Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Global State-Guided Reinforcement Learning-Assisted Differential Evolution for Multi-Objective Optimization

Zijian Cao

bosscao@163.com

Xi'an Technological University

Wenzheng Liu

Xi'an Technological University

Chuhang Qiao

Xi'an Technological University

Haowen Jia

Xi'an Technological University

Yu Li

Xi'an Technological University

Haoshuai Zhang

Xi'an Technological University

Feng Tian

Kunshan Duke University

Method Article

Keywords: Multi-objective optimization, Differential evolution, Reinforcement learning, Global state, Unmanned aerial vehicle path planning

Posted Date: October 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11075153/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Global State-Guided Reinforcement Learning-Assisted Differential Evolution for Multi-Objective Optimization

Zijian Cao[1*], Wenzheng Liu[1], Chuhang Qiao[1], Haowen Jia[1], Yu Li[1], Haoshuai Zhang[1], Feng Tian[2]

[1]School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China

[2]Kunshan Duke University, Kunshan 215316, China

bosscao@163.com, 1728208845@qq.com, 1029301841@qq.com, 603176912@qq.com, 3076140775@qq.com, 1251909074@qq.com, feng.tian978@dukekunshan.edu.cn

Corresponding author*: bosscao@163.com

Abstract: Multi-objective differential evolution must adapt its search behaviour while balancing convergence and diversity. However, existing reinforcement learning-assisted variants often regulate evolutionary operators or control parameters in isolation and make limited use of population-level multi-objective information. We propose a global state-guided reinforcement learning-assisted differential evolution algorithm for multi-objective optimization (GSG-RLDE). Its compact global state integrates the proportion of solutions on the first non-dominated front, the coefficient of variation of their finite crowding distances and the intergenerational change in that proportion, thereby representing convergence, distribution and search trend. Under this shared state, each individual independently samples a joint action comprising a mutation strategy and control-parameter interval from a common Q-table. An action-conditioned successful-parameter memory further reuses effective control settings. GSG-RLDE was evaluated against five representative algorithms on 31 problems from the ZDT, DTLZ, UF and WFG suites and in two multi-objective unmanned aerial vehicle path-planning scenarios. It achieved the best mean Friedman rank for both hypervolume and inverted generational distance across the benchmark problems. In both path-planning scenarios, GSG-RLDE obtained the highest mean hypervolume and a lower hypervolume standard deviation than all comparison algorithms, while generating short, smooth and obstacle-avoiding routes. These results indicate that global-state feedback combined with individual action execution enables effective adaptive search across benchmark and constrained engineering problems.

Keywords: Multi-objective optimization; Differential evolution; Reinforcement learning; Global state; Unmanned aerial vehicle path planning

1 Introduction

Multi-objective optimization problems (MOPs; Hua et al. 2021) arise widely in data mining (Bhuyan et al. 2022), power engineering (Al-Kaabi et al. 2022; Ali et al. 2023) and industrial design (Polo-Mendoza et al. 2023; Anosri et al. 2023). Their objectives are typically conflicting, so the aim is not a single optimum but a well-converged and well-distributed set of Pareto-optimal solutions (Wang et al. 2021). A general MOP can be formulated as:

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_m(x)), \\ \text{s.t.} G(x) = (g_1(x), g_2(x), \dots, g_m(x)), \end{cases} \tag{1}$$

where x is the decision vector $(x_1, \dots, x_n)$, $F(x)$ is the objective vector, and $G(x)$ denotes the constraints.

Conflicting objectives make conventional single-objective algorithms poorly suited to representing the structure of a solution set or balancing convergence and diversity (Coello Coello et al. 2007). Multi-objective evolutionary algorithms (MOEAs; Zhang and Xing 2017; Ma et al. 2019; Falcon-Cardona and Coello Coello 2020) address this limitation by generating multiple candidate solutions through population-based search. Differential evolution (DE; Storn and Price 1997) is widely used in continuous and engineering optimization because of its simple structure, few parameters and strong global search capability. Extending DE to multi-objective optimization has therefore become an important route to improving MOEA performance.

