algorithms

MDPI

Article

A Boosted Electromagnetic Wave Propagation Algorithm for Path Planning of Welding Manipulators in Complex Multi-Workpiece Scenarios

Chaochuan Jia [1,2], Feilong Yu [1], Xingyu Gao [1], Yaqi Yang [1], Han Xu [1], Maosheng Fu [1,2] and Yu Liu [1,*]

[1] School of Electronic Information and Artificial Intelligence, West Anhui University, Lu'an 237012, China; 03000076@wxc.edu.cn (C.J.); 2025210033@mail.wxc.edu.cn (F.Y.); 2025210019@mail.wxc.edu.cn (X.G.); 2025210032@mail.wxc.edu.cn (Y.Y.); 2024011392@mail.wxc.edu.cn (H.X.); fums@wxc.edu.cn (M.F.)
[2] Anhui Province Intelligent Hydraulic Machinery Joint Construction Subject Key Laboratory, Lu'an 237012, China
* Correspondence: 03000136@wxc.edu.cn

Abstract

To address the problems of the Electromagnetic Wave Propagation Algorithm (EMWPA)—insufficient initial-population coverage, an imbalance between exploration and exploitation, and a tendency to fall into local optima—in high-dimensional complex optimization problems, this paper proposes a boosted electromagnetic wave propagation optimization algorithm, BEMWPA. First, a cubic chaotic map is introduced in the population-initialization stage to enhance the uniformity of the initial-solution distribution and the search-space coverage. Second, nonlinear phase modulation is applied to the electric- and magnetic-field driving terms, and a differentiated probabilistic switching mechanism is constructed to improve the dynamic coordination between global exploration and local exploitation. Furthermore, a Beta-distribution opposition-based learning strategy is introduced to enhance the algorithm's ability to escape local optima by generating high-quality opposite candidate solutions. To verify the effectiveness of the proposed algorithm, systematic comparative experiments are conducted on the CEC2017 benchmark function set, and BEMWPA is combined with rapidly-exploring random tree (RRT) and applied to path planning of a welding manipulator in complex multi-workpiece scenarios. For a three-dimensional welding scenario containing 12 workpieces, 12 closed weld seams, and multiple obstacle constraints, BEMWPA-RRT reduces the initial inter-seam transfer path length of RRT from 586.00 mm to 479.11 mm, representing a relative reduction of 18.24%, and the complete end-effector path length is reduced from 2974.00 mm to 2867.11 mm, representing a relative reduction of 3.59%. Meanwhile, the optimized transfer path length is only 1.59 mm longer than the obstacle-free ideal transfer length of 477.52 mm, indicating that the proposed method can approach the geometric lower bound of this scenario while satisfying the obstacle-avoidance constraints. Kinematic verification on a seven-degrees-of-freedom welding manipulator further shows that the optimized Cartesian-space path can be converted into a continuously executable joint-space trajectory, providing an effective method for offline welding path planning of complex multi-workpiece tasks.

Keywords: electromagnetic wave propagation algorithm; BEMWPA; rapidly-exploring random tree; seven-degrees-of-freedom manipulator; multi-workpiece welding; path planning

Check for updates

Academic Editor: Juvenal Rodriguez-Resendiz

Received: 9 July 2026
Revised: 7 August 2026
Accepted: 8 August 2026
Published: 10 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Optimization problems are widespread in fields such as engineering design, path planning, and intelligent decision-making. As task scales and constraints continue to grow, many practical problems exhibit characteristics such as high dimensionality, nonlinearity, multimodality, and strong coupling. Traditional optimization methods often rely heavily on strong prior assumptions or gradient information, and they tend to suffer from slow convergence and stagnation in local optima within complex search spaces [1–3]. For path planning of welding manipulators in complex multi-workpiece scenarios, the planning result must not only satisfy path-length and obstacle-avoidance requirements but also account for the weld-seam visiting order, trajectory smoothness, joint-space executability, welding-torch pose constraints, and search stability in complex environments. This places higher demands on both the global exploration and local exploitation capabilities of the optimization algorithm [4].

The Electromagnetic Wave Propagation Algorithm (EMWPA) is a recently proposed physics-inspired optimization algorithm [5]. By simulating the propagation and coupling behaviors of the electric and magnetic fields, the algorithm guides individuals to update their positions in the search space and features a simple structure, few parameters, and strong search directionality. However, the original EMWPA adopts random initialization, which can easily lead to insufficient initial-population coverage, and its electromagnetic-field driving terms are sensitive to parameter variations, which may cause step-length fluctuation and search oscillation in complex problems. Meanwhile, when population diversity drops rapidly in the middle and late iterations, the algorithm lacks an effective population-reconstruction mechanism and tends to fall into local optima, reducing convergence accuracy. To solve the above problems, this paper proposes a multi-strategy improved electromagnetic wave propagation optimization algorithm, BEMWPA. First, a cubic chaotic map is introduced in the initialization stage to improve the uniformity of the initial population and the search-space coverage [6,7]. Recently, deterministic-chaos-based mechanisms have also been explored for low-cost and finely tunable intelligent space coverage, such as algorithms built on the Courbage–Nekorkin artificial neuron model, which further motivated the chaotic initialization adopted here [8]. Second, nonlinear phase modulation is applied to the electric- and magnetic-field driving processes, combined with a differentiated probabilistic switching mechanism, to enhance the dynamic balance between global exploration and local exploitation. Finally, a Beta-distribution opposition-based learning strategy is introduced in the population-update stage to generate opposite candidate solutions according to the current population distribution, improving the algorithm's ability to escape local optima [9,10]. These strategies act collaboratively within a unified framework, thereby improving the convergence accuracy, search stability, and path-planning applicability of the algorithm.

Taking path planning of welding manipulators in complex multi-workpiece scenarios as the application target, this paper builds an optimization model that comprehensively considers path length, weld-seam visiting order, obstacle avoidance, trajectory smoothness, and joint-motion constraints, and solves it using BEMWPA. Through comparative experiments with algorithms such as PSO and ALA, and with reference to the application experience of PSO-type methods in manipulator trajectory optimization research [11–13], the effectiveness of the proposed algorithm is verified in terms of path length, convergence speed, planning success rate, and algorithm stability.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the basic principle of EMWPA and the improvement mechanisms of BEMWPA. Section 4 describes the CEC2017 benchmark tests and the path-planning experiments for welding manipulators in complex multi-workpiece scenarios and analyzes the results. Section 5 summarizes the results and discusses future research directions.

2. Related Work

At present, the commonly used methods for manipulator path planning mainly include teaching and offline programming methods, graph-search and sampling-based planning methods, and intelligent optimization algorithms. Teaching and offline programming methods can ensure trajectory executability fairly well, but they depend heavily on manual experience. When the workpiece type, fixturing position, or number of weld seams changes, re-teaching or extensive trajectory adjustment is often required, making it difficult to meet the flexibility demands of complex multi-workpiece welding tasks. Graph-search and sampling-based planning methods, such as the A* algorithm, the probabilistic roadmap method, and the rapidly-exploring random tree, can search for feasible paths in obstacle environments, but they usually focus more on the reachability between a start and a goal point. Consequently, they have limited ability to comprehensively optimize the multi-seam visiting order, the overall idle-travel length, and multi-objective constraints. Moreover, the generated paths may contain many redundant nodes and insufficient smoothness [4,14–18].

In recent years, metaheuristic optimization algorithms have been widely applied to complex problems such as continuous optimization, engineering design, and path planning because they do not rely on gradient information, are highly adaptable, and are simple to implement. Inspired by natural laws, biological behaviors, and collective collaboration, researchers have proposed representative methods such as Particle Swarm Optimization (PSO), the Whale Optimization Algorithm (WOA), the Sparrow Search Algorithm (SSA), the Multi-Verse Optimizer (MVO), Cuckoo Search (CS), and the Artificial Lemming Algorithm (ALA), and they have gradually applied them to manipulator path planning and trajectory optimization [11,12,19–22]. Wu et al. applied PSO [23] to manipulator trajectory planning and proposed the RTPPSO method, which optimizes the joint angles and S-shaped velocity-profile parameters through adaptive weights and random perturbation, enabling the manipulator to obtain a smoother motion trajectory in a shorter time while improving acceleration variation and operating stability. Du et al. [24] addressed problems such as non-uniform motion and low efficiency in traditional manipulator trajectory planning by introducing an improved WOA. They used reinforcement learning to adjust the inertia weight and combining variable-neighborhood search to enhance local exploitation, thereby solving the time-optimal trajectory and providing a more continuous, smooth, and efficient motion process. Zhou et al. [25] applied an improved SSA to trajectory planning for the energy-consumption optimization of welding robots. They constructed joint-space trajectories with seventh-order B-splines and combined elite opposition-based learning, non-dominated sorting, and Gaussian–Cauchy mutation to solve the optimal time series, reducing energy consumption while guaranteeing trajectory continuity. Liu et al. [26] addressed the insufficient trajectory-planning efficiency of six-degrees-of-freedom industrial manipulators by introducing an improved MVO and building a trajectory optimization model with time, energy consumption, and jerk as objectives. They improved the global search capability through adaptive parameter adjustment and population mutation, thereby enhancing operating efficiency and trajectory quality. Zhang et al. [27] addressed the time-optimal trajectory planning of six-degrees-of-freedom serial manipulators by introducing the Adaptive Cuckoo Search (ACS) algorithm. By combining inverse-kinematics solving with quintic B-spline trajectory construction, they optimized the total motion time under strict dynamic constraints and verified the convergence speed, solution accuracy, and trajectory smoothness of the algorithm through simulation and experiments. Existing metaheuristic optimization algorithms have achieved good results in manipulator path planning and trajectory optimization. However, in complex multi-workpiece welding scenarios, they remain susceptible to factors such as the large number of weld seams, complex obstacle constraints, strong joint-space constraints, and high search dimensionality. Consequently, they still suffer from problems such as declining population diversity, stagnation in local optima, and insufficient late-stage convergence accuracy.

3. BEMWPA

The search framework of EMWPA draws on the mechanism of electromagnetic wave propagation [5,7], driving the evolution of individuals in the solution space by characterizing the periodic coupling of the electric and magnetic fields. Electromagnetic waves, formed through the mutual excitation of time-varying electric and magnetic fields, exhibit sinusoidal oscillation characteristics and can propagate in different directions. This physical process provides a natural framework for constructing search trajectories with directionality and periodicity. On this basis, BEMWPA employs multiple strategies acting jointly: the differentiated roles of the electric and magnetic fields correspond to global exploration and local exploitation, respectively, and periodic modulation dynamic adjusts the search direction and step length.

3.1. Cubic Chaotic Initialization

It should be noted that chaotic initialization is a supporting component adapted to EMWPA rather than a claimed novelty: the cubic map is employed only to improve the uniformity and diversity of the initial population, thus providing a better starting point for the subsequent field-driven search. To improve the distribution uniformity and diversity of the initial population in the search space, this paper introduces a cubic chaotic map mechanism in the population-initialization stage. Compared to traditional random-initialization methods, chaotic maps possess good ergodicity, nonperiodic, and sensitivity to initial values, and they can generate more uniformly distributed candidate solutions, thereby enhancing the global exploration capability of the algorithm in the early stage [6]. The cubic map is expressed by Equation (1).

$$x_{k+1} = 2.595 x_k \left(1 - x_k^2\right) \tag{1}$$

where $x_k \in (0,1)$ is the chaotic variable generated at the k-th iteration, and 2.595 is the control parameter, whose value directly affects the chaotic characteristics of the mapped sequence. In this specific implementation, an initial chaotic variable is first randomly generated within the interval (0,1), and then the above mapping relationship is used to recurse dimension by dimension to construct a chaotic-sequence matrix of size $N \times dim$. Subsequently, the chaotic sequence is linearly mapped onto the problem domain [lb, ub] as the positions of the initial population.

This strategy can effectively alleviate the individual-clustering phenomenon caused by random initialization and improve the population's coverage of the search space, thereby providing a more representative initial solution set for subsequent iterations. The cubic map is adopted here as a simple and widely used chaotic mechanism rather than an optimal one. Compared to the pseudo-random uniform initialization of the original EMWPA, it produces an ergodic, non-repetitive sequence intended to improve the diversity of the initial population, and it serves only as a supporting component. Maps with more uniform value-range coverage, such as the recently proposed fractal tent map, may bring further improvement and are left for future work.

3.2. Exploration and Exploitation Phases

3.2.1. Square-Root Nonlinear Phase-Modulation Strategy

Unlike generic phase- or step-scaling operators that are applied externally to a search algorithm, the modulation proposed here is embedded directly into the electric- and magnetic-field cosine driving terms of the electromagnetic-wave model and constitutes the core methodological novelty of BEMWPA. In the original EMWPA, the position update of an individual is driven by the electric-field component E_y (whose update model is given by Equation (2)) and the magnetic-field component B_z (whose update model is given by

Equation (3)) during electromagnetic wave propagation. The electric and magnetic fields are used to describe the dynamic variation of an individual when it is attracted by the target position during the search, and their intensities directly affect the update magnitude. The original electromagnetic propagation model can be expressed as follows:

$$E_y = E_{\max}\cos(\omega t - \varphi),\ \varphi = 2\frac{\pi}{\lambda} + \pi \tag{2}$$

$$B_z = B_{\max}\cos(\omega t - \varphi),\ \varphi = 2\frac{\pi}{\lambda} + \pi \tag{3}$$

where $E_{\max}$ and $B_{\max}$ denote the maximum amplitudes of the electric and magnetic fields, respectively; ω is the angular frequency; t is the current iteration number; φ is the phase-offset parameter; and λ is the wavelength parameter. In this study, the wavelength parameter is set to $\lambda = 4000$ ($\lambda > 0$), so the phase offset $\varphi = 2\pi/\lambda + \pi \approx 3.14$ always lies in the interval $(\pi, +\infty)$. Consequently $\varphi > 1$ holds strictly, the square-root term $\sqrt{\varphi} \approx 1.77$ is always well-defined, and $\sqrt{\varphi} < \varphi$ holds throughout the search.

It can be seen that the original EMWPA adopts a linear phase offset φ in the electromagnetic-field computation. This form can preserve the periodic-variation characteristics of the electromagnetic field and provide dynamic perturbation for the population search through the cosine function. However, in a complex high-dimensional search space, the linear phase offset adjusts the field-strength response rather directly. When the electric or magnetic field is in a large-response interval, an individual may produce a large position-update magnitude. By contrast, when the population gradually approaches a good region, an excessively strong field attraction may cause the search trajectory to oscillate, making the local-exploitation process insufficiently stable. Therefore, while the original linear phase-adjustment approach maintains search activity, it may also introduce step-length fluctuation and unstable late-stage convergence.

To address the above problems, this paper performs a nonlinear reconstruction of the phase term of the electromagnetic propagation model in BEMWPA. The improved electric- and magnetic-field expressions are given by:

$$E_y = E_{\max}\cos(\omega t - \sqrt{\varphi}) \tag{4}$$

$$B_z = B_{\max}\cos(\omega t - \sqrt{\varphi}) \tag{5}$$

This improvement does not change the basic periodic structure of the electromagnetic-field model, nor does it change the angular frequency ω. Instead, it recalibrates the phase offset. Since $\varphi > 1$ under usual conditions, we have $\sqrt{\varphi} < \varphi$. The square-root mapping can compress the action intensity of the original phase offset, transforming the electromagnetic-field response from linear-phase driving into nonlinear phase modulation. Through this treatment, the variation in the electric and magnetic fields during iterations becomes smoother, the abruptness in the individual position-update process is weakened, and the continuity of the search trajectory is thereby improved. It should be noted that, because the phase enters through a bounded cosine, this improvement does not originate from the cosine itself. Rather, reducing the phase-offset magnitude from φ to $\sqrt{\varphi}$ changes the effective sampling of the field-driving oscillation over the finite search horizon, thereby reducing the iteration-to-iteration fluctuation of the population step length. This effect is quantified in Section 3.2.2.

After computing the electric-field and magnetic-field components, BEMWPA determines through a probabilistic switching mechanism which electromagnetic-field component the current individual adopts for updating. Let the switching probability be SP; in this paper it is set to $SP = 0.3$.

In each iteration, for the i-th individual, a random number $r \in [0,1]$ is generated. When $r <$ SP, the individual is updated using the electric-field component E_y; otherwise, the magnetic-field component B_z is used for the position update. The update process can be expressed as follows:

$$X_i^{(t+1)} = X_i^t + \eta E_y(X_{target}^t - X_i^t), r < SP \quad (6)$$

$$X_i^{(t+1)} = X_i^t + \eta \mathrm{B_Z}(X_{target}^t - X_i^t), r \geq SP \quad (7)$$

where ${X_i}^t$ denotes the position of the i-th individual at the t-th iteration; ${X^t}_{target}$ denotes the current target position; and η is a random scaling factor, whose value range is $\eta \in [0.7,1]$ In the program implementation, η acts on each dimension of the individual in vector form, giving different dimensions differentiated update magnitudes, thereby enhancing the flexibility of the search process.

