International Journal of Engineering and Technology Innovation, vol. 14, no. 1, 2024, pp. 67-84

A Study on the Vehicle Routing Problem Considering Infeasible Routing Based on the Improved Genetic Algorithm

Xiao-Yun Jiang*, Wen-Chao Chen, Yu-Tong Liu

School of Economics and Management, Xiamen University of Technology, Fujian, China

Received 18 July 2023; received in revised form 13 September 2023; accepted 14 September 2023

DOI: https://doi.org/10.46604/ijeti.2023.12612

Abstract

The study aims to optimize the vehicle routing problem, considering infeasible routing, to minimize losses for the company. Firstly, a vehicle routing model with hard time windows and infeasible route constraints is established, considering both the minimization of total vehicle travel distance and the maximization of customer satisfaction. Subsequently, a Floyd-based improved genetic algorithm that incorporates local search is designed. Finally, the computational experiment demonstrates that compared with the classic genetic algorithm, the improved genetic algorithm reduced the average travel distance by 20.6% when focusing on travel distance and 18.4% when prioritizing customer satisfaction. In both scenarios, there was also a reduction of one in the average number of vehicles used. The proposed method effectively addresses the model introduced in this study, resulting in a reduction in total distance and an enhancement of customer satisfaction.

Keywords: vehicle routing problem, infeasible routing, hard time window, genetic algorithm

1. Introduction

Delivery is a crucial activity for logistics companies. Goods are typically transported from suppliers to distribution centers and then delivered to customers. The vehicle routing problem (VRP) was first introduced by Dantzig and Ramser [1] in 1959. It involves a distribution center and multiple customer locations, requiring the scheduling of a certain number of vehicles and the design of appropriate delivery routes that ensure orderly delivery to each customer location before returning to the distribution center. The objective of VRP is to organize efficient delivery routes that minimize travel distance, cost, and time while meeting customer demands. In 1987, Solomon [2] extended the problem to include time window constraints, known as the vehicle routing problem with time windows (VRPTW).

At the same time, he proved the NP-hardness property of VRPTW. In VRPTW, delivery vehicles are required to provide satisfactory service to customers within predefined time windows. Each customer location is assigned a time window that specifies the earliest and latest time for service commencement, ensuring that vehicles start serving customers within these time windows. These time windows are known as hard time windows. However, due to the uncertainty of road transportation, unexpected road conditions may render part of the path infeasible, making it impossible to implement pre-planned distribution routes. This can lead to vehicles being unable to provide services to customers within the specified time windows. Therefore, the key challenge in enterprise logistics delivery lies in the rational planning of delivery routes and the subsequent optimization of vehicle routing when routes are infeasible. This is essential to ensuring satisfactory customer service within the designated time windows.

* Corresponding author. E-mail address: jxycom@163.com

English language proofreader: Chih-Wen Teng

Most of the current VRP research has focused on deterministic environments. Cappanera et al. [3] generalized the VRP that integrates precedence and synchronization constraints. The scholar also studied a model and some techniques in the study. Wang et al. [4] developed an improved algorithm based on improved ant colony optimization to address a periodic VRPTW and service choice. Zhen et al. [5] investigated VRPTW and release dates in the context of multiple depots and multiple trips. The problem is formulated as a mixed integer programming model, and a hybrid particle swarm optimization algorithm and a hybrid genetic algorithm are developed to address the problem. Neira et al. [6] studied two integer programming models for the VRPTW, covering multiple trips involved, loading time dependency, and limited trip duration.

Bortfeldt and Yi [7] discuss the solution to minimize the number of vehicles used, the total travel distance, and the waiting time under the constraints of capacity and time window. They also proposed a real-world penalty function for addressing this problem. Yao et al. [8] extended the original VRPTW by introducing a high-dimensional spatio-temporal network flow model to enhance the realism of the model. Hien et al. [9] proposed a genetic algorithm inspired by the greedy search for the large-scale electric vehicle routing problem.

Through experiments, it is found that the proposed algorithm can better find clustered charging routes with more optimal travel distances. Zulvia et al. [10] studied a green VRP model for perishable goods that considers different traveling times and working hours. A many-objective algorithm optimizes cost and customer satisfaction. Dominguez et al. [11] investigated a partially randomized heuristic algorithm based on VRP and two-dimensional packing and tested the efficiency of their algorithm through experiments.

The VRP problem in uncertain environments has been less studied. Cai et al. [12] presented a dynamic VRPTW solved with a dynamical algorithm. Salavati-Khoshghalb et al. [13] worked on solving the VRP with stochastic demands through a hybrid recourse policy. It is a hybrid policy that predetermines a maximum proceeding threshold and a minimum restocking threshold. If the customer demand value is higher than the latter, a return trip will be executed to the depot; otherwise, the vehicle will proceed to serve the next customer. Xu et al. [14] investigate the distribution of emergency relief for electric vehicles (EVs).

The objectives are to find routes for EVs that meet all shelters within their respective time windows and minimize the total cost. It proposes a two-stage solution method. In the first stage, the minimum travel cost between any two vertices is obtained, and in the second stage, a genetic algorithm is applied to obtain the distribution scheme. Kim [15] proposed a dynamic VRP with fuzzy customer responses from the customer's perspective. The customer response is represented by fuzzy rules in the model. The obtained routing strategy can effectively reduce customer complaints and avoid losing potential customers.

In summary, most studies on VRPTW have focused on deterministic environments, assuming fixed road conditions and customer time windows throughout the delivery process. These studies have primarily addressed constraints related to time windows, limited battery capacity of EVs, and cargo loading. However, real-world delivery processes occur in uncertain environments where road conditions and customer time windows may vary. Existing literature has paid relatively less attention to modeling VRPTW in uncertain environments, particularly when routes are infeasible.

The objective of this study is to optimize vehicle routing when partial routes are infeasible, aiming to minimize losses for the enterprise. To achieve this objective, a mixed-integer linear programming (MILP) model for VRPTW is established. This model considers both infeasible route constraints and hard time window constraints for customers, with the goals of maximizing customer satisfaction and minimizing total vehicle travel distance. Based on the characteristics of the proposed model, a Floyd-based improved genetic algorithm is designed to solve the model. This algorithm incorporates local search operations to enhance search efficiency. Finally, computational experiments are conducted to validate the effectiveness of the proposed approach, demonstrating its capability to effectively optimize logistics delivery routes, particularly in scenarios involving infeasible routes. The results show reduced total vehicle travel distance and increased customer satisfaction.

The remaining sections of this study are organized as follows: Section 2 presents the problem description and modeling. Section 3 presents the research methodology. Section 4 discusses the computation experiments and analysis. Finally, Section 5 presents the conclusions.

2. Problem Description and Modeling

This section presents the VRP model developed in this paper. The initial part provides an overview of the model, including an introduction to its key parameters. The subsequent section delves into the mathematical model of the VRP established in this study, encompassing assumptions, hard time windows, and the mathematical formulation.

