An Enhanced Artificial Electric Field Algorithm with Sine Cosine Mechanism for Logistics Distribution Vehicle Routing

Hongyu Zheng 1, Juan Gao 1, Juxia Xiong 1,∗, Guanglei Yao 1, Hongjiang Cui 2,∗ and Lirong Zhang 3

1 School of Mathematics and Physics, Guangxi Minzu University, Nanning 530006, China; zhy19982021reg@163.com (H.Z.); gaojuan2022512@163.com (J.G.); 18178637417@163.com (G.Y.)
2 School of Locomotive and Rolling Stock Engineering, Dalian Jiaotong University, Dalian 116028, China
3 College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China; a13536308535@163.com
∗Correspondence: xiongjuxia@gxun.edu.cn (J.X.); chj@djtu.edu.cn (H.C.)

Abstract: Aiming at the scheduling problem of logistics distribution vehicles, an enhanced artificial electric field algorithm (SC-AEFA) based on the sine cosine mechanism is proposed. The development of the SC-AEFA was as follows. First, a map grid model for enterprise logistics distribution vehicle path planning was established. Then, an enhanced artificial electric field algorithm with the sine cosine mechanism was developed to simulate the logistics distribution vehicle scheduling, establish the logistics distribution vehicle movement law model, and plan the logistics distribution vehicle scheduling path. Finally, a distribution business named fresh enterprise A in the Fuzhou Strait Agricultural and Sideline Products Trading Market was selected to test the effectiveness of the method proposed. The theoretical proof and simulation test results show that the SC-AEFA has a good optimization ability and a strong path planning ability for enterprise logistics vehicle scheduling, which can improve the scheduling ability and efficiency of logistics distribution vehicles and save transportation costs.

Keywords: artificial electric field algorithm; sine cosine mechanism; Coulomb’s law; time window; path optimization

1. Introduction

Many scientific and engineering problems in the fields of industry, agriculture, transportation, information, and management, among others, belong to optimization problems, or can be transformed into optimization problems [1–4]. These optimization problems are one of the hotspots and difficulties recognized by the experts and scholars at home and abroad. Mature optimization theories and methods have been widely used in these fields [5–7]. With the deepening of human cognitive activities and the rapid development of computer science and technology, the optimization problems faced in the real world are becoming increasingly diversified, large-scale, and complex. Additionally, these optimization problems present complex features such as being highly dynamic, high-dimensional, nonlinear, multi-objective, multi-constraint, discontinuous, and non-differentiable. This makes it difficult for experts and scholars using traditional optimization theories and methods to solve problems from the perspective of analyzing the mathematical characteristics of the problem [8–12]. Therefore, seeking new intelligent optimization algorithms with high efficiency, parallelism, adaptability, and robustness to solve large-scale complex optimization problems has become an urgent and persistent research topic in the field of engineering application, which not only has important theoretical significance but also has a wide application prospect.

Although there are many types of optimization algorithms in the field of intelligent optimization, none of them can solve all optimization problems. Therefore, a large number of new optimization algorithms are constantly being proposed or improved. In recent
decades, meta-heuristic algorithms based on the population have achieved good performance in solving complex engineering optimization problems: for example, the genetic algorithm (GA) [13], the simulated annealing algorithm (SAA) [14], particle swarm optimization (PSO) [15], the bat algorithm (BA) [16], ant colony optimization (ACO) [17], the novel moth to fire algorithm (MFO) [18], the locust optimization algorithm (GOA) [19], the butterfly optimization algorithm (BOA) [20], and the sine cosine optimization algorithm (SCA) [21], among others [22–31]. These algorithms can be grouped into two broad categories: individual-based and group-based. The first class generates only one solution, while the second class generates multiple solutions.

The artificial electric field algorithm (AEFA) [32] is different from these swarm intelligence bionic algorithms; it simulates the process of the relative motion of the charged particles under Coulomb force in the electrostatic field. The AEFA has the advantages of a simple structure, less adaptive parameters, and a simple algorithm principle. With the increasingly scattered urban commercial centers and the increasing demand for logistics, vehicle routing problems have gradually become the focus of research by scholars of intelligent optimization algorithms. The vehicle routing problem (VRP) [33] is a crucial part of logistics distribution optimization. In the process of logistics distribution, by selecting the optimal transportation route for the distribution vehicles and rationally arranging the scheduling sequence of the vehicles, the empty driving rate and driving distance of the vehicles can be effectively reduced. This problem was first proposed by Dantzig and Ramser in 1959 and belongs to the NP-hard problem [34]. At present, scholars at home and abroad have carried out extensive and in-depth research on the VRP, and the VRP has been applied to many social fields such as the route optimization of garbage trucks and the delivery route optimization of chain stores [35]. Since the traditional vehicle routing problem has less constraints, the vehicle cost is usually taken as the total cost as the optimization objective, which leads to great limitations in practical applications [36]. Therefore, the vehicle routing problem with time window constraints has become a research hotspot. Currently, many novel intelligent algorithms are applied to solve the vehicle routing problem with time window constraints [37]. Tas solved the vehicle routing problem with time-dependent soft time windows and random travel times by using the tabu search algorithm and the adaptive large neighborhood search algorithm, and showed that the method could be used to solve some other large-scale problems [38]. Iqbal used artificial bee colonies that simulate the foraging behavior of bees and a hybrid hyper-heuristic algorithm to solve the multi-objective vehicle routing problem with time windows [39]. Wang proposed a method to describe the spatial distribution of solutions using a distribution estimation algorithm using the generalized Mallows distribution as a probabilistic model to solve vehicle routing problems with time windows [40].

This paper starts from an actual scientific engineering optimization problem, takes the artificial electric field algorithm as the research object, and uses the characteristics of the sine cosine algorithm (SCA) so that it is not easy to fall into a local optimum and has a fast convergence speed. The method of combining theoretical analysis, numerical calculation, and engineering application was adopted. The theoretical system of the artificial electric field algorithm for solving intelligent logistics scheduling problems was constructed. The performance of the new algorithm was tested and verified through practical engineering optimization problems to effectively and systematically research the framework, theory, and application of the new optimization algorithm. The work conducted in this paper is the first application of the artificial electric field algorithm to the VRPTW, which is an innovative application. The main problem faced in the solution process of the simulation experiment is how to code and optimize. After many attempts, it was finally decided to use the natural number encoding method, combined with ant colony optimization, to solve the encoding strategy of such problems, and the improved algorithm of this paper was used for loop iteration.

