International Journal of Innovative
Computing, Information and Control
Volume **21**, Number **6**, December **2025**

ICIC International ©2025 ISSN 1349-4198
pp. 1655–1669

RESEARCH ON UAV PATH PLANNING BASED ON NSGAP IN WIRELESS CHARGING NETWORK INSPECTION

Ran Quan[1], Feng Sun[1], Xiaoping Hu[1], Miao Yu[1], Jinxu Wang[2] and Peiwen Yu[3,*]

[1]State Grid Changchun Power Supply Company
No. 1206, Tonghua Road, Nanguan District, Changchun 130000, P. R. China
{ 445333133; 80989997; 327157000; 49834833 }@qq.com

[2]Shenyang Jiuzhou Feixiang Technology Co., Ltd.
No. 24-4, Shenbei Road, Shenbei New District, Shenyang 110122, P. R. China
81666688@qq.com

[3]College of Software
Jilin University
No. 2699, Qianjin Street, Changchun 130000, P. R. China
*Corresponding author: 956477751@qq.com

Received March 2024; revised July 2024

Abstract. *In the context of smart city construction, the automatic monitoring of urban power grids using wireless sensor networks has become a hot research and application focus. When transmitting signals, these networks typically convey only simple data due to distance limitations and signal interference. Consequently, in recent years, UAVs (Unmanned Aerial Vehicles) have been employed to periodically gather detailed information generated by sensor network nodes. This study initially establishes a model for the wireless sensor network and employs a greedy algorithm to cluster signal nodes for forming UAV signal coverage zones. Utilizing index arrays for multi-point crossover expands the search space, while adaptive mutation and crossover operators are introduced to avert getting trapped in local optima. An improvement to the NSGA-II (Non-dominated Sorting Genetic Algorithm II) algorithm, named NSGAP (Non-dominated Sorting Genetic Algorithm Plus), is developed herein. Through simulation experiments, the NSGAP algorithm is applied to scenarios of varying inspection scales and compared with other conventional algorithms. The experimental outcomes demonstrate that the refined NSGAP algorithm outperforms others, effectively addressing the issue of efficiency in urban power grid inspection.*

Keywords: Urban power grids, Wireless sensor networks, Path planning, NSGA-II algorithm, Multi-UAV collaboration

1. **Introduction.** In major cities and large industrial areas, electrical power is transmitted and distributed to users via the power grid. Urban power grid refers to the system that provides electricity supply and distribution for urban areas, which consist of transmission lines, distribution lines, substations, and distribution stations. Given China's vast geographical span – approximately 50 degrees latitude and 62 degrees longitude – the construction of urban power grids faces challenges from complex environments, urban development, weather conditions, and human factors, leading to wear and tear over time. Therefore, regularly inspecting urban power grid nodes is a crucial task to ensure stable and reliable power supply [1]. Several solutions have been proposed for urban power grid

DOI: 10.24507/ijicic.21.06.1655

inspection, primarily including manual inspection, robotic inspection, UAV (Unmanned Aerial Vehicle) inspection [2], and wireless sensors.

Wireless sensors can collect a large amount of real-time data from the power grid. However, due to cost considerations, they often only transmit important information back to the signal receiving base in real time. Therefore, UAVs can be used to periodically inspect the wireless sensor network, obtaining comprehensive and complex data on power grid operation over a period. Due to varying urban conditions, UAVs can simultaneously collect node information within multiple signal reception areas from high altitudes, facilitating operator tasks and becoming a method for periodically collecting node signals on a large scale.

The key issue in using UAVs for inspection is path planning, which can be divided into static and dynamic path planning and has been widely applied in various fields. Path planning is a typical multi-objective optimization problem, where multiple objectives cannot simultaneously achieve optimal status under the same conditions, necessitating the search for relatively optimal solutions. Metaheuristic algorithms, particularly Genetic Algorithms (GA) based on evolutionary theory, excel in solving multi-objective optimization problems with strong global search capabilities and maintaining solution diversity. This study focuses on using an improved genetic algorithm for static path planning.

This study aims to use multiple UAVs to collaborate in the signal reception tasks of a fixed-scale urban power grid, collecting comprehensive data from each node over a period while minimizing the number of UAVs, ensuring thorough node inspection, and achieving the shortest path and lowest total cost. Each UAV can cover multiple nodes, forming a dynamic mapping from sparse to dense point sets within a specific time frame. This study does not consider obstacle avoidance due to high-rise buildings in cities, and the positions of wireless sensor nodes are known and fixed.

Given that UAVs have a certain signal reception coverage range, the study first combines greedy and clustering algorithms to optimize the network, selecting suitable intermediate signal reception positions among nodes so that a single UAV can cover multiple wireless sensor nodes within a specific time. The study proposes an NSGAP algorithm based on the second generation of Non-dominated Sorting Genetic Algorithm (NSGA-II). The main technical contributions of this proposed method are summarized as follows:

1) Proposing adaptive multi-point crossover and mutation operators to simulate the impact of the environment on the population and the mutual influence between populations due to crowding, adaptively changing the crossover and mutation probabilities of gene loci to prevent premature convergence to local optima;

2) Treating the order of individual gene loci as genetic information and introducing the concept of an index array for multi-point crossover to expand the search space;

3) Conducting simulation experiments by simulating the distribution of urban sensors, performing multiple rounds of comparative and ablation experiments in a simulated environment, and analyzing the experimental results.

