APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 391–414
https://doi.org/10.14743/apem2025.3.548

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Dynamic Harris Hawks Optimization and deep reinforcement learning framework for autonomous vehicle path planning

Zou, Q.[a,e], **Yuan, X.**[b], **Liu, F.**[a,e,*], **Yin, Y.**[c], **Chen, P.**[d]

[a]School of Intelligence Technology, Geely University of China, P.R. China
[b]School of Accountancy, Hunan International Business Vocational College, P.R. China
[c]Geely Automobile Research Institute (Ningbo) Co., Ltd, P.R. China
[d]Research Services, Coventry University, Coventry, United Kingdom
[e]Visual Computing and Virtual Reality Key Laboratory of Sichuan Province, Sichuan Normal University, P.R. China

ABSTRACT

Urban intelligent transportation systems require real-time, near-optimal routing for autonomous vehicles navigating dynamic and uncertain traffic. We propose a Harris Hawks Optimization–deep reinforcement learning framework (HHO-DRL) that unites HHO's global exploration with DRL's adaptive policy search through (i) a dynamic-weight fusion scheme that continuously balances exploration and exploitation and (ii) a bidirectional experience-feedback loop that exchanges elite solutions between the two solvers. On 23 CEC-2014 benchmark functions and five classical multimodal tests, HHO-DRL lowers mean error by up to three orders of magnitude relative to PSO and adaptive HHO, demonstrating superior robustness and precision. In 30 × 30 grid-world simulations with 30 % obstacle density, it generates vehicle routes 35 % shorter than those produced by Grey Wolf Optimization and 25 % shorter than adaptive HHO, while preserving smooth, collision-free trajectories. These results confirm that the proposed dual-mechanism delivers fast, high-quality solutions for high-dimensional, dynamic path-planning and other complex engineering optimization tasks.

ARTICLE INFO

Keywords:
Dynamic path planning;
Harris Hawks optimization;
Deep reinforcement learning;
Autonomous vehicles;
Dynamic weight fusion;
Bidirectional feedback;
Intelligent transportation systems;
Real-time navigation

**Corresponding author:*
i@liufengyu.cn
(Liu, F.)

Article history:
Received 23 January 2025
Revised 28 May 2025
Accepted 19 June 2025

1. Introduction

In recent years, Intelligent Transportation Systems (ITS) have experienced rapid development, providing novel technological approaches for urban traffic management and vehicle dispatching [1-3]. In this context, enabling autonomous vehicles to perform real-time route planning in dynamic, complex, and uncertain road environments has become a core challenge. Traditional deterministic route planning methods often struggle to efficiently obtain near-global optimal solutions for high-dimensional, multimodal, and nonlinear complex optimization problems. In contrast, metaheuristic algorithms exhibit superior robustness and adaptability in addressing com-

plex decision-making tasks, making them widely applicable in optimization scenarios [4]. However, even metaheuristic approaches can become trapped in local optima or suffer from imbalanced search processes when dealing with problems that have pronounced multi-peak, dynamic, and stochastic characteristics [5].

As a newly developed bio-inspired algorithm, the Harris Hawks Optimization (HHO) algorithm mimics the group hunting behavior of hawks, achieving a balance between global and local searches [6]. Studies have shown that HHO can outperform classical metaheuristic algorithms (e.g., PSO, GWO, WOA) when handling multimodal functions and high-dimensional problems, demonstrating strong performance in complex optimization tasks [7, 8]. However, when applied to highly dynamic vehicle route planning contexts, HHO may still face challenges such as premature convergence and limited adaptability [9]. To address these issues, some researchers have enhanced HHO's performance by introducing adaptive parameters and hybrid strategies to improve the coordination between global exploration and local exploitation [10]. Nevertheless, in scenarios requiring real-time route planning—characterized by significant time-varying features—HHO still needs further refinements to enhance its convergence speed and optimization accuracy. Notably, attempts have been made to incorporate advanced strategies such as orthogonal learning or Lévy mutations to handle high-dimensional optimization [11], and multi-population schemes have also proven beneficial for complex multimodal functions.

Deep reinforcement learning (DRL) has shown remarkable potential in decision-making and control domains by continuously optimizing policies through a state-action-reward loop that interacts with the environment [12]. DRL has been successfully applied to intelligent decision-making, autonomous driving, and route planning, effectively extracting high-value strategies in complex and uncertain environments [13]. By leveraging deep neural networks, DRL can extract features from high-dimensional state spaces and achieve efficient mappings from perception to decision-making [14]. However, DRL often faces challenges such as insufficient exploration and slow policy convergence during the early stages of learning, requiring significant time to reach a satisfactory solution [15]. Additionally, in high-dimensional continuous action spaces, DRL is prone to becoming trapped in suboptimal policies due to its limited global search capability, which hinders thorough exploration [16]. Consequently, integrating DRL with metaheuristic algorithms has emerged as a natural choice to provide heuristic guidance and identify promising solution regions [17]. Previous studies suggest that combining evolutionary or swarm intelligence approaches with DRL can significantly improve exploration efficiency and convergence performance in complex tasks [18]. For instance, hybrid DRL-evolutionary frameworks have shown promise in improving policy adaptation under dynamic conditions [19], while knowledge-guided DRL strategies have facilitated navigation in complex urban scenes [20]. Yet, despite these efforts, systematically integrating HHO with DRL for dynamic vehicle route planning remains underexplored [21]. Moreover, applying adaptive fusion mechanisms or self-adaptive parameters to further accelerate convergence has only seen preliminary exploration in related fields [22]. Meanwhile, larger-scale verification scenarios using real-time traffic data underscore the need for enhanced scalability [23], and complex Internet of Vehicles environments call for robust handling of uncertainty [24] and multi-agent cooperation [25]. Ensuring smooth trajectories and safety under dynamic conditions has also attracted research attention [26].

To address this gap, this study proposes a vehicle route planning framework that integrates HHO with DRL, aiming to fully leverage HHO's global search advantages and DRL's policy-learning capabilities. Moving beyond a simple overlay approach, a dynamic weight fusion mechanism is introduced, which flexibly adjusts the involvement levels of HHO and DRL based on measurable indicators such as fitness improvement and population diversity. This mechanism achieves an adaptive balance between global exploration and local exploitation. When the algorithm becomes trapped in local optima or shows slow improvement, increasing DRL's global exploration proportion helps it escape these traps. Conversely, when the algorithm demonstrates significant progress or maintains high population diversity, enhancing HHO's local search intensity further refines the solution quality.

In addition, a two-way feedback mechanism is proposed, where high-quality solutions identified by DRL are fed back into HHO at the end of each iteration, guiding the next search phase toward

promising regions. Simultaneously, the sequence of excellent solutions filtered by HHO is incorporated into the DRL experience pool with weighted importance, accelerating DRL's policy updates. Through this bidirectional, closed-loop information exchange, HHO and DRL achieve collaborative evolution, addressing the limitations of their independent use and demonstrating enhanced adaptability in highly dynamic and complex route planning scenarios. This framework delivers significant improvements in global-local search synergy, adaptive coordination, and real-time decision-making, providing new insights for future research on the integration of HHO and DRL.

In terms of experiments, this study employs a variety of standard test functions and the complex CEC-2014 benchmark functions to systematically evaluate the proposed HHO-DRL algorithm. The results demonstrate outstanding performance in global, local, and combined search tasks. Compared to standalone HHO, standalone DRL, and other classical metaheuristic algorithms such as PSO, GWO, WOA, MHHO, and AHHO, the proposed method exhibits superior capabilities in avoiding local optima, improving search efficiency, and adapting to dynamic environments. Furthermore, when applied to real-world dynamic vehicle route planning scenarios—ranging from simple to highly complex road networks and traffic conditions—the HHO-DRL approach generates shorter, safer, and smoother routes. These findings validate the effectiveness of the dynamic weight fusion and bidirectional feedback mechanisms, while also providing empirical references for applying similar strategies to other high-dimensional, nonlinear decision-making problems. In the long run, this work establishes a foundation for the integration of HHO and DRL, opening new research directions for addressing complex route planning challenges in the field of intelligent transportation.

The remainder of this paper is organized as follows: Section 2 introduces the research background and related work, including the development status and theoretical foundations of HHO and DRL. Section 3 elaborates on the integrated algorithmic framework and the implementation details of the dynamic weighting and bidirectional feedback mechanisms. Section 4 presents experimental evaluations and performance analyses of the algorithm, comparing it with classical baseline methods to highlight its advantages. Section 5 concludes the study and discusses future research directions. Future work may consider enhancing robustness using stochastic model predictive control, extending multi-agent cooperative decision-making frameworks, and achieving distributional robustness in uncertain environments [27]. Ongoing studies also emphasize policy smoothness [28] and controlled interactions in connected autonomous vehicle ecosystems [29], which may further improve the adaptability and safety of autonomous driving models.

2. Related work

2.1 Harris Hawks Optimization algorithm

The Harris Hawks Optimization (HHO) algorithm simulates the hunting behavior of hawks preying on rabbits. When the escape energy $|E| \geq 1$, the algorithm performs the corresponding operation [30]. The decision for the hawks to encircle the prey is determined by $|E| \geq 0.5$, while the decision to dive is made by checking if a random number $r \geq 0.5$. The escape energy function is defined as follows:

$$E = 2E_0(1 - \frac{t}{T}) \tag{1}$$

Here, E_0 is a randomly initialized value within the range (-1, 1), t denotes the current iteration number, and T represents the total number of iterations. When the escape energy $|E| \geq 1$, a random walk is performed. The position update is given by the following formula:

$$X(t+1) = \begin{cases} X_k(t) - r_1|X_k(t) - 2r_2X(t)| \\ (X_k(t) - X_m(t)) - r_3(lb - r_4(ub - lb)) \end{cases} \tag{2}$$

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{3}$$

Here, X_k represents a randomly selected individual from the Harris hawk population; X_r denotes the position of the best individual in the Harris hawk population, i.e., the position of the "rabbit"; X_m corresponds to the mean position of the population; N is the population size; $r_1 \sim r_4$ follows a normal distribution [0,1]; lb and ub represent the lower and upper bounds of the problem, respectively [31].

