

Article

Adaptive Bitterling Fish Optimization with Evolutionary Game Theory: For Cross-Regional Emergency Repair Path Planning

Shuangqing Chen ¹, Chao Chen ^{1,*}, Junfei Liu ², Xingwang Wang ³, Zhe Xu ⁴, Yongbin Liu ⁴, Haibin Liang ⁴, Lulu Zhang ¹ and Yaqian Liu ¹

¹ Key Laboratory of Enhanced Oil and Gas Recovery, Ministry of Education, Northeast Petroleum University, Daqing 163318, China

² Daqing Longfeng Gas Branch of Kunlun Gas Co., Ltd., Daqing 163000, China

³ Exploration and Production Research Institute, SINOPEC, Beijing 102206, China

⁴ Gas Technology Institute of Petrochina Kunlun Gas Co., Ltd., Harbin 150001, China

* Correspondence: cchao0317@163.com

Abstract

Modern energy internets and large-scale industrial systems are becoming increasingly complex. Consequently, the rapid response capability of energy infrastructure during sudden failures has become a core element to ensure the stable operation of the social economy. Emergency repair path planning (ERPP) is a complex nonlinear combinatorial optimization problem. It is characterized by dynamic uncertainties, such as fluctuating task durations and variable traffic accessibility. This paper proposes a cross-regional emergency repair path planning (CR-ERPP) optimization model considering dynamic path conditions. The model takes into account jurisdiction ownership, cross-regional dispatch costs, path weights (congestion coefficient, grade coefficient, quality coefficient) and accident risk levels. The primary objective of this model is to minimize the total repair cost. Furthermore, an Adaptive Bitterling Fish Optimization with Evolutionary Game Theory (ABFO-EGT) is developed. It introduces adaptive mechanisms, evolutionary game theory, and a symmetric mutation strategy. These enhancements are designed to overcome the inherent limitations of traditional swarm intelligence algorithms, namely unbalanced search behavior and premature convergence to local optima. Performance analysis demonstrates that the ABFO-EGT algorithm exhibits superior convergence stability and global search capability. Case study results show that the proposed method significantly reduces the total repair cost. Specifically, the cost is reduced by 33.2% compared to manual decision-making, 27.6% compared to the GWO algorithm, and 9.1% compared to both the ACO and PSO algorithms. This study provides an efficient and reliable decision support tool for emergency management of large-scale energy systems.

Keywords: energy infrastructure; cross-regional emergency repair path planning; ABFO-EGT; evolutionary game theory; symmetric mutation strategy



Academic Editors: Guangdong Tian and Sergei Odintsov

Received: 9 May 2026

Revised: 15 July 2026

Accepted: 20 July 2026

Published: 22 July 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

1.1. Background of Cross-Regional Emergency Repair for Energy Infrastructure

With the increasing complexity of modern energy internet and large-scale industrial systems, the construction and maintenance of energy infrastructure have become arduous tasks [1–4]. More importantly, the rapid response capability of energy infrastructure in the face of sudden failures has become a core element ensuring the stable operation of the social

economy. The optimization of emergency repair path planning first identifies candidate repair teams. This identification relies on accident locations, jurisdiction ownership, and cross-regional dispatch conditions. Subsequently, candidate paths are comprehensively evaluated. This evaluation integrates road traffic conditions and accident risk levels. Finally, the optimal path scheme is selected. It minimizes the total repair cost while satisfying both cross-regional dispatch and road traffic constraints. As a critical link in the emergency management system, emergency repair path planning directly determines whether repair teams can arrive at the accident sites and implement repairs in the shortest time, thereby minimizing energy losses, environmental risks, and negative social impacts [5–8]. However, the actual repair environment is typically highly dynamic and uncertain. The duration of repair tasks, transportation network accessibility, and travel times are all subject to change. Meanwhile, the spatial distribution of multiple accident sites significantly increases the difficulty of dispatch and route coordination. This complexity endows the problem with typical stochastic and strongly constrained characteristics. Consequently, it is generally formulated as a complex nonlinear combinatorial optimization problem [9–11].

1.2. Existing Optimization Studies in Path Planning

In existing research, path planning models have evolved from simple shortest-path searches to optimization models that account for various complex constraints. Shin et al. proposed a mathematical model based on Mixed-Integer Linear Programming (MILP) and an Ant Colony Optimization (ACO) algorithm for optimal scheduling, which considers the constraints of post-disaster reconstruction and material transportation [12]. Choosumrong et al. developed a Spatial Decision Support System (SDSS) for emergency vehicle routing that incorporates geospatial information and real-time traffic conditions [13]. In their study on post-disaster road network restoration, Li et al. constructed a nonlinear model integrating logistics support, crew scheduling, and path planning [14]. Although these mathematical and spatial models offer a rigorous framework for handling constraints, they possess a critical limitation. When the problem scale expands, path weights change dynamically, or cross-regional coordination demands intensify, their computational complexity escalates significantly. Consequently, their applicability in complex real-world scenarios becomes limited.

1.3. Existing Intelligent Algorithms

To address such path planning models, metaheuristic algorithms have been widely adopted by researchers [15,16]. Swarm intelligence algorithms exhibit great flexibility in handling complex non-convex problems. Consequently, methods such as Particle Swarm Optimization (PSO) [17], Gray Wolf Optimizer (GWO) [18], and Golden Jackal Optimization (GJO) [19] have been successively applied to the field of path optimization. Nevertheless, emergency repair path planning models often involve complex road weight constraints and multi-dimensional decision variables. When dealing with such models, traditional swarm intelligence algorithms often face common challenges, including loss of population diversity, premature convergence to local optima, and insufficient convergence accuracy [20–23]. That is to say, traditional swarm intelligence algorithms excel at exploring wide solution spaces. However, their capabilities are limited in emergency scheduling. In such highly constrained and strongly coupled scenarios, their exploitation capability and convergence robustness remain insufficient.

Recently, an increasing number of studies have focused on multi-strategy fusion and hybrid mechanisms to overcome the limitations of traditional intelligence algorithms in complex environments. Specifically, embedding strategies such as chaotic mappings, opposition-based learning, symmetry enhancement, or adaptive mutations into the evolu-

tionary mechanism has been widely proven effective. These operators successfully break local optima traps and enhance population diversity, further boosting the capacity to solve highly constrained combinatorial optimization tasks [24–31]. For instance, in discrete combinatorial optimization scenarios, the set-based differential evolution algorithm incorporating both exponential and binomial crossover strategies has been proposed. Experimental results on the Traveling Salesman Problem demonstrate that this algorithm achieves superior solution accuracy and convergence performance compared to traditional discrete swarm intelligence algorithms [32]. Regarding continuous global optimization tasks, the Adaptive Stagnation Guided Differential Evolution (ASGDE) algorithm constructs a multi-layer adaptive fusion framework that effectively balances global exploration and local exploitation. This algorithm exhibits outstanding competitiveness in both high-dimensional multimodal benchmarks and practical engineering optimization problems [33]. These cutting-edge advancements fully confirm the tremendous potential of multi-strategy fusion and hybrid mechanisms in addressing complex engineering problems.

1.4. Bitterling Fish Optimization

Bitterling Fish Optimization (BFO) is an emerging swarm intelligence algorithm inspired by the unique reproductive and foraging behaviors of bitterling fish in nature [34]. This algorithm demonstrates exceptional global spatial exploration capabilities and rapid convergence potential. However, the standard BFO algorithm relies on a fixed search step size and a static position update mechanism. This reliance causes an inherent imbalance between exploration and exploitation. Consequently, the algorithm often suffers from premature convergence when solving complex constrained engineering optimization problems with dynamic weights.

1.5. Novelty and Contributions

To overcome the deficiencies of existing research, a cross-regional emergency repair path planning optimization model is constructed. This model specifically addresses energy infrastructure emergency scenarios involving multiple accident points and multiple repair teams. Furthermore, this paper proposes an Adaptive Bitterling Fish Optimization with Evolutionary Game Theory (ABFO-EGT). The highlights and main contributions of this paper are presented as follows.

- (1) Construction of a cross-regional emergency repair path planning optimization model considering dynamic path conditions

This model incorporates jurisdiction ownership, cross-regional dispatch costs, path weights, and accident risk levels into the objective function. The path weights specifically account for congestion, grade, and quality coefficients. This comprehensive approach achieves a mathematical representation that closely aligns with real-world repair scenarios.

- (2) Proposal of the Adaptive Bitterling Fish Optimization with Evolutionary Game Theory (ABFO-EGT)

The ABFO-EGT algorithm introduces adaptive mechanisms, evolutionary game theory, and a symmetric mutation strategy. These enhancements effectively address the search imbalance and the tendency of the traditional BFO algorithm to fall into local optima. Its optimization accuracy and robustness in high-dimensional weighted path spaces are remarkably improved.

- (3) Systematic verification of algorithmic performance and engineering case analysis

Extensive comparisons with 10 mainstream algorithms on the CEC 2017 benchmark functions verify the proposed method. The results demonstrate that the ABFO-EGT algorithm achieves excellent solution performance and robustness. Furthermore, an actual gas

pipeline network leakage accident is utilized as a case study. The emergency repair path obtained by the proposed algorithm significantly reduces the total repair cost. Specifically, the cost is reduced by 33.2% compared to manual decision-making, 27.6% compared to the GWO algorithm, and 9.1% compared to both the ACO and PSO algorithms.

2. Optimization Model Construction

Optimization modeling is essential for cross-regional emergency repair path planning of energy infrastructure. Empirical scheduling and classical path optimization approaches cannot simultaneously account for dynamic road conditions, cross-regional dispatch costs, and accident risk levels. As a result, they fail to achieve globally optimal resource allocation. The core objective of the proposed model is to minimize the total comprehensive repair cost under multiple constraints. This total cost comprises in-jurisdiction repair expenses and cross-regional dispatch repair costs.

Compared with conventional path planning problems, this task presents three core challenges. First, dynamic path weights break the spatial symmetry of network topology. The problem thus becomes a highly constrained optimization task on asymmetric directed graphs, rendering most classical path algorithms built on symmetric assumptions invalid. Second, the two-layer optimization structure of team assignment and path selection constitutes a typical NP-hard combinatorial optimization problem. Third, multiple strong constraints drastically compress the feasible region. This imposes higher demands on the constraint-handling capability of solution algorithms.

The accident emergency repair path planning optimization model proposed in this paper is established based on the following assumptions.

- (1) There exists at least one accessible path between each accident site and each repair team.
- (2) The occurrence of accident sites is random and requires repair by repair teams.

2.1. Non-Symmetric Weight Representation of the Model

In traditional path planning problems, physical network topologies usually exhibit spatial symmetry. The geometric distance between node i and node j satisfies the symmetry property. This is shown in Equation (1).

$$d_{ij} = d_{ji}, \quad \forall i, j \in V \quad (1)$$

However, this geometric symmetry is completely broken in cross-regional emergency repair scenarios under dynamic environments. Let $w_{ij}(t)$ denote the comprehensive path weight at time t . This weight integrates three factors: congestion coefficient, road grade coefficient, and pavement quality coefficient. Due to direction-dependent dynamic traffic flows and real-time sudden risks, the actual weight matrix shows significant non-symmetry. This is presented in Equation (2).

$$w_{ij}(t) \neq w_{ji}(t) \quad (2)$$

Therefore, the cross-regional emergency repair path planning (CR-ERPP) model is mathematically formulated as a highly constrained optimization problem based on a non-symmetric directed graph.

2.2. Objective Function

The total cost of accident point repair by dispatched teams is required to be minimized in the optimization model on the basis of weighted emergency repair path planning. The total emergency repair cost is mainly composed of two parts. These include the repair costs

for accidents within the teams' jurisdictions and the repair costs for accidents outside their jurisdictions. The specific objective function is presented in Equation (3).

$$\min F = \sum_{i=1}^m \sum_{j=1}^n \zeta_{ij} \times [D_{(i,j)} + (1 - \sigma_{ij}) \cdot C_{ij} \times (1 - \phi_j) \cdot D_{(i,j)}] \quad (3)$$

In the formula, $D_{(i,j)}$ is the weighted total length from repair team i to accident point j ; ζ_{ij} indicates a binary variable indicating whether repair team i is assigned to repair accident point j , defined as: $\zeta_{ij} = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; σ_{ij} indicates whether accident point j is within the jurisdiction of repair team i , defined as: $\sigma_{ij} = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; C_{ij} represents the cost-effectiveness of repair team i conducting cross-regional repair on accident point j ; ϕ_j denotes the importance level of accident point j , defined as: $\phi_j = \begin{cases} 1, & \text{if } g(x_j) \in \{\text{medium, high risk}\}; \\ 0, & \text{if } g(x_j) \in \{\text{low risk}\}; \end{cases}$ where $g(x_j)$ represents the risk level evaluation of accident j ; m is defined as the number of repair teams; and n is defined as the number of accident points.

Notably, the determination of $D_{(i,j)}$ is formulated as an optimization problem, and the corresponding objective functions for solving $D_{(i,j)}$ are presented in Equations (4)–(6).

$$\min D_{(i,j)} = \sum_{k=1}^M \gamma_k \cdot L_k \cdot w_k \quad (4)$$

$$w_k = w_k^1 + w_k^2 + w_k^3 = \sum_{e=1}^3 w_k^e \quad (5)$$

$$L_k = \sum_{h=1}^{H-1} \beta_h \cdot \beta_{h+1} \cdot l_{h,h+1} \quad (6)$$

In the formula, M is defined as the count of all connected road segments from the location of repair team i to accident point j ; γ_k is a binary variable indicating whether the k -th path is selected, defined as: $\gamma_k = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; L_k denotes the distance of the k -th path; w_k is defined as the total road weight for the k -th path; w_k^1 denotes the road congestion coefficient for the k -th path; w_k^2 is the road grade coefficient for the k -th path; w_k^3 is the road quality coefficient for the k -th path; H denotes the number of all routing points from the location of repair team i to accident point j ; h represents the h -th routing point; β_h is a binary variable indicating whether the h -th routing point is selected, defined as: $\beta_h = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; β_{h+1} indicates whether the $(h+1)$ -th routing point is selected, defined as: $\beta_{h+1} = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; and $l_{h,h+1}$ represents the distance between the h -th routing point and the $(h+1)$ -th routing point.

2.3. Constraints

2.3.1. Repair Team Scheduling Constraints

When multiple accidents occur in a certain area, the scheduling of repair teams must satisfy that each team can only handle one accident. The specific formula is presented in Equation (7).

$$\sum_{i=1}^m \zeta_{ij} = 1 \quad (7)$$

where ζ_{ij} is a binary variable indicating whether repair team i is assigned to repair accident point j , defined as: $\zeta_{ij} = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; m is defined as the number of repair teams.

2.3.2. Path Weight Constraints

Each weight of the path takes a value between 0 and 1. The specific formulas are presented in Equations (8)–(10).

$$w_k^1 \in [0, 1] \quad (8)$$

$$w_k^2 \in [0, 1] \quad (9)$$

$$w_k^3 \in [0, 1] \quad (10)$$

where w_k^1 represents the road congestion coefficient for the k -th path; w_k^2 represents the road grade coefficient for the k -th path; and w_k^3 represents the road quality coefficient for the k -th path.

2.3.3. Path Selection Constraints

When multiple paths exist from a repair team to an accident point, the repair team can only select one path. The specific formula is presented in Equation (11).

$$\sum_{k=1}^M \gamma_k = 1 \quad (11)$$

where γ_k is a binary variable indicating whether the k -th path is selected, defined as: $\gamma_k = \begin{cases} 1, & \text{if yes;} \\ 0, & \text{otherwise;} \end{cases}$; M is the number of paths.

2.3.4. Routing Point Quantity Constraints

When multiple paths exist from a repair team to an accident point, the number of routing points on all paths is subject to a quantity limit. The specific formula is presented in Equation (12).

$$0 \leq H < H_{\max} \quad (12)$$

where H denotes the total number of routing points from the location of a certain repair team to a certain accident point; $H_{\max}$ represents the maximum allowable number of routing points from the location of a certain repair team to a certain accident point.

