Journal of Network Intelligence ©2020 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **5**, Number **3**, August **2020**

Improved Flower Pollination Algorithm for the Capacitated Vehicle Routing Problem

Jiawen Zhuang[1], Hao Luo[2], Tien-Szu Pan[3] and Jeng-Shyang Pan[1,4,5]*

[1]School of Information Science and Engineering,
Fujian University of Technology,Fuzhou, 350118, China

[2]School of Aeronautics and Astronautics,
Zhejiang University, Hangzhou,310058,China

[3]Department of Electronic Engineering,
National Kaohsiung University of Science and Technology,Kaohsiung,Taiwan

[4]College of Computer Science and Engineering,
Shandong University of Science and Technology,Qingdao,266590,China

[5]College of Science and Engineering,
Flinders University,Adelaide,5042,Australia

Received May 2020; revised July 2020
(Corresponding author: Jeng-Shyang Pan(jengshyangpan@gmail.com))

Abstract. *Capacitated Vehicle Routing Problem (CVRP), as a transportation problem, is often use the meta-heuristic algorithm to solve it and can obtain an approximate optimal solution. Flower pollination algorithm (FPA) is a meta-heuristic algorithm and we propose an improved flower pollination algorithm (IFPA) in this paper. In IFPA, we enhance the global search capability of FPA by introducing a random jump perturbation in the global pollination phase and update the switching probability according to the global optimal value of each iteration. Then we test the IFPA by CEC2013, IFPA has better performance in convergence than FPA, particle swarm optimization (PSO), and differential evolution (DE). Finally, IFPA is also applied to solve CVRP in many instances. From the test results of the instances, IFPA is more suitable to solve the CVRP problem than FPA, PSO, and DE.*
Keywords: meta-heuristic algorithm, flower pollination algorithm, CEC2013, capacitated vehicle routing problem

---

1. **Introduction.** The meta-heuristic algorithm is widely used in transportation problems. Some meta-heuristic algorithms are population-based algorithm [1, 2]. Differential Evolution (DE) [3, 4, 5], Ant Colony Optimization [6, 7, 8], Cuckoo Search (CS) [9, 10, 11], Particle Swarm Optimization (PSO) [12, 13, 14], Artificial Bee Colony (ABC) algorithm [15, 16, 17], Cat Swarm Optimization (CSO) [18, 19, 20], Bat Algorithm (BA) [[21, 22, 23] and Grey Wolf Optimization (GWO) [24, 25, 26] were some popular population-based algorithms. Recently Meng et. al. proposed the QUasi-Ane TRansformation Evolutionary (QATRE) [27, 28, 29, 30] to implement the DE algorithm based on the Matrix process to avoid the bias of crossover operation. FPA, proposed by Yang [31], is a novel meta-heuristic algorithm. PSO is the foraging behavior of natural birds, ABC mimics bee, and the idea of FPA comes from the pollination mechanism of flowers.

Since the FPA was introduced, many researchers have proposed many improvements. Dubey et al. proposed a biologically inspired modified FPA, which uses constant factor rather than a random number to control the local pollination for increasing convergence, adding intensive exploitation for exploiting the best solution [32]. Abdel-Raouf et al. proposed an improved FPA with chaos. Which uses chaotic mapping to determine the switching probability, levy parameter, and local pollination random number [33]. Wang uses three strategy to improve FPA, such as local neighborhood search, dimension by dimension evaluation and improvement, and dynamic switching probability [34]. The compact FPA is also presented for the layouts of nodes in Wireless Sensor Network [35].

The history of the vehicle routing problem (VRP) can be traced back to Dantzig's work. Dantzig in 1959 firstly proposes the truck dispatching problem [36]. Many scholars have proposed many diverse models for diverse types of VRP and many methods to obtain the optimal or suboptimal solution of the VRP [37].

VRP is crucial whether in transportation or research. On the one hand, the optimization of vehicle usage scheme and vehicle driving path scheme is of great significance for improving the operation quality of the transportation system. On the other hand, it is a challenging research subject. One kind of VRP, CVRP is an NP-hard problem [38]. It usually takes a long computational time to get the optimal solution. Therefore, the meta-heuristic algorithm is often applied to solve the CVRP and obtain an approximate optimal solution.

Enhanced ABC algorithm, proposed by Szeto, is applied to solve CVRP [39]. Chen puts forward a new hybrid PSO for solving CVRP, which is hybrid discrete PSO and simulated annealing, and discrete PSO search optimum and simulated annealing get out of local optimum [38]. Mazzeo used ant colony algorithm to solve CVRP [40]. Teymourian proposes two hybrid algorithms for solving CVRP, which hybrid improved intelligent water drops algorithm and advanced cuckoo search algorithm [41]. Our work has some benefits and advantages. Comparing with some exact algorithm, such as Branch and Bound [42, 43], Branch and Cut [44, 45], IFPA is more suitable for large-scale problems and can obtain satisfactory solutions. Moreover, our work is more reliable because we test more instances than Wu [46] that just test 10 instances.

In this paper, our work has several contributions. One is to introduce random jump disturbance in the global pollination of FPA, and make the switching probability change with the fitness value, thus improving the FPA. The other one is that IFPA can effectively solve CVRP and get a better solution than the other three algorithms.

The following is the remaining of this paper. In section 2, the CVRP model is described, FPA and IFPA were described in section 3. In section 4, the experiments of CEC2013 functions [47] and CVRP are described. In section 5, a conclusion is given.

2. **CVRP.** Minimizing the distance or the total cost for all vehicles is the objective of CVRP. In this paper, we select distance as our goal. The customer's total demand on each route should not surpass the capacity of the vehicles servicing that route. The starting and ending point is the same depot. The CVRP model is as follows [38, 46]:

$$\min f(X) = \sum_{k=1}^{K_{vehicle}} \sum_{i=0}^{N_c} \sum_{j=0}^{N_c} c_{ij}^k x_{ij}^k \tag{1}$$

s. t.

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ travels from customer } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$y_i^k = \begin{cases} 1 & \text{if vehicle } k \text{ serves customer } i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\sum_{i=1}^{N} d_i y_i^k \leq Q_k, k = 1, 2, ..., K_{vehicle} \tag{4}$$

$$\sum_{k=1}^{K} y_t^k = 1, i = 1, 2, ..., N_c \tag{5}$$

$$\sum_{i=0}^{N} x_{ij}^k = y_j^k, j = 0, 1, 2, ..., N_c; k = 1, 2, ..., K_{vehicle} \tag{6}$$

$$\sum_{j=1}^{N} x_{ij}^k = y_i^k, i = 0, 1, 2, ..., N_c; k = 1, 2, ..., K_{vehicle} \tag{7}$$

$$\sum_{i=0}^{N} x_{it}^k = \sum_{j=0}^{N} x_{tj}^k, t = 1, 2, ..., N_c; k = 1, 2, ..., K_{vehicle} \tag{8}$$

$$\sum_{i,j \in S \times S} x_{ij}^k \leq |S| - 1, \mathrm{S} \subset \{1, 2, ..., N\}, S \neq \Phi, k = 1, 2, ..., K_{vehicle} \tag{9}$$

where is N_c the number of customers, and $K_{vehicle}$ is the number of vehicle, c_{ij}^k is the distance of traveling from the ith customer to the jth customer by the kth vehicle, d_i is the demand of the ith customer. The capacity of the kth vehicle is Q_k . The depot is node 0.

Formula (1) is the objective function that is to minimize the total distance by all vehicles. In formula (2), if the kth vehicle travels from the ith customer to the jth customer, x_{ij}^k equals to 1, otherwise equals to 0. In (3), if the kth vehicle serves the ith customer, y_i^k equals to 1, otherwise equals to 0. Formula (4) shows that total demand in each route should not exceed vehicles capacity. Formula (5) guarantee that every customer is served only once. Formula (6) and (7) ensures one vehicle serves one custom. Formula (8) ensures the continuity of the route so that every car coming in from the customer point would go out from that point, as well as back to the depot. Finally, formula (9) is to eliminate sub-loop.