When the escape energy $|E| < 1$, the algorithm performs local search. Depending on the energy level, four updating strategies are selected [32]. When the escape energy $|E| \geq 0.5$ and $r \geq 0.5$, the algorithm conducts a "soft encirclement." The position update is given by the following formula:

$$X(t+1) = \Delta X(t) - E|JX(t) - X(t)| \tag{4}$$

$$\Delta X(t) = X_r - X(t) \tag{5}$$

Here, $J \in [0,2]$ represents the update step size, and $\Delta X(t)$ indicates the difference between the position with the highest fitness value and the current position. When the escape energy meets conditions $|E| \geq 0.5$ and $r < 0.5$, the algorithm executes "hard encirclement," resulting in the direct capture of the rabbit. The position update formula is as follows:

$$X(t+1) = X_r(t) - E|\Delta X(t)| \tag{6}$$

When the escape energy meets conditions $|E| < 0.5$ and $r \geq 0.5$, the rabbit cannot escape, and the Harris hawks employ a more sophisticated soft encirclement strategy as follows:

$$Y = X_r(t) - E|JX_r(t) - X(t)| \tag{7}$$

$$Z = Y + S \times LF(D) \tag{8}$$

$$LF(D) = \frac{u\sigma}{100 \times |v|\frac{1}{\beta}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{1/\beta} \tag{9}$$

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \tag{10}$$

Here, D denotes the dimension of the problem, S represents a random vector in D dimension, $LF(x)$ corresponds to the Lévy flight formula, u, v is a random number within (0,1), and β is set to 1.5. When the escape energy meets conditions $|E| < 0.5$ and $r < 0.5$, the rabbit may attempt to flee, and the Harris hawks adopt a gradually convergent soft encirclement approach as follows:

$$Y = X_r(t) - E|JX_r(t) - X_m(t)| \tag{11}$$

$$Z = Y + S \times LF(D) \tag{12}$$

2.2 Deep reinforcement learning with actor-critic methods

In both deterministic and stochastic environments, policies represented by neural networks can be updated through gradient ascent, which typically requires estimating the value function. A commonly used architecture for this purpose is the actor-critic framework, where the actor is responsible for policy optimization, and the critic estimates the value function [33]. In deep reinforcement learning, both the actor and critic are represented by nonlinear neural networks. The actor updates its policy parameters using gradients derived from the policy gradient theorem, while the critic approximates the value function of the current policy.

In dynamic vehicle route planning, a Markov Decision Process (MDP) model is used to describe the transitions between different states of a vehicle [34]. It is defined as follows:

- State s: Describes the specific condition of the vehicle at a given time, such as position, speed, and so forth.
- Action a: The operations the vehicle can take, such as accelerating, decelerating, or turning.
- State transition probability $P(s'|s,a)$: The probability of moving from state s to state s' by taking action a.
- Reward function $R(s,a)$: The reward obtained by taking action a in state s.

The state transition equation is as follows:

$$P_{ss'} = P(s_{t+1} = s' | s_t = s, a_t = a) \tag{13}$$

In dynamic vehicle planning, the state at the next time step depends solely on the current state and the current action. In reinforcement learning, it is necessary to learn a policy (actor) π and a value function (critic) Q. The value function $Q^{\pi}(s, a)$ represents the expected total reward obtained by taking action a in state s.

$$Q^{\pi}(s_t, a_t) = E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s, a_t = a] \tag{14}$$

Here, $E[\cdot]$ represents the expected value; γ^k denotes the discount factor γ raised to the power of k, where $\gamma \in [0,1]$ is used to determine the present value of future rewards. A smaller value of γ reduces the weight of future rewards. R_{t+k+1} represents the reward received at time $t + k + 1$.

In reinforcement learning, the policy π defines the strategy an agent uses to decide which actions to take in different states. It is represented as a probability distribution over actions for a given state, indicating the likelihood of selecting a specific action in a particular state, as expressed by the following formula:

$$\pi(a|s) = P(a_t = a | s_t = s) \tag{15}$$

Here, $P(a_t = a | s_t = s)$ represents the conditional probability of taking action a at time t when the state is s.

Policy gradient methods optimize the policy by maximizing the cumulative reward. The gradient of the objective function $J(\theta)$ is given as:

$$\nabla_{\theta} J(\theta) = E[\nabla_{\theta} \, log \, \pi_{\theta} \, (a|s) Q^{\pi}(s, a)] \tag{16}$$

Here, $\nabla_{\theta} J(\theta)$ represents the gradient of the objective function $J(\theta)$ with respect to the parameter θ, indicating how to adjust θ to maximize the objective function $J(\theta)$. $\pi_{\theta}(a|s)$ denotes the parameterized policy π, while $Q^{\pi}(s, a)$ represents the action-value function under policy π, which is the expected total reward obtained by taking action a in state s and proceeding from state s.

The policy parameters θ are updated using gradient ascent to optimize the policy:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta) \tag{17}$$

Here, θ represents the parameter vector of the policy, which controls the behavior of the policy, and α denotes the learning rate, specifying the step size of each update.

DQN combines deep neural networks with Q-learning to estimate the Q-value function. The Q-value function is updated using the following formula:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \, max_{a'} \, Q \, (s_{t+1}, a') - Q(s_t, a_t)) \tag{18}$$

Here, $Q(s_t, a_t)$ represents the Q-value of taking action a_t in state s_t, which indicates the expected total reward for a given state and action. R_{t+1} denotes the immediate reward received at time $t + 1$, and $max_{a'} \, Q \, (s_{t+1}, a')$ represents the maximum Q-value in state s_{t+1}, indicating the Q-value of the optimal action in the next state.

DQN uses deep neural networks to approximate the Q-value function $Q(s, a; \phi)$, where ϕ represents the parameters of the neural network. The specific steps are as follows:

Step 1: Neural network architecture

A deep neural network is employed, where the input is the state s, and the output corresponds to the Q-values of all possible actions.

Step 2: Calculation of target Q-value

The target Q-value is computed using the Bellman equation:

$$y = R_{t+1} + \gamma \, max_{a'} \, Q \, (s_{t+1}, a'; \phi^{-}) \tag{19}$$

Here, y represents the target Q-value; R_{t+1} denotes the immediate reward received at time $t+1$; and $max_a, Q\ (s_{t+1}, a'; \phi^-)$ is the maximum Q-value in state s_{t+1}, derived using the target network parameters ϕ^-, representing the Q-value of the optimal action in the next state.

Step 3: Loss function

The loss function is defined as the mean squared error between the current Q-value and the target Q-value:

$$L(\phi) = E[(y - Q(s_t, a_t; \phi))^2] \tag{20}$$

Step 4: Gradient descent

The network parameters ϕ are updated using gradient descent as follows:

$$\phi \leftarrow \phi - \alpha \nabla_\phi L(\phi) \tag{21}$$

Here, $\nabla_\phi L(\phi)$ represents the gradient of the loss function $L(\phi)$ with respect to the parameters ϕ.

To address the issue of reward sparsity, reward shaping is introduced. The new reward function R is defined as:

$$R = R' + r(s, a, s') \tag{22}$$

Here, R' represents the target reward, and $r(s, a, s')$ denotes the additional reward introduced to guide the learning process. The additional reward can be defined using a potential function $\Phi(s)$, which maps state s to a real number to measure the potential value of the state, as shown in the following formula:

$$r(s, a, s') = \gamma \Phi(s') - \Phi(s) \tag{23}$$

Here, γ represents the discount factor, and s' denotes the next state resulting from action a. By this method, the additional reward can be incorporated into the original reward R', providing the agent with richer information during the learning process and thereby accelerating the learning speed.

In dynamic vehicle planning, the potential function needs to account for the following factors:

- Target position: The target location the vehicle needs to reach.
- Path smoothness: Ensuring the smoothness and feasibility of the vehicle's path.
- Obstacle avoidance: The vehicle must avoid obstacles.
- Continuity of states and actions: Considering the physical properties and dynamic constraints of the vehicle.

By carefully considering these factors, this study employs the Euclidean distance as the potential function, effectively guiding the vehicle toward the target position while meeting the requirements for path smoothness and obstacle avoidance [35, 36].

$$\Phi(s) = -\|s - s^*\|_2 + \lambda \cdot \text{PathSmoothess(s)} - \mu \cdot \text{ObstacleProximity(s)} \tag{24}$$

Here, s represents the current state, s^* denotes the target state, and $\|s - s^*\|_2$ indicates the Euclidean distance between the current state and the target state. PathSmoothness(s) measures the smoothness of the path, with smoother paths being preferable. ObstacleProximity(s) evaluates the proximity to obstacles, where greater distances from obstacles are preferred. λ and μ are weighting factors used to balance the importance of the target position, path smoothness, and obstacle avoidance.

3. Method

Dynamic vehicle planning aims to determine the optimal path and strategy in dynamic environments. By integrating Harris Hawks Optimization (HHO) and deep reinforcement learning, the proposed approach combines the strengths of both methods to achieve more efficient and effective dynamic vehicle planning. Furthermore, a dynamic weight fusion mechanism and a bidirectional feedback mechanism are introduced to improve the algorithm's adaptability and collaborative optimization capabilities.

The flowchart of the proposed algorithm is shown in Fig. 1.

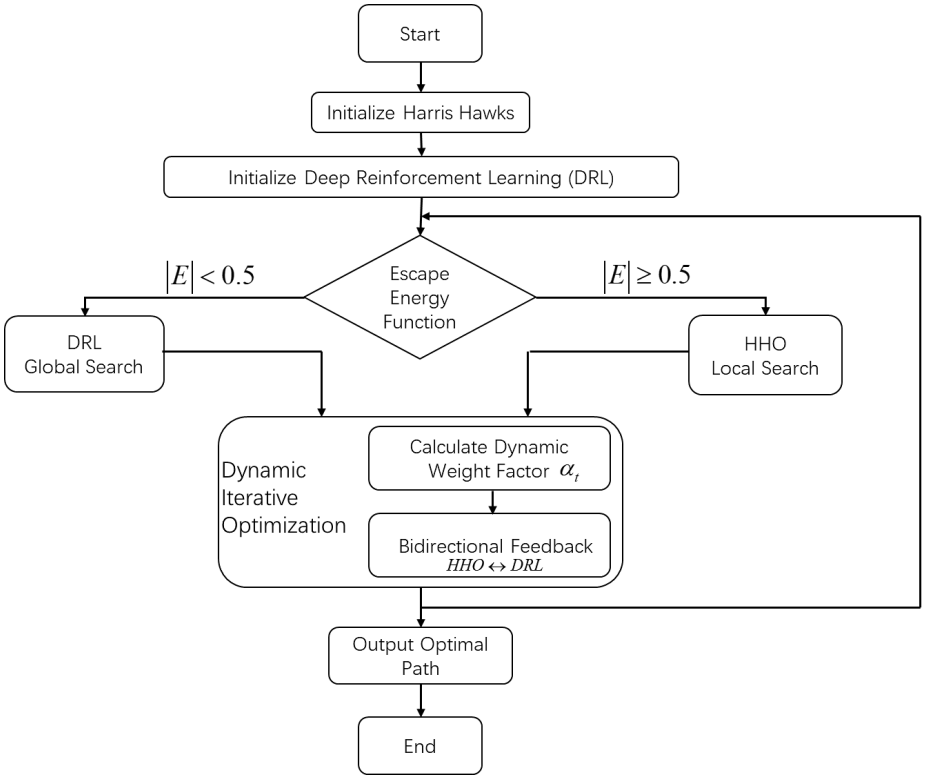


Fig. 1 Flowchart of the Harris Hawks Optimization algorithm integrated with deep reinforcement learning

A detailed description of the algorithm is given below.