The innovations and main contributions of this paper are as follows:
An enhanced artificial electric field algorithm with the sine cosine mechanism, named SC-AEFA, is proposed; An optimization scheduling algorithm for logistics distribution vehicles based on the SC-AEFA is proposed for the first time; A map grid model for enterprise logistics distribution vehicle path planning is established; Comprehensive experiments were designed and executed to prove the effectiveness of the SC-AEFA through test functions and the case of an actual distribution business, named fresh enterprise A.

2. Basic Knowledge

2.1. Artificial Electric Field Algorithm

The artificial electric field algorithm (AEFA) is a novel meta-heuristic optimization algorithm to simulate the mutual motion of charged particles in an electrostatic field. In order to simplify the operation, the AEFA only considers the attractive force between particles, ignoring the repulsive force between charged particles to ensure that particles with a large charge can attract the nearby particles with a low charge.

Each charged particle represents a feasible solution in the search space. The fitness value of each solution is determined by the charge of each charged particle. The higher the charge of the particles, the closer the feasible solutions are to the optimal solution. The motion principle of the charged particles is shown in Figure 1. Each circle represents a charged particle, and the size of the circle represents the amount of charge. The charged particle \( Q_1 \) is attracted to the other three charged particles and finally forms a resultant force \( F \) and acceleration in this direction. It can be seen that \( Q_4 \) has the largest amount of charge and has the greatest attraction to \( Q_1 \), so the direction of the resultant force on \( Q_1 \) is closer to the direction of attraction of \( Q_4 \) to \( Q_1 \). Therefore, in the iteration process of the AEFA, the charged particles with a small charge in the search space will move to the charged particles with a high charge, so that the algorithm converges to the optimal solution [41].

![A schematic diagram of the force on the charged particles.](image)

Figure 1. A schematic diagram of the force on the charged particles.
The mathematical model of the AEFA can be described by Equations (1) and (2) [42].

\[ V_{i}^{t+1} = \text{rand} \times V_{i}^{t} + a_{i}^{t} \]  
\[ X_{i}^{t+1} = X_{i}^{t} + V_{i}^{t} \]  

where \( \text{rand} \) is a random number in the interval \([0, 1]\); \( V_{i}^{t+1} \) represents the velocity of the \( i \)-th charged particle at the \((t + 1)\)-th iteration; \( X_{i}^{t+1} \) represents the position of the \( i \)-th charged particle at the \((t + 1)\)-th iteration; and \( a_{i}^{t} \) represents the acceleration of the \( i \)-th charged particle at the \( t \)-th iteration.

According to Newton’s second law, when the particle is subjected to the electric field force, the expression of acceleration is described as follows.

\[ a_{i}^{t} = \frac{Q_{i}^{t} \times E_{i}^{t}}{m_{i}^{t}} \]  

At the \( t \)-th iteration, the electric field intensity of the \( i \)-th particle is

\[ E_{i}^{t} = \frac{F_{i}^{t}}{Q_{i}^{t}} \]  

The resultant force of the charged particle in the solution space is

\[ F_{i}^{t} = \sum_{j=1, j \neq i}^{N} \text{rand} \times F_{ij}^{t} \]  

The resultant force is calculated according to the Coulomb force on the particle. The Coulomb force on the charged particle is modeled as follows.

\[ F_{ij}^{t} = K^{t} \times \frac{Q_{i}^{t} \times Q_{j}^{t} \times (X_{best}^{t} - X_{i}^{t})}{R_{ij}^{t} + \epsilon} \]  
\[ R_{ij}^{t} = \| X_{i}(t), X_{j}(t) \|_{2} \]  
\[ K^{t} = K_{0} \times \exp \left( -\alpha \times \frac{\text{iteration}}{\text{Maxiteration}} \right) \]  

In Equation (6), \( F_{ij}^{t} \) represents the Coulomb force of charged particle \( i \) and charged particle \( j \) at the \( t \)-th iteration; \( K^{t} \) is the Coulomb constant; \( X_{best}^{t} \) represents the optimal individual at the \( t \)-th iteration; \( R_{ij}^{t} \) represents the distance between charged particle \( i \) and charged particle \( j \); \( \epsilon \) is a very small positive number. In Equation (8), \( K_{0} = 500; \alpha = 30; \) iteration represents the current number of iterations; Maxiteration represents the maximum number of iterations. The change in the Coulomb constant makes the algorithm perform a global search in the early iteration and turn to a local search in the later iteration. The \( K^{t} \) iteration process is shown in Figure 2.
The charge of all particles in the first-generation population is the same (set to 1). Starting from the second generation, the iterative formula of the charge $Q_i$ is as follows:

$$Q_i^t = \frac{q_i^t}{\sum_{i=1}^{N} q_i^t}$$

$$q_i^t = \exp\left(\frac{\text{fit}_i^t - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}\right)$$

Among them, $q_i^t$ represents the charge of the $i$-th particle at the $t$-th iteration; $Q_i^t$ is the normalization of $q_i^t$; $N$ is the number of particles; $\text{fit}_i^t$ is the fitness value of the $i$-th individual at the $t$-th iteration; $\text{best}(t)$ and worst$(t)$ represent the optimal fitness value and the worst fitness value at the $t$-th iteration, respectively (the same is true in Algorithm 1).

The position of the algorithm at each iteration is determined by the greedy strategy:

$$p_i^{t+1} = \begin{cases} p_i^t & \text{when } \text{fit}(p_i^t) < \text{fit}(X_i^{t+1}) \\ X_i^{t+1} & \text{when } \text{fit}(p_i^t) \geq \text{fit}(X_i^{t+1}) \end{cases}$$

Equation (11) represents the selection method of the position of the individual in the next generation. If the fitness of the updated individual is worse than that of the previous generation, the position of the individual will not change; otherwise, the updated individual is used as the next generation.

To sum up, the pseudocode of the artificial electric field algorithm is described as follows.

