U.P.B. Sci. Bull., Series C, Vol. 84, Iss. 3, 2022 ISSN 2286-3540

A HYBRID FIREWORKS ALGORITHM FOR THE MULTI-TRIP VEHICLE ROUTING PROBLEM

Qiang SONG[1]

This paper investigates the muti-trip vehicle routing problem (MTVRP) with considerations of vehicle capacity and time constraints. The problem aims to determine a set of trips and assign each trip to a vehicle in a proper way. In this work, firstly, a mixed integer linear programming (MILP) model is formulated to optimize the total travelling time. Then, a hybrid fireworks algorithm (HFWA) is developed for solution generation since it has been proven to be NP-hard. In the algorithm design, a new coding scheme is proposed to accommodate the problem characteristic. Meanwhile, the opposition-based learning technique and the evolution mechanism of artificial bee colony (ABC) algorithm are embedded into FWA for balancing its exploration and exploitation abilities. Computational results indicate that HFWA is effective and efficient in solving MTVRP when compared to other algorithms.

Keywords: vehicle routing problem, multi-trip, fireworks algorithm, reverse learning

1. Introduction

Vehicle routing problem (VRP) is a research hotspot in the field of operations research and combinational optimization. A number of VRP variants have appeared in different practical scenarios over the years, such inbound logistics and express logistics [1,2]. Multi-trip vehicle routing problem (MTVRP) is a significant variant of VRP for logistics and transportation companies, in which a vehicle may execute multiple delivery tasks in the planning horizon [3]. In this regard, appropriate planning of MTVRP plays an important role in reducing transportation cost and increasing customer satisfaction [4,5]. This paper is motivated by this background and aims to investigate such a practical scheduling problem in effective manner.

This paper investigates the MTVRP with considerations of capacity and time constraints, in which the total travelling time is selected for improvement. As MTVRP has been proven to be NP-hard, the design of solution methodologies has attracted much attention from many researchers. The algorithms in this field can be classified into three categories: exact methods, heuristic and metaheuristic. The methods are capable of obtaining optimal solutions, and related algorithms include mixed integer programming modeling (MILP) and dynamic programming

[1] Associate Prof., Ph.D., School of Computer Science and Software, Zhaoqing University, China, e-mail: aysq168@163.com

(DP). However, the computation time will explosively grow with the increase of problem size [6]. Such a dilemma makes it impossible to apply these algorithms to practical cases. The heuristic methods can quickly obtain the approximate optimal solution of MTVRP by virtue of different scheduling rules [7]. Meanwhile, it is very difficult to select appropriate scheduling rules for different practical scenarios and some minor change of a specified scenario may lead to invalidation of the original selection of scheduling rules [8]. It should be noticed that it is almost impossible to solve practical-scale instances to optimality considering the exploding search space. Under such circumstances, metaheuristic algorithms have become the most appropriate approaches since they are problem-agnostic and able to find optimal approximation solutions in a reasonable time [9]. For example, Hsu [10] proposed a hybrid shuffled frog-leaping algorithm to solve the problem of disassembly process planning; Salhi [11] presented a hybrid genetic algorithm for traditional VRP; in order to solving the MTVRP, François [12] combined the heuristic algorithm with bin packing routines in order; Saxena [13] adopted a parallelized version of genetic algorithm to solve VRP based on OpenMP programming model. In this regard, this paper proposes a hybrid fireworks algorithm (HFWA) to the investigated MTVRP.

FWA is a relatively new meta-heuristic algorithm, which is inspired by emulating the fireworks explosion in the night [14]. Due to its simple and novel concept, FWA has attracted much attention of many researchers. So far, FWA has been successfully applied in solving different practical engineering problems, including numerical optimization, image fusion and job shop scheduling [15, 16]. Yang and Ke studied capacitated vehicle routing problem (CVRP) and modified FWA for solution generation [17]. They proposed a new method to generate sparks according to selection rule and designed a new method to determine the explosion amplitude for each firework. Wang et al. developed a new task scheduling method for fog computing, in which a hybrid algorithm is proposed by introduce the explosion radius detection mechanism of FWA into genetic algorithm [18]. Simulation results indicated that the hybrid method can achieve better execution time and ensure better load in a short time. Pang et al. designed an improved fireworks algorithm to minimize the makespan in permutation flow shop scheduling problems (PFSPs) and hybrid flow shop scheduling problems (HFSPs) [19]. Different improved strategies, non-linear decreasing radius and Cauchy mutation operators are utilized to enhance algorithm performance. Comprehensive experiments in these have validated FWA’s excellent performance when compared to traditional metaheuristics, like genetic algorithm and particle swarm optimization. Meanwhile, the FWA applications in these works have directive significance to model formulation and algorithm design in current research.

To our knowledge, little was published on addressing MTVRP by virtue of FWA-based algorithm. In this research, an effort was made to deal with such a practical scheduling problem by this efficient algorithm. To better adapt FWA to the investigated MTVRP, three modifications have been embedded into FWA. First, a novel solution representation architecture is developed to accommodate the problem characteristic. Second, an opposition-based learning initialization method is utilized to generate initial solutions with high qualities. Third, evolution strategy of ABC algorithm is embedded into FWA to strengthen the information exchange among different individuals, which aims to balance its exploration and exploitation abilities.

The rest of this paper is organized as follows. Section 2 formulates MTVRP with a MILP model. Section 3 presents the outlines of FWA, and Section 4 gives detailed designs of the proposed HFWA. Computational studies are designed and discussed in Section 5. Finally, Section 6 presents conclusions and future work.