2. Research Background.

In recent years, the application of UAVs in industrial, military, and commercial fields has been increasing, making the research on methods and technologies for UAV path planning increasingly important. Over the past few decades, many researchers have proposed different UAV path planning methods. To ensure the safety and efficiency of UAVs, path planning needs to consider many factors, such as weather conditions, flight altitude, obstacles, and fuel consumption.

Currently, UAV path planning algorithms are mainly divided into two categories: path planning based on traditional algorithms and path planning based on intelligent algorithms. Traditional path planning algorithms mainly include the A* algorithm, the

Dijkstra algorithm, and the artificial potential field method. Traditional algorithms often have large computational demands and complex processes when solving multi-dimensional problems, while path planning based on intelligent optimization algorithms can obtain better results in a shorter time.

The origin of intelligent optimization algorithms lies in simulating biological interactions or physical phenomena. These are a series of approximate optimization algorithms [3]. Compared with traditional algorithms, they have greater advantages in handling complex optimization problems. However, the most basic intelligent optimization algorithms have disadvantages such as easily falling into local optima, slow convergence speed, and poor convergence accuracy when solving path planning problems. Therefore, many scholars have improved basic intelligent optimization algorithms when addressing UAV path planning issues [4]. To solve the UAV path planning problem in a three-dimensional flight environment with obstacles, Mullen et al. [5] proposed a Hybrid Differential Symbiotic Organism Search (HDSOS) algorithm and introduced the concepts of traction function and perturbation strategy in the algorithm, which respectively improve the efficiency and robustness of the algorithm. Wu and Zhang [6] proposed a Golden Eagle Optimizer with Dual Learning Strategies (GEO-DLS). The dual learning strategies include personal example learning and mirror reflection learning, which respectively enhance the search capability and convergence accuracy of the GEO algorithm. Liu et al. [8] proposed a PSO algorithm to solve the path optimization problem in a complex three-dimensional environment. The chaotic strategy and Cauchy-Gaussian mutation strategy were introduced into the original sparrow search algorithm, which respectively enhance the population diversity and anti-stagnation of the algorithm.

The genetic algorithm is a search algorithm that simulates the process of natural selection and genetics, commonly used to solve optimization problems. It has also been widely used to solve path planning problems in recent years [8-12]. The basic idea is to simulate the evolutionary process in nature, performing operations such as gene coding, selection, crossover, and mutation on individuals to evolve better individuals generation by generation, ultimately finding the optimal solution. Since the genetic algorithm uses stochastic operations, it may fall into local optimum, especially in large search spaces.

There are various optimization algorithms within genetic algorithms, among which NSGA and NSGA-II are multi-objective optimization algorithms. NSGA is the Non-dominated Sorting Genetic Algorithm, and NSGA-II is the Non-dominated Sorting Genetic Algorithm II with elitism, which retains the advantages of NSGA and has the following improvements: it greatly reduces the complexity of the problem, allowing the algorithm to converge to the optimal solution set faster; it uses crowding distance and crowding operators, greatly enhancing the operability and versatility of NSGA-II; NSGA-II adopts an elitism strategy, which effectively preserves high-quality solutions generated during iterations, significantly improving robustness and convergence efficiency.

3. Improved Multi-Objective Genetic Algorithm.

This paper uses a genetic algorithm to solve the UAV path planning problem, aiming to find the optimal signal reception path and determine the optimal number of UAVs when dealing with cooperative signal reception tasks involving multiple UAVs.

3.1. NSGA-II algorithm.

NSGA-II algorithm is a multi-objective optimization design computation using Pareto optimal solutions [14]. In the following, the principle and main process of NSGA-II algorithm are introduced. First determine the Pareto dominance relation: for the minimization multi-objective optimization problem, for n objective components $f_i(x)$, $i = 1, \ldots, n$. Given any two decision variables X_a, X_b, suppose the following two conditions hold, it is called X_a dominate X_b, as shown in Equations (1) and (2).

$$\text{For } \forall i \in 1, 2, \ldots, n,\ f_i(X_a) <= f_i(X_b). \tag{1}$$

$$\text{For } \exists i \in 1, 2, \ldots, n,\ f_i(X_a) < f_i(X_b). \tag{2}$$

A decision variable is said to be non-dominated if no other decision variable dominates it. Pareto rank: In a set of solutions, the Pareto rank of the non-dominated solution is defined as 1, the non-dominated solution is deleted from the set of solutions, the Pareto rank of the remaining solution is defined as 2, and so on, the Pareto rank of all solutions in the set of solutions can be obtained. A fast dominance sort is then performed, assuming that the population size is P. The algorithm needs to calculate two parameters: n_p dominated by each individual p and the set S_p of solutions dominated by this individual. Traversing the entire population, the computational complexity of this parameter is $O(mN^2)$. In order to make the obtained solution more uniform in the objective space, crowding degree is introduced here n_d:

Set $n_d = 0$, $n \in 1, \ldots, N$, for each objective function:

1) Rank the individuals at this level according to the objective function, denote $f_m^{\max}$ as the maximum and $f_m^{\min}$ as the minimum;
2) The crowding degrees 1_d and n_d of the two boundaries after sorting are set to ∞;
3) $n_d = n_d + (f_m(i+1) - f_m(i-1)) \big/ \left(f_m^{\max} - f_m^{\min}\right)$.

From the perspective of the two-objective optimization problem, it is just like the sum of the side lengths of the maximum moment (the rectangle cannot touch other points in the objective space) that the individual can generate in the objective space.