**Algorithm 1** Pseudocode of the artificial electric field algorithm (AEFA).

- **Initialization:**
  - Randomly initialize $(X_1^t, X_2^t, \ldots, X_N^t)$ of population size $N$ in the search range $[X_{min}, X_{max}]$;
  - Initialize the velocity to 0;
  - Evaluate the fitness values $(\text{fit}_1^t, \text{fit}_2^t, \ldots, \text{fit}_N^t)$ of agent $X$;
  - Set iteration $t = 1$;
  - Reproduction and Updating

while stopping criterion is not satisfied do
  for $i = 1$ to $N$ do
    Evaluate the fitness values $\text{fit}_i^t$
    Calculate the total force in each direction $F_i^t$
    Calculate the acceleration $a_i^t$
    $V_i^{t+1} = \text{rand} \times V_i^t + a_i^t$
    $X_i^{t+1} = X_i^t + V_i^{t+1}$
  end for
end while
2.2. The Sine Cosine Algorithm and Its Basic Principles

The sine cosine algorithm (SCA) is a new meta-heuristic optimization algorithm based on mathematical trigonometric functions proposed by Seyedali Mirjalili in 2016. The SCA utilizes the mathematical properties of the sine and cosine functions, performs a global search and local search of the algorithm by adding adaptive parameters to change the amplitude of the sine and cosine functions, and finally finds the optimal solution. The update formula of the position of the individual in the SCA is described as follows [43].

\[
X_i^t = \begin{cases} 
X_i^t + r_1 \times \sin(r_2) \times |P_{\text{best}}^t - X_i^t|, & r_4 < 0.5 \\
X_i^t + r_1 \times \cos(r_2) \times |P_{\text{best}}^t - X_i^t|, & r_4 \geq 0.5
\end{cases}
\]  
\[
r_1 = a \left( 1 - \frac{\text{iteration}}{\text{Maxiteration}} \right)
\]  
\[
r_2 = \text{rand}(0, 2\pi)
\]  
\[
r_3 = \text{rand}(0, 2)
\]  
\[
r_4 = \text{rand}(0, 1)
\]  

In Equation (12), \(X_i^t\) represents the position of the \(i\)-th individual at the \(t\)-th iteration; \(P_{\text{best}}^t\) represents the optimal individual in the population at the \(t\)-th iteration; \(r_1, r_2, r_3, \text{ and } r_4\) are four key parameters in the sine cosine algorithm, where \(r_1\) controls the individual search direction and the degree of influence of the step size in the iteration process, \(r_2\) controls the search distance of the individual in the iterative process, \(r_3\) is a random weight coefficient, which determines the optimal individual’s impact on the current influence degree of the individual, and \(r_4\) is the conversion switch of the sine mechanism or cosine mechanism of the control algorithm. In Equation (13), \(a\) is a constant (generally 2); \(\text{iteration}\) is the current iteration number; \(\text{Maxiteration}\) is the maximum iteration number.

The iteration principle of the SCA is shown in Figure 3 [44], in which \(r_1 = 2, r_3 = 1\). It can be seen from the figure that when \(1 < |r_1 \sin(r_2)| < 2\) and \(1 < |r_1 \cos(r_2)| < 2\), the algorithm performs a global search; when the value of \(r_1 \sin(r_2)\) or \(r_1 \cos(r_2)\) belongs to the interval \([-1, 1]\), the algorithm performs a local search in the explored space. The purpose of a global search is to quickly browse the search space and lock the range of the optimal solution, while that of a local search is to find the optimal solution within the locked range.

![Figure 3. The iteration principle of the SCA.](image)

3. SC-AEFA

3.1. Fundamental

Too much global searching will make the algorithm inefficient, and too much local searching will make the algorithm fall into a local optimum prematurely, resulting in unsatisfactory final results. Therefore, balancing the global search and local search can
better improve the performance of the algorithm [45]. It was found that the position update of the artificial electric field algorithm is controlled by the position of the charged particle and the mutual attraction of the surrounding particles. The information interaction ability between the charged particles is strong, so the AEFA has a strong local search ability and a weak global search ability. Compared with the AEFA, the SCA can balance the global search and local development better. Therefore, the updating mechanism of the SCA is integrated into the AEFA (SC-AEFA), which changes the iterative method of the AEFA, makes the global search and local search of the algorithm achieve a dynamic balance, and enhances the optimization ability of the algorithm.

In order to carry out the global search as much as possible in the early stage of the algorithm, the local search of the optimization range is carried out in the later stage; it is specified that when the number of iterations is less than 200, the algorithm uses a modified sine cosine iteration formula to update the position.

### 3.2. Improved Artificial Electric Field Algorithm

Aiming at the shortcomings of the AEFA, this paper made the following improvements to the original artificial electric field algorithm: a partial repulsion factor was introduced on the basis of the original algorithm, that is, far away from the current worst individual \(X_{t \text{worst}}^t\). In addition, in order to reduce the influence of the step size in the later iteration of the algorithm and improve the role of the particle’s own position in the iteration, the parameter \(\gamma\) was added, which decreases as the number of iterations increases; the improved iterative formula is given as follows.

\[
X_{t+1}^i = X_t^i - X_{t \text{worst}}^t + \gamma V_t^i \quad (17)
\]

\[
\gamma = 2e^2 \left(1 - \frac{\text{iteration}}{\text{Maxiteration}}\right) \sin(t\pi)^3 \quad (18)
\]

### 3.3. Improved Sine Cosine Algorithm

In order to better coordinate the global search and local search of the SCA, the sine cosine algorithm was improved. Section 2.2 points out that the parameter \(r_1\) is extremely important and determines the influence of the individual search direction and step size during the iterative process. However, \(r_1\) as a linear function does not favor a sufficient global search in the early stage of the algorithm iteration; this will lead to a decline in the local optimization ability in the later stage of the algorithm. Therefore, the parameter \(r_1\) can be improved as follows [46].

\[
r_1 = 2 \times \frac{\text{iteration}}{\text{Maxiteration}} \cos(\pi t)^3 \quad (19)
\]

Compared with the previous one, the improved \(r_1\) expands the global search range of the SCA and uses the mathematical characteristics of the cosine function to make \(r_1\) oscillate up and down within a fixed range, which better balances the global search and local search of the algorithm.

### 3.4. Steps of SC-AEFA

Based on the improved artificial electric field algorithm, the improved sine cosine algorithm we introduced to obtain the artificial electric field algorithm with the sine cosine mechanism (SC-AEFA). Combining the advantages of the two algorithms, the SC-AEFA not only has a strong global search ability but also has the ability to avoid falling into a local optimum. The pseudocode of the SC-AEFA is shown in Algorithm 2.
**Algorithm 2** Pseudocode of the SC-AEFA.