2. Mathematical modeling

2.1 Problem description

Fig. 1 depicts the investigated MTVRP. Let $G=(V,E)$ be a complete and undirected graph, where $V=\{0,\ldots,n\}$ is the set of vertices and $E=\{(i,j)\,|\,i,j\in V, i\neq j\}$ the set of arcs. Vertice 0 represents the distribution center and $J=\{1,\ldots,n\}$ correspond to clients. The travelling time between two vertices is defined as t_{ij}. For every client i, the corresponding order size is q_i. A fleet of homogeneous vehicles are used to execute multiple trips in the planning horizon T_H.

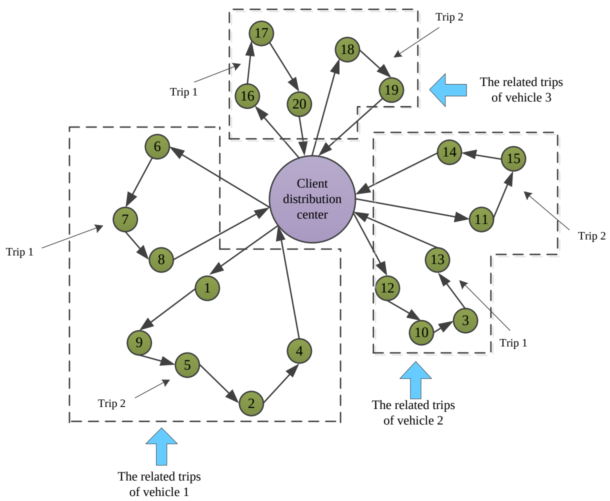

Fig. 1. Schematic diagram of MTVRP problem

In MTVRP, two important decisions should be made: (1) group clients into trips and (2) schedule these trips on vehicles. A solution is feasible if the following constrains are all satisfied:

(1) Every client is served only once.
(2) Every trip starts and ends at the distribution center.
(3) Trips assigned to a vehicle do not overlap with each other.
(4) The completion time of each vehicle does not exceed T_H .
(5) The demand sum in any trip does not exceed the vehicle capacity Q .

2.2 Mathematical model

The MTVRP is formulated in this subsection by virtue of a MILP model. First, some decision variables are introduced as follows:

0-1 decision variable, if arc (i, j) is assigned to the r-th trip of vehicle k , $x^r_{ijk}=1$; otherwise, $x^r_{ijk}=0$;

0-1 decision variable, if the vehicle k is used, $y_k=1$; otherwise, $y_k=0$;

:

The time point when vehicle k arrives at vertice j by arc (i, j) in its

: r-th trip;

Mathematically, the MTVRP can be formulated in the following manner.

$$\min F = \sum_{k\in K}\sum_{r\in R}\sum_{i\in V}\sum_{j\in V} x^r_{ijk} \cdot t_{ij} \tag{1}$$

s.t.

$$\sum_{k\in K}\sum_{r\in R}\sum_{i\in V} x^r_{ijk} = 1 \qquad \forall j \in J \tag{2}$$

$$x^r_{iik} = 0 \qquad \forall i \in V,\ \forall k \in K,\ \forall r \in R \tag{3}$$

$$\sum_{i\in V} x^r_{ijk} - \sum_{i\in V} x^r_{jik} = 0 \qquad \forall k \in K,\ \forall r \in R \tag{4}$$

$$x^r_{ijk} \le y_k \quad \forall i, j \in V \qquad \forall k \in K,\ \forall r \in R \tag{5}$$

$$\sum_{j\in J} x^r_{0jk} \le y_k \qquad \forall k \in K,\ \forall r \in R \tag{6}$$

$$\sum_{r\in R} x^r_{0jk} \le y_k \qquad \forall k \in K,\ \forall j \in J \tag{7}$$

$$\sum_{r\in R}\sum_{j\in J} x^r_{0jk} \ge y_k \qquad \forall k \in K \tag{8}$$

$$\sum_{j\in J} T^0_{0jk} \cdot x^0_{oji} = 0 \qquad \forall k \in K \tag{9}$$

$$\sum_{j\in J} T^r_{0jk} \cdot x^r_{oji} = \sum_{j\in J} t_{0j} \cdot x^r_{0ji} + \sum_{j\in J} T^{r-1}_{0jk} \cdot x^r_{0ji} \quad \forall k \in K,\ \forall r \in R/\{1\} \tag{10}$$

$$\sum_{j\in J} T^r_{j0k} \cdot x^r_{j0i} = \sum_{j\in J} t_{0j} \cdot x^r_{0ji} + \sum_{j\in J} T^r_{j0k} \cdot x^r_{j0i} \quad \forall k \in K,\ \forall r \in R \tag{11}$$

$$\sum_{k\in K}\sum_{r\in R}\sum_{i\in V} x^r_{ijk} \cdot \left(T^r_{ijk} - t_{ij}\right) = \sum_{k\in K}\sum_{r\in R}\sum_{i\in V} x^r_{jik} T^r_{jik} \quad \forall j \in J \tag{12}$$

$$\sum_{j \in J} T^r_{j0k} \cdot x^r_{j0i} \leq T_H \qquad \forall k \in K, \ \forall r \in R \tag{13}$$

$$\sum_{i \in V} \sum_{j \in V} x^r_{ijk} \cdot q_i \leq Q \qquad \forall k \in K, \ \forall r \in R \tag{14}$$

$$x^r_{ijk} \in \{0,1\} \qquad \forall k \in K, \ \forall i, j \in V, \forall r \in R \tag{15}$$

$$y_k \{0,1\}, \qquad \forall k \in K \tag{16}$$

$$T^r_{ijk} \geq 0 \qquad \forall k \in K, \ \forall i, j \in V, \forall r \in R \tag{17}$$