Step 1: Initialization

Initialize the positions and velocities of the Harris Hawks population.

$$X(0) = \{X_i(0) | i = 1,2,\ldots,N\} \tag{25}$$

Here, $X_i(0)$ represents the position of the i-th individual in the Harris Hawks population, and N denotes the population size.

Initialize the policy network (actor) parameters θ and the value network (critic) parameters ϕ in deep reinforcement learning, providing a starting point for subsequent policy updates and value estimation.

Step 2: Global exploration (HHO)

In the Harris Hawks Optimization algorithm, the escape energy function is utilized to determine whether global exploration or local exploitation behaviors will be performed during the current iteration:

$$E(t) = 2(1 - \frac{t}{T})R - 1 \tag{26}$$

Here, R is a random number within the range (-1, 1), t denotes the current iteration number, and T represents the total number of iterations.

When $|E(t)| \geq 1$ holds, global exploration is performed, and the update formula is as follows:

$$X(t+1) = X_{rand}(t) - r|X_{rand}(t) - 2rX(t)| \tag{27}$$

Here, X_{rand} represents the randomly selected individual position, and r is a random number.

Step 3: Local exploitation (DRL)

In deep reinforcement learning, the Markov Decision Process (MDP) model is used to describe the transitions of a vehicle between different states, as detailed in Section 2.2.1.

In each state, the actor network selects an action based on policy $\pi_\theta(a|s)$, while the critic network estimates the value function $V_\phi(s)$ of the current policy.

Step 4: Iterative optimization

During each iteration, the original decision-making process for selecting HHO's global exploration or DRL's local exploitation was determined by the energy function $E(t)$. However, to enable more flexible and adaptive regulation, a dynamic weight fusion mechanism and a bidirectional feedback mechanism are introduced:

To better control the involvement of HHO and DRL during different phases of practical experiments, a dynamic weight factor $\alpha_t \in [0,1]$ is defined to balance their contribution proportions in the t -th iteration. To ensure that the determination of α_t , this study employs two quantitative indicators: "Diversity" and "Improvement Ratio."

– Fitness improvement ratio I_t

Assume $f_{best}(t)$ represents the global optimal solution fitness (objective function value) in the t -th generation. Theoretically, the goal of the optimization problem is to minimize $f_{best}(t)$ as much as possible (or approach zero). The relative improvement rate between consecutive generations is defined as follows:

$$I_t = \frac{|f_{best}(t-1) - f_{best}(t)|}{max(\varepsilon, |f_{best}(t-1)|)} \tag{28}$$

Here, ε is a very small positive number used to prevent the denominator from becoming zero.

When I_t is relatively large, it indicates that the current iteration has significantly improved the optimal solution, reflecting good progress in the algorithm during this iteration. Conversely, when I_t is very small or approaches zero, it suggests that the algorithm has made limited progress in recent iterations and may be stuck in a local optimum or a stagnation state.

– Population diversity D_t

To measure the distribution of the HHO population in the search space, the population diversity is defined as follows:

$$D_t = \frac{1}{N}\sum_{i=1}^{N} \|X_i(t) - \bar{X}(t)\| \tag{29}$$

Here, $X_i(t)$ represents the position vector of the i -th individual in the t -th generation, and $\bar{X}(t)$ denotes the mean position vector of the population in the t -th generation. A very small D_t indicates that the population has converged to a narrow region, increasing the likelihood of getting stuck in local optima. Conversely, a relatively large D_t suggests that the search maintains high diversity and strong exploration potential.

– Determination method for dynamic weight α_t

The study aims to increase the weight of DRL when population diversity is low and the fitness improvement ratio is small, thereby helping to overcome stagnation. Conversely, when fitness improvement is significant or diversity remains high, preference is given to HHO for more refined local exploitation.

At iteration t, α_t quantifies the relative contribution of DRL's local exploitation versus HHO's global exploration at iteration t. From a theoretical sensitivity standpoint, abrupt or large fluctuations in α_t can destabilize convergence—leading to oscillations or premature stagnation—whereas smooth, moderate adjustments help preserve search stability and maintain overall algorithmic efficiency. Taking these factors into account, the dynamic weight is calculated as follows:

$$\alpha_t = \frac{\frac{D_{ref}}{D_t + \varepsilon} \cdot \frac{I_{ref}}{I_t + \varepsilon}}{1 + \frac{D_{ref}}{D_t + \varepsilon} \cdot \frac{I_{ref}}{I_t + \varepsilon}} \tag{30}$$

Here, D_{ref} and I_{ref} are reference constants, and ε is a very small positive number to prevent division by zero.

When D_t is small and I_t is also small, $\frac{D_{ref}}{D_t}$ and $\frac{I_{ref}}{I_t}$ will increase, making α_t approach 1, thereby increasing the proportion of DRL's global exploration. Conversely, when improvements are significant or diversity is sufficient, α_t approaches 0, allowing HHO to dominate in that generation.

Based on the above process, in each generation of the iteration, α_t is used to combine DRL (global exploration) and HHO (local exploitation) in a specific proportion. This combination is dynamically adjusted based on the data observed during the iterative process, rather than relying on fixed threshold values for decision-making.

To strengthen the deep collaboration between HHO and DRL during the search process, this study implements a bidirectional feedback mechanism through information exchange at the end of each iteration. The core objective of this mechanism is to transfer high-quality features identified by DRL during global exploration to the HHO population, enabling HHO to conduct more targeted local searches in the subsequent iteration. Simultaneously, the high-quality solutions identified by HHO during local refinement are incorporated into DRL's experience pool as weighted samples, allowing DRL to quickly focus on high-value state-action spaces.

Through this closed-loop interaction, both methods reinforce each other in subsequent iterations, leading to improved overall optimization performance.

– DRL → HHO feedback

At the end of the current iteration, DRL identifies several superior solutions during global exploration, such as discovering route segments that are significantly better than those in previous generations for path planning. Let the global optimal solution obtained by DRL after convergence in this iteration be $X^*_{DRL}(t)$. To enable HHO to focus more on this promising region in the next generation, this solution can serve as a "guiding center" to reinitialize some individuals in the HHO population.

Let $\beta \in (0,1)$ be a proportional factor controlling the number of individuals to be reinitialized. β quantifies the proportion of individuals to be reinitialized by the DRL module before HHO's global exploration. From a theoretical sensitivity standpoint, excessively large values of β may over-scatter the population, undermining local refinement and slowing convergence; conversely, overly small β values can limit beneficial feedback, reducing the synergy between DRL and HHO. Therefore, selecting a moderate, balanced β is theoretically recommended to sustain robust convergence efficiency. For example, βN individuals are reset, where N represents the population size. This can be expressed as:

$$X_i(t+1) = X^*_{DRL}(t) + \sigma \cdot N(0, I) \tag{31}$$

Here, $i \in \Omega_{DRL2HHO}$, $i \in \Omega_{DRL2HHO}$ represents the index set of individuals to be reinitialized, σ is a hyperparameter controlling the intensity of perturbation, and $N(0, I)$ is a Gaussian random vector with a mean of 0 and a covariance of the identity matrix.

Using this formulation, a portion of the HHO population in the next iteration will be concentrated around the high-potential region identified by DRL. This approach enhances the precision and effectiveness of the local search in the subsequent iteration.

– HHO → DRL feedback

After completing an iteration, HHO generates a set of candidate solutions, from which the top K high-quality solutions, denoted as $\left\{X^{(k)}_{HHO}(t)\right\}_{k=1}^{K}$, are selected. For the path planning problem, these high-quality solutions correspond to state-action-reward sequences, such as discrete samples of vehicle states and control decisions along the optimal paths. These sequences can then be directly added to DRL's experience replay buffer B as training samples for subsequent policy updates.

To encourage DRL to prioritize the high-quality information provided by HHO during training, higher sampling weights can be assigned to these newly added samples. Let the objective function of the problem be denoted as $f(\cdot)$. For each sequence corresponding to an HHO high-quality solution $X^{(k)}_{HHO}(t)$, the weight factor w_k is defined as:

$$w_k = \frac{\exp(-\lambda f(X_{HHO}^{(k)}(t)))}{\sum_{m=1}^{K} \exp(-\lambda f(X_{HHO}^{(m)}(t)))} \tag{32}$$

Here, $\lambda > 0$ is a parameter used to control the steepness of the weight distribution. A larger w_k indicates higher solution quality (lower objective function value), meaning that it should be prioritized in subsequent DRL training. From a theoretical sensitivity standpoint, excessively large values of w_k may overemphasize high-quality but noisy samples, risking policy divergence and instability. Conversely, overly small values of w_k can underutilize valuable heuristic information, slowing policy improvement. Therefore, selecting a moderate, balanced range for w_k is theoretically recommended to ensure robust and stable DRL updates.

After injecting these weighted samples into the experience pool, when DRL draws samples from the buffer B for training, weighted sampling is performed according to w_k, as follows:

$$P(sample(s_{k,j}, a_{k,j})) = \frac{w_k}{\sum_{m=1}^{K} w_m} \tag{33}$$

With weighted sampling, the policy gradient update formula for DRL can be expressed as:

$$\nabla_\theta J(\theta) \approx \sum_{k=1}^{K} \sum_{j=1}^{J_k} P(sample(s_{k,j}, a_{k,j})) \nabla_\theta \log \pi_\theta (a_{k,j} | s_{k,j}) Q_\phi(s_{k,j}, a_{k,j}) \tag{34}$$

Here, J_k represents the length of the state-action pair sequence for the corresponding solution (in path planning, this refers to the number of discrete states along the path), and Q_ϕ denotes the DRL value function estimation.

State-action pairs from high-quality solutions are more likely to be selected for policy updates, reinforcing DRL's preference for high-value regions and thereby accelerating policy convergence.

Through the mathematical description of the bidirectional feedback mechanism, this study explicitly defines the implementation method for repositioning population individuals in the DRL → HHO direction and establishes quantitative metrics and sampling strategies for converting high-quality solutions into weighted training samples in the HHO → DRL direction.