Since $SP = 0.3$, an individual adopts electric-field-driven updating with a probability of about 30% and magnetic-field-driven updating with a probability of about 70% in the expected sense. This setting means that the algorithm does not allocate the two search modes in equal proportion but is dominated by magnetic-field-driven updating while retaining a certain proportion of electric-field-driven search. Combined with the amplitude difference between the electric and magnetic fields, this probabilistic switching approach can maintain the search-perturbation capability while keeping the population-update process relatively stable.

Compared to the original EMWPA, BEMWPA reduces the direct influence of the linear phase offset on the field-strength response through square-root nonlinear phase modulation, making the driving of the individual position update by the electric and magnetic fields smoother. Meanwhile, the $SP = 0.3$ probabilistic switching mechanism further controls the action frequency of the two types of electromagnetic-field components during the update process, enabling the algorithm to maintain the necessary perturbation capability during the search while avoiding trajectory instability caused by an excessively high proportion of large updates. Through this improvement, the algorithm can form a more stable transition between global search and local exploitation. This helps alleviate the step-length fluctuation, local oscillation, and premature convergence of the original EMWPA in high-dimensional complex optimization problems, thereby improving the overall convergence accuracy and search stability.

3.2.2. Effectiveness Analysis of Square-Root Nonlinear Phase Modulation

To verify the impact of square-root nonlinear phase modulation on EMWPA performance, this paper constructs three forms of phase modulation for comparative analysis. Among them, BEMWPA introduces square-root phase terms into the electric-field and magnetic-field components, namely E_y and B_z; PEMWPA retains the unmodulated phase term φ; and TEMWPA adopts a squared phase term φ^2. Except for the phase term, the three algorithms keep the same population size, search framework, parameter settings, and boundary handling, so this comparison can directly reflect the impact of different phase-modulation methods on algorithm performance.

The average-ranking results in Figure 1 verify the effectiveness of the square-root phase-modulation strategy. As shown in Figure 1, the average rank of BEMWPA is 1.76, which is lower than 2.17 for PEMWPA and 2.07 for TEMWPA. This indicates that BEMWPA using the $\sqrt{\varphi}$ phase factor outperforms the comparison algorithms PEMWPA using φ and TEMWPA using φ^2 in terms of overall performance. Compared to PEMWPA and TEMWPA, the average rank of BEMWPA is reduced by about 18.89% and 14.98%, respectively, indicat-

ing that square-root nonlinear phase modulation can more effectively improve the overall optimization performance of the algorithm across different test functions.

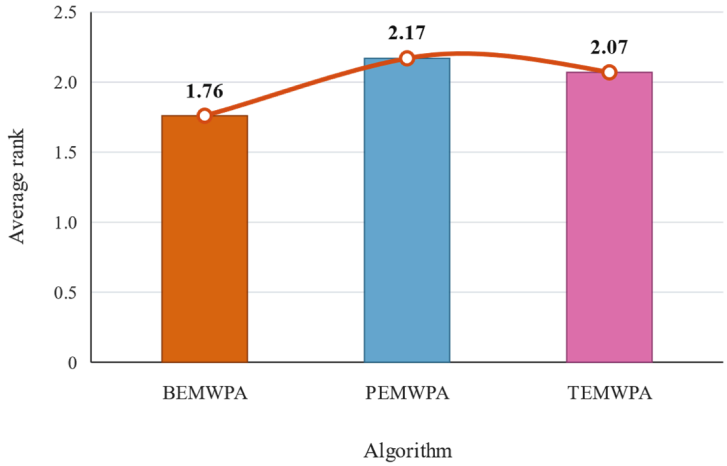


Figure 1. Average-ranking chart (BEMWPA is the proposed algorithm).

From the perspective of the phase-modulation mechanism, PEMWPA uses φ as the phase-adjustment factor. Its phase variation is linear, the search step length is rather directly affected by the original phase, and directional fluctuation easily occurs in a complex search space. TEMWPA uses φ^2 as the phase-adjustment factor, which further compresses the phase variation. Although this can attenuate oscillation to some extent, it may also reduce the exploration activity of the population. By contrast, BEMWPA uses $\sqrt{\varphi}$ for nonlinear phase modulation, which optimizes the effective phase distribution of the $cos(\cdot)$ term without changing the basic structure of the electromagnetic-field driving term, giving the individual-update process both search stability and direction-adjustment capability. Therefore, BEMWPA can better coordinate global exploration and local exploitation, thereby obtaining better average-ranking results.

To provide quantitative evidence for the smoother search behavior, the per-iteration fluctuation of the population step length is further measured for the three phase forms under identical settings (population size 30, 1000 iterations, dimension 50, and 30 independent runs) on a set of representative benchmark functions. Three indicators are used, for which a smaller value indicates a smoother update process: the standard deviation of the population step length, the roughness of the step-length increments (the standard deviation of successive differences), and the trajectory-smoothness index (the mean absolute second difference of the step-length sequence). Averaged over the benchmark set, the $\sqrt{\varphi}$ form (BEMWPA) attains the smallest value on all three indicators. Relative to the φ form (PEMWPA) and the φ^2 form (TEMWPA), it reduces the trajectory-smoothness index by 21.4% and 11.6%, the increment roughness by 11.3% and 12.7%, and the step-length standard deviation by 6.4% and 12.7%, respectively. These results quantitatively confirm that the square-root nonlinear phase modulation yields a smoother and more stable position-update process, consistent with the average-ranking advantage reported above.

3.3. Beta-Distribution Opposition-Based Learning Strategy

Different from the standard opposition-based learning that generates a fixed opposite solution ($x' = lb + ub - x$), the strategy proposed here is an adaptive, distribution-based generalization in which the Beta shape parameters and the boosted/stable modes are dynamically governed by the real-time population diversity and by dynamic boundaries. To enhance the population-reconstruction capability of BEMWPA in middle and late stages, this paper introduces a Beta-distribution opposition-based learning strategy

after the electric- and magnetic-field-driven updates. This strategy generates opposite candidate solutions based on the dynamic distribution range of the current population, so that opposition-based learning no longer relies on fixed global boundaries but performs adaptive perturbation according to the real-time search state. Compared to ordinary opposition-based learning, the Beta distribution can adjust the offset direction and perturbation intensity of candidate solutions through its shape parameters, thereby improving the effectiveness of the opposite candidate solutions.

Let the population at instant t be denoted as P(t), where the i-th individual is denoted as X_i(t), whose j-th dimension position is given by $X_{i,j}$(t). d^j_{max} and d^j_{min} denote the dynamic maximum and minimum of the current population along the j-th dimension, respectively. The opposite candidate solution generated based on the Beta distribution is expressed as follows:

$$\widetilde{x}_{i,j}(t) = (d^j_{max}(t) - d^j_{min}(t)) \cdot Beta(\alpha, \beta) + x_{i,j}(t) \tag{8}$$

where $\widetilde{x}_{i,j}(t)$ is the opposite candidate position generated for the i-th individual in the j-th dimension; Beta(α,β) denotes a random variable following the Beta distribution; and α and β are its shape parameters. This formula controls the generation range of the opposite candidate solution through the dynamic boundaries of the current population, allowing the opposite search to proceed around the current effective search region.

In the boosted-perturbation mode, the two shape parameters of the Beta distribution are jointly determined by the diffusion factor s, the peak-adjustment parameter $p_{i,j}$, and the relative-position parameter $m_{i,j}$:

$$\alpha = \begin{cases} s \cdot p_{i,j}, & m_{i,j} < 0.5 \\ s, & otherwise \end{cases} \tag{9}$$

$$\beta = \begin{cases} s, & m_{i,j} < 0.5 \\ s \cdot p_{i,j}, & otherwise \end{cases} \tag{10}$$

where $m_{i,j}$ describes the relative position of the current individual within the dynamic boundaries. When $m_{i,j} < 0.5$, α and β are assigned such that the Beta distribution is biased toward one side; when $m_{i,j} \geq 0.5$, the action direction of the parameters switches, so that the opposite candidate solution can form differentiated perturbation according to the individual's position. This treatment avoids all individuals adopting the same perturbation direction and helps maintain search diversity within the population.

The diffusion factor s is adaptively controlled by the current population diversity D_{pop}(t):

$$s = \left(\frac{1}{\sqrt{D_{pop}(t)}} \right)^{1+N(0,0.5)} \tag{11}$$

where N(0,0.5) denotes a Gaussian random variable with mean 0 and variance 0.5. From the relationship in the formula, when the population diversity is low, D_{pop}(t) is small and the diffusion factor increases accordingly, thereby enhancing the perturbation magnitude of the opposite candidate solution. When the population still maintains a good distribution, the perturbation intensity is relatively weakened to avoid destroying the existing good search region.

The peak-adjustment parameter $p_{i,j}$ is calculated as follows:

$$p_{i,j} = \begin{cases} \frac{(s-2)m_{i,j}+1}{s(1-m_{i,j})}, & m_{i,j} < 0.5 \\ \frac{2-s}{s} + \frac{s-1}{s \cdot m_{i,j}}, & otherwise \end{cases} \tag{12}$$

This parameter is used to further adjust the shape of the Beta distribution, so that the generation of opposite candidate solutions is controlled not only by the diffusion factor

but also by the current relative position of the individual. Through the piecewise setting of $p_{i,j}$, the algorithm can form different probability-distribution shapes in different position regions, thereby improving the diversity of opposite candidate solutions.

In the boosted-perturbation mode, the relative position is computed using the opposite relative position $m_{i,j}$:

$$m_{i,j} = \frac{(d^j_{max}(t) + d^j_{min}(t) - x_{i,j}(t)) - d^j_{min}(t)}{d^j_{max}(t) - d^j_{min}(t)} \tag{13}$$

This expression is equivalent to mapping the current individual $p_{i,j}(t)$ to the symmetric opposite position within the dynamic boundaries. In this way, BEMWPA can generate candidate solutions on the relatively opposite side of the current search region, thereby increasing the chance for the population to escape the local-optimum region.

To judge the degree of aggregation of the current population, the normalized population diversity needs to be computed:

$$D_{pop}(t) = \frac{1}{N}\sum_{i=1}^{N} CD(X_i(t), P(t)) \tag{14}$$

$$CD(X_i(t), P(t)) = \min_{C \in P(t), C \neq X_i(t)} d(C, Xi(t)) \tag{15}$$

where $CD(X_i(\mathrm{t}),P(\mathrm{t}))$ denotes the nearest normalized distance between individual $X_i(t)$ and the other individuals in the current population.

The normalized distance between individuals is computed as follows:

$$d(C, X_i(t)) = \sqrt{\frac{1}{dim}\sum_{j=1}^{dim}\left(\frac{(x_{i,j}(t) - c_j)}{d^j_{max}(t) - d^j_{min}(t)}\right)^2} \tag{16}$$

where C is any individual in the current population other than $X_i(\mathrm{t})$, and c_j is its component in the j-th dimension. This normalized distance can weaken the influence of differences in the value ranges of different dimensions on the diversity computation, allowing $D_{pop}(\mathrm{t})$ to reflect the degree of aggregation of the current population more accurately.

In addition to the boosted-perturbation mode, this paper also retains a stable-search mode. In this mode, the relative-position parameter $m_{i,j}$ no longer uses the opposite position but is determined directly by the original position of the current individual within the dynamic boundaries:

$$m_{i,j} = \frac{x_{i,j}(t) - d^j_{min}(t)}{d^j_{max}(t) - d^j_{min}(t)} \tag{17}$$

The corresponding diffusion factor is set as follows:

$$s = 0.1\sqrt{D_{pop}(t)} + 0.9 \tag{18}$$

Compared to the boosted-perturbation mode, the diffusion factor varies more gently in the stable-search mode, generating small candidate perturbations near the advantageous region, thereby improving the search stability in the local-exploitation stage.

For clarity, the complete procedure of the Beta-distribution opposition-based learning strategy is summarized as follows, and its expanded pseudocode is given in Algorithm 1 (Section 3.4). The strategy is applied to every individual of the population at each iteration. At each iteration, one of two perturbation modes is selected stochastically: with probability 0.5, the boosted mode is used, in which the diffusion factor is given by Equation (11) and the relative-position parameter is measured from the opposite point ($d_{max} + d_{min} - x_i$); otherwise the stable mode is used, in which the diffusion factor takes

the milder form $s = 0.1 \cdot \sqrt{D_{pop}} + 0.9$ and the relative-position parameter is measured from the current point x_i. Hence, the population diversity D_{pop} only scales the magnitude of the diffusion factor within a mode and does not decide which mode is activated. For each individual, the Beta candidate is first sampled within the dynamic range $[d_{min}, d_{max}]$ and then clamped to the static search bounds $[lb, ub]$ for boundary handling. A greedy criterion is then adopted, i.e., the candidate replaces the individual only when it achieves a lower objective value, and the global best is updated whenever the candidate improves it.

Algorithm 1: Pseudo-code for BEMWPA Algorithm

Initialize
Generate an initial population $\boldsymbol{X}$ of size N using Equation (1) cubic chaotic map
Determine the optimal solution (s) from the initial population set P
Define the switch probability (SP = 0.3)
Define φ as a predefined control parameter
for t = 1 to max_iteration **do**
    Compute E_y using the Equation (4)
    Compute B_z using the Equation (5)
    for i = 1 to N **do**
        if rand $\leq$ 0.3 then
            Update X_i using the electric-field-driven exploration rule in Equation (6)
        else
            Update X_i using the magnetic-field-driven exploitation rule in Equation (7)
        end
        Compute the objective function
        if current results outperform previous ones then
            Update the population accordingly
        end
    end
    Apply the Beta-distribution opposition-based learning strategy (Equations (8)–(18)) to the current population:
    Compute dynamic bounds d_{max}, d_{min}, and population diversity D_{pop}
    if rand < 0.5 then
        Compute s by Equation (12); compute the relative position m from the opposite point ($d_{max} + d_{min} - X_i$)
    else
        Compute s by Equation (18); compute the relative position m from the current point X_i by Equation (17)
    for i = 1 to N do
        Compute Beta shape parameters α, β from ref_i using Equations (9) and (10)
        Generate the candidate $\widetilde{X}_i$ by Equation (9); clamp $\widetilde{X}_i$ to [lb, ub]
        if $f(\widetilde{X}_i) < f(X_i)$ then $X_i \leftarrow \widetilde{X}_i$ (greedy); update best if improved;
end for
    Determine the best solution obtained in the current iteration
end
End Algorithm

3.4. Algorithm Flowchart and Pseudocode

Figure 2 illustrates the overall workflow of the BEMWPA algorithm, while Algorithm 1 presents its pseudocode. First, the algorithm initializes the control parameters and generates a population of N candidate solutions using the cubic chaotic map. This initialization strategy improves the distribution uniformity and diversity of the population. After boundary handling, the fitness value of each individual is evaluated, and the best-performing individual is selected as the current target solution. During the iteration process, the algorithm computes the electric-field component E_y and the magnetic-field component B_z according to the phase-modulation mechanism, and selects different position-update methods through random probability to achieve a balance between global exploration and local exploitation. After updating, boundary handling and fitness evaluation are performed on the new solution, and the best position is updated according to the fitness result.

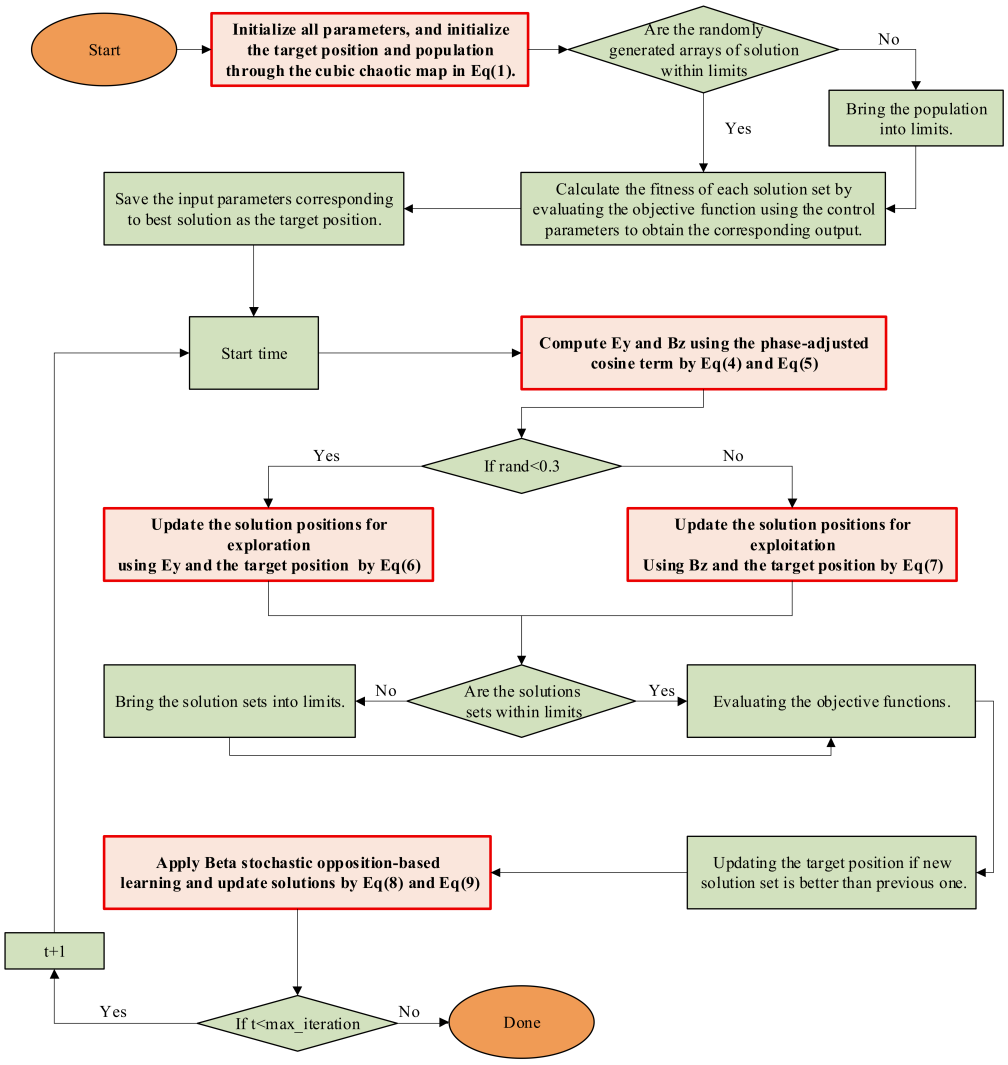


Figure 2. BEMWPA algorithm flowchart. Rectangles denote process steps; diamonds denote decision points; rounded boxes denote start/stop.