In general, it is not possible to make all objectives optimal at the same time. Therefore, the ideal situation is to make the one or two objectives with the largest weight best, and the other objectives as far as possible, so as to achieve the overall optimum. First, objective functions need to be constructed, each of which generates Pareto solution sets. Suppose the optimization problem has M objective functions and N decision objectives, i.e.,

$$\begin{cases} y = f(x) = [f_1(x), f_2(x), \ldots, f_m(x)] \\ x = (x_1, x_2, \ldots, x_N) \end{cases} \tag{3}$$

As shown in Equation (3), x is a decision variable composed of N decision objectives; y is the target vector.

On the basis of the above, the algorithm further adopts the elite retention strategy: The parent population C_i and offspring population D_i are first synthesized into population R_i. Generate a new parent population C_{i+1} from the R_i. Arrange the entire population in the parent population C_{i+1} in order of Pareto levels from low to high. Until at some level the individuals of the layer cannot all be placed in the parent population. Place individuals from each level in the parent population C_{i+1} in a round-robin fashion, ordered from highest to lowest crowding distance, until the parent population C_{i+1} is filled.

When using real-valued encoding, use SBX (Simulated Binary Crossover) to simulate binary crossover. It follows from Equations (4) and (5):

$$X_{1j}(t) = 0.5 \times [(1 + r_j)\, X_{1j}(t) + (1 - r_j)\, X_{2j}(t)] \tag{4}$$

$$X_{2j}(t) = 0.5 \times [(1 - r_j)\, X_{1j}(t) + (1 + r_j)\, X_{2j}(t)] \tag{5}$$

where

$$r_j = \begin{cases} (2u_j)^{\frac{1}{\eta+1}} & u_j < 0.5 \\ 1/(2(1-u_j))^{\frac{1}{\eta+1}} & \text{else} \end{cases} \tag{6}$$

Perform polynomial mutation:

$$X_j(t+1) = X_j(t) + \Delta j \tag{7}$$

Among them, Δj indicates in the j dimensions, the amount of variation between generation t to $t+1$.

3.2. **NSGAP algorithm.** The crossover strategy of NSGA-II algorithm, the chromosome gene fragments are often used as the crossover objects, but in the encoded genetic information, the order of gene loci is also a part of the genetic information. In the path planning problem, the order of the path points often affects the quality of the final result.

The multi-UAV inspection problem is actually the combination of grouping problem and sorting problem. That is, different inspection points are assigned to different UAVs. At the same time, the order of waypoints affects the total driving distance, thus affecting the total cost. Therefore, the crossover strategy in this paper is to regard the total order of waypoints and the UAV assignment results as independent genetic information. In the process of crossover, the path order of one parent generation is used as the index, and the sorting within the group is carried out according to the UAV path group of the other parent generation, that is, to ensure a relative order.

This paper uses real number coding, that is, m paths formed by n UAVs are encoded into a one-dimensional array with the inspection center (0 point) as the interval. Taking nine inspection points as an example, an array of 0-10 is generated. Assuming that three UAVs are used for distribution, through initialization of the population and subsequent selection operations, the parents selected are as follows.

0	1	3	0	4	2	7	8	9	0	5	6	0

0	2	7	3	0	1	4	5	0	6	0	8	9	0

FIGURE 1. Encoding diagram

Figure 1 depicts the motion paths of multiple UAVs, with each complete UAV path starting and ending at 0.

Therefore, based on the real number coding, this paper generates an index chromosome for one of the chromosomes of the parent generation. In the implementation, it is shown that for the array with 0 as the interval, the order of the waypoints other than 0 is extracted.

1	3	4	2	7	8	9	5	6

FIGURE 2. Index array diagram

For the other parent, the index array in the figure above is sorted to ensure that the local paths are relatively ordered within each UAV group.

0	3	2	7	0	1	4	5	0	6	0	8	9	0

FIGURE 3. Illustration of the sorting results

At the same time, this paper changes the specific strategy in the crossover and mutation process. The NSGA-II algorithm uses crowding distance to assess population density, but relying solely on crowding distance can sometimes lead to situations where crowding distance and Pareto levels do not match, causing unfair reproduction in the population.

Therefore, at this stage in the process, this paper first compares an individual's crowding distance with the average crowding distance of individuals in its Pareto frontier. This serves as a foundation for introducing cumulative classification ranking in subsequent operations. An iterative factor 'i' is introduced, and the crossover and mutation probabilities are determined. After this, the population can evolve and reproduce more freely within these constraints, which we refer to as population adaptive evolution. The crossover probability is denoted as P_c and the mutation probability as P_m:

$$P_c \begin{cases} p_{c,agv} + \dfrac{G-i}{G}\left(p_{c,agv_{j_{agv}}{}_{c,\max}}\right) \\ p_{c,agv} & , \text{ if } d_j(i) = d_{agv}(i) \\ p_{c,agv} - \dfrac{i}{G}\left(p_{c,agv} - p_{j_{agv}{}_{c,\min}}\right) \end{cases} \tag{8}$$

$$P_m \begin{cases} p_{m,agv} + \dfrac{G-i}{G}\left(p_{m,agv_{j_{agv}}{}_{m,\max}}\right) \\ p_{m,agv} & , \text{ if } d_j(i) = d_{agv}(i) \\ p_{m,agv} - \dfrac{i}{G}\left(p_{m,agv} - p_{j_{agv}{}_{m,\min}}\right) \end{cases} \tag{9}$$

$p_{c,agv} = \frac{p_{c,\max}+p_{c,\min}}{2}$ is the average crossover probability; $p_{m,agv} = \frac{p_{m,\max}+p_{m,\min}}{2}$ is the average mutation probability; $p_{c,\max}$, $p_{c,\min}$ are the maximum and minimum crossover probabilities, respectively, $p_{m,\max}$, $p_{m,\min}$ are the maximum and minimum mutation probabilities. G is the total number of iterations; i is the current iteration number; $d_j(i)$ is the jth individual in the current population of the ith generation; $d_{agv}(i)$ is the average crowding distance of the current population. Individuals with a crowding distance smaller than the average crowding distance have lower crossover and mutation probabilities, and vice versa.

