\Delta t_{i,x_{i,1}}, & b_{x_{i,1}} \leq c_{i,1} \\ \tau_{i,1} + 2k_{i,1}t'_{i,x_{i,1}} + (b_{x_{i,1}} - 1)\Delta t_{i,x_{i,1}}, & b_{x_{i,1}} > c_{i,1} \end{cases} \quad (4)$$

The arrival time and capacity at the next area can always be calculated based on the leaving time and capacity of the previous area:

$$\tau_{i,j+1} = \begin{cases} \tau'_{i,j} + t_{i,x_{i,j},x_{i,j+1}}, & y_{i,j} = 0 \\ \tau'_{i,j} + t'_{i,x_{i,j}} + t'_{i,x_{i,j+1}}, & y_{i,j} = 1 \end{cases} \quad (5)$$

$$c_{i,j+1} = \begin{cases} c_{i,j} - (b_{x_{i,j}} \bmod c_i), & y_{i,j} = 0 \\ c_i, & y_{i,j} = 1 \end{cases} \quad (6)$$

The number of times that v_i goes back and forth between the next area and isolated region depends on the intermediate decision and remaining capacity at the previous area:

$$k_{i,j+1} = \begin{cases} 0, & y_{i,j} = 0 \wedge b_{x_{i,j+1}} \leq c_{i,j} \\ 1 + \lfloor \frac{b_{x_{i,j+1}} - c_{i,j}}{c_i} \rfloor, & y_{i,j} = 0 \wedge b_{x_{i,j+1}} > c_{i,j} \\ \lfloor \frac{b_{x_{i,j+1}}}{c_i} \rfloor, & y_{i,j} = 1 \end{cases} \quad (7)$$

By analogy, the time at which v_i finally leaves $x_{i,j}$ is

$$\tau'_{i,j} = \begin{cases} \tau_{i,j} + (b_{x_{i,j}} - 1)\Delta t_{i,x_{i,j}}, & b_{x_{i,j}} \leq c_{i,j} \\ \tau_{i,j} + 2k_{i,j}t'_{i,x_{i,j}} + (b_{x_{i,j}} - 1)\Delta t_{i,x_{i,j}}, & b_{x_{i,j}} > c_{i,j} \end{cases} \quad (8)$$

The total transmission risk is evaluated as the sum of exposure durations of all high-risk individuals. For each area $a_{x_{i,j}}$, the exposure duration of the first individual is $\tau_{i,j}$, each next individual in the first round needs to wait another period of $\Delta t_{i,x_{i,j}}$, and the exposure duration of each individual in subsequent rounds should take the travel time of v_i between the area and the isolated region into consideration. Consequently, the sum of exposure durations of individuals in the area can be calculated by Equation (9).

$$T_{x_{i,j}} = \begin{cases} b_{x_{i,j}} (\tau_{i,j} + \frac{b_{x_{i,j}} - 1}{2} \Delta t_{i,x_{i,j}}), & b_{x_{i,j}} \leq c_{i,j} \\ b_{x_{i,j}} (\tau_{i,j} + \frac{b_{x_{i,j}} - 1}{2} \Delta t_{i,x_{i,j}}) + k_{i,j}t'_{i,x_{i,j}} (2b_{x_{i,j}} - (k_{i,j} + 1)c_{i,j}), & b_{x_{i,j}} > c_{i,j} \end{cases} \quad (9)$$

The main difficulty lies in the case when an area is assigned to two or more vehicles. Let A^+ be the set of such areas and $V(a)$ be the set of vehicles to which the area a is assigned. We use the following procedure to deal with this difficulty.

1. Initialize $T_{x_{i,j}} = 0$ for all i and j , and initialize a set $A^b = \{a_{x_{1,1}}, a_{x_{2,1}}, \dots, a_{x_{m,1}}\}$, i.e., A^b consists of the areas to be first visited by the vehicles.
2. Select from A^b the area a^* with the earliest arrival time.
3. If $a^* \notin A^+$, i.e., all b^* individuals can be loaded in one round, then use Equations (5)–(9) to calculate the sum of exposure durations of individuals in a^* and the arrival time and capacity of the corresponding vehicle at the next area.
4. Otherwise, let v_{i^*} be the first vehicle arriving at a^* , b^* be the number of individuals in a^* , j^* be the index of a^* in the sequence of v_{i^*} , and $b' = \min(b^*, c_{i^*})$, i.e., the number of individuals that

can be loaded by v_{i^*} in the current round. We first increase $T_{x_{i^*,j^*}}$ by the exposure durations of individuals loaded by v_{i^*} in this round:

$$T_{x_{i^*,j^*}} = T_{x_{i^*,j^*}} + b' \left(\tau_{i^*,j^*} + \frac{b'-1}{2} \Delta t_{i^*,x_{i^*,j^*}} \right) \tag{10}$$

and then update the number of remaining individuals in a^* as

$$b^* = b^* - b' \tag{11}$$

(4.1) If $b^* > 0$, temporally update the arrival time of v_{i^*} after the current round of transfer:

$$\tau_{i^*,1} = \tau_{i^*,1} + (b'-1) \Delta t_{i^*,x_{i^*,j^*}} + 2t'_{i,x_{i^*,1}} \tag{12}$$

and then go to step 2).

(4.2) Otherwise $b^* = 0$, use Equations (5) and (6) to calculate the arrival time and capacity of v_{i^*} at the next area.