In addition, the algorithm introduces a Beta random opposition-based learning strategy to perform perturbation updates on the population, enhancing population diversity and reducing the probability of falling into local optima. When the number of iterations reaches the maximum, the algorithm stops and outputs the best solution.

4. Experimental Analysis

This section conducts a systematic experimental evaluation of the proposed BEMWPA. The algorithm introduces three improvement mechanisms into the original EMWPA framework to enhance the initial-population quality, strengthen the search-adjustment capability, and improve the late-stage exploitation accuracy, thereby improving its comprehensive performance on complex optimization problems.

The experiments are conducted on the CEC2017 benchmark set [28] and BEMWPA was compared with nine algorithms: EMWPA, ALA, POA, GOOSE, HHO, DBO, PSO, BKA, and PO [5,11,12,29–34]. The unified experimental parameters are set as follows: the population size is N = 30 and the maximum number of iterations is 1000. To reduce the influence of randomness, each algorithm is run independently 30 times on each test function, and the results are statistically analyzed. In addition, this paper verifies the application capability of BEMWPA through a manipulator path-planning problem. All benchmark functions follow the official definitions of the CEC2017 technical report, in which F2 is officially excluded because of its unstable numerical behavior; the remaining 29 functions are therefore used, with dimension D = 50 and unified search domain [−100, 100]^D. With a population size of 30 and 1000 iterations, each run corresponds to an evaluation budget of about 3×10^4 function evaluations. All compared algorithms are executed under the same budget, the same search bounds, and the same termination criterion, so that the comparison is fair. Each algorithm is executed 30 times independently on every function with a different random seed per run, and the statistical results reported in Tables A1–A3 are obtained from these 30 independent runs.

All experiments are executed on a Windows 11 system with an Intel(R) Core(TM) i7-14700HX processor, and the algorithm implementation platform is MATLAB R2024a (The MathWorks, Inc., Natick, MA, USA).

4.1. Ablation Studies of the BEMWPA Algorithm

To quantify the individual contribution of each proposed mechanism and to verify their complementarity, a full-factorial ablation study is conducted on the CEC2017 benchmark suite (29 functions, 50 dimensions). Starting from the original EMWPA, the three mechanisms introduced in this paper—cubic chaotic initialization (denoted C), square-root nonlinear phase modulation (denoted D), and Beta-distribution opposition-based learning (denoted O)—are switched on and off independently, producing eight algorithm variants: EMWPA (baseline, none), CEMWPA (C), DEMWPA (D), OEMWPA (O), CDEMWPA (C+D), COEMWPA (C+O), DOEMWPA (D+O), and BEMWPA (the complete model, C+D+O). All variants use identical control parameters (population size N = 30 and 1000 iterations) and the same number of function evaluations. Each variant is run independently 30 times on every function and is ranked on each function by its mean objective value. Figure 3 reports the results: the radar chart (left) shows the ranking distribution of each variant across the 29 functions, and the bar chart (right) summarizes the average ranking of each variant over the whole suite, where a lower average rank indicates better overall performance.

As shown in Figure 3, among the three mechanisms, Beta-distribution opposition-based learning (O) contributes the most: introducing it alone (OEMWPA) sharply lowers the average rank from 6.55 for the baseline EMWPA to 3.31, a reduction of about 49%. By contrast, adding cubic chaotic initialization alone (CEMWPA, 5.79) or square-root phase modulation alone (DEMWPA, 6.28) yields only a marginal improvement over the baseline. Combining these two without opposition-based learning (CDEMWPA, 6.90) does not help—it yields a slightly worse rank than the baseline. This indicates that, on their own, the improved initial-population diversity and the smoother phase-driven search cannot be effectively converted into higher convergence accuracy and mainly serve as supporting

components. Once combined with opposition-based learning, however, both mechanisms become clearly beneficial: COEMWPA (C+O) and DOEMWPA (D+O) reach average ranks of 2.48 and 2.72, respectively, which are both better than the single-mechanism OEMWPA (3.31), showing that the three mechanisms are complementary rather than redundant. Finally, the complete model BEMWPA (C+D+O) attains the best average rank of 1.97, surpassing every partial variant, including the strongest pair COEMWPA (2.48), and reducing the average rank by about 70% relative to the baseline EMWPA. These results demonstrate that each proposed strategy makes a positive contribution when properly combined, and that their integration is the key to the superior comprehensive performance of BEMWPA.

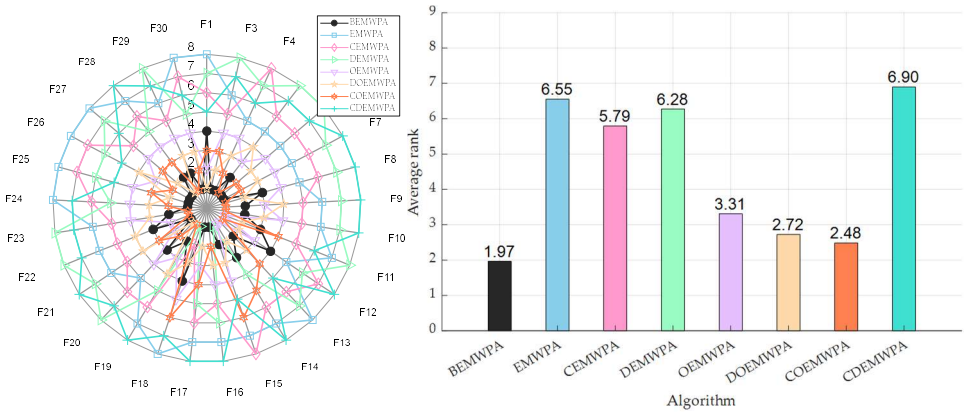


Figure 3. Ablation study of BEMWPA on CEC2017 (BEMWPA is the proposed algorithm).

4.2. Results and Analysis of the Test Functions on CEC2017

The CEC2017 benchmark set contains 30 test functions, among which F2 has been officially removed; therefore, this paper uses the remaining 29 functions for experimental evaluation. According to their characteristics, they can be divided into unimodal functions, simple multimodal functions, hybrid functions, and composition functions, and the experimental dimension is set to 50.

Among them, unimodal functions are mainly used to evaluate the local-exploitation capability and convergence accuracy of the algorithm. Conversely, multimodal, hybrid, and composition functions focus on examining the global search capability, the ability to maintain population diversity, and the ability to escape local optima.

4.2.1. Core Parameter Analysis

This section mainly evaluates the optimization performance of BEMWPA on the CEC2017 test function set, using the minimum, mean, and standard deviation as evaluation metrics, and ranking based on the mean (Avg) results. Smaller values of these three metrics indicate higher convergence accuracy and better stability of the algorithm. The above statistical-metric settings refer to the experimental evaluation methods commonly used in CEC benchmark tests and real-world constrained optimization problems [28,35].

Tables A1–A3 (Appendix A) present the statistical results on the CEC2017 50-dimensional test functions. Overall, BEMWPA outperforms the comparison algorithms on most functions, showing especially strong search capability and stability on unimodal, multimodal, and some hybrid functions. In terms of the Avg metric, BEMWPA achieves the best result on 20 of the 29 functions, accounting for 66.7%, which indicates its strong competitiveness in high-dimensional complex optimization problems. Compared to the original EMWPA, BEMWPA improves its ranking on all 29 functions, indicating that the improvement strategies effectively enhance the convergence accuracy, local-optimum es-

cape capability, and robustness of the algorithm. The overall results verify the effectiveness and application potential of BEMWPA in complex optimization problems.

Table 1 summarizes the parameter settings, implementation sources, stopping criterion, and boundary-handling method of all ten algorithms. For every algorithm, the population size is $N = 30$ and the stopping criterion is a maximum of 1000 iterations (about 3×10^4 function evaluations). The implementations follow the original references [5,11,12,29–34], and unless listed in Table 1, the default parameter values from those references are adopted. Out-of-range solutions are handled by clamping to the search bounds $[lb, ub]$.

Table 1. Parameter settings, implementation sources, and boundary-handling method of all algorithms.

Algorithm	Key Parameter Settings
BEMWPA	$N = 30$, $SP = 0.3$, $\eta \in [0.7,1]$, $E_{max} = 3$, $B_{max} = 1$, $\omega = 4$, $\lambda = 4000$
EMWPA	$SP = 0.3$, $\eta \in [0.7,1]$, $E_{max} = 3$, $B_{max} = 1$, $\omega = 4$, $\lambda = 4000$
ALA	Adaptive: $\theta = 2 \cdot \text{atan}(1 - t/T)$, energy E, factor G (no free constants)
POA	No free control parameters (fully adaptive)
GOOSE	coefficient threshold 0.17, $SW \in [10,14]$, sound speed 343.2
HHO	$E_0 \in [-1,1]$, $E_1 = 2(1 - t/T)$, random jump strength
DBO	$R = 1 - t/T$; producer/follower sub-swarm ratios as in original ref.
PSO	$w = 0.7$, $c_1 = c_2 = 1.2$, $V_{max} = 0.15 \cdot (ub - lb)$
BKA	$p = 0.9$, $n = 0.05 \cdot \exp(-2(t/T)^2)$, Cauchy mutation
PO	$\alpha = \text{rand}/5$, $\theta = \text{rand} \cdot \pi$, behavior switch $St \in \{1,2,3,4\}$

4.2.2. Convergence Analysis

Figure 4 shows the average convergence curves of BEMWPA and the comparison algorithms on selected CEC2017 test functions. BEMWPA can reduce the fitness value at a faster rate on most functions and ultimately obtain better convergence results, indicating that it has good coordination capability between global search and local exploitation.

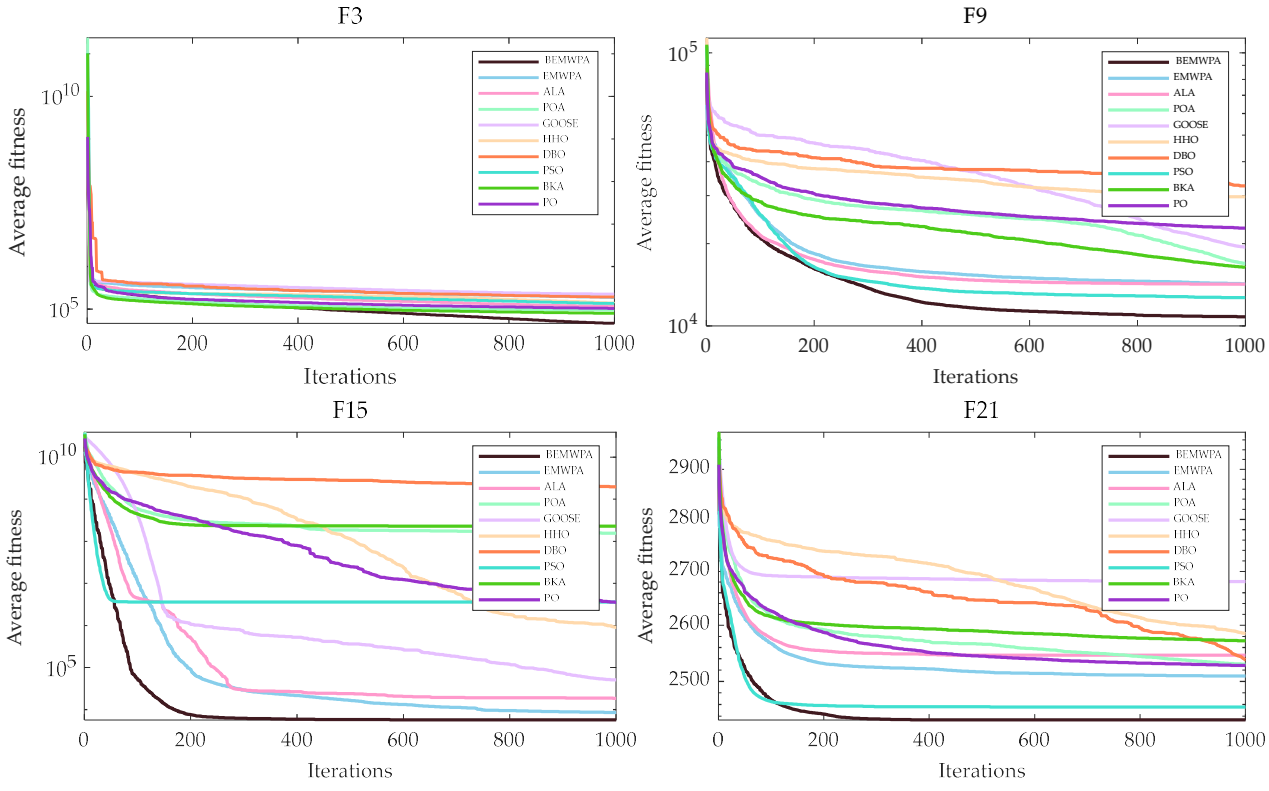


Figure 4. Convergence curves for four representative functions (F3, F9, F15, F21).

BEMWPA exhibits a faster convergence speed on some unimodal functions and maintains stable search performance on complex optimization problems, verifying the effectiveness of the improvement strategies in enhancing the comprehensive optimization capability of the algorithm.

4.2.3. Stability Analysis

Box plots can intuitively reflect the result distribution of multiple independent runs of an algorithm, where the box height indicates the degree of dispersion and the median line indicates the typical performance. Figure 5 shows the box-plot results of each algorithm on the 50-dimensional CEC2017 test functions.

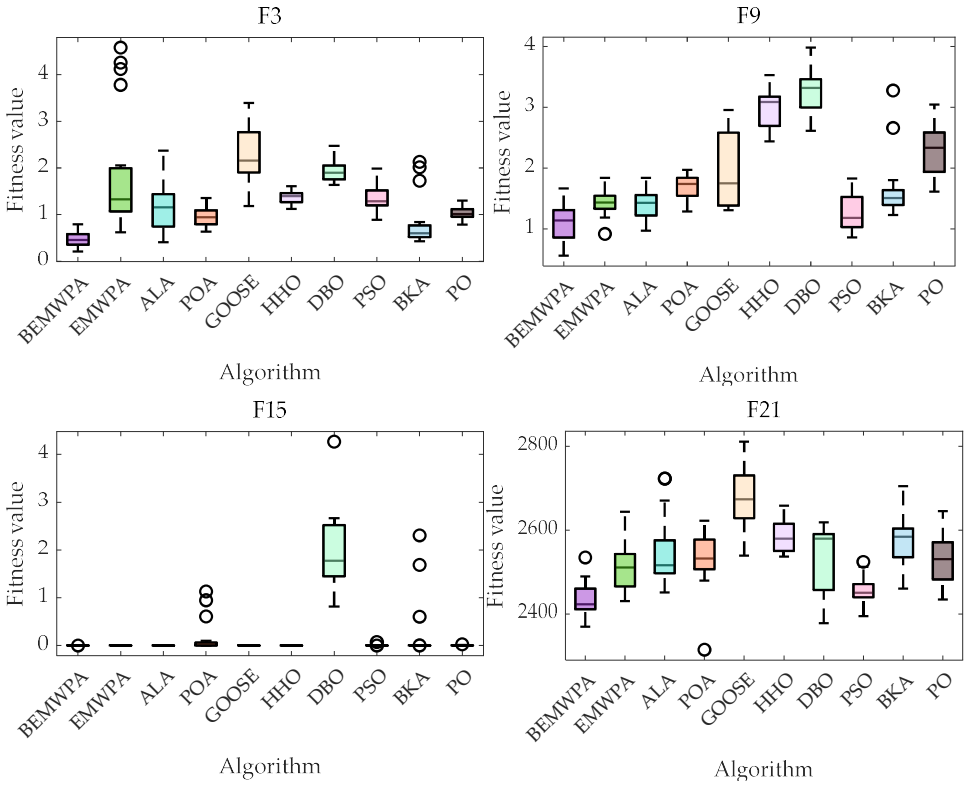


Figure 5. Box plots for four representative functions (F3, F9, F15, and F21).Box = interquartile range; central line = median; whiskers = range; circles = outliers.