Multi-objective variants of DE have been studied extensively. Abbass et al. (2002) proposed Pareto-frontier differential evolution (PDE), which generates new individuals using DE and retains non-dominated solutions for the next generation. Xue et al. (2003) developed multi-objective differential evolution (MODE), combining Pareto-based fitness ranking with crowding-distance reduction. Yao et al. (2010) incorporated non-dominated sorting and crowding distance into DE selection and reported better convergence and diversity than NSGA-II (Deb et al. 2002). Varadarajan and Swarup (2008) applied DE to optimal power flow with competing objectives, demonstrating its potential for constrained real-world MOPs. Beyond Pareto-dominance methods, MOEA/D-DE proposed by Li and Zhang (2006, 2009) decomposes continuous MOPs and generates trial solutions using DE/rand/1/bin. Qu and Suganthan (2010) instead combined normalized objective summation with diversified selection, avoiding non-dominated sorting. These studies establish the potential of DE for MOPs, but their search behaviour remains strongly dependent on predefined strategies and parameters.

Reinforcement learning (RL) enables state-feedback-based adaptation in evolutionary search. Existing studies have primarily addressed parameter control and strategy selection. Huynh et al. (2021) used Q-learning to adapt combinations of F and CR, whereas Sun et al. (2021) formulated parameter control as a Markov decision process for dynamic DE parameter adjustment. Liu et al. (2019) proposed QLPSO, which jointly models state information and the action dimension in a three-dimensional Q-table. This representation allows each particle to select a parameter configuration according to its own state. For multi-objective optimization, Liang et al. (2024) combined information entropy and RL to learn Pareto-front structure and adjust reference-vector distributions. Han et al. (2023) used RL to activate and adapt reference points, mitigating the exploration-exploitation conflict. Yu et al. (2024) combined Q-learning-driven mutation selection with Pareto-front relearning for multi-objective feature selection. RL-assisted methods thus provide state-responsive adaptation, but most regulate only one component and make limited joint use of dominance, solution distribution and evolutionary trends.

The central challenge is therefore to represent convergence, distribution quality and search trends in a compact state, then use this state to coordinate mutation strategies and control parameters. Without such joint feedback, search on complex Pareto fronts can stagnate, become locally crowded or lose exploratory capacity.

We address this challenge with GSG-RLDE. The algorithm constructs a global multi-objective state from non-dominated sorting and crowding distance, then uses a shared Q-table to coordinate strategy and parameter selection. Each individual samples and executes an action independently under the same state, allowing heterogeneous search without losing population-level guidance. An action-conditioned memory further reuses successful control-parameter information. The main contributions are as follows:

(1) We develop GSG-RLDE, which uses a shared Q-table to coordinate strategy and parameter selection. Each individual independently selects an action under the current global state, producing coordinated yet complementary search behaviours.

(2) We construct a compact global state from the first-front proportion, the variation in finite crowding distances and the intergenerational change in the first-front proportion, thereby representing convergence, distribution and search trend.

(3) We introduce a shared-policy, individual-execution mechanism. Individuals independently sample joint strategy-parameter actions from one shared Q-table under the same global state, coordinating the overall search while retaining heterogeneous exploration.

(4) We introduce an action-conditioned successful-parameter memory. Statistics of parameters associated with positive individual contributions are stored separately for each action and reused to guide subsequent parameter generation.

The remainder of this paper is organized as follows. Section 2 introduces DE, RL and multi-objective optimization. Section 3 presents GSG-RLDE. Section 4 evaluates the algorithm on 31 problems from four benchmark suites. Section 5 examines multi-objective unmanned aerial vehicle path planning, and Section 6 concludes the paper.

2 Theoretical background

2.1 Differential evolution

Differential evolution is a population-based stochastic search algorithm for continuous optimization. Each individual represents a candidate solution, and the population is updated iteratively through mutation, crossover and selection.

During initialization, DE randomly generates NP individuals within the feasible variable domain. For a problem of dimension D, let the ith individual be $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$. Its jth variable is expressed as:

$$x_{i,j} = x_{min}^{j} + rand(0,1) \cdot \left(x_{max}^{j} - x_{min}^{j}\right), j = 1,2, \dots, D \tag{2}$$

where $\{x_{min}^{j}, x_{max}^{j}\}$ denotes the lower and upper bounds of each dimension, and $\mathrm{rand}(0,1)$ is a uniformly distributed random number generated in $[0,1]$.