2.1. Problem description

VRP is a mathematical abstraction of the reasonable allocation of distribution vehicles and travel routes in logistics distribution activities. It has been the frontier and a key research topic in the fields of combinatorial optimization and operations for the past two decades. VRP involves the systematic planning of routes and vehicle quantities, taking into account various constraints and restrictions such as load limits, customer service time, and customer demand. The ultimate objective is to achieve specific distribution goals such as minimizing costs, reducing total distance, or minimizing travel time.

Nowadays, although most logistics companies own complete system platforms and supporting logistics infrastructure, many distribution problems still exist. Firstly, there is a lack of systematic and effective distribution routing. During the distribution process, the choice of distribution route is mainly subjective. Especially when encountering a road accident unexpectedly, the driver often chooses a distribution route based on his personal experience and fails to achieve overall optimization. Secondly, new drivers who are unfamiliar with the distribution route may make a choice that requires them to travel longer distances, and the transportation cost is increased under this circumstance. A lack of reasonable distribution design results in frequent occurrences of late or early arrivals and failure to meet customer satisfaction.

Against this background, this paper aims to address the VRPTW considering infeasible routing in logistics distribution. The single-objective VRP is complicated by the constraint of unexpected traffic congestion and a new objective of customer satisfaction. This is the VRPTW considering infeasible routing.

A general distribution problem can be described as a mathematical model: There is a directed graph, which represents a set of all nodes; 0 and n + 1 represent distribution centers; 1, 2, ..., n represent customers; A represents a set of arcs. A reasonable distribution route must start at node 0 and finally return to node n + 1 in the directed graph. Table 1 and Table 2 respectively show the parameters and decision variables involved in the VRPTW model constructed in this paper. The overarching objective of the model is to meet the needs of logistics companies, to improve customer satisfaction as much as possible, and to identify the relatively optimum routing and vehicle routing scheme.

Table 1 Symbols of parameter

Parameter	Definition
v	Speed of distribution vehicle
c_{ij}	Distance between customer node *i* and *j*
t_{ij}	Travel time from customer node *i* to *j*
s_i	Service time of customer node *i*
at_i	Lower bound of the satisfaction time window of the customer *i*
bt_i	Upper bound of the satisfaction time window of the customer *i*
a_i	Lower bound of the time window of the customer node *i*
b_i	Upper bound of the time window of the customer node *i*
E_t	Lower bound of the time window of the distribution center

Table 1 Symbols of parameter (continued)

Parameter	Definition
L_t	Upper bound of the time window of the distribution center
d_i	Demand of the customer node *i*
C	Maximum carrying capacity of the distribution vehicles
M	A positive number large enough
V_i	node *i*
n	Number of customer nodes

Table 2 Symbols of set

Set	Definition
K	Set of the distribution vehicles
N_c	Set of the customer nodes
$\Delta_{-(j)}$	Set of arcs for which the vehicle returns to node *j*
$\Delta^{+}_{(i)}$	Set of arcs for which the vehicle starts from node *i*
D_{ij}	Set of arcs for which the routing between node *i* and *j* is infeasible

2.2. Mathematical model

To address the multi-objective VRPTW constructed in this paper, a mathematical model based on the following basic assumptions is established. The meanings of the symbols used in this paper are listed in Table 3.

(1) There is a one-way flow of goods, namely pure goods distribution.

(2) Both the start point and destination of the travel route are located in distribution centers.

(3) The location coordinates of distribution centers and customer nodes are known.

(4) The demand on every customer node is known.

(5) The vehicle has a loading capacity, and the maximum loading capacity is known.

(6) Each vehicle serves along one route only.

(7) Overloading is prohibited, namely not surpassing the maximum loading capacity.

(8) The demand at each customer node must be met.

(9) The vehicle runs at a constant speed.

(10) The traffic resources on the road are not considered.

(11) The vehicle must serve each customer node within the customer's time window and return to the distribution center within its time window. Waiting time is required if the vehicle arrives in advance.

Table 3 Symbols of parameter

Variable	Definition
w_{ik}	The time when vehicle *k* starts serving node *i*
x_{ijk}	Whether vehicle *k* heads for node *j* from node *i*, if yes, x_{ijk} = 1; if not, x_{ijk} = 0
S_i	Satisfaction of customer *i*

In logistics distribution, providing timely service for customers is a key factor that influences customer satisfaction, which will decrease as the gap between real and expected service time widens. The traditional method for quantifying customer satisfaction is evaluating it on a scale from 0 to 1 [16] representing the maximum and minimum levels, respectively. This is a relatively simple method. However, in the multi-objective case, dimensions between different objectives should be considered because they will result in a significantly different optimization effect.

This paper uses hard time windows and improves the traditional quantification method of customer satisfaction. For the problem, 10 and 0 represent the maximum and minimum levels of customer satisfaction, respectively. A satisfaction time window is embedded into the customer's hard time window, and customer satisfaction is measured by the inner time window. Fig. 1 shows how the two-time windows are related to customer satisfaction. a_i represents the lower bound of the time window of customer node i, and b_i represents the upper bound of the time window, respectively, and the lower and upper bound of the satisfaction time windows are represented by at_i and bt_i, respectively. Therefore, a new way of calculating customer satisfaction is proposed in Fig. 1.

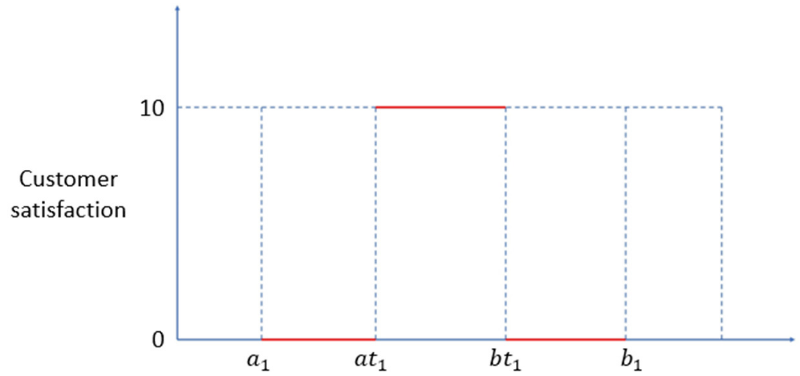

Fig. 1 Time windows related to customer satisfaction

The mathematical model of the VRPTW based on the assumptions above is constructed as follows:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij} x_{ijk} \tag{1}$$

Objective Eq. (1) represents the minimum total distance;

$$\max \sum_{i\in N_c} S_i \tag{2}$$

Objective Eq. (2) represents maximum customer satisfaction;

$$S.t. \sum_{k\in K} \sum_{j\in \Delta^+_{(j)}} x_{ijk} = 1,\ \forall i \in N \tag{3}$$