3. Adaptive Bitterling Fish Optimization with Evolutionary Game Theory (ABFO-EGT)

The standard Bitterling Fish Optimization (BFO) algorithm exhibits inherent limitations. Inspired by reinforcement learning, swarm evolution mechanisms, and adaptive learning mechanisms [35–39], this paper proposes an Adaptive Bitterling Fish Optimization with Evolutionary Game Theory (ABFO-EGT) algorithm to overcome these limitations. By introducing an evolutionary game theory-based decision mechanism, a symmetric mutation strategy, and an adaptive parameter evolution mechanism, this algorithm constructs an optimization framework with self-organizing characteristic.

The iterative optimization process of the ABFO-EGT relies on a closed-loop environmental feedback mechanism. At each iteration, individuals within the population act as rational agents, executing specific evolutionary behaviors (strategies) and interacting with

the objective function (the environment). The resulting fitness improvements serve as environmental rewards (payoffs). These payoffs directly drive the replicator dynamics equation. Consequently, this process continuously guides the population to adaptively adjust the selection probabilities between global exploration and local exploitation for the subsequent iteration.

The high compatibility between the ABFO-EGT algorithm and the cross-regional emergency repair path planning (CR-ERPP) model stems from leveraging this closed-loop feedback mechanism to navigate the complexities of dynamic road networks. Specifically, the evolutionary game theory (EGT) module utilizes dynamic strategy adaptability to track real-time traffic weight fluctuations across the non-symmetric network. Meanwhile, a symmetric mutation strategy effectively counteracts the search direction bias induced by spatial asymmetric constraints. Finally, while strictly satisfying multiple scheduling constraints, an adaptive parameter evolution mechanism significantly enhances local exploitation precision and convergence robustness within complex, high-dimensional path spaces.

3.1. Evolutionary Game Theory-Based Adaptive Strategy Switching Mechanism

Exploration and exploitation are typically balanced in the BFO algorithm through preset thresholds or linear weights. Furthermore, the optimization process is formulated as a symmetric strategy evolutionary game model in this work. Consequently, the replicator dynamic equation drives the adaptive switching of strategies.

3.1.1. Theoretical Motivation and Integration Points of EGT and BFO

Evolutionary game theory (EGT) is a mathematical framework. It simulates the adaptive evolution of strategies in a bounded rational population under dynamic environments. It emphasizes that individuals obtain payoffs through interactions with the environment and other individuals. Individuals then dynamically adjust their strategy choices based on these payoffs. Compared with traditional adaptive methods, EGT has three unique advantages. These advantages address the limitations of the BFO algorithm.

(1) Dynamic Strategy Adaptability

EGT does not require preset stage transition points. It automatically adjusts the selection probability of each strategy using the replicator dynamics equation. The adjustment is based on the actual performance of each strategy in the current iteration. This enables the algorithm to dynamically switch between exploration and exploitation according to the real-time optimization state.

(2) Symmetric Strategy Competition

In EGT, all strategies compete on an equal footing. Their selection probabilities are determined solely by their relative payoffs. This symmetric competition mechanism ensures that no strategy is permanently eliminated. It maintains the structural symmetry of the search process.

(3) Population-Level Evolution

EGT models strategy evolution at the population level, not the individual level. This is highly consistent with the swarm intelligence essence of the BFO algorithm. It allows the algorithm to maintain population diversity while ensuring convergence.

Based on the above theoretical motivations, the integration of EGT and BFO is established on four fundamental mapping levels. This forms a unified optimization framework.

(1) Mapping Between Game Participants and Algorithm Individuals

Each individual in the BFO population corresponds to a participant in the evolutionary game.

(2) Mapping Between Pure Strategies and Optimization Behaviors

The two core optimization behaviors of the algorithm (global exploration and local exploitation) correspond to the two pure strategies in the evolutionary game. This mapping transforms the problem of balancing exploration and exploitation into a strategy selection problem in the game.

(3) Mapping Between Game Payoffs and Fitness Increments

The fitness improvement obtained by an individual after executing a search strategy is mapped to the economic payoff of the game player under that strategy. The magnitude of the payoff qualitatively and quantitatively reflects how well the strategy fits the current search environment.

(4) Mapping Between Evolutionary Game Processes and Algorithm Iteration Processes

Each iteration of the BFO algorithm is equivalent to one generation of update in the evolutionary game. The replicator dynamics equation replaces the rigid probability control of traditional BFO. It drives the adaptive evolution of strategy selection probabilities with the iteration process.

The mechanism diagram of the evolutionary game theory-based adaptive strategy switching mechanism is shown in Figure 1.

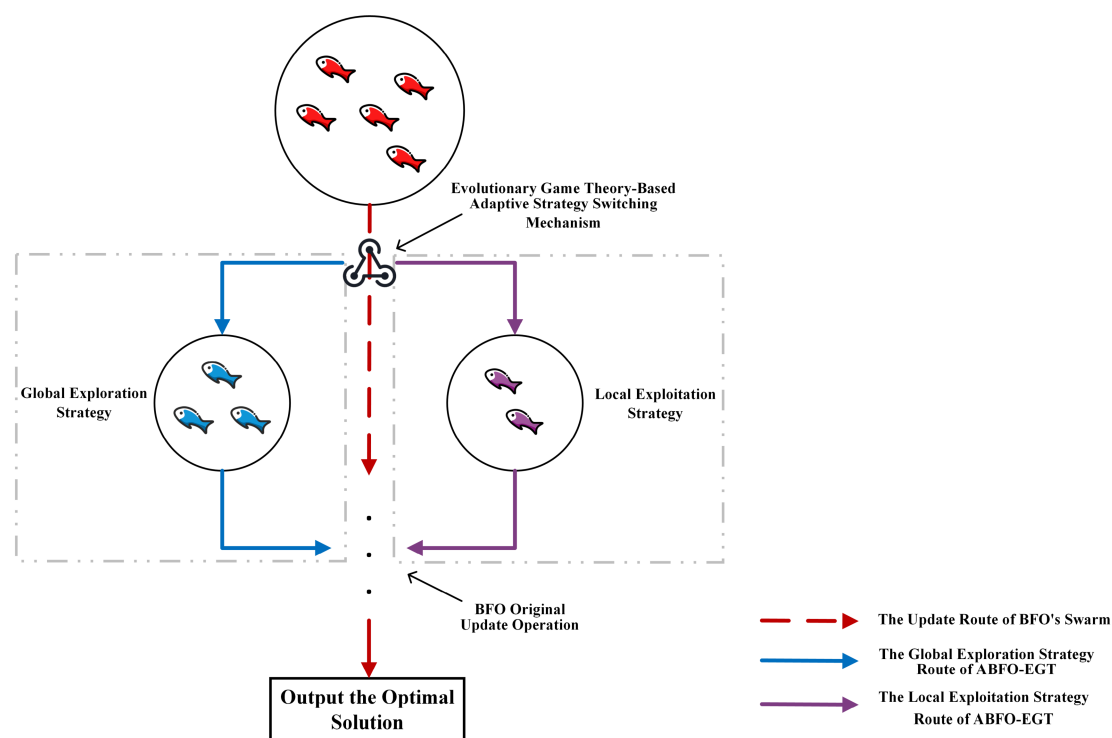


Figure 1. Mechanism diagram of the evolutionary game theory-based adaptive strategy switching mechanism.

3.1.2. Basic Definitions of the Evolutionary Game Model

The two core optimization strategies of the algorithm are defined as the pure strategies of the evolutionary game. In this game, the average fitness improvement achieved by each strategy within one iteration serves as the payoff.

(1) Pure Strategy Set

$$S = \{s_1, s_2\} \quad (13)$$

where S denotes the pure strategy set; s_1 denotes the global exploration strategy; and s_2 denotes the local exploitation strategy.

(2) Strategy Selection Probability

Let $P_m(t)$ denote the probability that an individual selects strategy s_m ($m = 1, 2$) at the t -th iteration. $P_m(t)$ satisfies $P_1(t) + P_2(t) = 1$. In the initial state, $P_1(t) = P_2(t) = 0.5$, which ensures initial symmetry of strategy selection.

(3) Skew-symmetric Payoff Matrix

The payoff matrix is constructed as shown in Equation (14).

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (14)$$

where A denotes the payoff matrix; a_{ij} denotes the payoff for a player choosing strategy s_i when the opponent chooses strategy s_j . In this model, $a_{11} = a_{22} = 0$ and $a_{12} = -a_{21} = 1$. The value of 1 is chosen for normalization purposes. This construction preserves the zero-sum property and yields a skew-symmetric payoff structure, which is appropriate for modeling the competition between exploration and exploitation.

(4) Strategy Payoff

The average fitness improvement of a strategy per iteration is defined as the payoff, which directly reflects the optimization efficiency of the strategy. The calculation formula for the payoff $R_m(t)$ of strategy s_m at the t -th iteration is presented in Equation (15).

$$R_m(t) = \begin{cases} \frac{\Delta f_{m,total}(t)}{N_m(t)}, & N_m(t) > 0 \\ 0, & N_m(t) = 0 \end{cases} \quad (15)$$

where $R_m(t)$ denotes the average payoff of strategy s_m at the t -th iteration; m denotes the strategy index, where $m = 1$ corresponds to the global exploration strategy and $m = 2$ corresponds to the local exploitation strategy; $N_m(t)$ is the total number of strategy s_m used at the t -th iteration; t is the current iteration number, $1 \leq t \leq T$, where T is the maximum number of iterations; and $\Delta f_{m,total}(t)$ is the total fitness improvement brought by strategy s_m at the t -th iteration, i.e., the sum of fitness differences between the original solutions and new solutions for all individuals successfully improved by this strategy (for minimization problems, the individual fitness improvement is calculated only when $f(x^{new}) < f(x^{old})$).

3.1.3. Replicator Dynamics Equation-Based Strategy Probability Update

The replicator dynamics equation is the core of evolutionary game theory. Its core logic dictates that strategies with payoffs higher than the population average will experience an increased selection probability. Conversely, strategies with payoffs lower than the average will suffer a decreased selection probability. The detailed update procedure is presented as follows.

- (1) Calculate the population average payoff at the t -th iteration, which serves as the criterion for evaluating the merits of strategies. The corresponding formula is given in Equation (16).

$$\bar{R}(t) = \frac{1}{2} \sum_{m=1}^2 R_m(t) \quad (16)$$

where $\bar{R}(t)$ denotes the population average payoff at the t -th iteration, i.e., the mean of the average payoffs of the two strategies; m denotes the strategy index; $R_m(t)$ denotes the average payoff of strategy s_m at the t -th iteration.

- (2) Update the strategy selection probability for the next generation by the replicator dynamics equation.

$$P_m(t+1) = P_m(t) \cdot \left(1 + \alpha \cdot \frac{R_m(t) - \bar{R}(t)}{\bar{R}(t)}\right) \quad (17)$$

In the formula, $P_m(t+1)$ denotes the selection probability of strategy s_m at the $(t+1)$ -th iteration; α denotes the learning rate, set to 0.1, which controls the step size of probability update; $P_m(t)$ is the selection probability of strategy s_m at the t -th iteration; $R_m(t)$ denotes the average payoff of strategy s_m at the t -th iteration; and $\bar{R}(t)$ denotes the population average payoff at the t -th iteration.

- (3) Boundary constraints and normalization are applied to the updated probabilities. This step effectively prevents the complete elimination of any strategy due to an excessively low probability. This method ensures that the algorithm maintains both global exploration and local exploitation capabilities throughout all iterations.

$$P_m(t+1) = \max(P_m(t+1), 0.1) \quad (18)$$

$$P_m(t+1) = \frac{P_m(t+1)}{\sum_{m=1}^2 P_m(t+1)} \quad (19)$$

In the formula, $P_m(t+1)$ denotes the selection probability of strategy s_m at the $(t+1)$ -th iteration; $\max(\cdot)$ is the mathematical maximum operator; and m is the strategy index.

3.2. Symmetric Mutation Strategy

Symmetry is a fundamental principle in optimization theory. Symmetric search operators can ensure uniform coverage of the solution space while preventing directional bias. In particular, spatially symmetric mutation guarantees that the probability of generating a solution in any direction is equal to that of generating a solution in the opposite direction. This property is crucial for avoiding premature convergence to local optima caused by one-sided search behaviors. In contrast, conventional mutation operators employed in swarm intelligence algorithms often exhibit asymmetric search biases, resulting in incomplete exploration of the solution space.

The traditional BFO algorithm adopts fixed-structure position update operators. Consequently, its single mutation mode cannot simultaneously satisfy the dual requirements of global exploration and local exploitation. Accordingly, a symmetric mutation strategy is designed for the game model described in Section 3.1. This strategy corresponds to the global exploration and local exploitation strategies, respectively. This provides an executable optimization mechanism for game-theoretic decision-making.

3.2.1. Global Exploration Strategy (s_1)

This strategy corresponds to the pure strategy of global exploration within the game model. The global exploration strategy achieves the core objective of expanding the search

space and avoiding local optima. It is especially applicable to optimization tasks in the early iteration stages. The mutation formula is expressed as Equation (20).

$$v_i(t) = x_{r1}(t) + F_i(t) \cdot (x_{r2}(t) - x_{r3}(t)) + F_i(t) \cdot (x_{r4}(t) - x_{r5}(t)) \quad (20)$$

where $r1, r2, r3, r4, r5$ denote five mutually distinct individual indices randomly selected from the population, none of which equal the current individual index i ; $v_i(t)$ is the mutated candidate solution vector of the i -th individual at the t -th iteration; $F_i(t)$ is the mutation factor of the i -th individual at the t -th iteration, with a value range of $(0, 1.2]$; and $x_{rk}(t)$ is the position vector of the rk -th individual at the t -th iteration ($k = 1, 2, 3, 4, 5$).

3.2.2. Local Exploitation Strategy (s_2)

This strategy corresponds to the pure strategy of local exploitation within the game model. The core objective of the local exploitation strategy is to guide the search using elite solutions. This guidance accelerates the algorithm's convergence toward the global optimum. Furthermore, it meets the requirements for high-precision optimization in the late iteration stages. The mutation formula is given in Equation (21).

$$v_i(t) = x_i(t) + F_i(t) \cdot (x_{wpbest}(t) - x_i(t)) + F_i(t) \cdot (x_{r1}(t) - x_{r2}(t)) \quad (21)$$

where $v_i(t)$ is the mutated candidate solution vector for the i -th individual at the t -th iteration; $x_{wpbest}(t)$ is the weighted elite solution at the t -th iteration (refer to Section 3.3.1 for the specific calculation), which provides the convergence direction for mutation; $x_i(t)$ is the position vector of the i -th individual at the t -th iteration; $r1$ is an individual index randomly selected from the current population, such that $r1 \neq i$; $r2$ denotes an individual index randomly selected from the union population (comprising the current population and the external archive), such that $r2 \neq i$ and $r2 \neq r1$; $F_i(t)$ is the mutation factor of the i -th individual at the t -th iteration, with a value range of $(0, 1.2]$; and $x_{rk}(t)$ is the position vector of the rk -th individual at the t -th iteration ($k = 1, 2$).

3.2.3. Spatial Symmetric Mutation

Both of the aforementioned strategies are spatial symmetric mutations. The mutation factor F is controlled by a probability density function. Its probability distribution is strictly symmetric about the origin. In other words, its probability density function is an even function, as expressed in Equation (22).

$$f(F) = f(-F), \quad \forall F \in (-\infty, +\infty) \quad (22)$$

Formally, individual i generates a new mutated state v_i around multiple individuals (e.g., x_1, x_2, x_3). This process is shown in Equation (23).

$$v_i = \begin{cases} x_1 + F \cdot (x_2 - x_3), & \text{if } m = 1 \\ x_i + F \cdot (x_1 - x_2), & \text{if } m = 2 \end{cases} \quad (23)$$

where m denotes the strategy index.

Because $f(F)$ exhibits the property of an even function, the probability of generating a mutation step in any positive spatial direction is strictly equal to the probability of generating an equivalent step in the opposite negative direction. Assuming the step size is δ ($\delta > 0$), this relationship is presented in Equation (24).

$$\begin{cases} P(v_i \in [x_1, x_1 + \delta]) = P(v_i \in [x_1 - \delta, x_1]), & \text{if } m = 1 \\ P(v_i \in [x_i, x_i + \delta]) = P(v_i \in [x_i - \delta, x_i]), & \text{if } m = 2 \end{cases} \quad (24)$$

This structural symmetry helps reduce directional bias and improves the balance of spatial exploration.