During the iteration process, not only the crossover and mutation probabilities between individuals differ, but also the crossover and mutation probabilities between generations should vary. A common scenario is that the periodic changes in the environment can cause changes in the biological clocks of organisms, which in turn affect their genetic expression. For example, diurnal variations can affect the biological clocks of many organisms, thereby regulating their metabolic activities, behavioral patterns, and so on. This change may affect the genetic expression patterns of organisms, thereby influencing their genetic characteristics. Here, trigonometric functions are used to simulate the impact of periodic changes on biological genetics.

$$cxp = \frac{r}{1+\frac{n}{e^n}} + u(1-r)\sin n \tag{10}$$

$$mup = \begin{cases} (1-cxp)t, & cxp < 0.5 \\ t \times rand(\tau-\varepsilon, \tau+\varepsilon), & cxp \geq 0.5 \end{cases} \tag{11}$$

where cxp and mup represent the final mutation rates for each generation. r, t, τ, ε, and u represent disturbance parameters:

$$r = p_{c,agv} + rand(-\varepsilon, \varepsilon) \tag{12}$$

$$u = r \times rand(\tau-\varepsilon, \tau+\varepsilon) \tag{13}$$

$$t = p_{m,agv} + rand(-\varepsilon, \varepsilon) \tag{14}$$

ε is set to 0.05 in this study. τ is set to 0.6. Calculate according to this formula. During the initial period of population update, the evolutionary individuals have a high probability of crossover and mutation, and during the update period, the probability of crossover and mutation of the evolutionary individuals improves the overall search ability of the algorithm. Therefore, in the later stages of iteration, the crossover probability and mutation probability tend to stabilize.

3.3. **Algorithm embedding and application.** Firstly, to cluster sensor network nodes, we use a clustering algorithm. Each point in the set may simultaneously appear within the coverage area of a patrol zone. Here, we define that each point in the set is a neighbor to one another. The first step of the algorithm is to select a coordinate point based on the greedy algorithm's idea, following the principle of maximum neighbors. After identifying the neighbor nodes for each node, they are assigned to the nearest coordinate point. To ensure that the number of neighbors in each node set Z is maximized, a patrol cluster is established with the coordinate point as the center of the circle, and a patrol cluster is denoted by N_i.

The specific procedure is as follows.

Algorithm 1. Clustering algorithm application

Input: Wireless sensor coordinate point collection N, maximum reception distance R
Output: Signal receiving nodes collection C
1 Define the input N (wireless sensor coordinate points or urban power grids nodes).
2 Initialize an empty set C to store signal receiving nodes.
3 For each node i in N, find its neighbor nodes within the maximum distance R (Euclidean distance must not exceed the radius of the signal receiving area). Store the neighbors in set Z_i.
4 Repeat the following steps until N is empty:
5 Identify the node i with the largest neighbor set Z_i.
6 Form a cluster N_i using node i and its neighbors Z_i.
7 Remove the nodes in N_i from N.
8 Collect the coordinates of signal reception points within cluster N_i and save them as new signal receiving nodes in C.
9 After updating N, determine if N is empty.
10 If N is not empty, return to step 3.
11 Output the set C of signal receiving nodes.

Then apply the signal receiving nodes collection C to NSGAP, the steps are as Algorithm 2.

Algorithm 2. NSGAP path planning

Input: Signal receiving nodes collection C
Output: Optimal paths P
1 Initialize UAV paths in random order according to C.
2 Calculate cost and constraint values for the initial population. Use fast non-dominated sorting to determine Pareto levels and crowding distances.
3 Selection, crossover, and mutation: Create a new population and generate offspring through these operations.
4 Calculate objective and constraint values for offspring, and determine their Pareto levels and crowding distances.
5 Form a new parent population by combining parents and offspring, selecting elite individuals until the total number of survivors equals n.
6 Repeat steps 2-5 until reaching the termination condition.
7 Output the optimal paths P.

4. Model Construction.

4.1. Definition of relevant parameters. Firstly, the relevant parameters of the model are defined in this paper, as shown in Table 1.

TABLE 1. Related parameters and symbolic meanings

Parameter notation	Meanings
$G = \{N\}$	The whole sensor network
$N = \{0, 1, 2, \ldots, n\}$	The set of sensor nodes in the whole sensor network, $i = 0$ represents the initial position of the UAV
$C = \{0, 1, 2, \ldots, c\}$	The set of signal receiving points in the whole sensor network, $i = 0$ represents the initial position of the UAV
$P = \{(i, j) \mid i, j \in N, i \neq j\}$	The path set of each UAV
(x_i, y_i)	Geographic location coordinates of node i
R	UAV signal coverage radius
$M = \{1, 2, 3, \ldots, m\}$	UAV collection
F_m	The initial cost of the UAV
Q_i	The total amount of information transmitted by node i (bit)
u	Signal reception rate
d_{ij}	The travel distance between any two signal receiving points i, j, $i, j \in C$ (km)
E^d	UAV total distance energy consumption emissions (W/Km)
e_m^d	UAV m bit distance energy consumption emissions (W/Km)
E^t	UAV total time energy consumption emissions (W/h)
e_m^t	UAV m bit time energy consumption emissions (W/h)
t_{ijm}	UAV m receives the time from reception point i to j (h)
v	The running speed of the UAV, always maintains a constant speed (km/h)
S	Total cost ($)
s	Unit energy cost ($)
$B_{\max}$	UAV flight battery capacity
T_m	The total time from departure to return of the UAV m (h)

The foundation of UAV path planning lies in environmental modeling, typically encompassing terrain modeling, obstacle, and threat zone modeling. Assuming there are no other threats to flight in the terrain, indicating a favorable environment, the UAVs maintain a consistent flight altitude, and fly at a uniform velocity. As their altitude remains constant, the three-dimensional model can be simplified to two dimensions. In the subsequent model construction, UAV will be used to represent the UAVs.