3. FPA and IFPA.

3.1. FPA.

It is estimated that tens of thousands of flowering plants exist in nature and can be seen everywhere. FPA is an algorithm abstracted from the pollination mechanism. Pollination could be taken by abiotic form or biotic form and achieved through self-pollination or cross-pollination. FPA as a population-based heuristic algorithm, its local pollination contains abiotic and self-pollination, which indicates the exploiting in the search area, its global pollination contains biotic and cross-pollination, which represents global exploration, the proportion of above two pollination process is controlled by its switching probability. The global pollination of FPA is modeled as follows [31].

$$Pollen_i^{t+1} = Pollen_i^t + Levy_{step} \times (Pollen_i^t - Pollen_*) \tag{10}$$

where $Pollen_i^t$ is the ith pollen of the tth iteration, $Pollen_*$ is the best pollen of the current iteration among all pollens. In this paper $Levy_{step} > 0$ and draw from Levy distribution. FPAs local pollination for exploiting is modeled as follows [31].

$$Pollen_i^{t+1} = Pollen_i^t + \varphi_{\text{rand}} \times (Pollen_p^t - Pollen_q^t) \tag{11}$$

where $Pollen_p^t$ and $Pollen_q^t$ are pollens from the identical plant diverse flowers, and φ_{rand} is the uniform distribution number in [0,1]. The FPAs pseudo-code shows in Figure 1.

FPA

Fitness Function $F(Pollen)$, solution $Pollen_i = (pol_1, pol_2 \ldots, pol_d)$
Initialize $Pollen$ randomly and switching probability $p \in [0,1]$
Calculate $F(Pollen^t)$ and find the best solution $Pollen_*$.
For $t = 1, \ldots,$ MaxIteration **do**
    For $i = 1, \ldots, N$ **do**
        If $rand \leq p$ **then**
            Global pollination via (10)
        else
            Local pollination via (11)
        end if
        Calculate $F(Pollen_i^{t+1})$
        Update them, when new pollens are better.
    end For
        Update the best solution $Pollen_*$
end For

FIGURE 1. FPA's pseudo-code.

3.2. **IFPA.** In terms of software and hardware, FPA has strong practicability. It also has some shortcomings such as slow convergence rate and worse precision. Given the shortcomings of FPA, we propose an IFPA in this paper. In IFPA, we introduce a random jump perturbation operator in the global pollination phase that enhances FPAs global search ability, and a novel switching probability updating strategy that using global optimum to control the switching probability.

The shortcomings of FPA, such as slow convergence rate and easy to fall into local solutions, are closely related to its global pollination. When the global pollination area is more narrow, the solution will easily fall into local solution, while more wide will slow the convergence rate. How to balance the degree of global pollination is a very important thing. Therefore, we propose a random jump perturbation, which makes the particle jump to a randomly designated point around, and controls whether it jumps, whether it jumps to the point or away from the point by the operator. The randomness of global pollination is enhanced, and the possibility of jumping out of local solutions is improved. It is also restricted to a one-step jump to prevent its convergence too slow. The enhanced global pollination formula is as follows:

$$Pollen_i^{t+1} = Pollen_i^t + Levy_{step} \times (Pollen_i^t - Pollen_*) + \alpha \times (Pollen_i^t - Pollen_k) \tag{12}$$

where α is the random integer number in [-1,0,1], and $Pollen_k$ is the random pollen different from $Pollen_i^t$.

The switching probability controls the ratio of global pollination and local pollination. When the switching probability is larger, the FPAs global search ability is reinforced, which is helpful to shun the local optimal solution, but the local pollination is weakened and the optimization accuracy is low. When the switching probability is smaller, the situation is reversed.

Based on this, we design a dynamic switching probability operator. If the optimal value found at this generation is less than the optimal value found in the previous generation, we think it may drops into the local optimal solution, so we increase the global pollination proportion and make the switching probability of the next generation $p^{t+1} = 1.5p^t$. Conversely, we think it approaches the optimal solution, so we increase the local pollination proportion, that is, reducing the global pollination proportion $p^{t+1} = 0.8p^t$.

In this section, the step of IFPA will be described. We set the total population is POPSIZE . Firstly, pollen is initialized randomly, and its fitness value is calculated, the original switching probability parameter is initialized. Then for each iteration pollen updates by (12) and (13), and novel switching probability updating strategy is performed after iteration. While meeting the max number of iteration or calculate fitness is just less than the threshold fitness, terminate the algorithm. The following are the detailed steps.

Step 1. Initialization: Produce POPSIZE pollens $Pollen^t$ with D dimensions, where POPSIZE is pollen number, t is the present iteration and set $t = 1$, initialize switching probability p=0.8 .

Step 2. Evaluation: Calculate $f(Pollen_i^t)$ for the ith pollen. Find the populations best pollen $Pollen_*$ and its fitness.

Step 3. Pollen Update: According to the present switching probability, the ith pollen $Pollen_i^t$ choose enhanced global pollination by (12), or local pollination by (11), to generate new pollen $Pollen_i^{t+1}$. In (12), α is the random integer number in [-1,0,1]. Calculate $f(Pollen_i^{t+1})$. Update pollen, when new pollens are better. Update the best pollen $Pollen_*$ and its fitness in population.

Step 4. Perform Strategy: For every iteration, Comparing best pollens fitness of this iteration with that of the previous iteration. If its no better than that of the previous iteration, update the switching probability with $p^{t+1} = 1.5p^t$, otherwise, $p^{t+1} = 0.8p^t$.

Step 5. Termination: Step 3 to 5 are repeated until reaching the given threshold fitness, or maximum iteration. Finally, record the best fitness $f\ (Pollen_*)$ and best pollen $Pollen_*$ among the population.

3.3. **Solution representation.** Both the FPA and the IFPA are continuous optimization algorithms, which are often used in continuous optimization problems. For CVRP is a discrete problem, we need to encode and decode each solution, that is, to discretize the FPA and IFPA algorithm.

There are many papers about continuous optimization algorithms to discrete for solving the CVRP problem, such as Wu proposes a real number encoding method, solutions dimension equals to the number of custom, and the decoding only requires a sorting and integer operation [46]. Szeto also proposes a real number encoding method, the solution is n + m dimensions, that is m vehicles and n customers, and the path is the sequence of numbers between the two zeros in the solution [39]. Ai et al. propose another real number coding method named SR-1. the solution is n + 2m dimensions. Each solution has n dimension that is corresponding to customers, and the 2m dimension is the coordinate of vehicles. The n-dimensional customer produces the customer priority matrix, and the 2m dimensional produces the vehicle priority matrix. The path is obtained by combining the two matrices, that is, the feasible solution [48].

In this paper, the SR-1 encoding method is used for encoding, and local optimization methods such as 2-opt, 1-1 exchange, and 1-0 exchange are adopted [48].

4. **Experiment and Application.** We utilized 28 benchmark functions of the CEC2013 to test our proposed algorithm comparing with PSO [12],FPA [31],and DE [49]. Detail function is shown in table 1, and experimental results are shown in table 2 to 6. Next,

we select some functions convergence curves to compare the rate of convergence of 4 algorithms, details are shown in figure 2 to 6. Finally, we apply the IFPA to solve the CVRP, the result is shown in table 7 to 8.

TABLE 1. Function of CEC2013.

No.	**Type**	**Optimum**	**No.**	**Type**	**Optimum**
F1	Unimodal	-1400	F15	Basic Multimodal	100
F2	Unimodal	-1300	F16	Basic Multimodal	200
F3	Unimodal	-1200	F17	Basic Multimodal	300
F4	Unimodal	-1100	F18	Basic Multimodal	400
F5	Unimodal	-1000	F19	Basic Multimodal	500
F6	Basic Multimodal	-900	F20	Basic Multimodal	600
F7	Basic Multimodal	-800	F21	Composition	700
F8	Basic Multimodal	-700	F22	Composition	800
F9	Basic Multimodal	-600	F23	Composition	900
F10	Basic Multimodal	-500	F24	Composition	1000
F11	Basic Multimodal	-400	F25	Composition	1100
F12	Basic Multimodal	-300	F26	Composition	1200
F13	Basic Multimodal	-200	F27	Composition	1300
F14	Basic Multimodal	-100	F28	Composition	1400
Search Range: $[-100, 100]^D$					

4.1. **Experiment with CEC2013 function.** In the CEC2013 test, for the randomness of the meta-heuristic algorithm and comparing fairly, algorithms run independently for 51 times. For D=2, the iteration is 400. For D=5, the iteration is 800. For D=10, the iteration is 1000. For D=20, the iteration is 1500. For D=30, the iteration is 2000. Four algorithms maximum function evaluation is the same in different dimensions, for each algorithm has the same function evaluation in each iteration. The range of each dimension of the decision variable is [-100,100]. The learning factor of PSO, c_1 and c_2 equal to 2, and the inertia constant ω equals to 0.8, the number of population is 400. FPA and IFPAs switching probability p=0.8 and $\lambda = 1.5$. The number of pollen is 400, As for DE, F=2 and CR=0.9. Therefore, the result obtained is reliable.