As shown in Figure 5, BEMWPA has a smaller box width, a lower median, and fewer outliers on most functions, indicating its good search stability and result consistency. By contrast, EMWPA, GOOSE, DBO, and POA fluctuate greatly on some functions and tend to stagnate in local optima. Especially on hybrid functions and high-complexity functions, BEMWPA still maintains a relatively concentrated result distribution, reflecting strong robustness and local-optimum escape capability.

4.2.4. Statistical Analysis

Table 2 presents the Wilcoxon rank-sum test results between BEMWPA and each comparison algorithm on the 50-dimensional CEC2017 test functions. The results show that there are significant differences between BEMWPA and most algorithms, indicating that its performance improvement has good statistical reliability. Specifically, BEMWPA achieves significant advantages over DBO, PO, POA, and HHO on 29, 28, 26, and 26 functions, respectively, reflecting its strong high-dimensional optimization capability.

In addition, BEMWPA shows significant differences compared to GOOSE, BKA, and ALA on 25, 25, and 20 functions, respectively. Even compared to EMWPA and PSO, it

achieves significant advantages on 22 and 19 functions, respectively. Overall, BEMWPA can stably outperform the comparison algorithms on most test functions, verifying the effectiveness of chaotic initialization, the probabilistic switching mechanism, and the opposition-based learning strategy in enhancing the global search capability, local-exploitation capability, and robustness of the algorithm.

Table 2. Summary of Wilcoxon rank-sum test results on CEC2017 (50 dimensions).

Alg.	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
$p < 0.05$	22	20	26	25	26	29	19	25	28
$p > 0.05$	7	9	2	4	3	0	10	4	1

To provide a more rigorous statistical assessment for the ten algorithms over the 29 functions, a Friedman test with Nemenyi post hoc analysis is further performed on the mean results, and win/tie/loss statistics and effect sizes are reported. As shown in Table 3, the Friedman test yields $\chi^2 = 147.30$ ($df = 9$, $p = 3.2 \times 10^{-27}$) with Kendall's $W = 0.564$, indicating that the ten algorithms differ significantly and consistently across the benchmark. BEMWPA attains the best mean rank of 1.41, well ahead of the second-best ALA (3.45). Table 4 reports the Nemenyi post hoc p-value for BEMWPA against each competitor, the win/tie/loss count at the 0.05 level, and the rank-biserial effect size r. BEMWPA is significantly better than every comparison algorithm except ALA ($p = 0.24$), with an overall win/tie/loss of 213/44/4 over the 29 functions and large effect sizes ($r \geq 0.66$), which quantitatively confirms the effectiveness and robustness of BEMWPA.

Table 3. Friedman test and mean ranks of the ten algorithms on the CEC2017 50-dimensional functions.

Alg.	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
Mean rank	1.41	4.07	3.45	6.38	6.45	6.34	9.45	3.93	6.76	6.76

Table 4. Post hoc Nemenyi p-values, win/tie/loss counts, and rank-biserial effect sizes of BEMWPA versus each comparison algorithm (CEC2017, 50-D).

BEMWPA vs.	Nemenyi p	Win/Tie/Loss	Effect Size r
EMWPA	0.029	22/7/0	1.00
ALA	0.238	16/12/1	0.72
POA	<0.001	26/2/1	0.93
GOOSE	<0.001	24/4/1	0.93
HHO	<0.001	26/3/0	1.00
DBO	<0.001	29/0/0	1.00
PSO	0.050	18/11/0	0.66
BKA	<0.001	24/4/1	0.93
PO	<0.001	28/1/0	1.00
Total	—	213/44/4	—

4.2.5. Radar Chart

Figure 6 presents the radar chart of the average ranking of each algorithm on the CEC2017 test functions. It can be seen that the ranking of BEMWPA is closer to the center on most functions, and its overall performance is better than that of the original EMWPA, indicating that the proposed improvement strategies effectively enhance the comprehensive optimization capability of the algorithm. Meanwhile, the ranking distribution of BEMWPA across different types of functions is relatively balanced, indicating that it has good adaptability on unimodal, multimodal, and complex composition functions. By

contrast, although PSO and ALA perform well on some functions, their overall rankings fluctuate greatly and their stability is relatively insufficient.

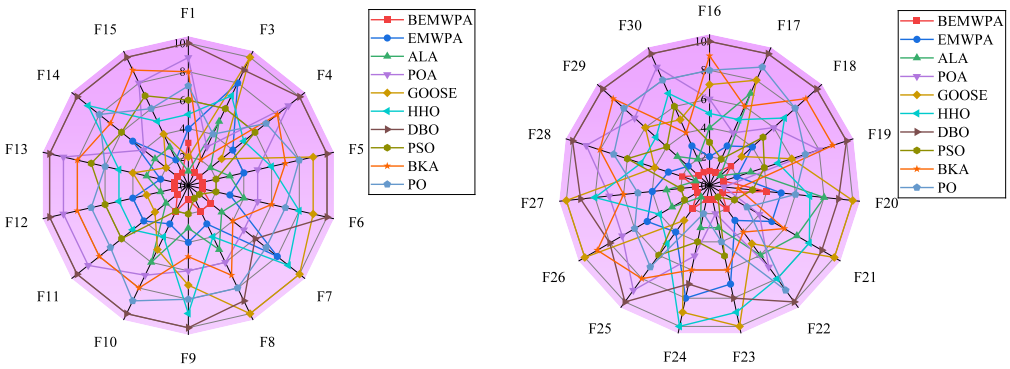


Figure 6. CEC2017 radar chart (BEMWPA is the proposed algorithm).

Figure 7 shows the average-ranking results of each algorithm on the CEC2017 test functions. BEMWPA achieves the lowest average-ranking value of 1.41, significantly better than 4.07 for EMWPA, indicating its better overall optimization performance. Meanwhile, the ranking variation of BEMWPA is relatively stable, further verifying its stable search capability and robustness across different test functions.

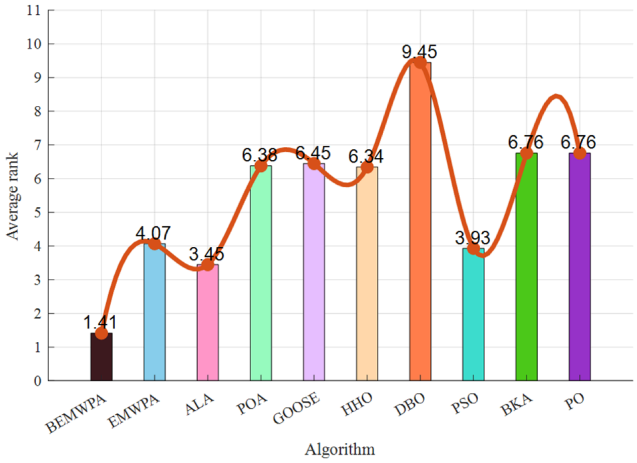


Figure 7. CEC2017 ranking chart. (BEMWPA is the proposed algorithm).

To quantify the computational overhead of the proposed strategies, the time complexity of BEMWPA is analyzed. For a population of size N, dimension D, and T iterations, the original EMWPA has a per-iteration complexity of $O(N{\cdot}D)$ plus N fitness evaluations, i.e., $O(T{\cdot}N{\cdot}(D + C_f))$, where C_f is the cost of one fitness evaluation. BEMWPA adds three components: cubic chaotic initialization adds a one time cost of $O(N{\cdot}D)$; square-root phase modulation only replaces the phase term and adds negligible cost; and the Beta-distribution opposition-based learning strategy, which is applied to the whole population at each iteration and computes the population diversity through pairwise normalized distances, adds a cost of $O(N^2{\cdot}D)$ and requires one additional round of fitness evaluations. The overall complexity of BEMWPA is therefore $O(T{\cdot}(N^2{\cdot}D + N{\cdot}C_f))$. Compared to EMWPA, it introduces an extra $O(N^2{\cdot}D)$ diversity term and about three times as many fitness evaluations per iteration, which is the cost of the improved accuracy and robustness.

Table 5 further reports the measured average runtime per run over the 29 functions. Although BEMWPA performs more evaluations per iteration than the baseline, its actual runtime is only 0.097 s, the second lowest among the ten algorithms—close to the original

EMWPA (0.019 s) and far below the eight recent competitors (0.367–9.715 s)—because the population size $N = 30$ keeps the $O(N^2 \cdot D)$ term small while the fitness evaluation dominates the runtime of the other algorithms. Hence, the accuracy and robustness gains of BEMWPA are obtained at a low and practical computational cost.

Table 5. Average runtime per run of all algorithms on the CEC2017 50-dimensional functions.

Alg.	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
Average runtime per run (s)	0.097	0.019	0.388	0.621	0.367	0.847	0.424	0.402	0.754	9.715

4.3. Path-Planning Experiment for a Welding Manipulator in Complex Multi-Workpiece Scenarios

To verify the engineering applicability of the proposed improved algorithm in complex manipulator welding scenarios, this paper further constructs a three-dimensional welding path-planning scenario containing 12 workpieces, 12 closed weld seams, and multiple obstacle constraints. Unlike a single mathematical test function or a small-scale workpiece scenario, this experiment directly maps the algorithm performance to the collision-free transfer distance of the manipulator end-effector, the path smoothness, and the kinematic realizability [4,13,23–27]. The welding segments are given by the process task, with a total length of 2388.00 mm; the inter-seam transfer segments are generated by the path-planning algorithm. The ideal transfer length under obstacle-free conditions is 477.52 mm, while the complete path length obtained by the initial RRT planning is 2974.00 mm, of which the transfer path length is 586.00 mm. Therefore, the inter-seam transfer segments are the main optimization target affecting the cycle time of complex multi-workpiece welding and the idle-travel energy consumption of the manipulator.

4.3.1. Workpiece Welding Scenario and RRT Initial-Path Construction

The multi-point welding path-planning program takes the closed weld seams of 12 workpieces, WP-A to WP-L, as the task object. The four bottom edges of each workpiece form a closed weld seam. The program first determines the workpiece visiting order and the entry corner of each closed weld seam through a nearest-neighbor strategy and local 2-opt improvement, and then generates collision-free transfer paths between adjacent weld seams. The welding order obtained in the current experiment is WP-A, WP-E, WP-H, WP-C, WP-K, WP-G, WP-L, WP-D, WP-J, WP-I, WP-F, and WP-B, forming 11 inter-seam transfer segments in total. In the path-planning program, RRT [36–39] is responsible for rapidly constructing a feasible initial path in the three-dimensional Cartesian space containing workpiece obstacles and platform constraints, after which BEMWPA and the comparison algorithms continuously optimize the intermediate nodes of the RRT path. To provide a feasible channel for subsequent optimization, RRT adopts a goal-biased sampling strategy to determine the expansion direction and generates new nodes through nearest-neighbor search and step-length constraints; its sampling, nearest-neighbor selection, and node-expansion processes are expressed below. It should be noted that the transfer paths are planned in the full three-dimensional workspace. The RRT sampler draws candidate points over the entire x–y–z range ($z \in [18, 180]$), and the platform, the extruded workpiece blocks, and the cylindrical pillars are all modeled as finite-height 3D solids with full 3D collision checking. Vertical (over-the-top) transfer is therefore permitted rather than excluded by construction. The optimized paths mostly route laterally around the obstacle footprints because the objective minimizes the transfer-path length, and detouring around is generally shorter than passing over the top when lateral clearance is available.

$$q_{\text{rand}} = \begin{cases} q_{\text{goal}}, & \rho < p_g \\ q_{\text{free}}, & \rho \geq p_g \end{cases} \tag{19}$$

$$q_{\text{near}} = \underset{q_i \in T}{\operatorname{argmin}} \|q_i - q_{\text{rand}}\|_2 \tag{20}$$

$$q_{\text{new}} = q_{\text{near}} + \frac{\eta (q_{\text{rand}} - q_{\text{near}})}{\|q_{\text{rand}} - q_{\text{near}}\|_2 + \varepsilon} \tag{21}$$

To ensure the geometric feasibility of the expansion edge, after each new node is generated, the program performs collision detection on the candidate segment. Only when the candidate segment does not intersect the obstacle set is the node merged into the random tree. Its constraint relationship can be expressed as follows:

$$\text{line}(q_{\text{near}}, q_{\text{new}}) \cap \Omega_{\text{obs}} = \varnothing \tag{22}$$

In Equations (19)–(22), q denotes the node state in the planning space, which in this experiment corresponds to the candidate path point of the manipulator end-effector in the three-dimensional workspace; q_{rand} is the random sampling state; q_{goal} is the goal state; q_{free} is a random state within the free space C_{free}; ρ is a random number in the interval [0,1]; and p_g is the goal-bias probability. q_i is the i-th node in the random tree T; q_{near} is the nearest node with the minimum Euclidean distance to the sampling point; q_{new} is the new node obtained by expanding from q_{near} toward q_{rand}; η is the single-step expansion length, ε is a small positive quantity preventing a zero denominator; and $\|\cdot\|_2$ denotes the Euclidean norm. Ω_{obs} denotes the obstacle set; line(q_{near},q_{new}) is the candidate connecting segment between the two nodes; and $\cap$ and $\varnothing$ denote the set intersection and the empty set, respectively. Equation (21) indicates that only when the candidate connecting segment does not intersect any obstacle does the expansion edge satisfy the obstacle-avoidance feasibility requirement.

When the distance between the new node and the goal point is smaller than the threshold, the algorithm completes the path connection and obtains the initial transfer path by backtracking through the parent-node indices. Subsequently, a random-shortcut smoothing procedure is used to compress the redundant polyline: if the direct segment between two non-adjacent nodes is collision-free, the intermediate nodes are removed. This treatment can reduce the degree of path detour, but because the RRT nodes are generated by random expansion, the resulting path still has local redundancy and corner mutations, and therefore requires further swarm-intelligence optimization.

Figure 8 shows the kinematic verification results after the BEMWPA-optimized path is mapped onto the seven-degrees-of-freedom welding manipulator. In the figure, the workpieces, the welding platform, and the end-effector motion trajectory together constitute the multi-workpiece welding execution scenario. The joint postures of the manipulator vary continuously with the path points, and the end-effector can complete the tracking along the optimized closed weld seams and inter-seam transfer segments. This result indicates that the Cartesian-space path obtained by BEMWPA-RRT not only satisfies the obstacle-avoidance requirement at the geometric level but can also be converted, through inverse-kinematics solving, into a joint-space trajectory executable by the Welding7R manipulator, providing a kinematic-feasibility basis for the subsequent path-length statistics and multi-algorithm comparison.

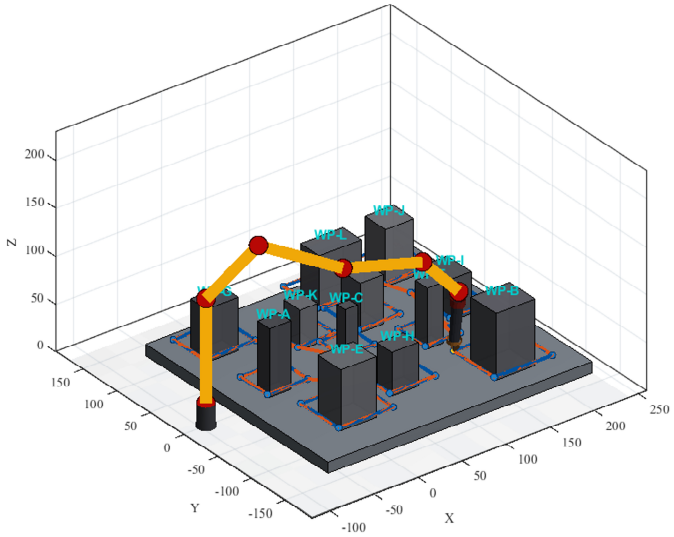


Figure 8. Kinematic verification of the seven-degrees-of-freedom welding manipulator under the BEMWPA-optimized path.

4.3.2. BEMWPA-RRT Path-Node Optimization Model

After obtaining the RRT feasible path, this paper takes the intermediate nodes of each transfer segment as continuous optimization variables and searches for a shorter and smoother collision-free path within the neighborhood of the RRT nodes. Let the path of the s-th transfer segment be $P_s = \{p_{s,1}, p_{s,2}, \ldots, p_{s,n_s}\}$, where $p_{s,1}$ and p_{s,n_s} are determined by the endpoints of the adjacent weld seams and do not participate in the optimization. The remaining intermediate nodes constitute the algorithm individual:

$$X_s = [p_{s,2}, p_{s,3}, \ldots, p_{s,n_s-1}] \in \mathrm{R}^{3(n_s-2)} \tag{23}$$

In Equation (23), X_s denotes the optimization-variable vector of the s-th inter-seam transfer segment, composed of the intermediate path nodes excluding the fixed start point $p_{s,1}$ and the end point $p_{s,ns}$; $p_{s,i} \in R^3$ is the i-th three-dimensional path node; n_s is the total number of nodes in this transfer segment; and $R^{3(\mathrm{ns}-2)}$ denotes the dimension of the variables to be optimized in this transfer segment.