The objective function (1) is to minimize the total travelling time. Constraint (2) ensures that each client is visited exactly once by a vehicle in a trip. Constraint (3) states that there are no routes or trips between same vertices, and the flow conservation at each client is defined in Constraint (4). Constraint (5) guarantees that only selected vehicles can execute trips. Constraints (6) and (7) indicates that each trip start from the distribution center and trips assigned to a vehicle do not overlap with each other. Constraint (8) ensures that there exists at least on trip for each selected vehicle. For each vehicle, Equations (9) and (10) are used to calculate the arrival time at its first client vertice, while equation (11) defines the arrival times point at its last client vertice. The time points when a vehicle arrives at other client vertices are calculated in Equation (12). Constraints (13) guarantees that all trips should be completed within the planning horizon. Constraint (14) takes into consideration the vehicle capacity, i.e., all demand assigned to a trip may not exceed the limitation. Finally, Equations (15)~(17) defines boundary values of all decision variables.

3. FWA algorithm

FWA is a new meta-heuristic algorithm, which is inspired by emulating the fireworks explosion in the night [20]. FWA is a population-based evolutionary meta-heuristic in nature, and the individual in algorithm is referred as firework. Two types of mutation operations are utilized to generate offspring individuals, which are called explosion sparks and Gaussian sparks. Numerical results indicate that FWA works very well on some practical engineering problems. This section presents a detailed introduction of this novel algorithm.

The outlines of basic FWA are stated as follows:

Algorithm 1. *Outlines of FWA*

1: *Initialize algorithm parameters.*
2: *Randomly generate* n *individuals.*
3: *Evaluate the solution performance of* n *individuals.*
4: ***Repeat***
5:     *calculate the number of explosion sparks for each individual;*
6:     *calculate the explosion amplitude for each individual;*

7: *generate sparks for each individual;*
8: *generating Gaussian sparks for each individual;*
9: *select the best individual into offspring generation;*
10: *randomly select the other* $n-1$ *offspring individuals based on a probability;*
11: ***until** any algorithm stopping criteria is met.*
12: *Output the best individual.*

In FWA's evolution, the explosion sparks number s_i generated for parent firework x_i (i.e., the i-th individual in population) is determined by its solution performance. Given the evaluation function $f(\cdot)$, s_i is obtained according to the following expression:

$$s_i = M \cdot \frac{f_{\max} - f(x_i) + \varepsilon}{\sum_{j=1}^{n} f_{\max} - f(x_j) + \varepsilon} \tag{18}$$

$$s_i = \begin{cases} s_{\min} & if\ s_i < s_{\min} \\ s_{\max} & if\ s_i > s_{\max} \\ round(s_i) & else \end{cases} \tag{19}$$

where $s_{\min}$ and $s_{\max}$ are boundary of s_i. In addition, M represents the total number of sparks and $f_{\max}$ defines the evaluation of the worst individual in current population. Parameter ε is a small constant to avoid zero division. In addition, the explosion amplitude A_i for x_i is calculated by:

$$A_i = A \cdot \frac{f(x_i) - f_{\min} + \varepsilon}{\sum_{j=1}^{n} f(x_j) - f_{\min} + \varepsilon} \tag{20}$$

where A defines the maximum explosion amplitude, and $f_{\min}$ is the evaluation of the best individual in current population.

Explosion sparks in FWA are generated by taking the flowing steps:

Algorithm 2. Explosion sparks generation in FWA

1: *Initialize all solution arrays of the explosion sparks, i.e., set* $spark_i \leftarrow x_i$.
2: *Calculate the offset displacement, i.e., set* $\Delta x = A_i \cdot rand(-1, 1)$.
3: *Set* $z_d = round(rand(0, 1))$, *where* $d = 1, 2, \cdots, D$.
4: *For* $d = 1, 2, \cdots, D$, *where* $zd == 1$ *do*
5: $spark_{i,d} \leftarrow spark_{i,d} + \Delta x$
6: *if* $spark_{i,d}$ *out of bounds then*
7: *Set* $spark_{i,d} \leftarrow x_{min,d} + |spark_{i,d}| \,\%(x_{max,d} - x_{min,d})$
8: *end if*
9: *end for*

In FWA, the Gaussian sparks are generated according to Algorithm 3.

Algorithm 3. Gaussian sparks generation in FWA

1: *Initialize all solution arrays of Gaussian sparks, i.e., set* $\tilde{x}_i \leftarrow x_i$.
2: *Calculate the offset displacement, i.e., set* $e \leftarrow Gaussian(1,1)$.
3: *Set* $z_d = round(rand(0, 1))$*, where* $d = 1, 2, \cdots, D$.
4: *For* $d = 1, 2, \cdots, D$*, where* $zd == 1$ *do*
5: $\tilde{x}_{i,d} \leftarrow \tilde{x}_{i,d} \cdot e$
6: *if* $spark_{i,d}$ *out of bounds then*
7: *Set* $\tilde{x}_{i,d} \leftarrow x_{min,d} + |\tilde{x}_{i,d}| \,\%(x_{max,d} - x_{min,d})$
8: *end if*
9: *end for*

The distance-based selection strategy in FWA is used to select other n-1 fireworks to formulate offspring generation. The selection probability $p(x_i)$ of x_i is calculated by:

$$R(x_i) = \sum_{j=1}^{n} dis(x_i - x_j) = \| x_i - x_j \| \tag{21}$$

$$p(x_i) = \frac{R(x_i)}{\sum_{j=1}^{n} R(x_i)} \tag{22}$$

where $R(x_i)$ represents the Euclid distance between two different solution array. Such a selection strategy is to able to ensure that individuals in low crowded regions may have a higher probability to be selected for next generation.