Table 1 shows that, the diversity of test functions guarantees the reliability of our experiment. Table 2 to 6 show the algorithm result of D=2, 5, 10, 20, 30. In each table, the data form is Mean/Std, and in the final line, the form is the recording best number in Mean/ Std. The Mean is the mean value of 51 runs and Std is stand deviation. In each table, if an algorithm achieves the best of the four algorithms in a function test, the number of records is increased by one. Note that two algorithms get the same best results in the same function, we record both algorithms getting the best results.

As shown in Table 2, with D=2, In mean value, IFPA achieves the best of the four algorithms in 19 functions. As for standard deviation, IFPA obtains the best in 20 functions. From table 3, with D=5, the IFPA obtains a better convergence precision effect in 18 test functions in Mean item. In the Std item, IFPA is more stable than other algorithms in 17 functions. In table 4, with D=10, the IFPA has the better mean than others in Mean item, but it has the approximate stability in standard deviation with FPA. In table 5 and 6, whether in Mean item or Std item, IFPA achieves the best result in most function than the other 3 algorithms. IFPA has better convergence precision and stability than FPA, PSO, and DE.

TABLE 2. Mean/Std result of algorithms with D=2.

No.	IFPA	FPA	PSO	DE
F1	0/0	1.87e-07/2.01e-07	0/0	0/0
F2	0/0	2.21e-09/3.39e-09	3.37e-12/8.97e-12	2.18e-13/2.23e-13
F3	3.12e-13/8.25e-13	1.34e-06/2.65e-06	2.72e-11/7.31e-11	9.46e-12/9.77e-12
F4	0/0	1.19e-09/1.31e-09	4.55e-12/8.46e-12	2.63e-13/2.6e-13
F5	0/0	2.88e-05/2.37e-05	1.02e-09/2.45e-09	0/0
F6	0/0	3.53e-07/3.99e-07	0/0	0/0
F7	6.92e-07/2.38e-07	0.083102/0.045169	9.10e-08/1.23e-07	2.19e-05/9.93e-06
F8	1.76e-07/2.40e-07	1.2018/0.70158	3.87e-08/6.42e-08	5.08e-07/3.71e-07
F9	4.70e-04/2.73e-04	1.32e-02/4.62e-03	2.89e-03/2.06e-02	7.59e-05/1.27e-04
F10	7.05e-07/1.01e-06	3.58e-03/2.47e-03	3.51e-09/1.67e-08	4.45e-04/4.73e-04
F11	0/0	0.001201/0.001071	0/0	0/0
F12	0/0	2.31e-03/2.35e-03	0/0	1.23e-10/2.58e-10
F13	0/0	2.96e-03/3.59e-03	0/0	3.30e-12/6.59e-12
F14	6.64e-13/3.49e-12	9.96e-02/9.04e-02	1.85e-02/7.42e-02	6.42e-06/2.12e-05
F15	2.21e-05/4.67e-05	8.09e-02/7.44e-02	5.51e-02/1.20e-01	0.14206/0.13946
F16	0.07264/0.045947	0.053375/0.03245	0.084802/0.068451	1.44e-02/8.73e-03
F17	8.33e-03/8.16e-03	0.10403/0.082287	0.0067931/0.02717	0.012614/0.010241
F18	0.05494/0.037594	0.14435/0.070128	0.090263/0.39181	0.12375/0.075685
F19	0/0	3.74e-07/7.70e-07	0/0	0/0
F20	3.78e-06/4.87e-06	1.09e-02/6.58e-03	7.66e-04/3.81e-03	1.28e-03/1.21e-03
F21	1.07e-10/1.05e-10	0.30095/0.17872	7.45e-08/1.02e-07	1.99e-08/1.48e-08
F22	7.75e-10/1.25e-09	0.56451/0.43635	7.50e-08/2.52e-07	9.73e-07/9.34e-07
F23	2.33e-05/4.08e-05	1.331/1.0559	2.46e-08/3.10e-08	3.97e-05/4.39e-05
F24	1.29e-08/2.65e-08	0.10127/0.070979	3.60e-09/4.12e-09	1.23e-06/1.05e-06
F25	1.80e-08/2.54e-08	0.30529/0.24379	2.29e-08/3.07e-08	2.97e-07/2.02e-07
F26	1.08e-08/2.66e-08	0.017173/0.013251	8.55e-03/6.09e-02	4.55e-05/1.45e-04
F27	0.32485/0.53116	9.6213/5.6215	0.15043/0.33134	0.6618/0.46372
F28	1.64e-10/2.17e-10	0.31096/0.24254	6.39e-08/6.66e-08	3.23e-08/1.59e-08
Total	19/20	0/0	13/12	7/7

Figure 2 to 6 is the convergence curves of IFPA, FPA, PSO, and DE in CEC2013. As shown in figure 2, with D=2, the blue curve is IFPA, the red curve is FPA, and the yellow curve is PSO and the line with the right-pointing triangle marker symbol is DE. In figure 2, we iterate 400 times and take 4 samples for every 100 iterations to plot the convergence curves. The optimization curves of F14, F15, F20, F26 reflect the IFPA has a better convergence rate than FPA, PSO, and DE. Similarly, in Figure 3 to 6, the curves are about four algorithms at D=5, 10, 20, 30. IFPA has a good rate of convergence than FPA, PSO, and DE.

4.2. **CVRP Application.** In this section, we apply IFPA, FPA, PSO, and DE for solving CVRP. Just as CVRP is a discrete problem, we use SR-1 methods for solution representation and decoding. We test in 26 instances of A and 22 instances of B. For more effectively, we run 10 times for getting more reliable data. The particles or pollens are 50. The iteration is 1000. The other parameters are the same as above mentioned. The CVRP experiment is shown in table 7 to 8. As is shown in table 7, IFPA, FPA, PSO, and DE are tested in 26 instances, such as A-n32-k5. A-n32-k5 means 32 points including depot.

Table 3. Mean/Std result of algorithms with D=5.

No.	IFPA	FPA	PSO	DE
F1	0/0	7.81e-05/3.26e-05	0/0	0.51347/0.23659
F2	1.01e-10/7.70e-11	1.22e-02/8.90e-03	0/0	1.29e+04/4.93e+03
F3	6.16e-05/6.21e-05	455.5759/392.846	5.0084/23.8662	3.06e+5/1.63e+5
F4	7.89e-13/5.59e-13	7.04e-03/3.18e-03	0/0	321.3541/132.8914
F5	0/0	1.48e-03/5.38e-04	4.41e-11/9.60e-11	0.22899/0.088351
F6	4.31e-09/4.03e-09	1.85e-03/1.32e-03	0.9249/1.684	0.7528/0.24972
F7	0.071877/0.026677	2.1302/0.63397	0.36214/0.54636	3.2476/0.85147
F8	11.0371/4.7572	15.0488/3.9881	15.8611/8.0451	14.8774/2.4767
F9	0.74865/0.16424	0.57891/0.17962	0.61816/0.58909	1.8228/0.27069
F10	0.047564/0.013241	0.06869/0.019296	0.154/0.099475	1.0703/0.15078
F11	7.30e-09/1.75e-08	1.7728/0.4158	3.5701/2.7871	6.1993/1.4919
F12	0.97526/0.44587	2.1412/0.68232	3.1021/1.7646	9.2642/1.8296
F13	1.1133/0.51809	2.3675/1.0669	4.9868/2.3768	9.9102/2.3477
F14	1.5731/0.80576	34.7244/18.0338	132.67/91.474	128.4688/49.9083
F15	77.1249/36.0592	66.4823/40.8411	128.347/90.1638	222.477/68.1575
F16	0.45206/0.10925	0.3015/0.067634	0.35045/0.20416	0.51869/0.12252
F17	4.2517/1.173	5.8637/1.7956	6.2929/1.9921	14.6506/2.1267
F18	5.6477/1.1148	6.3041/1.6026	6.9291/1.4929	15.9153/2.706
F19	0.037976/0.038452	0.10808/0.047396	0.21276/0.092409	0.69442/0.19399
F20	0.29899/0.074889	0.52189/0.14732	0.28232/0.26971	0.70315/0.12869
F21	31.2294/46.312	19.3249/20.963	231.3725/104.8622	153.6254/20.0888
F22	150.062/25.1565	195.3489/51.1928	338.8589/146.7542	373.5481/71.197
F23	269.0217/45.3829	269.9081/41.8249	316.7802/145.0557	441.5375/86.7905
F24	57.0541/27.8568	78.8627/20.0993	96.3009/32.4902	93.4406/22.2511
F25	91.3159/23.3488	104.125/9.2509	106.165/2.8333	108.6289/10.6242
F26	34.0242/23.4768	29.4642/16.9507	85.141/36.0697	70.7598/18.9784
F27	294.4923/46.5782	261.3828/49.422	312.1647/41.8169	311.5479/39.8722
F28	68.6314/46.8564	60.254/39.1287	241.1765/125.1822	151.7258/14.6781
Total	18/17	7/4	4/4	0/4