On this basis, the path cost is jointly composed of a length term, a smoothness term, and a collision-penalty term. Among them, the path length measures the idle-travel distance between weld seams, and the smoothness term suppresses sharp turns between consecutive path segments. The two can be expressed as follows:

$$L(P_s) = \sum_{i=1}^{n_s-1} \| p_{s,i+1} - p_{s,i} \|_2 \tag{24}$$

$$S(P_s) = \sum_{i=2}^{n_s-1} \arccos\left(\frac{(p_{s,i} - p_{s,i-1}) \cdot (p_{s,i+1} - p_{s,i})}{\| p_{s,i} - p_{s,i-1} \|_2 \| p_{s,i+1} - p_{s,i} \|_2 + \varepsilon} \right) \tag{25}$$

In Equations (24) and (25), $L(P_s)$ is the length of the s-th transfer path; $S(P_s)$ is a smoothness metric based on the accumulation of the angles between adjacent path segments; $p_{s,i-1}$, $p_{s,i}$, and $p_{s,i+1}$ are three consecutive path nodes within the same transfer segment; $arccos(\cdot)$ is used to compute the angle between adjacent path segments; "$\cdot$" denotes the vector inner product; $||\cdot||_2$ is the Euclidean norm; and ε is a small positive quantity preventing a zero denominator.

To further clarify the calculation of the collision constraint, this paper converts the segment-level collision-detection result into a 0–1 variable. If the segment formed by

adjacent path nodes intersects the obstacle set, the segment is recorded as a collision segment. The collision-indicator variable, the collision-segment count, and the path-level collision flag are defined as:

$$I_{s,i}^{c} = \begin{cases} 1, & \text{line}(p_{s,i}, p_{s,i+1}) \cap \Omega_{\text{obs}} \neq \varnothing \\ 0, & \text{line}(p_{s,i}, p_{s,i+1}) \cap \Omega_{\text{obs}} = \varnothing \end{cases} \tag{26}$$

$$C(P_s) = \sum_{i=1}^{n_s-1} I_{s,i}^{c} \tag{27}$$

$$I_c(P_s) = \begin{cases} 1, & C(P_s) > 0 \\ 0, & C(P_s) = 0 \end{cases} \tag{28}$$

In Equations (26)–(28), for the i-th adjacent-node connecting line in the s-th path segment, Ω_{obs} is the obstacle set; line($p_{s,i}$,$p_{s,i+1}$) denotes the candidate segment between adjacent nodes; $C(P_s)$ denotes the number of collision segments occurring in path P_s; and I_c indicates whether the entire candidate path has a collision. If $C(P_s) > 0$, then $I_c = 1$; otherwise, $I_c = 0$.

Based on the above definitions of the length, smoothness, and collision constraints, the comprehensive fitness function of the s-th candidate path is given by:

$$F(P_s) = L(P_s) + \omega_s S(P_s) + \omega_c \sum_{i=1}^{n_s-1} I_{s,i}^{c} \| p_{s,i+1} - p_{s,i} \|_2 + M I_c(P_s) \tag{29}$$

In Equation (29), $F(P_s)$ is the comprehensive fitness function of the s-th candidate path; w_s and w_c are the smoothness weight and the collision-penalty weight, respectively; and M is the large penalty coefficient for infeasible paths. In this experiment, $w_s = 0.08$, $w_c = 10^5$, and $M = 5 \times 10^4$. This setting makes the algorithm prioritize satisfying the obstacle-avoidance constraints and further compresses the inter-seam idle travel and reduces path turning within the feasible domain.

To confirm that the results are not sensitive to these hand-chosen weights, we further varied the smoothness weight w_s over {0.02, 0.08, 0.32} and the collision-penalty weight over $\{2.5 \times 10^4, 1 \times 10^5, 4 \times 10^5\}$ (a sixteen-fold range each) on the same twelve-workpiece scenario and the same RRT initial path, with five independent runs per combination. All nine combinations produced collision-free paths (zero colliding segments), and the cumulative transfer-path length varied by, at most, 13.4% from the baseline setting ($w_s = 0.08$, $w_c = 1 \times 10^5$), with no systematic trend with respect to either weight, a variation comparable to the run-to-run randomness of the optimizer. Therefore, the algorithm is robust to these weights within a wide range. Regarding dimensional consistency, the path-length and collision terms have the dimension of length (mm); the smoothness term is an accumulated turning angle and is dimensionless; w_s and w_c act as scaling factors that bring these terms to a magnitude comparable with the path length; and M is set much larger than any feasible path length so that infeasible paths are always rejected. The same weights are used for all compared algorithms, so the comparison is unaffected.

The complete path consists of fixed welding segments and optimized transfer segments, and its length is computed as follows:

$$L_{\text{full}} = L_{\text{weld}} + L_{\text{trans}}, \qquad L_{\text{trans}} = \sum_{s=1}^{N_T} L(P_s) \tag{30}$$

$$IR_{\text{trans}} = \frac{L_{\text{RRT}}^{\text{trans}} - L_{\text{alg}}^{\text{trans}}}{L_{\text{RRT}}^{\text{trans}}} \times 100\% \tag{31}$$

In Equations (30) and (31), L_{full} is the total length of the complete welding path; L_{weld} is the length of the fixed welding segments; L_{trans} is the total length of the inter-seam transfer paths; N_T is the total number of inter-seam transfer segments; IR_{trans} is the reduction rate relative to the initial RRT transfer path; and $L_{\mathrm{RRT}}^{\mathrm{trans}}$ and $L_{\mathrm{alg}}^{\mathrm{trans}}$ are the initial RRT transfer-path length and the transfer-path length obtained by the optimization algorithm, respectively.

Through the above modeling, BEMWPA continuously optimizes the intermediate nodes within the neighborhood of the RRT feasible channel. Consequently, the candidate path simultaneously balances path-length compression and local-turning suppression while satisfying the obstacle-avoidance constraints, thereby improving the quality of the feasible path in complex multi-workpiece welding scenarios.

4.3.3. Trajectory Generation and Kinematic Verification of the Seven-Degrees-of-Freedom Welding Manipulator

The path optimization yields a discrete pose sequence of the end-effector in three-dimensional space, which still needs to be converted into a joint-space trajectory of the manipulator. The program first converts each path point p_i into a homogeneous pose matrix T_i, and solves through inverse kinematics for the corresponding joint angles q_i:

$$T_i = \begin{bmatrix} R_i & p_i \\ 0\,0\,0 & 1 \end{bmatrix}, \quad q_i = IK(T_i) \tag{32}$$

In Equation (32), Ti is the homogeneous pose matrix corresponding to the i-th end-effector path point; R_i and p_i are the end-effector attitude rotation matrix and position vector, respectively; qi is the manipulator joint-angle vector obtained by the inverse-kinematics operator $IK(\cdot)$; and $IK(\cdot)$ denotes the inverse-kinematics solution mapping from the end-effector pose to the joint space.

In the twelve-workpiece multi-point welding verification, the program establishes the Welding7R seven-degrees-of-freedom welding manipulator model. The model consists of base yaw J1, upper arm J2, elbow J3, forearm J4, additional forearm segment J5, wrist pitch J6, and wrist roll J7; the main geometric parameters are set as $d_1 = 110$, $s_2 = 68$, $s_3 = 92$, $s_4 = 78$, $s_5 = 52$; the wrist-joint twist angles are $\alpha_6 = -\pi/2$ and $\alpha_7 = \pi/2$; the base transform is transl(−95,0,24); and the end-tool offset is transl(0,0,65). Compared to a non-redundant manipulator model, this redundant-degree-of-freedom structure can provide greater null-space adjustment capability for attitude adjustment and link obstacle-avoidance while keeping the end welding torch reaching the target welding point.

The program solves the joint angles using damped inverse kinematics [40,41] with position-task priority and attitude constraints corrected in the null space. The core iterative form of its position task can be written as follows:

$$\Delta q = J_p^T \left(J_p J_p^T + \lambda I \right)^{-1} e \tag{33}$$

$$q^{k+1} = q^k + \alpha \Delta q \tag{34}$$

In Equations (33) and (34), J_p is the end-effector position-task Jacobian matrix; $J_p^T \left(J_p J_p^T + \lambda I \right)^{-1}$ is the damped pseudo-inverse form; λ is the damping factor; I is the identity matrix; e is the end-effector position-error vector; Δq is the joint increment in one iteration; q^k and q^{k+1} denote the joint-angle vectors at the k-th and $(k + 1)$ iterations, respectively; and α is the update step length. In the actual solving process, the program superimposes the welding-torch axial-attitude constraint and the reference-attitude correction term in the null space of the position task. This maintains the torch orientation, improves the trajectory continuity of the seven-degrees-of-freedom redundant manipulator, and reduces the link-collision risk.

After obtaining the complete Cartesian path optimized by BEMWPA, the program performs densified sampling on the path and uses the joint solution of the previous sampling point as the initial value for the next sampling point to maintain the continuity of the joint-solution sequence. The path-densification constraint and the joint-angular-velocity estimation can be expressed as follows:

$$P_d = \text{densify}(P^*, \Delta s), \quad \left\| p_{j+1} - p_j \right\|_2 \leq \Delta s \tag{35}$$

$$\dot{q}_k(t_j) \approx \frac{q_k(t_{j+1}) - q_k(t_j)}{\Delta t} \tag{36}$$

In Equations (35) and (36), P_d denotes the densified Cartesian path-point sequence; densify$(\cdot)$ denotes the path-densification sampling operator; P^* is the complete end-effector path optimized by BEMWPA; Δs is the maximum spatial step length allowed between adjacent sampling points; and p_j and p_{j+1} are adjacent densified path points. $q_k(t_j)$ is the angular-velocity estimate of the k-th joint at the discrete instant t_j; $q_k(t_j)$ is the corresponding joint angle; t_j and t_{j+1} are adjacent discrete sampling instants; and Δt is the time interval between adjacent sampling points. Equation (35) is used to limit the spatial distance between adjacent Cartesian sampling points, and Equation (36) is used to estimate the joint angular velocity from the discrete joint sequence, providing a basis for the subsequent velocity-constraint analysis rather than serving alone as a velocity-optimization objective.

This trajectory-generation method, through path densification, continuous inverse solving, and discrete velocity estimation, can suppress joint mutations between adjacent path points and provide a discrete joint-trajectory basis for the subsequent velocity-constraint analysis. The graphical simulation further verifies whether the manipulator end-effector can move continuously along the weld seams and transfer segments under the optimized path, providing a kinematic basis for the transition of the algorithm results from a geometric path to a manipulator-executable trajectory.

4.3.4. Comparison of Multi-Algorithm Path-Optimization Results and Motion Verification

To verify the path-optimization capability of BEMWPA-RRT in complex multi-workpiece welding tasks, this paper selects EMWPA, ALA, POA, GOOSE, HHO, DBO, PSO, BKA, and PO as comparison algorithms, and performs statistics under the same twelve-workpiece scenario, the same RRT initial-path set, and the same collision constraints. This experiment contains 11 inter-seam transfer segments in total, with a fixed welding-segment length of 2388.00 mm, an RRT initial complete-path length of 2974.00 mm, and a corresponding inter-seam transfer length of 586.00 mm. Table 6 presents the cumulative optimization results of each algorithm over the 11 transfer segments.

As shown in Table 6, BEMWPA achieves the shortest cumulative transfer-path length of 479.11 mm, representing a relative reduction of 18.24% compared with the initial RRT transfer-path length of 586.00 mm. The corresponding complete-path length is reduced from 2974.00 mm to 2867.11 mm, representing a relative reduction of 3.59%. Since the fixed welding segment of 2388.00 mm remains unchanged across all algorithms, the relative reduction of the complete path is diluted by the welding-process length. If only the optimizable inter-seam transfer segments are considered, the advantage of BEMWPA is more direct. Furthermore, the BEMWPA transfer path is only 1.59 mm longer than the obstacle-free ideal transfer length of 477.52 mm, i.e., about 0.33% higher, indicating that it is already close to the geometric lower bound of this scenario while satisfying the obstacle-avoidance constraints.

In terms of statistical stability, the mean cumulative length of BEMWPA is 479.28 mm with a standard deviation of only 0.14 mm, which is significantly lower than that of algorithms such as GOOSE, HHO, DBO, and PO, indicating its more stable convergence

characteristics in welding environments with multiple coupled segments and dense obstacles. PSO, ALA, and BKA can also obtain relatively short paths, but their best cumulative transfer lengths are 490.76 mm, 497.85 mm, and 498.41 mm, respectively, which are still higher than that of BEMWPA. The best transfer length of DBO is 622.79 mm, which is higher than the initial RRT transfer length, indicating that it tends to deviate from the effective feasible channel in this type of multi-obstacle narrow-channel path-optimization problem.

Table 6. Statistical results of multi-algorithm inter-seam transfer-path optimization in the twelve-workpiece scenario.

Algorithm	Optimal Transition Length (mm)	Average Cumulative Length (mm)	Standard Deviation (mm)	Worst Cumulative Length (mm)	Compared to RRT Shortening Rate (%)
BEMWPA	479.11	479.28	0.14	479.57	18.24
PSO	490.76	502.05	5.43	512.07	16.25
ALA	497.85	543.39	27.14	579.60	15.04
BKA	498.41	521.31	11.48	546.69	14.95
POA	504.09	534.62	7.24	549.12	13.98
EMWPA	519.14	550.67	11.50	568.40	11.41
GOOSE	521.58	616.33	56.52	751.18	10.99
HHO	539.24	598.68	29.89	649.43	7.98
PO	559.59	631.78	18.45	656.40	4.51
DBO	622.79	753.82	38.80	822.52	−6.28

In Figure 9, the gray dashed line indicates the RRT initial transfer length of 586.00 mm, and the colored bars indicate the best cumulative transfer length of each algorithm. Except for DBO, all of the other algorithms are below the RRT baseline, and BEMWPA is the shortest. The manipulator experiment takes the final feasible-path quality and path-compression effect as the main metrics. For all ten algorithms, the same welding scenario, the same RRT initial-path set, the same decision variables (the intermediate transfer nodes), the same cost function and collision constraints, and the same number of iterations and independent runs are used. No algorithm-specific post-processing is applied to any method. Therefore, this paper does not take the running time as a direct comparison metric under an equal computational budget.

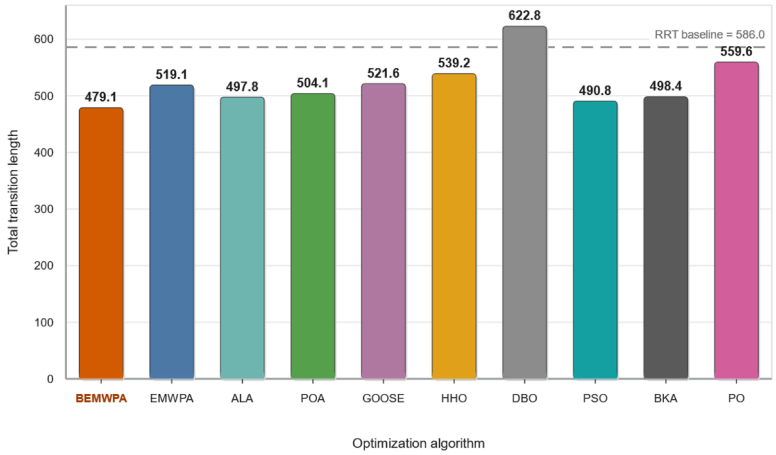


Figure 9. Comparison of inter-seam transfer-path lengths of different optimization algorithms in the twelve-workpiece scenario (BEMWPA is the proposed algorithm).

Figure 10 shows the spatial path of the twelve-workpiece multi-point welding optimized by BEMWPA. The blue closed polyline represents the fixed welding segments,

and the orange polyline represents the inter-seam transfer segments optimized by BEMWPA. Compared with the RRT initial transfer length of 586.00 mm, BEMWPA compresses the cumulative transfer length to 479.11 mm and reduces the complete-path length to 2867.11 mm, indicating that it can effectively reduce the idle travel between workpieces while maintaining obstacle-avoidance feasibility.

