Hindawi
Complexity
Volume 2021, Article ID 8571524, 14 pages
https://doi.org/10.1155/2021/8571524

Research Article

Knowledge and Behavior-Driven Fruit Fly Optimization Algorithm for Field Service Scheduling Problem with Customer Satisfaction

Bin Wu, Hui-Jun Jiang, Chao Wang, and Min Dong

Department of Industrial Engineering, Nanjing Tech University, Nanjing 211816, China

Correspondence should be addressed to Bin Wu; wubinnjtech@163.com

Received 25 July 2020; Revised 7 January 2021; Accepted 22 January 2021; Published 3 February 2021

Academic Editor: Zhile Yang

The field service scheduling problem (FSSP) is the key problem in field services. Field service pays particular attention to customer experience, that is, customer satisfaction. Customer satisfaction described by customer behavior characteristics based on the prospect theory is considered as the primary optimization goal in this paper. The knowledge of the insertion feasibility on the solution is analysed based on the skill constraint and time window. According to the knowledge, an initialization method based on the nearest heuristic algorithm is constructed. Based on the prior knowledge of the FSSP and the endowment of the Fruit Fly Optimization Algorithm, two operators are defined according to the matrix encoding method. Based on these two operators, three search strategies are then proposed, and the smell-based search strategy and vision-based search strategy for the FOA are redesigned. To verify the performance of the algorithms, the proposed operators and strategies are tested and analysed in the well-known benchmark. Through comparison with the state-of-the-art algorithms, the results show that the proposed HFOA is an effective and efficient method to solve the FSSP with customer satisfaction.

1. Introduction

Field services usually refer to the series of service activities performed to meet customer needs at the customer's designated location. Field services are widely provided in various industries, such as the maintenance and protection of electric energy networks and mobile communication systems, after-sales installation and maintenance of home appliances, furniture repair, home care, and the maintenance, repair, and operation (MRO) services of manufacturing enterprises. For example, the MRO services of the Chinese industrial market are more than 500 billion dollars in 2016. With the development of big data, mobile Internet, and other information technologies, innovative business models continue to emerge, such as the online to offline (O_2O) business model. Based on the rapidly growing and customized service needs of customers, the role of field service is becoming increasingly important. The efficiency and quality of field services are closely related to the rational allocation of workers and reasonable planning routes. If worker allocation or route planning is unreasonable, it often results in long travel time for workers, long waiting time for customers, staff skills not matching tasks, and other issues. As a result, worker satisfaction is reduced, which negatively affects worker attitude, service quality, and service efficiency, thereby reducing customer satisfaction and ultimately leading to corporate profit decline. Therefore, the field service scheduling problem (FSSP) is critical to field service.

Field service as a form of service extremely emphasizes customer satisfaction. However, little has been done to investigate the customer satisfaction in the current study about FSSP. Almost all models emphasize the cost, such as the shortest travel time for workers, the most tasks to complete. In fact, customer satisfaction is the first. All the companies try their best to improve the customer satisfaction even if the cost increases. Thus, customer satisfaction as the first optimization goal is presented in our study.

On the other hand, the algorithm for solving field service scheduling is not rich. Newer and advanced solution methods are highly desirable. Therefore, the Fruit Fly

Optimization Algorithm (FOA) is proposed to solve the problem. The FOA is a novel global optimization method based on swarm intelligence, which is proposed by Wenchao Pan [1]. As a new global optimization method, FOA has the advantages of fast convergence, simple operation, and convenient application, and so on. Therefore, FOA has given rise to many researchers' attention and has successfully been used in some areas in recent years, such as tuning of PID controller [2], neural network [3], machine scheduling [4], multidimensional knapsack problems [5], and joint replenishment problems [6]. However, the FOA has not been applied to FSSP in addition to our work [7].

This paper contributes to this research area in three ways: firstly, the customer satisfaction is formulated into the FSSP model as an optimization goal, and it is described according to the behavioral features of the customers based on prospect theory; secondly, the knowledge of the insertion feasibility on the solution is analysed. And the improved nearest neighbour insertion algorithm for initialization based on the knowledge is presented. Thirdly, the FOA is presented for FSSP. The matrix-based representation and three matrix-based search strategies are designed according to the problem characteristics.

The rest of the paper is organized as follows. Section 2 reviews the field service scheduling problem. Section 3 illustrates customer satisfaction based on the prospect theory. In Section 4, the mixed-integer programming model of FSSP is presented. After that, the hybrid FOA for FSSP is discussed in detail in Section 5. Thereafter, the simulation experiment and comparisons are provided in Section 6. Finally, the conclusions and future work are discussed in Section 7.

2. Literature Review

Lesaint et al. [8] firstly put forward the concept of field service scheduling. They developed a heuristic algorithm to schedule 20,000 engineers for British Telecom maintenance and repair. However, the details of the algorithm and the experimental results have not been published. Considering the types and priorities of tasks, worker skills, customer time windows, and other constraints, the model of FSSP was established and a greedy randomized adaptive search procedure (GRASP) was proposed for solving the problem by Xu and Chiu [9]. A construction heuristic and an adaptive large neighbourhood search algorithm for FSSP were presented by Cordeau et al. [10]. Akjiratikarl et al. [11] presented the shortest travel distance of workers as the optimization target and proposed the particle swarm optimization algorithm to optimize the solution. The time delay and overtime of workers were also considered in the optimization goal by Petrakis et al. [12], and the heuristic algorithm with insertion and neighbourhood was proposed to optimize the model.

Goel and Meisel [13] investigated FSSP about the maintenance of electricity networks. To solve the problem, they combined the large neighbourhood search algorithm with the mathematical programming method. Xu et al. [14] investigated the nondominated sorting genetic algorithm-II (NSGA-II) to the cross-trained workers scheduling problem. The FSSP model considering both the constraint of worker's skills and routing optimization was presented by Jiang and Wang [15]. The genetic algorithm based on the segment of the chromosome encoding method was investigated to solve the model. Kovacs et al. [16] solved the FSSP considering the team building by using adaptive large neighbourhood search.

Damm et al. [17] proposed a customized biased random key genetic algorithm to solve the multiobjective FSSP. The main objectives include the sum of priority values associated with the tasks and the idle time of the employees. Cortés et al. [18] developed the branch-and-price algorithm to solve a real FSSP. Hashimoto et al. [19] studied the FSSP model for France Telecom, and the task priority and skill level were considered in the model. The improved GRASP was investigated to solve the model. A parallel adaptive large neighbourhood search with the postoptimization procedure based on the mathematical programming was proposed to solve FSSP by Pillac et al. [20]. Decerle et al. [21] studied the memetic algorithm for the home health care routing and scheduling problem with time window and synchronization constraints.

A few papers also considered the multiperiod scheduling. Chen et al. [22] considered the service time affected by the technician's experience and learning rate. The rolling-horizon framework was built, and the variant of the record-to-record travel algorithm (RTR) was presented to solve the daily routes. Furthermore, Chen et al. [23] also considered the dynamic requests in daily decision making and proposed an approximate dynamic programming-based solution approach for the problem. Qin et al. [24] proposed several local search operators and integrated them into a tabu search framework to solve the multiperiod FSSP. Considering the team building, Zamorano and Stolletz [25] proposed a branch-and-price algorithm for the multiperiod FSSP.

Additional papers consider the dynamic and stochastic service requests. Souyris et al. [26] focused on uncertainty associated with service times, and the branch-and-price method was investigated for the problem. The dynamic workforce scheduling problem about British Telecom (BT) was researched by Borenstein et al. [27]. The problem was decomposed into subproblems, and a rule-based system was proposed for the problem. In addition, Tang et al. [28] and Cao and Han [29] studied the field service scheduling problem based on the M/G/m queuing model by simulation.

The foregoing short review shows that the FSSP has attracted increased attention. However, the study of the FSSP is still in its infancy. Since the important applications of the problem, closer to reality models and more advanced methods for solving the problem are highly desirable. To the best of our knowledge, it seems that no research considers the customer satisfaction in the optimization goal for FSSP. And FOA is also the first reported application of FSSP in our study.

The FOA has been successfully applied to various areas in recent years. To name a few, a two-stage adaptive FOA was proposed by Wang et al. [6] to solve the unrelated parallel machine scheduling problem with additional resource constraints. Hu et al. [30] used the FOA to optimize the parameter for the generalized regression neural network. Meng and Pan [31] investigated an improved FOA with the harmony search algorithm for solving the multidimensional knapsack problem. Shen et al. [32] used the FOA to tune support vector machine (SVM) parameter, and it was applied to medical diagnosis. Zheng et al. [33] presented the improved FOA to solve the semiconductor final testing scheduling problem. To keep the diversity of the population and avoid the algorithm falling into local optimum, Niu et al. [34] and Li et al. [35] proposed the multipopulation concept. Niu et al. [34] divided the population into two subpopulations: one was responsible for the exploration and the other was responsible for the exploitation. Li et al. [35] proposed the four subpopulations in the whole algorithm, and each subpopulation evolved separately.