(4.3) For each other $v_{i'} \in V(a^*)$, restore its arrival time at a^* to the value of the previous round (if exists, because after returning to the isolated region, $v_{i'}$ can directly go to the next area instead of revisiting a^*), and use Equations (5) and (6) to calculate its arrival time and capacity at the next area.

5. Remove a^* from both A^+ and A^b ; if a^* is not the last area in the sequence, add the next area to A^b .
6. Go to step 2) until A^b is empty.

For illustration, consider a simple example shown in Figure 2. Suppose that $x_1 = \{1, 2, 3\}$, $y_1 = \{0, 1\}$, $x_2 = \{2, 3\}$, $y_2 = \{1\}$. According to the above procedure, A^b is initialized as $\{a_1, a_2\}$, for which we have $\tau_{1,1} = 5$ and $\tau_{2,1} = 10$. In the first iteration, we select $a^* = a_1$ and $i^* = 1$, and then calculate

$$\begin{aligned} b' &= 10 \\ T_1 &= 10(5 + 0.5(10-1)/2) = 72.5 \\ \tau'_{1,1} &= 5 + 0.5(10-1) = 9.5 \\ \tau_{1,2} &= 9.5 + 10 = 19.5 \end{aligned}$$

In the second iteration, we have $A^b = \{a_2\}$, $a^* = a_2$ and $i^* = 2$, and then calculate

$$\begin{aligned} b' &= 20 \\ T_2 &= 15(10 + 0.5(20-1)/2) = 221.25 \\ \tau_{2,2} &= 10 + 0.5(20-1) + 2 \times 5 = 29.5 \end{aligned}$$

In the third iteration, we select $a^* = a_2$ and $i^* = 1$, and then calculate

$$\begin{aligned} b' &= 5 \\ T_2 &= 221.25 + 5(19.5 + 0.5(5-1)/2) = 323.75 \\ \tau'_{1,2} &= 19.5 + 0.5(5-1) = 21.5 \\ \tau_{1,3} &= 21.5 + 5 + 10 = 36.5 \end{aligned}$$

As now $b^* = 0$, we restore $\tau_{2,2} = 10$ and recalculate $\tau'_{2,2} = 10 + 0.5(20-1) = 19.5$ and $\tau'_{2,3} = 19.5 + 5 + 10 = 34.5$. Similarly, in the remaining iterations, individuals in a_3 are first transferred by v_2 followed by v_1 .

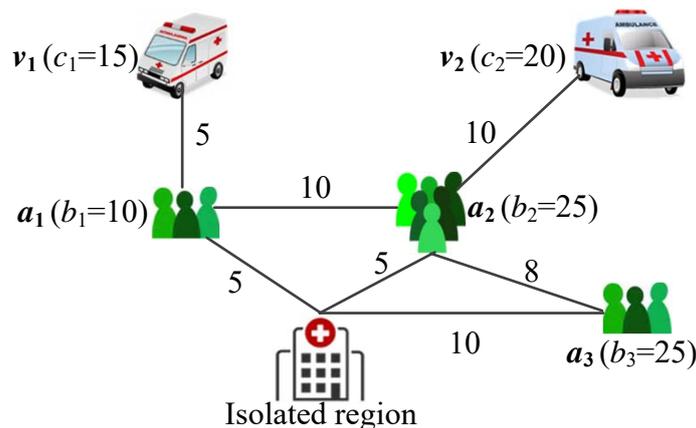


Figure 2. A simple problem instance with two vehicles and three areas. We assume that the vehicles have the same velocity, and the label on a line denotes the travel time between two areas; we also assume that the loading time interval Δt_j is 0.5 for all three areas.

The objective of the problem is to minimize the sum of exposure durations of all individuals, and hence the problem are formulated as

$$\min \quad f(X, Y) = \sum_{j=1}^n T_j \tag{13}$$

s.t. Equations (1)–(12)

$$\bigcup_{i=1}^m \mathbf{x}_i = A \tag{14}$$

$$1 \leq x_{i,j} \leq n, \quad 1 \leq i \leq m; 1 \leq j \leq n_i \tag{15}$$

$$y_{i,j} \in \{0, 1\}, \quad 1 \leq i \leq m; 1 \leq j \leq n_i - 1 \tag{16}$$

Due to the inclusion of Y into the decision, the considered problem is significantly more complex than the basic VRP.

4. Method

In a large-scale epidemic, we often need to schedule multiple vehicles to transfer hundreds to thousands of high-risk individuals distributed in many areas. The solution space of such a problem instance is huge, for which exact algorithms such as branch-and-bound can be impractical, and most metaheuristics for VRPs also converge slowly and cannot produce a satisfactory solution within a short response time.

To efficiently solve the problem, we adopt the WWO metaheuristic [3], which uses a small-size population (typically of 5–10 solutions) and hence requires low computational resources. We adapt the operators of WWO to evolve solutions to the problem, and develop a gradual neighborhood search method to improve solution accuracy. Algorithm 1 presents the pseudocode of the algorithm framework and the following subsections describe its operators in detail.