DE then generates a mutant vector. The classical DE/rand/1 strategy is defined as:

$$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}), r_1 \neq r_2 \neq r_3 \neq i \tag{3}$$

where X_{r1}, X_{r2} and X_{r3} are mutually distinct individuals randomly sampled from the current population, and $F \in [0,1]$ is the scaling factor. Their difference vectors define search directions and promote global exploration.

Crossover combines the target vector X_i with the mutant vector V_i to produce a trial vector $U_i = (u_{i,1}, u_{i,2}, \dots, u_{i,D})$. Binomial crossover is commonly expressed as:

$$u_{i,j} = \begin{cases} v_{i,j}, & rand_j(0,1) \leq CR \text{or} j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{4}$$

where $CR \in [0,1]$ is the crossover rate and j_{rand} is a randomly selected dimension, ensuring that the trial vector inherits at least one component from the mutant vector.

Finally, selection determines the individuals retained in the next generation. For a minimization problem, DE commonly uses greedy selection:

$$X_i^{G+1} = \begin{cases} U_i^G, & if\ f(U_i^G) < f(X_i^G) \\ X_i^G, & otherwise \end{cases} \tag{5}$$

where G is the generation index and $f(\cdot)$ is the objective function. Together, these operations determine the search range, information exchange and solution retention in DE.

2.2 Q-learning

Reinforcement learning is commonly formulated as a Markov decision process (MDP), comprising states, actions, rewards and state transitions. An agent selects an action from state feedback and updates its policy using the reward returned by the environment. Fig. 1 illustrates a typical MDP.

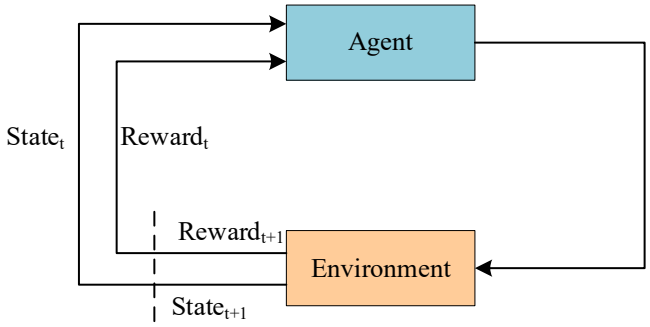


Fig. 1 Markov decision process

At time step t, the agent observes state s_t, selects action a_t and interacts with the environment. The environment returns an immediate reward r_t and transitions to state s_{t+1}. Repeated interactions allow the agent to update its policy from experience and progressively improve its long-term return.

Q-learning is a model-free RL algorithm that learns the value of state-action pairs through repeated environmental interactions. It does not require a predefined state-transition model. Instead, it iteratively updates the action-value function $Q(s,a)$ towards the optimum. For state s_t and action a_t, the Q-value is updated as:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \cdot [r_t + \gamma \cdot max_{a \epsilon A} Q(s_{t+1},a) - Q(s_t,a_t)] \tag{6}$$

where the learning rate α controls the update step, r_t is the immediate reward obtained after action a_t, and the discount factor γ weights future returns. The term $max_{a \epsilon A} Q(s_{t+1},a)$ is the maximum Q-value available in state s_{t+1}. By balancing action exploration with the use of learned experience, Q-learning progressively maps states to decisions. Its workflow is shown in Fig. 2.

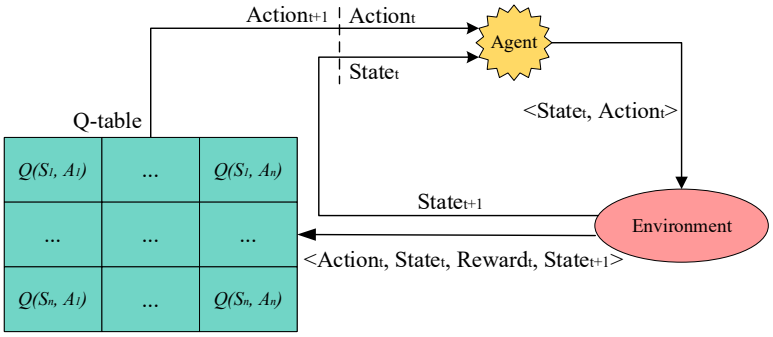


Fig. 2 Q-learning workflow

2.3 Non-dominated sorting and crowding distance

Pareto dominance compares the relative quality of multi-objective solutions but cannot distinguish among mutually non-dominated individuals. NSGA-II therefore uses non-dominated sorting to assign dominance ranks and crowding distance to preserve diversity within each front. We use these measures to characterize population dominance and objective-space distribution for state construction and environmental selection.