The UAV's flying altitude is set as H, and its flying speed is always maintained as V. Here, we define N as the collection of n sensor nodes, $N = \{n_i \mid i = 1, 2, 3, \ldots, n\}$. Each sensor node at position n_i ($n_i \in N$) is located at (x_i, y_i). Let M be the collection of m UAVs, $M = \{m_i \mid i = 1, 2, 3, \ldots, m\}$. The initial position of each UAV is $(0, 0)$.

The optimization objectives for the multiple UAV system are both the UAV power consumption, denoted as $E(P)$, and the number of UAVs, m. The signal receiving area for the UAVs is donated as R, and it is assumed that there are z nodes covered within R.

Use Algorithm 1 to cluster the N power grid nodes, input the N power grid nodes and the reception range R, forming the UAV signal receiving nodes collection C. Use Algorithm 2 to process C, thereby generating the final optimal paths P.

4.2. **The UAV path model.** The UAV's starting and ending points are both $(0, 0)$. After starting from the initial point, the UAV traverses the signal receiving area along a predefined path, sequentially inspecting the electrical wireless sensor nodes in the area, and then returns to the endpoint. Each UAV carries a certain flight battery capacity, and in this problem, there is a maximum flight time for the UAV denoted as $T_{\max}$. This represents the maximum duration the UAV's battery can sustain its flight. Knowing this maximum duration allows us to determine the UAV's maximum total cost.

Therefore, the first condition to be satisfied is that the total energy consumption of a single UAV is less than the capacity of its carried battery, as shown in Equation (15):

$$B_{\max} \geq E^d + E^t \tag{15}$$

At the same time, for any UAV m, its distance energy consumption and time energy consumption E^d, E^t. In the process of signal reception, the total cost of the UAV is the energy consumption required for its own flight. This energy consumption is mainly related to flight distance and flight time, where hovering for signal reception is counted in the time-of-flight energy consumption. UAV flight distance energy consumption is as follows:

$$E^d = \sum_{i=1,j=1}^{n} e_m^d d_{ij} \tag{16}$$

In this mission, the UAV performs overlay signal reception on the surrounding nodes at the signal receiving point, so the hovering time at a signal receiving point is the time required for information transmission of the surrounding node with the largest amount of information. That is, the required time and energy consumption:

$$E^t = \sum_{i=1,j=1}^{n} \left(\frac{\max(Q_i)}{u} + t_{ijm} \right) e_m^t \tag{17}$$

That is, the total flight time of the UAV is

$$T_m = \frac{\max(Q_i)}{u} + t_{ijm} \tag{18}$$

Therefore, the total cost is the total energy consumption cost of each UAV in flight time and flight distance, plus the initial cost of the UAV itself:

$$S = mF_m + \sum_{m=1}^{m} s \left(E_m^d + E_m^t \right) \tag{19}$$

Therefore, we need to find the UAV optimal path satisfying the minimum total cost, i.e.,

$$S = \min \left(P_{E_m^d} + P_{E_m^t} + P_{F_u} \right) \tag{20}$$

For n nodes, set a set c of signal transmission nodes with a total of C. The path of the UAV is planned based on the signal transmission nodes, and the signal transmission point position is planned based on the signal coverage radius of the UAV. The signal coverage radius of the UAV is R, and it is set that there are z nodes under the coverage of R.

For C, there is $C = \{C_1, C_2, C_3, \ldots, C_i | 0 < i \leq c\}$, for each transmitting node, its overlay node can be expressed as $C_i = \{N_1, N_2, \ldots, N_j | 0 < j \leq n\}$.

Use $P = \{0, i, \ldots, j, C + 1\}$, $i, j \in C$ to show UAV path, when the UAV passes through nodes i and j, it needs t_{ij} as its time consumption:

$$t_{ij} = \frac{d_{ij}}{v}, \quad \forall i, j \in A \tag{21}$$

5. **Simulation Experiments and Analysis.** The simulation parameters are set as shown in Table 2.

TABLE 2. Simulation parameter setting table for single UAV path planning

Experimental parameter	Parameter values	Experimental parameter	Parameter values
Signal reception radius	500 m	P	30 dBm
e_m^t	2 J	H	10 m
V	6 m/s	Crossover probability	0.8
$B_{\max}$	15 W	Mutation probability	0.06

Suppose there are a total of m UAVs available for the signal reception of 100 nodes, as shown in Figure 4. These 100 nodes are still scattered over a 50 × 50 square kilometer grid. Initially, a clustering algorithm is used to group all nodes, assuming a UAV signal reception radius of 5 kilometer, resulting in 21 clusters denoted as N_i.