Constraint Eq. (3) restricts each customer to one route only;

$$\sum_{j\in \Delta^+_{(0)}} x_{0jk} = 1,\ \forall k \in K \tag{4}$$

$$\sum_{i\in \Delta_{-(n+1)}} x_{i,n+1,k} = 1,\ \forall k \in K \tag{5}$$

Constraints Eqs. (4) to (5) represent vehicle departs (ends) at the distribution center;

$$\sum_{i\in \Delta_{-(j)}} x_{ijk} - \sum_{j\in \Delta^+_{(j)}} x_{jik} = 0\ ,\ \forall k \in K,\ \forall j \in N \tag{6}$$

Constraint Eq. (6) represents the flow restriction for vehicle k on the route;

$$S_i \begin{cases} 0 & w_{ik} < at_i,\ w_{ik} > bt_i \\ 10 & at_i \le w_{ik} \le bt_i \end{cases} \tag{7}$$

Constraint Eq. (7) represents the piece-wise function of customer satisfaction;

$$w_{ik} + s_i + t_{ij} - w_{jk} \le \left(1 - x_{ijk}\right)M,\ \forall k \in K,\ \forall (i, j) \in A \tag{8}$$

Constraint Eq. (8) represents that the travel time of the vehicle *k* is continuous;

$$t_{ij} = \frac{c_{ij}}{v} \tag{9}$$

Constraint Eq. (9) represents that the travel time of the vehicle from node *i* to *j* is equal to the ratio of the distance to the vehicle speed;

$$c_{ij} \begin{cases} c_{ij} & (i, j) \notin D_{ij} \\ +\infty & (i, j) \in D_{ij} \end{cases} \tag{10}$$

Constraint Eq. (10) represents that the distance between two nodes is infinite when the route between them is infeasible;

$$a_i\Big(\sum_{i \in \Delta^+_{(i)}} x_{ijk}\Big) \le w_{ik} \le b_i\Big(\sum_{i \in \Delta^+_{(i)}} x_{ijk}\Big),\ \forall k \in K,\ \forall i \in N \tag{11}$$

Constraint Eq. (11) represents that vehicle *k* must start serving customer *i* between its lower and upper bound of time windows;

$$E_t \le w_{ik} \le L_t,\ \forall k \in K,\ \forall i \in \{0, n+1\} \tag{12}$$

Constraint Eq. (12) represents that vehicle *k* must leave (return to) the distribution center between its lower and upper bound of time windows;

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+_{(i)}} x_{ijk} \le C,\ \forall k \in K \tag{13}$$

Constraint Eq. (13) represents that the initial load of vehicle *k* in the distribution center must not be greater than its maximum capacity;

$$x_{ijk} \in \{0,1\},\ \forall k \in K,\ \forall (i, j) \in A \tag{14}$$

Constraint Eq. (14) represents 0-1 variable for whether vehicle *k* leaves node *i* for *j*. To address multi-objective functions, this paper uses a weighted sum approach to convert them into single-objective problems. Section 3.3.1 will provide further details on this process.

3. Research Methodology

In this section, an analysis of the VRP issues addressed in the paper is presented. Given the substantial complexity of this problem, the paper proposes the utilization of Floyd’s algorithm to enhance the genetic algorithm and introduces local search to augment the algorithm's search capabilities. Moreover, an analysis is undertaken concerning several crucial components within the improved genetic algorithm.

3.1. Genetic algorithm

Almost all VRPs and even VRP variants are NP-hard problems. NP-hard problems are hard to solve. This is why algorithms for solving VRP are studied extensively. In this study, the VRP is addressed with a proposed genetic algorithm, and an attempt is made to improve the algorithm.

An intelligent algorithm is an abstract summary of some structures and laws in nature [17]. The genetic algorithm is an adaptive global intelligent algorithm formed by biological heredity and evolution processes [18]. It is a random search method evolved from the evolutionary law of the biological community. Optimization is an evolutionary process starting from a single population to a better population based on the rule of survival of the fittest. During the evolutionary process, crossover and mutation occur thanks to genetic operators, and optimum individuals survive in the last after several iterations and become the approximate optimal solution to the problem.

The genetic algorithm mainly consists of five steps: initialization, selection, crossover, mutation, and reorganization. During the preparatory stage of the genetic algorithm, initial solutions are randomly generated. They can be regarded as genes that are encoded into chromosomes. Chromosomes then come together to form a population. Afterward, based on the objective function, a fitness function is set as the measure for the quality of staining. The selection step means selecting ideal chromosomes based on the fitness function. Next, in the crossover step, two chromosomes are randomly selected from the set and subjected to crossovers at one or more locations, resulting in the creation of new chromosomes. The mutation step means the formation of a new chromosome after random crossover occurring at two different locations on a chromosome.

However, it should be noted that fitness is still the measure for the new chromosomes resulting from crossover and mutation. The final reorganization step means putting the new chromosomes back into the population. After the predetermined number of iterations has been completed, decoding is performed to generate the final distribution scheme.

3.2. Floyd's algorithm

In 1962, Robert Floyd first proposed Floyd's algorithm, which is an exact algorithm implemented through the iteration of the routing table. It is an efficient and simple method for identifying the shortest route between any two nodes and in weighted graphs.

Create a distance routing table $Dist_1 = Dist_{ij}^{(1)}$ from V_i to V_j directly with one step, and $Dist_1 = Dist_{ij}^{(1)}$ is also the shortest-distance routing table for direct connection between nodes. If there is no direct incident edge between V_i and V_j, let $C_{ij} = +\infty$.

Calculate the two-step shortest distance routing table. Assuming that it takes two steps from V_i to V_j via an intermediate node V_b, the shortest distance from the node V_i to V_j is

$$Dist_{ij}^{(2)} = \min\left\{c_{ib} + c_{bj}\right\} \tag{15}$$

The shortest distance routing table is expressed as $Dist_2 = Dist_{ij}^{(2)}$.

(1) Calculate the m-step shortest distance routing table. Assuming that V_i is connected to V_j via the intermediate node V_r, the shortest distance from V_i to V_r with $k-1$ steps is $Dist_{ir}^{(m-1)}$ and the shortest distance from V_r to V_j with $k-1$ steps is $Dist_{rj}^{(m-1)}$, the shortest distance from V_r to V_j with m steps is

$$Dist_{ij}^{(m)} = \min\left\{Dist_{ib}^{(m-1)} + Dist_{bj}^{(m-1)}\right\} \tag{16}$$

The shortest distance routing table is expressed as $Dist_m = Dist_{ij}^{(m)}$.

(2) Compare the routing tables $Dist_m$ and $Dist_{m-1}$. When $Dist_m = Dist_{m-1}$, the routing table with the shortest distance between any two nodes is $Dist_m$. If there are Q nodes in the graph G and $C_{ij} \geq 0$, the number of iterations $iter$ is expressed by,

$$iter - 1 < \frac{\lg(M-1)}{\lg 2} \leq iter \tag{17}$$

The operation of Floyd's algorithm is done after the steps above to generate an iterated shortest-distance routing table. This table can help the search for the shortest distance and quickest speed of distribution with the genetic algorithm.