2.3.1 Non-dominated sorting

Non-dominated sorting partitions a population according to Pareto dominance. Solutions not dominated by any other population member form the first front. Repeating this procedure on the remaining individuals yields the second and subsequent fronts. The front index is the non-domination rank, and a lower rank indicates a more favourable dominance relation within the current population.

Individuals with lower non-domination ranks receive higher priority during environmental selection. The rank describes relative dominance within the population and does not directly measure distance from the true Pareto front. Fig. 3 illustrates non-dominated sorting.

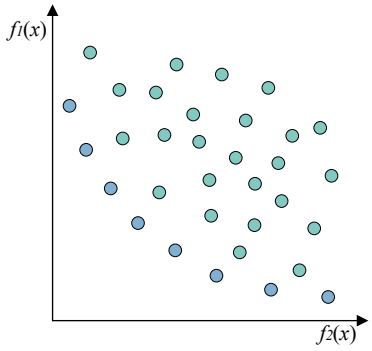


Fig. 3 Illustration of non-dominated sorting

2.3.2 Crowding distance

Non-domination rank alone cannot determine retention order among individuals on the same front. Crowding distance is therefore used to estimate local solution density in objective space.

Individuals on each front are first sorted separately by each objective. The normalized distances to adjacent solutions are then summed across objectives. Boundary solutions are assigned an infinite crowding distance to preserve the extent of the Pareto front. For interior solutions, a larger crowding distance indicates a sparser neighbourhood and consequently a higher retention priority. Fig. 4 illustrates this measure.

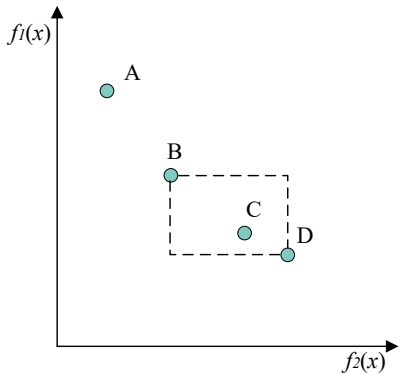


Fig. 4 Illustration of crowding distance

Non-domination rank characterizes population dominance, whereas crowding distance describes local distribution within a front. GSG-RLDE uses the first-front proportion and statistics of its finite crowding distances to construct a global state. This state provides multi-objective information for adaptive strategy and parameter selection.

3 Proposed algorithm

3.1 Motivation

Conventional DE was developed mainly for single-objective optimization, whereas applications in data mining, power engineering and industrial design commonly involve conflicting objectives. These problems require both Pareto-front convergence and a uniform solution distribution, placing greater demands on adaptive search. However, DE mutation strategies and control parameters are often predefined and cannot respond dynamically to population state or objective-space distribution. GSG-RLDE addresses this limitation by using RL to perceive the evolutionary state and adapt search strategies and parameters under multi-objective selection. The resulting control seeks to balance exploration and exploitation while improving convergence and solution diversity.

3.2 Definitions

3.2.1 Multi-objective state-space definition

To obtain a converged and well-distributed Pareto set, GSG-RLDE constructs the RL state using non-dominated sorting and crowding distance. Non-dominated sorting describes Pareto relations and the relative convergence state, whereas crowding distance measures density within a front and characterizes solution diversity.

The proportion of individuals on the first non-dominated front, p_{nd}, serves as a proxy for the overall convergence state. Distribution uniformity is described by the coefficient of variation of crowding distance on that front, $crowd_{CV}$. This coefficient is the ratio of the standard deviation to the mean crowding distance. A larger $crowd_{CV}$ indicates a less uniform distribution of non-dominated solutions.

GSG-RLDE further evaluates the search trend from the change in the proportion of non-dominated solutions between consecutive generations, Δp_{nd}:

$$\Delta p_{nd} = p_{nd}^{t} - p_{nd}^{t-1} \tag{7}$$

where p_{nd}^{t} and p_{nd}^{t-1} are the proportions of first-front solutions in the current and previous generations, respectively.

The global state combines p_{nd}, $crowd_{CV}$ and Δp_{nd}. A low or decreasing non-dominated proportion indicates an exploration or recovery stage. A high $crowd_{CV}$ indicates an uneven distribution that requires stronger diversity maintenance. Otherwise, the search is considered to be progressing steadily. Table 1 defines the states.