Customs are serviced by 5 vehicles, and the best-known solution result is 784. Symbols like M in the table refer to the mean value of 10 runs. Symbols like S in the table refer to the stand deviation of 10 runs. The last line of table 7 is to calculate the performance of each algorithm in each instance. If an algorithm achieves the best result in an instance, then its corresponding term (Mean/ Stand Deviation) is added to 1. Whether in Mean item or Stand Deviation item IFPA performs better than the other three algorithms. IFPA has 17 best performances in Mean item and 13 in Stand Deviation item. Similarly, in table 8, 22 instances are used to test the algorithms, the parameters are the same as the above mentioned. From table 8, IFPA gets 13 better performance in Mean item and Stand Deviation. And Figure 7 is the route map of IFPA on A-n46-k7 instance in the 9th run. On the whole, the IFPA we proposed has better convergence accuracy and stability than FPA, PSO, and DE in CVRP.

5. **Conclusions.** we propose IFPA in this paper, which is enhanced by random jump perturbation operator and its switching probability varies with the global optimum of

Table 4. Mean/Std result of algorithms with D=10.

No.	IFPA	FPA	PSO	DE
F1	6.36e-09/2.15e-09	0.065679/0.016957	0/0	1.03e+03/2.61e+02
F2	3.01e-03/1.07e-03	534.7903/156.5574	5.10e-11/2.14e-10	6.18e+06/2.09e+06
F3	3.81e+06/1.48e+06	1.27e+07/4.89e+06	1.11e+06/5.29e+06	2.06e+09/5.04e+08
F4	1.19e-05/1.06e-05	36.7133/10.6556	2.36e-13/2.13e-13	9.27e+03/2.38e+03
F5	7.26e-06/1.78e-06	0.28127/0.055384	4.12e-07/7.70e-07	69.5393/12.8463
F6	2.04e-03/8.53e-04	0.09001/0.063173	0.62535/1.4642	83.4214/16.3068
F7	14.0407/2.885	36.8664/5.8414	27.5713/22.4623	67.2262/8.6703
F8	20.2718/0.055033	20.2731/0.064192	20.3044/0.059066	20.2513/0.068758
F9	4.7602/0.61592	5.0435/0.49091	4.7367/1.5897	8.0841/0.53867
F10	0.15883/0.028049	0.15119/0.019017	0.37457/0.27929	140.8073/32.7892
F11	7.8359/1.2888	14.8415/2.3642	29.3414/12.9624	60.1365/7.5711
F12	16.1268/2.8026	23.0094/4.5841	26.259/12.4732	71.1428/8.1376
F13	19.0003/3.7562	30.9777/5.1307	33.3101/9.8706	69.5204/8.8836
F14	331.9764/103.3957	375.9019/70.3603	913.4666/299.0018	1145.9695/133.0936
F15	905.9532/145.697	666.3938/85.8658	838.6646/312.1282	1347.0284/130.3205
F16	0.90926/0.12904	0.70121/0.11803	0.45422/0.37162	0.93323/0.13905
F17	29.802/4.2068	37.5183/4.6169	28.8639/9.0038	114.4804/11.5771
F18	37.5978/4.6184	42.9112/6.0584	27.7559/9.5898	112.9044/12.3419
F19	1.4639/0.23712	1.2385/0.20948	1.2731/0.54824	19.9641/7.7656
F20	3.0182/0.1694	3.2515/0.15586	3.1036/0.51009	3.7233/0.20011
F21	92.4868/26.4427	96.636/26.9239	394.3077/42.0356	482.665/24.4713
F22	403.488/68.4806	570.5885/93.3868	1116.0011/358.705	1468.5114/125.6417
F23	1103.5654/140.2435	929.1687/92.0711	1035.2271/295.0504	1595.5527/170.019
F24	148.2738/8.0375	142.6049/6.8801	206.1124/26.5891	211.4401/15.2614
F25	184.219/24.6832	178.255/20.352	212.9408/11.5838	212.4234/10.3249
F26	123.8798/3.6747	131.3682/4.8194	173.9133/36.4968	187.402/10.9944
F27	429.1122/18.199	407.1113/4.6333	446.6245/33.2932	593.5751/31.5038
F28	100.0807/34.5168	132.4951/6.7595	427.753/198.1379	595.4341/83.8035
Total	11/11	7/11	9/4	1/2

last and now. The random jump perturbation operator improves pollen diversity. The novel switching probability updating strategy controls the proportion of global exploration and local exploitation, that gradually weakens the global effect and enhances the local effect. Then we test the IFPA by CEC2013. IFPA has better convergence precision and speed than FPA, PSO, and DE. Finally, four algorithms are used to solve CVRP. From the test results of the selected CVRP test instances, IFPA is more powerful to solve CVRP than FPA, PSO, and DE.

In the future, the IFPA could be further improved, such as hybrid [50], and adding chaotic mapping [51]. IFPA proposed in this paper could also be applied to other fields, such as power system problems [32] and wireless sensor networks problems [35].

Acknowledgment. This paper is supported by Zhejiang Province Basic Public Welfare Research Program (Grant No. LGG19F020021), Shanghai Automotive Industry Science and Technology Development Foundation (Grant No. 1815).

Table 5. Mean/Std result of algorithms with D=20.

No.	IFPA	FPA	PSO	DE
F1	4.44e-03/8.10e-04	0.96891/0.16029	3.25e-13/1.59e-13	1.14e+04/1.58e+03
F2	293.101/66.0257	8.00e+04/1.74e+04	5.16e+03/8.22e+03	9.30e+07/1.65e+07
F3	2.50e+08/5.51e+07	2.66e+08/6.41e+07	4.81e+08/5.82e+08	4.49e+11/3.86e+11
F4	0.5643/0.40979	2638.4207/616.7311	28.6263/58.552	3.50e+04/5.17e+03
F5	0.052009/0.0081726	2.061/0.26684	6.93e-03/8.55e-03	1182.8672/218.9587
F6	1.0304/0.17338	1.2066/0.32829	2.5239/3.7579	1550.454/301.9796
F7	31.4515/4.0581	60.9592/6.4386	87.7145/37.1132	776.8381/421.3689
F8	20.6868/0.063644	20.7163/0.054111	20.7154/0.075301	20.689/0.075988
F9	16.6607/1.0958	16.9111/0.78845	17.5202/2.2349	22.8569/0.78677
F10	0.46008/0.051919	0.94274/0.071128	0.19415/0.16484	1388.3123/216.9299
F11	69.1726/5.2839	61.4735/6.3978	96.1593/25.4954	264.678/25.0193
F12	78.0355/8.2337	98.3027/12.0344	99.5538/31.8529	285.329/18.044
F13	94.4584/9.2919	123.2666/10.3973	140.74/33.032	280.5053/24.4151
F14	1818.6308/205.7626	1592.7746/114.837	2529.8888/424.7988	3860.7122/182.7896
F15	3273.9183/211.677	2104.1733/157.8468	2274.3553/454.4577	3974.1847/183.4144
F16	1.5065/0.21288	1.4441/0.18399	0.26759/0.48276	1.6406/0.23894
F17	125.6101/10.0439	143.458/13.9367	94.1394/27.1916	541.0395/45.5588
F18	135.002/8.6502	163.4766/17.9539	96.5926/18.0419	538.7545/38.7699
F19	7.5846/0.89521	7.2342/1.0007	5.9695/3.0782	7.31e+03/3.63e+03
F20	7.6657/0.2251	8.2891/0.23037	9.1568/0.85547	9.9697/0.066736
F21	116.9193/2.7018	157.2523/18.2347	305.8824/90.3588	1465.8885/80.36
F22	2431.1419/156.0366	2048.1337/160.2527	3015.8449/623.0757	4519.0503/209.4897
F23	3531.5691/237.1801	2861.2171/209.892	2883.7446/623.5493	4609.5522/213.7189
F24	248.1246/3.0734	253.2957/2.513	249.7151/5.1618	279.4762/2.8596
F25	254.5481/2.7645	264.0839/2.6636	248.2443/5.2724	289.0877/2.9087
F26	2.00e+02/6.10e-04	200.0463/0.011738	211.4108/39.4809	211.4372/2.8607
F27	647.6199/110.5125	405.556/2.6698	776.9549/53.6965	969.5953/28.4846
F28	946.5572/195.4429	1788.742/124.0558	2105.988/530.8711	3249.9454/183.67
Total	14/16	6/9	8/1	0/2

REFERENCES

[1] H. Ceylan, H. Ceylan, S. Haldenbilen,and O. Baskan, Transport energy modeling with meta-heuristic harmony search algorithm, an application to Turkey, *Energy Policy*, vol.36, no.7, pp.2527–2535, 2008.