3.3. Floyd-based improved genetic algorithm

The VRPTW model designed in this paper is a combinatorial optimization model that is strongly NP-hard. To seek a better solution within the valid time, this paper proposes a "Floyd-based improved genetic algorithm." Using the Floyd algorithm to obtain the minimum travel cost between any two customer nodes, a local search procedure is incorporated to destroy the current operator and repair it later. Compared with the classic genetic algorithm, the proposed algorithm can quickly iterate the infeasible routing to obtain new routing by incorporating Floyd's algorithm, avoiding the effect of the infeasible routing on the general distance matrix. The flow chart of the proposed algorithm is shown in Fig. 2.

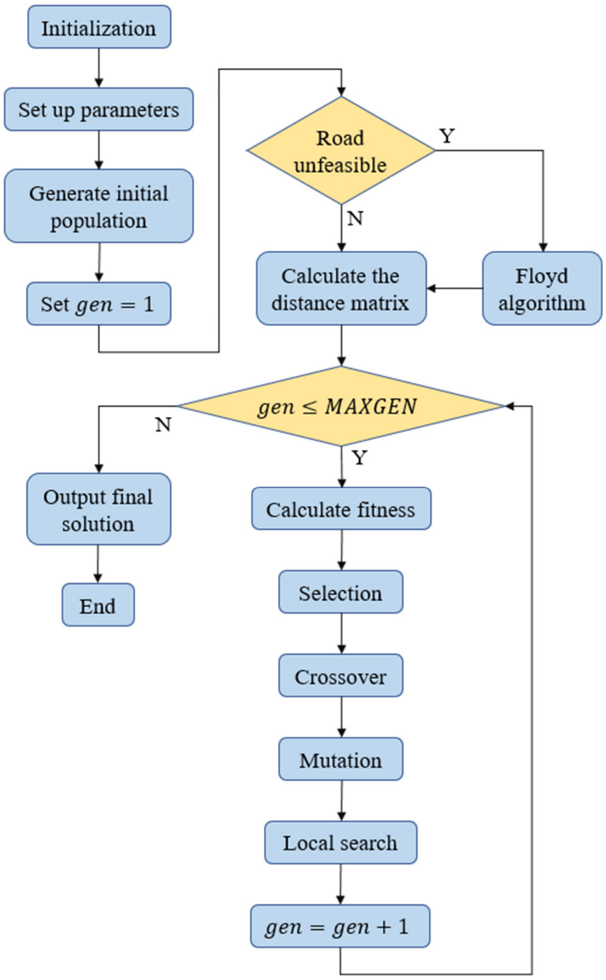

Fig. 2 Proposed algorithm's flow chart

3.3.1. Fitness function

A key step in using the genetic algorithm is the encoding of chromosomes, and simple encoding can quicken the solving process. To address the VRP in this paper, integer encoding is used to encode both distribution centers and customer nodes in the chromosomes.

If this encoding method cannot guarantee that each decoded distribution route satisfies the loading capacity and time window constraints, a penalty function can be used to prevent the distribution routes from violating the constraints whenever possible. Considering that this model is a multi-objective planning, a weighted summation is used to transform the multi-objective into a single objective. Therefore, the total distribution cost, customer satisfaction, and penalty function are expressed as follows:

$$f(x) = \mu \times c(r) + (1-\mu) \times s(r) + \alpha \times q(r) + \beta \times \omega(r) \tag{18}$$

$$c(r) = \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \tag{19}$$

$$s(r) = \sum_{i \in N_c} S_i \tag{20}$$

$$q(r) = \sum_{k=1}^{n} \max\left\{ \left(\sum_{i \in N} d_i \sum_{j \in \Delta^+_{(i)}} x_{ijk} - C \right), 0 \right\} \tag{21}$$

$$\omega(r) = \sum_{i=1}^{n} \max\left\{ (l_i - b_i), 0 \right\} \tag{22}$$

r is the routing scheme, $f(r)$ is the total cost of the current scheme; $c(r)$ is the total distance; $s(r)$ is customer satisfaction; $\mu \in (0,1)$ is the weight of the total distance; $q(r)$ is the number of routes infringing the loading capacity constraint; $\omega(r)$ is the number of customer nodes violating the time window constraint; α is the penalty factor for violation of loading constraint; β is the penalty factor for violation of time window constraint.

For this model, the lower the total cost of vehicle delivery, the better. However, in the selection step of genetic algorithms, individuals with greater fitness are usually selected. Therefore, the fitness function should be set to the reciprocal of the cost function, namely, $fitness = 1/f(r)$.

3.3.2. Population initialization

Before the population initialization step, an initial solution to the VRP is constructed. This initial solution does not necessarily accept the time window and loading constraints. However, a high-quality initial solution can, to some extent, lower the difficulty of searching for the genetic algorithm. The Population initialization is constructed as shown in Algorithm 1:

Algorithm 1: population initialization pseudocode

```
Begin
  Input: Number of customers: n, Number_of_vehicles: k, Route(k) = [ ]
  Output: Population_initialization Δ
  Selected_customer = randomly select a customer j, j ∈ {1, 2, ⋯ , n} from all customers
  Generate_sequence: Seq = [j, j + 1, ⋯ , n, 1, ⋯ , j − 1]
  For i = 1 to n do
    Insert Seq(i) to Route(k)
    If Route satisfies loading constraint then
      If Route is empty then
        Insert Seq(i) to Route(k)
      Else if Route(k) has only one customer then
        Insert Seq(i)and sort by lower bound of time window
      Else if number of customers lr visited on the Route(k) > 1 then
        Traverse the first and last insertion positions
        If lower bound for Seq(i)'s time window ≤ lower bound for the first customer in Route(k) then
          Insert Seq(i) into the first stop
        Else if lower bound for Seq(i)'s time window ≥ lower bound for the last customer in Route(k) then
          Insert Seq(i)into the last stop
        Else
          Traverse intermediate positions between consecutive customers
          If lower bound for the previous customer ≤ lower bound for Seq(i)'s time window ≤ lower bound for the
          next customer then
            Insert Seq(i) in the middle of the two
    Else
      Update k = k + 1
  End For
  Return Population_initialization Δ
End
```

3.3.3. Selection

For the VRP, a binary tournament selection, also known as the elitism strategy, is adopted. Binary tournament selection compares two individuals and selects the one with greater fitness into the offspring population. If the number of population is $NIND$, $NIND$ cycles will be needed. In each cycle, two individuals are randomly selected for comparison, and the one with greater fitness is selected. When the newly selected $NIND$ individuals are duplicates, only one of them is retained.