4. The proposed HFWA algorithm

4.1 The solution presentation

The solution to MTVRP is to group clients into trips and schedule trips on homogeneous vehicles under some constraints. In this regard, a novel solution representation architecture is proposed to accommodate problem characteristics on purpose of paving the way for FWA deployment.

Consider the scenario where n clients are to be served by m vehicles, the solution can be defined by n real numbers in interval $[1, m+1)$. The i-th number in solution array stores the information about the assign information of client i.

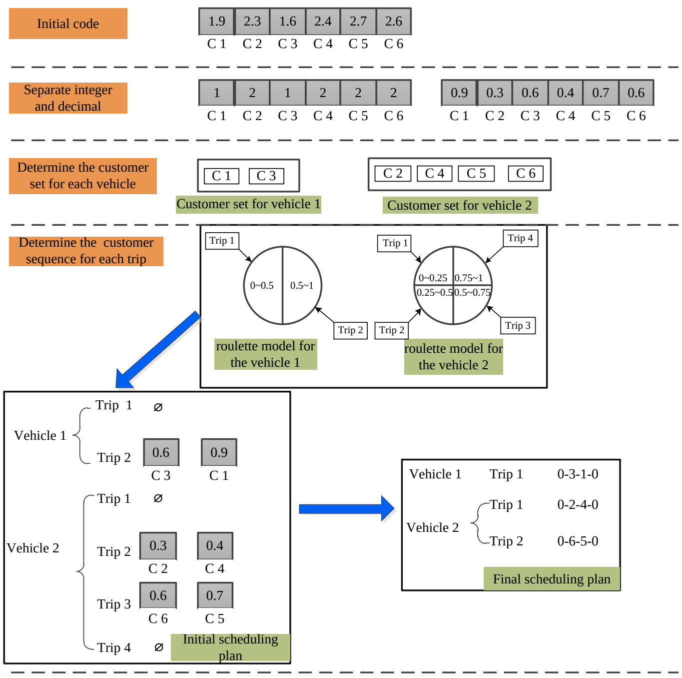

Fig. 2. The schematic diagram of encoding and decoding

The decoding process includes three steps. First, the integer and fractional parts of all coding values are separated to form two arrays. Second, the integer part of the i-th number is used to determine the vehicle index for client i. Then, the number of clients assigned to each vehicle can be known with certainty, and the trip size of a vehicle equals the number of clients assigned to it. On basis of this, the roulette rule is used to determine the trip index and visit priority of each client by virtue of the fractional part of the i-th number. Some minor adjustments are utilized to modify the final schedule, i.e., delete empty trips.

To facilitate understanding of above-mentioned description, Fig.2 presents an illustrative example. In this case, six clients are to be served by two vehicles, and thus the solution can be denoted by six real numbers in interval [1, 3). Given a solution array (1.9, 2.3, 1.6, 2.4, 2.7, 2.6), the interpreted schedule is as follows:

- Vehicle 1: only one trip with visit sequence 0→3→1→0
- Vehicle 2: two trips with visit sequences 0→2→4→0 and 0→6→5→0.

4.2 Evaluation function

The proposed solution representation architecture has efficiently taken advantage of the real-coding mechanism, which makes it possible to solve MTVRP with FWA. Meanwhile, to eliminate infeasible solution in the algorithm evolution, this research adopts the penalty function to establish the evaluation function. The expression is defined as follows:

$$F' = F + \lambda_1 \cdot \sum_{k\in K}\sum_{r\in R} \max\left(\sum_{j\in J} T_{j0k}^{r} \cdot x_{j0i}^{r} - T_H, 0 \right) + \lambda_2 \cdot \sum_{k\in K}\sum_{r\in R} \max\left(\sum_{i\in V}\sum_{j\in V} x_{ijk}^{r} \cdot q_i - Q, 0 \right) \quad (23)$$

where λ_1 and λ_2 are two penalty factors.

4.3 Fuse the initialization strategy of reverse learning

The opposition-based learning initialization method is embedded into FWA to diversify initial individuals and to find good initial solutions [21,22]. To be more specific, the initial population is created in a random manner, and the corresponding opposition-based population is generated. Then a greedy selection method is used to find some good solutions inside these two groups in order to form initial population with high qualities.

Give the population size n, let $x_i = (x_{i1}, \cdots, x_{id}, \cdots, x_{iD})$ be the solution array of the $i-th$ solution. The opposition-based $x_i^o = (x_{i1}^o, \cdots, x_{id}^o, \cdots, x_{id}^o)$ of x_i is defined as follows:

$$x_{id}^o = x_d^l + x_d^u - x_{id} \quad (24)$$

where $d = 1, 2, \cdots, D$ and is the length of solution array. In addition, $[x_d^l, x_d^u]$ boundary values of each dimension of soliton array. On basis of above-mentioned description, the Opposition-based learning initialization method is stated as follows

Step 1. Set $i \leftarrow 1$ and $d \leftarrow 1$. Then, go to Step 2.

Step 2. If $i \le D$, go to Step 3; otherwise, go to Step 7.

Step 3. If $d \le D$, go to Step 4; otherwise, go to Step 5.

Step 4. Generated solution array value x_{id} in a random way and calculated its opposition-based value x_{id}^o. Then, go to Step 5.

Step 5. Set $d \leftarrow d+1$. If $d > D$, set $d \leftarrow 1$ and go to Step 6; otherwise, go to Step 3.

Step 6. Set $i \leftarrow i+1$, and then go to Step 2.

Step 7. Rank these solutions according to their evaluation function values, and then select n best solutions to form the initial population.