[2] Z. Beheshti, and S. M. H. Shamsuddin, A review of population-based meta-heuristic algorithms, *International Journal of Advances in Soft Computing and its Applications*, vol.5, no.1, pp.1–35, 2013.

[3] R. Storn,and K. Price, Differential Evolution A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341–359, 1997.

[4] J. S. Pan, N. Liu, and S. C. Chu, A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning, *IEEE Access*, vol.8, pp.17691 –17712, 2020.

[5] Z. Meng,J. S. Pan, and W. Zheng, Differential evolution utilizing a handful top superior individuals with bionic bi-population structure for the enhancement of optimization performance, *Enterprise Information Systems*, vol.14, no. 2, pp.221 –242, 2020.

[6] M. Dorigo,M. Birattari, and T. Stutzle, Ant colony optimization, *IEEE computational intelligence magazine*, vol.1, no. 4, pp.28 –39, 2006.

Table 6. Mean/Std result of algorithms with D=30.

No.	IFPA	FPA	PSO	DE
F1	0.21005/0.047085	1.6724/0.25503	1.99e-07/5.28e-07	3.08e+04/3.25e+03
F2	1.21e+04/3.07e+03	4.30e+05/8.47e+04	2.03e+05/1.21e+05	3.23e+08/5.15e+07
F3	2.55e+09/4.54e+08	1.74e+09/3.65e+08	5.61e+09/3.84e+09	3.06e+11/1.90e+11
F4	12.9991/3.9302	1.04e+04/1.50e+03	1.73e+03/1.52e+03	6.94e+04/8.63e+03
F5	0.66805/0.10691	3.9109/0.51136	10.5765/7.9876	4296.206/732.8512
F6	28.439/5.1056	25.3144/3.9004	31.177/23.6261	3157.9869/489.7223
F7	83.0242/7.7852	126.5768/13.1464	129.9279/38.3035	512.4852/180.6913
F8	20.8958/0.046283	20.8951/0.055199	20.9351/0.043094	20.9168/0.044212
F9	30.4468/0.87152	29.8517/1.0739	31.2823/3.3743	38.8395/0.97836
F10	1.0346/0.010592	1.1713/0.034491	0.2426/0.19423	3937.4196/404.4814
F11	146.4998/10.0488	119.3996/11.975	230.243/70.2358	564.6809/37.7562
F12	155.3292/11.7631	216.5263/23.5823	203.2822/58.1465	598.8115/34.9327
F13	183.596/12.183	251.7995/19.8788	293.5716/62.6283	588.3639/43.931
F14	4438.1955/240.4685	3105.9184/198.4526	4270.3727/730.5661	6824.879/263.9466
F15	6095.9478/238.5257	4342.9191/244.6017	4228.5025/710.9887	7212.0084/289.73
F16	2.0469/0.24339	2.1131/0.203	0.088048/0.36601	2.2689/0.19327
F17	227.0424/13.001	268.1081/18.3024	215.1615/51.2164	1229.9005/71.2603
F18	234.7922/13.4408	311.0879/23.3268	210.304/46.0948	1218.2389/86.2569
F19	15.0655/1.3518	15.7596/1.5256	22.5986/10.6053	1.23e+05/5.45e+04
F20	12.5443/0.22803	12.9485/0.18218	14.3407/0.99597	14.8701/0.12199
F21	343.4379/14.6055	255.4671/7.0003	326.5464/86.3824	3078.0647/151.8024
F22	5322.834/283.3256	3772.413/215.3321	4925.924/722.8289	7552.3091/229.4233
F23	6248.6021/376.254	5170.8379/260.857	4909.9687/925.0429	7894.2917/246.663
F24	283.7803/3.5149	287.9095/2.85	286.6392/6.6789	332.527/4.2635
F25	295.4344/3.7622	310.0104/2.9473	287.8172/5.451	351.0006/4.3878
F26	200.1414/0.094272	200.263/0.068347	267.9438/89.8859	238.554/7.3077
F27	1065.1655/118.6661	456.1333/123.1239	1157.0917/72.5686	1403.9799/32.3949
F28	377.3541/38.5245	1157.1869/73.9425	1763.5459/945.2889	4267.33/316.0937
Total	11/14	9/8	8/2	0/4

[7] H. Al-Qaheri,A. Mustafi, and S. Banerjee, Digital watermarking using ant colony optimization in fractional Fourier domain, *Journal of Information Hiding and Multimedia Signal Processing*, vol.1, no. 3, pp.179 –189, 2010.

[8] S. C. Chu, J. F. Roddick,C. J. Su, and J. S. Pan, Constrained Ant Colony Optimization for Data Clustering, *Pacific Rim International Conference on Artificial Intelligence*, Auckland, New Zealand, pp.210–214, 2009.

[9] X. S. Yang, and S. Deb, Cuckoo search via Lvy flights, *2009 World congress on nature and biologically inspired computing (NaBIC)*, Coimbatore, India, pp.210–214, 2009.

[10] P. C. Song, J. S. Pan,and S. C. Chu, A parallel compact cuckoo search algorithm for three-dimensional path planning, *Applied Soft Computing*, vol.94, 2020.

[11] J. S. Pan, P. C. Song, S. C. Chu and Y. J. Peng, Improved Compact Cuckoo Search Algorithm Applied to Location of Drone Logistics Hub, *Mathematics*, vol.8, no. 3, 2020.

[12] J. Kennedy, and R. C. Eberhart, Particle Swarm Optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Western Australia, pp.1942–1948, 1995.

[13] K. Wang, Y. Q. Liu, D. Liu and Q. Shi, Improved particle swarm optimization algorithm based on gaussian-grid search method, *J. Inf. Hiding Multimed. Signal Process*, vol.9, no. 4, pp.1031 –1037, 2018.

Table 7. CVRP result of algorithms in Instance A.