3.2.4. External Archive

The external archive with a maximum capacity of $2 \times N_p$ is established to store superior individuals replaced during the iterative process. The original solution is stored in the external archive when a new solution outperforms it. Once the archive reaches its full capacity, older individuals within the archive are randomly replaced.

This design facilitates the reuse of historically superior solutions. Such reuse supplements population diversity during local exploitation. Ultimately, it prevents population homogenization in the late iteration stages.

3.3. Adaptive Parameter Evolution Mechanism

Adaptive parameter evolution mechanism is a key component of metaheuristic algorithms, as fixed parameter settings often lead to unsatisfactory performance across different problem domains. Symmetric parameter adjustment ensures that the search intensity evolves smoothly and symmetrically throughout the optimization process, thereby avoiding abrupt changes that may disrupt the balance between exploration and exploitation.

To address the high parameter sensitivity of the BFO, ABFO-EGT implements a self-adaptive evolution mechanism for core parameters throughout the iterative process. Accordingly, this study designs an adaptive parameter optimization mechanism to achieve dynamic adjustment of three core parameters: elite ratio, mutation factor, and spawning step size.

3.3.1. Dynamic Adaptive Elite Ratio ($p(t)$)

The elite ratio $p(t)$ directly determines the size of the elite pool in the local exploitation strategy. In this paper, a linearly decreasing $p(t)$ is designed to achieve adaptive adjustment between the early and late iteration stages, as expressed in Equation (25).

$$p(t) = p_{\max} - (p_{\max} - p_{\min}) \cdot \frac{t}{T} \quad (25)$$

where $p_{\max}$ is the initial maximum elite ratio, with 20% of the individuals selected as elites in the early stage of the iteration ($p_{\max} = 0.2$); $p_{\min}$ is the final minimum elite ratio, where at least the two best individuals are selected as core elites in the later stages ($p_{\min} = 2/N_p$); N_p is the number of individuals in the population; t denotes the current iteration number; and T denotes the maximum number of iterations.

The number of elite individuals is calculated based on $p(t)$, and a lower bound is set to ensure the effectiveness of the elite pool. The calculation formula is shown in Equation (26).

$$p_{\text{num}}(t) = \max(2, \text{round}(N_p \cdot p(t))) \quad (26)$$

where $p_{\text{num}}(t)$ denotes the number of elite individuals at the t -th iteration; $p(t)$ denotes the elite ratio; $\max(\cdot)$ denotes the mathematical maximum operator; $\text{round}(\cdot)$ is the rounding function; 2 represents the minimum number of elite individuals; N_p denotes the number of individuals in the population; and t denotes the current iteration number.

On this basis, the weighted elite solution $x_{wpbest}(t)$ is calculated, which assigns higher weights to more excellent elite individuals to strengthen the guiding effect of the optimal solution.

$$x_{wpbest}(t) = \sum_{e=1}^{p_{\text{num}}(t)} w_e \cdot x_e^{\text{best}}(t) \quad (27)$$

$$w_e = \frac{p_{num}(t) - e + 1}{\sum_{k=1}^{p_{num}(t)} k} \quad (28)$$

where $x_{w_{pbest}}(t)$ denotes the weighted elite solution at the t -th iteration; t denotes the current iteration number; $p_{num}(t)$ denotes the number of elite individuals at the t -th iteration; e is the index of elite individuals; w_e is the weight of the e -th elite individual, satisfying $\sum_{e=1}^{p_{num}(t)} w_e = 1$, and the higher the rank, the greater the weight; and $x_e^{best}(t)$ denotes the position vector of the e -th elite individual after all population individuals at the t -th iteration are sorted in ascending order of fitness.

3.3.2. Adaptive Mechanism of the Mutation Factor (F) Based on a Historical Memory Pool

To address the poor adaptability of a fixed mutation factor, this paper maintains a memory pool M_F of size $H = 10$ to store historically successful mutation factors. This mechanism realizes the adaptive generation of the F .

(1) Mutation factor sampling during each iteration

An index r_{idx} is randomly selected from the memory pool, and the mutation factor F is generated using a Cauchy distribution. The theoretical rationale for adopting the Cauchy distribution lies in its structurally symmetric and heavy-tailed probability density function. Both the Cauchy and standard normal distributions possess structurally symmetric probability density functions, which strictly ensure equal probabilities of positive and negative step-size perturbations. However, compared with the normal distribution, the theoretical advantage of adopting the Cauchy distribution lies in its heavy-tailed property. This unique characteristic enables the algorithm to occasionally generate large mutational jumps, effectively preventing the algorithm from stagnating in symmetric local traps.

$$F = M_F(r_{idx}) + 0.1 \cdot \tan(\pi \cdot (rand - 0.5)) \quad (29)$$

In the formula, F denotes the mutation factor; $M_F(r_{idx})$ is the historically successful mutation factor stored at the r_{idx} -th position in the memory pool; and $rand$ denotes a uniformly distributed random number in the range of $[0, 1]$; 0.1 acts as the shift parameter; 0.5 serves as the scale parameter.

(2) Memory pool updating

After each iteration, the memory pool is updated using the weighted Lehmer mean based on the successful mutation F values and their corresponding fitness improvements. The weights are determined by the magnitude of the fitness improvement from the successful mutations.

$$w_j = \frac{\Delta f_j}{\sum_{l=1}^k \Delta f_l} \quad (30)$$

$$M_F(k_{ptr}) = \frac{\sum_{j=1}^k w_j \cdot F_j^2}{\sum_{j=1}^k w_j \cdot F_j} \quad (31)$$

where k denotes the number of individuals with successful mutations in the current iteration; F_j is the mutation factor used in the j -th successful mutation; Δf_j denotes the fitness improvement brought by the j -th successful mutation; w_j is the weight of the j -th successful mutation; k_{ptr} is the update pointer of the memory pool, which is updated by $k_{ptr} = \text{mod}(k_{ptr}, H) + 1$ after each update to realize the rolling update of the memory pool; H represents the predefined total capacity of the memory pool; and $M_F(k_{ptr})$ is the updated value at the pointer position of the memory pool.

3.3.3. Power-Law Decaying Spawning Step Size ($SD(t)$)

Aiming at the spawning behavior of BFO, this paper designs a power-law decaying step size. This design successfully expands the search space during the early iterations. Subsequently, it improves the optimization accuracy in the late iterations.

$$SD(t) = SD_{initial} \cdot \left(1 - \frac{t}{T}\right)^\gamma \quad (32)$$

where $SD_{initial}$ denotes power-law decaying spawning step size; $SD_{initial}$ denotes the initial spawning step size, set to 1; γ is the power-law decay coefficient, set to 5; T denotes the maximum number of iterations; and t denotes the current iteration number.

The pseudocode of ABFO-EGT is Algorithm 1. The flowchart of ABFO-EGT is shown in Figure 2.

3.4. Mapping Mechanism from Continuous to Discrete Space

ABFO-EGT operates in a continuous search space. However, emergency repair path planning is inherently a discrete combinatorial optimization problem. A mapping mechanism is therefore required to bridge this gap. This study adopts the Smallest Position Value (SPV) rule combined with a Path Reconstructor. It converts continuous position vectors into discrete path solutions.

(1) SPV Rule

For each continuous position vector $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ generated by the ABFO-EGT, each dimension is sorted according to its numerical value, and a discrete priority sequence $P_i = [p_1, p_2, \dots, p_D]$ is thus obtained.

(2) Path Reconstructor

The priority sequence P_i is used to determine the traversal order of road segments or nodes. For a given accident point, the priority sequence is decoded into a feasible path satisfying the constraint conditions in Section 2.2 through the path reconstruction function. If the generated sequence leads to inaccessibility or violates the maximum routing point constraint, the greedy repair strategy is triggered, and invalid segments are replaced with the nearest feasible connections. Thus, it is ensured that each continuous position vector can be mapped into a rigorous, feasible and discrete emergency repair path.

3.5. Feasibility Rule

The feasibility rule is a core evaluation mechanism. It combines the objective function value and the degree of constraint violation. It quantitatively compares the quality of individual solutions in stochastic optimization algorithms. When solving constrained optimization problems, this rule only requires standardizing the constraints. It equivalently transforms complex constrained problems into unconstrained ones. This significantly reduces the complexity of algorithm design. Generally, the mathematical model of a single-objective constrained optimization problem is expressed as Equation (33).

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_j(x) \leq 0, \quad j = 1, 2, \dots, p \\ & g_k(x) = 0, \quad k = 1, 2, \dots, q \end{aligned} \quad (33)$$

where $x = [x_1, x_2, \dots, x_n]$ is the n -dimensional optimization variable. $h_j(x)$ is the j -th inequality constraint. $g_k(x)$ is the k -th equality constraint.

Algorithm 1: Adaptive Bitterling Fish Optimization with Evolutionary Game Theory (ABFO-EGT)

Input: Population size N , Dimension D , Maximum iterations T , Search space boundaries $[lb, ub]$, Objective function f_{obj}

Output: The global optimal solution and its fitness value

1: Initialize algorithm parameters: N, D, T

2: Initialize the population X randomly within $[lb, ub]$

3: Set iteration counter $t = 1$

4: Evaluate the fitness value for each individual in X

5: Find and record the optimal individual X_{best}

6: Initialize the External Archive

7: **while** $t < T$ **do**

8: Calculate the adaptive elite ratio $p(t)$

9: Calculate spawning step size $SD(t)$

10: Calculate number of elites $p_{num}(t)$

11: Calculate weighted elite solution $x_{wpbest}(t)$

12: **if** $t > 1$ **then**

13: Update the strategy selection probabilities $P_m(t)$

14: **else**

15: The strategy selection probability is set to the initial value $P_1(t) = P_2(t) = 0.5$.

16: **end if**

17: Sample from the historical memory pool M_F

18: Generate the mutation factor F via the Cauchy distribution

19: Execute foraging behavior

20: Update the external archive and global optimal solution

21: **for** $i = 1$ to N **do**

22: Generate a random number $rand$

23: **if** $rand < P_1(t)$ **then**

24: The i -th individual executes mutation strategy 1

25: **else**

26: The i -th individual executes mutation strategy 2

27: **end if**

28: Update the external archive and strategy payoffs

29: **end for**

30: Update the historical memory pool M_F

31: Carry out spawning behavior

32: Perform weight learning, differential update, and mating behavior

33: Record the global optimal fitness

34: $t = t + 1$

35: **end while**

36: **Output** the global optimal solution and its fitness value

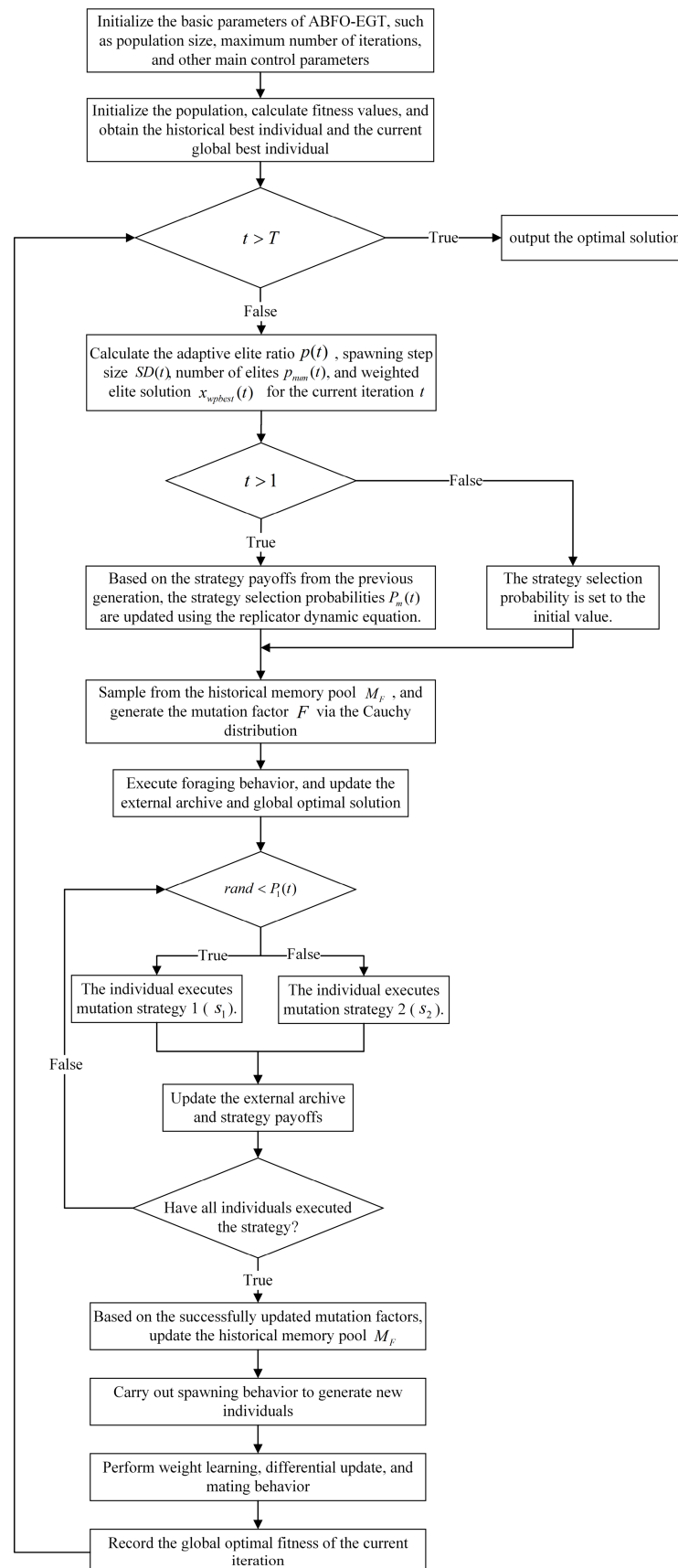


Figure 2. Flowchart of ABFO-EGT.

The above model contains p inequality constraints and q equality constraints. To unify the constraint forms, a tolerance parameter ε is introduced. Equality constraints are transformed into inequality constraints, as shown in Equation (34).

$$|g_k(x)| \leq \varepsilon \quad (34)$$

Through the transformation in Equation (28), the original problem becomes a pure inequality-constrained nonlinear optimization problem. It now contains $p + q$ inequality constraints. The essence of this problem is to find a decision vector x that minimizes the objective function $f(x)$. This vector x must lie within the feasible region satisfying all constraints. To quantitatively measure how much individual solutions in the population deviate from the constraints, a single constraint violation function ($\psi_i(x)$) is defined as Equation (35).

$$\psi_i(x) = \begin{cases} \max\{0, h_i(x)\}, & i = 1, 2, \dots, p \\ \max\{0, |g_{i-p}(x)| - \varepsilon\}, & i = p + 1, p + 2, \dots, p + q \end{cases} \quad (35)$$

Based on the single constraint violation function, the total constraint violation degree of an individual solution is further defined. This comprehensively represents its total violation of all constraints ($\Phi(x)$), as shown in Equation (36).

$$\Phi(x) = \sum_{i=1}^{p+q} \psi_i(x) \quad (36)$$

Combining the individual's constraint violation degree and objective function value, an individual comparison rule can be established. This rule is used to select high-quality individuals for the next iteration. Let a and b be any two individuals in the population. The specific selection rules are as follows.

- (1) If both a and b are infeasible solutions, the individual with the smaller constraint violation degree is better. That is, if $\Phi(a) < \Phi(b)$, a is superior to b .
- (2) If both a and b are feasible solutions, the evaluation is based on the objective function value. This study aims to minimize the total emergency repair cost. The individual with the smaller objective function value is better. That is, if $f(a) < f(b)$, a is superior to b .
- (3) If a is a feasible solution and b is an infeasible solution, the feasible solution is always better. That is, a is superior to b .