3.3.4. Crossover

The crossover of the genetic algorithm is represented as follows. Select two parent individuals $Parent_1$ and $Parent_2$.

(1) Randomly select the intersection position of the two parent individuals $Parent_1$ and $Parent_2$;

(2) Move forward the crossover segment of $Parent_2$ before that of $Parent_1$ and the crossover segment of $Parent_1$ before that of $Parent_2$;

(3) Delete the second repeated locus from front to back, and mark the repeated loci in the two parent individuals;

(4) Then delete the second repeated locus to leave two offspring individuals;

3.3.5. Mutation

Mutation means the exchange of two genes on the chromosome of a parent individual at two randomly selected mutation positions $Parent_1$ and $Parent_2$ to form new individuals. This article mainly adopts a single-point mutation approach, which can enhance the search for feasible regions.

3.3.6. Local search

In the large neighborhood search algorithm [19], local search embraces the ideas of “destruction” and “repair” in the large-scale neighborhood search algorithm. This means using the destruction operator to delete several customers from the previous solution and using the repair operator to reinsert the deleted customers into the destroyed solution.

The destruction operator does not remove several customers randomly; instead, it removes several similar customers based on a similarity calculation formula. Similarly, the repair operator does not insert the removed customer into any insertion position of any route randomly; instead, it reinserts the removed customer into a position where the total travel distance would increase least, provided that the loading and time window constraints are satisfied.

(1) Destruction operator

The destruction operator is used to remove several relevant customers as expressed in:

$$R(i,j)=\frac{1}{c_{ij}+\phi_{ij}} \tag{23}$$

$$c'_{ij}=\frac{c_{ij}}{\max c_{ij}} \tag{24}$$

If i and j are not served by the same vehicle, $\emptyset_{ij}=1$; if yes, $\emptyset_{ij}=0$, c'_{ij} is the normalized value of c_{ij} and falls in the range of [0,1]. It can be found that a larger $R(i,j)$ means a greater correlation between customer i and customer j. On this basis, assume that there are n customers and q customers to be removed, and the random element is D. Then, the pseudocode of the destruction operator is as in Algorithm 2.

Algorithm 2: destruction operator

```
Begin
  Input: the solution: r, the number of customers to be deleted: q, the random element: D
  Output: the solution after destruction: r_d, set of removed customers: I
  Select customer i_seed from the solution randomly and put i_seed into the set I
  While |r| < q do
    Select the customer i_curr from the set I randomly
    Sort the customers who are in the current solution r but not in the set I as follows:
I < j ⇒ R(i_curr, Rank[i] < R(i_curr, Rank[j], and store the sorted result in the sorted sequence Rank.
    Calculate the serial numbers of randomly selected customers m ← ⌈rand^D |Rank|⌉; I ← I ∪ {Rank_m}
  End while
  Remove the customers in the set I from the solution r_d to get a destroyed solution
  Return the solution r_d and set I
End
```

where $rand$ is random numbers from 0 to 1, $|Rank|$ is the number of customers in the set $Rank$, and $\lceil \quad \rceil$ means rounded up to an integer.

(2) Repair operator

Given the set I of customers removed and the destroyed solution r_d, the backfilling and insertion of the operator are performed. The pseudocode of the repair operator is as in Algorithm 3:

Algorithm 3: repair operator

```
Begin
  Input: the solution after destruction: r_d, set of removed customers: I
  Output: solution: r
r ← r_d
  while |I| > 0 do
    Calculate the minimum insertion cost of each customer z_i = min_{m⊂M} Δf_{i,m} in I as well as the serial number p_i of the
    corresponding insertion route of z_i and the insertion position pos_i on this route
    If the customer can't be inserted into any route in the current solution,
      Create a new route should be created
    Select the customer i_max of max_{i∈I} z_i from I
    Insert the customer i_max back to Position pos_{i_max} on Route p_{i_max} in r
    I ← I{i_max}, Remove the customer i_max from I
  End while
  Return solution r
End
```

where $\Delta f_{i,m}$ refers to the distance increment after the customer i is inserted into the route m where the total travel distance would increase least, provided that the constraints are satisfied.

3.3.7. Reorganization

In the binary tournament selection, only one of the duplicate individuals is retained, and others are deleted. Therefore, the selected offspring individuals must be less than the original population. The new offspring individuals surviving crossover, mutation, and local search are also less than the original population. This necessitates the reorganization step to combine the offspring individuals surviving local search with the original population initiated at the beginning of the current iteration, to keep the population number unchanged.

Assuming that the amount of population is $NIND$, and the number of individuals selected by a binary tournament is $NIND_1$, the reorganization step follows a principle: $NIND_1$ offspring individuals surviving local search in the current iteration are retained, and only the top $NIND - NIND_1$ individuals which have the highest fitness in the original population initiated at the beginning of the current iteration are retained.

4. Computational Experiment and Analysis

In this section, a comparative analysis is conducted between improved genetic algorithms and classic genetic algorithms for addressing VRP problems. It includes a computational experiment using a Solomon benchmark and a designed simulation experiment to validate the effectiveness of the improved genetic algorithms.

4.1. Comparison with classic genetic algorithm

In this section, a case study is conducted using the Solomon benchmark test instances for the VRP (available for download from the website https://www.sintef.no/projectweb/top/vrptw). This dataset consists of 56 instances. Each instance contains a distribution center and 100 customer nodes. In the experiments, this study focuses primarily on the instances from Group C1. The maximum runtime of the algorithm was set to 480 seconds in the experiment. And the algorithm will end when it finds the best-known solution (BKS). Ten independent repetitions of the experiments are conducted for each instance, utilizing both the classic genetic algorithm and the improved genetic algorithm for solutions. The performance results and the BKS of the instances are presented in Table 4.

Table 4 The performance results of C1

Instance	BKS	Classic genetic algorithms				Improved genetic algorithm			
		Best	Mean	Worst	Average time/s	Best	Mean	Worst	Average time/s
C101	828.94	1282.48	1509.54	1697.49	480	**828.94**	**828.94**	**828.94**	**128.09**
C102	828.94	1455.71	1576.48	1672.07	480	**828.94**	**828.94**	**828.94**	**229.66**
C103	828.06	1410.15	1557.47	1694.61	480	**828.06**	**846.24**	**893.21**	**428.7**
C104	824.78	1362.22	1440.75	1586.2	480	**824.78**	**882.61**	**956.34**	**480**
C105	828.94	1188.37	1366.55	1592.62	480	**828.94**	**828.94**	**828.94**	**192.63**
C106	828.94	1198.24	1330.71	1457.92	480	**828.94**	**828.94**	**828.94**	**186.41**
C107	828.94	1217.73	1357.33	1465.54	480	**828.94**	**832.28**	**862.37**	**265.96**
C108	828.94	1150.43	1276.79	1396.44	480	**828.94**	**828.94**	**828.94**	**284.87**
C109	828.94	1088.66	1631.41	1841.74	480	**820.4**	**831.84**	**905.09**	**302.03**

The overall results indicate that in all test instances, the improved genetic algorithm outperforms classic genetic algorithms. Classic genetic algorithms exhibit relatively poor convergence capability within a limited time. This superiority can be attributed to the incorporation of local search operations, which maintains a balance between intensification and diversification, effectively leveraging the current search space. Therefore, the integration of local search operations into genetic algorithms can significantly enhance search efficiency when confronted with large-scale problems. It is worth mentioning that the improved genetic algorithms find a solution superior to BKS in the instance C109.