A knowledge-guided multiobjective FOA was investigated for the multiskill resource-constrained project scheduling problem by Wang and Zheng [36]. To improve the local and global searching ability and avoid premature convergence, Han et al. [37] proposed the improvement of FOA with trend search and coevolution. Wu et al. [38] added adaptive strategies and multigroup mechanisms to the FOA. The strategies included the adaptive adjustment mechanism for the search direction, the iteration step, and the crossover and mutation mechanism. In addition, some FOAs with other intelligent optimization algorithms are proposed. Yuan et al. [39] used the chaotic sequence to enhance the global searching capability of FOA. Lei et al. [40] applied the FOA combining gene expression profiles for identifying the protein network structure. Peng et al. [41] used the FOA to the flow shop rescheduling problem. The FOA with the local search is used to improve the performance of a multistart variable neighbourhood descent algorithm. However, to the best of our knowledge, the FOA has not yet been applied to the FSSP.

3. Customer Satisfaction Description Based on the Prospect Theory

The FSSP can be described as follows: a few n spatially distributed tasks must be served exactly once by one of m workers. Workers have different skills and different proficiency levels for every skill. They start their service from their home, visit locations to fulfill tasks, and return home. The assignment of a task to a worker is only possible if the worker has a certain skill required by the task. Each task has service time and time windows. Their services are forbidden outside the deadline. The optimization goals of the FSSP are usually to minimize the total travel time of the workers, the total service time, and so on. Unlike the previous studies, customer satisfaction is the primary goal of optimization in this paper. Therefore, how to describe the customer satisfaction is the key to solve the problem.

Field service scheduling is like a vehicle routing problem [10]. However, the two problems differ in the following aspects. First, the objective in the vehicle routing problem is usually to minimize the number of vehicles required or the total distance traveled, whereas FTSP tries to maximize the number of jobs completed within a given time frame. Second, the vehicles in VRP have physical capacities while the technicians in FSSP have different skills as required by different jobs. The customer satisfaction is represented based on time windows in the traditional vehicle routing problem [42], as shown in Figure 1 ($[E_i, L_i]$ represents the time window that the customer can tolerate and U represents the customer satisfaction). From the customer satisfaction point of view, if the customer is serviced in the time window, the satisfaction is 100%; otherwise, the satisfaction is 0. The method has a large gap from reality. The other researchers [43, 44] described customer satisfaction as a triangular fuzzy number or trapezoidal fuzzy number based on the time window, as shown in Figures 2 and 3. But the psychological characteristics of customers are not considered in these methods. For example, when customers are waiting for service, they tend to be inner anxiety and dissatisfaction (i.e., loss sensitive) as the waiting time increases nonlinearly. The prospect theory is a kind of behavioral decision theory which has great influence on behavioral science. It is based on the bounded rationality of human beings and can describe the decision-making behavior of people more reliably, such as dependency, loss avoidance, and decrease of sensitivity. Prospect theory has obvious advantages in describing human psychological characteristics such as satisfaction. Therefore, the customer satisfaction is described based on the prospect theory in this paper.

In the prospect theory, the individual needs to select a reference point in the initial stage of the decision, and the value of the reference point is zero. The individual evaluation is carried out relative to the reference point. In the decision-making stage, according to the value function of the income and loss of subjective evaluation (as shown in Figure 4), the specific form is as follows:

$$g(x) = \begin{cases} x^{a}, & x > 0, \\ -\gamma(-x)^{\beta}, & x < 0, \end{cases} \tag{1}$$

where α and ß are the risk attitude coefficients, usually between (0,1); the larger the value, the more inclined to risk. γ is the loss of aversion coefficient, greater than 1 that is more sensitive to the loss of income characteristics [45].

In the field service scheduling, the customer satisfaction is mainly affected by the arrival time of workers. When the workers' arrival time is later than the customer's psychological expectations, it will cause the customer's dissatisfaction; as the delay time increases, the dissatisfaction will increase. Because the impact of customer satisfaction is the gap between the customer's psychological expectations of arrival time and the actual situation, the greater the gap, the higher the customer dissatisfaction, rather than the arrival time itself. For the customer, the worker's arrival delay is a loss, this loss will cause changes in customer psychological perception. Therefore, the customer dissatisfaction can be described by the prospect theory. Customer dissatisfaction with worker arrival time is shown in Figure 5, where R_i represents the arrival time reference point for customer i, a_i represents the actual time to start the service, and T_i is the customer's maximum tolerance start time. The customer's dissatisfaction function is expressed as (2).

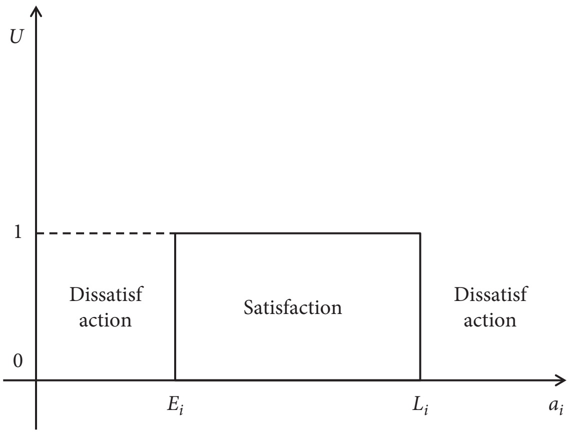

Figure 1: Time window description.

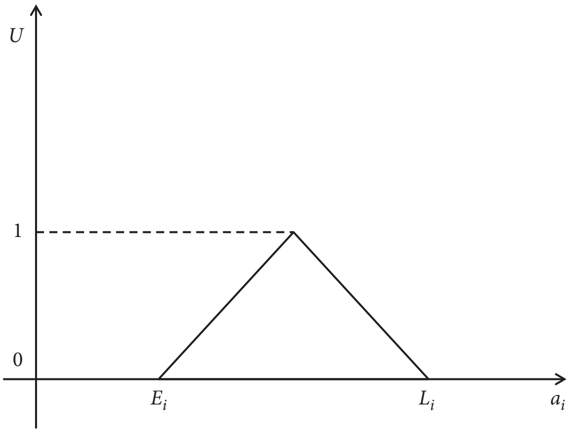

Figure 2: Triangular fuzzy number description.

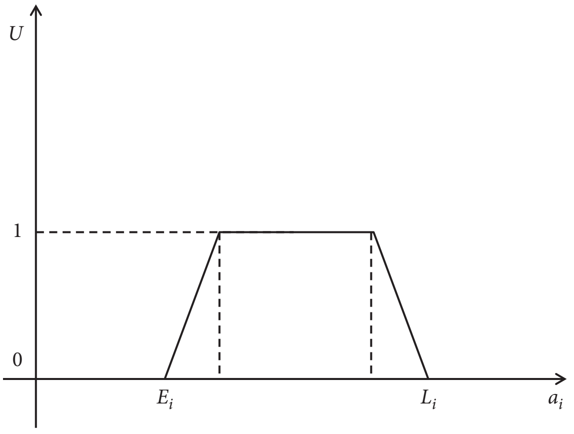

Figure 3: Trapezoidal fuzzy number.

$$U(a_i) = \begin{cases} 1, & a_i > T_i, \\ \gamma\left(\dfrac{a_i}{R_i - 1}\right)^{\beta}, & R_i < a_i \leq T_i, \quad i = 1, 2, \ldots, n. \\ 0, & 0 \leq a_i \leq R_i. \end{cases} \tag{2}$$

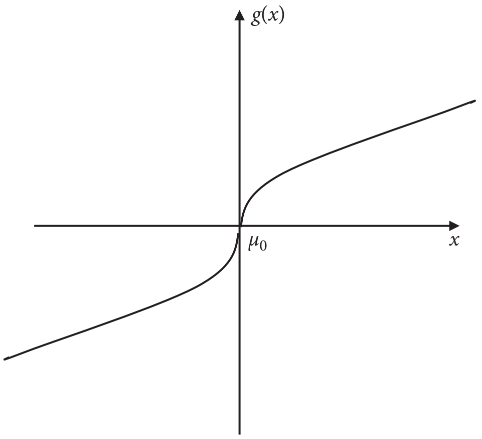

Figure 4: The value function in the prospect theory.