3.6. Computational Complexity Analysis

The computational complexity of the proposed ABFO-EGT is a crucial metric for evaluating its applicability in dynamic emergency repair planning. This complexity primarily depends on population initialization, adaptive parameter computation, EGT strategy switching, symmetric mutation, and objective function evaluation. To quantify this, let N denote the population size, D denote the dimension of the decision variables, T represent the maximum number of iterations, and C_{obj} indicate the computational cost of objective function evaluation.

3.6.1. Time Complexity

During the initialization phase, generating the population and evaluating initial fitness requires a runtime of $O(N \times D + N \times C_{obj})$. Within the iterative loop, sorting the population to determine weighted elite solutions incurs a complexity of $O(N \log N)$. The EGT-based probability update relies solely on calculating average strategy payoffs, which executes

in linear time, $O(N)$. Executing symmetric mutation, foraging, and spawning behaviors requires traversing all dimensions; this demands $O(N \times D)$ operations per generation. Finally, evaluating the fitness of the updated individuals takes $O(N \times C_{obj})$.

Accordingly, the overall time complexity of ABFO-EGT is rigorously formulated as $O(N \times (D + C_{obj}) + T \times (N \log N + N \times D + N \times C_{obj}))$. For complex engineering problems, $N \log N$ is generally smaller than or comparable to $N \times D$. After omitting lower-order terms, the asymptotic time complexity simplifies to $O(T \times N \times (D + C_{obj}))$. This theoretically demonstrates that, compared to standard swarm intelligence algorithms, the integrated symmetric mutation mechanism and evolutionary game theory module do not increase the polynomial order of the time complexity.

3.6.2. Space Complexity

The memory consumption of ABFO-EGT primarily stems from three components. First, storing the main population matrix requires $O(N \times D)$. Second, maintaining an external archive to retain historically superior individuals incurs $O(N_A \times D)$, where N_A denotes the predefined archive capacity. Third, allocating the historical memory pool for mutation factors takes $O(M_F)$, where M_F represents the memory pool size. Consequently, the upper bound of the total space complexity is $O((N_A + N) \times D + M_F)$. Given that both N_A and M_F are predefined finite constants, the overall space complexity scales linearly with the problem dimension. This linear scaling demonstrates high memory efficiency.

4. Solution Performance Analysis of ABFO-EGT

4.1. Ablation Study and Comparative Experiments with Multiple Algorithms

This paper designs two sets of verification schemes to systematically verify the optimization performance of the proposed ABFO-EGT and the effectiveness of the overall improvement strategies. These schemes comprise an ablation study and comparative experiments involving multiple algorithms. All experiments are carried out on the CEC 2017 benchmark test suite [40], which is widely used in the IEEE evolutionary computation community. This test suite contains 30 test functions with diverse characteristics. They are classified into four groups: unimodal functions (F1~F3), multimodal functions (F4~F10), hybrid functions (F11~F20), and composition functions (F21~F30). These groups comprehensively cover the full spectrum of optimization problems, ranging from simple unimodal to complex multimodal, hybrid, and composition scenarios.

4.1.1. Design of the Ablation Study

The traditional BFO algorithm suffers from inherent limitations, namely unbalanced search behavior and premature convergence to local optima. The key contribution of this study lies in addressing these issues. Specifically, it introduces adaptive mechanisms and evolutionary game theory to construct a complete algorithmic framework for ABFO-EGT. An ablation study is conducted to directly verify the comprehensive performance improvements of these strategies. In this study, the original BFO algorithm serves as the control group for comparison with the proposed ABFO-EGT algorithm.

4.1.2. Design of Comparative Experiments with Multiple Algorithms

The 9 swarm intelligence optimization algorithms are selected as the horizontal comparison benchmarks. Specifically, they are Particle Swarm Optimization (PSO) [41], Crested Porcupine Optimizer (CPO) [42], Hippopotamus Optimization (HO) [43], Pelican Optimization Algorithm (POA) [44], Goose Optimization Algorithm (GOOSE) [45], Newton–Raphson-Based Optimizer (NRBO) [46], Parrot Optimizer (PO) [47], Dingo Optimization Algorithm (DOA) [48], and Golden Jackal Optimization (GJO) [49].

4.1.3. Unified Experimental Parameter Settings

Completely identical experimental configurations are adopted for all algorithms: the maximum number of iterations is $T = 1000$, the population size is set to $N_p = 30$, and the dimension of the decision variables is $D = 30$. The influence of algorithmic randomness on the results is eliminated by independently running each algorithm 30 times on each test function. The mean (Mean) and standard deviation (Std) of the optimal values are recorded as the core evaluation metrics. Furthermore, all competing algorithms adopt the default parameters and structures as specified in their original literature. The main control parameter settings for all benchmark algorithms are shown in Table 1.

Table 1. Main control parameter settings for all benchmark algorithms.

Intelligent Algorithms	Main Control Parameter	Value
PSO	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	the learning factor	2
	the learning factor	2
	the range of inertia weights	[0.6, 0.9]
	the maximum speed limit	6
CPO	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	the minimum population size	15
	the number of cycles	2
	the convergence rate	0.2
	the strategy trade-off factor	0.8
HO	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	the mode switching threshold	0.6
	Levy flight parameters	1.5
	the Levy flight scaling factor	0.05
POA	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	Local disturbance coefficient	0.2
GOOSE	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	the coefficient threshold	0.17
	the threshold for switching between development and exploration modes	0.5
	the internal sub strategy threshold of the development mode	0.2
NRBO	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	the determining factor of the trap avoidance operator	0.6
PO	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	Levy flight parameters	1.5
	the probability threshold of communication behavior	0.5

Table 1. Cont.

Intelligent Algorithms	Main Control Parameter	Value
DOA	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	Hunting or scavenger rate	0.5
	Group attack or persecution rate	0.7
	Survival Rate Threshold	0.3
	Minimum attacking population	2
GJO	the lower bound of the search space boundary	−100
	the upper bound of the search space boundary	100
	Levy flight parameters	1.5
	the energy parameter benchmark value	1.5
	the Levy flight scaling factor	0.05

4.1.4. Experimental Results

The detailed test results are presented in Table 2.

As demonstrated in Table 2, the ABFO-EGT exhibits a comprehensive performance that is significantly superior to that of the BFO and the other nine algorithms in solving the CEC 2017 benchmark test suite.

(1) Unimodal Functions (F1–F3)

ABFO-EGT achieves the optimal mean and standard deviation across all three unimodal functions. Unimodal functions contain a single global optimum. Therefore, algorithmic performance primarily depends on local exploitation precision and convergence speed. ABFO-EGT utilizes an adaptive parameter evolution mechanism to gradually narrow the search range in later iterations. The local exploitation strategy guided by weighted elite solutions can accurately point to the optimal solution region, and the power-law decaying spawning step size further improves the local search accuracy. Therefore, it shows overwhelming convergence accuracy advantages on unimodal functions.

(2) Multimodal Functions (F4–F10)

ABFO-EGT yields the optimal mean results on all seven multimodal functions. Multimodal functions feature numerous local optima. Algorithmic success here hinges on the dynamic exploration–exploitation balance and the ability to escape local stagnation. On one hand, the EGT-based adaptive strategy switching mechanism dynamically adjusts the exploration–exploitation ratio based on the current search state. This prevents premature convergence. On the other hand, the symmetrical mutation strategy guarantees an unbiased search. Simultaneously, the heavy-tailed Cauchy distribution generates large-step mutations, assisting the population in jumping out of local optima. Together, these mechanisms maintain population diversity while ensuring high convergence efficiency on multimodal problems.

(3) Hybrid (F11–F20) and Composition Functions (F21–F30)

ABFO-EGT achieves the optimal mean on nine out of 10 hybrid functions and all 10 composition functions. These benchmarks exhibit complex landscape characteristics, including multiple subspaces, multimodality, and asymmetry. They place stringent demands on global search capability and adaptability. The symmetric mutation strategy effectively counteracts search bias caused by asymmetrical constraints in complex spaces. Furthermore, the EGT-based adaptive strategy switching mechanism regulates the search mode, while an external archive supplements population diversity. This synergy enables the algorithm to stably converge to the global optimal region even in high-dimensional, complex solution spaces.

Table 2. Optimization Results between ABFO-EGT and the Other 10 Swarm Intelligent Algorithms on 30-Dimensional Benchmark Functions.

f		BFO	PSO	CPO	HO	POA	GOOSE	NRBO	PO	DOA	GJO	ABFO-EGT
f_1	Mean	4.0402×10^3	5.1694×10^7	2.0056×10^{10}	1.5781×10^8	1.5853×10^{10}	8.7578×10^3	1.7542×10^{10}	4.3099×10^9	4.9636×10^{10}	1.1592×10^{10}	1.4407×10^3
	Std	4.5951×10^3	2.2957×10^8	3.7426×10^9	2.1856×10^8	6.7585×10^9	6.0213×10^3	3.7353×10^9	2.0629×10^9	8.6625×10^9	3.1580×10^9	2.0767×10^3
	Mean Rank	2	4	10	5	8	3	9	6	11	7	1
	Std Rank	2	5	9	4	10	3	8	6	11	7	1
f_2	Mean	1.4112×10^5	3.7754×10^{21}	2.9384×10^{38}	3.4923×10^{29}	7.6614×10^{31}	1.4226×10^{18}	4.1042×10^{36}	4.0317×10^{31}	1.1350×10^{45}	2.2284×10^{35}	200.0000
	Std	4.4470×10^5	9.7352×10^{21}	5.3218×10^{38}	8.5102×10^{29}	1.6561×10^{32}	5.2284×10^{18}	1.3785×10^{37}	1.7889×10^{32}	4.5498×10^{45}	9.8345×10^{35}	1.6703×10^{-5}
	Mean Rank	2	4	10	5	7	3	9	6	11	8	1
	Std Rank	2	4	10	5	6	3	9	7	11	8	1
f_3	Mean	2.3906×10^3	1.3701×10^4	1.4072×10^5	5.2870×10^4	3.3171×10^4	6.1460×10^4	5.2002×10^4	3.8783×10^4	7.0454×10^4	5.9138×10^4	300.0000
	Std	1.4589×10^3	6.0204×10^3	3.2286×10^4	7.0757×10^3	9.7230×10^3	2.2377×10^4	9.5496×10^3	8.9406×10^3	1.1988×10^4	1.1011×10^4	2.5261×10^{-9}
	Mean Rank	2	3	11	7	4	9	6	5	10	8	1
	Std Rank	2	3	11	4	7	10	6	5	9	8	1
f_4	Mean	484.0606	504.8582	4.7694×10^3	639.7744	1.9625×10^3	505.6656	1.8572×10^3	709.2180	1.1503×10^4	1.2782×10^3	422.3861
	Std	31.3955	23.4622	1.0359×10^3	100.1930	1.2882×10^3	17.9835	791.1137	208.6547	3.6262×10^3	670.3518	25.0149
	Mean Rank	2	3	10	5	9	4	8	6	11	7	1
	Std Rank	4	2	9	5	10	1	8	6	11	7	3
f_5	Mean	659.5706	692.5863	848.4875	727.0491	766.8151	859.8683	839.8560	769.8543	854.3649	737.9295	602.9832
	Std	34.5144	28.6782	23.5369	46.3730	24.2780	63.9560	32.9809	55.4532	49.6790	51.9615	28.0230
	Mean Rank	2	3	9	4	6	11	8	7	10	5	1
	Std Rank	6	4	1	7	2	11	5	10	8	9	3
f_6	Mean	632.5852	648.6380	667.1167	656.7096	657.0820	673.7337	671.1188	663.8735	679.5919	638.1620	604.6932
	Std	12.8479	6.1240	7.8696	6.7660	9.1110	10.1842	10.9824	8.4310	11.6736	8.2217	3.3445
	Mean Rank	2	4	8	5	6	10	9	7	11	3	1
	Std Rank	11	2	4	3	7	8	9	6	10	5	1
f_7	Mean	1.1101×10^3	874.3820	1.2421×10^3	1.1494×10^3	1.2543×10^3	2.0106×10^3	1.2565×10^3	1.2114×10^3	1.3420×10^3	1.0435×10^3	837.4320
	Std	109.8873	40.6420	44.4987	70.0521	68.4244	849.3855	90.3281	76.3329	61.7964	57.8341	33.1528
	Mean Rank	4	2	7	5	8	11	9	6	10	3	1
	Std Rank	10	2	3	7	6	11	9	8	5	4	1
f_8	Mean	938.7922	944.9972	1.1156×10^3	968.8468	983.1943	1.0903×10^3	1.0855×10^3	1.0070×10^3	1.1120×10^3	967.9307	907.3227
	Std	39.8314	24.6502	20.9729	33.5239	30.9281	88.5956	24.7137	30.3595	40.5720	27.9125	19.6924
	Mean Rank	2	3	11	5	6	9	8	7	10	4	1
	Std Rank	9	3	2	8	7	11	4	6	10	5	1

Table 2. Cont.

f		BFO	PSO	CPO	HO	POA	GOOSE	NRBO	PO	DOA	GJO	ABFO-EGT
f_9	Mean	3.5165×10^3	5.4770×10^3	9.6131×10^3	5.1101×10^3	5.4008×10^3	9.3998×10^3	7.2844×10^3	6.4954×10^3	9.6521×10^3	5.5742×10^3	1.5188×10^3
	Std	838.2435	1.3444×10^3	2.0960×10^3	787.1813	652.7018	3.3132×10^3	1.1845×10^3	1.3227×10^3	2.1526×10^3	1.0865×10^3	529.0248
	Mean Rank	2	5	10	3	4	9	8	7	11	6	1
	Std Rank	4	8	9	3	2	11	6	7	10	5	1
f_{10}	Mean	5.0920×10^3	4.4043×10^3	9.3528×10^3	5.2143×10^3	5.1970×10^3	5.7648×10^3	7.8870×10^3	6.5357×10^3	8.8248×10^3	6.7201×10^3	4.2782×10^3
	Std	432.0904	673.0189	343.6051	542.4686	576.1689	474.7211	515.8649	665.9877	887.3771	1.6764×10^3	500.0533
	Mean Rank	3	2	11	5	4	6	9	7	10	8	1
	Std Rank	2	9	1	6	7	3	5	8	10	11	4
f_{11}	Mean	1.3008×10^3	1.2246×10^3	8.7736×10^3	1.5375×10^3	2.1198×10^3	1.3519×10^3	2.9571×10^3	1.5345×10^3	6.8097×10^3	3.3221×10^3	1.2654×10^3
	Std	80.2695	50.1265	3.0702×10^3	128.4198	739.0550	72.3225	909.9598	153.2002	4.9634×10^3	1.2436×10^3	46.4071
	Mean Rank	3	1	11	6	7	4	8	5	10	9	2
	Std Rank	4	2	10	5	7	3	8	6	11	9	1
f_{12}	Mean	5.7645×10^5	1.3402×10^6	2.5770×10^9	6.4560×10^7	1.7577×10^9	4.6737×10^6	1.6609×10^9	1.7717×10^8	7.9167×10^9	9.3751×10^8	3.0104×10^4
	Std	3.9810×10^5	6.3032×10^5	6.6617×10^8	6.6740×10^7	1.9621×10^9	3.1192×10^6	7.4317×10^8	1.2145×10^8	3.8043×10^9	8.1785×10^8	2.7415×10^4
	Mean Rank	2	3	10	5	9	4	8	6	11	7	1
	Std Rank	2	3	7	5	10	4	8	6	11	9	1
f_{13}	Mean	2.0916×10^4	2.4292×10^5	7.8132×10^8	1.6458×10^5	3.4715×10^7	1.4191×10^5	2.6355×10^8	1.1269×10^6	1.1505×10^9	7.1997×10^7	1.8780×10^4
	Std	2.3275×10^4	9.8836×10^5	4.5471×10^8	2.2487×10^5	9.2641×10^7	8.4031×10^4	1.7919×10^8	2.8387×10^6	1.2854×10^9	7.6536×10^7	1.7445×10^4
	Mean Rank	2	5	10	4	7	3	9	6	11	8	1
	Std Rank	2	5	10	4	8	3	9	6	11	7	1
f_{14}	Mean	7.2433×10^3	3.0872×10^4	2.1042×10^6	1.9735×10^5	1.6944×10^4	2.5610×10^4	1.5435×10^5	3.1805×10^5	1.5724×10^5	6.2288×10^5	1.6543×10^3
	Std	4.4326×10^3	3.3862×10^4	1.1209×10^6	3.0997×10^5	1.9489×10^4	2.2243×10^4	3.0806×10^5	2.7551×10^5	3.8463×10^5	6.5815×10^5	78.9715
	Mean Rank	2	5	11	8	3	4	6	9	7	10	1
	Std Rank	2	5	11	8	3	4	7	6	9	10	1
f_{15}	Mean	1.3256×10^4	1.2878×10^4	5.8844×10^7	3.1705×10^4	4.0052×10^4	4.7769×10^4	1.5753×10^6	7.4005×10^4	4.6017×10^7	5.8705×10^6	3.7635×10^3
	Std	1.3882×10^4	1.1076×10^4	2.8278×10^7	2.8674×10^4	2.9542×10^4	4.5614×10^4	3.2625×10^6	6.1462×10^4	1.6782×10^8	1.0622×10^7	5.7222×10^3
	Mean Rank	3	2	11	4	5	6	8	7	10	9	1
	Std Rank	3	2	10	4	5	6	8	7	11	9	1
f_{16}	Mean	2.2741×10^3	2.3991×10^3	3.0026×10^3	2.4700×10^3	2.3085×10^3	2.9102×10^3	2.6044×10^3	2.4359×10^3	2.8170×10^3	2.3610×10^3	2.1380×10^3
	Std	197.5573	255.9444	179.6391	253.1504	166.4464	400.6331	246.9690	214.1876	430.8615	221.0232	210.3347
	Mean Rank	2	5	11	7	3	10	8	6	9	4	1
	Std Rank	3	9	2	8	1	10	7	5	11	6	4
f_{17}	Mean	1.8572×10^5	6.9637×10^5	1.5327×10^7	6.3033×10^5	2.4797×10^5	3.5500×10^5	1.5971×10^6	2.0438×10^6	1.1887×10^7	2.3043×10^6	8.8585×10^3
	Std	1.4795×10^5	6.8074×10^5	9.9862×10^6	4.9629×10^5	2.8528×10^5	1.8992×10^5	1.6491×10^6	2.0721×10^6	3.4831×10^7	2.2054×10^6	7.6892×10^3
	Mean Rank	2	6	11	5	3	4	7	8	10	9	1
	Std Rank	2	6	10	5	4	3	7	8	11	9	1