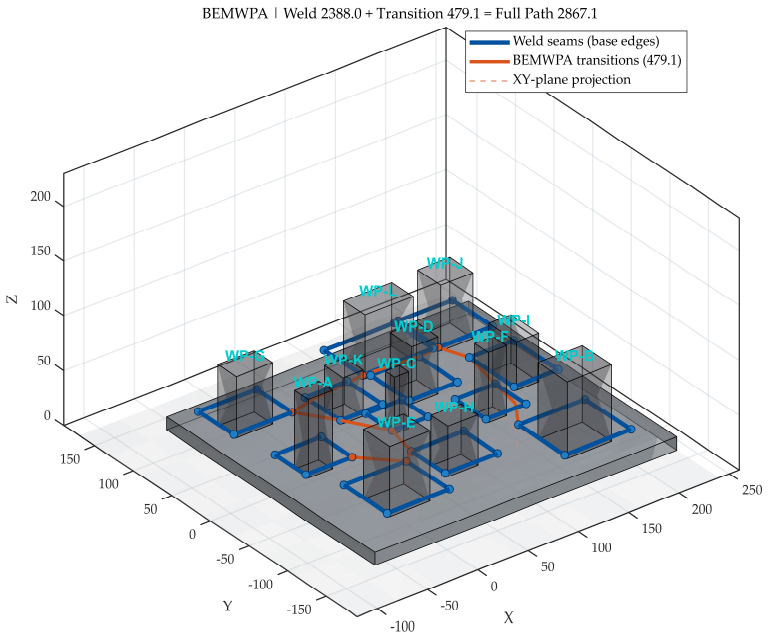


Figure 10. Spatial path of twelve-workpiece multi-point welding optimized by BEMWPA.

On this basis, Figure 11 further presents the path results of the nine comparison algorithms, other than BEMWPA, in the same twelve-workpiece scenario. The value at the bottom of each subfigure is the cumulative inter-seam transfer length of the corresponding algorithm; the blue segments represent the fixed welding segments, and the orange segments represent the inter-seam transfer segments obtained by that algorithm. A comparison with Figure 10 reveals that PSO, ALA, and BKA can also form relatively short transfer paths, but their cumulative lengths are still longer than that of BEMWPA. Additionally, the transfer segments of algorithms such as DBO and PO are longer, indicating that their ability to compress effective channels in multi-obstacle coupled scenarios is relatively insufficient.

To further verify the manipulator executability of the optimized path, this paper uses the Welding7R seven-degrees-of-freedom welding manipulator for kinematic simulation. The model contains a raised base, a shortened upper arm, an elbow link, a forearm, an additional forearm segment, a wrist-pitch joint, and a wrist-roll joint; the base pose is transl(−95,0,24), and the end-tool offset is transl(0,0,65). The program first performs densified sampling on the complete Cartesian path optimized by BEMWPA. It then uses the joint solution of the previous sampling point as the initial value to solve the joint-space trajectory point by point through damped task-priority inverse kinematics. Finally, it checks for collisions between the manipulator links and the workpiece obstacles during the solving process.

As shown in Figure 8, the end-effector can move continuously along the optimized closed weld seams and inter-seam transfer paths, and the manipulator links do not interfere with the workpiece obstacles in the visualization results. It can thus be seen that BEMWPA-RRT can not only reduce the inter-seam idle-travel length in Cartesian space but can also be converted into a joint-space trajectory executable by the redundant-degree-of-freedom welding manipulator, providing a feasible basis for offline welding path planning of complex multi-workpiece tasks.

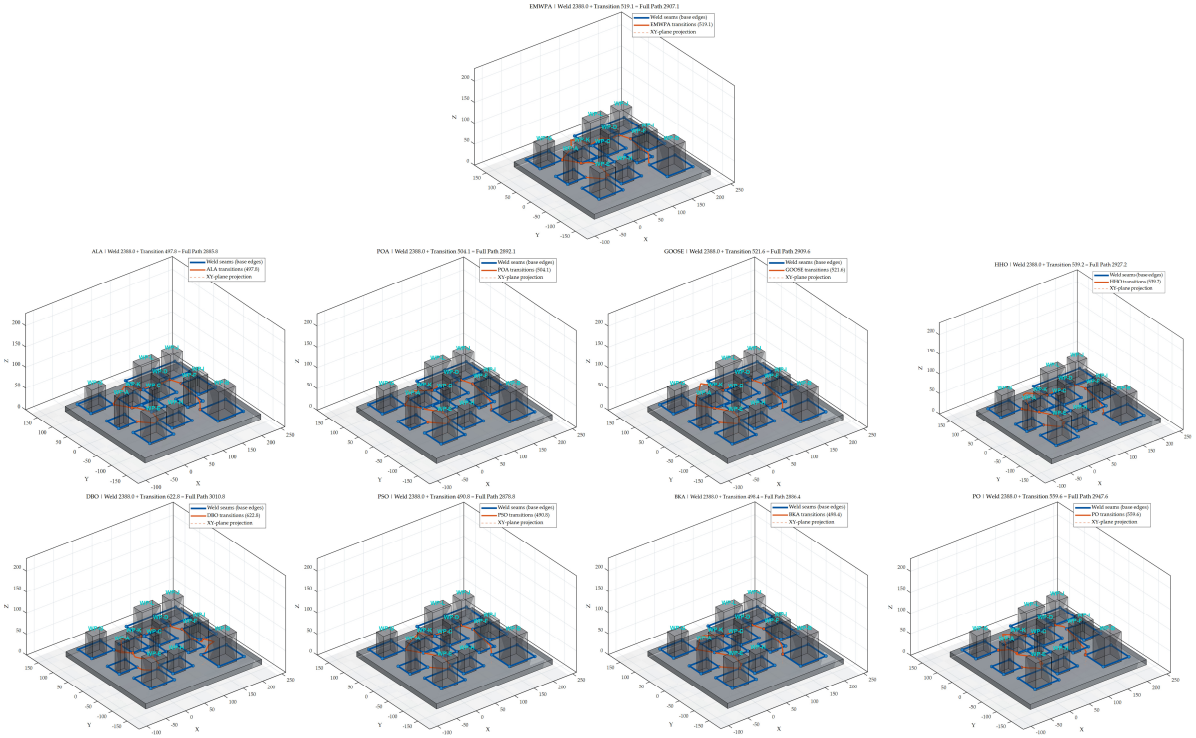


Figure 11. Visual comparison of inter-seam transfer paths of nine comparison algorithms in the twelve-workpiece scenario.

4.3.5. Generalization Verification in an Expanded Scenario with Sixteen Workpieces and Two Obstacles

To further verify the generalization and robustness of the proposed BEMWPA beyond the twelve-workpiece case studied above, a larger and more challenging engineering scenario is constructed. The number of workpieces is increased from 12 to 16, and two additional cylindrical obstacle pillars are introduced into the workspace, so that the inter-loop transfer paths must avoid a denser set of three-dimensional obstacles. As in the previous experiment, the base of each workpiece is welded along all four edges, yielding sixteen closed weld loops with a fixed total weld length of 3012.0. The welding order of the loops is first determined by the nearest-neighbor and 2-opt methods, and the inter-loop transfer paths are then generated by RRT and refined by each optimization algorithm. Since the total weld length is fixed, the total inter-loop transition length is adopted as the performance metric, where a shorter transition length indicates a more efficient collision-free transfer scheme.

Figure 12 shows the complete welding path planned by BEMWPA in this expanded scenario. The sixteen workpieces (WP-A–WP-P) are welded in the optimized order, and the transfer segments (orange) smoothly bypass the two obstacle pillars (OBS-1 and OBS-2) and the surrounding workpieces without collision. BEMWPA compresses the inter-loop transition length to 707.8, so that the full welding path is 3012.0 + 707.8 = 3719.8. Compared to the raw RRT transfer length of 1012.6, BEMWPA shortens the transition path by about 30.1%, demonstrating its effectiveness in a complex three-dimensional environment with multiple obstacles.

To assess the relative performance of BEMWPA, the same transfer-planning task is solved by all ten algorithms under identical maps and welding order, and the resulting total transition lengths are compared in Figure 13. BEMWPA attains the shortest transition length of 707.8, outperforming every competitor, including the second-best PSO (757.6) and the original EMWPA (829.4). Relative to EMWPA, BEMWPA reduces the transition length by about 14.7%. The remaining algorithms follow with 770.9 (ALA), 777.1 (GOOSE),

779.4 (BKA), 799.8 (HHO), 817.8 (POA) and 863.5 (PO), whereas DBO yields the longest path (997.6), which is close to the raw RRT baseline. These results confirm that BEMWPA retains its advantage in a larger and more obstacle-dense scenario, providing consistent and generalizable performance for real multi-workpiece welding applications.

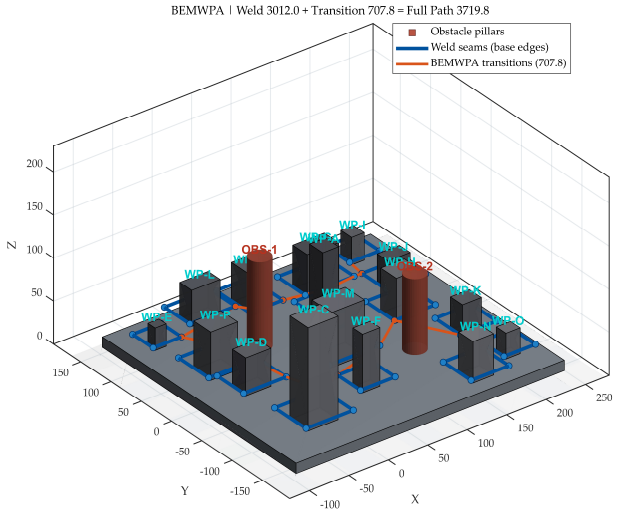


Figure 12. Complete welding path planned by BEMWPA in the expanded scenario with sixteen workpieces and two obstacle pillars (weld 3012.0 + transition 707.8 = full path 3719.8).

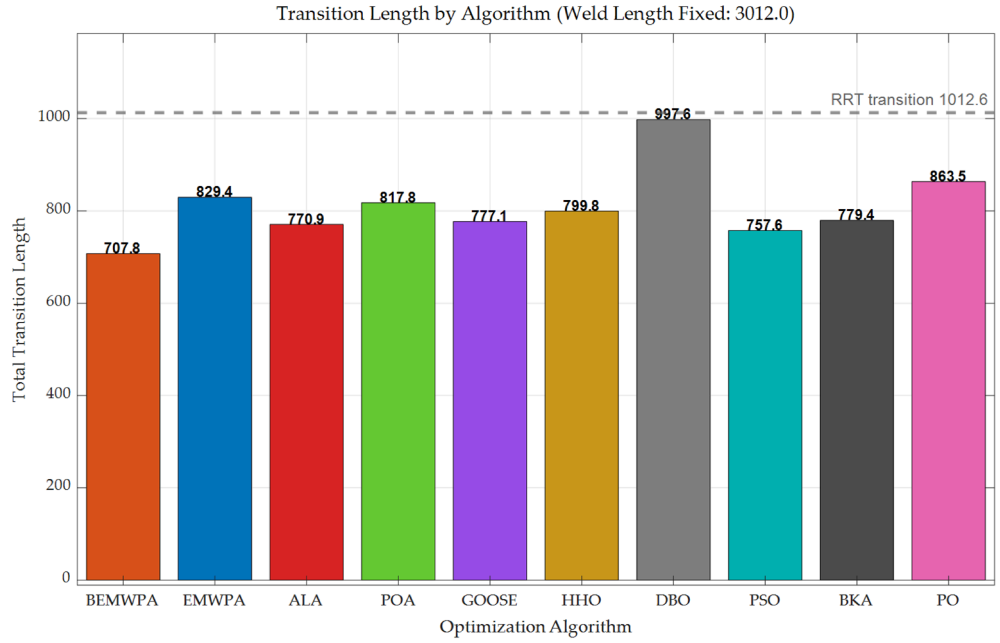


Figure 13. Comparison of the total inter-loop transition length of the ten algorithms in the expanded sixteen-workpiece, two-obstacle scenario (weld length fixed at 3012.0) (BEMWPA is the proposed algorithm).

Figure 14 provides a visual comparison of the transfer paths produced by the nine competing algorithms in the same expanded scenario. Consistent with the quantitative results, the comparison algorithms generate visibly longer or more tortuous transfer segments around the two obstacle pillars—DBO in particular produces markedly redundant detours (997.6). By contrast, the BEMWPA path in Figure 12 remains the most compact and smooth, further confirming the superior obstacle-avoidance efficiency and the generalization capability of the proposed method in complex multi-workpiece welding tasks.

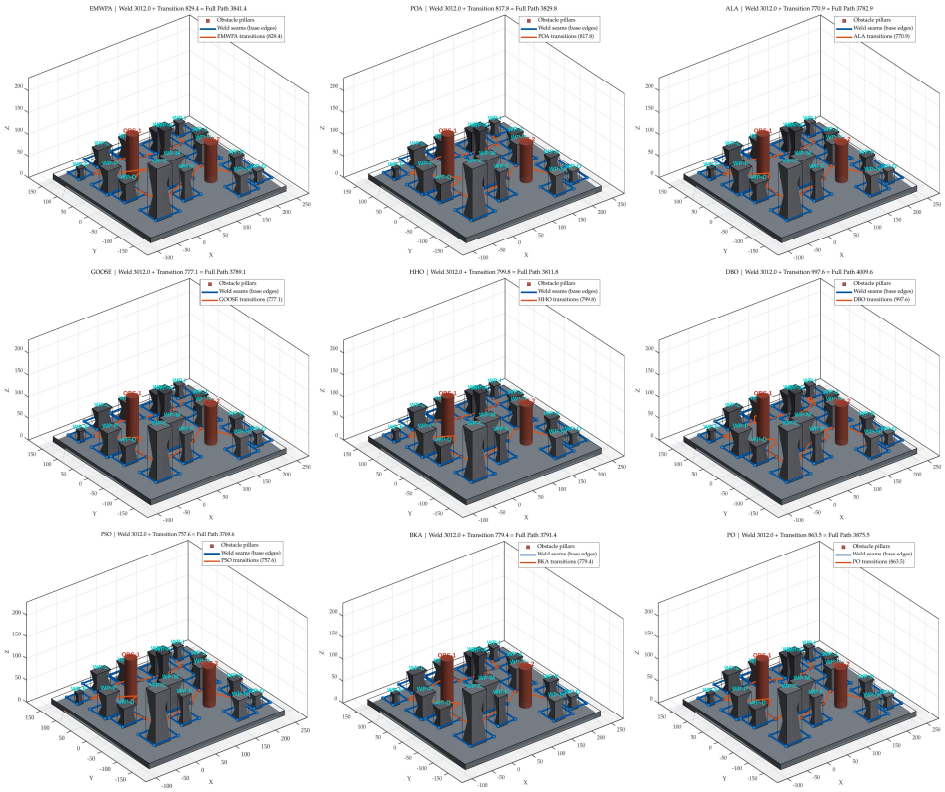


Figure 14. Visual comparison of the inter-loop transfer paths of the nine comparison algorithms in the expanded sixteen-workpiece, two-obstacle scenario.

5. Conclusions and Outlook

Aiming to solve the problems of premature convergence, insufficient population diversity, and imbalance between exploration and exploitation of the electromagnetic wave propagation optimization algorithm in high-dimensional complex optimization problems, this paper proposes a boosted electromagnetic wave propagation optimization algorithm, BEMWPA. Through the collaborative action of cubic chaotic initialization, nonlinear electromagnetic-field phase modulation, and a Beta-distribution opposition-based learning strategy, the proposed algorithm effectively improves the search-space coverage, the global exploration capability, and the local-exploitation stability. Systematic experimental results based on the IEEE CEC2017 standard test functions show that BEMWPA outperforms most comparison algorithms in terms of convergence accuracy, stability, and comprehensive ranking, verifying the effectiveness of the proposed improvement strategies on complex multimodal optimization problems.

In the path-planning experiment for the welding manipulator in complex multi-workpiece scenarios, this paper constructs a three-dimensional scenario containing 12 workpieces, 12 closed weld seams, and multiple obstacle constraints and combines BEMWPA with the RRT initial feasible path to continuously optimize the inter-seam transfer nodes. The experimental results show that BEMWPA-RRT reduces the inter-seam transfer path from 586.00 mm to 479.11 mm, representing a relative reduction of 18.24%. The complete path is reduced from 2974.00 mm to 2867.11 mm, representing a relative reduction of 3.59%. In addition, the optimized transfer path is only 1.59 mm longer than the obstacle-free ideal transfer length of 477.52 mm, indicating that the method can approach the geometric lower bound of the scenario while satisfying the obstacle-avoidance constraints. Kinematic verification on the seven-degrees-of-freedom welding manipulator further shows that the optimized end-effector path can be converted into a continuously executable joint-

space trajectory, providing a feasible basis for offline welding path planning of complex multi-workpiece tasks.

The main contributions of this paper are as follows:

(1) A population-initialization method based on the cubic chaotic map is proposed, which enhances the search-space coverage by improving the uniformity and diversity of the initial-solution distribution, thereby alleviating premature population convergence at its source.

(2) A search mechanism with differentiated electric- and magnetic-field driving is constructed, combined with a probabilistic switching strategy, to achieve dynamic coordination between global exploration and local exploitation. Among them, the electric-field driving focuses on expanding the search range, while the magnetic-field driving focuses on local fine search, thereby improving the adaptive optimization capability of the algorithm at different search stages.