Initialization;
Randomly initialize \( (X^1_t, X^2_t, \ldots, X^N_t) \) of population size \( N \) in the search range \([X_{min}, X_{max}]\);
Initialize the velocity to 0;
Evaluate the fitness values \( (fit^1_t, fit^2_t, \ldots, fit^N_t) \) of agent \( X \);
Set various parameters;
Reproduction and Updating

while Stopping Criterion is not satisfied do

\[ V_{i}^{t+1} = rand \times V_{i}^t + a_{i}^t \]
if iteration < 200 do
\[ X_{i}^{t+1} = r_1 \times \sin(r_2) \times |r_3 \times P_{best}^t - X_{i}^t| \]
else
\[ X_{i}^{t+1} = r_1 \times \cos(r_2) \times |r_3 \times P_{best}^t - X_{i}^t| \]
end if
else
\[ X_{i}^{t+1} = X_{i}^t - X_{worst} + \gamma \times V_{i}^t \]
end if

end for
end while

4. Simulation Experiment and Result Analysis

In order to test the performance of the SC-AEFA, the artificial electric field algorithm (AEFA), artificial electric field algorithm with chaotic mapping (IAEFA), butterfly optimization algorithm (BOA), and grasshopper optimization algorithm (GOA) were used as a control group, and six groups of functions (as shown in Table 1) were used for testing [47–50]. The software used was MATLAB R2016a, the operating environment was a 64-bit Windows 10 operating system, and the computer hardware parameters were as follows: the processor type was an AMD Ryzen 5 3500U, the on-board RAM was 12.0 GB, and the system was a 64-bit operating system.

**Table 1.** The test function.

| Function | Search Space | Theoretical Value |
|----------|--------------|-------------------|
| \( f_1(x) = \sum_{i=1}^{D} x_i^2 \) | \([-100, 100]\) | 0 |
| \( f_2(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i| \) | \([-10, 10]\) | 0 |
| \( f_3(x) = \left( \sum_{i=1}^{D} x_i \right)^2 \) | \([-100, 100]\) | 0 |
| \( f_4(x) = \max\{ |x_i|, 1 \leq i \leq D \} \) | \([-100, 100]\) | 0 |
| \( f_5(x) = \sum_{i=1}^{D} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2] \) | \([-30, 30]\) | 0 |
| \( f_6(x) = \sum_{i=1}^{D} (|x_i + 0.5|)^2 \) | \([-100, 100]\) | 0 |

The above five algorithms used the same population size \( N = 50 \), dimension \( D = 30 \), and maximum number of iterations \( \text{Maxiteration} = 1000 \). In order to ensure the validity of the experimental results, each group of functions was run independently 20 times, and the mean value and variance of the running results of each algorithm were compared. The mean value represents the final solution accuracy of the algorithm, and the variance
represents the stability of the algorithm. The running results of the five algorithms on the above six standard test functions are shown in Table 2.

Table 2. A comparison of the optimization results of different algorithms.

| Function | Algorithm | Mean Value   | Variance   |
|----------|-----------|--------------|------------|
| $f_1$    | GOA       | $7.02 \times 10^{-4}$ | $2.03 \times 10^2$ |
|          | BOA       | $2.54 \times 10^{-13}$ | $1.02 \times 10^{-12}$ |
|          | AEFA      | $1.09 \times 10^{-24}$ | $9.56 \times 10^{-2}$ |
|          | IAEFA     | $1.29 \times 10^{-6}$ | $4.46 \times 10^{-6}$ |
|          | SC-AEFA   | $5.79 \times 10^{-82}$ | $6.23 \times 10^{-89}$ |
| $f_2$    | GOA       | $6.17 \times 10^{-4}$ | $4.58 \times 10^0$ |
|          | BOA       | $9.22 \times 10^{-12}$ | $1.36 \times 10^{-9}$ |
|          | AEFA      | $5.79 \times 10^{-12}$ | $9.87 \times 10^0$ |
|          | IAEFA     | $1.46 \times 10^{-2}$ | $4.50 \times 10^{-3}$ |
|          | SC-AEFA   | $1.32 \times 10^{-46}$ | $6.15 \times 10^{-67}$ |
| $f_3$    | GOA       | $2.76 \times 10^2$ | $7.73 \times 10^2$ |
|          | BOA       | $1.94 \times 10^{-12}$ | $2.24 \times 10^{-13}$ |
|          | AEFA      | $1.13 \times 10^3$ | $4.27 \times 10^2$ |
|          | IAEFA     | $8.55 \times 10^{-4}$ | $1.26 \times 10^1$ |
|          | SC-AEFA   | $1.83 \times 10^{-89}$ | $7.38 \times 10^{-99}$ |
| $f_4$    | GOA       | $9.63 \times 10^0$ | $4.00 \times 10^0$ |
|          | BOA       | $9.88 \times 10^{-12}$ | $1.45 \times 10^{-1}$ |
|          | AEFA      | $2.00 \times 10^0$ | $9.74 \times 10^{-1}$ |
|          | IAEFA     | $1.30 \times 10^{-3}$ | $9.86 \times 10^{-4}$ |
|          | SC-AEFA   | $4.39 \times 10^{-47}$ | $2.55 \times 10^{-66}$ |
| $f_5$    | GOA       | $1.76 \times 10^2$ | $3.00 \times 10^4$ |
|          | BOA       | $2.90 \times 10^1$ | $3.58 \times 10^{-10}$ |
|          | AEFA      | $2.77 \times 10^1$ | $2.27 \times 10^3$ |
|          | IAEFA     | $2.40 \times 10^1$ | $1.24 \times 10^1$ |
|          | SC-AEFA   | $2.31 \times 10^{-1}$ | $3.36 \times 10^{-2}$ |
| $f_6$    | GOA       | $2.39 \times 10^{-1}$ | $7.06 \times 10^{-1}$ |
|          | BOA       | $5.88 \times 10^0$ | $3.76 \times 10^{-10}$ |
|          | AEFA      | $1.50 \times 10^{-2}$ | $1.59 \times 10^0$ |
|          | IAEFA     | $0.00 \times 10^0$ | $0.00 \times 10^0$ |
|          | SC-AEFA   | $0.00 \times 10^0$ | $0.00 \times 10^0$ |

Table 2 shows the mean and variance of 20 independent runs of the test functions for the GOA, BOA, AEFA, IAEFA, and SC-AEFA. It can be seen from the operation results that the performance and accuracy of the SC-AEFA were significantly stronger than those of the other four algorithms. For the function $f_6$, the SC-AEFA can directly search for the optimal value 0, and the optimization effect can reach 100%. Since the function $f_5$ belongs to the banana function, the optimization result of the SC-AEFA is not ideal, but it is still many orders of magnitude more accurate than that of the other four algorithms. Therefore, the performance of the SC-AEFA has obvious competitive advantages.

The convergence trend comparison diagram of the five algorithms is presented below to observe the differences between the five algorithms more intuitively.