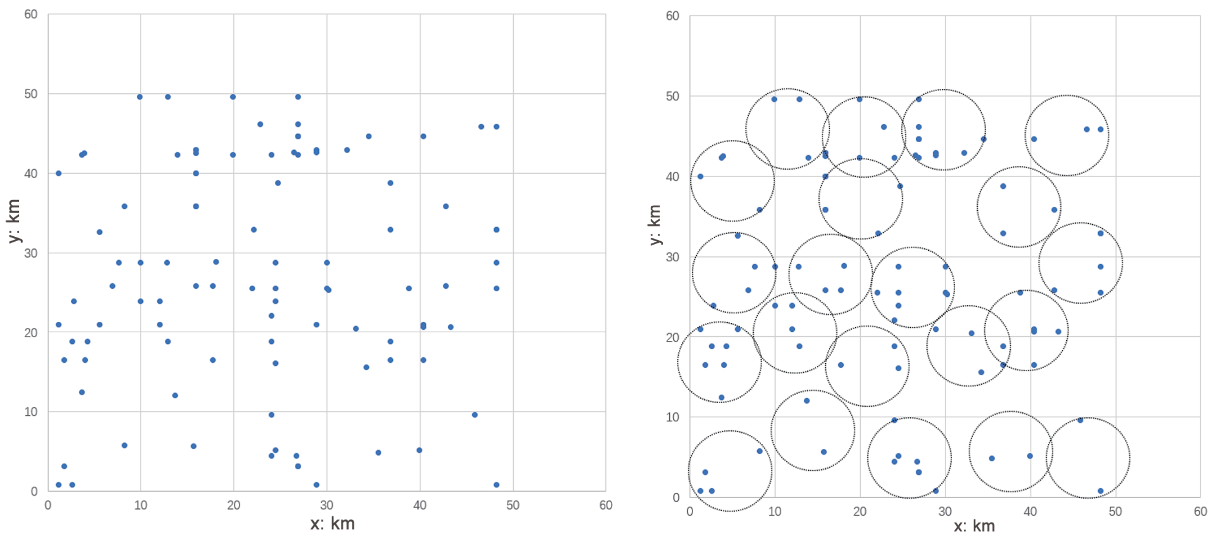

FIGURE 4. Location of 100 nodes clustering

Suppose the algorithm's crossover and mutation probabilities are fixed, with a crossover probability of 0.5 and a mutation probability set to 0.55. These numerical values are equal to the average crossover probability $p_{c,agv}$, and average mutation probability $p_{m,agv}$ of the traditional algorithm.

Figures 5 and 6 show that when NSGAP algorithm is applied to multi-UAV path planning, Figure 5 shows the optimal path map for a group of simulations, with different colors representing different UAVs. In Figure 6, it can be seen that NSGAP converges faster and achieves stronger optimization effects in practical applications compared to NSGA-II.

Comparative experiment. The CEC2022 optimization function test set has a total of 12 single-objective test functions with boundary constraints (similar to the CEC2013, CEC2014, CEC2017, and CEC2020 optimization function test sets). These are unimodal functions (F1), multimodal functions (F2-F5), hybrid functions (F6-F8), and combined functions (F9-F12). The test dimensions are 2, 10 and 20, which is less than CEC2013, CEC2014 and CEC2017, and the same as CEC2020 optimized function test set.

All the test functions are solving minimization problems.