instance	BKV	M/IFPA	M/FPA	M/PSO	M/DE	S/IFPA	S/FPA	S/PSO	S/DE
A-n45-k6	944	1030.5	1027.9	1033.5	1062.9	12.1339	24.0434	37.2305	22.2989
A-n45-k7	1146	1168.3	1171.7	1174.7	1186.4	4.83	2.7976	12.2776	5.2642
A-n46-k7	914	935.281	942.0607	943.4805	963.2721	4.371	3.0185	13.3023	5.1939
A-n48-k7	1073	1112.2	1113.7	1126.3	1133.3	7.2015	4.2584	10.983	9.2926
A-n53-k7	1010	1062.9	1059.2	1067.1	1089.9	8.949	12.742	19.2893	12.0678
A-n54-k7	1167	1215.7	1219.3	1229.2	1244.6	8.0626	11.8938	20.2852	7.1321
A-n55-k9	1073	1101.3	1104.8	1118.1	1131.1	7.7377	8.3472	15.5771	5.2664
A-n61-k9	1035	1187.8	1171.2	1165.2	1212.1	16.0567	14.5459	32.3751	8.7822
A-n62-k8	1290	1343.3	1341.3	1352.9	1373.7	4.4773	8.4849	17.6934	8.7893
A-n63-k9	1634	1695.5	1706	1706.1	1750.4	14.1201	10.6184	19.477	16.0561
A-n63-k10	1315	1379.9	1377.7	1375.6	1410.7	8.6684	10.8543	17.5371	11.229
A-n64-k9	1402	1489.9	1483.9	1477.6	1513.5	13.2491	14.427	16.9814	11.7299
A-n65-k9	1177	1266.9	1270.6	1280	1315.6	18.594	15.3545	13.8924	20.7444
A-n69-k9	1168	1224.4	1227.8	1235	1262.8	6.5358	8.8949	17.8868	11.1658
A-n80-k10	1764	1891.1	1881.9	1894.9	1924.8	13.5098	9.6443	23.066	17.89
A-n32-k5	784	787.2555	787.917	795.8927	792.3797	0.3843	0.7428	14.1107	3.0042
A-n33-k5	661	670.0689	670.9565	685.5917	674.4119	4.8057	5.1674	8.1119	6.2998
A-n33-k6	742	742.8916	743.1867	744.9353	746.0909	0.3603	0.6962	1.4121	1.9903
A-n34-k5	778	789.835	790.6263	794.8318	796.4842	2.0684	2.1155	3.6343	4.2444
A-n36-k5	799	811.0082	814.3318	822.3429	826.3277	1.7347	3.4134	13.8168	3.4316
A-n37-k5	669	682.6576	688.265	692.2565	698.9449	4.7332	3.3061	11.5006	6.6086
A-n37-k6	949	965.9209	963.2131	972.7341	975.0433	3.795	4.4946	14.4937	7.4291
A-n38-k5	730	746.009	749.1232	756.0405	763.9066	4.7791	4.1212	11.4014	8.4083
A-n39-k5	822	842.3224	849.0114	850.5909	864.5575	6.5871	7.0213	9.9891	9.974
A-n39-k6	831	841.0361	843.8896	844.0845	850.2126	3.0245	3.909	5.6321	6.6134
A-n44-k7	937	972.0752	970.1763	981.2069	993.4208	6.011	9.741	18.607	11.442
Best Number		17	6	3	0	14	7	1	4

Table 8. CVRP result of algorithms in Instance B.

instance	BKV	M/IFPA	M/FPA	M/PSO	M/DE	S/IFPA	S/FPA	S/PSO	S/DE
B-n45-k5	751	764.8276	763.642	771.8425	778.9442	3.238	3.0577	7.6194	5.0931
B-n45-k6	678	726.9834	733.6402	740.6682	750.3614	11.0083	12.2689	19.1044	9.3091
B-n50-k7	741	749.5179	747.5284	752.6085	756.7734	2.0153	1.8771	6.0432	4.5204
B-n50-k8	1313	1342.4	1339.8	1342.9	1348	2.4195	2.9506	6.3554	5.4062
B-n51-k7	1032	1056.6	1062.7	1070.7	1080.1	5.2001	5.7941	10.3877	9.6801
B-n52-k7	747	761.5051	760.8908	768.4612	770.0915	2.8904	1.5867	6.2445	3.7758
B-n56-k7	707	737.8803	740.6999	745.1967	759.6426	3.3764	3.152	8.0153	5.6094
B-n57-k7	1153	1262.9	1283.3	1336.6	1304	15.8288	23.6088	50.9952	15.4473
B-n57-k9	1598	1631.8	1630.9	1628.1	1652.9	4.6436	6.7512	10.9261	5.2717
B-n63-k10	1537	1575.1	1573.9	1581.4	1605.2	6.1593	8.963	6.2592	8.5888
B-n64-k9	861	933.3173	942.5308	932.8179	962.9793	12.2904	6.0937	18.0562	12.8969
B-n67-k10	1033	1100.8	1109.1	1114.9	1131.7	8.5255	7.8119	15.358	7.1263
B-n68-k9	1304	1319.8	1321.2	1325.6	1338.5	3.1125	4.6249	7.3213	3.3454
B-n78-k10	1266	1294.1	1291.4	1293.6	1322.1	7.0016	7.8446	13.9793	7.7938
B-n31-k5	672	676.9321	677.2138	682.1923	682.1957	1.2436	1.3565	5.6781	2.0508
B-n34-k5	788	791.0146	791.5426	793.8962	796.0825	0.748	1.4244	4.0826	2.2652
B-n35-k5	955	960.5567	960.6174	961.1713	965.1459	1.3105	1.6925	2.3593	1.6148
B-n38-k6	805	814.0256	814.868	817.02	824.7458	1.2714	3.841	3.8877	2.6668
B-n39-k5	549	554.8778	556.5571	561.5804	562.0877	1.6345	1.7212	3.2397	3.1788
B-n41-k6	829	839.9814	840.1659	847.5659	850.2074	3.2247	2.3487	5.4337	3.6756
B-n43-k6	742	754.3639	757.4055	758.2226	763.7189	1.986	2.6673	6.0856	3.835
B-n44-k7	909	931.5067	931.0228	939.7941	941.664	2.8862	3.4877	4.9014	4.9147
Best Number		13	7	2	0	13	6	0	3

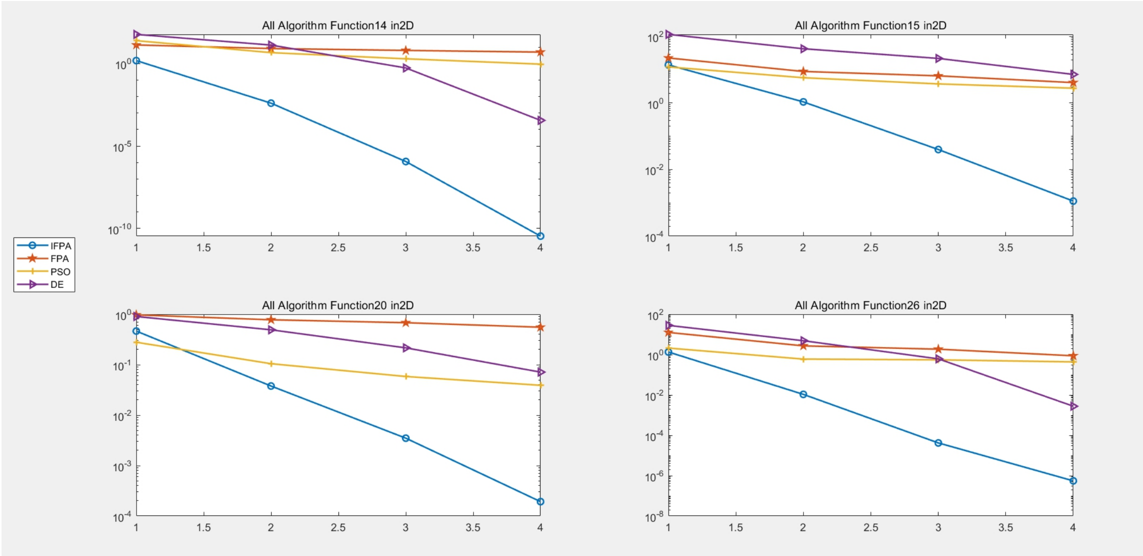

FIGURE 2. The optimization curves of the functions at D=2.

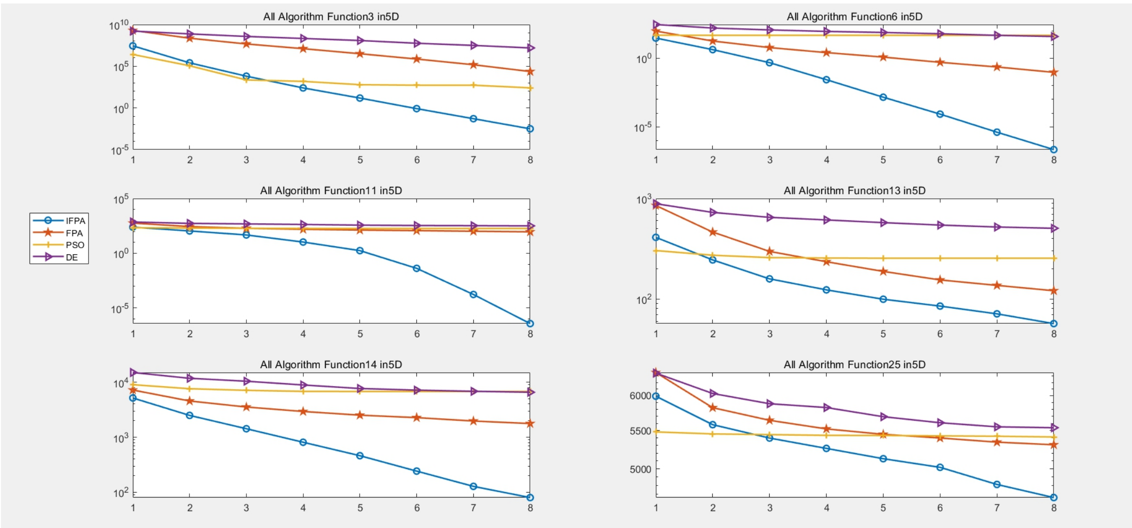

FIGURE 3. The optimization curves of the functions at D=5.