It can be seen from Figure 4 that the SC-AEFA had obvious advantages in the convergence speed compared with the other algorithms. In addition, as shown in Figure 4b, although the SC-AEFA falls into a local optimum in the early stage of the iteration, as the number of iterations increases, the SC-AEFA escapes this situation. This shows that the SC-AEFA has a strong ability to deviate from the local optimum, that is, it has a strong global search ability and robustness. In Figure 4b,d,e, the BOA and SC-AEFA showed little difference in the convergence speed, but from the analysis of the results, the SC-AEFA had a higher accuracy.
many orders of magnitude more accurate than that of the other four algorithms. Therefore, the performance of the SC-AEFA has obvious competitive advantages.

The convergence trend comparison diagram of the five algorithms is presented below to observe the differences between the five algorithms more intuitively.

It can be seen from Figure 4 that the SC-AEFA had obvious advantages in the convergence speed compared with the other algorithms. In addition, as shown in Figure 4b, although the SC-AEFA falls into a local optimum in the early stage of the iteration, as the number of iterations increases, the SC-AEFA escapes this situation. This shows that the SC-AEFA has a strong ability to deviate from the local optimum, that is, it has a strong global search ability and robustness. In Figure 4b,d,e, the BOA and SC-AEFA showed little difference in the convergence speed, but from the analysis of the results, the SC-AEFA had a higher accuracy.

Figure 4. The comparison of the convergence results of different algorithms.

5. Practical Problem Solving and Analysis

5.1. Model Analysis

Unlike the traditional vehicle routing problem, this paper added time window constraints. Each customer has its own service time, and arriving at the customer point earlier or later will be penalized by the time window, which will increase the cost. Furthermore, this is not limited to one cargo vehicle. Within the range of the distribution center, additional delivery vehicles can be dispatched, but each vehicle has a usage cost. On the premise of the above conditions, the distribution vehicle needs to minimize the distribution cost while completing the transportation task.

5.2. Model Assumptions and Variable Descriptions

5.2.1. Model Assumptions

1. The delivery vehicle only delivers a single type of goods;
2. The needs of customers in the delivery process are sequentially met;
3. The specifications of the delivery vehicles are unified;
4. After the delivery vehicle completes the delivery task, it needs to return to the delivery center;
5. The whole distribution process does not consider the influence of other special factors (traffic congestion, vehicle failure, etc.) and the driving speed is constant;
6. The time when the delivery vehicle leaves the distribution center is calculated from 0:00;
7. The road gradient of each driving route in the distribution network is 0;
8. The service time of the delivery vehicles at each demand point is equal (the service time can also be set according to the actual situation).

5.2.2. Symbol and Variable Description

1. Parameter Symbol Description (as shown in Table 3)
2. Decision Variable Description When the vehicle $k$ drives from node $i$ to node $j$, $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$. When the vehicle $k$ is put into use, $y_k = 1$; otherwise, $y_k = 0$. 
5.3. Establishment of VRPTW Model

The vehicle routing problem with the time window constraints (VRPTW) optimization model is as follows:

\[
\min Z = \left( c_0 \sum_{k \in K} y_k + c_1 \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ijk} \right) + e_{pu} \sum_{i \in N} \max \left( e_i - t^1_{ij}, 0 \right) + l_{pu} \sum_{i \in N} \max \left( t^1_{ij} - l_i, 0 \right)
\]  

The objective function indicates that the total delivery cost was the lowest. The total cost consists of the vehicle fixed cost, travel distance cost, and time window penalty cost. The model constraints are as follows.

\[
\sum_{j \in N, j \neq j} x_{ijk} = 1, \forall i \in N, k \in K
\]  

\[
\sum_{j \in V, j \neq j} x_{0jk} \leq 1, k \in K
\]  

\[
\sum_{i \in K} y_k \leq |K|
\]  

\[
\sum_{i \in V, j \neq j} x_{ijk} - \sum_{i \in V, j \neq j} x_{ijk} = 0, \forall j \in V, k \in K
\]  

\[
0 \leq h^1_k \leq h^1_k - p_i x_{ijk} + c \left( 1 - x_{ijk} \right), \forall i \in V, \forall j \in V, i \neq j, k \in K
\]  

\[
0 \leq h_{0k} \leq c, k \in K
\]  

\[
t_{0i} = \max \left[ 0, \left( e_i - t^1_{ij} \right) \right], i \in N
\]  

\[
t^2_{ij} = t^1_{ij} + t_{0i} + t_{oi} + c_{ij}, i \in N \cup F
\]  

\[
t_{ij} = \frac{d_{ij}}{S}, \forall i, j \in V
\]  

\[
t^1_{ij} = \sum_{i \in V, j \in V, j \neq j} x_{ijk} \left( \bar{t}^2_{ij} + t_{ij} \right), \forall k \in K
\]
Equation (21) indicates that the demand of each customer point is satisfied only once by one vehicle. Equation (22) indicates that each vehicle only serves customer points on one path. Equation (23) indicates that the number of vehicles put into use does not exceed the total number of vehicles in the distribution center. Equation (24) indicates that the traffic flow remains balanced at each customer point. Equation (25) represents the residual cargo weight of the vehicle at customer point \( j \). Equation (26) indicates that the load of the vehicle does not exceed the maximum load of the vehicle in the process of driving. Equation (27) indicates the waiting time \( t_{\omega i} (= e_i - t_1^i) \) for the vehicle when it arrives at the customer point early. If the vehicle does not arrive in advance, \( t_{\omega i} = 0 \). Equation (28) represents the time when the vehicle departs from customer point \( i \) to the next customer point. Equation (29) represents the driving time of the vehicle from customer point \( i \) to customer point \( j \). Equation (30) represents the time when the vehicle arrives at customer point \( j \).

6. Problem Solving

6.1. Coding Strategy

In accordance with the characteristics of the problem, this paper used the natural number coding method and coding strategy of ant colony optimization, and the SC-AEFA for the cyclic iteration.

6.2. Solving Steps

On the basis of determining the coding strategy, the flow chart of the algorithm is shown in Figure 5.

![Figure 5. The flow chart of the algorithm.](image-url)
The specific steps of the algorithm are described as follows.

Step 1. Initialize the population and various parameters, obtain the position information of each node of the problem model, establish a coordinate system, and calculate the distance matrix;

Step 2. Update the population;

Step 3. Compare the cost of each path with the previous generation, and save the best path;

Step 4. If the algorithm reaches the maximum number of iterations, output the optimal solution; otherwise, go back to Step 2.

6.3. Example Verification

Taking a distribution business called fresh enterprise A in the Fuzhou Strait Agricultural and Sideline Products Trading Market as an example, the optimization problem of the vehicle distribution path was solved.