Table 2. Cont.

f		BFO	PSO	CPO	HO	POA	GOOSE	NRBO	PO	DOA	GJO	ABFO-EGT
f_{18}	Mean	2.2660×10^5	4.4018×10^5	2.6492×10^7	8.0217×10^5	2.1500×10^5	3.8309×10^5	2.3824×10^6	2.1799×10^6	2.3121×10^6	4.6745×10^6	1.1230×10^4
	Std	1.7722×10^5	3.1480×10^5	1.3607×10^7	7.5064×10^5	1.9968×10^5	2.8530×10^5	2.0422×10^6	1.7506×10^6	7.3529×10^6	7.9492×10^6	1.0801×10^4
	Mean Rank	3	5	11	6	2	4	9	7	8	10	1
	Std Rank	2	5	11	6	3	4	8	7	9	10	1
f_{19}	Mean	1.8801×10^4	8.4862×10^3	8.5816×10^7	1.7913×10^6	9.9342×10^5	6.5791×10^5	6.3929×10^6	3.1834×10^6	7.4498×10^7	3.4731×10^7	2.8742×10^3
	Std	2.0206×10^4	5.9864×10^3	3.9168×10^7	1.0611×10^6	7.0653×10^5	3.8453×10^5	8.2501×10^6	1.9811×10^6	2.0644×10^8	7.2715×10^7	1.8306×10^3
	Mean Rank	3	2	11	6	5	4	8	7	10	9	1
	Std Rank	3	2	9	6	5	4	8	7	11	10	1
f_{20}	Mean	2.6432×10^3	2.6851×10^3	3.2427×10^3	2.6094×10^3	2.4979×10^3	3.1115×10^3	2.7203×10^3	2.6712×10^3	2.8627×10^3	2.5559×10^3	2.4827×10^3
	Std	190.9154	145.2887	175.4232	103.8650	118.0178	245.4746	209.9605	172.3405	257.1391	194.0339	124.2504
	Mean Rank	5	7	11	4	2	10	8	6	9	3	1
	Std Rank	7	4	6	1	2	10	9	5	11	8	3
f_{21}	Mean	2.4410×10^3	2.4706×10^3	2.6314×10^3	2.5298×10^3	2.5493×10^3	2.6631×10^3	2.6010×10^3	2.5376×10^3	2.6848×10^3	2.4732×10^3	2.4069×10^3
	Std	36.3533	42.6600	21.9942	41.4819	55.6830	66.5792	38.5542	56.4874	70.7146	50.0957	23.2840
	Mean Rank	2	3	9	5	7	10	8	6	11	4	1
	Std Rank	3	6	1	5	8	10	4	9	11	7	2
f_{22}	Mean	5.1608×10^3	4.6855×10^3	6.5919×10^3	5.6443×10^3	4.6125×10^3	7.9428×10^3	6.0973×10^3	3.8647×10^3	8.1270×10^3	5.8987×10^3	3.4643×10^3
	Std	2.2505×10^3	2.2759×10^3	2.1782×10^3	2.2305×10^3	1.3659×10^3	853.5523	2.3814×10^3	1.8778×10^3	1.6056×10^3	3.1039×10^3	1.6449×10^3
	Mean Rank	5	4	9	6	3	10	8	2	11	7	1
	Std Rank	8	9	6	7	2	1	10	5	3	11	4
f_{23}	Mean	2.9543×10^3	3.1011×10^3	3.1963×10^3	2.9906×10^3	3.0315×10^3	3.4728×10^3	3.0449×10^3	2.9823×10^3	3.3391×10^3	2.8967×10^3	2.7904×10^3
	Std	87.0254	137.6229	46.2933	87.7792	96.6220	130.4690	47.9312	80.5470	123.2087	46.9429	44.6342
	Mean Rank	3	8	9	5	6	11	7	4	10	2	1
	Std Rank	6	11	2	7	8	10	4	5	9	3	1
f_{24}	Mean	3.1162×10^3	3.2092×10^3	3.3648×10^3	3.2115×10^3	3.2016×10^3	3.6701×10^3	3.1982×10^3	3.0942×10^3	3.5865×10^3	3.0840×10^3	2.9895×10^3
	Std	77.5141	93.5645	51.9980	150.2453	106.6311	116.5219	58.5146	65.7727	200.7552	56.5214	55.7003
	Mean Rank	4	7	9	8	6	11	5	3	10	2	1
	Std Rank	6	7	1	10	8	9	4	5	11	3	2
f_{25}	Mean	2.9126×10^3	2.9074×10^3	3.8204×10^3	3.0134×10^3	3.2519×10^3	2.9162×10^3	3.4050×10^3	3.0620×10^3	4.9866×10^3	3.3421×10^3	2.8901×10^3
	Std	24.0568	17.3218	181.5571	64.2494	180.4301	19.2884	183.8836	50.2524	735.2303	221.3407	7.4039
	Mean Rank	3	2	10	5	7	4	9	6	11	8	1
	Std Rank	4	2	8	6	7	3	9	5	11	10	1
f_{26}	Mean	6.3576×10^3	5.7673×10^3	8.5012×10^3	7.3524×10^3	7.3127×10^3	1.0654×10^4	7.8260×10^3	7.3959×10^3	9.6972×10^3	5.9782×10^3	5.2672×10^3
	Std	1.3451×10^3	2.1113×10^3	447.7004	1.4922×10^3	1.4014×10^3	2.6486×10^3	1.0917×10^3	1.3032×10^3	1.1404×10^3	580.5816	763.7374
	Mean Rank	4	2	9	6	5	11	8	7	10	3	1
	Std Rank	7	10	1	9	8	11	4	6	5	2	3

Table 2. Cont.

f		BFO	PSO	CPO	HO	POA	GOOSE	NRBO	PO	DOA	GJO	ABFO-EGT
f_{27}	Mean	3.3079×10^3	3.4255×10^3	3.9188×10^3	3.4594×10^3	3.3660×10^3	4.2411×10^3	3.4338×10^3	3.3395×10^3	3.8004×10^3	3.3477×10^3	3.2374×10^3
	Std	43.6860	153.3206	157.8229	88.4217	67.4441	316.5171	94.3379	42.1473	391.4884	63.5189	20.3871
	Mean Rank	2	6	10	8	5	11	7	3	9	4	1
	Std Rank	3	8	9	6	5	10	7	2	11	4	1
f_{28}	Mean	3.3044×10^3	3.4792×10^3	3.8438×10^3	3.4848×10^3	3.3711×10^3	3.9991×10^3	3.4384×10^3	3.3511×10^3	3.7971×10^3	3.3611×10^3	3.2325×10^3
	Std	61.3382	212.7759	118.2558	134.5975	73.2407	371.4221	71.3185	73.4295	328.5728	54.6049	20.7922
	Mean Rank	2	7	10	8	5	11	6	3	9	4	1
	Std Rank	3	9	7	8	5	11	4	6	10	2	1
f_{29}	Mean	4.1709×10^3	4.1523×10^3	5.5395×10^3	5.0308×10^3	4.5016×10^3	4.9847×10^3	5.0847×10^3	4.7109×10^3	5.9243×10^3	4.4446×10^3	3.7940×10^3
	Std	345.5527	274.7747	273.4482	348.6004	348.0051	464.3135	578.5096	300.7228	1.3271×10^3	427.2083	198.3823
	Mean Rank	3	2	10	8	5	7	9	6	11	4	1
	Std Rank	5	3	2	7	6	9	10	4	11	8	1
f_{30}	Mean	1.0335×10^4	5.0650×10^4	1.2215×10^8	1.6822×10^7	1.0794×10^7	1.9441×10^6	6.5598×10^7	3.1879×10^7	1.7204×10^8	3.6496×10^7	7.8100×10^3
	Std	4.6640×10^3	4.1415×10^4	4.0166×10^7	1.8675×10^7	9.2741×10^6	9.9312×10^5	4.1621×10^7	2.6838×10^7	1.3612×10^8	3.1937×10^7	2.8815×10^3
	Mean Rank	2	3	10	6	5	4	9	7	11	8	1
	Std Rank	2	3	9	6	5	4	10	7	11	8	1

Notably, ABFO-EGT yields a suboptimal result only on the hybrid function F11. This represents the sole test instance among all benchmarks where EGT failed to achieve top performance. The reason lies in the structure of F11. It is a separable hybrid function composed of multiple low-dimensional unimodal sub-functions. Its overall landscape is relatively smooth and lacks deceptive local optimal structures. On such structurally simple problems, the overhead of strategy selection and payoff calculation introduced by the EGT mechanism does not offer a distinct advantage. Consequently, PSO shows a slightly faster convergence trend due to its simpler velocity update rules. However, regarding standard deviation, ABFO-EGT still exhibits superior stability on F11. This confirms that the algorithm's robustness remains unaffected by problem landscapes.

Overall, ABFO-EGT achieves optimal results on 29 out of 30 benchmark functions. This demonstrates the strong adaptability of its proposed improvements across diverse search spaces. Specifically, the adaptive parameter mechanism ensures local exploitation precision on unimodal functions. The adaptive strategy switching mechanism maintains the exploration–exploitation balance on multimodal functions. Finally, the symmetrical mutation strategy enhances global search capabilities on complex functions. These three mechanisms produce a strong synergistic effect, collectively elevating the overall optimization performance of the algorithm.

To further investigate the convergence speed of the ABFO-EGT, typical test functions were selected from the test suite to plot the iterative descent curves, as shown in Figure 3.

From the perspective of overall convergence trends, the ABFO-EGT algorithm demonstrates significantly superior convergence speeds and final accuracies compared to the BFO and the other nine algorithms across the six representative test functions. As illustrated in Figure 3a,c,d, ABFO-EGT converges to fitness values far below those of the other algorithms. It achieves this within a remarkably small number of iterations, exhibiting overwhelming convergence efficiency. In Figure 3b,e,f, other swarm intelligence algorithms exhibit obvious stagnation during the iterative process, reflecting their susceptibility to being trapped in local optima. By contrast, after a short period of stagnation in the later iteration stage, the convergence curve of ABFO-EGT can decline again. This reveals that ABFO-EGT has an excellent capability to escape local optima and can effectively search for global optimal solutions within complex solution spaces. The iterative descent curve of ABFO-EGT consistently remains below those of all other algorithms. This superior performance holds true for both simple unimodal functions and extremely complex composition functions. Consequently, its final convergence results are significantly better than those of the other 10 algorithms. This further confirms that the ABFO-EGT algorithm has excellent adaptability when dealing with different types of optimization problems.

4.2. Comparison of Computational Costs

Furthermore, a comparative analysis of computational costs was conducted to comprehensively evaluate the practicality of ABFO-EGT. This analysis compared the algorithms across two dimensions: running time and memory usage. Each algorithm was subjected to 20 independent runs to obtain the average values for both running time and memory consumption. The comparison results are shown in Figure 4.

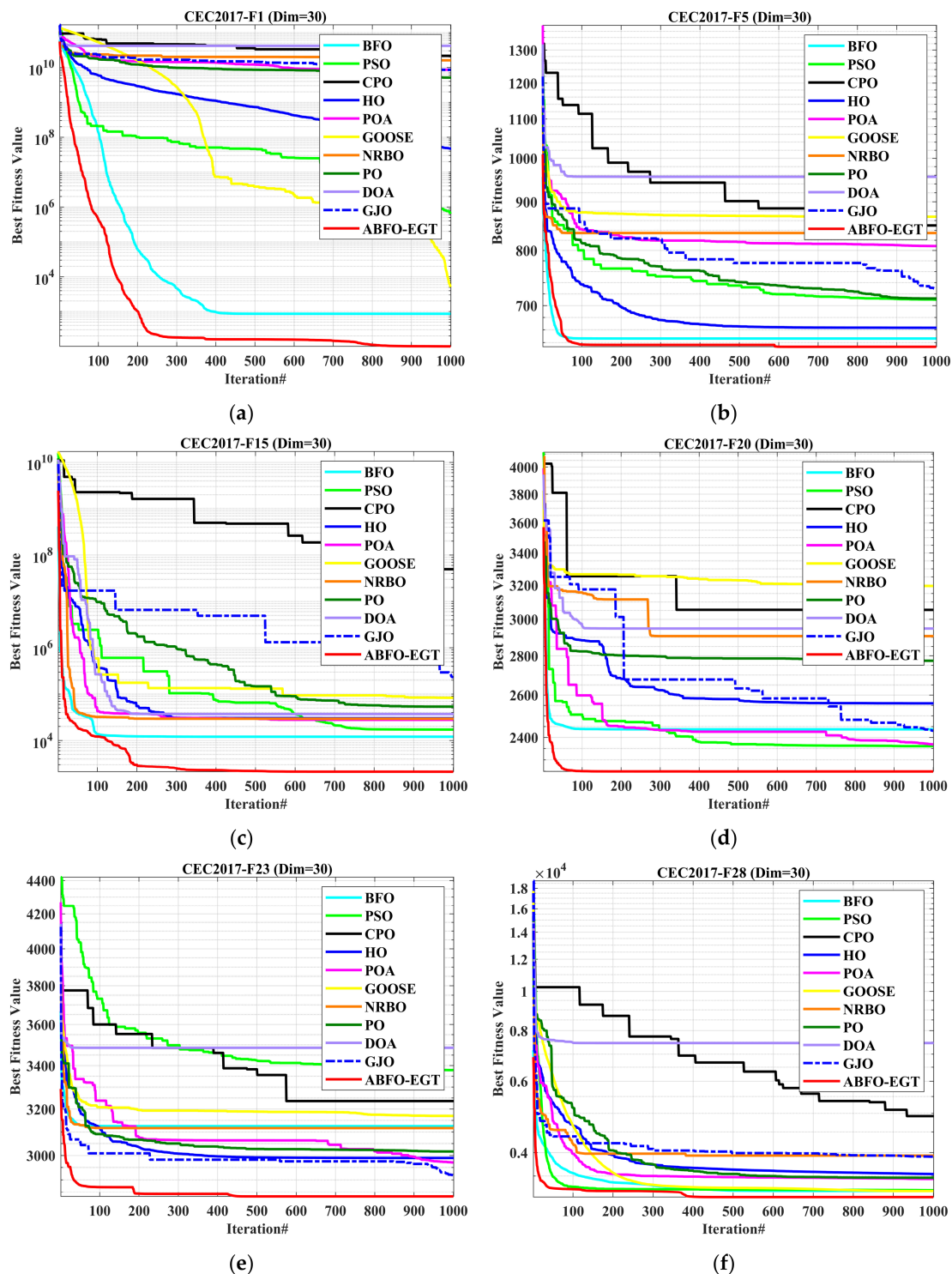


Figure 3. Convergence Curves of the ABFO-EGT Algorithm on the Benchmark Test Functions. (a) The convergence curve of f_1 ; (b) the convergence curve of f_5 ; (c) the convergence curve of f_{15} ; (d) the convergence curve of f_{20} ; (e) the convergence curve of f_{23} ; (f) the convergence curve of f_{28} .