4.2. Experiment design

To verify the effectiveness of the proposed method, a simulation experiment is designed in this paper. The experiment includes one distribution center and 20 distribution customer nodes. The coordinates are numbered 0, 2, ..., 20, where 0 represents the distribution depot. Suppose the maximum loading capacity and speed of a delivery vehicle are 150 kg and 20 km/h, respectively. The lower bound of the satisfaction time window is increased by 5 minutes from its lower bound of the time window, while the upper bound is reduced by 5 minutes from its upper bound of the time window. The coordinate, demand, service time window, and required service time at each customer node are shown in Table 5 below.

Table 5 Customer node information

Serial no.	X coordinate/km	Y coordinate/km	Demand/kg	Lower bound of the time window	Upper bound of the time window	Service time/min
0	40	50	0	7:00	21:00	0
1	45	68	40	17:00	17:30	32

Table 5 Customer node information (continued)

Serial no.	X coordinate/km	Y coordinate/km	Demand/kg	Lower bound of the time window	Upper bound of the time window	Service time/min
2	45	70	10	16:30	17:00	8
3	42	66	40	8:00	9:30	32
4	42	68	10	16:00	16:30	8
5	42	65	20	7:20	8:00	16
6	40	69	10	15:00	15:40	8
7	40	66	40	9:40	10:20	32
8	38	68	30	10:40	11:30	25
9	38	70	10	14:40	15:20	7
10	35	66	5	12:00	12:30	5
11	35	69	17	13:30	14:20	15
12	25	85	3	15:10	16:10	4
13	22	75	16	8:30	9:30	18
14	22	85	23	14:10	15:10	20
15	20	80	31	12:40	13:40	30
16	42	60	15	13:40	15:20	16
17	23	80	20	17:00	17:50	20
18	36	62	5	11:10	12:00	5
19	21	86	7	13:40	15:00	4
20	37	65	10	9:30	11:10	8

Given the stated conditions and without considering infeasible routing, this paper tackles the problem using the improved algorithm. In consideration of the diverse requirements of contemporary enterprises, the values of μ are set to 0.8 and 0.2 for separate problem-solving instances. This differentiation signifies the varying priorities of enterprises, with one emphasizing total distance while the other prioritizes customer satisfaction. Setting the parameters of the improved genetic algorithm based on the specifics of the model constructed in this paper. The algorithm parameters are shown in Table 6. The final solution obtained with different weights μ is shown in Table 7.

Table 6 Parameter setting of improved genetic algorithm

Name of parameter	Value
Penalty function coefficient for loading constraint violation	100
Penalty function coefficient for violation of time window constraint	100
Number of population	50
Maximum number of iterations	200
Mutation probability	0.1
Crossover probability	0.9

Table 7 The final solution obtained with different weights μ

Weight	$\mu = 0.8$	$\mu = 0.2$
Number of vehicles	3	3
Distance	182.82 km	188.53 km
Customer satisfaction	180	200

As shown in Table 7, this paper uses an improved genetic algorithm to generate a set of solutions with different weights μ. The solutions can be utilized to better meet the diverse decision-making needs of enterprises, thereby reducing transportation costs or improving customer satisfaction.

4.3. Presentation of experiment results

To simulate the occurrence of infeasible paths in the model, this paper randomly generates 15 infeasible routes before distribution. The specific node information is shown in Table 8. This experiment is conducted under the premise of heightened emphasis on customer satisfaction by the company, thus setting $\mu = 0.2$. The parameters are the same as those of the improved genetic algorithm shown in Table 6. The routes of vehicles solved by the classic genetic algorithm are shown in Fig. 3. The results of the classic genetic algorithm are shown in Table 9 and Table 10.

Table 8 Nodes with infeasible routing

Serial no.	Two customer nodes with infeasible routing
1	0.20
2	1.16
3	1.19
4	2.9
5	2.10
6	4.6
7	4.17
8	4.20
9	9.19
10	10.11
11	10.14
12	10.18
13	11.12
14	11.14
15	16.17

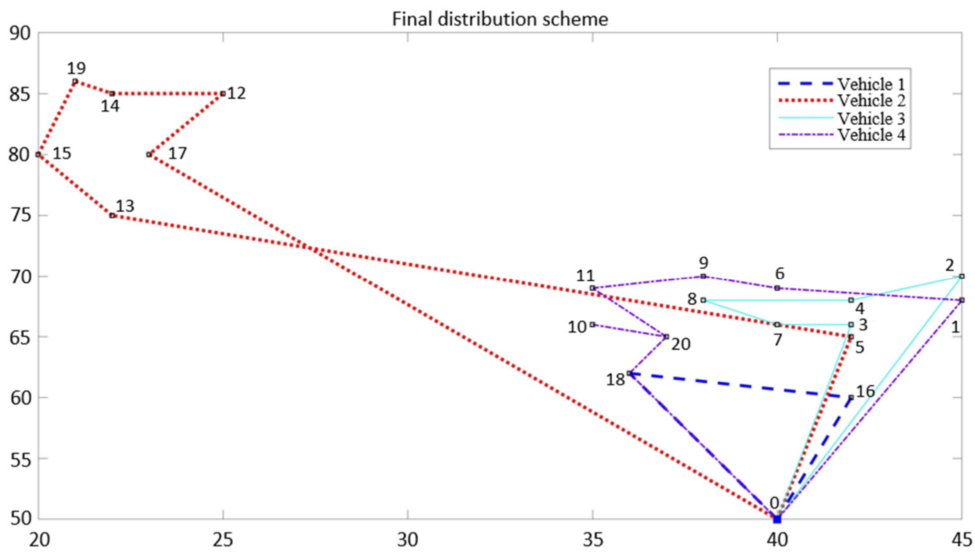

Fig. 3 Final distribution scheme of classic genetic algorithm

Table 9 Classic genetic algorithm

Final solution	-
Number of vehicles	4
Total traveling distance	225.52 km
Customer satisfaction	200
Number of routes infringing the constraints	0
Number of customers infringing the constraints	0

Table 10 Distribution route solved by the classic genetic algorithm

Distribution route	Running sequence
Route 1	0→18→16→0
Route 2	0→5→13→15→19→14→12→17→0
Route 3	0→3→7→8→4→2→0
Route 4	0→18→20→10→20→11→9→6→1→0

In Table 10 customer node 18 and the second passing node 20 are intermediate nodes. When the improved genetic algorithm designed is used to solve the VRPTW considering infeasible routing, the curve tends to be stable when more than 90 iterations are performed, and it finally converges to 195.73 km while the customer satisfaction is 200, where f_1 is the total distance; f_2 is customer satisfaction, as shown in Fig. 4. The routes of vehicles are shown in Fig. 5. The final solution is shown in Table 11 and Table 12.