Table 1 State-space partitioning

State	Transition condition	Interpretation
State_1	$p_{nd} < 0.25 \vee \Delta p_{nd} < -\frac{1}{N}$	Low or decreasing proportion of non-dominated solutions
State_2	$p_{nd} \geq 0.25 \wedge \Delta p_{nd} \geq -\frac{1}{N} \wedge crowd_{CV} \leq 0.8$	Evolutionary exploration stage
State_3	$crowd_{CV} > 0.8$	Non-uniform distribution of non-dominated solutions

3.2.2 Action-space definition

Mutation strategy determines the search direction, whereas the control parameters F and CR govern the magnitude of differential perturbation and the crossover intensity, respectively. To capture their coupled effects, GSG-RLDE defines each combination of a mutation strategy and parameter intervals as an action.

We use DE/rand/1 and DE/current-to-pbest/1. The former emphasizes global exploration, whereas the latter exploits high-quality solutions to accelerate local search. Each control parameter, F and CR, is assigned one of two intervals: [0.1, 0.5] or [0.6, 1.0]. Combining two mutation strategies with four interval pairs yields eight candidate actions, as listed in Table 2.

Table 2 Action space

F interval	CR interval	DE/current-to-pbest/1	DE/rand/1
[0.1, 0.5]	[0.1, 0.5]	Action 1	Action 5
[0.1, 0.5]	[0.6, 1.0]	Action 2	Action 6
[0.6, 1.0]	[0.1, 0.5]	Action 3	Action 7
[0.6, 1.0]	[0.6, 1.0]	Action 4	Action 8

Large F or CR values generally increase search perturbation and help escape local optima, but may reduce stability. Small values produce conservative search and stable convergence, although they can weaken global exploration. The two intervals provide complementary search behaviours. Jointly encoding strategies and parameters enables GSG-RLDE to select evolutionary behaviour according to the current search state.

3.2.3 Reward function

GSG-RLDE uses a composite reward to evaluate both the population-wide evolutionary trend and the contribution of individual actions. Convergence improvement is first measured by the change in the first-front proportion, Δp_{nd}. When $\Delta p_{nd} > 0$, the current operation has produced more non-dominated solutions and

promoted convergence towards the Pareto front. A negative change instead indicates possible search deterioration.

For diversity, $crowd_{CV}$ measures the non-uniformity of the non-dominated solution distribution. Because a larger $crowd_{CV}$ indicates greater imbalance, the population-level global reward is defined as:

$$R_{global} = \Delta p_{nd} + \lambda(crowd_{CV}^{t-1} - crowd_{CV}^{t}) \tag{8}$$

where λ balances convergence improvement and diversity preservation. The reward favours a larger proportion of non-dominated solutions while penalizing local crowding in objective space.

For action a, GSG-RLDE calculates the mean individual improvement r_a in the current generation. Combining this local contribution with the global reward gives the final reward:

$$r = \omega r_a + (1 - \omega) R_{global} \tag{9}$$

where ω controls the relative influence of individual action contributions and the population-wide evolutionary trend. This reward allows GSG-RLDE to adapt strategy selection to the multi-objective search state and coordinate convergence and diversity across stages.

3.2.4 Action-conditioned successful-parameter memory

Inspired by successful-parameter adaptation in JADE (Zhang and Sanderson 2009) and related adaptive control-parameter methods (Brest et al. 2006), GSG-RLDE maintains an action-conditioned successful-parameter memory. The mechanism stores effective control parameters separately for each action and reuses them in subsequent generations.

For each action *a*, an independent cyclic memory M_a with a capacity equal to the population size *NP* is maintained. Each record contains the first-front proportion p_{nd} the crowding-distance coefficient of variation $crowd_{CV}$, and the parameter statistics $\mu_{F,a}$ and $\mu_{CR,a}$. The former two quantities retain the population context in which the record was generated, whereas parameter generation uses $\mu_{F,a}$ and $\mu_{CR,a}$ The memory structure is illustrated in Table 3.