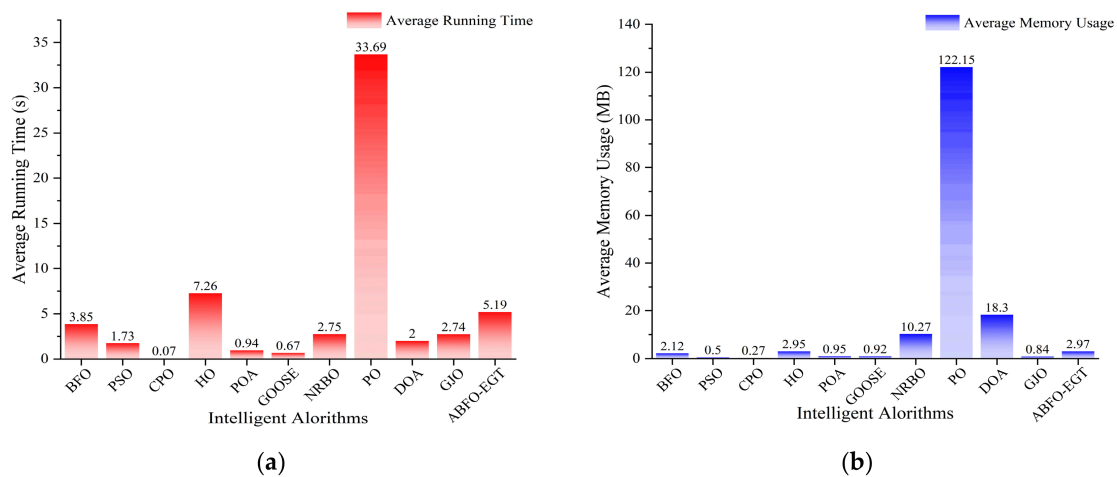


Figure 4. Comparison of Computational Costs among the 11 Algorithms. (a) Average Running Time; (b) Average Memory Usage.

As illustrated in Figure 4, regarding the comparison of running times, CPO demonstrates the most prominent performance with a running time of only 0.07 s. In contrast, the PO algorithm exhibits the highest computational overhead at 33.39 s. The running time of the ABFO-EGT algorithm is 5.19 s, representing an increase of 1.34 s compared to the BFO algorithm (3.85 s). This increment in running time is primarily attributed to the integration of the evolutionary game theory (EGT) mechanism. Specifically, the requirement to calculate the strategy evolution process and the payoff matrix during each iteration theoretically increases the computational burden. Nevertheless, compared to the HO (7.26 s) and PO (33.69 s) algorithms, ABFO-EGT maintains a competitive level of computational efficiency, and its running time is at a moderate level. In terms of average memory usage, the PO algorithm exhibits extremely high memory requirements, with an occupancy of up to 122.15 MB, which is significantly higher than that of other algorithms. In contrast, the CPO (0.27 MB) and PSO (0.5 MB) algorithms are extremely lightweight. The memory usage of ABFO-EGT is 2.97 MB, slightly higher than that of the BFO algorithm (2.12 MB). This indicates that ABFO-EGT exhibits excellent performance in memory control. Although additional storage structures are introduced to implement strategic games and dynamic population updates, its total memory occupancy is still controlled within 5 MB, which is much lower than that of PO and DOA. This demonstrates that the algorithm has low dependence on hardware resources and good scalability when solving complex engineering problems.

4.3. Statistical Analysis and Significance Testing

This study adopts non-parametric test methods to conduct an in-depth analysis of the experimental results. This approach verifies the effectiveness of ABFO-EGT from a mathematical statistics perspective. It also objectively evaluates the performance differences between ABFO-EGT and the other 10 swarm intelligence optimization algorithms. The analysis process is divided into two stages. First, the Friedman test performs a global evaluation of the overall algorithm rankings. Subsequently, the Bonferroni–Dunn test is utilized for post hoc analysis to identify specific significant differences.

4.3.1. Friedman Test for Individual Test Functions

To objectively evaluate the statistical significance of the performance differences between the proposed ABFO-EGT and the 10 competitor algorithms, the Friedman non-parametric test was conducted on each of the 30 CEC 2017 benchmark functions. Based on the mean obtained from 30 independent runs, this test verifies whether a statistically

significant difference exists in the overall performance of the 11 algorithms on each specific function. The core statistical results are detailed in Table 3.

Table 3. Friedman test results of 11 swarm intelligence optimization algorithms for individual test functions within the 30 CEC 2017 benchmark functions.

Function	Chi-Square	<i>p</i> -Value
f_1	192.773	5.1760×10^{-36}
f_2	186.664	9.6655×10^{-35}
f_3	175.845	1.7053×10^{-32}
f_4	188.100	4.8584×10^{-35}
f_5	166.755	1.3025×10^{-30}
f_6	164.245	4.3013×10^{-30}
f_7	161.473	1.6088×10^{-29}
f_8	171.291	1.4989×10^{-31}
f_9	158.364	7.0514×10^{-29}
f_{10}	156.091	2.0750×10^{-28}
f_{11}	184.882	2.2681×10^{-34}
f_{12}	188.418	4.1716×10^{-35}
f_{13}	170.218	2.5000×10^{-31}
f_{14}	146.855	1.6525×10^{-26}
f_{15}	149.945	3.8250×10^{-27}
f_{16}	138.882	7.1420×10^{-25}
f_{17}	101.255	3.0561×10^{-17}
f_{18}	132.100	1.7411×10^{-23}
f_{19}	162.027	1.2358×10^{-29}
f_{20}	100.409	4.5132×10^{-17}
f_{21}	163.391	6.4598×10^{-30}
f_{22}	80.391	4.2083×10^{-13}
f_{23}	162.718	8.8965×10^{-30}
f_{24}	153.209	8.1442×10^{-28}
f_{25}	188.218	4.5910×10^{-35}
f_{26}	135.909	2.8994×10^{-24}
f_{27}	158.118	7.9235×10^{-29}
f_{28}	181.255	1.2861×10^{-33}
f_{29}	140.264	3.7214×10^{-25}
f_{30}	179.182	3.4643×10^{-33}

Table 3 presents the Friedman test results for the 11 algorithms on individual test functions. At the significance level of $\alpha = 0.05$, the *p*-values of all 30 CEC 2017 benchmark functions (ranging from 5.1760×10^{-36} to 4.2083×10^{-13}) are all well below the threshold. Therefore, the null hypothesis is strongly rejected for all test cases. Furthermore, the large Chi-square statistics (ranging from 80.391 to 192.773) indicate that the algorithm performances significantly deviate from the random state. These results statistically confirm the existence of significant differences in the optimization capabilities of the 11 algorithms on each function.

4.3.2. Global Performance Evaluation Based on the Friedman Test

In this section, the mean (Mean) and standard deviation (Std) of the optimal values obtained by the 11 algorithms across 30 independent tasks ($N = 30$) were subjected to the Friedman non-parametric test. The calculated experimental statistics and the resulting average rankings are presented in Table 4.

Table 4. Friedman Test Results.

/		Mean	Std
Test Results	N	30	30
	Chi-square	214.073	107.655
		1.8578×10^{-40}	1.5841×10^{-18}
Friedman Test Value	BFO	2.633	4.167
	PSO	3.967	5.267
	CPO	9.933	6.300
	HO	5.500	5.900
	POA	5.500	6.033
	GOOSE	7.400	6.700
	NRBO	7.900	6.900
	PO	5.967	6.233
	DOA	10.100	9.700
	GJO	6.067	7.167
	ABFO-EGT	1.033	1.633

The experimental results show that the Chi-square value of the Friedman test based on the mean is 214.073, with a corresponding p -value of 1.8578×10^{-40} . The Chi-square value based on the standard deviation is 107.655, with a corresponding p -value of 1.5841×10^{-18} . At the significance level $\alpha = 0.05$, both p -values are far smaller than the threshold, which strongly demonstrates that there are extremely significant statistical differences in the performance distributions of the 11 algorithms. According to the ranking data in Table 4, ABFO-EGT ranks first in terms of both Mean and Std, with ranking values of 1.033 and 1.633, respectively. This demonstrates that ABFO-EGT not only possesses the best solution accuracy but also exhibits exceptional robustness.

4.3.3. Post Hoc Significance Analysis Based on Bonferroni–Dunn Test

To further determine whether ABFO-EGT is significantly superior to the other 10 algorithms, this study conducted the Bonferroni–Dunn test using ABFO-EGT as the control (baseline) algorithm. For the Bonferroni–Dunn test, the criterion for judging a significant difference between two algorithms is that the difference in their performance rankings must exceed a critical difference (CD). The calculation results for the critical difference in the Bonferroni–Dunn test are as follows. $CD_{0.05}$ and $CD_{0.1}$ represent the critical differences at the significance levels of 0.05 and 0.1, respectively.

$$CD_{0.05} = 2.21, CD_{0.1} = 1.99$$

Based on the above results, the Bonferroni–Dunn test results for the 11 algorithms are presented in Figure 5. The ranking position of the baseline algorithm (ABFO-EGT) is represented by the pink horizontal line. Meanwhile, the critical thresholds at 95% and 90% significance levels are given by the red solid line and black dashed line respectively.

As shown in Figure 5a, the rankings of PSO, CPO, HO, POA, GOOSE, NRBO, PO, DOA, and GJO are all significantly beyond the threshold at the 95% significance level. This confirms that the solution accuracy of ABFO-EGT is significantly superior to that of these nine algorithms. Although BFO has the closest ranking to the control algorithm, the leading advantage of ABFO-EGT is still very obvious. In Figure 5b, except for the control algorithm, the rankings of the remaining ten algorithms are all significantly beyond the threshold at the 95% significance level. This further confirms that ABFO-EGT has more excellent anti-interference ability than other algorithms when solving complex optimization problems.

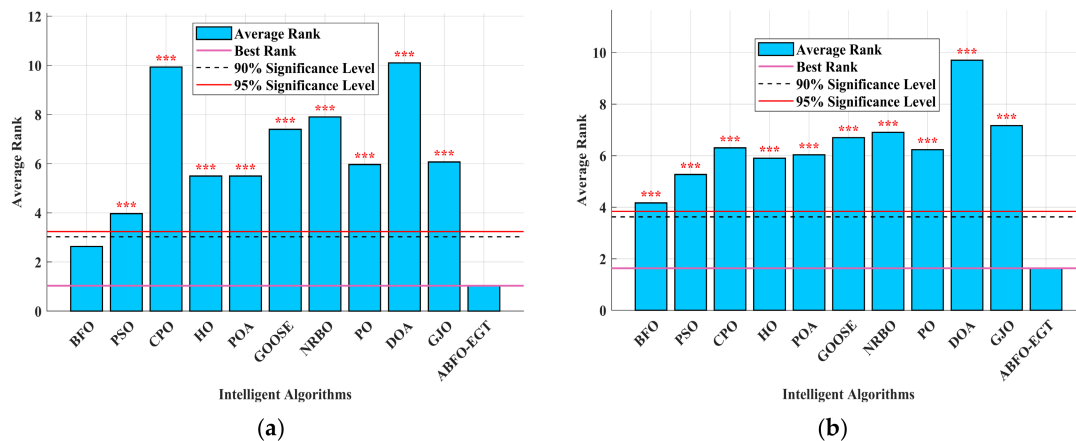


Figure 5. Bar chart of Bonferroni–Dunn test results. (a) Mean; (b) Standard Deviation (The *** indicates significant differences).

Combining the results of the Friedman test and Bonferroni–Dunn test, the ABFO-EGT algorithm statistically significantly outperforms the comparison algorithms. ABFO-EGT achieves an extremely low average ranking value and robust performance under the strict critical difference test. This fully demonstrates that the introduction of adaptive mechanisms and game theory strategies conclusively and effectively improves optimization performance.

5. Parameter Sensitivity Analysis and Orthogonal Experiments

The performance of intelligent optimization algorithms relies heavily on hyperparameters. Unreasonable parameter settings are likely to induce premature convergence to local optima or degrade the convergence rate. For the purpose of systematically analyzing the impact of core parameters on ABFO-EGT and identifying the optimal parameter combination, this study adopts the Orthogonal Array Design (OAD) method for parameter tuning. Compared with the traditional single-factor variable method, orthogonal experiments can obtain comprehensive and reliable statistical conclusions with the minimum number of experiments. This approach significantly improves the efficiency and scientificity of parameter tuning.

5.1. Experimental Design

5.1.1. Selection of Factors and Levels

Based on a profound analysis of the ABFO-EGT algorithm mechanism, four key hyperparameters are adopted as experimental factors. Specifically, three levels are established for every single factor. Among them, Level 2 is the initial default value of the algorithm, serving as the central reference point of the experiment. Level 1 and Level 3 are reasonably extended to both sides based on the default value to cover the effective value range of the parameters. The experimental factors and level settings are listed in Table 5.

Table 5. Settings of Experimental Factors and Levels.

Factor Code	Parameter Symbol	Parameter Description	Level 1	Level 2	Level 3
A	H	Memory Pool Size	5	10	15
B	α	Evolutionary Game Theory Learning Rate	0.05	0.10	0.20
C	γ	Power-Law Decay Coefficient of Spawning Step Size	3	5	7
D	$SD_{initial}$	Initial Spawning Step Size	0.5	1.0	2.0

5.1.2. Selection of Orthogonal Array and Experimental Scheme

For experiments with 4 factors and 3 levels, there are two standard orthogonal arrays available: $L_9(3^4)$ and $L_{27}(3^{13})$. The L_9 array only requires 9 groups of experiments with a small computational load, but it cannot examine the interaction effects between parameters. The L_{27} array requires 27 groups of experiments. Although the computational load is increased, it can more comprehensively investigate the potential interaction effects between parameters, resulting in higher reliability of experimental results. Considering the rigor requirements of this study, the $L_{27}(3^{13})$ orthogonal array is selected, and the first 4 columns are used to arrange the experimental factors.

To eliminate the influence of algorithmic randomness on the results, each parameter combination was executed 30 times independently under identical test conditions. The arithmetic mean of the final fitness values obtained from the 30 runs was used as the primary evaluation metric. Meanwhile, the standard deviation served as a supplementary metric. All experimental tests are carried out in a uniform hardware and software environment, so that the fairness and comparability of experimental results can be ensured.

5.2. Experimental Results and Analysis

5.2.1. Range Analysis

The most widely adopted intuitive analysis method in orthogonal experiments is range analysis. The average response value (K) and range value (R) for each factor at various levels are calculated to judge the primary and secondary influence order on algorithm performance. It is demonstrated that a higher R value leads to a more prominent effect of factor level fluctuation on algorithm performance. For the minimization optimization problem investigated in this study, a smaller K value represents superior algorithm performance at that specific level. The relevant range analysis results are listed in Table 6.

Table 6. Results of range analysis.