6.3.1. Basic Data

The model data used in this paper are shown in Tables 4 and 5 and Figure 6 including the specific information of the distribution vehicle and the geographic location of the customer points and distribution centers.

Table 4. The delivery vehicle model parameter table.

| Vehicle Model  | Curb Weight (kg) | Total Mass (kg) | Rated Load (kg) | Vehicle Start-Up Cost |
|----------------|------------------|-----------------|-----------------|-----------------------|
| NJL5033XXYBEV8 | 1900              | 3500            | 1500            | 120                   |

Table 5. The customer and distribution center node details.

| Distribution Center | X-Coordinate | Y-Coordinate | Demand | Left Time Window | Right Time Window | Service Time |
|---------------------|--------------|--------------|--------|-----------------|------------------|--------------|
| 0                   | 56.69        | 60.22        | 0      | 0               | 1236             | 0            |

| Customer Number | X-Coordinate | Y-Coordinate | Demand | Left Time Window | Right Time Window | Service Time |
|-----------------|--------------|--------------|--------|-----------------|------------------|--------------|
| 1               | 56.88        | 69.44        | 30     | 65              | 146              | 90           |
| 2               | 56.40        | 69.74        | 10     | 727             | 782              | 90           |
| 3               | 56.26        | 67.66        | 10     | 15              | 67               | 90           |
| 4               | 57.02        | 63.34        | 20     | 621             | 702              | 90           |
| 5               | 57.56        | 66.95        | 10     | 534             | 605              | 90           |
| 6               | 57.77        | 61.75        | 20     | 357             | 410              | 90           |
| 7               | 54.89        | 63.89        | 10     | 567             | 620              | 90           |
| 8               | 55.53        | 62.11        | 40     | 384             | 429              | 90           |
| 9               | 56.57        | 55.21        | 20     | 99              | 148              | 90           |
| 10              | 57.24        | 55.79        | 20     | 179             | 254              | 90           |
| 11              | 57.71        | 57.79        | 10     | 278             | 345              | 90           |
| 12              | 58.98        | 54.80        | 10     | 732             | 777              | 90           |
| 13              | 55.92        | 59.61        | 40     | 169             | 224              | 90           |
| 14              | 56.44        | 56.30        | 10     | 622             | 703              | 90           |
| 15              | 57.47        | 58.43        | 10     | 261             | 316              | 90           |
| 16              | 57.62        | 54.94        | 10     | 358             | 405              | 90           |
| 17              | 58.26        | 72.76        | 20     | 200             | 237              | 90           |
| 18              | 56.90        | 71.41        | 30     | 31              | 100              | 90           |
| 19              | 55.48        | 68.91        | 20     | 751             | 816              | 90           |
| 20              | 56.12        | 65.11        | 10     | 283             | 344              | 90           |
| 21              | 56.48        | 64.88        | 10     | 665             | 716              | 90           |
| 22              | 57.35        | 64.26        | 20     | 383             | 434              | 90           |
| 23              | 57.88        | 54.73        | 20     | 567             | 624              | 90           |
| 24              | 56.12        | 58.83        | 10     | 166             | 235              | 90           |
| 25              | 56.58        | 64.15        | 20     | 68              | 149              | 90           |
| 26              | 57.32        | 56.66        | 10     | 16              | 80               | 90           |
| 27              | 56.56        | 67.70        | 10     | 541             | 600              | 90           |
6.3.2. Analysis of Results

Under the premise of ensuring the same population size, dimension, number of iterations, and related parameters, the above VRPTW was solved by using the ACO algorithm, BOA, AEFA, and SC-AEFA. The results are described below.

It can be seen from Table 6 that the SC-AEFA obtained better results under the premise of less time, which shows that the SC-AEFA is superior to the other three algorithms in terms of both the computing speed and the optimization accuracy. In order to more clearly observe the convergence trend of the SC-AEFA and the other three algorithms, the algorithm iteration convergence graph is shown in Figure 7.

Table 6. The comparison of the results.

| Algorithm   | Required Vehicle | Algorithm Time | Cost    |
|-------------|------------------|----------------|---------|
| AEFA        | 4                | 87.60 s        | 559.35  |
| SC-AEFA     | 4                | 78.76 s        | 556.86  |
| BOA         | 4                | 80.68 s        | 561.36  |
| ACO         | 4                | 80.04 s        | 678.75  |

Figure 6. The distribution map of the customer points and distribution centers.

Figure 7. The comparison of the algorithm convergence effect. 

Model parameter settings: the speed of the delivery vehicle is \( v = 60 \) km/h; the driving cost per mileage is CNY 0.4; the penalty coefficient for being earlier than the time window is \( early = 0.05 \); the penalty coefficient for being later than the time window is \( late = 0.1 \); the penalty coefficient for violating the capacity constraint is \( \alpha = 100 \).
From the convergence comparison chart, it can be seen that the effect of the ant colony algorithm was the worst, and the convergence trend was very weak. The BOA was prone to premature convergence, resulting in it taking a long time to jump out of the local optimum, and the convergence accuracy was not high. Both the AEFA and SC-AEFA had better convergence trends, but the SC-AEFA had a significantly higher accuracy.

It can be seen from Table 7 that the optimal value, the worst value, and the average value obtained by the algorithm proposed in this paper were better than those of the other three algorithms. At the same time, in the 20 independent repeated experiments, the SC-AEFA algorithm’s earliest convergence iteration number was 23, which was the fastest among the four algorithms. Additionally, the optimal value obtained by the algorithm proposed was 556.12, while the theoretical value of the example was 553.81. In comparison, it can be found that the degradation degree of the algorithm’s performance was not obvious, which indicates that the optimization results of the algorithm proposed in this paper are effective in solving the VRPTW.

Table 7. The comparison of results.

| Algorithm | The Optimal Value | The Worst Value | The Average Value | The Earliest Convergence Number |
|-----------|-------------------|-----------------|-------------------|--------------------------------|
| AEFA      | 562.78            | 593.49          | 560.78            | 40                             |
| SC-AEFA   | 556.12            | 568.38          | 556.86            | 23                             |
| BOA       | 559.66            | 677.94          | 561.36            | 37                             |
| ACO       | 656.01            | 679.23          | 678.75            | 33                             |

7. Conclusions

In order to improve the defects of the AEFA, this paper proposed an improved artificial electric field algorithm (SC-AEFA) that integrates the sine cosine mechanism. First, the iterative formula of the original artificial electric field algorithm was improved to reduce the influence of the step size in the later iteration to highlight the importance of individual positions. In addition, a repulsive factor was introduced in the iterative process to keep away from the current worst individual. Second, some parameters of the sine cosine algorithm were improved, so that the sine cosine algorithm could better control the dynamic balance between the global search and local search. Finally, the improved sine cosine iterative formula was introduced into the artificial electric field algorithm, so that the algorithm can increase the scope of the global search and improve the ability of local optimization. Through the comparison experiment of the test functions, it was proven that the optimization accuracy of the algorithm proposed was significantly improved, and the stability had more significant advantages.