This mechanism works in conjunction with the dynamic weight fusion mechanism: the former adjusts the proportion of HHO and DRL involvement in each iteration, while the latter utilizes information feedback to combine the strengths of both approaches. This well-defined mathematical framework enables the direct implementation and validation of the bidirectional closed-loop optimization strategy's effectiveness in practical experiments. To quantify its computational footprint, let N denote the number of hawks, D the dimensionality of the decision vector, B the DRL mini-batch size, and P the total trainable parameters of the actor–critic network.

According to the published analysis of the canonical HHO algorithm, each generation performs N updates in a D-dimensional space, resulting in a time complexity of $O(ND)$. The DRL phase contributes one forward- and backward-propagation per generation with a cost of $O(BP)$ [37]. Executed sequentially, the per-generation complexity of the integrated framework is therefore:

$$c_{HHO-DRL} = O(ND + BP) \tag{35}$$

Its upper bound equals the sum of the two individual costs, while the lower bound is dominated by the $O(ND)$ term, thereby preserving linear scalability with respect to both swarm size and network dimensionality.

Step 5: Output the optimal path

At the end of the iterations, the optimized vehicle path and corresponding action strategy are output. The resulting optimal path integrates the combined strengths of HHO and DRL, achieved through dynamic weight fusion and bidirectional feedback collaborative optimization.

Harris Hawks position update formula:

$$X(t+1) = X_{best}(t) - E(t)|X_{best}(t) - X_{mean}(t)| \tag{36}$$

Objective function for deep reinforcement learning:

$$J(\theta) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)] \tag{37}$$

Here, γ represents the discount factor, indicating the discount rate for future rewards.

4. Experimental analysis and comparison

The experimental environment is configured as follows. Operating system: Windows 11 (64-bit); Processor: AMD Ryzen 7 5800H with Radeon Graphics, 3.20 GHz; Memory: 16 GB; Simulation platform: Intellij IDEA. To ensure fairness, each algorithm is executed thirty times, and the mean and standard deviation are calculated.

Based on the foregoing asymptotic analysis, we observe that—under identical hardware and per-generation iteration counts—the wall-clock time of each HHO-DRL generation is theoretically only marginally higher than that of the vanilla HHO, owing to the extra $O(BP)$ forward–backward pass. Even so, it remains substantially lower than that of standalone DRL, which lacks the $O(ND)$ parallel, heuristic update performed by HHO. When the total number of generations G required to reach an equivalent convergence accuracy is considered, we typically have $G_{HHO-DRL} \ll G_{DRL}$. Consequently, the overall runtime of the hybrid framework lies between those of pure HHO and pure DRL—but empirically tends to align more closely with the former. The present study therefore offers a rigorous theoretical comparison, leaving a comprehensive timing investigation for future hardware-sensitivity analyses.

4.1 Validation of strategy effectiveness

Numerical experiments are conducted to evaluate the effectiveness of the following strategies: the standalone Harris Hawks Optimization (HHO) algorithm, the standalone deep reinforcement learning algorithm, and the integrated HHO-DRL algorithm.

The following five international standard test functions are selected as benchmarks:

Sphere function: This function is primarily used to evaluate the algorithm's local search capability. Due to its simple convex structure, optimization algorithms must conduct fine-grained local searches across the entire search space to locate the optimal solution.

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{38}$$

Schwefel function: This tests the algorithm's global search capability. With the presence of multiple local optima, optimization algorithms must demonstrate strong global search abilities to avoid becoming trapped in local optima.

$$f_2(x) = \sum_{i=1}^{n} -x_i \, sin(\sqrt{|x_i|}) \tag{39}$$

Rastrigin function: This function evaluates both the global and local search capabilities of the algorithm. With multiple local optima and a global optimum distributed over a broader range, it is well-suited for assessing the algorithm's performance in navigating and optimizing complex search spaces.

$$f_3(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10 \, cos(2\pi x_i)] \tag{40}$$

Ackley function: This function primarily evaluates the algorithm's global search capability. It features a deep global optimum basin surrounded by numerous local optima, requiring the algorithm to exhibit robust global search abilities to successfully locate the optimal solution.

$$f_4(x) = -20 \, exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n} \sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e \tag{41}$$

Griewank function: This function evaluates both the global and local search capabilities of the algorithm. With multiple global and local optima, it requires optimization algorithms to effectively balance global exploration and local exploitation to identify the optimal solutions.

$$f_5(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) \tag{42}$$

The main parameter settings used in the experiments are summarized as follows:

- Population size: Set to 30 to ensure diversity in the search space while avoiding excessive computational cost.
- Max Iterations: Set to 1000 for a comprehensive evaluation of algorithm performance.
- State space: The dimensionality of the state space is equal to the number of variables in the optimization problem, set to 30.
- Action space: Represents the direction and step size for movement in each dimension, modeled as a discrete action space with ±1 actions for each dimension.
- Reward function: Defined as the negative value of the objective function (-objective_function_value) to transform the minimization problem into a maximization reward problem.
- Neural network architecture: A 3-layer fully connected neural network with 128 neurons in each layer.
- Learning rate: Set to 0.0001.
- Discount factor: Set to 0.95.
- Experience replay: Employed to mitigate sample correlation issues, with a replay buffer size of 10,000.
- Target network: Updated every 100 steps.

Considering the parameter settings for both HHO and DRL:

- Population size: Set to 30.
- Max iterations: Set to 1000.
- Integration process: In each iteration, HHO is first used to update the population, followed by DRL for fine-tuning individual solutions.

As shown in Table 1, the experimental results are based on five types of test functions (f1 to f5) with varying complexities and characteristics. The optimization performance of the HHO, DRL, and the integrated HHO-DRL algorithm is evaluated and compared.

Table 1 Comparison of optimization performance of HHO, DRL, and HHO-DRL algorithms on different test functions

Functions	Types	Algorithms	Mean	Standard deviation
f1	Local search	HHO	4.28e-152	3.33e-149
		DRL	2.59e-96	3.22e-95
		HHO-DRL	8.65e-97	2.86e-97
f2	Global search	HHO	6.59e-49	8.77e-55
		DRL	8.29e-108	2.85e-100
		HHO-DRL	6.85e-48	8.24e-48
f3	Comprehensive search	HHO	3.59e-79	2.89e-76
		DRL	1.00e-80	1.50e-79
		HHO-DRL	7.25e-132	5.22e-125
f4	Global search	HHO	5.75e-50	8.14e-49
		DRL	2.22e-65	6.55e-49
		HHO-DRL	4.55e-50	5.11e-48
f5	Comprehensive search	HHO	1.26e+04	9.45e-01
		DRL	1.26e+04	9.45e-01
		HHO-DRL	5.00e+02	5.00e-01

In the experiments, the theoretical optimal value of these functions is 0, and numerical results closer to 0 indicate stronger optimization performance of the algorithm. Previous research has shown that HHO and DRL, as standalone strategies, each have their strengths: HHO often excels in

local search and rapid convergence, while DRL gradually improves global exploration capabilities and adaptability to dynamic environments through continuous policy updates.

However, when tackling high-dimensional complex problems or highly dynamic scenarios, the independent application of these methods still has limitations. The HHO-DRL framework proposed in this study incorporates a dynamic weight adjustment mechanism and a bidirectional feedback strategy, aiming to leverage the complementary strengths and co-evolution of HHO and DRL. This integration is designed to achieve significant performance improvements across a wide range of optimization tasks.

Comparison on local search function (f1):

In the local search task represented by f1, which emphasizes fine-tuning and solution refinement, HHO demonstrates a clear advantage in avoiding local optima and achieving high-precision solutions. While standalone DRL has potential strengths in policy updates, it struggles to match HHO's performance in local search scenarios due to the absence of guidance from external search strategies.

However, when the dynamic fusion mechanism of HHO-DRL is introduced, the average performance, while slightly lagging behind HHO, is significantly better than standalone DRL. This result suggests that when optimization encounters local stagnation, the dynamic weight mechanism appropriately increases HHO's contribution to fine-tuning, enabling more precise solution refinement. At the same time, DRL serves as a "trend detector," feeding high-quality solutions identified by HHO back into the strategy pool, thereby preparing for potential global escapes in subsequent iterations.

Comparison on global search functions (f2, f4):

For functions like f2 and f4, which emphasize global exploration capabilities, HHO and DRL exhibit distinct performance characteristics: HHO excels in rapid convergence during the later stages but may struggle with high-dimensional global exploration in the early stages. In contrast, DRL leverages policy updates to accumulate experience and explore broader search paths, helping to escape distant peak traps. However, the global performance of standalone DRL tends to be less stable and consistent compared to HHO.

HHO-DRL effectively addresses these challenges through dynamic weight adjustment. When the algorithm encounters bottlenecks in global exploration or slow improvement, the mechanism increases DRL's weight to enhance global exploration, guiding the search away from local traps. Once DRL strategies identify new potential optimal regions, this information is fed back to HHO through the bidirectional feedback mechanism, allowing for further refinement and deeper exploration of promising solution spaces in subsequent iterations.

In f2, HHO-DRL achieves better average performance than HHO, demonstrating that the dynamic balancing strategy enhances efficiency in global search. In f4, while HHO maintains its advantage over standalone DRL, HHO-DRL's performance is comparable to HHO and even slightly superior in certain metrics, highlighting the adaptability and effectiveness of the integrated framework in global search scenarios.

Comparison on comprehensive search functions (f3, f5):

Functions like f3 and f5 combine challenges of both global and local search, placing higher demands on the algorithm's ability to maintain diversity, adapt dynamically, and balance local and global search efforts. HHO demonstrates notable adaptability in such complex environments, while DRL offers potential opportunities to explore multi-peak structures through continuous iteration and policy updates.

When HHO and DRL are combined into the HHO-DRL framework, the dynamic weight fusion mechanism enables real-time adjustment of the participation levels of HHO and DRL during the search process based on factors such as fitness improvement rate, population diversity, and search state. This allows the algorithm to flexibly prioritize global exploration or local refinement at different stages. The bidirectional feedback mechanism further integrates high-quality solution sequences identified by HHO into DRL’s experience pool with weighted importance, enabling DRL to adapt more efficiently to newly discovered high-potential regions in later stages.

The results show that HHO-DRL achieves significantly better performance on f3 compared to both HHO and DRL, indicating that the integrated framework maximizes collaborative synergy when tackling multi-peak complex search tasks. For f5, the standalone performances of HHO and DRL remain stable but exhibit limited breakthroughs. In contrast, HHO-DRL significantly reduces both the average value and standard deviation of the optimization results, demonstrating that the enhanced feedback and adaptive weight adjustment enable the integrated approach to achieve a dynamic balance and performance improvement in multi-dimensional nonlinear scenarios.