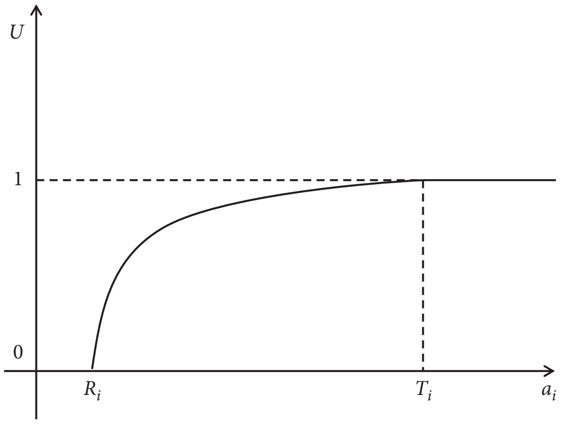

Figure 5: Dissatisfaction function for customer i.

$T_i = R_i[1 + (\gamma^{-1})^{\beta^{-1}}]$ is the maximum allowable start time corresponding to the loss value.

4. Model Formulation

The model assumes that all the information is known in advance in the static environment. Suppose a multigraph $G = \{V, E\}$, where the vertex set is divided into two disjoint subsets $V = \{D, C\}$. Set $D = \{0, 1, 2, \ldots, m-1\}$ represents the locations of the engineers' homes. Set $C = \{0, 1, 2, \ldots, n-1\}$ represents the tasks' locations. $E = \{(i, j)\}, i, j \in V$ is the arc set and $c_{ij} = c_{ji}$ is the travel time from i to j. $[R_i, T_i]$ denotes the expected start time and the endured latest start time for task i. $\lambda_{ki} \in (0, M]$ denotes the proficiency level of worker k for task i. The smaller λ_{ki} denotes the higher the proficiency, which means a shorter time required to complete task i. $\lambda_{ki} > M$ indicates that the worker does not possess a required skill and cannot complete the task. a_i denotes the actual start time of task i. s_i denotes the standard service time of task i. Each worker k has a default working time window $[b_k, e_k]$, and any work performed after e_k is considered as overtime. Let z_k be the remaining time after technician k returns to the home for the day. Q denotes a large enough integer. The binary decision variables $x_{ijk} = 1$, $i, j \in V, k \in D$ if and only if the worker k travels from node i directly to node j. The binary decision variables $y_{ik} = 1$, $i \in C, k \in D$ if and only if task i is serviced by worker k.

$$\max \frac{\sum_{i=0}^{n-1} (1 - U_i)}{n} + \frac{\sum_{k=0}^{m-1} z_k / (e_k - b_k)}{m}, \tag{3}$$

$$\sum_{k\varepsilon\ D} y_{ik} = 1, \quad \forall i \in C, \tag{4}$$

$$\sum_{j\in V} x_{ijk} = \sum_{j\in V} x_{jik} = y_{ik}, \quad \forall i \in C, k \in D, \tag{5}$$

$$\sum_{j\in C} x_{ijk} = 0, \quad \forall i \in D, k \in \frac{D}{\{i\}}, \tag{6}$$

$$a_j = \max\left\{R_j, a_i + \lambda_{ki} s_i + c_{ij}\right\}, \quad \forall i, j \in C, \forall k \in D, \tag{7}$$

$$a_j \le T_j, \quad \forall j \in C, \tag{8}$$

$$a_i + \lambda_{ki} s_i + c_{ij} \le a_j + Q\left(1 - x_{ijk}\right), \quad \forall k \in D, i \ne j \in C, \tag{9}$$

$$a_k + c_{kj} \le a_j + Q\left(1 - x_{kjk}\right), \quad \forall k \in D, j \in C, \tag{10}$$

$$a_j + \lambda_{kj} s_j + c_{jk} \le e_k - z_k + Q\left(1 - x_{jkk}\right), \quad \forall j \in C, k \in D, \tag{11}$$

$$0 \le z_k \le e_k - b_k, \quad \forall k, \tag{12}$$

$$x_{ijk}, y_{ik} \in \{0, 1\}, \quad \forall i, j \in C, k \in D, \tag{13}$$

$$a_i, s_i, z_k \in R_+, \quad \forall i, j \in C, k \in D. \tag{14}$$

The objective function (3) has two parts; the first part represents maximizing the average satisfaction of all customers. U_i is calculated according to equation (2). The second part represents maximizing the average of remaining time ratio of workers. We include in the objective function the remaining time rather than the travel time because the remaining time can be used to allocate more jobs when new jobs are requested, while idle time is wasted just like travel time. Maximizing remaining time is the same as minimizing travel time plus idle time. The purpose of using the denominator is to remove the dimension. Constraint (4) ensures that each task is visited by exactly one worker. Flow conservation constraint (5) ensures that a worker arriving at a task location also leaves that location, and each task is visited exactly once. Constraint (6) is necessary for the assignment of workers to their homes. Equality (7) sets the arriving time at the task i. Constraint (8) ensures that the arriving time does not exceed the endured latest start time for task i. When two tasks are consecutively executed by the same worker, constraint (9) ensures that the beginning of service j starts up after the completion time of its predecessor i plus the travel time between them. Constraint (9) together with (10) and (11) is aimed to eliminate subtours. Constraint (12) guarantees that the idle time of workers that when they return to their home is less than or equal to the size of their time windows. Constraint (13) defines variables x and y as binaries, and constraint (14) defines a_i, s_i, z_k as nonnegative real variables.

5. Hybrid Fruit Fly Optimization Algorithm for FSSP

5.1. Standard FOA. The FOA is to simulate the foraging behavior of fruit fly swarms. Fruit flies have a keen sense of smell; they can detect a source of food at 40 km. Furthermore, they have outstanding vision and can find where other flies are gathering and flying in that direction. There are two main steps in FOA: smell-based search and vision-based search. In the smell-based procedure, a fruit fly search for food randomly around the swarm location. In the visual procedure, a fruit flies fly to the best location of the swarm according to its excellent vision. The two procedures are repeated until meeting the termination criterion. The standard FOA procedures are summarized as follows:

Step 1: initialization. The parameters of the algorithm (i.e., population size and maximum number of iterations) and the initial center location of the swarm are initialized

Step 2: smell-based search procedure

Step 2.1: every fruit fly randomly determines a candidate location for a food source

Step 2.2: evaluate the smell concentration values for each fruit fly

Step 3: vision-based search procedure

Step 3.1: find the fruit fly with the best smell concentration and record the best location

Step 3.2: the fruit fly flies toward the best location and updates the center location of the swarm

Step 4: termination condition. If the termination conditions are met, the algorithm ends; otherwise, return to Step 2

The standard FOA is mainly applied for continuous optimization. To solve the FSSP, the knowledge of the solution is analysed, and then the encoding method, the smell-based search, and the vision-based search are redesigned. The detailed implementations of the hybrid FOA are introduced, respectively, including the nearest insertion heuristic algorithm for initialization; the matrix-based representation and the decoding procedure; and three matrix-based search strategies and postoptimization process.

5.2. Solution Knowledge about Task Insertion Feasibility. Because of the presence of time windows, two factors should be considered when a task is inserted into existing routes: the skill constraint and time window. Here, a feasible insertion property is proposed. For worker f, the existing feasible routes are $i_1, i_2, \ldots, i_p$. If a task h is inserted after $k -$ th position (between i_k and i_{k+1}), then the start service time in h

is $a_h = \sum_{t=1}^{k-1} c_{i_t i_{t+1}} + \sum_{t=1}^{k} \lambda_{f i_t} s_{i_t} + \sum_{t=1}^{k} w_{i_t} + c_{i_k h}$, where $w_{i_t} = \max\left\{R_{i_t} - (a_{i_{t-1}} + \lambda_{f i_{t-1}} s_{i_{t-1}} + c_{i_{t-1} i_t}), 0\right\}$ represents the wait time in the i_t. The delay time in the i_{k+1} is as follows:

$$\begin{aligned}\Delta D_{i_{k+1}} &= a_{i_k} + \lambda_{f i_k} s_{i_k} + c_{i_k h} + \lambda_{fh} s_h + w_h + c_{h i_{k+1}} \\ &\quad -\left(a_{i_k} + \lambda_{f i_k} s_{i_k} + c_{i_k i_{k+1}}\right) \\ &= \lambda_{fh} s_h + w_h + c_{i_k h} + c_{h i_{k+1}} - c_{i_k i_{k+1}}.\end{aligned} \tag{15}$$

When the h inserts into the route, if $\Delta D_{i_{k+1}} > 0$, then i_{k+1} will delay. Due to the "absorption" of the waiting time on the original route, the delay time in i_t after the insertion of h is as follows:

$$\Delta D_{i_t} = \max\left\{\Delta D_{i_{k+1}} - \sum_{t=k+1}^{p-1} w_{i_t}, 0\right\}, \quad t = k+2, \ldots, p. \tag{16}$$

Suppose the original arriving time in i_t is a_{i_t}, then the maximum delay that does not violate the time window constraint in i_t is $T_{t_i} - a_{t_i}$. Thus, if a task h is inserted after k – th position (between i_k and i_{k+1}), it should meet the following properties.