[14] M. Zhao, J. S. Pan, and S. T. Chen, Entropy-Based Audio Watermarking via the Point of View on the Compact Particle Swarm Optimization, *Journal of Internet Technology*, vol.16, no. 3, pp.485 –493, 2015.
[15] D. Karaboga, and B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, *International fuzzy systems association world congress*, Cancun, Mexico, pp.789–798, 2007.
[16] P. W. Tsai, M. K. Khan, J. S. Pan, and B. Y. Liao, Interactive artificial bee colony supported passive continuous authentication system, *IEEE Systems Journal*, vol.8, no. 2, pp.395 –405, 2012.
[17] T. S. Pan, T. K. Dao, J. S. Pan,and T. T. Nguyen, An unmanned aerial vehicle optimal route planning based on compact artificial bee colony, in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, J. S. Pan, P. W. Tsai, and H. C. Huang (eds.), Cham, Springer, pp.361 –369, 2017.
[18] L. Kong, J. S. Pan, P. W. Tsai, Sn. Vaclav, and J. H. Ho, A Balanced Power Consumption Algorithm Based on Enhanced Parallel Cat Swarm Optimization for Wireless Sensor Network, *International Journal of Distributed Sensor Networks*, vol.11, pp.729680:1 –729680:10, 2015.

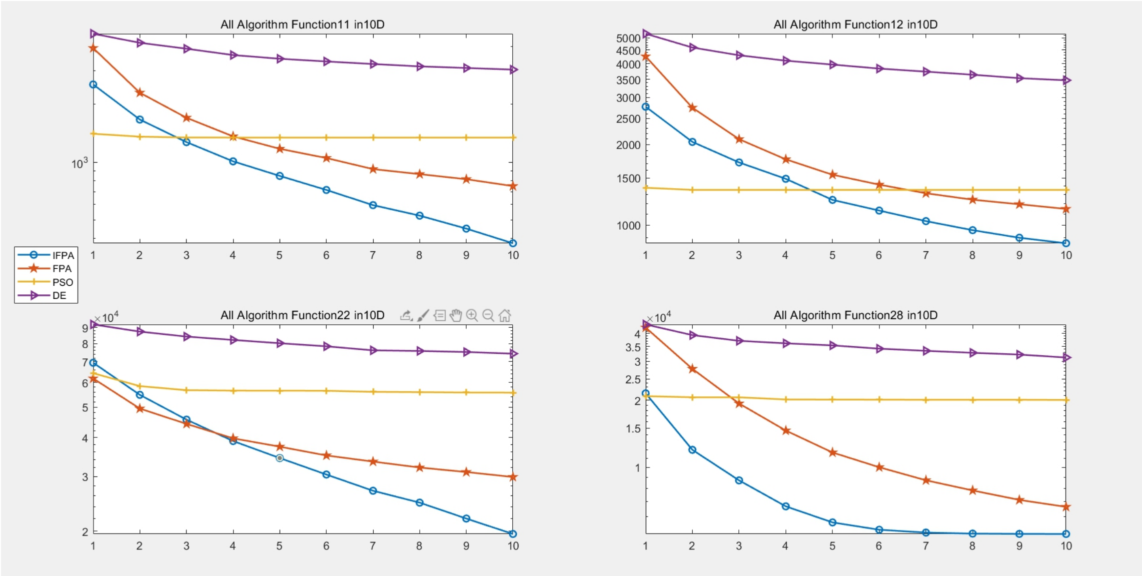

FIGURE 4. The optimization curves of the functions at D=10.

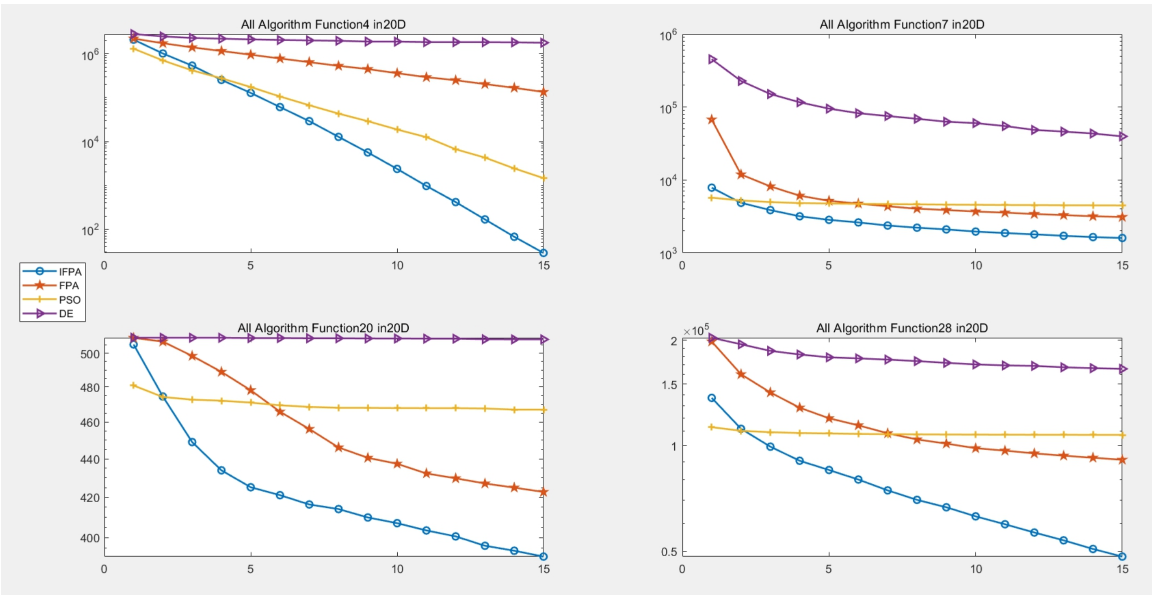

FIGURE 5. The optimization curves of the functions at D=20.

[19] P. W. Tsai, J. S. Pan,S. M. Chen, and B. Y. Liao, Enhance Parallel Cat Swarm Optimization Method Based on the Taguchi Method, *Expert Systems with Applications*, vol.39, no.7, pp.6309 –6319, 2012.
[20] S. C. Chu, P. W. Tsai, and J. S. Pan, Cat Swarm Optimization, *Pacific Rim International Conference on Artificial Intelligence*, Guilin, China, pp.854–858, 2006.
[21] T. T. Nguyen, J. S. Pan, and T. K. Dao, A Novel Improved Bat Algorithm Based on Hybrid Parallel and Compact for Balancing an Energy Consumption Problem, *Information*, vol.10, no.6, 2019.
[22] T. T. Nguyen, J. S. Pan, and T. K. Dao, A Compact Bat Algorithm for Unequal Clustering in Wireless Sensor Networks, *Applied Sciences*, vol.9, no.10, 2019.
[23] S. J. Cai and P. W. Tsai, Echolocation Guided Evolved Bat Algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, vol.7, no.1, pp.153–162, 2016.
[24] S. Mirjalili, S. M. Mirjalili, and A. Lewis, Grey wolf optimizer, *Advances in engineering software*, vol.69, pp.46–61, 2014.