Factor Code	Mean Value of Level 1 (K_1)	Mean Value of Level 2 (K_2)	Mean Value of Level 3 (K_3)	Range Value (R)	Rank
A	3.9158×10^3	2.7467×10^3	2.9055×10^3	1.1691×10^3	1
B	3.2591×10^3	3.2077×10^3	3.1012×10^3	1.5787×10^2	4
C	3.4161×10^3	3.1147×10^3	3.0372×10^3	3.7889×10^2	3
D	3.2383×10^3	3.4715×10^3	2.8582×10^3	6.1327×10^2	2

Ranked by the degree of influence, the impact of the four parameters on algorithm performance is ordered as follows: memory pool size (A) > initial spawning step size (D) > power-law decay coefficient of spawning step size (C) > evolutionary game theory learning rate (B). The influence mechanism of each parameter can be explained combined with the algorithm mechanism.

(1) Memory Pool Size (A)

Memory pool size displays the largest range value, making it the most critical factor affecting algorithm performance. The memory pool stores historically successful mutation factors, serving as the foundation for the symmetrical mutation strategy. A pool that is too small leads to insufficient historical experience and low diversity in sampled mutation factors. Consequently, the algorithm struggles to adapt to complex search landscapes. Conversely, an overly large pool introduces outdated experience from early iterations, which interferes with current mutation strategy selection. Therefore, an optimal threshold exists for the memory pool size. In this study, Level 2 (a size of 10) achieves the optimal performance, striking the best balance between historical richness and timeliness.

(2) Initial Spawning Step Size (D)

The initial spawning step size ranks second in influence. It determines the search scope during the early iterations of the algorithm. A larger initial step size enhances early population diversity, expands the global search range, and reduces the risk of falling into local optima. However, an excessively large step size causes overly dispersed early search behavior, thereby reducing convergence speed. Within the experimental range of this study, Level 3 (an initial step size of 2.0) yields the best performance. This indicates that appropriately increasing the initial step size is more advantageous for solving complex problems.

(3) Power-Law Decay Coefficient of Spawning Step Size (C)

The decay coefficient ranks third in impact. This parameter dictates the contraction rate of the step size. A larger decay coefficient causes a rapid contraction of the step size, which improves late-stage local exploitation precision but may lead to insufficient exploration. A smaller coefficient maintains a large-scale search for a longer duration, offering stronger exploration capabilities at the expense of exploitation accuracy. Experimental results indicate that increasing the decay coefficient within the set range improves algorithm performance. This suggests that the benchmark problems in this study are better suited to a faster step-size contraction rhythm.

(4) Evolutionary Game Theory Learning Rate (B)

The EGT learning rate exhibits the smallest range value, exerting the weakest influence on algorithm performance. This result validates the strong robustness of the EGT-based adaptive strategy switching mechanism. The replicator dynamics equation inherently possesses self-stabilizing properties. Here, strategy selection probabilities are primarily driven by actual strategy payoffs, while the learning rate merely controls the step size of probability updates. Over a wide range of values, the algorithm automatically adjusts strategy probabilities to reasonable levels via payoff feedback. Consequently, fluctuations in the learning rate have a minimal impact on overall performance.

5.2.2. Main Effect Analysis

To more intuitively demonstrate the influence trend of the level changes of each factor on the algorithm performance, the main effect plot of the parameters is presented in Figure 6.

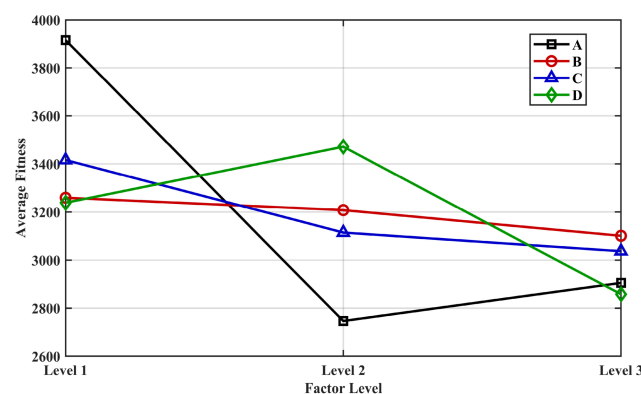


Figure 6. Main Effect Plot (A–D are factor codes).

It can be observed that the curve of parameter A (memory pool size H) has the largest span. This indicates that the algorithm's optimization performance is highly sensitive to changes in this parameter. Therefore, it serves as the core control parameter affecting the solution accuracy of the ABFO-EGT. The main effect curve of parameter A drops sharply

between Level 1 and Level 2, reaches the lowest point at Level 2, and then rises slightly at Level 3. This indicates that there is an obvious nonlinear optimal threshold for the selection of parameter A. The dynamic balance between global exploration and local exploitation will be destroyed by overly small or large values, and Level 2 is regarded as the optimal level to preserve this balance.

For parameter D (initial spawning step size $SD_{initial}$), its main effect curve follows a trend of initial growth followed by subsequent decline, with the optimal fitness performance attained at Level 3. This suggests that a higher initial value can effectively enhance the population diversity in the algorithm's early initialization stage. Consequently, it mitigates the likelihood of premature stagnation in local optima within intricate multi-dimensional solution spaces.

The main effect curves of parameter C (power-law decay coefficient of spawning step size γ) and parameter B (evolutionary game theory learning rate α) both show a trend of monotonically decreasing fitness with the increase in level. In particular, for parameter C, the fitness value is further improved when it is increased to Level 3. This means that within the set experimental range, assigning higher values to these two parameters is more conducive to improving the convergence accuracy of the algorithm.

Based on the above analysis, each parameter functions independently at different levels. The optimal parameter configuration that maximizes the algorithm performance can be determined by selecting the lowest average fitness point corresponding to each factor in the main effect plots. Therefore, the optimal parameter configuration for maximizing algorithm performance can be explicitly determined. It is identified by selecting the lowest average fitness point corresponding to each factor in the main effect plots. The optimal parameter combination is determined as: $H = 10$, $\alpha = 0.2$, $\gamma = 7$, and $SD_{initial} = 2$.

6. Case Study

The natural gas pipeline network is located in Heilongjiang Province, China, with pure natural gas as the transmission medium. The landforms are dominated by mountains, hills and plains, resulting in obvious undulations in the pipeline network layout. The natural gas pipeline network in this area was completed and put into operation in June 2003, and some pipelines pass through urban roads. The pipeline network needs to supply gas to residential users, enterprises and public institutions within the coverage area. The 3PE anti-corrosion coating is adopted for the key protection areas of the pipelines. The pipeline network data are presented in Table 7.

Table 7. Pipeline Network Data.

Name	Value	Remarks
Pipe Diameter	DN60–DN200	The pipe diameter at the accident point is DN150
Pipe Material	X52 Spiral Steel Pipe	-
Total Length of Pipeline Network	20 km	-
Design Pressure	2.5 MPa	-
Operating Pressure	0.01–1.2 MPa	The operating pressure of the pipeline at the accident point is 1.0 MPa
Average Daily Gas Transmission Volume	600,000 m ³	-

A gas pipeline leakage accident occurred in the existing urban sub-high-pressure gas pipeline, and the detailed leakage parameters are presented in Table 8.

Table 8. Leakage Parameter Table.

Name	Value
Adiabatic Index of Natural Gas	1.29
Atmospheric Ambient Pressure	101,325 Pa
Leakage Source Pressure	1 MPa
Critical Pressure Ratio	0.547541
Pipe Diameter	150 mm
Leakage Hole Diameter	20 mm
Leakage Duration	3600 s

After calculation, the total leakage mass is 687.8592 kg after the leakage accident lasts for 1 h. Aiming at the two most likely chain accidents after leakage, namely jet fire and vapor cloud explosion (VCE), the accident consequence assessment and accident risk assessment are completed, and the core results are presented in Table 9.

According to the results of accident consequence and risk assessment, the accident risk level is categorized as “Medium Risk”. The cross-regional emergency repair cost is empirically calibrated based on the historical emergency response data of gas companies, which quantitatively reflects the additional time delays, communication costs and logistics expenses incurred by cross-departmental coordination compared to intra-jurisdictional dispatch. The locations of the emergency repair team and the accident point have been marked in Figure 7. The parameters of the emergency repair path optimization model are presented in Tables 10 and 11.

Table 9. Results of Accident Consequence and Risk Assessment.

Accident Consequence	Core Parameter	Value	Accident Consequence Level	Accident Risk Level
Jet Fire	Maximum Hazard Radius	0.48 m	1 (Minor accident)	Low Risk
	Fatality Area	0.72 m ²		
	Number of Fatalities	0.00006029		
	Economic Loss	2.028 million CNY		
Vapor Cloud Explosion	Maximum Hazard Radius	106.4 m	2 (Major accident)	Medium Risk
	Fatality Area	35,547.8144 m ²		
	Number of Fatalities	2.9621		
	Economic Loss	13.457 million CNY		

Table 10. Parameter Settings for Emergency Repair Path Optimization.

Parameter Name	Value	Remarks
m	4	Number of teams (B1, B2, B3, B4)
n	1	Number of accident points (A1)
σ_{11}, σ_{21}	1	Accident point A1 is within the jurisdiction of teams B1 and B2
σ_{31}, σ_{41}	0	Accident point A1 is not within the jurisdiction of teams B3 and B4
C_{31}, C_{41}	2.5	Cross-regional emergency repair cost of teams B3 and B4 for accident point A1
$g(x_1)$	Medium Risk	Risk level assessment of accident A1
ϕ_1	1	Importance level of accident point A1

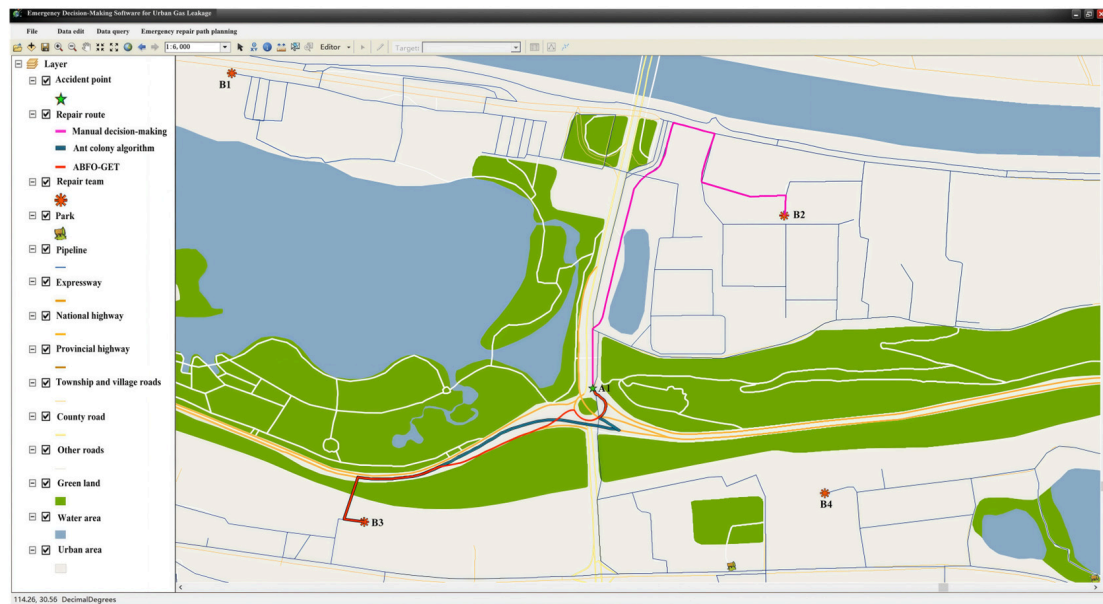


Figure 7. Emergency Repair Paths of the Three Methods with the Lowest Total Cost.

Table 11. Road Data of Emergency Repair Teams.

Repair Team	Road Number	Road Length (m)	Total Road Weight
B1	1	2308.78	1.8
	2	2198.57	1.8
	3	2255.34	1.8
	4	2033.82	1.67
	5	2261.35	1.73
	6	2167.62	1.8
	7	2262.47	1.8
B2	1	1645.87	2
	2	1865.06	2
	3	1349.64	2
	4	1714.11	2
	5	2456.39	2.1
B3	1	1001.99	1.8
	2	1499.04	1.85
	3	1768.10	1.85
	4	1102.40	1.8
	5	1384.26	1.8
B4	1	1567.83	2
	2	1590.09	2
	3	2214.93	2
	4	1624.83	2

This study conducts a comparative analysis to verify the superiority of the proposed method in emergency repair path optimization. The repair scheduling paths generated by ABFO-EGT are compared against four other approaches. These include a manual decision-making strategy, the Ant Colony Optimization (ACO) algorithm, Particle Swarm Optimization (PSO), and the Gray Wolf Optimizer (GWO). The parameters of the ABFO-EGT are set to Level 2 in Section 5.1.1. The visualization results of the emergency repair paths generated by the three methods with the lowest total cost are presented in Figure 7. The statistical results of total emergency path repair cost for different methods are shown in Table 12.

Table 12. Total Emergency Repair Path Cost Results of the Five Methods.

Method Name	Selected Emergency Repair Team	Road Number	Road Length (m)	Total Road Weight	Total Cost (F)
Manual Decision-Making	B2	3	1349.64	2	2699.29
Ant Colony Optimization (ACO)	B3	4	1102.40	1.8	1984.32
Particle Swarm Optimization (PSO)	B3	4	1102.40	1.8	1984.32
Gray Wolf Optimizer (GWO)	B3	5	1384.26	1.8	2491.67
ABFO-EGT	B3	1	1001.99	1.8	1803.58

Results indicate that the manual decision-making strategy relies solely on the shortest distance principle. It allocates the repair task of accident point A1 to team B2 strictly within its jurisdiction. Consequently, this strategy results in the highest total path cost. The ACO algorithm breaks through the administrative jurisdiction limitation of manual scheduling. It allocates the repair task of A1 to team B3 outside its jurisdiction. As a result, its total path cost is significantly lower than that of the manual decision-making scheme. PSO algorithm selects the same scheme as the ACO algorithm. GWO algorithm achieves a lower total path cost than the manual decision-making scheme. However, its total path cost is significantly higher than that of ACO and PSO algorithms. Notably, the ABFO-EGT also allocates the repair task of A1 to team B3 outside its jurisdiction. However, it achieves a further reduction in the total path cost. Overall, its comprehensive performance remains superior to that of other methods. Quantitative results demonstrate a significant reduction in the total emergency repair path cost for the ABFO-EGT algorithm. Specifically, the cost is reduced by 33.2% compared to the manual decision-making strategy, and by 27.6% compared to the GWO algorithm. Furthermore, it achieves a 9.1% reduction compared to both the ACO and PSO algorithms. These results prove that the proposed algorithm can successfully overcome the inherent limitations of traditional scheduling methods. Ultimately, it achieves a more economical and efficient emergency repair path scheduling.

7. Conclusions

To address the emergency repair scenario of urban natural gas pipeline networks, this paper constructs a cross-regional emergency repair path planning optimization model considering dynamic path conditions. An Adaptive Bitterling Fish Optimization algorithm with Evolutionary Game Theory (ABFO-EGT) is proposed. Compared with other swarm intelligence algorithms, this algorithm fundamentally balances the global exploration and local exploitation capabilities. ABFO-EGT exhibits excellent convergence accuracy and robustness in the CEC 2017 benchmark function tests. It achieves the optimal mean results on 29 test functions and the optimal standard deviation results on 21 test functions. ABFO-EGT maintains a significant lead in both the Friedman test and Bonferroni–Dunn test, demonstrating its superior global optimization performance. In the engineering case study, the emergency repair path obtained by ABFO-EGT demonstrates a significantly lower comprehensive cost compared to other methods. Quantitative results show that the total cost is reduced by 33.2% compared to the manual decision-making strategy. Furthermore, it achieves a cost reduction of 27.6% against the Gray Wolf Optimizer (GWO), and a 9.1% reduction against both the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms. Overall, the model and algorithm proposed in this paper

exhibit excellent engineering practicability. They provide efficient and accurate decision support for the emergency repair scheduling of energy infrastructures.

8. Discussion

The optimization model and ABFO-EGT algorithm proposed in this paper exhibit satisfactory solution accuracy, stability, and engineering applicability. This is demonstrated in both the CEC 2017 benchmark function tests and the case study. However, the famous No Free Lunch (NFL) theorem states a key principle: no single optimization method is universally ideal for all problems. Therefore, evaluating a novel method requires a critical analysis of its strengths and weaknesses. This analysis must incorporate both experimental and theoretical perspectives.