4.4 Local evolutionary method of fusing swarm search

To better adapt FWA for MTVRP, this subsection designs a local search method by taking advantage of evolution strategies of ABC algorithm [23]. In each iteration of FWA, some ABC-based evolution strategies are used to enhance the performance FWA's offspring. The proposed ABC-based local search includes three significant components: individual selection probability model, individual update method and offspring selection strategy.

(1) Individual selection probability model

The probability model is used to select candidates from current population for ABC-based local search. To ensure that good soliton has higher opportunities to be selected, all individuals in current population are ranked firstly according to evaluation function values and then the selected probability r_i of the $i-th$ solution x_i is calculated as follows:

$$r_i = \frac{\exp(-rank_i)}{\sum_{j=1}^{n} \exp(-rank_j)} \tag{25}$$

where $rank_i$ denotes the rank index of solution x_i.

(2) Individual update method

The local search adopts the evolution strategy in ABC algorithm to update individuals. In this regard, given a selected solution x_i and another random solution x_r ($k \neq i$), the mutant individual $v_i = (v_{i1}, \cdots, v_{id}, \cdots, v_{iD})$ is calculated as follows:

$$v_{id} = x_{id} + \mathrm{rand}(-1,1) \cdot (x_{id} - x_{kd}) \tag{26}$$

where $\mathrm{rand}(-1,1)$ represents a random value on interval [-1,1].

(3) Offspring selection strategy

Give a selected solution x_i and its mutant solution v_i, the ABC-based local search adopts greedy selection strategy for individual retention. In other words, the solution with a better evaluation function is selected to form offspring generation.

4.5 The implementation process of HFW algorithm

Based on above descriptions, the outlines of HFWA algorithm are stated as follows:

Algorithm 4. *Outlines of HFWA*

1: Initialize algorithm parameters.
2: Generate n individuals by virtue of opposition-based learning initialization method
3: Evaluate the solution performance of n individuals.
4: ***Repeat***
5: calculate the number of explosion sparks for each individual;
6: calculate the explosion amplitude for each individual;
7: generate sparks for each individual;
8: generating Gaussian sparks for each individual;

9: *select the best individual into offspring generation;*
10: *randomly select the other $n-1$ offspring individuals based on a probability;*
11: *Build individual selection probability model and select n candidate solutions*
12: *Generated ABC-based mutant solutions by the proposed local search algorithm*
13: *Select solutions with better performance to formulate the offspring generation*
14: ***until** any algorithm stopping criteria is met.*
15: *Output the best individual.*

5. Simulation experiment and result analysis

5.1 Experimental description

To verify the optimization performance of the HFWA built in this paper, Matlab 2016a programming platform is used for simulation. The computer parameters are Intel Core I5-8250U CPU 1.6GHZ, and 8GB memory. We refers the relevant steps in literature [24] to generate the MTVRP test example in this paper, and the parameters are set as follows: the total quantity of customers $n=40$, the total quantity of vehicles $m=3$, the vehicle load capacity $Q=10$ (unit: ton); Table 1 shows the horizontal and vertical coordinates and demand values of all nodes, among which, 0 represents the distribution center; the planning period $[0, T_H]$ is defined as $[0,8]$, the unit is hour, and the delivery speed is set as 60 km/h.

Table 1

Parameters of problem

Number i	X-coordinate /km	Y-coordinate /km	Demand q_i /ton	Number i	X-coordinate /km	Y-coordinate /km	Demand q_i /ton
0	60	50	--	21	61	45	3
1	13	40	2	22	36	12	2
2	60	16	2	23	73	10	2
3	50	72	1	24	86	43	4
4	95	28	1	25	93	77	4
5	34	21	2	26	4	56	3
6	9	42	3	27	11	46	3
7	16	13	1	28	33	71	1
8	78	66	1	29	34	78	1
9	69	20	3	30	27	19	4
10	34	58	3	31	86	40	2
11	92	2	3	32	11	40	1
12	71	29	1	33	19	33	1
13	45	85	2	34	34	40	3

14	18	95	1	35	62	25	2
15	62	90	2	36	58	22	1
16	32	16	2	37	1	62	2
17	46	3	2	38	18	4	3
18	8	2	2	39	26	99	2
19	28	62	3	40	35	25	3
20	73	44	1				

5.2 Parameter calibration of HFWA

The iterative evolution of HFWA involves the following algorithm control parameters: the fireworks population number n, the iteration number G, the explosive spark parameter SN, the Gaussian spark number GN, the basic explosion radius A, the explosion radius integration parameters a and b. According to relevant studies, parameters a and b have little influence on optimization performance of the algorithm, so the recommended values in literature research are as follows: a=0.1, b=0.2 [17]. To obtain the optimal solution capability of HFWA, orthogonal experimental method is adopted to verify the remaining five parameters [25]. Each parameter is set at 4 levels respectively. Table 2 summarizes the values of each algorithm parameter at different levels. Meanwhile, Table 3 shows the orthogonal experimental arrangement with 5 factors and 4 levels. Based on this, the HFWA is used to conduct 15 independent simulation experiments under different algorithm parameter arrangements. The mean value $\bar{F}$ of the decision target obtained by the algorithm is taken as the response variable, Table 4 carries out the range analysis on the test results.