Algorithm 1: The hybrid WWO and neighborhood search algorithm for the quarantine vehicle routing problem.

```

1 Use the procedures described in Section 4.1 to initialize a population  $P$  of solutions;
2 while the stopping condition is not met do
3   foreach solution  $Z = (X, Y)$  in the population do
4     Calculate the wavelength of  $Z$  according to Equation (17);
5     for  $d = 1$  to  $N_X$  do
6       Let  $\hat{d} = \text{rand}(1, n_X)$ ;
7       Reverse the subsequence  $X[d, \hat{d}]$ ;
8     Call Algorithm 5 to improve the direction vector  $Y$  according to the new sequence vector  $X$ ;
9     Call Algorithm 4 to improve the new solution  $Z$  by balancing the areas among the subsequences;
10    if  $Z$  is better than the original solution then
11      Replace the original solution with  $Z$  in the population;
12      if  $Z$  is the newly found best solution then
13        for  $i = 1$  to  $m$  do
14          for  $k = 1$  to  $\lceil \alpha n_i \rceil$  do
15            Clone  $Z$  to a new solution  $Z'$ ;
16            Swap two randomly selected areas in  $x_i$ ;
17            if  $Z'$  is better than the  $Z$  then
18              Let  $Z = Z'$ ;
19    Update the population size;
20 return the best solution found so far.

```

4.1. Solution Encoding, Initialization, and Improving

In the proposed hybrid algorithm, each solution is represented by two vectors, $X = \{x_1, x_2, \dots, x_m\}$ (which we call the sequence vector) and $Y = \{y_1, y_2, \dots, y_m\}$ (which we call the direction vector), each of which consists of m sub-vectors. To split two adjacent subsequences in X , we insert a symbol '0' between them.

The algorithm starts by randomly initializing a population P of solutions. To create a random solution, we first generate X as a random permutation of n areas and then randomly select $m - 1$ positions to insert separators '0', and thus obtain m subsequence vectors for m vehicles. As an area may be assigned to multiple vehicles, we check X and select all areas satisfying $b_{x_{i,j}} < c_i$, the set of which is denoted by A^\dagger . For each $a_j \in A^\dagger$, let $c(j)$ be the total capacity of vehicles to which a_j has been assigned, we use $(b_j - c(j)) / b_j$ as the probability that a_j will be assigned to more vehicles, and this probability-based assignment is recursively applied until there are no remaining individuals in a_j , as shown in Algorithm 2.

Next, we generate Y by using the procedure shown in Algorithm 3 to determine the direction $y_{i,j}$ on each area $x_{i,j}$ ($1 \leq i \leq m$). That is, the probability of the vehicle v_i goes from a_j to the isolated region is proportional to its loading rate (a fully-loaded vehicle must go to the isolated region).

Usually, a randomly generated solution has much room for improvement. An obvious case is that when a vehicle has completed its job, another one still has many areas to visit. We use the greedy procedure shown in Algorithm 4 to tentatively improve a solution by continually moving an area from the sequence of the vehicle with the maximum completion time to that of the vehicle with the minimum completion time.

Algorithm 2: The procedure for assigning the individuals in an area to probably more than one vehicle.

Input: area a_j and vehicle v_i to which a_j is initially assigned

- 1 Let $b'_j = b_j - c_i, V(a_j) = \{v_i\}$;
- 2 **while** $b'_j > 0$ **do**
- 3 **if** $rand() > b'_j/b_j$ **then**
- 4 **return**;
- 5 **else**
- 6 Randomly select a vehicle $v_{i'}$ from $V \setminus V(a_j)$;
- 7 Insert a_j into $x_{i'}$;
- 8 $b'_j \leftarrow b'_j - c_{i'}$;
- 9 $V(a_j) \leftarrow V(a_j) \cup \{v_{i'}\}$;

Algorithm 3: The procedure for generating the direction vector.

Input: subsequence x_i and the beginning dimension j_0 (set to 1 by default)

- 1 Let $j = j_0$;
- 2 **while** $j < n_i - 1$ **do**
- 3 **if** $b_{x_{i,j}} = c_i$ **then**
- 4 $y_{i,j} \leftarrow 1$;
- 5 **else if** $b_{x_{i,j}} < c_i$ **and** $rand() < b_{x_{i,j}}/c_i$ **then**
- 6 $y_{i,j} \leftarrow 1$;
- 7 **else**
- 8 **if** $rand() < 0.5$ **then** $y_{i,j} \leftarrow 1$;
- 9 **else** $y_{i,j} \leftarrow 0$;
- 10 $j \leftarrow j + 1$;

Algorithm 4: The procedure that improves a solution by balancing areas among vehicles.

- 1 **while** *true* **do**
- 2 Let v_{i_L} be the vehicle with the minimum completion time and v_{i_U} be that with the maximum completion time;
- 3 Among all areas that are assigned to v_{i_U} but have not been visited by v_{i_U} before the completion time of v_{i_L} , select an area a_j with the minimum $t_{i_U}(j)$ calculated as follows:

$$t_{i_U}(j) = \begin{cases} \tau_{i_U, m_{i_U}} + t_{i_U, x_{i_U, m_{i_U}}} j' & c_{i_U, m_{i_U}} \geq b_j \\ \tau_{i_U, m_{i_U}} + t'_{i_U, x_{i_U, m_{i_U}}} + t'_{i_U, j'} & \text{otherwise} \end{cases}$$
- 4 Move a_j from x_{i_U} to the end of x_{i_L} ;
- 5 **if** the completion time of v_{i_L} is later than that of v_{i_U} **then**
- 6 Undo the last movement and **return**;

4.2. Water Wave Optimization for Evolving Solutions

WVO is a metaheuristic that takes inspiration from natural wave motions and has exhibited state-of-the-art performance in solving a variety of continuous, combinatorial, and constrained optimization problems [3,44,45]. In WVO, a solution is analogous to a wave, which has a wavelength inversely proportional to the solution fitness. The better (worse) the solution, the smaller (larger) the range in which the solution explores. As the problem is to minimize the objective function