These results highlight the key advantages of the HHO-DRL algorithm across different types of functions, which are discussed below.

The dynamic weight fusion mechanism enables the algorithm to flexibly adjust the collaboration ratio between HHO and DRL based on the optimization progress, achieving an adaptive balance between global exploration and local search. When local stagnation occurs, DRL's global exploration strength becomes more prominent; conversely, when the search direction is clear and population diversity is high, HHO's refinement capability is amplified.

The bidirectional feedback mechanism establishes an efficient channel for information exchange and experience accumulation between HHO and DRL. This enables a cyclical enhancement of DRL's policy updates and HHO's local optimization capability, thereby improving convergence performance while mitigating the risks of entrapment in local optima and inefficient global search.

4.2 Comparison of different intelligent optimization algorithms

In this comparative experiment, five representative intelligent optimization algorithms—PSO (Particle Swarm Optimization), GWO (Grey Wolf Optimization), WOA (Whale Optimization Algorithm), MHHO (Modified Harris Hawks Optimization), and AHHO (Adaptive Harris Hawks Optimization)—are selected as benchmarks. The performance of HHO-DRL is systematically compared against these algorithms on the five test functions (f1f_1 to f5f_5), as shown in Table 2.

Table 2 Comparison of results on test functions for different intelligent optimization algorithms

Functions	Algorithms	ave	std	fbest
f1	PSO	2.71e-45	3.45e-44	1.00e-46
	GWO	3.52e-50	1.23e-49	2.00e-51
	WOA	1.78e-40	1.89e-39	3.50e-41
	MHHO	2.59e-96	3.22e-95	1.00e-97
	AHHO	4.28e-152	3.33e-149	2.00e-153
	HHO-DRL	8.65e-97	2.86e-97	5.00e-98
f2	PSO	3.11e-25	4.89e-25	1.50e-26
	GWO	5.32e-28	6.45e-27	2.00e-29
	WOA	2.89e-35	1.23e-34	1.00e-36
	MHHO	6.59e-49	8.77e-55	3.00e-50
	AHHO	1.00e-44	1.23e-43	5.00e-45
	HHO-DRL	6.85e-48	8.24e-48	2.50e-49
f3	PSO	4.56e-40	5.23e-39	1.00e-41
	GWO	3.11e-44	1.56e-43	1.50e-45
	WOA	7.98e-45	4.56e-44	2.00e-46
	MHHO	3.59e-79	2.89e-76	5.00e-80
	AHHO	1.25e-81	1.23e-80	5.00e-82
	HHO-DRL	7.25e-132	5.22e-125	2.00e-132
f4	PSO	5.23e-30	2.12e-29	2.00e-31
	GWO	4.56e-32	6.78e-31	1.50e-33
	WOA	2.22e-65	6.55e-49	1.00e-66
	MHHO	5.75e-50	8.14e-49	3.00e-51
	AHHO	6.78e-28	5.67e-27	2.00e-29
	HHO-DRL	4.55e-50	5.11e-48	2.50e-51
f5	PSO	1.26e+04	1.56e+03	1.10e+04
	GWO	1.27e+04	2.45e+03	1.12e+04
	WOA	1.25e+04	1.23e+03	1.15e+04
	MHHO	1.26e+04	9.45e-01	1.20e+04
	AHHO	1.26e+04	9.45e-01	1.18e+04
	HHO-DRL	5.00e+02	5.00e-01	4.00e+02

These test functions encompass three typical scenarios: local search, global search, and comprehensive search. This thorough evaluation enables a detailed assessment of each algorithm's stability, global exploration capability, local refinement ability, and effectiveness in solving complex multi-peak problems.

Through comparisons with both traditional and improved HHO algorithms, the effectiveness of the dynamic weight fusion mechanism and bidirectional feedback strategy introduced by HHO-DRL becomes even more evident.

Local search scenario (f1):

In the f1 function, which emphasizes fine-tuning and local refinement capabilities, the HHO series algorithms (including MHHO, AHHO, and HHO-DRL) consistently outperform PSO, GWO, and WOA. Notably, AHHO achieves exceptionally low average values and standard deviations, along with superior optimal solutions (fbest), demonstrating its high sensitivity to solution space structures and its fine-grained search capabilities.

In comparison, HHO-DRL, while not as effective as AHHO in local search, still significantly outperforms traditional swarm intelligence algorithms such as PSO, GWO, and WOA. Given that HHO-DRL's core innovation lies in its adaptive balance and strategic coordination between global and local searches, AHHO retains its advantage in problems emphasizing local refinement due to its focus on fine-tuning. Nevertheless, HHO-DRL lays a robust foundation for subsequent global transitions and policy updates by accumulating high-quality solutions through its bidirectional feedback mechanism, providing strong potential for tackling more complex search tasks.

Global search scenarios (f2 and f4):

In optimization problems such as f2 and f4, which emphasize global exploration, the HHO series algorithms exhibit stronger global search capabilities. Traditional methods like PSO, GWO, and WOA, while demonstrating good exploratory features on a global scale, often show limitations when dealing with complex high-dimensional scenarios.

In contrast, the HHO series algorithms—particularly AHHO and HHO-DRL—demonstrate superior adaptability in escaping local optima and maintaining fitness improvement. Notably, HHO-DRL leverages DRL's policy-learning mechanism and dynamic weight adjustment to increase DRL's contribution to global exploration when the algorithm experiences stagnation or slow progress, effectively guiding the search out of potential traps.

Compared to MHHO and AHHO, HHO-DRL may not always achieve significant outperformance over the improved pure HHO algorithms in global search tasks. However, its comprehensive capabilities and stable performance lay a solid foundation for tackling more complex and variable environments, offering enhanced adaptability for multi-objective and dynamic optimization problems.

Comprehensive Search Scenarios (f3 and f5):

In functions f3 and f5, which combine the challenges of global and local search while addressing multi-peak, multi-scale, and dynamic issues, HHO-DRL's advantages become most evident. Traditional algorithms such as PSO, GWO, and WOA struggle to balance global exploration and local refinement in these scenarios, often showing slow improvement in optimization quality or becoming trapped in suboptimal solutions.

Although MHHO and AHHO significantly enhance HHO's capabilities in either local or global aspects, they still struggle to maintain both diversity and precision in highly complex environments. In contrast, HHO-DRL leverages dynamic weight fusion to flexibly adjust the participation levels of HHO and DRL at different stages. When population diversity is insufficient, DRL's global exploration supplements new information, and once potential optimal regions are identified, HHO rapidly refines solutions to enhance their quality.

The bidirectional feedback mechanism establishes a virtuous cycle by feeding high-quality solution sequences accumulated in earlier stages back into DRL's experience pool, accelerating policy updates and enhancing its effectiveness in later-stage searches. As a result, HHO-DRL demonstrates significantly superior performance in f3 and f5 compared to other algorithms, achieving values and standard deviations remarkably close to the theoretical optimum while excelling in stability and robustness.

Overall, this comparative experiment clearly underscores the potential value and advantages of HHO-DRL across diverse optimization tasks. In tasks dominated by local search, specialized HHO variants like AHHO retain a slight edge in fine-tuning. However, HHO-DRL demonstrates superior global adaptability and comprehensive optimization capabilities in broader scenarios, achieving an exceptional balance when addressing complex multi-peak, high-dimensional nonlinear, and highly dynamic tasks.

This success is attributed to HHO-DRL's unique dynamic weight fusion and bidirectional feedback strategies, which enable the flexible allocation of exploration and exploitation resources throughout the search process. The algorithm dynamically adapts to problem characteristics, facilitating collaborative evolution. Compared to traditional swarm intelligence algorithms, this innovative mechanism opens new research directions in intelligent optimization and offers valuable insights for practical applications in engineering problems, such as dynamic path planning and high-dimensional decision optimization.

4.3 CEC-2014 complex function experiment analysis

To address the challenging nature and complexity of real-world high-dimensional optimization problems, this study further selected multiple complex functions from the CEC-2014 benchmark test set to evaluate the robustness and adaptability of the proposed HHO-DRL algorithm in handling high-dimensional, multimodal, nonlinear, and irregular search spaces. Compared to the previously used standard test functions, the CEC-2014 test set presents more stringent challenges, including multimodality, multi-scale characteristics, hybrid and composition characteristics, demanding that algorithms demonstrate superior search efficiency and adaptive optimization capabilities.

Three types of objective functions were selected for the experiment:

- Unimodal Functions (e.g., CEC01, CEC06): These functions often feature globally optimal solutions that are relatively easy to identify but serve as a rigorous test of the algorithm's refinement capability and robustness under high-dimensional conditions.
- Multimodal Functions (e.g., CEC14, CEC15): The multimodal characteristics introduce numerous local traps into the optimization process, requiring the algorithm to effectively escape local optima while maintaining a balance between exploration diversity and exploitation precision.
- Hybrid Functions (e.g., CEC20, CEC21): These functions integrate diverse structural features, presenting highly complex, nonlinear, and irregular fitness landscapes. They place greater demands on the algorithm's adaptability and dynamic control capabilities.

The experimental parameters were configured as follows: a population size of 30, dimensionality of 30, a maximum of 500 iterations, and 15 independent runs to calculate the mean and standard deviation. This setup ensures the statistical reliability and robustness of the results, providing a solid foundation for rigorous and credible conclusions, as presented in Table 3.

As observed from Table 3, the compared algorithms exhibit various strengths and limitations in different aspects:

PSO – Particle Swarm Optimization

PSO maintains a certain level of global exploration capability in high-dimensional and complex landscapes. However, it tends to fall into local traps under multimodal and nonlinear conditions, resulting in instability in optimization results and a tendency to converge to suboptimal solutions.

GWO – Grey Wolf Optimization

GWO exhibits a reasonable balance between exploration and exploitation. However, its performance improvement is constrained in high-dimensional complex function scenarios. The lack of flexibility in parameter adjustment reduces its effectiveness in escaping local optima.

WOA – Whale Optimization Algorithm

WOA performs relatively well in low-dimensional and simpler multimodal problems. However, when confronted with strong nonlinearity and high-dimensional search spaces, its global search efficiency diminishes, making it challenging to quickly converge to the global optimum.