In addition, this paper took the vehicle routing problem with time window constraints as the research object and established the VRPTW with the vehicle cost, distance cost, and time window penalty cost as the objective function. This problem was solved by the ACO algorithm, BOA, AEFA, and SC-AEFA. The results showed that the optimization ability of the SC-AEFA was significantly improved compared to the AEFA, and that it had a certain robustness. The results obtained by the algorithm proposed are not the best solution, so there is still much room for improvement in the performance of the algorithm. The main problem we face is that we cannot find the best solution range in the global search, so we cannot find the best result in the local search. In the following research, based on this problem, strategies to enhance the global search will be added into the algorithm.

Future research can be further carried out from the following two aspects:

Methods for solving discrete problems such as path optimization and location selection can be studied in more depth, since the artificial electric field algorithm is more suitable for solving discrete problems. The improvement mechanism of this paper is relatively simple. The performance of the artificial electric field algorithm has not been fully explored, so the algorithm still has much room for improvement. Applying the improved artificial electric
field algorithm to the system scheduling problem can be carried out to further verify the performance of the algorithm in future studies.

**Author Contributions:** Conceptualization, H.Z.; Methodology, H.Z. and J.X.; Writing—original draft preparation, H.Z.; Writing—review and editing, J.X.; Software, H.C. and G.Y.; Validation, L.Z., J.X., L.Z. and J.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Guangxi Science and Technology Base and Special Talents Project under Grant No. 2021AC19436, the Key Laboratory of Software Engineering in Guangxi Minzu University under Grant No. 2022-18XJSY-03, the Talent Introduction Project of Guangxi Minzu University under Grant No.2020KJQD05, the Innovation Training Program for College Students under Grant No.202110608009, and the Research Foundation for Civil Aviation University of China (3122022PT02).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are included within the article.

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

**References**

1. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Zhou, Y.; Chen, H.; Deng, W. Quantum differential evolution with cooperative coevolution framework and hybrid mutation strategy for large scale optimization. *Knowl.-Based Syst.* 2021, 224, 107080. [CrossRef]
2. Li, X.; Zhao, H.; Yu, L.; Chen, H.; Deng, W.; Deng, W. Feature extraction using parameterized multi-synchrosqueezing transform. *IEEE Sens. J.* 2022, 1. [CrossRef]
3. Li, G.; Li, Y.; Chen, H.; Deng, W. Fractional-Order Controller for Course-Keeping of Underactuated Surface Vessels Based on Frequency Domain Specification and Improved Particle Swarm Optimization Algorithm. *Appl. Sci.* 2022, 12, 3139. [CrossRef]
4. Li, X.; Shao, H.D.; Jiang, H.K.; Xiang, J.W. Modified Gaussian convolutional deep belief network and infrared thermal imaging for intelligent fault diagnosis of rotor-bearing system under time-varying speeds. *Struct. Health Monit.* 2022, 21, 339–353.
5. Chen, H.; Zhang, Q.; Luo, J.; Xu, Y.; Zhang, X. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* 2019, 86, 105884. [CrossRef]
6. Cui, H.; Guan, Y.; Chen, H. Rolling Element Fault Diagnosis Based on VMD and Sensitivity MCKD. *IEEE Access.* 2021, 9, 120297–120308. [CrossRef]
7. Jin, T.; Xia, H. Lookback option pricing models based on the uncertain fractional-order differential equation with Caputo type. *J. Ambient Intell. Humaniz. Comput.* 2020, 1–14. [CrossRef]
8. Li, T.; Shi, J.; Deng, W.; Hu, Z. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* 2022, 121, 108731. [CrossRef]
9. Zhao, H.; Liu, J.; Chen, H.; Li, Y.; Xu, J.; Deng, W. Intelligent diagnosis using continuous wavelet transform and gauss convolutional deep belief network. *IEEE Sens. J.* 2022. [CrossRef]
10. Zhang, X.; Wang, H.; Du, C.; Fan, X.; Cui, L.; Chen, H.; Deng, F.; Tong, Q.; He, M.; Yang, M.; et al. Custom-Molded Offloading Footwear Effectively Prevents Recurrence and Amputation, and Lowers Mortality Rates in High-Risk Diabetic Foot Patients: A Multicenter, Prospective Observational Study. *Diabetes Metab. Syndr. Obes. Targets Ther.* 2022, 15, 103–109. [CrossRef]
11. Jin, T.; Xia, H.; Deng, W.; Li, Y.; Chen, H. Uncertain Fractional-Order Multi-Objective Optimization Based on Reliability Analysis and Application to Fractional-Order Circuit with Caputo Type. *Circuits Syst. Signal Process.* 2021, 40, 5955–5982. [CrossRef]
12. Deng, W.; Xu, J.; Gao, X.-Z.; Zhao, H. An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Trans. Man Cybern. Syst.* 2022, 52, 1578–1587. [CrossRef]
13. Holland, J.H. Genetic Algorithms and the Optimal Allocation of Trials. *SIAM J. Comput.* 1973, 2, 88–105. [CrossRef]
14. KirkPatrick, S.; Gelatt, C.; Vecchi, M. Optimization by simulated annealing. *Science* 1983, 220, 671–680. [CrossRef]
15. Brits, R.; Engelbrecht, A.P.; van den Bergh, F. Locating multiple optima using particle swarm optimization. *Appl. Math. Comput.* 2006, 189, 1859–1883. [CrossRef]
16. Yang, X.S.; Gandomi, A.H. Bat Algorithm: A Novel Approach for Global Engineering Optimization. *Eng. Comput.* 2012, 29, 464–483. [CrossRef]
17. Dorigo, M.; Maniezzo, V. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst.* 1996, 26, 29–41. [CrossRef] [PubMed]
18. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* 2015, 89, 228–249. [CrossRef]
19. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper Optimization Algorithm: Theory and application. *Adv. Eng. Softw.* 2017, 105, 30–47. [CrossRef]
Appl. Sci. 2022, 12, 6240

20. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. Soft Comput. 2018, 23, 715–734. [CrossRef]

21. Mirjalili, S. ACO: A Sine Cosine Algorithm for Solving Optimization Problems. Knowl.-Based Syst. 2016, 96, 120–133. [CrossRef]

22. Ran, X.; Zhou, X.; Lei, M.; Tepsan, W.; Deng, W. A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots. Appl. Sci. 2021, 11, 11202. [CrossRef]

23. Zheng, J.J.; Yuan, Y.; Zou, L.; Deng, W.; Guo, C.; Zhao, H. Study on a novel fault diagnosis method based on VMD and BLM. Symmetry 2019, 11, 747. [CrossRef]

24. He, Z.Y.; Shao, H.D.; Zhong, X.; Zhao, X.Z. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. Knowl.-Based Syst. 2020, 207, 106396. [CrossRef]

25. Tian, C.; Jin, T.; Yang, X.; Liu, Q. Reliability analysis of the uncertain heat conduction model. Comput. Math. Appl. 2022, 119, 131–140. [CrossRef]

26. Zhang, Z.-H.; Min, F.; Chen, G.-S.; Shen, S.-P.; Wen, Z.-C.; Zhou, X.-B. Tri-Partition State Alphabet-Based Sequential Pattern for Multivariate Time Series. Cogn. Comput. 2021, 1–19. [CrossRef]

27. Wu, X.; Wang, Z.C.; Wu, T.H.; Bao, X.G. Solving the Family Traveling Salesperson Problem in the Adleman–Lipton Model Based on DNA Computing. IEEE Trans. NanoBiosci. 2021, 21, 75–85. [CrossRef]

28. Dai, S.L.; He, S.; Wang, M.; Yuan, C. Adaptive neural control of underacted surface vessels with prescribed performance guarantees. IEEE transactions on neural networks and learning systems. IEEE Trans. Neural Netw. Learn. Syst. 2018, 30, 3686–3698. [CrossRef]

29. Zhou, Y.; Zhang, J.; Yang, X.; Ling, Y. Optimal reactive power dispatch using wave wave optimization algorithm. Oper. Res. 2020, 20, 2537–2553. [CrossRef]

30. Zhao, H.M.; Li, H.D.; Jin, Y.; Dang, X.J.; Deng, W. Feature extraction for data-driven remaining usefule life prediction of rolling bearings. IEEE Trans. Instrum. Meas. 2021, 70, 351910. [CrossRef]

31. Xu, Y.; Chen, H.; Luo, J.; Zhang, Q.; Jiao, S.; Zhang, X. Enhanced Moth-flame optimizer with mutation strategy for global optimization. Inf. Sci. 2019, 492, 181–203. [CrossRef]

32. Anita; Anupam, Y. AFEA: Artificial electric field algorithm for global optimization. Swarm Evol. Comput. 2019, 48, 93–108.

33. Zhang, W.B.; Su, Q.; Cheng, G.L. Vehicle routing problem with time windows based on dynamic demands. Ind. Eng. Manag. 2016, 21, 68–74.

34. DANTZIG, G.B.; RAMSER, J.H. The truck dispatching problem. Manag. Sci. 1959, 6, 80–91. [CrossRef]

35. Kim, G.; Ong, Y.S.; Heng, C.K.; Tan, P.S.; Zhang, N.A. City Vehicle Routing Problem (City VRP): A Review. IEEE Trans. Intell. Transp. Syst. 2015, 16, 1654–1666. [CrossRef]

36. Tao, W.H.; Zhao, C.C.; Sun, Y.B.; Liu, C.L.; Sun, Z.X.; Sun, Z. Research on vehicle routing problem based on improved dragonfly algorithm. Comput. Technol. Dev. 2020, 30, 170–175.

37. Li, Z.P.; Zhang, Y.W. Multiple demands vehicle routing problem with time windows and service order constraints. Control Dec. 2019, 34, 1565–1570.

38. Tas, D.; Dellaert, N.; van Woensel, T.; De Kok, T. The time-dependent vehicle routing problem with soft time windows and stochastic travel times. Transp. Res. Part C Emerg. Technol. 2014, 46, 66–83. [CrossRef]

39. Iqbal, S.; Kaykobad, M.; Rahman, M.S. Solving the multi-objective vehicle routing problem with soft time windows with the help of bees. Swarm Evol. Comput. 2015, 24, 50–60. [CrossRef]

40. Wang, K.; Choi, S.H.; Lu, H. A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows. Comput. Ind. Eng. 2019, 130, 75–96.

41. Wang, T.; Liu, J.X.; Wei, B.; Ye, G.Q.; Liu, H.; Li, J.J.; Zhu, D.L. Optimal operation and dispatch of water supply pump station in T city. Water Res. Pow. 2021, 39, 120–123+132.

42. Zhao, F.; He, Y. Routing algorithm of software defined internet of things based on artificial electric field optimization. Comput. Eng. Des. 2021, 42, 2725–2732.

43. Xu, S.J.; Long, W. Improved sine cosine algorithm for solving high-dimensional optimization problems. Appl. Res. Comput. 2018, 35, 2574–2577.

44. Yong, L.Q.; Li, Y.H.; Jia, W. Literature survey on research and application of sine cosine algorithm. Comput. Eng. Appl. 2020, 56, 26–34.

45. Xuan, H.N.; Miao, C.L.; Zhao, D. System-level fault diagnosis based on bat algorithm. Comput. Eng. Sci. 2016, 38, 640–647.

46. Lin, J.; He, Q. Fusion sine cosine and mutation selection grasshopper optimization algorithm. J. Chin. Mini-Micro. Comput. Syst. 2021, 42, 706–713.

47. Li, X.Y. Chaotic gravitational constants for the artificial electric field algorithm. Micro. Appl. 2021, 37, 60–62,70.

48. Zhong, K.; Zhou, G.; Deng, W.; Zhou, Y.; Luo, Q. MOPMA: Multi-objective marine predator algorithm. Comput. Methods Appl. Mech. Eng. 2021, 385, 114029. [CrossRef]

49. Wei, Y.Y.; Zhou, Y.Q.; Luo, Q.F.; Deng, W. Optimal reactive power dispatch using an improved slime mould algorithm. Energy Rep. 2021, 7, 8742–8759. [CrossRef]

50. Deng, W.; Zhang, X.X.; Zhou, Y.Q.; Liu, Y.; Zhou, X.B.; Chen, H.L.; Zhao, H.M. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. Inf. Sci. 2022, 585, 441–453. [CrossRef]