(3) A Beta-distribution opposition-based learning strategy and perturbation-update mechanism are introduced, which enhance the population's ability to escape local optima by generating high-quality opposite candidate solutions, further improving the population diversity and global search performance of the algorithm.

In addition, the systematic experimental analysis based on the IEEE CEC2017 benchmark functions verifies the effectiveness of the proposed strategies. The manipulator path-planning application results show that BEMWPA can still maintain good optimization performance and stability in three-dimensional obstacle-avoidance path-optimization scenarios.

Although BEMWPA has achieved good experimental results in numerical optimization and complex multi-workpiece welding path planning, there is still room for further research. First, some key parameters in the current algorithm still rely on empirical settings; subsequent work may incorporate adaptive parameter-adjustment mechanisms so that the electric-field, magnetic-field, and probabilistic-switching parameters can be dynamically adjusted with the population state. Second, the current manipulator verification mainly focuses on path reachability, continuity, and interference-free visualization; future work may further introduce quantitative metrics such as joint velocity, acceleration, welding-torch pose constraints, and minimum safety distance, so as to form a more complete robot-execution performance-evaluation system. Third, our comparison set was deliberately restricted to EMWPA and eight recent (2023–2025) swarm/physics-inspired metaheuristics to keep the comparison within a consistent algorithm class; a comprehensive benchmark against strong adaptive differential-evolution-type optimizers such as DE, CMA-ES, and L-SHADE is another important direction of future work. In addition, BEMWPA can be extended to dynamic obstacles, multi-objective optimization, and real-time replanning scenarios, providing a more stable solution method for intelligent path planning in complex manufacturing environments.

Author Contributions: Conceptualization, C.J. and Y.L.; methodology, F.Y.; software, F.Y.; validation, X.G., Y.Y. and H.X.; formal analysis, F.Y.; investigation, X.G.; resources, M.F.; data curation, Y.Y.; writing—original draft preparation, F.Y.; writing—review and editing, Y.L. and C.J.; visualization, H.X.; supervision, Y.L.; project administration, M.F.; funding acquisition, C.J. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Open project of Anhui Dabieshan Academy of Traditional Chinese Medicine (TCMADM-2024-07), the University Key Research Project of Department of Education, Anhui Province (2024AH051994), the High-level Talent Start-up Project of West Anhui University (WGKQ2023003, WGKQ2021052), the Horizontal Project of West Anhui University (0045022007, 0045023151), and the University Innovation Team Project of Department of Education Anhui Province (2023AH010078).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author. The code developed in this study is not publicly available but can be provided by the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ALA	Artificial Lemming Algorithm
POA	Pelican Optimization Algorithm
GOOSE	GOOSE
HHO	Harris Hawks Optimization
DBO	Dung Beetle Optimizer
PSO	Particle Swarm Optimization
BKA	Black-winged Kite Algorithm
PO	Parrot Optimizer
EMWPA	Electromagnetic Wave Propagation Algorithm
BEMWPA	Boosted Electromagnetic Wave Propagation Algorithm
RRT	Rapidly-exploring Random Tree

Appendix A

Appendix A reports the complete per-function statistical results of BEMWPA and the nine comparison algorithms on the CEC2017 benchmark suite in 50 dimensions. For each of the 29 functions, the minimum (Min), mean (Avg), and standard deviation (Std) over 30 independent runs are listed, together with the rank of each algorithm based on the mean value. Tables A1–A3 correspond to functions F1–F10, F11–F20, and F21–F30, respectively.

Table A1. Statistical results for CEC2017 (F_1–F_{10}) in 50 dimensions.

Function	Metrics	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
F_1	Min	4.68×10^{4}	3.46×10^{7}	5.69×10^{2}	3.44×10^{10}	3.27×10^{4}	1.69×10^{8}	5.85×10^{10}	1.23×10^{9}	9.37×10^{9}	1.27×10^{10}
	Std	1.25×10^{7}	2.93×10^{8}	1.04×10^{4}	7.41×10^{9}	1.34×10^{4}	6.38×10^{7}	6.79×10^{9}	4.20×10^{9}	3.15×10^{10}	5.41×10^{9}
	Avg	5.95×10^{6}	2.72×10^{8}	9.24×10^{3}	4.56×10^{10}	5.30×10^{4}	2.77×10^{8}	7.03×10^{10}	6.28×10^{9}	3.28×10^{10}	2.18×10^{10}
	Rank	3	4	**1**	9	2	5	10	6	8	7
F_3	Min	2.09×10^{4}	6.24×10^{4}	4.10×10^{4}	6.32×10^{4}	1.18×10^{5}	1.12×10^{5}	1.64×10^{5}	8.92×10^{4}	4.32×10^{4}	7.89×10^{4}
	Std	1.62×10^{4}	1.24×10^{5}	5.09×10^{4}	1.90×10^{4}	5.99×10^{4}	1.39×10^{4}	2.11×10^{4}	2.72×10^{4}	5.17×10^{4}	1.25×10^{4}
	Avg	4.63×10^{4}	1.87×10^{5}	1.18×10^{5}	9.44×10^{4}	2.26×10^{5}	1.38×10^{5}	1.93×10^{5}	1.35×10^{5}	7.95×10^{4}	1.04×10^{5}
	Rank	**1**	8	5	3	10	7	9	6	2	4
F_4	Min	4.76×10^{2}	5.87×10^{2}	4.82×10^{2}	2.41×10^{3}	5.35×10^{2}	7.28×10^{2}	1.19×10^{4}	7.33×10^{2}	9.48×10^{2}	1.58×10^{3}
	Std	4.68×10^{1}	5.98×10^{1}	4.89×10^{1}	2.13×10^{3}	4.94×10^{1}	8.18×10^{1}	1.71×10^{3}	5.08×10^{2}	7.45×10^{3}	1.05×10^{3}
	Avg	5.66×10^{2}	7.05×10^{2}	5.66×10^{2}	6.83×10^{3}	5.94×10^{2}	8.63×10^{2}	1.54×10^{4}	1.26×10^{3}	4.35×10^{3}	2.70×10^{3}
	Rank	**1**	4	2	9	3	5	10	6	8	7
F_5	Min	7.12×10^{2}	8.03×10^{2}	7.96×10^{2}	8.58×10^{2}	8.74×10^{2}	8.79×10^{2}	1.07×10^{3}	7.68×10^{2}	8.19×10^{2}	9.03×10^{2}
	Std	6.47×10^{1}	4.87×10^{1}	5.36×10^{1}	2.99×10^{1}	1.82×10^{2}	3.34×10^{1}	2.36×10^{1}	6.28×10^{1}	1.27×10^{2}	4.18×10^{1}
	Avg	8.33×10^{2}	8.81×10^{2}	8.72×10^{2}	9.19×10^{2}	1.05×10^{3}	9.36×10^{2}	1.10×10^{3}	8.50×10^{2}	9.47×10^{2}	9.90×10^{2}
	Rank	**1**	4	3	5	9	6	10	2	7	8

Table A1. *Cont.*

Function	Metrics	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
F_6	Min	6.30×10^2	6.52×10^2	6.54×10^2	6.55×10^2	6.70×10^2	6.69×10^2	6.74×10^2	6.45×10^2	6.59×10^2	6.58×10^2
	Std	8.15×10^0	7.10×10^0	7.87×10^0	6.22×10^0	7.04×10^0	3.55×10^0	5.94×10^0	6.71×10^0	1.01×10^1	6.83×10^0
	Avg	6.55×10^2	6.63×10^2	6.65×10^2	6.69×10^2	6.81×10^2	6.77×10^2	6.84×10^2	6.56×10^2	6.72×10^2	6.75×10^2
	Rank	**1**	3	4	5	9	8	10	2	6	7
F_7	Min	1.17×10^3	1.56×10^3	1.51×10^3	1.53×10^3	1.69×10^3	1.78×10^3	1.68×10^3	1.09×10^3	1.50×10^3	1.59×10^3
	Std	1.50×10^2	1.59×10^2	8.15×10^1	8.27×10^1	1.65×10^3	6.81×10^1	4.75×10^1	1.56×10^2	9.50×10^1	9.35×10^1
	Avg	1.43×10^3	1.81×10^3	1.68×10^3	1.76×10^3	3.28×10^3	1.88×10^3	1.76×10^3	1.33×10^3	1.73×10^3	1.76×10^3
	Rank	2	8	3	5	10	9	6	**1**	4	7
F_8	Min	1.05×10^3	1.09×10^3	1.08×10^3	1.18×10^3	1.21×10^3	1.14×10^3	1.27×10^3	1.02×10^3	1.14×10^3	1.20×10^3
	Std	5.38×10^1	6.01×10^1	6.10×10^1	3.27×10^1	1.64×10^2	3.78×10^1	3.78×10^1	6.09×10^1	1.28×10^2	4.91×10^1
	Avg	1.13×10^3	1.19×10^3	1.22×10^3	1.24×10^3	1.41×10^3	1.21×10^3	1.40×10^3	1.13×10^3	1.28×10^3	1.31×10^3
	Rank	2	3	5	6	10	4	9	**1**	7	8
F_9	Min	5.59×10^3	9.16×10^3	9.72×10^3	1.28×10^4	1.31×10^4	2.44×10^4	2.61×10^4	8.59×10^3	1.23×10^4	1.61×10^4
	Std	2.89×10^3	1.92×10^3	2.39×10^3	1.85×10^3	6.00×10^3	3.08×10^3	3.75×10^3	2.93×10^3	4.85×10^3	3.99×10^3
	Avg	1.08×10^4	1.43×10^4	1.42×10^4	1.69×10^4	1.95×10^4	2.98×10^4	3.26×10^4	1.27×10^4	1.64×10^4	2.28×10^4
	Rank	**1**	4	3	6	7	9	10	2	5	8
F_{10}	Min	5.45×10^3	6.91×10^3	7.12×10^3	7.61×10^3	7.81×10^3	8.00×10^3	1.20×10^4	6.82×10^3	7.56×10^3	9.11×10^3
	Std	1.16×10^3	1.25×10^3	1.28×10^3	9.40×10^2	7.07×10^2	6.28×10^2	9.17×10^2	8.25×10^2	2.36×10^3	1.28×10^3
	Avg	8.46×10^3	8.80×10^3	9.09×10^3	9.19×10^3	8.96×10^3	8.93×10^3	1.41×10^4	8.12×10^3	9.93×10^3	1.12×10^4
	Rank	2	3	6	7	5	4	10	**1**	8	9

Bold values indicate the best result.

Table A2. Statistical results for CEC2017 (F_{11}–F_{20}) in 50 dimensions.

Function	Metrics	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
F_{11}	Min	1.26×10^3	1.42×10^3	1.25×10^3	2.80×10^3	1.26×10^3	1.53×10^3	9.17×10^3	1.49×10^3	1.61×10^3	2.12×10^3
	Std	6.75×10^1	3.71×10^2	8.66×10^1	1.71×10^3	1.50×10^2	1.78×10^2	2.34×10^3	3.50×10^2	5.52×10^3	1.80×10^3
	Avg	1.36×10^3	1.72×10^3	1.40×10^3	6.43×10^3	1.48×10^3	1.79×10^3	1.35×10^4	1.86×10^3	4.06×10^3	3.89×10^3
	Rank	**1**	4	2	9	3	5	10	6	8	7
F_{12}	Min	5.03×10^5	5.82×10^6	1.47×10^6	5.22×10^9	2.00×10^6	9.16×10^7	2.51×10^{10}	1.07×10^7	4.89×10^7	4.55×10^8
	Std	1.00×10^6	3.13×10^7	9.06×10^6	5.58×10^9	2.09×10^7	1.11×10^8	7.57×10^9	2.76×10^9	1.72×10^{10}	2.25×10^9
	Avg	2.24×10^6	2.48×10^7	6.85×10^6	1.47×10^{10}	1.95×10^7	2.20×10^8	3.48×10^{10}	1.90×10^9	7.16×10^9	2.40×10^9
	Rank	**1**	4	2	9	3	5	10	6	8	7
F_{13}	Min	1.76×10^3	1.11×10^4	7.43×10^3	5.57×10^6	3.04×10^4	1.79×10^6	5.61×10^9	1.71×10^4	6.13×10^5	2.13×10^6
	Std	2.00×10^3	9.20×10^3	1.89×10^4	5.15×10^9	1.71×10^5	4.08×10^6	5.75×10^9	6.86×10^8	2.00×10^9	4.39×10^8
	Avg	4.34×10^3	2.42×10^4	3.08×10^4	4.77×10^9	1.65×10^5	5.33×10^6	1.54×10^{10}	4.11×10^8	4.62×10^8	1.78×10^8
	Rank	**1**	2	3	9	4	5	10	7	8	6
F_{14}	Min	1.17×10^4	2.42×10^4	4.64×10^3	7.94×10^4	1.90×10^4	2.30×10^5	1.10×10^6	6.56×10^3	6.63×10^3	2.43×10^5
	Std	7.21×10^4	3.41×10^5	2.61×10^5	2.82×10^5	5.18×10^4	2.27×10^6	5.83×10^6	2.14×10^6	3.19×10^6	1.92×10^6
	Avg	7.29×10^4	3.30×10^5	1.50×10^5	3.03×10^5	9.89×10^4	2.35×10^6	7.83×10^6	8.08×10^5	8.33×10^5	2.20×10^6
	Rank	**1**	5	3	4	2	9	10	6	7	8

Table A2. *Cont.*

Function	Metrics	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
F_{15}	Min	1.69×10^3	2.34×10^3	3.38×10^3	5.83×10^4	1.38×10^4	3.95×10^5	8.19×10^8	7.73×10^3	3.28×10^4	6.82×10^4
	Std	3.35×10^3	5.25×10^3	1.17×10^4	3.32×10^8	3.79×10^4	2.99×10^5	8.05×10^8	1.59×10^7	6.27×10^8	5.83×10^6
	Avg	5.68×10^3	8.55×10^3	1.86×10^4	1.55×10^8	5.08×10^4	9.00×10^5	1.98×10^9	3.59×10^6	2.30×10^8	3.57×10^6
	Rank	**1**	2	3	8	4	5	10	7	9	6
F_{16}	Min	2.44×10^3	2.59×10^3	3.05×10^3	3.35×10^3	3.57×10^3	3.28×10^3	4.84×10^3	3.01×10^3	3.32×10^3	3.61×10^3
	Std	4.03×10^2	4.76×10^2	6.44×10^2	6.31×10^2	6.84×10^2	7.20×10^2	4.73×10^2	3.97×10^2	1.57×10^3	7.51×10^2
	Avg	3.45×10^3	3.58×10^3	4.05×10^3	4.56×10^3	4.62×10^3	4.36×10^3	5.58×10^3	3.78×10^3	4.86×10^3	4.84×10^3
	Rank	**1**	2	4	6	7	5	10	3	9	8
F_{17}	Min	2.89×10^3	2.80×10^3	2.99×10^3	2.76×10^3	3.20×10^3	2.79×10^3	4.09×10^3	2.85×10^3	3.18×10^3	3.66×10^3
	Std	2.38×10^2	4.25×10^2	4.98×10^2	5.40×10^2	4.82×10^2	5.19×10^2	5.08×10^2	3.27×10^2	6.17×10^2	4.16×10^2
	Avg	3.40×10^3	3.60×10^3	3.74×10^3	3.64×10^3	4.01×10^3	3.71×10^3	4.71×10^3	3.41×10^3	3.74×10^3	4.35×10^3
	Rank	**1**	3	7	4	8	5	10	2	6	9
F_{18}	Min	6.48×10^4	6.56×10^4	1.25×10^5	6.14×10^5	1.71×10^5	6.33×10^5	1.85×10^6	2.20×10^5	2.12×10^5	1.40×10^6
	Std	4.16×10^5	1.85×10^6	2.63×10^5	2.19×10^6	5.89×10^5	3.91×10^6	2.20×10^7	1.33×10^6	3.70×10^7	5.26×10^6
	Avg	4.56×10^5	1.33×10^6	4.03×10^5	2.98×10^6	1.00×10^6	4.95×10^6	1.96×10^7	1.40×10^6	1.75×10^7	8.79×10^6
	Rank	2	4	**1**	6	3	7	10	5	9	8
F_{19}	Min	5.07×10^3	2.95×10^3	2.58×10^3	3.99×10^5	4.08×10^5	3.10×10^5	1.83×10^8	5.86×10^3	7.85×10^4	2.85×10^5
	Std	6.93×10^3	1.46×10^4	2.03×10^4	1.03×10^8	1.17×10^6	1.11×10^6	6.48×10^8	1.29×10^6	1.63×10^8	2.50×10^7
	Avg	1.62×10^4	2.17×10^4	2.67×10^4	3.54×10^7	1.63×10^6	1.49×10^6	1.09×10^9	4.05×10^5	4.74×10^7	1.82×10^7
	Rank	**1**	2	3	8	6	5	10	4	9	7
F_{20}	Min	2.60×10^3	2.78×10^3	3.03×10^3	2.59×10^3	3.29×10^3	3.06×10^3	3.36×10^3	2.54×10^3	2.56×10^3	3.06×10^3
	Std	3.97×10^2	3.53×10^2	5.06×10^2	2.50×10^2	3.45×10^2	3.09×10^2	2.77×10^2	3.38×10^2	2.64×10^2	2.63×10^2
	Avg	3.37×10^3	3.44×10^3	3.73×10^3	3.14×10^3	3.98×10^3	3.58×10^3	3.79×10^3	3.21×10^3	3.15×10^3	3.51×10^3
	Rank	4	5	8	**1**	10	7	9	3	2	6

Bold values indicate the best result.