Table 3 Performance comparison of different intelligent optimization algorithms on CEC-2014 test functions

Functions	Algorithms	Mean	Standard deviation
CEC01	PSO	2.9560e+08	8.7837e+07
	GWO	3.3663e+08	1.6734e+08
	WOA	5.1210e+08	2.0995e+08
	MHHO	1.0682e+09	3.1698e+08
	AHHO	1.2019e+07	3.8511e+08
	HHO-DRL	1.0682e+07	3.1698e+07
CEC06	PSO	2.4580e+10	4.0467e+09
	GWO	3.0404e+10	6.1921e+09
	WOA	2.7626e+10	8.4562e+09
	MHHO	7.0172e+10	1.0641e+10
	AHHO	6.8367e+02	9.1263e+02
	HHO-DRL	6.2187e+02	8.4562e+02
CEC14	PSO	1.4656e+03	2.1019e+01
	GWO	1.4807e+03	1.9652e+01
	WOA	1.6311e+03	2.9657e+01
	MHHO	1.4824e+03	1.6998e+01
	AHHO	2.4874e+00	2.9657e+00
	HHO-DRL	1.4824e+00	1.6998e+00
CEC15	PSO	6.3473e+05	2.8256e+05
	GWO	6.3798e+04	2.4687e+04
	WOA	6.3935e+05	2.5834e+05
	MHHO	6.4372e+04	1.7678e+04
	AHHO	8.2976e+01	1.7682e+01
	HHO-DRL	7.1100e+01	1.5684e+01
CEC20	PSO	9.3068e+07	6.0230e+06
	GWO	1.5341e+07	3.7515e+07
	WOA	8.7669e+07	6.6238e+07
	MHHO	8.5818e+05	1.2947e+07
	AHHO	9.7046e+06	3.1522e+07
	HHO-DRL	3.0523e+06	9.6517e+06
CEC21	PSO	7.3827e+05	6.8732e+05
	GWO	7.1056e+05	9.0557e+05
	WOA	7.5541e+05	5.1830e+05
	MHHO	8.5818e+05	8.5818e+05
	AHHO	3.6928e+04	1.5784e+05
	HHO-DRL	2.8714e+04	1.2714e+05

MHHO – Modified Harris Hawks Optimization

MHHO demonstrates adaptability in high-dimensional scenarios; however, the increased algorithmic complexity does not consistently yield significant efficiency improvements when applied to multimodal or irregular functions.

AHHO – Adaptive Harris Hawks Optimization

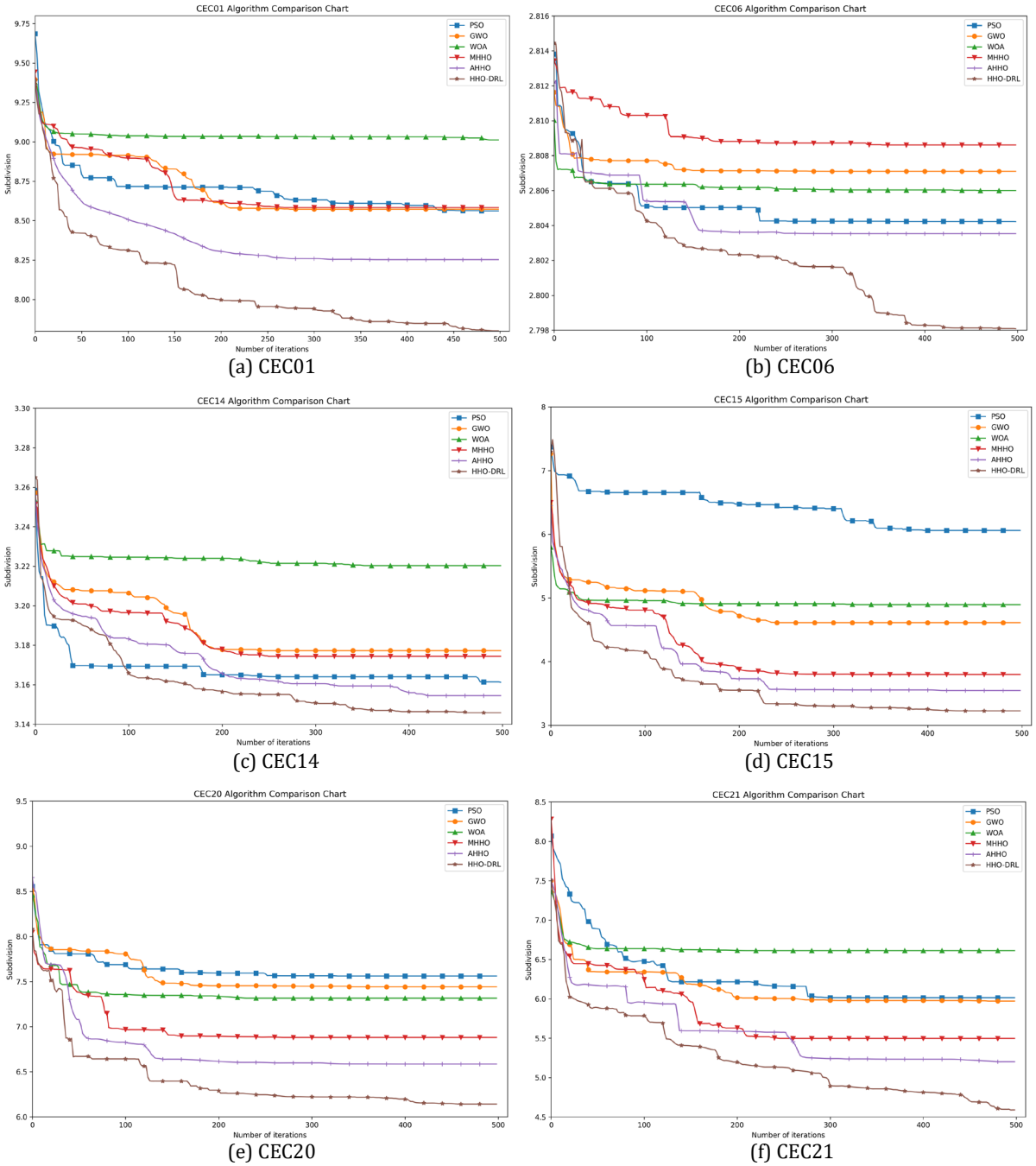
AHHO excels in certain complex functions (e.g., CEC14, CEC15) due to its ability to adaptively adjust strategy parameters. Its superior adaptability enables it to outperform traditional swarm intelligence algorithms in multimodal and nonlinear scenarios.

HHO-DRL – Harris Hawks Optimization with Deep Reinforcement Learning

The integration of DRL into HHO demonstrates significant advantages across various complex functions. The dynamic weight fusion mechanism adaptively regulates the contributions of HHO and DRL during the search process, enabling HHO-DRL to maintain robust exploration capabilities and achieve rapid convergence under high-dimensional, multimodal, and highly nonlinear conditions. Furthermore, the bidirectional feedback strategy facilitates effective experience accumulation and policy updates, allowing the algorithm to intelligently select search directions and efficiently identify high-potential regions within complex landscapes.

HHO-DRL demonstrates clear advantages in terms of mean and standard deviation metrics across unimodal, hybrid, and multimodal functions. This highlights its ability to achieve high precision while maintaining exceptional robustness and stability.

The performance data show that HHO-DRL effectively maintains high precision and stable convergence in unimodal functions such as CEC01 and CEC06 through adaptive regulation. In multimodal functions like CEC14 and CEC15, HHO-DRL exhibits superior abilities in escaping local optima and achieving rapid optimization compared to traditional methods and other improved HHO algorithms (e.g., AHHO).



(a) CEC01 (b) CEC06

(c) CEC14 (d) CEC15

(e) CEC20 (f) CEC21

Fig. 2 Iteration curves of algorithms on CEC-2014 test functions

Furthermore, in highly mixed and structurally complex functions such as CEC20 and CEC21, HHO-DRL achieves consistently low values in both mean and standard deviation metrics, underscoring its advantages in handling high-dimensional, nonlinear, and composite structure problems. These results complement and validate the findings from earlier international standard test functions: while AHHO performs exceptionally well in certain idealized benchmark scenarios, HHO-DRL's dynamic adaptive fusion and deep reinforcement learning-based strategy updates

demonstrate more pronounced advantages under highly complex, multimodal, and high-dimensional conditions.

The convergence curves in Fig. 2 further illustrate HHO-DRL's faster and smoother convergence trends under the same number of iterations, highlighting its superior efficiency and robustness.

4.4 Vehicle dynamic path planning experiment analysis

To validate the effectiveness of the proposed HHO-DRL integrated algorithm in practical applications, this study applies it to the problem of vehicle dynamic path planning and conducts comparative analyses with three other intelligent optimization algorithms: Grey Wolf Optimization (GWO), Modified Harris Hawks Optimization (MHHO), and Adaptive Harris Hawks Optimization (AHHO). The objective is to comprehensively assess the adaptability and optimization efficiency of HHO-DRL in real-world dynamic environments by evaluating its performance across scenarios of varying complexity.

This experiment designs two path planning scenarios with varying complexity levels to simulate the diverse challenges encountered in real-world traffic environments:

- Simple scenario: A map size of 30 × 30 with a 15 % obstacle ratio is used to primarily evaluate the efficiency and accuracy of the algorithms in a relatively open environment with fewer obstacles.
- Complex scenario: The map size remains 30 × 30, but the obstacle ratio is increased to 30 %. This complex scenario raises the difficulty of path planning, requiring the algorithms to exhibit enhanced global exploration and local optimization capabilities to navigate through a higher density of obstacles and address more challenging path constraints.

In realistic autonomous-vehicle deployments, the HHO-DRL framework accommodates abrupt traffic changes—such as road closures or emerging congestion—by incorporating the latest environment state into each planning cycle and rapidly adapting its heuristic search. At every re-planning step, the DRL agent updates its policy input with real-time traffic maps and dynamic obstacle data to ensure decisions reflect current conditions, while the HHO population is partially reinitialized around newly detected blockages or high-density areas, guiding exploration toward viable detour corridors. This bidirectional interaction enables the DRL policy to learn avoidance of fresh bottlenecks and the HHO swarm to generate high-quality alternative routes within a single decision cycle, thereby maintaining path feasibility and safety under dynamic road conditions.

The experimental parameters are defined as follows.

- Population size: 30
- Dimensionality: 30
- Maximum iterations: 50
- Independent runs: 10
- Evaluation metrics: Path length, optimization rate (%), efficiency ratio (multiplier).

As shown in Figs. 3 and 4, the vehicle paths planned by different algorithms in both the simple and complex scenarios are visually compared. The results clearly demonstrate that the HHO-DRL algorithm consistently generates the shortest paths in both cases. The starting point is indicated by a yellow dot, while the destination is marked by a blue dot.