Equation (13), we employ the following equation to calculate the wavelength $\lambda(Z)$ of each solution $Z = (X, Y)$:

$$\lambda(Z) = \frac{f(Z)}{\sum_{Z' \in P} f(Z')} \quad (17)$$

The original WWO has three evolutionary operators: propagation, refraction, and breaking. Here, we adopt the strategy of improved WWO that discards refraction and reduces the population size with an increasing number of generations. We adapt the propagation operator to evolve a solution $Z = (X, Y)$ to this problem as follows: let $n_X = \sum_{i=1}^m n_i$ be the total length of X , for each dimension $d < n_X$, with a probability of $\lambda(Z)$, reverse a subsequence from dimension d , where the length of the subsequence is randomly set between $[1, n_X - d]$. Note that if the subsequence does not contain a separator, then the reversal only changes the route of one vehicle, but does not change the set of areas assigned to the vehicle; otherwise, it changes the routes of and reassigns the areas among two or more vehicles; in particular, after a reversal, it is probable that an area occurs in a subsequence more than once, and in such a case we remove the duplicates. The expected number of reversal on the solution is $\lambda(Z)n_X$, so a high (low) fitness solution will be changed to a short (large) extent. If the best resulting solution is better, it will replace the original one in the next generation. The pseudocode of the propagation operation is given in Lines 5–7 of Algorithm 1.

The breaking operator is conducted only on each newly found best solution $Z^* = (X^*, Y^*)$. It tries to improve each subsequence x_i^* by performing a local search for $\lceil \alpha n_i \rceil$ times, each swapping two randomly selected areas in x_i^* , where α is a control parameter less than 1. Among $\lceil \alpha n_i \rceil$ swaps, the best resulting one, if decreasing the total exposure time, is retained in x_i^* . The pseudocode of the breaking operation is given in Lines 14–19 of Algorithm 1.

4.3. Gradual Neighborhood Search for Direction Improving

The purpose of the gradual neighborhood search method is to fine-tune the direction vector Y of each solution $Z = (X, Y)$. For each sub-vector y_i , the method gradually performs at most $(n_i - 1)$ neighborhood search operations, each changing the decision $y_{i,j}$ (if the current capacity of v_i allows) and then updating the subsequent dimensions from $j + 1$ to $n_i - 1$ according to the heuristic described in Algorithm 3. If the best neighboring sub-vector is better than the original sub-vector y_i , it will replace y_i in the solution. Algorithm 5 presents the pseudocode of the gradual neighborhood search method.

Algorithm 5: The gradual neighborhood search method for improving the direction vector of a solution.

```

1 for  $i = 1$  to  $m$  do
2   Let  $y'_i = y_i$ ;
3   for  $j = 1$  to  $n_i - 1$  do
4     if  $y_{i,j} = 1$  and  $c_{i,j} \geq c_i$  then
5       continue;
6      $y_{i,j} \leftarrow 1 - y_{i,j}$ ;
7     Call Algorithm 3 to regenerate the subsequent dimensions from  $j + 1$  to  $n_i - 1$ ;
8     if the total exposure time of  $x_i$  with  $y'_i$  is better then
9        $y_i \leftarrow y'_i$ ;

```

5. Results and Discussion

We test the proposed method on seven real-world instances of vehicle routing for transferring high-risk individuals in Hangzhou, China, from 28 January to 3 February 2020, the peak period of the COVID-19 epidemic in the city. Table 2 summarizes the basic characteristics of the seven instances.

Table 2. Summary of the basic characteristics of the seven real-world instances. \bar{b} : average number of individuals per area; K_V : number of types of vehicles; \bar{c} : the average capacity of vehicles; \bar{t} : average travel time between two areas (in minutes).

ID	n	m	\bar{b}	K_V	\bar{c}	\bar{t}
J28	27	6	13.9	2	11.0	44.5
J29	60	9	10.8	2	11.0	71.3
J30	127	16	10.4	4	10.5	60.7
J31	191	16	8.4	4	10.5	35.7
F01	93	15	10.6	4	10.8	54.2
F02	102	15	10.3	4	10.8	39.3
F03	98	12	8.2	4	9.3	48.2

To validate the performance of the hybrid WWO and neighbor search algorithm, we compare it with the following five methods:

- A greedy heuristic method, which always makes each vehicle move to the closest area with untransferred individuals. This method was used by most emergency management officers in the past.
- A standard integer programming algorithm implemented by CPLEX solver (version 12.6, with the automatic cut generator switched on).
- An ACO algorithm based on [19], which uses an artificial ant to construct and iteratively improve the tour of each vehicle.
- A memetic algorithm combining GA and local search for multi-trip VRP, in which a vehicle can go back to the depot and be re-loaded [46]. We adapt the algorithm for our problem by changing the objective function from the total travel time to the total exposure time.
- A tabu search algorithm for VRP with split deliveries and pickups [47].

The computational environment is a workstation with an i7-6500 2.5GH CPU, 8GB DDR4 RAM, and an NVIDIA Quadro M500M card. In each instance, as it is required to produce a solution within ten minutes, the maximum CPU time of each algorithm is set to 600 seconds. The last four algorithms are stochastic; in a real-world application, we typically run an algorithm once, or simultaneously run several instances of the algorithm with different random seeds and submit the best solution obtained to the decision-maker. However, to make the comparison more objective, we make up the number of runs of the ACO, memetic, tabu search, and WWO algorithms to 30 after the application.