Property 1. For worker f, the existing feasible routes are $i_1, i_2, \ldots, i_p$. If a task h is inserted after k – th position (between i_k and i_{k+1}), the necessary and sufficient conditions for insertion are as follows:

$$\begin{aligned} a_h &\leq T_h, \\ \Delta D_{i_t} + a_{i_t} &\leq T_{i_t}, \quad t = k+1, \ldots, p, \\ \lambda_{fh} &\leq M. \end{aligned} \tag{17}$$

According to Property 1, we can check whether the task can be inserted into the position. However, maybe several positions are feasible in the route for the task which is the optimum position. The optimum insertion position is the one that the average degree of dissatisfaction is minimum after insertion in all possible insertion positions. The determination of the optimal insertion position can be made according to the following steps:

Step 1: find out the set of all possible insertion positions in the current route for the candidate task, and set the customer's dissatisfaction to be 1.

Step 2: calculate the average dissatisfaction of all customers after inserting the task into the leftmost position of the current feasible insertion position set. If the dissatisfaction is smaller than the original value, the average customer dissatisfaction is set to the current value, and the current insertion position is set as the optimal insertion position, and the position is deleted from the feasible set, and go to Step 3.

Step 3: check the feasible insertion position collection. If the collection is not empty, go to Step 2; otherwise, end.

5.3. Heuristic for Initialization Based on Solution Knowledge.

A random initialization method is simple but may produce invalid solutions. To generate an initial solution with good quality, an insertion heuristic initialization method utilizing knowledge of the problem is proposed. The nearest insertion heuristic algorithm (NIH) is a classical local search algorithm. The core idea is described as follows: the nearest task that meets the constraint is selected from the nonserviced task collection and then inserted into a route. Based on the above property, the improved NIH is proposed in this paper. We revise the criteria for selecting candidate tasks, not only considering the distance, but also the service time of the worker. Suppose i is the last task in the current routes, j represents the unserviced task, and N_{ij} represents the generalized costs between i and j. N_{ij} can be defined as follows:

$$N_{ij} = \eta_1 \times c_{ij} + \eta_2 \times \lambda_{kj} \times s_j, \tag{18}$$

where η_1 and η_2 are weight coefficients, where $\eta_1 + \eta_2 = 1$. All unserviced task generalized costs are calculated according to formula (18). Then the smallest N_{ij} is selected, and the task j is inserted into the routes following task i. The procedure for NIH is as follows.

U represents the task set, RS_krepresents the k – th route set, CL_k represents the travel time of the k – th routes, and V_j represents the task set of worker j.

Step 1: $U = C, k = 0, \mathrm{RS}_k = \varnothing, CL_k = 0, j = 0$.

Step 2: check the status of U. If it is null, then go to Step 5. Otherwise, select the minimum costly task i. If $T_i > c_{ki}$, then initialize RS_k, insert task i into RS_k, remove i from U, and update $CL_k = c_{ki} + \lambda_{ki} s_i$.

Step 3: according to Property 1, select an unserviced task from U, then calculate the generalized costs N_{ij} of every task, and search the minimum N_{ij}. Then insert the corresponding task into RS_k, and update CL_k and U. Repeat this procedure until no task is selected.

Step 4: $k = k + 1$, $CL_k = 0$, go back to Step 2.

Step 5: map the initialization solution to the encoding population. Calculate the concentration of the initial solution, and select the maximum fitness as the current optimal solution.

5.4. Representation and Decoding Procedure.

To determine a feasible schedule for the FSSP, it needs to assign a suitable worker for each task and arrange the execution orders of task for each worker. In the HFOA, the solution is represented by an $m \times n$ matrix, where the m rows denote m workers and n columns represent the n tasks.

$$G = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{m-10} & g_{m-11} & g_{m-12} & \cdots & g_{m-1n-1} \end{bmatrix}. \tag{19}$$

The elements in the matrix $g_{ji} \neq 0$ represent that task i is assigned to worker j. Otherwise, $g_{ji} = 0$. Because all tasks are processed only once, $|g_{ji}| = \sqrt{\sum_{j=0}^{m-1} g_{ji}^2}$, meaning there is only one nonzero element in each column. For every row j, the collection of nonzero elements $G^j = (g_{jl}, g_{jm}, \ldots, g_{ji})$ represents the tasks that are assigned to worker j. These elements are arranged from small to large, and the corresponding task order is worker's service route. Using the encoding scheme, the tasks and the routes for workers can be represented simultaneously, and the search operator applied to the representation influences the schedule directly. To map a solution to a feasible schedule, the decoding procedure is outlined as follows:

Step 1: generate the initial allocation plan for every worker.

For $j = 0 \,\text{to}\, m - 1$ {select all the nonzero elements $(g_{jl}, g_{jm}, \ldots, g_{ji})$ of row j. Then sequence $(g_{jl}, g_{jm}, \ldots, g_{ji})$ from small to large, and then the corresponding tasks are $(m, j, \ldots, l)$.}

Step 2: the amendment of the illegal solution. For the routes $(l, k, \ldots, i)$ of worker j, each task is checked whether the arrival time of the workers meets constraint (8) in turn. If constraint (8) is violated, the task is deleted and reinserted into the routes according to Property 1, and the procedure with the determination of the optimal insertion position is applied. If it is unable to insert, the task will be assigned to other workers. There are three principles of assignment: (a) assign to the nearest workers; (b) assign to the workers who have the shortest service time; and (c) assign to the workers who have the least task. Repeat the above procedure until all tasks in the route are feasible. For the routes that are reallocated, the process is also performed.

Step 3: according to the feasible solution, the corresponding coding value is corrected, and the correction process is to exchange the corresponding value.

Example 1. Consider 3 workers and 8 tasks in FSSP, and the encoding is as follows:

$$G = \begin{bmatrix} 0 & 2.87 & 0 & 9.65 & 0 & 0 & 1.94 & 0 \\ 3.12 & 0 & 0 & 0 & 8.75 & 0 & 0 & 0 \\ 0 & 0 & 1.43 & 0 & 0 & 13.84 & 0 & 7.61 \end{bmatrix}. \tag{20}$$

According to the decoding algorithm, for worker 1, the tasks are (7, 2, 4), for worker 2, the tasks are (1, 5), and for worker 3, the tasks are (3, 8, 6). Then check and revise the routes, and the workers' travel routes are as follows: no. 1 route is 2-7-4; no. 2 route is 5-1; and no. 3 route 8-6-3. Finally, the code will be encoded and the resulting code is as follows:

$$G' = \begin{bmatrix} 0 & 1.94 & 0 & 9.65 & 0 & 0 & 2.87 & 0 \\ 8.75 & 0 & 0 & 0 & 3.12 & 0 & 0 & 0 \\ 0 & 0 & 13.84 & 0 & 0 & 7.61 & 0 & 1.43 \end{bmatrix}. \tag{21}$$

5.5. Fruit Fly Search Strategies Based on Prior Knowledge. The classical FOA utilizes real number coding and is mainly applied to function optimization. Therefore, the strategies of classical FOA are not applicable to the FSSP. In this paper, three strategies of fruit fly search are proposed according to the characteristics of the FSSP. First, we define two operators on two matrices.

posSwap(m, k) represents that the m – th row (or column) and k – th row (or column) are exchanged in a matrix A. A represents an individual fruit fly. All nonzero elements in G are calculated as follows:

$$g_{li} = g_{li} + \text{rand} \times \left(g_{li}^{\text{best}} - g_{li}\right), \quad g_{li} \neq 0, l = m \,\text{or}\, k, i > 0, \tag{22}$$

where g_{li}^{best} denotes the corresponding value for the current best individual. From the perspective of social psychology, this operation indicates that fruit flies use population experience information to guide their search. valueSwap(m, k) represents that the element g_{mk} in A is reassigned to a new value according to the following formula:

$$g_{mk} = g_{mk} + rand \times \left(g_{mk}^{\text{nb}} - g_{mk}\right), \tag{23}$$

where g_{mk}^{nb} denotes the corresponding value in the neighbour that is selected randomly within the field of view. From the perspective of social psychology, this operator represents the comparison of two positions of fruit flies on the horizon and determines which position is closer to the food source. It is worth noting that when g_{mk} is updated, the other elements in column k will be set to zero to guarantee a feasible solution.

Based on these two operators, three search strategies are defined:

(1) Worker-swap (W-sp): $m \in D, k \in D$, posSwap(m, k). This strategy indicates that all tasks are exchanged between two workers, and it also causes service order changes.

(2) Task-swap (T-sp): $m \in C, k \in C$, posSwap(m, k). This strategy indicates that two tasks are exchanged between two workers. The service orders may also change simultaneously.