(1) Strengths

Experimentally, the proposed method effectively solves the emergency repair path optimization problem. This model takes into account jurisdiction ownership, cross-regional dispatch costs, path weights, and accident risk levels. Theoretically, this success stems from a deep integration. It combines evolutionary game theory (EGT) and symmetric mutation strategies. Standard swarm intelligence algorithms often suffer from premature convergence. This is primarily due to their greedy selection mechanisms. In contrast, EGT provides a rigorous mathematical framework. It simulates rational decision-making in multi-agent systems. This ensures a dynamic balance between global exploration and local exploitation. This theoretical mechanism effectively maintains population diversity. Furthermore, it mathematically enhances the capability to escape local optima.

(2) Weaknesses

Despite these advantages, this study still has certain limitations in practical application scenarios.

① Algorithmic Weaknesses

The introduction of the evolutionary game payoff matrix is computationally demanding. It inevitably increases the asymptotic time complexity per iteration. This may affect the real-time responsiveness of the algorithm. This issue is particularly prominent when scaled to ultra-large networks. Second, the algorithm exhibits inherent parameter sensitivity. This study determined optimal parameter levels through rigorous orthogonal experiments and main effects plot analysis. However, these strictly tuned static parameters may limit the algorithm. Its adaptability could be restricted when transferred to entirely different objective spaces.

② Modeling Weaknesses

The current model has not yet incorporated dynamic factors such as road closure, traffic disturbance, temporary control and concurrent multiple accidents. Therefore, its adaptability in complex real-time environments still has room for improvement. The current objective function does not consider factors such as response time and task load balancing during the team scheduling process.

In summary, subsequent research can be further expanded in the following three aspects.

- (1) Future research can utilize data-driven, lightweight surrogate models. These models can approximate some highly complex exact calculations. Precise game evaluations will be reserved only for elite individuals. This approach will significantly compress the computing time per iteration.
- (2) Dynamic road networks and concurrent accident mechanisms will be introduced into the model. Real-time traffic information, road condition changes, and multi-accident scenarios will be integrated. This will create an optimization model that closely aligns

with actual emergency response processes. Consequently, the algorithm's decision-making capabilities in complex emergency scenarios will be enhanced.

- (3) The model will be further extended from a multi-objective cooperative optimization perspective. It will build upon the comprehensive repair cost. Key factors like team response timeliness, task distribution balance, and accident handling priorities will be jointly considered. This will facilitate the construction of a robust multi-objective decision-making framework. This framework will more accurately meet the diverse requirements of practical emergency dispatching.

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

Funding: This research was funded by the National Natural Science Foundation of China (No. 52104065, 51674086), Heilongjiang Province' Basic Research Support Program for Excellent Young Teachers (No. YQJH2023086), Northeast Petroleum University Talent Introduction and Research Initiation Funding Project (No. 2021KQ18), Northeast Petroleum University National Natural Science Foundation Cultivation Fund (No. 2023GPL-14), Heilongjiang Outbound Postdoctoral Funding (No. 216240064), and Qingyan Talent Support Program Project of Daqing Heilongjiang Province (No. DQXY202403).

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to express their sincere and heartfelt gratitude to the anonymous reviewers and the editorial team of the journal for their insightful comments, professional revisions, and constructive suggestions throughout the peer-review process. The authors also gratefully acknowledge the administrative and technical support provided by the research institution and laboratory colleagues during the completion of this work. Sincere thanks are extended to all members of the research group for their valuable academic discussions, continuous encouragement, and practical assistance in algorithm programming, experimental design, data statistics and result analysis. In addition, the authors would like to appreciate all researchers and scholars whose published works and literature provided important theoretical foundations and research references for this study.

Conflicts of Interest: Junfei Liu was employed by the Daqing Longfeng Gas Branch of Kunlun Gas Co., Ltd., Xingwang Wang was employed by the SINOPEC, Zhe Xu, Yongbin Liu and Haibin Liang were employed by the Gas Technology Institute of Petrochina Kunlun Gas Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, Z.; Xue, Q. To fully exert the important role of natural gas in building a modern energy security system in China: An understanding of China's National 13th Five-Year Plan for Natural Gas Development. *Nat. Gas Ind. B* **2017**, *4*, 270–277. [\[CrossRef\]](#)
2. Li, J.; Zhang, S.; Wang, X.; Cao, X.; Wang, J.; Wu, Z. Vertical gas dispersion from high-pressure pipeline failures: Characterizing hazardous cloud formation in open environments. *Nat. Gas Ind. B* **2026**, *13*, 111–127. [\[CrossRef\]](#)
3. Wang, X.; Hou, T.; Gao, W.; Yu, K.; Zhang, T.; Tan, Y. Experimental study on the diffusion process of natural gas from buried pipelines to underground confined spaces. *Nat. Gas Ind. B* **2024**, *11*, 603–615. [\[CrossRef\]](#)
4. Dong, X.; Gong, Q.; Li, Y.; Kong, Z. The process and trends of China's energy transition and the strategic repositioning of natural gas. *Nat. Gas Ind. B* **2025**, *12*, 750–765. [\[CrossRef\]](#)
5. Lin, Y.; Chen, B.; Wang, J.; Bie, Z. A Combined Repair Crew Dispatch Problem for Resilient Electric and Natural Gas System Considering Reconfiguration and DG Islanding. *IEEE Trans. Power Syst.* **2019**, *34*, 2755–2767. [\[CrossRef\]](#)
6. Biswas, S.; Cavdar, B.; Geunes, J. A review on response strategies in infrastructure network restoration. *arXiv* **2024**, arXiv:2407.14510. [\[CrossRef\]](#)

7. Li, X.; Ma, J.; Pasman, H.; Zhang, R. Dynamic risk investigation of urban natural gas pipeline accidents using Stochastic Petri net approach. *Process Saf. Environ. Prot.* **2023**, *178*, 933–946. [\[CrossRef\]](#)
8. Bu, F.; Lu, Q.; Jia, B.; Gao, X.; Zhang, H.; Wang, W.; Wang, Z. A Review of Research on Leakage and Diffusion Characteristics of Buried Gas Pipelines. *J. Pipeline Sci. Eng.* **2025**, 100426. [\[CrossRef\]](#)
9. Alkhaleel, B.A.; Liao, H.; Sullivan, K.M. Risk and resilience-based optimal post-disruption restoration for critical infrastructures under uncertainty. *Eur. J. Oper. Res.* **2022**, *296*, 174–202. [\[CrossRef\]](#)
10. Cui, X.; Li, B.; Wang, S.; Zhang, S.; Yang, X.; Ren, Y.; Xiao, Y. Timeliness-oriented rush repair optimization of workforce scheduling and routing for logically complex systems under uncertainty. *Reliab. Eng. Syst. Saf.* **2025**, *253*, 110574. [\[CrossRef\]](#)
11. Qiu, D.; Wang, Y.; Zhang, T.; Sun, M.; Strbac, G. Hierarchical multi-agent reinforcement learning for repair crews dispatch control towards multi-energy microgrid resilience. *Appl. Energy* **2023**, *336*, 120826. [\[CrossRef\]](#)
12. Shin, Y.; Kim, S.; Moon, I. Integrated optimal scheduling of repair crew and relief vehicle after disaster. *Comput. Oper. Res.* **2019**, *105*, 237–247. [\[CrossRef\]](#)
13. Panahi, S.; Delavar, M. A GIS-Based Dynamic Shortest Path Determination in Emergency Vehicles. *World Appl. Sci. J.* **2008**, *3*, 88–94.
14. Li, S.; Ma, Z.; Teo, K.L. A new model for road network repair after natural disasters: Integrating logistics support scheduling with repair crew scheduling and routing activities. *Comput. Ind. Eng.* **2020**, *145*, 106506. [\[CrossRef\]](#)
15. Puente-Castro, A.; Rivero, D.; Pazos, A.; Fernandez-Blanco, E. A review of artificial intelligence applied to path planning in UAV swarms. *Neural Comput. Appl.* **2022**, *34*, 153–170. [\[CrossRef\]](#)
16. Zhou, J.; Li, Z.; Liu, S.; Li, C.; Zhao, Y.; Zhou, Z.; Liang, G. Hybrid genetic algorithm for parametric optimization of surface pipeline networks in underground natural gas storage harmonized injection and production conditions. *Nat. Gas Ind. B* **2025**, *12*, 234–250. [\[CrossRef\]](#)
17. Mohemmed, A.W.; Sahoo, N.C.; Geok, T.K. Solving shortest path problem using particle swarm optimization. *Appl. Soft Comput.* **2008**, *8*, 1643–1653. [\[CrossRef\]](#)
18. Zhang, S.; Zhou, Y.; Li, Z.; Pan, W. Grey wolf optimizer for unmanned combat aerial vehicle path planning. *Adv. Eng. Softw.* **2016**, *99*, 121–136. [\[CrossRef\]](#)
19. Miao, F.; Yan, G.; Xie, Y. Optimization of UAV-based moving target search in maritime search and rescue: An enhanced multi-strategy golden jackal optimization algorithm. *Ocean Eng.* **2025**, *341*, 122841. [\[CrossRef\]](#)
20. Larsen, R.B.; Jouffroy, J.; Lassen, B. On the premature convergence of particle swarm optimization. In *Proceedings of the 2016 European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 1922–1927. [\[CrossRef\]](#)
21. Nezami, O.M.; Bahrampour, A.; Jamshidlou, P. Dynamic Diversity Enhancement in Particle Swarm Optimization (DDEPSO) Algorithm for Preventing from Premature Convergence. *Procedia Comput. Sci.* **2013**, *24*, 54–65. [\[CrossRef\]](#)
22. Hu, G.; Huang, F.; Chen, K.; Wei, G. MNEARO: A meta swarm intelligence optimization algorithm for engineering applications. *Comput. Methods Appl. Mech. Eng.* **2024**, *419*, 116664. [\[CrossRef\]](#)
23. Chen, S.; Liu, T.; Dong, H.; Bao, Y.; Yu, Z.; Guan, B.; Jiang, X.; Zhou, S.; Wang, S. Generic layout optimization design methodology for China's loop-star natural gas field pipeline network. *Nat. Gas Ind. B* **2024**, *11*, 616–629. [\[CrossRef\]](#)
24. He, Q.; Feng, Y.; Ma, L.; Lu, X.; Zhang, J.; Ma, C. Dynamic Interval Prediction of Subway Passenger Flow Using a Symmetry-Enhanced Hybrid FIG-ICPO-XGBoost Model. *Symmetry* **2026**, *18*, 288. [\[CrossRef\]](#)
25. Zhou, B.; Shi, C.; Yan, N.; Chu, Y. Improved Chimpanzee Optimization Algorithm Based on Multi-Strategy Fusion and Its Application in Multiphysics Parameter Optimization. *Symmetry* **2026**, *18*, 108. [\[CrossRef\]](#)
26. Zhang, D.; Wang, Z.; Zhao, Y.; Sun, F. Multi-Strategy Fusion Improved Dung Beetle Optimization Algorithm and Engineering Design Application. *IEEE Access* **2024**, *12*, 97771–97786. [\[CrossRef\]](#)
27. Ye, H.; Zhao, W. Multi-Strategy Fusion Improves Time Optimal Trajectory Planning of Particle Swarm Optimization Algorithm. *ES Gen.* **2025**, *8*, 1515. [\[CrossRef\]](#)
28. Wu, Z.; Zhou, T.; Fan, J.; Zhang, R.; Li, Z.; Hu, K. A Feature Selection Method Based on an Improved Sand Cat Swarm Optimization Algorithm with Multi-Strategy Fusion. *Entropy* **2026**, *28*, 595. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Li, L.; Ding, Y.; Zhou, X.; Zhu, X.; Wu, Z.; Peng, W.; Zhang, J.; Jia, C. Multi-Strategy Fusion Improved Walrus Optimization Algorithm for Coverage Optimization in Wireless Sensor Networks. *Biomimetics* **2026**, *11*, 72. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Zhao, X.; Duan, Y.; Liu, F. A multi-strategy enhanced RIME-based metaheuristic with adaptive strategy collaboration for global optimization and art image segmentation. *Sci. Rep.* **2026**, *16*, 22484. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Seyyedabbasi, A.; Arasteh, B.; Gurhanli, A.; Rasheed, J. Multi-strategy Sea Horse Optimization algorithm for UAV path planning. *Front. Robot. AI* **2026**, *13*, 1792384. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Kuninaga, Y.; Maeda, M.; Chikuba, Y. Set-based differential evolution with exponential and binomial crossovers for discrete optimization problem. *Artif. Life Robot.* **2026**, *31*, 441–456. [\[CrossRef\]](#)

33. Liu, J.; Ouyang, H.; Li, S.; Zou, D.; Li, F. ASGDE: Adaptive stagnation guided differential evolution algorithm for global optimization problems. *Clust. Comput.* **2026**, *29*, 409. [\[CrossRef\]](#)
34. Zareian, L.; Rahebi, J.; Shayegan, M.J. Bitterling fish optimization (BFO) algorithm. *Multimed. Tools Appl.* **2024**, *83*, 75893–75926. [\[CrossRef\]](#)
35. He, C.; Ouyang, H.; Huang, W.; Li, S.; Zhang, C.; Ding, W.; Zhan, Z.-H. An adaptive heuristic algorithm with a collaborative search framework for multi-UAV inspection planning. *Appl. Soft Comput.* **2025**, *174*, 112969. [\[CrossRef\]](#)
36. Li, X.; Guo, Y. Particle Swarm Optimization Based on Cubic Chaotic Mapping and Random Differential Mutation. *Algorithms* **2026**, *19*, 297. [\[CrossRef\]](#)
37. Liu, G.; Su, X.; Zhou, R.; Tian, Y. Reinforcement learning-based adaptive particle swarm optimization. *Complex Intell. Syst.* **2026**. [\[CrossRef\]](#)
38. Chen, B.; Ouyang, H.; Li, S.; Zou, D. Differential evolution algorithm with a complementary mutation strategy and data Fusion-Based parameter adaptation. *Inf. Sci.* **2024**, *668*, 120522. [\[CrossRef\]](#)
39. Qin, L.; Li, Z.; Lu, X.; Gao, Y. Differential Evolution for Deep Learning: A Comprehensive Survey. *Arch. Comput. Methods Eng.* **2026**. [\[CrossRef\]](#)
40. Wu, G.; Mallipeddi, R.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization*; Technical Report; Nanyang Technological University: Singapore, 2017. Available online: https://www.researchgate.net/publication/317228117_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2017_Competition_and_Special_Session_on_Constrained_Single_Objective_Real-Parameter_Optimization (accessed on 8 May 2026).
41. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995*; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948. [\[CrossRef\]](#)
42. Abdel-Basset, M.; Mohamed, R.; Abouhawwash, M. Crested Porcupine Optimizer: A new nature-inspired metaheuristic. *Knowl.-Based Syst.* **2024**, *284*, 111257. [\[CrossRef\]](#)
43. Amiri, M.H.; Mehrabi Hashjin, N.; Montazeri, M.; Mirjalili, S.; Khodadadi, N. Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm. *Sci. Rep.* **2024**, *14*, 5032. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Trojovský, P.; Dehghani, M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors* **2022**, *22*, 855. [\[CrossRef\]](#) [\[PubMed\]](#)
45. 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. [\[CrossRef\]](#)
46. Sowmya, R.; Premkumar, M.; Jangir, P. Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107532. [\[CrossRef\]](#)
47. 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. [\[CrossRef\]](#) [\[PubMed\]](#)
48. Peraza-Vázquez, H.; Peña-Delgado, A.F.; Echavarría-Castillo, G.; Morales-Cepeda, A.B.; Velasco-Álvarez, J.; Ruiz-Perez, F. A Bio-Inspired Method for Engineering Design Optimization Inspired by Dingoes Hunting Strategies. *Math. Probl. Eng.* **2021**, *2021*, 9107547. [\[CrossRef\]](#)
49. Chopra, N.; Ansari, M.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [\[CrossRef\]](#)

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.