Table A3. Statistical results for CEC2017 (F_{21}–F_{30}) in 50 dimensions.

Function	Metrics	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
F_{21}	Min	2.54×10^3	2.70×10^3	2.56×10^3	2.69×10^3	2.90×10^3	2.72×10^3	2.83×10^3	2.57×10^3	2.74×10^3	2.67×10^3
	Std	4.92×10^1	8.95×10^1	1.50×10^2	6.13×10^1	9.74×10^1	1.01×10^2	4.80×10^1	8.20×10^1	7.28×10^1	8.28×10^1
	Avg	2.61×10^3	2.82×10^3	2.87×10^3	2.80×10^3	3.10×10^3	2.88×10^3	2.94×10^3	2.68×10^3	2.84×10^3	2.80×10^3
	Rank	**1**	5	7	4	10	8	9	2	6	3
F_{22}	Min	8.63×10^3	2.47×10^3	8.20×10^3	1.05×10^4	9.50×10^3	9.91×10^3	1.37×10^4	8.80×10^3	8.68×10^3	1.14×10^4
	Std	8.74×10^2	2.07×10^3	1.21×10^3	6.58×10^2	8.32×10^2	9.94×10^2	1.17×10^3	8.73×10^2	1.81×10^3	1.07×10^3
	Avg	1.03×10^4	1.05×10^4	1.08×10^4	1.13×10^4	1.07×10^4	1.16×10^4	1.56×10^4	1.01×10^4	1.07×10^4	1.31×10^4
	Rank	2	3	6	7	5	8	10	**1**	4	9
F_{23}	Min	2.96×10^3	3.28×10^3	3.10×10^3	3.28×10^3	3.95×10^3	3.65×10^3	3.56×10^3	3.27×10^3	3.36×10^3	3.27×10^3
	Std	1.51×10^2	2.68×10^2	1.99×10^2	1.24×10^2	1.75×10^2	1.83×10^2	1.89×10^2	1.59×10^2	2.11×10^2	1.23×10^2
	Avg	3.18×10^3	3.77×10^3	3.52×10^3	3.51×10^3	4.24×10^3	3.96×10^3	3.87×10^3	3.57×10^3	3.71×10^3	3.53×10^3
	Rank	**1**	7	3	2	10	9	8	5	6	4

Table A3. *Cont.*

Function	Metrics	BEMWPA	EMWPA	ALA	POA	GOOSE	HHO	DBO	PSO	BKA	PO
F_{24}	Min	3.20 × 10^3	3.67 × 10^3	3.31 × 10^3	3.39 × 10^3	4.16 × 10^3	4.12 × 10^3	3.84 × 10^3	3.46 × 10^3	3.46 × 10^3	3.38 × 10^3
	Std	1.18 × 10^2	3.26 × 10^2	1.69 × 10^2	1.33 × 10^2	1.37 × 10^2	2.13 × 10^2	2.02 × 10^2	1.80 × 10^2	2.52 × 10^2	1.26 × 10^2
	Avg	3.38 × 10^3	4.19 × 10^3	3.66 × 10^3	3.68 × 10^3	4.40 × 10^3	4.43 × 10^3	4.15 × 10^3	3.66 × 10^3	3.85 × 10^3	3.64 × 10^3
	Rank	**1**	8	3	5	9	10	7	4	6	2
F_{25}	Min	3.03 × 10^3	3.12 × 10^3	3.06 × 10^3	5.04 × 10^3	3.05 × 10^3	3.16 × 10^3	1.00 × 10^4	3.24 × 10^3	3.39 × 10^3	4.02 × 10^3
	Std	3.35 × 10^1	4.47 × 10^1	2.08 × 10^1	9.57 × 10^2	2.25 × 10^1	5.87 × 10^1	8.14 × 10^2	2.13 × 10^2	1.52 × 10^3	3.99 × 10^2
	Avg	3.10 × 10^3	3.18 × 10^3	3.09 × 10^3	6.32 × 10^3	3.11 × 10^3	3.25 × 10^3	1.12 × 10^4	3.59 × 10^3	4.85 × 10^3	4.63 × 10^3
	Rank	2	4	**1**	9	3	5	10	6	8	7
F_{26}	Min	2.91 × 10^3	4.09 × 10^3	3.84 × 10^3	1.05 × 10^4	6.18 × 10^3	6.17 × 10^3	1.27 × 10^4	6.33 × 10^3	1.15 × 10^4	6.45 × 10^3
	Std	3.59 × 10^3	2.18 × 10^3	2.40 × 10^3	1.17 × 10^3	3.86 × 10^3	1.44 × 10^3	5.54 × 10^2	1.58 × 10^3	2.02 × 10^3	2.18 × 10^3
	Avg	9.10 × 10^3	1.17 × 10^4	1.07 × 10^4	1.27 × 10^4	1.47 × 10^4	1.08 × 10^4	1.38 × 10^4	1.06 × 10^4	1.39 × 10^4	1.23 × 10^4
	Rank	**1**	5	3	7	10	4	8	2	9	6
F_{27}	Min	3.42 × 10^3	3.62 × 10^3	**3.38 × 10^3**	3.76 × 10^3	4.11 × 10^3	3.92 × 10^3	4.57 × 10^3	3.55 × 10^3	3.78 × 10^3	3.55 × 10^3
	Std	2.31 × 10^2	3.81 × 10^2	3.00 × 10^2	4.00 × 10^2	8.46 × 10^2	4.66 × 10^2	2.69 × 10^2	**1.64 × 10^2**	3.35 × 10^2	3.88 × 10^2
	Avg	3.78 × 10^3	4.11 × 10^3	3.82 × 10^3	4.24 × 10^3	5.71 × 10^3	4.53 × 10^3	5.09 × 10^3	3.80 × 10^3	4.21 × 10^3	4.13 × 10^3
	Rank	**1**	4	3	7	10	8	9	2	6	5
F_{28}	Min	3.29 × 10^3	3.33 × 10^3	**3.29 × 10^3**	5.18 × 10^3	3.35 × 10^3	3.59 × 10^3	6.68 × 10^3	3.74 × 10^3	3.67 × 10^3	4.41 × 10^3
	Std	**4.11 × 10^1**	7.25 × 10^1	4.12 × 10^1	1.05 × 10^3	2.13 × 10^2	1.38 × 10^2	8.07 × 10^2	6.97 × 10^2	1.87 × 10^3	5.08 × 10^2
	Avg	3.36 × 10^3	3.48 × 10^3	3.35 × 10^3	6.94 × 10^3	3.51 × 10^3	3.85 × 10^3	8.14 × 10^3	4.68 × 10^3	5.28 × 10^3	5.21 × 10^3
	Rank	2	3	**1**	9	4	5	10	6	8	7
F_{29}	Min	**4.04 × 10^3**	4.52 × 10^3	4.77 × 10^3	5.27 × 10^3	5.28 × 10^3	5.46 × 10^3	7.19 × 10^3	5.02 × 10^3	4.85 × 10^3	5.55 × 10^3
	Std	**4.09 × 10^2**	4.82 × 10^2	5.09 × 10^2	1.05 × 10^3	6.58 × 10^2	4.45 × 10^2	1.79 × 10^3	5.19 × 10^2	4.62 × 10^3	1.11 × 10^3
	Avg	4.81 × 10^3	5.43 × 10^3	5.45 × 10^3	6.97 × 10^3	6.55 × 10^3	6.18 × 10^3	1.02 × 10^4	5.58 × 10^3	7.83 × 10^3	7.26 × 10^3
	Rank	**1**	2	3	7	6	5	10	4	9	8
F_{30}	Min	7.80 × 10^5	1.09 × 10^6	**7.56 × 10^5**	5.08 × 10^7	3.30 × 10^7	3.52 × 10^7	5.27 × 10^8	8.22 × 10^6	1.98 × 10^7	7.48 × 10^7
	Std	**3.03 × 10^5**	1.57 × 10^6	2.26 × 10^6	8.82 × 10^7	9.97 × 10^6	1.19 × 10^7	1.10 × 10^9	1.16 × 10^8	2.04 × 10^7	9.62 × 10^7
	Avg	1.08 × 10^6	2.88 × 10^6	2.68 × 10^6	2.37 × 10^8	4.59 × 10^7	5.33 × 10^7	1.93 × 10^9	4.75 × 10^7	4.37 × 10^7	2.04 × 10^8
	Rank	**1**	3	2	9	5	7	10	6	4	8

Bold values indicate the best result.

References

1. Houssein, E.H.; Saeed, M.K.; Hu, G.; Al-Sayed, M.M. Metaheuristics for Solving Global and Engineering Optimization Problems: Review, Applications, Open Issues and Challenges. *Arch. Comput. Methods Eng.* **2024**, *31*, 4485–4519. [CrossRef]
2. Tang, J.; Liu, G.; Pan, Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1627–1643. [CrossRef]
3. Alorf, A. A survey of recently developed metaheuristics and their comparative analysis. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105622. [CrossRef]
4. Mazaheri, H.; Goli, S.; Nourollah, A. A survey of 3D space path-planning methods and algorithms. *ACM Comput. Surv.* **2024**, *57*, 1–32. [CrossRef]
5. Abdullah, M.; Alamgeer, K.S.; Hafeez, G.; Alghamdi, B.; Alsafran, A.S.; Babar, M.Z. Electromagnetic wave propagation algorithm: A novel electromagnetic wave propagation-inspired optimizer for engineering applications. *Ain Shams Eng. J.* **2025**, *16*, 103615. [CrossRef]

6. Limane, A.; Zitouni, F.; Harous, S.; Lakbichi, R.; Ferhat, A.; Almazyad, A.S.; Jangir, P.; Mohamed, A.W. Chaos-enhanced metaheuristics: Classification, comparison, and convergence analysis. *Complex Intell. Syst.* **2025**, *11*, 177. [CrossRef]
7. Zelinka, I.; Diep, Q.B.; Snášel, V.; Das, S.; Innocenti, G.; Tesi, A.; Schoen, F.; Kuznetsov, N.V. Impact of chaotic dynamics on the performance of metaheuristic optimization algorithms: An experimental analysis. *Inf. Sci.* **2022**, *587*, 692–719. [CrossRef]
8. Kvitko, D.; Rybin, V.; Bayazitov, O.; Karimov, A.; Karimov, T.; Butusov, D. Chaotic path-planning algorithm based on Courbage–Nekorkin artificial neuron model. *Mathematics* **2024**, *12*, 892. [CrossRef]
9. Lakbichi, R.; Zitouni, F.; Harous, S.; Ferhat, A.; Limane, A.; Almazyad, A.S.; Xiong, G.; Mohamed, A.W. Opposition-based learning techniques in metaheuristics: Classification, comparison, and convergence analysis. *PeerJ Comput. Sci.* **2025**, *11*, e2935. [CrossRef] [PubMed]
10. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M. Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **2008**, *12*, 64–79. [CrossRef]
11. Kennedy, J.; Eberhart, R. *Particle Swarm Optimization*; IEEE: New York, NY, USA, 1995; pp. 1942–1948. [CrossRef]
12. Xiao, Y.; Cui, H.; Khurma, R.A.; Castillo, P.A. Artificial lemming algorithm: A novel bionic meta-heuristic technique for solving real-world engineering optimization problems. *Artif. Intell. Rev.* **2025**, *58*, 1. [CrossRef]
13. Dai, G.; Zhang, Q.; Xu, B. A novel framework for trajectory planning in robotic arm developed by integrating dynamical movement primitives with particle swarm optimization. *Sci. Rep.* **2025**, *15*, 29656. [CrossRef] [PubMed]
14. Chen, G.; Luo, N.; Liu, D.; Zhao, Z.; Liang, C. Path planning for manipulators based on an improved probabilistic roadmap method. *Robot. Comput.-Integr. Manuf.* **2021**, *72*, 102196. [CrossRef]
15. Sepehri, A.; Moghaddam, A.M. A motion planning algorithm for redundant manipulators using rapidly exploring randomized trees and artificial potential fields. *IEEE Access* **2021**, *9*, 26059–26070. [CrossRef]
16. Gao, Q.; Yuan, Q.; Sun, Y.; Xu, L. Path planning algorithm of robot arm based on improved RRT* and BP neural network algorithm. *J. King Saud. Univ.-Comput. Inf. Sci.* **2023**, *35*, 101650. [CrossRef]
17. Qi, J.; Yuan, Q.; Wang, C.; Du, X.; Du, F.; Ren, A. Path planning and collision avoidance based on the RRT* FN framework for a robotic manipulator in various scenarios. *Complex Intell. Syst.* **2023**, *9*, 7475–7494. [CrossRef]
18. Merckaert, K.; Convens, B.; Nicotra, M.M.; Vanderborght, B. Real-time constraint-based planning and control of robotic manipulators for safe human–robot collaboration. *Robot. Comput.-Integr. Manuf.* **2024**, *87*, 102711. [CrossRef]
19. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
20. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control. Eng.* **2020**, *8*, 22–34. [CrossRef]
21. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]
22. Yang, X.-S.; Deb, S. *Cuckoo Search via Lévy Flights*; IEEE: New York, NY, USA, 2009; pp. 210–214. [CrossRef]
23. Wu, N.; Jia, D.; Li, Z.; He, Z. Trajectory planning of robotic arm based on particle swarm optimization algorithm. *Appl. Sci.* **2024**, *14*, 8234. [CrossRef]
24. Du, J.; Hou, J.; Wang, H.; Chen, Z. Application of an improved whale optimization algorithm in time-optimal trajectory planning for manipulators. *Math. Biosci. Eng.* **2023**, *20*, 16304–16329. [CrossRef] [PubMed]
25. Zhou, Y.; Han, G.; Wei, Z.; Huang, Z.; Chen, X.; Wu, J. Optimal trajectory planning of robot energy consumption based on improved sparrow search algorithm. *Meas. Control.* **2024**, *57*, 1014–1021. [CrossRef]
26. Liu, J.; Wang, H.; Li, X.; Chen, K.; Li, C. Robotic arm trajectory optimization based on multiverse algorithm. *Math Biosci. Eng.* **2023**, *20*, 2776–2792. [CrossRef] [PubMed]
27. Zhang, L.; Wang, Y.; Zhao, X.; Zhao, P.; He, L. Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm. *J. Mech. Sci. Technol.* **2021**, *35*, 3171–3181. [CrossRef]
28. Awad, N.H.; Ali, M.Z.; Liang, J.J.; Qu, B.Y.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report; Nanyang Technological University: Singapore, 2016. Available online: https://github.com/P-N-Suganthan/CEC2017-BoundContrained (accessed on 7 August 2026).
29. Trojovský, P.; Dehghani, M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors* **2022**, *22*, 855. [CrossRef] [PubMed]
30. Hamad, R.K.; Rashid, T.A. GOOSE algorithm: A powerful optimization tool for real-world engineering challenges and beyond. *Evol. Syst.* **2024**, *15*, 1249–1274. [CrossRef]
31. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
32. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [CrossRef]

33. Wang, J.; Wang, W.-C.; Hu, X.-X.; Qiu, L.; Zang, H.-F. Black-winged kite algorithm: A nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artif. Intell. Rev.* **2024**, *57*, 98. [CrossRef]
34. Lian, J.; Hui, G.; Ma, L.; Zhu, T.; Wu, X.; Heidari, A.A.; Chen, Y.; Chen, H. Parrot optimizer: Algorithm and applications to medical problems. *Comput. Biol. Med.* **2024**, *172*, 108064. [CrossRef] [PubMed]
35. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2020**, *56*, 100693. [CrossRef]
36. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
37. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Technical Report TR 98-11; Computer Science Department, Iowa State University: Ames, IA, USA, 1998.
38. Kuffner, J.J.; LaValle, S.M. *RRT-Connect: An Efficient Approach to Single-Query Path Planning*; IEEE: New York, NY, USA, 2000; pp. 995–1001. [CrossRef]
39. Gammell, J.D.; Barfoot, T.D.; Srinivasa, S.S. Batch informed trees (bit*): Informed asymptotically optimal anytime search. *Int. J. Robot. Res.* **2020**, *39*, 543–567. [CrossRef]
40. Nakamura, Y.; Hanafusa, H. Inverse kinematic solutions with singularity robustness for robot manipulator control. *J. Dyn. Syst. Meas. Control.* **1986**, *108*, 163–171. [CrossRef]
41. Chiaverini, S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans. Robot. Autom.* **1997**, *13*, 398–410. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.