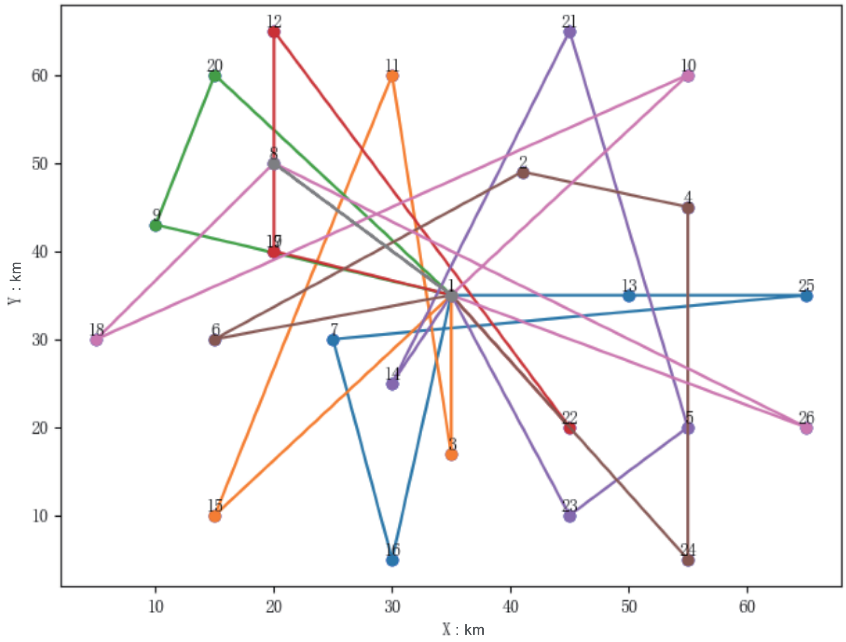

FIGURE 5. NSGAP algorithm for multi-UAV path planning

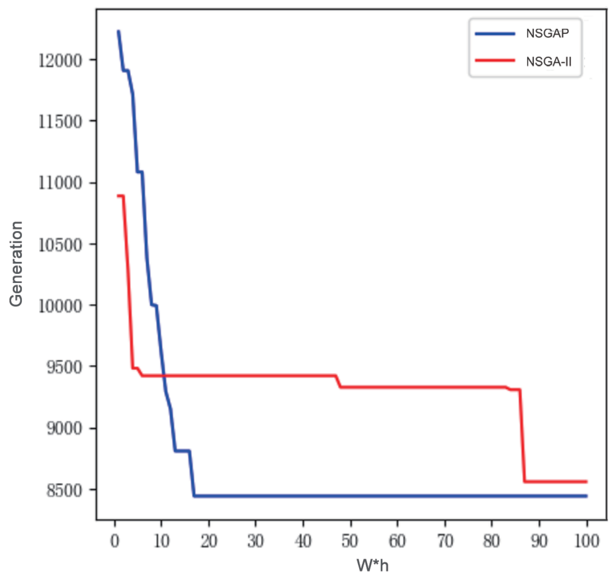

FIGURE 6. Multi-UAV fitness score of NSGA-II and NSGAP

In the comparison experiments, the initial population number of all algorithms is set to 50, test dimension is 20, and the number of iterations is 100.

In Table 3, we can see that NSGAP achieved the best result in 6 out of 12 test functions and the second-best result in 3 test functions, indicating that it performs the best among these algorithms in the minimization problems of CEC2022. Compared with other algorithms, combining with Figure 7, we can see that NSGAP and NSGA-II have some similarities in the convergence curve, with both achieving optimal or near-optimal solutions multiple times. It is evident that in minimization problems, these two algorithms perform better than others, and compared to NSGA-II, NSGAP more frequently achieves the lowest objective function value. The adaptive adjustment of generational mutation rate and crossover rate makes the whole algorithm less likely to fall into local optimal

TABLE 3. CEC2022 test scores

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**F12**
woa	24042.29	743.40	0.15	1107.0	908.32	1817130.60	2485.96	6034.89	3024.60	2040.40	10320.75	3615.92
aro	11107.23	539.54	0.029	**930.51**	**902.07**	5616621.97	2172.64	5124.26	2710.39	2817.85	4173.37	3039.23
goa	24043.47	779.34	0.058	939.32	903.64	368193.57	10385.70	3795.42	3678.54	2436.05	3809.06	4073.22
ssa	15467.79	918.82	0.59	1085.08	903.83	129256120.17	2202.23	3966.89	4026.29	2606.08	4119.98	3061.34
zoa	15576.34	971.40	0.36	1265.77	908.17	**50202.06**	2154.61	4020.18	2894.74	1821.91	2728.86	3173.05
NSGAP	**2315.41**	**485.95**	**0.000047**	936.00	905.75	152535.73	**1896.54**	6040.17	**2563.60**	2805.89	2903.86	**2900.00**
NSGA-II	4163.36	468.50	0.00099	950.29	906.88	340375.92	1961.81	**2995.52**	2574.49	**976.32**	**2625.53**	2913.29

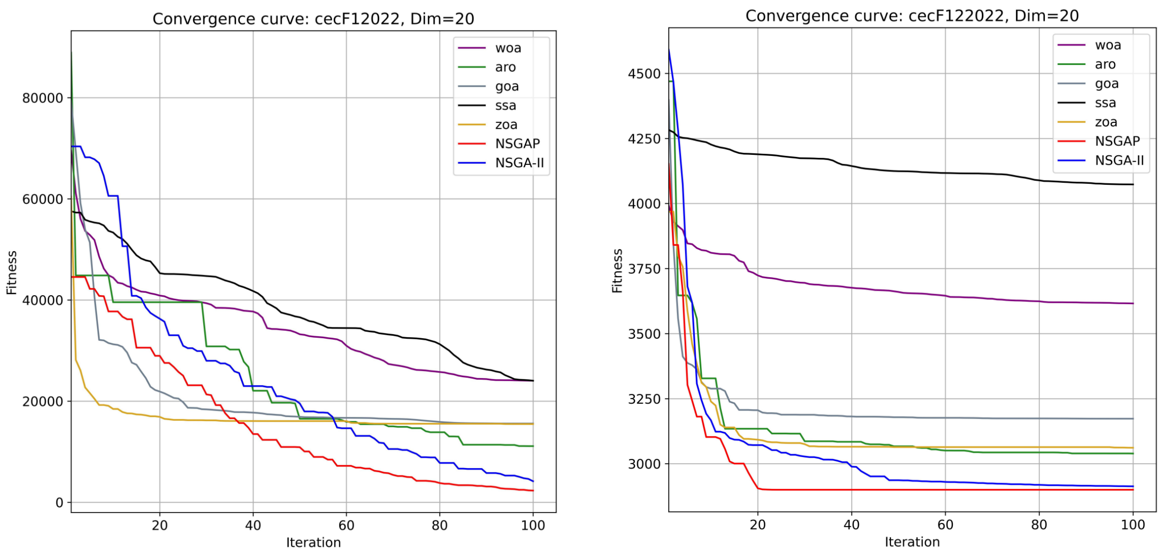

FIGURE 7. (color online) CEC2022 optimization function test result

solutions. The complete genetic information crossover ensures the expansion of the search space.

From Figure 7, it can be seen that NSGAP also has a faster convergence speed. The faster iteration speed in experiments may stem from improved crossover and mutation strategies, which allow the algorithm to converge more quickly to the vicinity of local optimal solutions.

6. **Summary.** This study models wireless sensor networks to simulate UAVs' regular inspections of these networks. The NSGA-II algorithm was improved by adding an adaptive evolution strategy and enhancing the operators, which to some extent improved the algorithm's speed, efficiency, stability, and accuracy.