(3) Task-reallocation (T-re): $m \in C, k \in D$, valueSwap(m, k). This strategy indicates that task k is assigned to worker m. It can also change the service order.

The aforementioned strategies have to perform the decoding procedure that is described in Section 5.4 after execution, and if it encounters an illegal solution, it is corrected according to Property 1 and the procedure of determination of the optimal insertion position. Based on the strategies, the smell-based search and vision-based search are implemented as illustrated in Figures 6 and 7. As we know, the smell-based search is primarily responsible for exploration. Its search range is relatively large in the FOA. The vision-based search is primarily responsible for exploitation. It performs fine-grained searching in a relatively small area. According to the above principles, for smell-based search, W-sp, T-sp, and T-re are carried out in turn. We then select the best solution from the three results. Next, if the best new solution is better than the original solution,

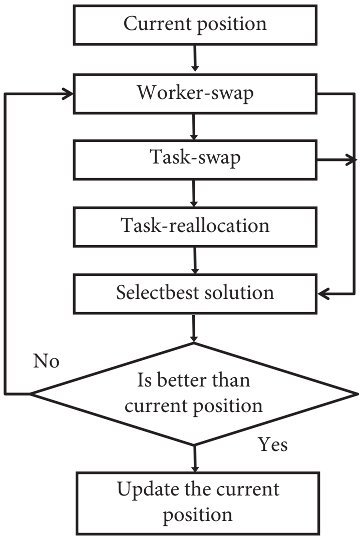

Figure 6: Smell-based search procedure.

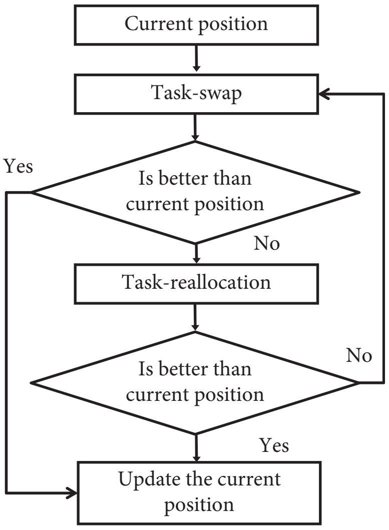
Current position
Task-swap
Yes
Is better than current position
No
Task-reallocation
No
Is better than current position
Yes
Update the current position

Figure 7: Vision-based search procedure.

we update the original solution using the new solution. Otherwise, we repeat the above procedure until the original solution is improved. While the best acceptance principle is utilized in the smell-based search, the better acceptance principle is utilized in the vision-based search, only T-sp and T-re are carried out, and whenever a better solution is found, the previous solution will be updated.

5.6. Postoptimization Procedure. To improve the quality of the solution, two local search algorithms (2-Opt and Or-Opt) are carried out as the postoptimization procedure. The 2-Opt and Or-Opt algorithms will compare every possible valid combination of the swapping to find the better solution. Note that the nodes should be checked according to Property 1 before swapping. The postoptimization process utilizes a better acceptance strategy to update the solution. Based on the above design, the flowchart of the HFOA is presented in Figure 8.

5.7. Computational Complexity Analysis. To analyze the efficiency of the algorithm, computational complexity analysis is necessary. Assume that the population size is P, and the maximum iteration number is NC, the task number is n and the worker is m. According to the procedure of HFOA, its computational complexity can be analysed as follows: in the initialization procedure, the computational complexity of NIH is $O(n^2)$ and the computational complexity of the encoding procedure is $O(P \times n \times m)$. For each iteration, in the decoding procedure, there are find and sort process and the computational complexity is $O(P \times n^2 \times m)$. The computational complexity of fitness evaluation is $O(P \times n)$. In the smell-based search, W-sp, T-sp, and T-re are carried out in turn. For each search strategy, the computational complexity is $O(P \times n)$. Then, the decoding and fitness evaluation procedure are carried out after the search strategy. Therefore, the computational complexity of smell-based search is approximate to $O(3P \times n^2 \times m)$. Similar to smell-based search, the computational complexity of vision-based search procedure is $O(2P \times n^2 \times m)$. In the postoptimization procedure, the computational complexities of 2-Opt and Or-Opt are, respectively, $O(n^2)$ and $O(n^3)$. Usually, it is reasonable to take the highest power item as the computational complexity of the algorithm. Therefore, the analyses above show that the computational complexity is $O\{NC \times P \times (6 \times n^2 \times m + n^3)\}$. The algorithm approximates an $O(n^3)$ level algorithm.

6. Experimental Results and Comparisons

6.1. Test Instance. The HFOA is coded in C++ and executed on a PC with an Intel Core i5 2450M@2.50 GHz CPU. Since there is no public standard benchmark for the FSSP, we built the instances based on the well-known multidepot vehicle routing problem with time windows which were designed by Cordeau [46].

The new instances keep the original coordinates. The number of workers is equal to the number of vehicles in the original instances. There are multiple vehicles in each warehouse in the original instances. Therefore, while setting the original warehouse location as a worker's location, it is also necessary to set some customer locations as worker positions. In addition, for each instance, it is necessary to generate a matrix of workers' proficiency levels for tasks. The matrix is generated randomly, and the range of λ_{ki} is $[0.5, 3]$. For each worker, $b_k = 0, e_k = 1000$. The time windows of the customer in the original instance are $[E_i, L_i]$. In our instance, the time window of the task is $[R_i, T_i]$. $T_i = L_i, R_i = T_i/(1 + (\gamma^{-1})^{\beta^{-1}})$. The service time is set to the standard customer service time. There are 20 instances in our experiments, and the detailed information is described in Table 1.

PR01 represents 4 depots and 48 customers, with 2 vehicles in each depot in the MDVRPTW dataset. Here, we modify it to 8 workers and 44 tasks. 4 workers are located at

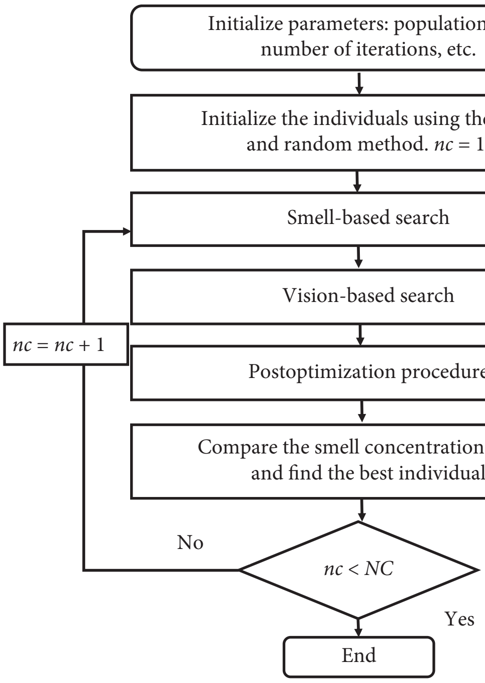
Initialize parameters: population size, number of iterations, etc.
Initialize the individuals using the NIH and random method. $nc = 1$.
Smell-based search
Vision-based search
$nc = nc + 1$
Postoptimization procedure
Compare the smell concentration value, and find the best individual
No
$nc < NC$
Yes
End

Figure 8: Flowchart of the HFOA for FSSP.

Table 1: Instance description.

Instance	$(m \times n)$
PR01	8×44
PR02	12×88
PR03	16×132
PR04	20×176
PR05	24×220
PR06	28×264
PR07	12×66
PR08	18×132
PR09	24×198
PR10	30×264
PR11	4×48
PR12	8×92
PR13	12×136
PR14	16×180
PR15	20×224
PR16	24×268
PR17	6×72
PR18	12×138
PR19	18×204
PR20	24×270

the original depot and the other four are randomly selected from the group of customers. The other instances are similarly converted. It is noted that the difference between PR01 and PR11 is only in the time window. The time window of PR01 is more compact, but PR11 is more relaxed. As a result, PR01 needs more workers.

The parameters of our algorithm are set as follows: the parameters of dissatisfaction function $\gamma = 2.55, \beta = 0.88$, population size $P = 100$, and iteration number $NC = 1000$. To conveniently evaluate and compare the algorithms, the relative percentage increase (RPI) is defined as follows:

$$\mathrm{RPI} = \frac{f_A - f_b}{f_b} \times 100\%, \tag{24}$$

where A denotes HFOA, its variants, and the state-of-the-art algorithms. f_A represents the average result of 20 runs for algorithm A, and f_b is the best result among the algorithms.