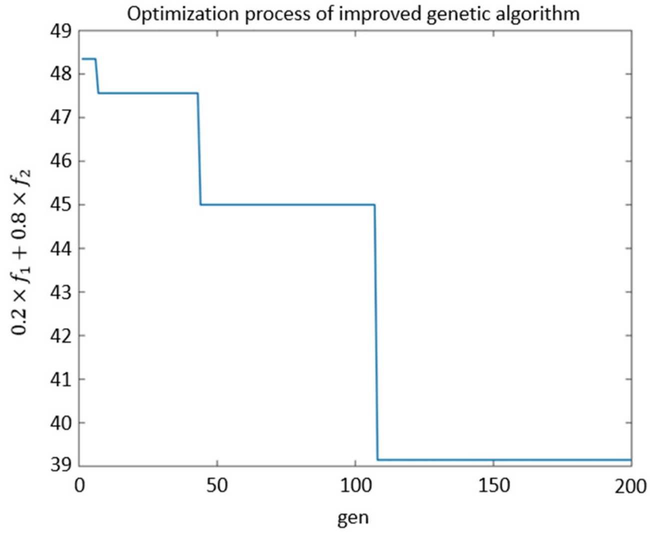

Fig. 4 Optimization process of improved genetic algorithm

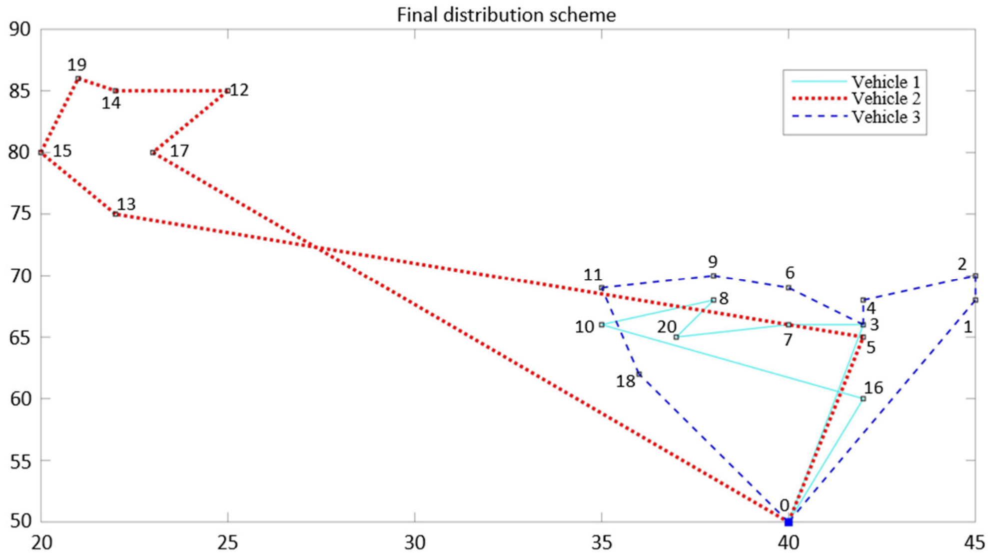

Fig. 5 Final distribution scheme

Table 11 Final solution

Final solution	-
Number of vehicles	3
Total distance	195.73 km
Customer satisfaction	200
Number of routes infringing the constraints	0
Number of customers infringing the constraints	0

Table 12 Distribution route

Distribution route	Running sequence:
Route 1	0→3→7→20→8→10→16→0
Route 2	0→5→13→15→19→14→12→17→0
Route 3	0→18→11→9→6→3→4→2→1→0

In distribution route 3, customer node 3 is the intermediate node. In the same case of maximizing customer satisfaction, compared with the classical genetic algorithm, the improved genetic algorithm improves the total distance by 29.79 km and uses one less vehicle. The quality of the final solution is significantly improved.

To compare the convergence ability of the algorithm under different decision preferences. This study conducted ten independent iterative experiments for both $\mu = 0.8$ and $\mu = 0.2$. The maximum runtime for each experiment was set at 30 seconds, and the algorithm parameters were consistent with those specified in Table 6. The results are presented in Table 13 and Table 14. The solutions generated by the two algorithms are presented in Table 15.

Table 13 Comparison of algorithms under $\mu = 0.8$

Parameter	Classic genetic algorithm			Improved genetic algorithm		
	Best	Mean	Worst	Best	Mean	Worst
Number of vehicles	3	4	4	3	3	3
Distance	207.49 km	240.67 km	255.85 km	190.41 km	191.02 km	195.73 km
Customer satisfaction	190	184	170	200	182	180

Table 14 Comparison of algorithms under $\mu = 0.2$

Parameter	Classic genetic algorithm			Improved genetic algorithm		
	Best	Mean	Worst	Best	Mean	Worst
Number of vehicles	3	4	5	3	3	3
Distance	225.52 km	241.65 km	268.66 km	195.73 km	197.16 km	203.77 km
Customer satisfaction	200	200	200	200	200	200

Table 15 Solutions of two algorithms

Parameter	Classic genetic algorithm		Improved genetic algorithm	
Weight	$\mu = 0.8$	$\mu = 0.2$	$\mu = 0.8$	$\mu = 0.2$
Number of vehicles	3	4	3	3
Distance	207.49 km	225.52 km	190.41 km	195.73 km
Customer satisfaction	190	200	180	200

When the occurrence of infeasible routing manifests within the system, the solutions derived from the improved genetic algorithm dominate those produced by the classical genetic algorithm. The application of improved genetic algorithms proves to be more efficacious in addressing the problems of this study.

4.4. Comparison of the improved algorithm

The robustness and parallelism of the proposed algorithm have been verified under different conditions. The classic genetic algorithm and the improved algorithm are compared under the same parameters. The results show that the latter is superior in all aspects. In this case of $\mu = 0.8$, the average travel distance is reduced by 49.65 km, marking a 20.6% improvement compared to the pre-improvement average. Additionally, the average vehicle count is reduced by 1. Similarly, for $\mu = 0.2$, there is an average reduction of 1 vehicle, and the average travel distance is decreased by 44.49 km, representing an 18.4% improvement. The analysis in Table 15 shows that the classical genetic algorithm's solutions are dominated by the improved genetic algorithm's solutions.