Table 3 Structure of the action-conditioned successful-parameter memory

Action	**Record**	$\boldsymbol{p_{nd}}$	$\boldsymbol{crowd_{CV}}$	$\boldsymbol{\mu_{F,1}}$	$\boldsymbol{\mu_{CR,a}}$
1	1	p_{nd}	$crowd_{CV}$	$\mu_{F,1}$	$\mu_{CR,1}$
1	⋮	⋮	⋮	⋮	⋮
1	H	p_{nd}	$crowd_{CV}$	$\mu_{F,1}$	$\mu_{CR,1}$
2, ..., A	1, ..., H	p_{nd}	$crowd_{CV}$	$\mu_{F,a}$	$\mu_{CR,a}$

For individual *i*, the contribution of its trial vector is evaluated by comparing each objective with that of the corresponding target vector:

$$d_i = \frac{n_i^+ - n_i^-}{M} \tag{10}$$

where n_{i+} and n_{i-} denote the numbers of improved and degraded objectives, respectively, and M is the number of objectives. For action a, only individuals with $d_i > 0$ are used to update the memory. Let I_{a+} denote this set. The normalized contribution of individual i is

$$w_i = \frac{d_i}{\sum_{j \in I_a^+} d_j}, \qquad i \in I_a^+ \tag{11}$$

The successful F values are summarized using a weighted Lehmer mean, whereas the successful CR values are summarized using a weighted arithmetic mean:

$$\mu_{F,a} = \frac{\sum_{i \in I_a^+} w_i F_i^2}{\sum_{i \in I_a^+} w_i F_i} \tag{12}$$

$$\mu_{CR,a} = \sum_{i \in I_a^+} w_i CR_i \tag{13}$$

The resulting parameter statistics are stored with the current population descriptors in the memory associated with action a. Records are replaced cyclically. If no individual selecting that action produces a positive contribution, the corresponding memory remains unchanged.

When action a is selected, one record is sampled uniformly from M_a. Its parameter statistics are used to generate the control parameters. Specifically, F_i is sampled using a Cauchy perturbation and CR_i using a Gaussian perturbation:

$$F_i = \mu_{F,a}^h + 0.1 \tan[\pi(u - 0.5)], \qquad u \sim U(0,1) \tag{14}$$

$$CR_i = \mu_{CR,a}^h + 0.1z, \qquad z \sim N(0,1) \tag{15}$$

The generated parameters are then restricted to the intervals encoded by the selected action; values outside the admissible range are replaced by the nearest boundary.

This action-conditioned memory enables GSG-RLDE to reuse successful parameter information without expanding the reinforcement-learning state space. Parameter records are retrieved according to the selected action.

3.2.5 Global state-guided individual decision mechanism

Multi-objective optimization must improve convergence while preserving diversity across conflicting objectives. Drawing on the coordination principle of centralized training and decentralized execution (CTDE; Lowe et al. 2017),, GSG-RLDE uses shared global information to coordinate individual evolutionary actions. Its implementation uses one shared tabular value function rather than separate agent-specific policies.

Non-dominated sorting and crowding distance first characterize the population-wide evolutionary state. Under this shared state, every individual samples an action from the same 3 × 8 Q-table and executes the selected mutation strategy and parameter intervals independently. Action selection uses an epsilon-Softmax rule: a random action is selected with probability 0.05; otherwise, an action is sampled from the Softmax distribution with temperature T = 1.

This shared-policy, individual-execution design maintains a coherent search direction while allowing different strategies and parameter settings within the same generation. It reduces dependence on a single operator and supports stage-dependent exploration and exploitation. Fig. 5 shows the mechanism.

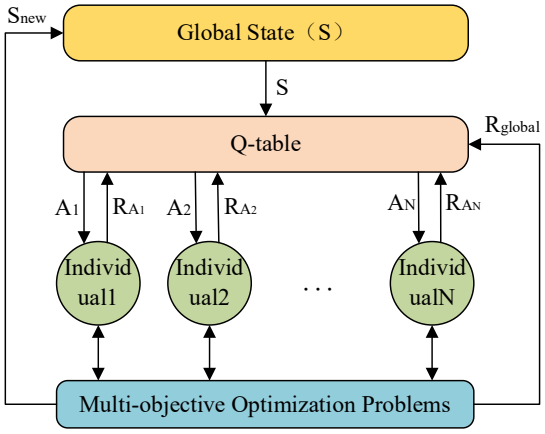


Fig. 5 Global state-guided individual decision mechanism

3.3 Algorithmic framework

Fig. 6 outlines GSG-RLDE. The algorithm initializes the population and evaluates the objective functions. Non-dominated sorting and crowding distance then yield the first-front proportion p_{nd} and its crowding-distance coefficient of variation $crowd_{CV}$, which determine the initial global state. The shared Q-table and action-conditioned parameter memories are initialized concurrently.