Table 3 presents the results in terms of the average exposure time (in minutes) obtained by the algorithms on the test instances. For the last four algorithms, we present both mean value and standard deviation (in parenthesis). Figure 3 also illustrates the comparative results, where the maximum and minimum values obtained by WWO are given by error lines. As we can observe, the proposed algorithm achieves the shortest average exposure time (shown in bold in Table 3) on each instance. On instance J28 where the numbers of areas and high-risk individuals are relatively small, the differences among the algorithms are not significant, and the result of CPLEX is the second best. However, with an increasing number of areas/individuals, the performance of CPLEX becomes the worst among all algorithms on the six remaining instances. This is because, for large instances, the exact algorithm of CPLEX can only explore a very small fraction of the solution space within the limited running time. The ACO algorithm performs the worst on instance J28, but it outperforms Greedy and CPLEX on the remaining instances. The memetic algorithm performs the second-worst on J28, the second-best on J29, and the third-best (below tabu search and WWO) on the last four instances. This demonstrates that the population-based ACO and memetic metaheuristics can explore the large solution space in an efficient (if not the most efficient) way. The performance of the greedy heuristic is unstable, worse than that of CPLEX on J28 and F01 while better than that of CPLEX on the remaining instances. That is, on some specific instances, the greedy heuristic can sometimes reach a good solution, but its general performance is far from satisfactory. The tabu search algorithm starts from a greedy

solution and then iteratively improves it. It performs relatively well on medium-size instances F01, F02, and F03; on small-size instances, its performance heavily relies on the unstable initial solution; on large instances, it cannot effectively explore the large solution spaces as the population-based metaheuristics. The average exposure time obtained by the greedy heuristic method used by the organization in the past is 337 minutes, while that of the proposed WWO-based algorithm is 206 minutes, which demonstrates that the proposed algorithm can significantly increase the efficiency of epidemic control.

Table 3. Results of the comparative algorithms on the test instances.

ID	Greedy	CPLEX	ACO	Memetic	Tabu Search	WWO
J28	220	213	236 (14.19)	223 (5.11)	218 (8.16)	203 (5.87)
J29	263	284	212 (20.75)	192 (3.88)	208 (9.15)	175 (7.20)
J30	351	384	338 (24.52)	295 (10.97)	333 (16.25)	277 (13.49)
J31	363	475	252 (18.39)	226 (12.73)	253 (18.20)	191 (9.28)
F01	416	397	261 (23.30)	257 (15.45)	249 (17.03)	242 (10.07)
F02	306	353	226 (16.06)	207 (14.66)	187 (15.05)	152 (5.25)
F03	390	448	267 (20.40)	251 (17.80)	245 (28.46)	215 (6.70)

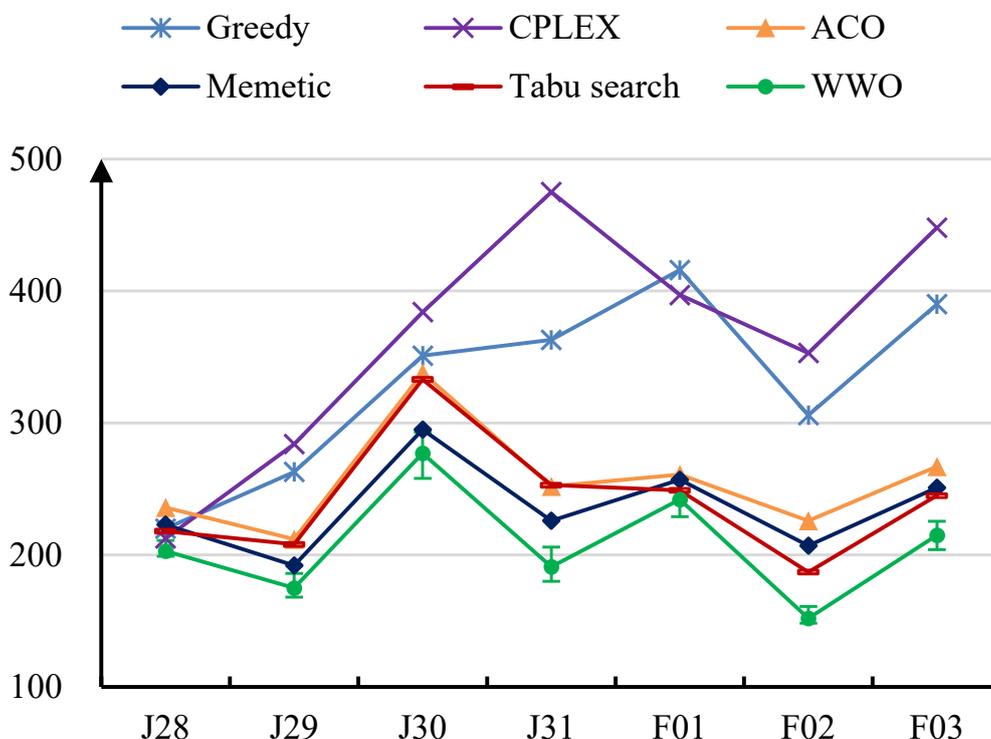


Figure 3. Comparative results of the algorithms on the test instances. The horizontal axis denotes the test instance, and the vertical axis denotes the average exposure time (in minutes).