Table 2

Parameter setting of orthogonal test

Level	Factors				
	n	G	SN	GN	A
1	30	1000	10	8	0.4
2	40	1200	20	10	0.6
3	50	1500	30	14	0.8

Table 3

Test results of orthogonal experiment

No.	factors					$\bar{F}$	No.	factors					$\bar{F}$
	n	G	SN	GN	A			n	G	SN	GN	A	
1	1	1	1	1	1	19.63	15	2	2	3	1	3	18.74
2	1	1	1	1	2	20.23	16	2	3	1	2	1	20.43
3	1	1	1	1	3	17.98	17	2	3	1	2	2	19.27
4	1	2	2	2	1	18.13	18	2	3	1	2	3	21.09
5	1	2	2	2	2	19.62	19	3	1	3	2	1	18.92

6	1	2	2	2	3	20.71	20	3	1	3	2	2	18.34
7	1	3	3	3	1	21.08	21	3	1	3	2	3	20.97
8	1	3	3	3	2	19.79	22	3	2	1	3	1	20.96
9	1	3	3	3	3	19.78	23	3	2	1	3	2	21.22
10	2	1	2	3	1	19.30	24	3	2	1	3	3	19.19
11	2	1	2	3	2	19.67	25	3	3	2	1	1	17.71
12	2	1	2	3	3	19.47	26	3	3	2	1	2	18.24
13	2	2	3	1	1	19.88	27	3	3	2	1	3	18.46
14	2	2	3	1	2	20.77							

Table 4

Range analysis of orthogonal experiment

Level	Factors				
	n	G	SN	GN	A
1	58.98	58.18	60.00	57.21	58.68
2	59.54	59.74	57.11	59.16	59.05
3	58.01	58.61	59.42	60.15	58.80
4	1.53	1.56	2.90	2.94	0.37
range	4	3	2	1	5
grade	3	1	2	1	1

According to the orthogonal experimental results, the explosive spark parameter SN and the Gaussian sparks number GN have the greatest influence on the HFWA. These two parameters are used to balance the global search and the local mining of the HFWA, thus directly determining the optimization quality of the solution algorithm. Secondly, population size n and iteration number G also have great influence on the optimization performance of HFWA. Relatively speaking, the basic explosion radius parameter A has little influence on the optimization performance of HFWA, but it also needs to be set to ensure that the algorithm achieves better performance. To sum up, the five parameters of HFWA algorithm are set as follows: the number of fireworks population n is set as 50, the quantity of iterations G is set as 1000, the explosive spark parameter SN is set as 20, the quantity of Gaussian sparks GN is set as 8, and the basic explosion radius A is set as 0.6.

5.3 Analysis of test results

To verify the optimization effect of HFWA, it is compared with FWA, SA and PSO. To ensure the equity of algorithm comparison, the population size of the three comparison algorithms is set to the same value of HFWA, the population size of FWA, PSO and the number of internal cycles of SA are set to the same value as the population size of HFWA, and the other parameters are maintained to the same value as the references. For the above calculation example, the decision-making method runs independently for 15 times. Table 5 statistics the test results. Statistical indicators include: optimal value F^{opt}, worst value F^{wor}, mean value F^{mean}, standard deviation F^{std}, average percentage relative deviation (APRD), which are the parameters of the decision target [29], and average running time of the algorithm. APRD is calculated as follows:

$$APRD = \frac{F^{mean} - F^{lb}}{F^{lb}} \times 100\% \tag{22}$$

Table 5

Comparison of test results of the four algorithms

Algorithms	F^{opt}	F^{wor}	F^{mean}	F^{std}	$APRD$ /%	time/s
HFWA	17.19	17.71	18.29	1.46	3.00	576.30
FWA	17.89	18.70	19.25	1.52	8.79	544.75
SA	17.73	18.37	19.00	1.93	6.87	418.89
PSO	18.03	19.08	19.19	1.82	10.98	535.12

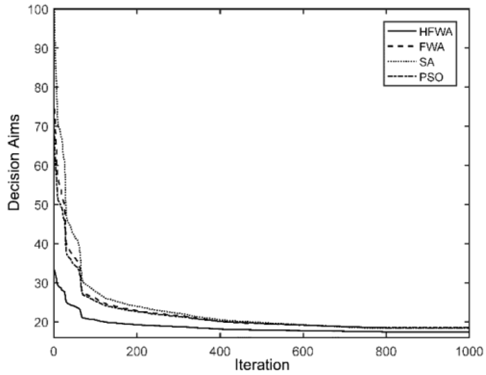

Fig. 3. Comparison of optimal evolution curves

In the expression, F^{mean} represents the mean value of the 15 times solution results of an algorithm, F^{lb} is the low-order value of the current example, and F^{lb} is replaced by the optimal value Z^{opt} of the four algorithms. Meanwhile, Fig. 3 shows the iterative process of the optimal solution evolution curve of the four algorithms, and the corresponding scheduling scheme is shown in Fig. 4. In addition, Table 6-9 summarizes the details of the optimal decision scheme.