6.2. Effectiveness of Three Key Features of the HFOA

6.2.1. Effectiveness of Initialization Method. Firstly, to certificate the effectiveness of the hybrid initialization method for the HFOA, the experiments are implemented to compare the hybrid initialization method with the other two methods. The two methods are NIH and random method, and the corresponding HFOAs with the two approaches are named n-HFOA and r-HFOA. Six instances (PR01, PR05, PR10, PR11, PR15, and PR20) are selected for the test. The RPI results are shown in Table 2. It can be seen from Table 2 that the optimization results of n-HFOA are significantly better than the random method (r-HFOA), and the RPI is improved by 4.76% on average. The optimization results of the hybrid initialization method (HFOA) are better than the n-HFOA, with an average increase of 3.51%. This is because NIH can generate an initial solution with better quality, which helps the algorithm to locate promising regions within the large search space. And the random method can generate diversity distribution initial solution. Therefore, the hybrid initialization method is more effective and efficient.

To evaluate the parameters η_1 and η_2, we implement another comparison experiment. 9 levels are set for η_1 as shown in Table 3. As $\eta_1 + \eta_2 = 1$, the η_2 is not independent and then it is not listed in the table. From Table 3, when $\eta_1 = 0.5$, the results are significantly better than other values. As for η_1 and η_2, moderate values are preferred that is because η_1 and η_2 determine the weight of travel time and service time. A too large η_1 and a too small η_2 may cause overfocusing on the travel time when inserting the new task in the route. On the contrary, a too small η_1 and a too large η_2 may cause overfocusing on the service time when inserting the new task in the route. According to the results of the experiment, the weight coefficient of NIH is set $\eta_1 = 0.5, \eta_2 = 0.5$.

6.2.2. Effectiveness of Fruit Fly Search Strategies. Secondly, we verified the effectiveness of the three search strategies presented above. The three search strategies were executed separately. The results of the RPI on the three search strategies are shown in Table 4. It can be seen from the table that the results of T-sp are the best and the results of W-sp are the worst because W-sp mainly exchanges customers between the workers, although it can change the customers' order, but more limited, relatively rough search. The ability of T-re is stronger than that of W-sp, and the optimization result of RPI is improved by 23.53% on

Table 2: Comparisons of the RPI values for different initialization methods.

	HFOA	*n*-HFOA	*r*-HFOA
PR01	0.93	4.92	8.23
PR05	0.78	2.96	10.27
PR10	0.12	3.80	11.71
PR11	0.74	3.97	8.58
PR15	0.66	3.69	9.91
PR20	0.73	5.70	10.88
Avg.	0.66	4.17	9.93

Table 3: Comparisons of the RPI values for different η_1 values.

η_1	PR01	PR05	PR10	PR11	PR15	PR20	Avg.
0.1	3.31	2.63	3.19	3.62	2.17	2.83	2.96
0.2	2.11	1.19	2.56	3.28	2.79	2.44	2.40
0.3	1.54	1.88	2.05	1.89	1.67	1.85	1.81
0.4	1.75	1.49	0.65	1.48	1.51	1.73	1.44
0.5	0.93	0.78	0.12	0.74	0.66	0.73	0.66
0.6	1.13	0.84	0.21	0.71	0.93	0.91	0.78
0.7	1.34	1.89	1.12	2.07	1.63	1.77	1.63
0.8	1.83	2.45	1.68	2.51	1.91	2.11	2.08
0.9	2.16	2.42	2.65	2.16	2.67	2.92	2.50

Table 4: Comparisons of the RPI values for different search strategies.

	T-sp	T-re	W-sp
PR01	13.95	36.67	56.05
PR05	18.21	29.30	51.71
PR10	11.32	28.41	57.87
PR11	19.36	34.26	48.63
PR15	18.09	36.32	61.76
PR20	16.07	22.44	52.58
Avg.	16.17	31.23	54.76

average. T-re is like an insertion operator, and the search precision is better than that of W-sp. The T-sp operator has the strongest optimization ability in the three search strategies, which is 15.06% higher than the T-re, since T-sp operator does not only change the assignment of tasks between workers, but also changes the order of the tasks. It is like a 2-Opt operator that integrates between innerroutes and interroutes.

Next, to compare the performance of the smell search and vision search, we carry out computational experiments to compare the HFOA with different search procedures. The full version that the smell search and vision search are all executed is named HFOA, the version that only the smell search is executed is named HFOAss, and the version that only the vision search is executed is named HFOAvs. 6 instances (RP01, PR05, PR10, PR11, PR15, and PR20) are tested. The average RPI for the 6 instances is 0.66% for HFOA, 1.89% for HFOAss, and 3.71% for HFOAvs. The RPI values are shown in Figure 9. From the figure, we can see that the full version is significantly better than the others. The HFOAss is more effective than HFOAvs. The smell search executes the three strategies (T-re, T-sp, and W-sp) in turn and selects the best result to update the solution. On the contrary, the vision search is primarily responsible for exploitation, only T-sp and T-re are carried out, and whenever a better solution is found, the solution will be updated.

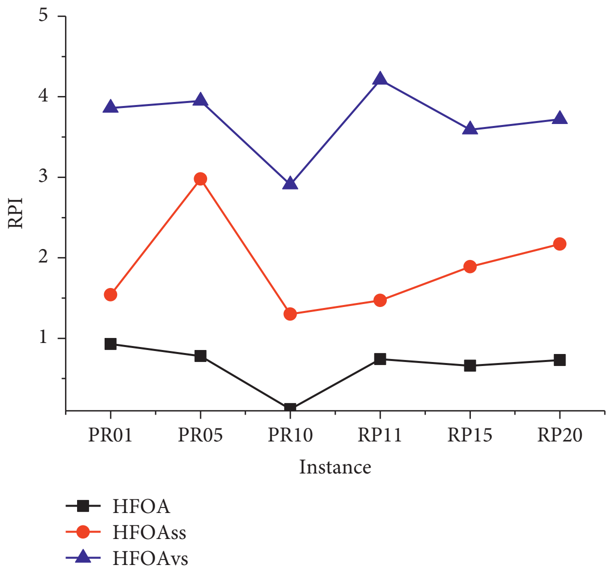

Figure 9: Comparison of smell and vision strategies.

6.2.3. *Effectiveness of Postoptimization.* Furthermore, to verify the effectiveness of the postoptimization search, we carry out computational experiments to compare the HFOA with different postoptimization procedures. Figure 10 shows the effect of the postoptimization procedure on the algorithm. It is clear from the comparisons that the hybrid procedure using 2-Opt and Or-Opt can produce significantly better results than other methods. The postoptimization procedures (2-Opt + Or-Opt) lead to a performance improvement of approximately 3.56%. Using a postoptimization algorithm alone, Or-Opt has a slight advantage over 2 -Opt.

6.3. *Comparison with State-of-the-Art Algorithms.* To verify the performance of the proposed algorithm, we compared the HFOA with the four algorithms: GRASP [9], genetic algorithm (GA) [15], the record-to-record travel algorithm (RTR) [22], and memetic algorithm (MA) [21]. The parameters for the GRASP, GA, RTR, and MA are set to the corresponding value according to the existing literature. The RPI values for the five algorithms are shown in Table 5. The best value is marked in bold. From the table, we can see that HFOA is significantly superior to the other four algorithms. The average RPI value of the HFOA is 0.57%, which is much lower than those four algorithms. Simultaneously, the HFOA performs the best for 18 out of 20 instances, except PR09 and PR19. The MA has an average RPI value of 1.63% and it is ranked second. Meanwhile, it gets the best result in PR09 and PR19. For the other three algorithms, the average RPI of GA is equal to 3.81%, and it is ranked

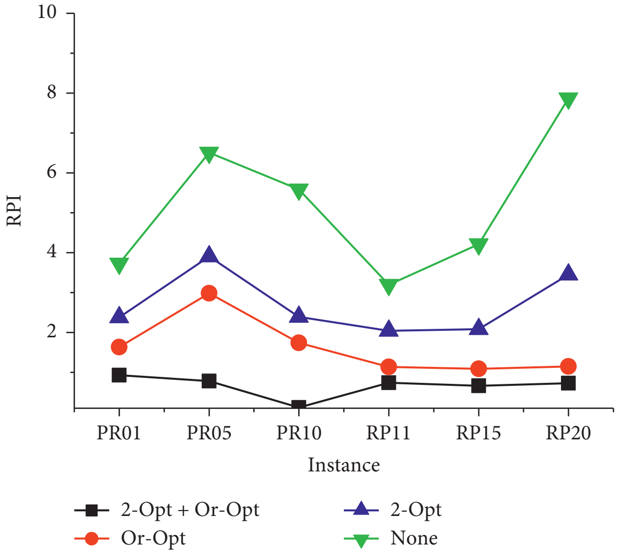

Figure 10: Comparisons of postoptimization.

Table 5: Comparisons of HFOA with the state-of-the-art algorithms.