Inevitably, the exposure time calculated based on the objective function of a solution is not equal to the actual exposure time obtained by the implementation of the solution. Thereby, we compare the calculated and actual exposure times on the test instances in Figure 4, from which we can observe that, except on F01 the calculated time is longer than the actual time, on each remaining instance the actual time is longer. This is because the solutions obtained by our algorithm have high qualities, i.e., they schedule the vehicles in an efficient and “compact” way; when a vehicle is delayed by an unexpected incident, many areas (especially those requiring multiple rounds of transfer) may be seriously affected, which can cause a significant increase of the total exposure time. Nevertheless, the calculated time does not deviate much from the actual time, which demonstrates the feasibility and accuracy of the solution produced by our approach.

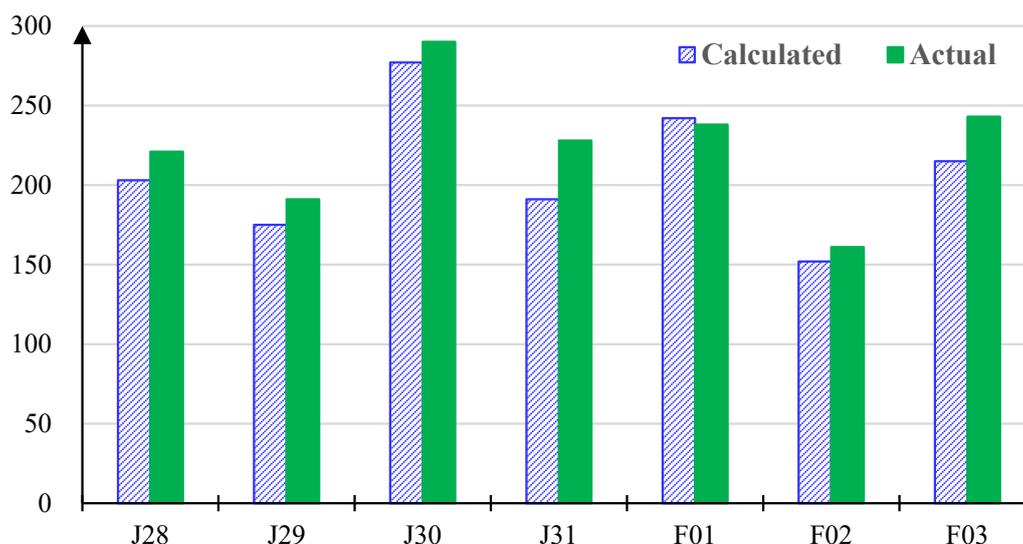


Figure 4. Comparison of the calculated and actual average exposure times on the test instances. The horizontal axis denotes the test instance, and the vertical axis denotes the average exposure time (in minutes).

In summary, the application of our method can be divided into the following steps.

- (1) In preparation for an epidemic, establish a database to manage the information of quarantine vehicles, services for medical isolation, and routes from most densely populated areas to the isolated regions;
- (2) Whenever there are requirements for high-risk individual transfer, input the number and distribution of individuals, and run the algorithm;
- (3) Arrange staffs to prepare the vehicles (typically, during the execution of the algorithm);
- (4) Within the response time, take the best-known solution found by the algorithm, and send it to the staffs for implementation;
- (5) Monitor the operations; if the implementation deviates from the plan, or new requirement comes, adjust the current solution or run the algorithm to solve the new instance.

In practice, we must pay attention to the changes in information, including traffic conditions, vehicle conditions, and the number of high-risk individuals, and appropriately respond to the changes. For example, if the number of individuals in an area increases a small amount during the implementation of an existing solution, we typically make the vehicle for this area to transfer the new individuals and, if the vehicle is delayed too much, the last area(s) of the vehicle can be reassigned to other vehicle(s) that have completed their jobs. However, if the number of individuals increases a large amount, we often need to construct and solve a new instance for the remaining tasks.

6. Conclusions

Transferring high-risk individuals to an isolated region in a timely manner is critical for epidemic control. In this paper, we present a problem of scheduling quarantine vehicles to transfer high-risk individuals from a set of dispersed areas to an isolated region, which is more difficult than most existing VRPs. To solve this problem, we propose a hybrid WWO and neighborhood search algorithm, which adapts the WWO metaheuristic to efficiently explore the solution space and utilizes neighborhood search to improve the solution accuracy. Computational results demonstrate that the proposed algorithm significantly outperforms several existing algorithms and obtains high-quality solutions on the test instances.

Currently, our problem assumes that available drivers for quarantine vehicles are sufficient. During the peak period of COVID-19, we found that this assumption holds for the instances in

Hangzhou and many other cities in China, but does not hold for that in Wuhan, the epidemic center where the number of high-risk individuals is huge and hence a vehicle should have several drivers in rotation. Therefore, an ongoing study is to include the scheduling of drivers in our problem. Our future work will integrate more other components, such as combining the use of electric and fuel vehicles [48], early primary screening and automatic grading of individuals [49], to enhance the effectiveness of epidemic control.

Author Contributions: Conceptualization, M.-X.Z. and Y.-J.Z.; methodology, Y.-J.Z.; software, H.-F.Y. and J.-Y.W.; validation, M.-X.Z.; formal analysis, H.-F.Y.; investigation, M.-X.Z.; data curation, H.-F.Y.; writing—original draft preparation, M.-X.Z.; writing—review and editing, J.-Y.W.; supervision, Y.-J.Z.; funding acquisition, Y.-J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China under Grant 61872123 and Zhejiang Provincial Outstanding Youth Science Foundation under Grant LR20F030002.