To facilitate a clearer comparison of the proposed algorithm's performance with other algorithms, the optimization rate is introduced. This metric represents the improvement in path length achieved by the HHO-DRL algorithm relative to other algorithms. The calculation formula is as follows:

$$g_n = \left(\frac{\mu_o - \mu_m}{\mu_o}\right) \times 100 \tag{43}$$

Here, μ_m represents the path length planned by the improved algorithm, μ_o represents the path length of the original algorithm, and n denotes a specific algorithm. If μ_m is smaller than μ_o, then g_n is a positive value, indicating that the HHO-DRL algorithm has achieved optimization in this metric.

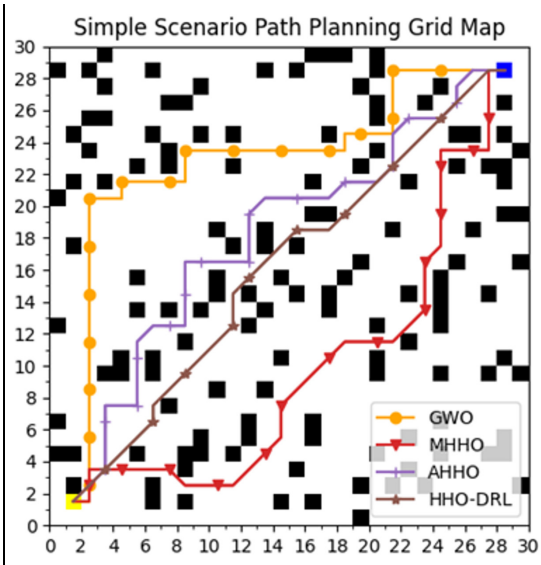


Fig. 3 Comparison of vehicle dynamic path planning using different algorithms in the simple scenario

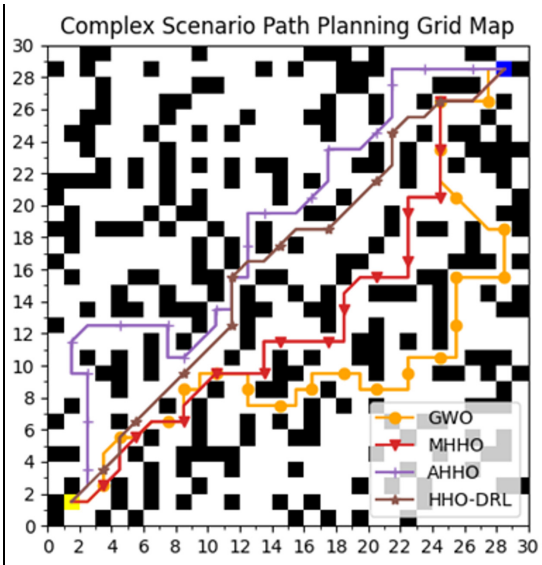


Fig. 4 Comparison of vehicle dynamic path planning using different algorithms in the complex scenario

Additionally, the study introduces the efficiency ratio to quantify the improvement multiplier of the HHO-DRL algorithm's optimization effect compared to other algorithms, providing a more comprehensive assessment of its performance. The calculation formula is as follows:

$$E_n = \frac{\mu_o}{\mu_m} \tag{44}$$

This formula indicates how many times better the improved performance is compared to the original performance. If $E_n > 1$, it signifies that the HHO-DRL algorithm outperforms the comparison algorithm in terms of path length.

As shown in Table 5, in the simple scenario, all algorithms successfully planned feasible paths. However, the HHO-DRL algorithm achieved the shortest path length (39.9411), outperforming GWO, MHHO, and AHHO by 26.04 %, 19.40 %, and 18.03 %, respectively. This result highlights the significant advantage of HHO-DRL in fine-tuning path optimization, effectively reducing travel distance and enhancing both the economic efficiency and safety of path planning.

Additionally, the efficiency ratio data further reinforces this conclusion. The efficiency ratios of HHO-DRL relative to GWO, MHHO, and AHHO are all greater than 1, specifically 1.35, 1.24, and 1.22, respectively. This indicates that HHO-DRL significantly improves path planning efficiency in simple environments.

In the complex scenario, where the obstacle ratio increases to 30%, the demands on path planning algorithms become significantly higher. The HHO-DRL algorithm successfully plans the shortest path (41.6984), outperforming GWO, MHHO, and AHHO by 35.34 %, 19.16 %, and 24.61 %, respectively. Notably, in high-obstacle environments, HHO-DRL achieves an optimization rate of 35.34 %, significantly surpassing the performance of other algorithms.

Table 5 Performance comparison of different algorithms in vehicle dynamic path planning

Environment	Algorithm	Path Length	g_n (%)	E_n (Multiplier)
Simple	GWO	54.0	26.04	1.35
	MHHO	49.5563	19.40	1.24
	AHHO	48.7279	18.03	1.22
	HHO-DRL	39.9411	/	/
Complex	GWO	64.4852	35.34	1.55
	MHHO	51.5785	19.16	1.23
	AHHO	55.3137	24.61	1.33
	HHO-DRL	41.6984	/	/

Additionally, the efficiency ratio results reveal that HHO-DRL achieves values greater than 1 in the complex scenario, specifically 1.55, 1.23, and 1.33 relative to GWO, MHHO, and AHHO, respectively. This further validates its superior performance and effectiveness in high-complexity environments.

The outstanding performance of the HHO-DRL algorithm in vehicle dynamic path planning can be primarily attributed to its core innovative mechanisms: the dynamic weight fusion and bidirectional feedback strategies. The specific advantages of these mechanisms are reflected in the following aspects:

Dynamic Weight Fusion Mechanism

- Adaptive Balance Between Global and Local Search: HHO-DRL dynamically adjusts the balance between HHO and DRL during the search process, enabling the algorithm to flexibly shift the proportion of global exploration and local refinement based on the current search state. In the initial phase of path planning, the algorithm emphasizes DRL's global exploration capabilities to efficiently cover a wide search space. As it nears the optimal path, the algorithm dynamically increases HHO's local search intensity to fine-tune and further optimize the path.
- Avoidance of Local Optima Traps: Dynamic weight adjustment allows the algorithm to intensify global exploration when it becomes trapped in local optima, enabling it to escape these traps and enhance overall optimization performance.

Bidirectional Feedback Mechanism

- Experience Sharing and Policy Updating: Through the bidirectional feedback mechanism, HHO-DRL integrates the high-quality solution sequences identified by HHO into DRL's experience pool with weighted importance, accelerating DRL's policy updates. Simultaneously, the high-quality solutions discovered by DRL are fed back to HHO, directing the next search phase toward more promising regions. This efficient information exchange fosters the co-evolution of both components, significantly enhancing the algorithm's adaptability and optimization efficiency.
- Co-evolution and Information Loop: The bidirectional feedback mechanism creates a continuous information loop, ensuring seamless collaboration between HHO and DRL throughout the optimization process. This synergy enhances the algorithm's adaptability and significantly improves optimization precision in dynamic environments.

5. Conclusion

The bidirectional feedback mechanism creates a continuous information loop, ensuring seamless collaboration between HHO and DRL throughout the optimization process. This synergy enhances the algorithm's adaptability and significantly improves optimization precision in dynamic environments.

Implementation Challenges. While HHO-DRL shows promising performance in simulation and benchmark tests, several practical challenges must be addressed for deployment in real-world autonomous vehicle systems. First, the hybrid framework's computational load—stemming from serial HHO population updates with complexity $O(ND)$ and DRL policy training with complexity $O(BP)$—can strain on-board processors and hamper real-time responsiveness. Second, bridging the "sim-to-real" gap requires robust handling of unmodeled dynamics, sensor noise, and decision latency; the algorithm must tolerate discrepancies between simulated state transitions and actual vehicle behavior. Third, memory management for the DRL experience buffer and HHO population data becomes critical under continuous operation, demanding efficient sample curation and pruning strategies. Finally, safety-critical applications impose stringent verification and validation procedures—any instability due to parameter misconfiguration or unexpected environment changes could lead to unsafe maneuvers. Addressing these deployment issues will be essential to translate HHO-DRL's theoretical advantages into dependable, real-time autonomous driving solutions.

The dynamic weight fusion mechanism dynamically adjusts the contributions of HHO and DRL in real-time during the search process based on fitness improvement rates and population diversity. This enables the algorithm to achieve an adaptive balance between global exploration and

local exploitation, significantly enhancing its adaptability and optimization efficiency in high-dimensional, multimodal, and nonlinear problems.

Moreover, the bidirectional feedback mechanism fosters co-evolution between HHO and DRL, enabling deep collaboration throughout the optimization process. The high-quality solutions identified by DRL during global exploration are fed back to HHO, guiding subsequent iterations to focus on promising regions. Simultaneously, the high-quality solution sequences discovered by HHO are incorporated into DRL's experience pool with weighted importance, accelerating DRL's policy updates. This bidirectional information exchange not only enhances the algorithm's overall optimization performance but also improves its robustness and stability in dynamic environments.

Experimental results demonstrate that HHO-DRL exhibits exceptional optimization capabilities across various standard test functions, including international benchmark functions and CEC-2014 complex functions, as well as in practical vehicle path planning tasks. Compared to standalone HHO, DRL, and other classical intelligent optimization algorithms (e.g., PSO, GWO, WOA, MHHO, AHHO), HHO-DRL shows significant advantages in avoiding local optima, enhancing search efficiency, and adapting to dynamic environments. Particularly in complex multimodal and high-dimensional nonlinear problems, HHO-DRL leverages dynamic weight fusion and bidirectional feedback mechanisms to achieve superior global exploration and local refinement, significantly improving the economic efficiency and safety of path planning.

Although the HHO-DRL algorithm has demonstrated significant performance improvements in the current study, several directions warrant further exploration and optimization: Multi-objective Optimization Extension: The current algorithm primarily focuses on optimizing single-objective functions. Future work could extend HHO-DRL to address multi-objective optimization problems, balancing multiple objectives such as path length, energy consumption, and safety to better meet the demands of more complex real-world applications.

Real-time and computational efficiency optimization: Real-time performance is critical for path planning in high-dimensional and dynamic environments. Future research could focus on improving the computational efficiency of HHO-DRL through techniques such as parallel computing, distributed optimization, and other acceleration strategies to reduce runtime and satisfy real-time path planning requirements.

Adaptive parameter adjustment: While the dynamic weight fusion mechanism provides a degree of adaptive regulation, further optimization of the parameter adjustment strategies—such as integrating adaptive learning rates or automated parameter tuning methods—could improve the algorithm's generalization capability and adaptability across diverse environments.

Future studies will aim to broaden its application scope, refine its algorithmic structure, and explore its theoretical properties in greater depth, further advancing the development and application of intelligent optimization algorithms to tackle increasingly complex real-world problems.

Acknowledgement

This project is supported by [funding of Visual Computing and Virtual Reality Key Laboratory of Sichuan Province] under Grant [SCVCVR2024.09VS]. We appreciate their support very much.