Instance	GRASP	GA	RTR	MA	HFOA
PR01	8.06	4.47	8.07	2.45	**0.93**
PR02	7.11	3.23	7.85	2.02	**0.25**
PR03	7.32	5.94	10.61	1.47	**0.74**
PR04	9.81	2.47	10.83	1.41	**0.67**
PR05	10.3	2.08	12.79	1.95	**0.78**
PR06	5.58	1.46	9.50	1.78	**0.94**
PR07	8.19	2.59	9.24	1.75	**0.19**
PR08	11.42	4.08	9.29	2.53	**0.95**
PR09	10.92	4.92	10.87	**0.55**	0.64
PR10	10.79	6.25	11.85	1.71	**0.12**
PR11	3.56	3.55	13.79	1.03	**0.74**
PR12	10.67	4.64	9.04	1.22	**0.23**
PR13	7.83	4.56	15.63	0.91	**0.61**
PR14	10.91	3.62	11.77	1.53	**0.79**
PR15	6.89	1.55	16.75	1.96	**0.66**
PR16	4.76	4.31	10.85	1.36	**0.45**
PR17	5.31	5.05	12.08	1.53	**0.17**
PR18	8.63	4.48	12.52	1.72	**0.02**
PR19	13.12	4.84	9.01	**0.78**	0.87
PR20	8.91	2.06	11.95	2.95	**0.73**
Avg.	8.51	3.81	11.21	1.63	0.57

third. The average RPI of GRASP and RTR are 8.51% and 11.21%. Based on the above analysis, we can conclude that HFOA has better performance than other algorithms.

6.4. Application to Solve Vehicle Routing Problem with Time Windows. Furthermore, to verify the ability of HFOA to solve the NP-hard problem, the well-known combined optimization problem vehicle routing problem with time windows is provided in this section. The dataset including 56 Solomon instances with 100 customers is tested in the experiments. These instances are categorized into six different categories: R1, R2, C1, C2, RC1, and RC2. Each category contains R which indicates that customers are distributed randomly, C categories are clustered, and RC means customers are distributed both randomly and clustered. Moreover, the R1, C1, and RC1 instances have narrower time windows for the depot and all customers are associated with very tight time windows. By contrast, the R2, C2, and RC2 instances have a long scheduling horizon that permits many customers to be serviced by the same vehicle, and the time windows defined for the customers are wider.

All instances are repeatedly solved 20 times. The HFOA results are compared to those obtained by other algorithms in the literature. We select six state-of-the-art algorithms that have obtained the best results. These algorithms include the following: genetic algorithm and set partitioning (CGH) [47], genetic algorithm (GA) [48], hybrid ant colony with

Table 6: Comparison of the HFOA with the state-of-the-art methods.

Dataset	Best (TD)	CGH (%)	GA (%)	ACOT (%)	MAPSO (%)	AHSA (%)	TS (%)	HFOA (%)
R1	1156.36	2.34	5.50	4.91	4.64	5.20	4.58	4.31
R2	877.20	2.59	12.37	8.56	8.53	11.00	11.75	7.84
C1	826.70	0.20	2.95	1.84	0.20	1.01	0.20	0.91
C2	589.24	0.11	5.24	3.99	0.11	6.99	0.11	0.22
RC1	1338.18	0.26	2.13	5.79	3.44	3.24	3.81	2.96
RC2	1003.95	1.19	10.41	11.60	11.52	10.17	11.30	8.68
Avg.	965.27	1.11	6.43	6.11	4.74	6.27	5.29	4.15

Table 7: The computational time (average) of the state-of-the-art methods.

	R1	R2	C1	C2	RC1	RC2	Avg.
CGH	—	—	—	—	—	—	3600.00
GA	—	—	—	—	—	—	835.00
ACOT	698.00	655.00	210.00	142.00	317.00	407.00	404.83
MAPSO	**22.58**	**22.04**	**18.55**	**19.55**	**23.43**	**21.58**	**21.29**
AHSA	64.32	635.09	108.14	141.54	109.88	442.07	250.17
TS	—	—	—	—	—	—	—
HFOA	118.90	123.67	98.84	95.16	123.20	131.37	115.19

CPU type: CGH-; GA: Intel Pentium II 266 MMX; ACOT: Intel Pentium 1000 MHz; MAPSO: Intel Core i5 2430M@ 2.40 GHz; AHSA: Intel Xeon 2.8 GHz; TS-; HFOA: Intel Core i5 2450M@2.50 GHz.

tabu search (ACOT) [49], a multiadaptive particle swarm optimization (MAPSO) [50], adaptive harmony search algorithm (AHSA) [51], and a probabilistic tabu search (TS) [52]. The RPI values of the six algorithms and HFOA over all tested instances are presented and compared in Table 6. The best-known results of average total distances (TD) for each category also are reported. The results tabulated in Table 5 show that HFOA achieved competitive results compared to others, among the total 7 heuristics, HFOA is ranked second. The proposed algorithm can achieve the solutions close to the best-known solutions for the cluster problem sets C1 and C2. Moreover, HFOA can obtain comparable solutions for the problem sets R1, R2, RC1, and RC2 compared with other metaheuristics.

To compare the convergence speed with the state-of-the-art algorithms, the average computational time is reported in Table 7. We report the average running time in seconds. The best computational times are highlighted in bold font. It is noted that the CGH and GA do not report the runtime for each category and only provide the total average computational time; TS does not report the running time. CGH and TS also do not report the CPU type. As can be seen from the table, the computational time of HFOA is very competitive with CGH, GA, ACOT, and AHSA. Furthermore, we could observe that the algorithm converges faster in the clustered instances. Considering the tradeoff of quality and computation time, HFOA is the competitive approach among the seven methods.

7. Conclusion

In this paper, we research the field service scheduling problem with customer satisfaction. The customer satisfaction that considered customer behavioral features is described based on the prospect theory. It provides a new way to describe customer satisfaction. The mathematical model for FSSP with customer satisfaction is presented. To solve the NP model, the knowledge of the insertion feasibility on the solution is research. And an effective HFOA is proposed for the model. Based on the prior knowledge of the FSSP and endowment of the Fruit Fly Optimization Algorithm, a novel encoding method based on matrix is proposed for HFOA. To improve the solution quality, an effective nearest neighbour insertion heuristic is developed to generate the initial solution. Additionally, three search strategies and postoptimization procedure are applied to the HFOA.

To verify the ability of HFOA, the well-known VRPTW dataset including 56 Solomon instances with 100 customers is tested in the experiments. The results illustrate that the HFOA algorithm provides excellent results compared to the other six metaheuristic methods. The proposed HFOA is also successfully applied to FSSP. The comparisons verified the effectiveness of the developed heuristic and search strategies. Through comparison with the other four algorithms, the results show that the proposed HFOA is an effective and efficient method to solve the FSSP. Although this work is presented in terms of FSSP with customer satisfaction, the main idea of the HFOA could also be applied to many other FSSP problems such as dynamic task arriving and dynamic road network information, and we will extend the HFOA to these problems in the future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Additional Points

(1) The customer satisfaction that considered the customer behavioral characteristics is described based on the prospect theory, and the mathematical model

for field service scheduling problem with customer satisfaction is formulated.

(2) The knowledge of the insertion feasibility on the solution is analysed. According to the knowledge, the improved nearest neighbour insertion algorithm for initialization is proposed.

(3) The Fruit Fly Optimization Algorithm is presented for FSSP. A matrix encoding method is designed, and three search operators based on swarm behavior and prior knowledge of the problem are proposed. Based on the operators, the smell-based search strategy and the vision-based search strategy are redesigned.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This project was supported by the National Social Science Foundation of China (Grant no. 20BGL025) and National Natural Science Foundation of China (Grant nos. 71671089 and 71701092).

References

[1] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[2] W. Sheng and Y. Bao, "Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle," *Nonlinear Dynamics*, vol. 73, no. 1-2, pp. 611–619, 2013.

[3] P.-W. Chen, W.-Y. Lin, T.-H. Huang, and W.-T. Pan, "Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service," *Applied Mathematics & Information Sciences*, vol. 7, no. 2, pp. 459–465, 2013.

[4] X.-l. Zheng and L. Wang, "A two-stage adaptive fruit fly optimization algorithm for unrelated parallel machine scheduling problem with additional resource constraints," *Expert Systems with Applications*, vol. 65, pp. 28–39, 2016.

[5] L. Wang, X.-L. Zheng, and S.-Y. Wang, "A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem," *Knowledge-Based Systems*, vol. 48, pp. 17–23, 2013.

[6] L. Wang, Y. Shi, and S. Liu, "An improved fruit fly optimization algorithm and its application to joint replenishment problems," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4310–4323, 2015.

[7] Wu B, J. Cheng, and M Dong, "Hybrid fruit fly optimisation algorithm for field service scheduling problem," *International Journal of Automation and Control*, vol. 14, no. 5-6, pp. 554–570, 2020.

[8] D. Lesaint, C. Voudouris, and N. Azarmi, "Dynamic workforce scheduling for British telecommunications plc," *Interfaces*, vol. 30, no. 1, pp. 45–56, 2000.