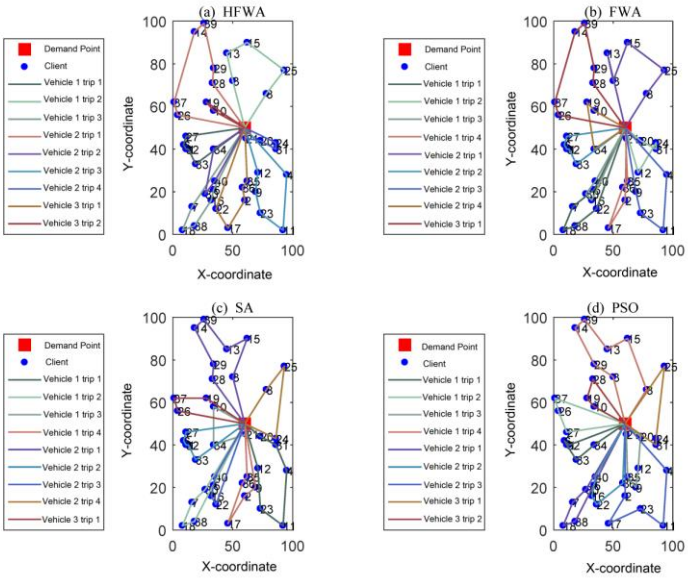

Fig. 4. Optimal scheme of the four algorithms

Table 6

The optimal scheme of HFWA algorithm

Vehicle No.	Trip No.	Line arragement	Line length /km	Time-consumption /km	Loads /ton
1	1	0→27→6→32→1→33→0	112.07	1.87	10
1	2	0→3→13→15→25→8→0	132.11	2.20	10
1	3	0→5→16→38→18→7→0	144.06	2.40	10
2	1	0→28→29→39→14→37→26→0	172.84	2.88	10
2	2	0→34→30→40→0	95.35	1.59	10
2	3	0→12→9→23→11→4→0	131.82	2.20	10
2	4	0→24→31→20→21→0	60.67	1.01	10
3	1	0→35→36→2→17→22→0	113.91	1.90	9
3	2	0→10→19→0	68.59	1.14	6

Table 7

Optimal scheme obtained by FWA

Vehicle No.	Trip No.	Line arrangement	Line length/km	Time-consumption /hour	Loads /ton
1	1	0→5→7→18→38→22→0	147.09	2.45	10
1	2	0→20→31→24→12→0	75.14	1.25	8
1	3	0→40→30→16→0	95.23	1.59	9
1	4	0→21→2→17→36→35→0	105.77	1.76	10
2	1	0→8→25→15→3→13→0	149.94	2.50	10
2	2	0→33→1→32→6→27→0	112.07	1.87	10
2	3	0→4→11→23→9→0	130.22	2.17	9
2	4	0→34→19→10→0	85.07	1.42	9
3	1	0→26→37→14→39→29→28→0	172.84	2.88	10

Table 8

Optimal scheme obtained by SA algorithm

Vehicle No.	Trip No.	Line arrangement	Line length /km	Time-consumption/hour	Loads /ton
1	1	0→12→23→11→4→31→20→0	132.52	2.21	10
1	2	0→38→18→7→30→0	143.90	2.40	10
1	3	0→10→34→21→0	77.76	1.30	9
1	4	0→35→36→17→2→9→0	112.83	1.88	10
2	1	0→28→29→14→39→13→15→3→0	160.69	2.68	10
2	2	0→27→6→32→1→33→0	112.07	1.87	10
2	3	0→40→5→16→22→0	95.46	1.59	9
2	4	0→24→25→8→0	104.32	1.74	9
3	1	0→19→37→26→0	124.20	2.07	8

Table 9

Optimal scheme obtained by PSO algorithm

Vehicle No.	Trip No.	Line arrangement	Line length /km	Time-consumption /hour	Loads /ton
1	1	0→34→33→32→6→1→0	110.39	1.84	10
1	2	0→27→26→37→0	128.29	2.14	8
1	3	0→21→35→9→12→20→0	72.40	1.21	10
1	4	0→3→29→14→39→13→15→8→0	167.79	2.80	10
2	1	0→38→18→7→30→0	143.90	2.40	10
2	2	0→40→5→16→22→36→0	102.76	1.71	10
2	3	0→4→11→23→17→2→0	169.13	2.82	10
3	1	0→25→24→31→0	108.21	1.80	10
3	2	0→10→19→28→0	78.91	1.32	7

Based on the comparison of the 15 times results with independent running, it can be seen from the above optimization results: as far as the optimization index F^{opt}, the optimal solution of HFWA algorithm is 17.19, which is superior to FWA (F^{opt} is17.89), SA (F^{opt} is 17.73) and PSO (F^{opt} is 18.03). At the same time, the indexes F^{opt} of the four algorithms are 18.29, 19.25, 19.00 and 19.19, corresponding $APRD$ indexes are 3.00%, 8.79%, 6.87% and 10.98% respectively. The performance of HFWA is the best, so the hybrid algorithm proposed in this paper can obtain the high-quality scheduling scheme of MTVRP. In addition, as far as F^{std} index, the index data corresponding to the four algorithms are 1.46, 1.52, 1.93 and 1.82 respectively, and HFWA performs the best. In other words, HFWA not only achieves high quality scheme, but also has strong solution stability. Finally, in terms of algorithm running time, the value corresponding to SA is the minimum (418.89 seconds), while the value corresponding to HFWA is the maximum (576.30 seconds). In general, the running time of the tested algorithm is within the same order of magnitude, and both of them are relatively short, indicating that evolutionary algorithm has strong practical application value for solving MTVRP problem.

6. The Conclusion

This paper investigated MTVRP with considerations of capacity and time constraints. A MILP model is formulated to minimize total transportation time, and a hybrid approach named HFWA is proposed for solution generation. In algorithm design, a novel solution representation architecture is utilized to accommodate MTVRP's characteristic. In addition, an opposition-based learning initialization method is introduced to generate initial solutions with high qualities. In order to balance the algorithm's exploration and exploitation abilities, the solution mutation strategy of ABC is embedded into the FWA's evolution. Computational results indicate that HFWA is effective and efficient in solving MTVRP when compared to other metaheuristic algorithms.

With respect to future research, an interesting research hotpot worth researching is to apply HFWA to more complicated problems, such as dynamic MTVRPs. Meanwhile, another research direction is to formulate new mathematical models of MTVRPs with the consideration of green manufacturing or sustainability.

Acknowledgement

This paper is supported by the National Nature Science Foundation of China (No. 60773212).