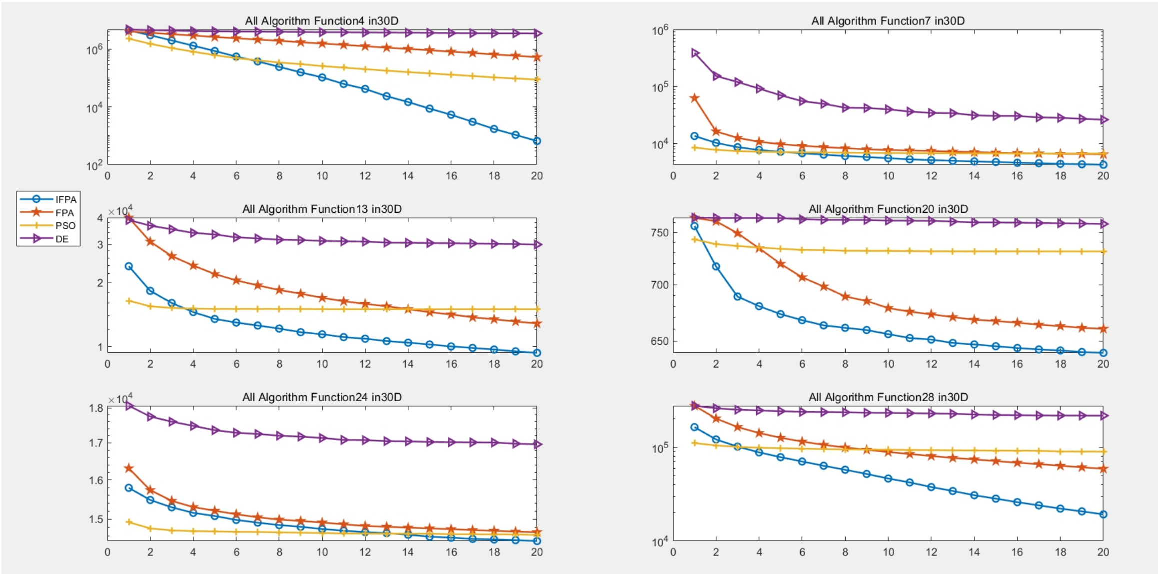

FIGURE 6. The optimization curves of the functions at D=30.

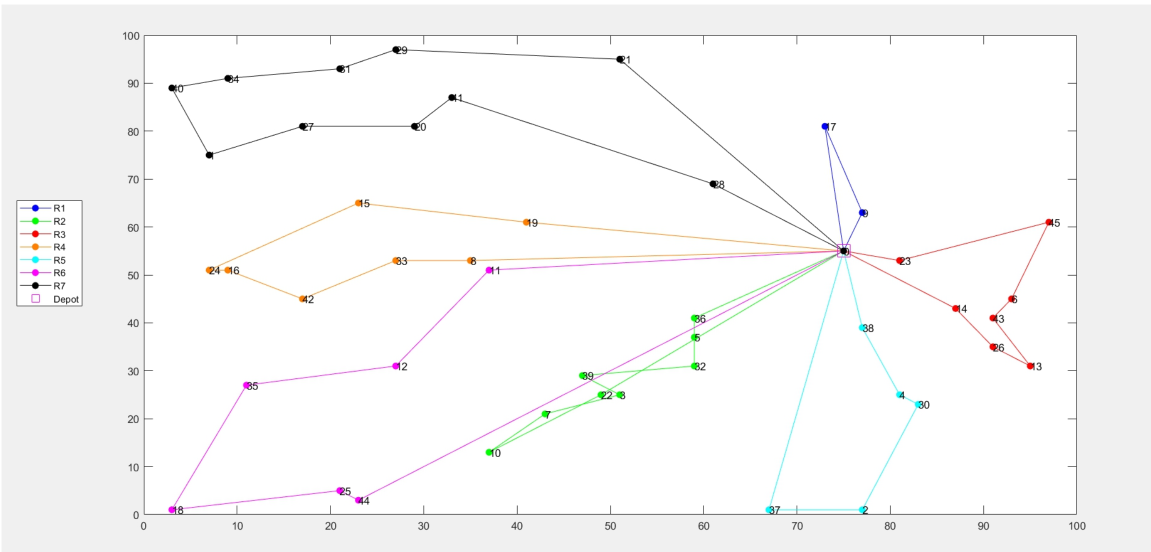

FIGURE 7. The vehicle route of the A-n46-k7 by IFPA.

[25] P. Hu, J. S. Pan, and S. C. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, *Knowledge-Based Systems*, vol.195, 2020.

[26] J. S. Pan, P. Hu, and S. C. Chu, Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power, *Processes*, vol.7, no.11, 2019.

[27] Z. Meng, and J. S. Pan, QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): an enhanced structure for differential evolution, *Knowledge-Based Systems*, vol.155, pp.35–53, 2018.

[28] Z. Meng, J. S. Pan and H. Xu, QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization, *Knowledge-Based Systems*, vol.109, pp.104–121, 2016.

[29] Z. G. Du, J. S. Pan, S. C. Chu, H. J. Luo, and P. Hu, Quasi-Affine Transformation Evolutionary Algorithm With Communication Schemes for Application of RSSI in Wireless Sensor Networks, *IEEE Access*, vol.8, pp.8583–8594, 2020.

[30] N. Liu, J. S. Pan, and T. T. Nguyen, A bi-population QUasi-Affine TRansformation Evolution algorithm for global optimization and its application to dynamic deployment in wireless sensor networks, *EURASIP Journal on Wireless Communications and Networking*, vol.2019, 2019.

[31] X. S. Yang, Flower Pollination Algorithm for Global Optimization, *International conference on unconventional computing and natural computation*, Orlans, France, pp.240–249, 2012.

[32] H. M. Dubey, M. Pandit, and B. K. Panigrahi, A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems, *Cognitive Computation*, vol.7, no.5, pp.594–608, 2015.

[33] O. Abdel-Raouf, M. Abdel-Baset, and I. El-Henawy, An improved flower pollination algorithm with chaos, *International Journal of Education and Management Engineering*, vol.4, no.2, pp.1–8, 2014.

[34] R. Wang, and Y. Zhou, Flower pollination algorithm with dimension by dimension improvement, *Mathematical Problems in Engineering*, vol.2014, pp.1–9, 2014.

[35] T. T. Nguyen, J. S. Pan, and T. K. Dao, An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network, *IEEE Access*, vol. 7, pp. 75985–75998, 2019.

[36] G. B. Dantzig, and J. H. Ramser, The truck dispatching problem, *Management science*, vol.6, no.1, pp.80–91, 1959.

[37] P. Toth, and D. Vigo (eds.), *Vehicle routing: problems, methods, and applications*, Society for Industrial and Applied Mathematics, Philadelphia, America, 2014.

[38] A. Chen, G. Yang, and Z. Wu, Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem, *Journal of Zhejiang University-Science A*, vol.7, no.4, pp.607–614, 2006.

[39] W. Y. Szeto, Y. Wu, and S. C. Ho, An artificial bee colony algorithm for the capacitated vehicle routing problem, *European Journal of Operational Research*, vol.215, no.1, pp.126–135, 2011.

[40] S. Mazzeo, and I. Loiseau, An ant colony algorithm for the capacitated vehicle routing, *Electronic Notes in Discrete Mathematics*, vol.18, pp.181–186, 2004.

[41] E. Teymourian, V. Kayvanfar, G. H. M. Komaki, and M. Zandieh, Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem, *Information Sciences*, vol.334335, pp.354–378, 2016.

[42] G. Laporte, Y. Nobert, A branch and bound algorithm for the capacitated vehicle routing problem, *Operations-Research-Spektrum*, vol.5, no.2, pp.77–85, 1983.

[43] M. Fischetti, P. Toth, D.Vigo, A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs, *Operations Research*, vol.42, no.5, pp.846–859, 1994.

[44] J. Lysgaard, A. N. Letchford, R. W. Eglese, A new branch-and-cut algorithm for the capacitated vehicle routing problem, *Mathematical Programming*, vol.100, no.2, pp.423–445, 2004.

[45] N. R. Achuthan, L. Caccetta, S. P. Hill, An improved branch-and-cut algorithm for the capacitated vehicle routing problem, *Transportation Science*, vol.37, no.2, pp.153–169, 2003.

[46] B. Wu, W. Wang, Y. Zhao, X. Xu, and F. Yang, A novel real number encoding method of particle swarm optimization for vehicle routing problem, *2006 6th World Congress on Intelligent Control and Automation*, Dalian, China, pp.3271–3275, 2006.

[47] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernndez-Daz, *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimizatio*, Technical Report, 2013.

[48] T. J. Ai, and V. Kachitvichyanukul, Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem, *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 380–387, 2009.

[49] K. Fleetwood, An introduction to differential evolution, *Proceedings of Mathematics and Statistics of Complex Systems (MASCOS) One Day Symposium*,Brisbane, Australia, pp.785–791, 2004.

[50] O. Abdel-Raouf, I. El-henawy, and M. Abdel-Baset, A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles, *International Journal of Modern Education and Computer Science*, vol. 6, no. 3, pp. 38–44, 2014.

[51] H. T. T. Binh, N. T. Hanh, L. V. Quan and N. Dey, Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks, *Neural computing and applications*, vol. 30, no. 7, pp. 2305–2317, 2018.