At each generation, GSG-RLDE uses the current global state S_t to select an evolutionary action a_{t_i} for every individual. A record from the corresponding action memory is sampled to generate the control parameters. Guided by the shared global state, individuals evolve independently to produce and evaluate offspring.

After offspring evaluation, positive-contribution records update the corresponding action memories, after which environmental selection produces the next population and updates non-dominated sorting, crowding distance and the global state S_{t+1}. The composite reward then combines a population-level global reward with action-level local rewards. These components quantify changes in convergence and distribution, and the mean improvement contributed by each action, respectively.

Finally, the reward updates the shared Q-table, allowing adaptive control of evolutionary strategies and parameters. The procedure repeats until the termination criterion is met.

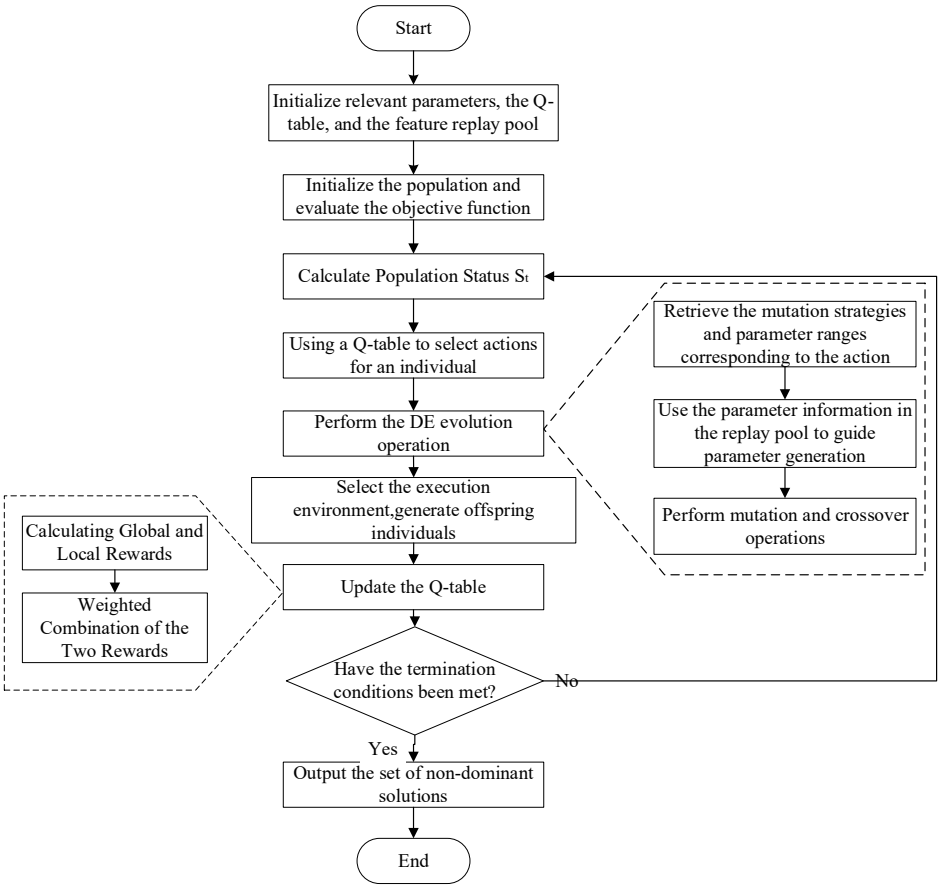


Fig. 6 Workflow of GSG-RLDE

Table 4 Pseudocode for GSG-RLDE

Algorithm 1 GSG-RLDE

(1) Initialize the population size NP, state space, action space, Q-tables, feature replay pools, and reward weights ω and λ;
(2) Initialize population P_0;
(3) Evaluate P_0;
(4) Apply non-dominated sorting to P_0 to obtain FrontNo, and calculate CrowdDis;
(5) Calculate p_{nd}^0 and $crowd_{CV}^0$, and obtain the initial state according to Table 1;
(6) while FES $\leq$ FESMAX
(7) for i=1 to NP
(8) For individual i under current state S_t, select action a_i^t from the Q-table;
(9) Map a_i^t to the corresponding mutation strategy and parameter intervals;
(10) Generate control parameters (F_i, CR_i) from the intervals and successful historical information in the replay pool;
(11) Use the selected mutation strategy and F_i to generate mutant vector V_i;
(12) Apply crossover with CR_i to generate trial vector U_i;
(13) Evaluate U_i;
(14) end
(15) Apply selection to the population and trial-vector set U to obtain offspring population P_{t+1};
(16) Update FrontNo and CrowdDis from the offspring population;
(17) Calculate p_{nd}^{t+1} and $crowd_{CV}^{t+1}$, and obtain offspring state S_{t+1};
(18) Calculate global reward R_{global} using Eq. (8);
(19) Calculate the mean individual improvement contributed by each action to obtain local action reward r_a;
(20) Calculate the composite reward using Eq. (9);
(21) for i=1 to NP
(22) Update the Q-table using the reward for each action a_i^t and R_{global};
(23) end
(24) Update population state $S_{t+1} \rightarrow S_t$;
(25) Update the replay pools;
(26) end
(27) Return the non-dominated solution set;

4 Experimental results and analysis

4.1 Experimental framework