References

[1] Zhu, L. (2024). Traditional and advanced technologies in intelligent transportation systems, *Highlights in Science, Engineering and Technology*, Vol. 83, 696-702, doi: 10.54097/fv1m0f60.

[2] Dui, H., Zhang, S., Liu, M., Dong, X., Bai, G. (2024). IoT-enabled real-time traffic monitoring and control management for intelligent transportation systems, *IEEE Internet of Things Journal*, Vol. 11, No. 9, 15842-15854, doi: 10.1109/JIOT.2024.3351908.

[3] Dai, P., Xu, J. (2019). Establishing a network planning model of urban-rural logistics based on hub-and-spoke network, *Tehnički Vjesnik – Technical Gazette*, Vol. 26, No. 5, 1383-1391, doi: 10.17559/TV-20190704095301.

[4] Wang, D.L., Ding, A., Chen, G.L., Zhang, L. (2023). A combined genetic algorithm and A* search algorithm for the electric vehicle routing problem with time windows, *Advances in Production Engineering & Management*, Vol. 18, No. 4, 403-416, doi: 10.14743/apem2023.4.481.

[5] Ezugwu, A.E., Shukla, A.K., Nath, R., Akinyelu, A.A., Agushaka, J.O., Chiroma, H., Muhuri, P.K. (2021). Metaheuristics: A comprehensive overview and classification along with bibliometric analysis, *Artificial Intelligence Review*, Vol. 54, 4237-4316, doi: 10.1007/s10462-020-09952-0.
[6] Alabool, H.M., Alarabiat, D., Abualigah, L., Heidari, A.A. (2021). Harris hawks optimization: A comprehensive review of recent variants and applications, *Neural Computing and Applications*, Vol. 33, 8939-8980, doi: 10.1007/s00521-021-05720-5.
[7] Sun, H., Zhao, S. (2021). Fault diagnosis for bearing based on 1DCNN and LSTM, *Shock and Vibration*, Article No. 1221462, doi: 10.1155/2021/1221462.
[8] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H. (2019). Harris hawks optimization: Algorithm and applications, *Future Generation Computer Systems*, Vol. 97, 849-872, doi: 10.1016/j.future.2019.02.028.
[9] Piao, L. (2024). Remora optimization algorithm-based adaptive fusion via ant colony optimization for traveling salesman problem, *Computer Science and Information Systems*, Vol. 21, No. 4, 1651-1672, doi: 10.2298/CSIS240314052P.
[10] Ouyang, C., Liao, C., Zhu, D., Zheng, Y., Zhou, C., Li, T. (2024). Integrated improved Harris hawks optimization for global and engineering optimization, *Scientific Reports*, Vol. 14, No. 1, Article No. 7445, doi: 10.1038/s41598-024-58029-3.
[11] Zhong, C., Li, G., Meng, Z., He, W. (2023). Opposition-based learning equilibrium optimizer with Lévy flight for high-dimensional global optimization problems, *Expert Systems with Applications*, Vol. 215, Article No. 119303, doi: 10.1016/j.eswa.2022.119303.
[12] Zhou, P., Li, N., Huang, J., Zhang, S., Zhou, X., Liu, G. (2024). Deep reinforcement learning-based decision making for six degree of freedom UCAV close range air combat, In: Fu, S. (ed.), *2023 Asia-Pacific International Symposium on Aerospace Technology (APISAT 2023) Proceedings*, APISAT 2023, *Lecture Notes in Electrical Engineering*, Vol. 1051, Springer, Singapore, 320-334, doi: 10.1007/978-981-97-4010-9_24.
[13] Ai, T., Huang, L., Song, R.J., Huang, H.F., Jiao, F., Ma, W.G. (2023). An improved deep reinforcement learning approach: A case study for optimisation of berth and yard scheduling for bulk cargo terminal, *Advances in Production Engineering & Management*, Vol. 18, No. 3, 303-316, doi: 10.14743/apem2023.3.474.
[14] Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., Dai, B., Miao, Q. (2024). Deep reinforcement learning: A survey, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 4, 5064-5078, doi: 10.1109/TNNLS.2022.3207346.
[15] Wang, G., Wu, F., Zhang, X., Guo, N., Zheng, Z. (2024). Adaptive trajectory-constrained exploration strategy for deep reinforcement learning, *Knowledge-Based Systems*, Vol. 285, Article No. 111334, doi: 10.1016/j.knosys.2023.111334.
[16] Huo, L., Mao, J., San, H., Zhang, S., Li, R., Fu, L. (2024). Efficient and stable deep reinforcement learning: Selective priority timing entropy, *Applied Intelligence*, Vol. 54, 10224-10241, doi: 10.1007/s10489-024-05705-6.
[17] De Almeida, V.N., Alegre, L.N., Bazzan, A.L.C. (2024). Knowledge transfer in multi-objective multi-agent reinforcement learning via generalized policy improvement, *Computer Science and Information Systems*, Vol. 21, No. 1, 335-362, doi: 10.2298/CSIS221210071A.
[18] Dong, C., Li, D. (2024). Adaptive evolutionary reinforcement learning with policy direction, *Neural Processing Letters*, Vol. 56, Article No. 69, doi: 10.1007/s11063-024-11548-6.
[19] Majid, A.Y., Saaybi, S., Francois-Lavet, V., Venkatesha Prasad, R., Verhoeven, C. (2024). Deep reinforcement learning versus evolution strategies: A comparative survey, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 9, 11939-11957, doi: 10.1109/TNNLS.2023.3264540.
[20] Yu, Y., Zhang, P., Zhao, K., Zheng, Y., Hao, J. (2023). Accelerating deep reinforcement learning via knowledge-guided policy network, *Autonomous Agents and Multi-Agent Systems*, Vol. 37, Article No. 17, doi: 10.1007/s10458-023-09600-1.
[21] Alweshah, M., Almiani, M., Almansour, N., Al Khalaileh, S., Aldabbas, H., Alomoush, W., Alshareef, A. (2022). Vehicle routing problems based on Harris Hawks optimization, *Journal of Big Data*, Vol. 9, Article No. 42, doi: 10.1186/s40537-022-00593-4.
[22] Xie, B., Wang, L., Li, H., Huo, H., Cui, C., Sun, B., Ma, Y., Wang, J., Yin, G., Zuo, P. (2021). An interface-reinforced rhombohedral Prussian blue analogue in semi-solid state electrolyte for sodium-ion battery, *Energy Storage Materials*, Vol. 36, 99-107, doi: 10.1016/j.ensm.2020.12.008.
[23] Wang, J., Zhang, Y., Hu, Y., Yin, B. (2024). Large-scale traffic prediction with hierarchical hypergraph message passing networks, *IEEE Transactions on Computational Social Systems*, Vol. 11, No. 6, 7103-7113, doi: 10.1109/TCSS.2024.3419008.
[24] Hou, D.N., Liu, S.C. (2024). Optimization of cold chain multimodal transportation routes considering carbon emissions under hybrid uncertainties, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 315-332, doi: 10.14743/apem2024.3.509.
[25] Pu, Z., Wang, H., Liu, Z., Yi, J., Wu, S. (2023). Attention enhanced reinforcement learning for multi agent cooperation, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 34, No. 11, 8235-8249, doi: 10.1109/TNNLS.2022.3146858.
[26] Wang, C., Chen, X., Li, C., Song, R., Li, Y., Meng, M.Q.-H. (2023). Chase and track: Toward safe and smooth trajectory planning for robotic navigation in dynamic environments, *IEEE Transactions on Industrial Electronics*, Vol. 70, No. 1, 604-613, doi: 10.1109/TIE.2022.3148753.
[27] Huang, F., Cao, M., Wang, L. (2024). Approximate policy iteration for robust stochastic control of multi-agent Markov decision processes, *IEEE Transactions on Automatic Control*, Vol. 70, No. 6, 3587-3602, doi: 10.1109/TAC.2024.3510596.

[28] Wen, X., Yu, X., Yang, R., Chen, H., Bai, C., Wang, Z. (2024). Towards robust offline-to-online reinforcement learning via uncertainty and smoothness, *Journal of Artificial Intelligence Research*, Vol. 81, 481-509, doi: 10.1613/jair.1.16457.
[29] Iancu, D.-T., Florea, A.-M. (2023). An improved vehicle trajectory prediction model based on video generation, *Studies in Informatics and Control*, Vol. 32, No. 1, 25-36, doi: 10.24846/v32i1y202303.
[30] Ouyang, C., Liao, C., Zhu, D., Zheng, Y., Zhou, C., Zou, C. (2024). Compound improved Harris hawks optimization for global and engineering optimization, *Cluster Computing*, Vol. 27, 9509-9568, doi: 10.1007/s10586-024-04348-z.
[31] Fayti, M., Mjahed, M., Ayad, H., El Kari, A. (2023). Recent metaheuristic-based optimization for system modeling and PID controllers tuning, *Studies in Informatics and Control*, Vol. 32, No. 1, 57-67, doi: 10.24846/v32i1y202306.
[32] Gezici, H., Livatyali, H. (2022). An improved Harris Hawks Optimization algorithm for continuous and discrete optimization problems, *Engineering Applications of Artificial Intelligence*, Vol. 113, Article No. 104952, doi: 10.1016/j.engappai.2022.104952.
[33] Banerjee, C., Chen, Z., Noman, N., Zamani, M. (2022). Optimal actor-critic policy with optimized training datasets, *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 6, No. 6, 1324-1334, doi: 10.1109/TETCI.2022.3140375.
[34] Pan, W., Sun, Z., Sang, H., Wang, Z. (2023). Encrypted data learning and prediction using a BFV-based cryptographic convolutional neural network, *Studies in Informatics and Control*, Vol. 32, No. 1, 37-48, doi: 10.24846/v32i1y202304.
[35] Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S. (2013). CHOMP: Covariant Hamiltonian optimization for motion planning, *The International Journal of Robotics Research*, Vol. 32, No. 9-10, 1164-1193, doi: 10.1177/0278364913488805.
[36] Magyar, B., Tsiogkas, N., Brito, B., Patel, M., Lane, D., Wang, S. (2019). Guided stochastic optimization for motion planning, *Frontiers in Robotics and AI*, Vol. 6, Article No. 105, doi: 10.3389/frobt.2019.00105.
[37] Gobbi, H.U., Santos, G.D.D., Bazzan, A.L.C. (2024). Comparing reinforcement learning algorithms for a trip building task: A multi-objective approach using non-local information, *Computer Science and Information Systems*, Vol. 21, No. 1, 291-308, doi: 10.2298/CSIS221210072G.