Simulation results demonstrate that the enhanced NSGA-II algorithm enhances the optimization of the algorithm, accelerates convergence speed, significantly improves the accuracy of convergence direction, and better solves the problem of trajectory planning for multiple UAVs. Furthermore, future research will focus on operation selection, continuous optimization of other communication channels, continuous improvement of algorithm efficiency, and emphasis on time and space constraints in trajectory planning under more complex conditions.

When intelligent optimization methods are used to solve path planning model problems, related methods such as ant colony algorithm, particle snail optimization method, and genetic algorithm are used. In recent years, in-depth research and intensive training of artificial intelligence methods such as decision optimization (TSP, VRP, etc.) have performed well, enabling fast path planning. Reinforcement learning has been used in several studies under the framework of multi-agent planning with promising results. However, existing research on dynamic programming is not ideal, planning multiple routes without considering adjustments and limiting collaboration time and space.

Acknowledgment. This research is funded by Science and Technology Project of State Grid Jilin Electric Power Co., Ltd. No. SGJLCC00KJJS2302568.

REFERENCES

[1] L. Tang, Y. Han, A. S. Zalhaf et al., Resilience enhancement of active distribution networks under extreme disaster scenarios: A comprehensive overview of fault location strategies, *Renewable and Sustainable Energy Reviews*, vol.189, 113898, 2024.

[2] F. Deng, Y. Shan, Z. Xie et al., Unmanned aerial vehicle power inspection offloading strategy based on game theory and reinforcement learning, *Power System Technology*, vol.45, no.9, pp.3649-3657, 2021.

[3] M. Samadi and M. F. Othman, Global path planning for autonomous mobile robot using genetic algorithm, *2013 International Conference on Signal-Image Technology & Internet-Based Systems*, pp.726-730, 2023.

[4] A. Tuncer and M. Yildirim, Dynamic path planning of mobile robots with improved genetic algorithm, *Computers & Electrical Engineering*, vol.38, no.6, pp.1564-1572, 2022.

[5] R. J. Mullen, D. Monekosso, S. Barman et al., A review of ant algorithms, *Expert Systems with Applications*, vol.36, no.6, pp.9608-9617, 2019.

[6] H. S. Wu and F. M. Zhang, Wolf pack algorithm for unconstrained global optimization, *Mathematical Problems in Engineering*, vol.2022, 2022.

[7] K. L. Hoffman, M. Padberg and G. Rinaldi, Traveling salesman problem, *Encyclopedia of Operations Research and Management Science*, no.1, pp.1573-1578, 2023.

[8] Y. Liu, X. Zhang, Y. Zhang et al., Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach, *Chinese Journal of Aeronautics*, vol.32, no.6, pp.1504-1519, 2019.

[9] X. Fu and Y. Hu, Improved particle swarm optimization based three-dimensional path planning, *Optoelectronics & Control*, vol.28, no.3, pp.86-89, 2021.

[10] C. Yang, Y. Pei and P. Liu, Research on 3D spatial path planning using an improved particle swarm algorithm, *Computer Engineering and Applications*, vol.55, no.11, pp.117-122, 2019.

[11] Y. Xie and Y. Kong, Improved particle swarm optimization for 3D path planning, *Computer Measurement & Control*, vol.30, no.3, pp.179-182+191, DOI: 10.16526/j.cnki.11-4762/tp.2022.03.030, 2022.

[12] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, vol.6, no.1, pp.58-73, 2002.

[13] X. Ma and H. Yang, Greenhouse monitoring and control system with multi-objective control algorithm based on NSGA-II, *Practical Rural Technology*, no.1, pp.28-31, 2019.

[14] Z. Han, S. Wang, X. Dong and X. Ma, Improved NSGA-II algorithm for multi-objective scheduling problem in hybrid flow shop, *Proc. of 2017 9th International Conference on Modelling, Identification and Control (ICMIC2017)*, pp.815-820, 2017.

Author Biography

Ran Quan holds a Master's degree in Engineering, awarded by Northeast Electric Power University, China in June 2020. The individual has contributed to academia with a publication in the "Journal of Power Systems and Their Automatic Control" and an additional paper presented at an EI-indexed international conference. Since joining State Grid Changchun Power Supply Company, China in August 2020, he has specialized in the operation and maintenance of transmission lines. Presently, he is serving in the Construction Department of State Grid Changchun Power Supply Company, China, continuing his focus on the research and practice of transmission line operation and maintenance.

Feng Sun graduated from the School of Business and Economics of Jilin University, China in 1994. He has been working at the Transmission Operation and Maintenance Center of State Grid Changchun Power Supply Company, China since 1995, with a research focus on transmission line operation and maintenance.

Xiaoping Hu graduated from Northeast Electric Power University, China in 2007. He employed at the Transmission Operation and Maintenance Center of State Grid Changchun Power Supply Company, China from 2007 to present. Research direction includes transmission line operation and maintenance.

Miao Yu graduated from Northeast Electric Power University, China in 2006. He employed at the Transmission Operation and Maintenance Center of State Grid Changchun Power Supply Company, China from 2006 to present. Research direction includes transmission line operation and maintenance.

Jinxu Wang graduated from Jilin University with a major in Electronic Science and Engineering, China in 2014. He employed at Shenyang Jiuzhou Feixiang Technology Co., Ltd., China from 2018 to present. Research direction includes aerospace equipment research and development, and promotion of drone technology services.

Peiwen Yu graduated with a Bachelor's degree in Software Engineering and a concurrent Bachelor's degree in Japanese from Jilin University, China in 2022. Currently he is a postgraduate student in the Software Engineering program at the College of Software, Jilin University, China. He has published one first-author EI conference paper and one second-author EI conference paper. Research direction includes data mining.