Acknowledgments: The authors would like to thank Zhejiang Provincial Center for Disease Control and Prevention for information sharing.

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

Abbreviations

The following abbreviations are used in this manuscript:

VRP	vehicle routing problem
COVID-19	novel coronavirus pneumonia
GA	genetic algorithm
ACO	ant colony optimization
WWO	water wave optimization

References

1. Eksioglu, B.; Vural, A.V.; Reisman, A. The vehicle routing problem: A taxonomic review. *Comput. Ind. Eng.* **2009**, *57*, 1472–1483. [\[CrossRef\]](#)
2. Hussain, K.; Mohd Salleh, M.N.; Cheng, S.; Shi, Y. Metaheuristic research: a comprehensive survey. *Artif. Intell. Rev.* **2019**, *52*, 2191–2233. [\[CrossRef\]](#)
3. Zheng, Y.J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [\[CrossRef\]](#)
4. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [\[CrossRef\]](#)
5. Solomon, M.M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **1987**, *35*, 254–265. [\[CrossRef\]](#)
6. Bräysy, O.; Dullaert, W.; Gendreau, M. Evolutionary algorithms for the vehicle routing problem with time windows. *J. Heuristics* **2004**, *10*, 587–611. [\[CrossRef\]](#)
7. Potvin, J.Y. State-of-the art review—Evolutionary algorithms for vehicle routing. *INFORMS J. Comput.* **2009**, *21*, 518–548. [\[CrossRef\]](#)
8. Prodhon, C.; Prins, C. *Metaheuristics for Vehicle Routing Problems*; Springer: Cham, Switzerland, 2016; pp. 407–437. [\[CrossRef\]](#)
9. Baker, B.M.; Ayechev, M. A genetic algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2003**, *30*, 787–800. [\[CrossRef\]](#)
10. Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2004**, *31*, 1985–2002. [\[CrossRef\]](#)
11. Ghoseiri, K.; Ghannadpour, S.F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Appl. Soft Comput.* **2010**, *10*, 1096–1107. [\[CrossRef\]](#)
12. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **2013**, *40*, 475–489. [\[CrossRef\]](#)
13. Xu, D.; Xiao, R. An improved genetic clustering algorithm for the multi-depot vehicle routing problem. *Int. J. Wirel. Mob. Comput.* **2015**, *9*, 1–7. [\[CrossRef\]](#)

14. Zhu, Q.; Qian, L.; Li, Y.; Zhu, S. An improved particle swarm optimization algorithm for vehicle routing problem with time windows. In Proceedings of the 2006 IEEE Int'l Conference Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 1386–1390. [CrossRef]
15. Ai, T.J.; Kachitvichyanukul, V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.* **2009**, *36*, 1693–1702. [CrossRef]
16. Zhang, J.; Yang, F.; Weng, X. An evolutionary scatter search particle swarm optimization algorithm for the vehicle routing problem with time windows. *IEEE Access* **2018**, *6*, 63468–63485. [CrossRef]
17. Marinakis, Y.; Marinaki, M.; Migdalas, A. A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. *Inform. Sci.* **2019**, *481*, 311–329. [CrossRef]
18. Wu, H.; Tao, F.; Qiao, Q.; Zhang, M. A chance-constrained vehicle routing problem for wet waste collection and transportation considering carbon emissions. *Int. J. Environ. Res. Public Health* **2020**, *17*, 458. [CrossRef] [PubMed]
19. Bell, J.E.; McMullen, P.R. Ant colony optimization techniques for the vehicle routing problem. *Adv. Eng. Inform.* **2004**, *18*, 41–48. [CrossRef]
20. Yu, B.; Yang, Z.Z.; Yao, B. An improved ant colony optimization for vehicle routing problem. *Eur. J. Oper. Res.* **2009**, *196*, 171–176. [CrossRef]
21. Wu, W.; Tian, Y.; Jin, T. A label based ant colony algorithm for heterogeneous vehicle routing with mixed backhaul. *Appl. Soft Comput.* **2016**, *47*, 224–234. [CrossRef]
22. Wu, L.; He, Z.; Chen, Y.; Wu, D.; Cui, J. Brainstorming-based ant colony optimization for vehicle routing with soft time windows. *IEEE Access* **2019**, *7*, 19643–19652. [CrossRef]
23. Huang, Y.H.; Blazquez, C.A.; Huang, S.H.; Paredes-Belmar, G.; Latorre-Nuñez, G. Solving the feeder vehicle routing problem using ant colony optimization. *Comput. Ind. Eng.* **2019**, *127*, 520–535. [CrossRef]
24. Szeto, W.Y.; Wu, Y.; Ho, S.C. An artificial bee colony algorithm for the capacitated vehicle routing problem. *Eur. J. Oper. Res.* **2011**, *215*, 126–135. [CrossRef]
25. Zhang, S.; Lee, C.K.M.; Choy, K.L.; Ho, W.; Ip, W.H. Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transp. Res. Part D* **2014**, *31*, 85–99. [CrossRef]
26. Ng, K.K.H.; Lee, C.K.M.; Zhang, S.Z.; Wu, K.; Ho, W. A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion. *Comput. Ind. Eng.* **2017**, *109*, 151–168. [CrossRef]
27. Zheng, Y.J.; Ling, H.F.; Xue, J.Y. Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations. *Comput. Oper. Res.* **2014**, *50*, 115–127. [CrossRef]
28. Berghida, M.; Boukra, A. EBBO: an enhanced biogeography-based optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Int. J. Adv. Manuf. Technol.* **2015**, *77*, 1711–1725. [CrossRef]
29. Zheng, Y.; Ling, H.; Xue, J.; Chen, S. Population classification in fire evacuation: A multiobjective particle swarm optimization approach. *IEEE Trans. Evol. Comput.* **2014**, *18*, 70–81. [CrossRef]
30. Zheng, Y.J.; Chen, S.Y.; Ling, H.F. Evolutionary optimization for disaster relief operations: A survey. *Appl. Soft Comput.* **2015**, *27*, 553–566. [CrossRef]
31. Daskin, M.; Haghani, A. Multiple vehicle routing and dispatching to an emergency scene. *Environ. Plan. A* **1984**, *16*, 1349–1359. [CrossRef]
32. Haghani, A.; Tian, Q.; Hu, H. Simulation model for real-time emergency vehicle dispatching and routing. *Transp. Res. Rec.* **2004**, *1882*, 176–183. [CrossRef]
33. Vargas-Villamil, F.D. Vehicle routing and dispatching for emergency personnel evacuation from off-shore oil platforms. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 4–16 June 2006; p. 6. [CrossRef]
34. Wang, C.X.; Deng, X.M. Multi-depot emergency vehicle routing and transportation optimization with fuzzy variables. *J. Syst. Manag.* **2011**, *20*. Available online: http://en.cnki.com.cn/Article_en/CJFDTotal-XTGL201103005.htm (accessed on 15 March 2020).
35. Wohlgenuth, S.; Oloruntoba, R.; Clausen, U. Dynamic vehicle routing with anticipation in disaster relief. *Socio-Econ. Plan. Sci.* **2012**, *46*, 261–271. [CrossRef]