[9] J. Xu and S. Y. Chiu, "Effective heuristic procedures for a field technician scheduling problem," *Journal of Heuristics*, vol. 7, no. 5, pp. 495–509, 2001.

[10] J.-F. Cordeau, G. Laporte, F. Pasin, and S. Ropke, "Scheduling technicians and tasks in a telecommunications company," *Journal of Scheduling*, vol. 13, no. 4, pp. 393–409, 2010.

[11] C. Akjiratikarl, P. Yenradee, and P. R. Drake, "PSO-based algorithm for home care worker scheduling in the UK," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 559–583, 2007.

[12] I. Petrakis, C. Hass, and M. Bichler, "On the impact of real-time information on field service scheduling," *Decision Support Systems*, vol. 53, no. 2, pp. 282–293, 2012.

[13] A. Goel and F. Meisel, "Workforce routing and scheduling for electricity network maintenance with downtime minimization," *European Journal of Operational Research*, vol. 231, no. 1, pp. 210–228, 2013.

[14] Z. Xu, X. G. Ming, M. Zheng, M. Li, L. He, and W. Song, "Cross-trained workers scheduling for field service using improved NSGA-II," *International Journal of Production Research*, vol. 53, no. 4, pp. 1255–1272, 2015.

[15] J. J. Jiang and L. Y. Wang, "Field product service scheduling with service matching based on genetic algorithm," *Computer Integrated Manufacturing Systems*, vol. 18, no. 11, pp. 2573–2577, 2012, in Chinese.

[16] A. A. Kovacs, S. N. Parragh, K. F. Doerner, and R. F. Hartl, "Adaptive large neighborhood search for service technician routing and scheduling problems," *Journal of Scheduling*, vol. 15, no. 5, pp. 579–600, 2012.

[17] R. B. Damm, M. G. C. Resende, and D. P. Ronconi, "A biased random key genetic algorithm for the field technician scheduling problem," *Computers & Operations Research*, vol. 75, pp. 49–63, 2016.

[18] C. E. Cortés, M. Gendreau, L. M. Rousseau, S. Souyris, and A. Weintraub, "Branch-and-price and constraint programming for solving a real-life technician dispatching problem," *European Journal of Operational Research*, vol. 238, no. 1, pp. 300–312, 2014.

[19] H. Hashimoto, S. Boussier, M. Vasquez, and C. Wilbaut, "A grasp-based approach for technicians and interventions scheduling for telecommunications," *Annals of Operations Research*, vol. 183, no. 1, pp. 143–161, 2011.

[20] V. Pillac, C. Guéret, and A. L. Medaglia, "A parallel matheuristic for the technician routing and scheduling problem," *Optimization Letters*, vol. 7, no. 7, pp. 1525–1535, 2013.

[21] J. Decerle, O. Grunder, A. Hajjam El Hassani, and O. Barakat, "A memetic algorithm for a home health care routing and scheduling problem," *Operations Research for Health Care*, vol. 16, pp. 59–71, 2018.

[22] X. Chen, B. W. Thomas, and M. Hewitt, "The technician routing problem with experience-based service times," *Omega*, vol. 61, pp. 49–61, 2016.

[23] X. Chen, B. W. Thomas, and M. Hewitt, "Multi-period technician scheduling with experience-based service times and stochastic customers," *Computers & Operations Research*, vol. 82, pp. 1–14, 2017.

[24] H. Qin, W. Ming, Z. Zhang, Y. Xie, and A. Lim, "A tabu search algorithm for the multi-period inspector scheduling problem," *Computers & Operations Research*, vol. 59, pp. 78–93, 2015.

[25] E. Zamorano and R. Stolletz, "Branch-and-price approaches for the multiperiod technician routing and scheduling problem," *European Journal of Operational Research*, vol. 257, no. 1, pp. 55–68, 2017.

[26] S. Souyris, C. E. Cortés, F. Ordóñez, and A. Weintraub, "A robust optimization approach to dispatching technicians

under stochastic service times," *Optimization Letters*, vol. 7, no. 7, pp. 1549–1568, 2013.

[27] Y. Borenstein, N. Shah, E. Tsang, R. Dorne, A. Alsheddy, and C. Voudouris, "On the partitioning of dynamic workforce scheduling problems," *Journal of Scheduling*, vol. 13, no. 4, pp. 411–425, 2010.

[28] Q. Tang, G. R. Wilson, and E. Perevalov, "An approximation manpower planning model for after-sales field service support," *Computers & Operations Research*, vol. 35, no. 11, pp. 3479–3488, 2008.

[29] Y. R. Cao and C. F. Han, "Simulation for after-sales field service queuing approximation M/G/m model," *Industrial Engineering and Management*, vol. 14, no. 5, pp. 104–107, 2009.

[30] R. Hu, S. Wen, Z. Zeng, and T. Huang, "A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm," *Neurocomputing*, vol. 221, pp. 24–31, 2017.

[31] T. Meng and Q.-K. Pan, "An improved fruit fly optimization algorithm for solving the multidimensional knapsack problem," *Applied Soft Computing*, vol. 50, pp. 79–93, 2017.

[32] L. Shen, H. Chen, Z. Yu et al., "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[33] X.-l. Zheng, L. Wang, and S.-Y. Wang, "A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem," *Knowledge-Based Systems*, vol. 57, pp. 95–103, 2014.

[34] J. Niu, W. Zhong, Y. Liang, N. Luo, and F. Qian, "Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization," *Knowledge-Based Systems*, vol. 88, pp. 253–263, 2015.

[35] T. Li, L. Gao, P. Li, and Q. Pan, "An ensemble fruit fly optimization algorithm for solving range image registration to improve quality inspection of free-form surface parts," *Information Sciences*, vol. 368, pp. 953–974, 2016.

[36] L. Wang and X.-L. Zheng, "A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem," *Swarm and Evolutionary Computation*, vol. 38, pp. 54–63, 2018.

[37] X. Han, Q. Liu, H. Wang, and L. Wang, "Novel fruit fly optimization algorithm with trend search and co-evolution," *Knowledge-Based Systems*, vol. 141, pp. 1–17, 2018.

[38] L. Wu, Q. Liu, X. Tian, J. Zhang, and W. Xiao, "A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems," *Knowledge-Based Systems*, vol. 144, pp. 153–173, 2018.

[39] X. Yuan, Y. Liu, Y. Xiang, and X. Yan, "Parameter identification of BIPT system using chaotic-enhanced fruit fly optimization algorithm," *Applied Mathematics and Computation*, vol. 268, pp. 1267–1281, 2015.

[40] X. Lei, Y. Ding, H. Fujita, and A. Zhang, "Identification of dynamic protein complexes based on fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 105, pp. 270–277, 2016.

[41] K. Peng, Q.-K. Pan, L. Gao, X. Li, S. Das, and B. Zhang, "A multi-start variable neighbourhood descent algorithm for hybrid flowshop rescheduling," *Swarm and Evolutionary Computation*, vol. 45, pp. 92–112, 2019.

[42] 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, 2014.

[43] R. Chen, M. Gen, and T. Tozawa, "Vehicle routing problem with fuzzy due-time using genetic algorithms," *Japanese Journal of Fuzzy Theory and Systems*, vol. 7, no. 5, pp. 1050–1061, 1995.

[44] J. Tang, Z. Pan, R. Y. K. Fung, and H. Lau, "Vehicle routing problem with fuzzy time windows," *Fuzzy Sets and Systems*, vol. 160, no. 5, pp. 683–695, 2009.

[45] A. Tversky and D. Kahneman, "Advances in prospect theory: cumulative representation of uncertainty," *Journal of Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.

[46] http://neo.lcc.uma.es/vrp/vrp-instances/description-for-files-of-cordeaus-instances/.

[47] G. B. Alvarenga, G. R. Mateus, and G. De Tomi, "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 34, no. 6, pp. 1561–1584, 2007.

[48] K. C. Tan, L. H. Lee, Q. L. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 281–295, 2001.

[49] B. Yu, Z. Z. Yang, and B. Z. Yao, "A hybrid algorithm for vehicle routing problem with time windows," *Expert Systems with Applications*, vol. 38, no. 1, pp. 435–441, 2011.

[50] Y. Marinakis, M. Marinaki, and A. Migdalas, "A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows," *Information Sciences*, vol. 481, pp. 311–329, 2019.

[51] E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "An adaptive hybrid algorithm for vehicle routing problems with time windows," *Computers & Industrial Engineering*, vol. 113, pp. 382–391, 2017.

[52] Y. Rochat and D. Taillard É, "Probabilistic diversification and intensification in local search for vehicle routing," *Journal of Heuristics*, vol. 1, no. 1, pp. 47–167, 1995.