R E F E R E N C E S

[1]. *M. Huang, X. Chen*, "Logistics Vehicle Scheduling Based on Genetic Algorithm", 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2020.

[2]. *H. Duan, Y. Ye, Z. Lei, M. Wang*, "Express delivery logistics with high-speed railway: a perspective of payment scheme and forecast information sharing", in Complex & Intelligent Systems, **vol.** 3, no. 18, 2021, pp. 1-13.

[3]. *A. Cinar, F. S. Salman and B. Bozkaya*, "Prioritized single nurse routing and scheduling for home healthcare services", in European Journal of Operational Research, **vol.** 289, no. 3, 2021, pp. 867-878.

[4]. *F. Hernandez, D. Feillet, R. Giroudeau and O. Naud*, "Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows", in European Journal of Operational Research, **vol.** 249, no. 2, 2016, pp.551-559.

[5]. *A. Mr, C. Gpb, C Gc, et al.*, "Biomechanical study of autograft anatomic reconstruction in lateral ankle instability", in Revista Española de Cirugía Ortopédicay Traumatología (English Edition), **vol.** 65, no. 2, 2021, pp.124-131.

[6]. *Y. Wang, Q. Li, X. Guan, et al.*, "Two-echelon collaborative multi-depot multi-period vehicle routing problem", in Expert Systems with Applications, **vol.** 167, no. 114201, 2020, pp.5-8.

[7]. *B. Pba, A. MD, A. Ta, et al.*, "Modeling and evaluating agile assembly systems using mixed-integer linear programming", in Procedia CIRP, **vol.** 93, Apr. 2020, pp.1073-1078.

[8]. *M. Battarra, M. Monaci and D. Vigo,* "An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem", in Computers & Operations Research, **vol.** 36, no.11, 2009, pp. 3041-3050.

[9]. *Hashemi, S., M. Salari, and M. Ranjbar,* "Multi-Trip Open Vehicle Routing Problem with Time Windows: A case study", in Operation Research, **vol.** 27, no.1, 2020.

[10]. *H.Hsu, C.Wang,* "A Hybrid Approach Combining Improved Shuffled Frog-leaping Algorithm with Dynamic Programming for Disassembly Process Planning", in IEEE Access, **vol.** 9, no. 4. 2021, pp: 57743-57756.

[11]. *S. Salhi, R. Petch,* "A GA based heuristic for the vehicle routing problem with multiple trips", in Journal of Mathematical Modelling and Algorithms, **vol.** 6, no. 4, Jul. 2007, pp.591-613.

[12]. *V. François, Y. Arda, Y. Crama and G. Laporte,* "Large neighborhood search for multi-trip vehicle routing", in European Journal of Operational Research, **vol.** 255, no. 2, Apr. 2016, pp. 422-441.

[13]. *R. Saxena, M. Jain, K. Malhotra and K. Vasa,* "An optimized openmp-based genetic algorithm solution to vehicle routing problem", in Smart Computing Paradigms: New Progresses and Challenges. Springer, Singapore, **vol.** 767, Dec. 2019, pp. 237-245.

[14]. *Y. Tan,* Fireworks algorithm, Heidelberg, Germany: Springer, 2015.

[15]. *X. Chen, X. Wei, G. Yang and W. Du,* "Fireworks explosion based artificial bee colony for numerical optimization", in Knowledge-Based Systems, **vol.** 188, no. 5, Jan. 2020.

[16]. *Y. Chen, L. Li, X. Zhao, et al.,* "Simplified hybrid fireworks algorithm", in Knowledge-Based Systems, **vol.** 173, no. 1, Jun. 2019, pp. 128-139.

[17]. *S. Ye, H. Ma, S. Xu, et al.,* "An effective fireworks algorithm for warehouse-scheduling problem", in Transactions of the Institute of Measurement and Control, **vol.** 39, no. 1, 2017, pp. 75-85.

[18]. *J. Ji, H. Xiao and C. Yang,* "HFADE-FMD: a hybrid approach of fireworks algorithm and differential evolution strategies for functional module detection in protein-protein interaction networks", in Applied Intelligence, **vol.** 51, no. 6788, 2021, pp. 1-15.

[19]. *Y. Xue, B. Zhao, T. Ma and W. Pang.* "A self-adaptive fireworks algorithm for classification problems", in IEEE Access, **vol.** 6, no. 1, 2018, pp. 44406-44416.

[20]. *X. Li, S. Han and C. Gong,* "Analysis and improvement of fireworks algorithm", in Algorithms, **vol.** 10, no. 1, Feb. 2017, pp.1-22.

[21]. *S. Mahdavi, S. Rahnamayan and K. Deb,* "Opposition based learning: A literature review", in Swarm and evolutionary computation, **vol.** 39, no.1 Apr. 2018, pp. 1-23.

[22]. *S. Liu, Z. Zhang, C. Guan and X. Zhu,* "An improved fireworks algorithm for the constrained single-row facility layout problem", in International Journal of Production Research, **vol.** 59, no. 8, Feb. 2021, pp. 2309-2327.

[23]. *Y. Xue, J. Jiang, B. Zhao and T. Ma,* "A self-adaptive artificial bee colony algorithm based on global best for global optimization", in Soft Computing, **vol.** 22, no. 9, 2018, pp. 2935-2952.

[24]. *D. Cattaruzza, N. Absi, D. Feillet and T. Vidal,* "A memetic algorithm for the multi trip vehicle routing problem", in European Journal of Operational Research, **vol.** 236, no. 3, 2014, pp. 833-848.

[25]. *R. Ruiz, T. Stützle,* "An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives", in European Journal of Operational Research, **vol.** 187, no.3, Jun. 2008, pp.1143-1159.