5. Conclusion

This study focuses on optimizing vehicle delivery paths to address the occurrence of infeasible routes. Considering the infeasible routes and hard time window constraints for customers, the MILP model is developed to solve the VRPTW. The model aims to maximize customer satisfaction and minimize total vehicle travel distance as optimization objectives. To solve the model, a Floyd-based improved genetic algorithm is designed, incorporating local search operations to enhance search efficiency. Computational experiments are conducted to validate the effectiveness of the proposed method. The main findings of this study can be summarized as follows:

(1) There is a lack of research on the VRPTW model in uncertain environments, with most studies focused on deterministic environments. To fill this research gap, this study establishes a VRPTW MILP model that considers infeasible routes and hard time window constraints for customers, to maximize customer satisfaction and minimize total vehicle travel distance.

(2) This study considers infeasible routes and hard time window constraints for customers in the VRPTW model, making it more applicable to the real world. To address this complex problem, a Floyd-based improved genetic algorithm is proposed to solve the model. The algorithm incorporates the local search operation of adaptive large neighborhood search to improve solution quality and convergence speed.

(3) To validate the effectiveness of the proposed method, a computational experiment is designed. The experiment involves customer nodes with coordinates, demands, service time windows, and required service time, as well as simulated infeasible routing. The results show that the Floyd-based improved genetic algorithm reduces the average travel distance by 20.6% and 18.4% under $\mu = 0.8$ and $\mu = 0.2$. Moreover, the average number of vehicles used is reduced by one in each case. The solutions of the classical genetic algorithm are dominated by the solutions of the improved genetic algorithm. It is worth mentioning that the improved genetic algorithm finds a solution superior to BKS in the instance C109.

In conclusion, the proposed method in this study demonstrates outstanding performance in solving the model constructed in this paper as well as large-scale problems, successfully achieving the goal of reducing total vehicle travel distance while significantly improving customer satisfaction. These results provide strong support for practical applications in the field of logistics delivery and offer valuable insights for further research and improvement.

Acknowledgements:

The project was supported by the Natural Science Foundation of Fujian Province of China (No. 2022J02053) and the Xiamen University of Technology Innovation and Technology Project (No. YKJCX2022029).

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," Management Science, vol. 6, no. 1, pp. 80-91, October 1959.

[2] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," Operations Research, vol. 35, no. 2, pp. 254-265, March-April 1987.

[3] P. Cappanera, C. Requejo, and M. G. Scutella, "Temporal Constraints and Device Management for the Skill VRP: Mathematical Model and Lower Bounding Techniques," Computers & Operations Research, vol. 124, article no. 105054, December 2020.

[4] Y. Wang, L. Wang, G. C. Chen, Z. Q. Cai, Y. Q. Zhou, and L. N. Xing, "An Improved Ant Colony Optimization Algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice," Swarm and Evolutionary Computation, vol. 55, article no. 100675, June 2020.

[5] L. Zhen, C. L. Ma, K. Wang, L. Y. Xiao, and W. Zhang, "Multi-Depot Multi-Trip Vehicle Routing Problem with Time Windows and Release Dates," Transportation Research Part E-Logistics and Transportation Review, vol. 135, article no. 101866, March 2020.

[6] D. A. Neira, M. M. Aguayo, R. De la Fuente, and M. A. Klapp, "New Compact Integer Programming Formulations for the Multi-Trip Vehicle Routing Problem with Time Windows," Computers & Industrial Engineering, vol. 144, article no. 106399, June 2020.

[7] A. Bortfeldt and J. M. Yi, "The Split Delivery Vehicle Routing Problem with Three-Dimensional Loading Constraints," European Journal of Operational Research, vol. 282, no. 2, pp. 545-558, April 2020.

[8] Y. Yao, X. N. Zhu, H. Y. Dong, S. N. Wu, H. L. Wu, L. C. Tong, et al., "ADMM-Based Problem Decomposition Scheme for Vehicle Routing Problem with Time Windows," Transportation Research Part B-Methodological, vol. 129, pp. 156-174, November 2019.

[9] V. Q. Hien, T. C. Dao, and H. T. T. Binh, “A Greedy Search Based Evolutionary Algorithm for Electric Vehicle Routing Problem,” Applied Intelligence, vol. 53, no. 3, pp. 2908-2922, February 2023.

[10] F. E. Zulvia, R. J. Kuo, and D. Y. Nugroho, “A Many-Objective Gradient Evolution Algorithm for Solving a Green Vehicle Routing Problem with Time Windows and Time Dependency for Perishable Products,” Journal of Cleaner Production, vol. 242, article no. 118428, January 2020.

[11] O. Dominguez, A. A. Juan, B. Barrios, J. Faulin, and A. Agustin, “Using Biased Randomization for Solving the Two-Dimensional Loading Vehicle Routing Problem with Heterogeneous Fleet,” Annals of Operations Research, vol. 236, no. 2, pp. 383-404, January 2016.

[12] X. H. Cai, L. Jiang, S. H. Guo, H. J. Huang, and H. W. Du, “TLHSA and SACA: Two Heuristic Algorithms for Two Variant VRP Models,” Journal of Combinatorial Optimization, vol. 44, no. 4, pp. 2996-3022, November 2022.

[13] M. Salavati-Khoshghalb, M. Gendreau, O. Jabali, and W. Rei, “A Hybrid Recourse Policy for the Vehicle Routing Problem with Stochastic Demands,” EURO Journal on Transportation and Logistics, vol. 8, no. 3, pp. 269-298, 2019.

[14] P. Xu, Q. X. Liu, and Y. H. Wu, “Energy Saving-Oriented Multi-Depot Vehicle Routing Problem with Time Windows in Disaster Relief,” Energies, vol. 16, no. 4, article no. 1992, February 2023.

[15] G. Kim, “Dynamic Vehicle Routing Problem with Fuzzy Customer Response,” Sustainability, vol. 15, no. 5, article no. 4376, March 2023.

[16] S. F. Ghannadpour, S. Noori, R. Tavakkoli-Moghaddam, and K. Ghoseiri, “A Multi-Objective Dynamic Vehicle Routing Problem with Fuzzy Time Windows: Model, Solution and Application,” Applied Soft Computing, vol. 14, pp. 504-527, January 2014.

[17] H. Yao, “Application of Artificial Intelligence Algorithm in Mathematical Modelling and Solving,” Applied Mathematics and Nonlinear Sciences, vol. 7, no. 1, pp. 449-456, January 2022.

[18] Y. Xue, Q. Sun, C. Li, W. Dang, and F. Hao, “Distribution Network Monitoring and Management System Based on Intelligent Recognition and Judgement,” Applied Mathematics and Nonlinear Sciences, vol. 7, no. 1, pp. 685-694, January 2022.

[19] P. Shaw, “A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems,” APES Group, Dept of Computer Science, University of Strathclyde, Working paper, July 14, 1997.

Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (https://creativecommons.org/licenses/by-nc/4.0/).