36. Chini, G.; Poli, C.; Oddi, G.; Pietrabissa, A.; Grigioni, M. Receding horizon multi-vehicle routing for emergency scenarios. In Proceedings of the 22nd Mediterranean Conference Control and Automation, Palermo, Italy, 16–19 June 2014; pp. 374–379. [[CrossRef](#)]
37. Ye, Z.; Yin, Y.; Zong, X.; Wang, M. An optimization model for evacuation based on cellular automata and ant colony algorithm. In Proceedings of the 7th International Symposium on Computational Intelligence and Design, Hangzhou, China, 13–14 December 2014; Volume 1, pp. 7–10. [[CrossRef](#)]
38. Shahparvari, S.; Abbasi, B. Robust stochastic vehicle routing and scheduling for bushfire emergency evacuation: An Australian case study. *Transp. Res. Part A* **2017**, *104*, 32–49. [[CrossRef](#)]
39. Sun, Y.; Zhang, L.; Fei, T.; Li, Y. Application of crossover mutation ant colony algorithm in emergency logistics vehicle routing problem. In Proceedings of the 2012 International Conference on Systems and Informatics, Yantai, China, 19–20 May 2012; pp. 86–89. [[CrossRef](#)]
40. Chen, M. Improved genetic algorithm for emergency logistics distribution vehicle routing problems. In Proceedings of the IEEE International Conference on Security, Pattern Analysis, and Cybernetics, Wuhan, China, 18–19 October 2014; pp. 385–388. [[CrossRef](#)]
41. Du, M.; Yi, H. Research on multi-objective emergency logistics vehicle routing problem under constraint conditions. *J. Ind. Eng. Manag.* **2013**, *6*, 258–266. [[CrossRef](#)]
42. Wang, X.; Choi, T.M.; Liu, H.; Yue, X. A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 545–556. [[CrossRef](#)]
43. Fallahi, A.E.; Sefrioui, I. A linear programming model and memetic algorithm for the Emergency Vehicle Routing. In Proceedings of the 4th World Conference on Complex Systems (WCCS), Ouarzazate, Morocco, 22–25 April 2019; pp. 1–5. [[CrossRef](#)]
44. Zheng, Y.J.; Lu, X.Q.; Du, Y.C.; Xue, Y.; Sheng, W.G. Water wave optimization for combinatorial optimization: Design strategies and applications. *Appl. Soft Comput.* **2019**, *83*, 105611. [[CrossRef](#)]
45. Singh, G.; Rattan, M.; Gill, S.S.; Mittal, N. Hybridization of water wave optimization and sequential quadratic programming for cognitive radio system. *Soft Comput.* **2019**, *23*, 7991–8011. [[CrossRef](#)]
46. Cattaruzza, D.; Absi, N.; Feillet, D.; Vidal, T. A memetic algorithm for the Multi Trip Vehicle Routing Problem. *Eur. J. Oper. Res.* **2014**, *236*, 833–848. [[CrossRef](#)]
47. Qiu, M.; Fu, Z.; Eglese, R.; Tang, Q. A Tabu Search algorithm for the vehicle routing problem with discrete split deliveries and pickups. *Comput. Oper. Res.* **2018**, *100*, 102–116. [[CrossRef](#)]
48. Liao, W.; Liu, L.; Fu, J. A comparative study on the routing problem of electric and fuel vehicles considering carbon trading. *Int. J. Environ. Res. Public Health* **2019**, *16*, 3120. [[CrossRef](#)]
49. Song, Q.; Zheng, Y.J.; Yang, J. Effects of Food Contamination on Gastrointestinal Morbidity: Comparison of Different Machine-Learning Methods. *Int. J. Environ. Res. Public Health* **2019**, *16*, 838. [[CrossRef](#)] [[PubMed](#)]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).