All comparative experiments were conducted in MATLAB R2022a using PlatEMO (Tian et al. 2017). This open-source platform integrates representative MOEAs, standard benchmark suites and commonly used performance indicators. It therefore provides a consistent environment for implementation, testing and comparison. All multi-objective experiments used this platform to support consistent settings and reproducible results.

4.2 Compared algorithms and parameter settings

We compared GSG-RLDE with five representative algorithms: NSGA-II (Deb et al. 2002), MOEA/D (Zhang and Li 2007), MOEA/D-DE (Li and Zhang 2006, 2009), MOCGDE (Tian et al. 2022) and RLMODE (Yu et al. 2024). NSGA-II and MOEA/D represent classical non-dominated sorting and decomposition frameworks, respectively. MOEA/D-DE incorporates DE into MOEA/D, MOCGDE combines DE with conjugate-gradient search, and RLMODE is a recent RL-assisted multi-objective DE method. Table 5 summarizes these algorithms.

Table 5 Compared multi-objective optimization algorithms

Algorithm	Year	Key feature
NSGA-II	2002	Classical baseline based on non-dominated sorting and crowding distance
MOEA/D	2007	Classical decomposition-based multi-objective baseline
MOEA/D-DE	2009	Representative integration of DE into the MOEA/D framework
MOCGDE	2022	Multi-objective algorithm combining DE with conjugate-gradient search
RLMODE	2024	Multi-objective DE based on RL-assisted offspring generation and Pareto-front relearning

Experiments used the ZDT, DTLZ, UF and WFG benchmark suites, covering different objective dimensions, front geometries and search difficulties. ZDT assesses convergence and distribution in bi-objective problems, whereas DTLZ tests scalability to more objectives. UF introduces nonlinear, multimodal and discontinuous features. WFG provides configurable separability, dependence and front geometry for testing robustness in complex search landscapes. Each problem was run independently 30 times. Table 6 reports the settings.

Table 6 Benchmark settings

Problem	Population size	Decision variables	Objectives	Maximum evaluations
ZDT1–ZDT3	100	30	2	10,000
ZDT4,ZDT6	100	10	2	10,000
DTLZ1	105	7	3	30,000
DTLZ2~DTLZ6	105	12	3	30,000
DTLZ7	105	22	3	30,000
UF1–UF7	100	30	2	100,000

Problem	Population size	Decision variables	Objectives	Maximum evaluations
UF8~UF10	100	30	3	100,000
WFG1–WFG9	105	12	3	30,000

Performance was evaluated using hypervolume (HV) and inverted generational distance (IGD). HV measures the objective-space volume dominated by the non-dominated set relative to a reference point, jointly reflecting convergence and diversity. A larger HV indicates a higher-quality solution set. It is defined as:

$$HV(P) = Vol\left(\bigcup_{x\in P}\left[f_1(x), z_{ref}^1\right] \times \cdots \times \left[f_m(x), z_{ref}^m\right]\right) \tag{16}$$

where P is the non-dominated solution set, z_{ref} is the reference point, and $Vol(\cdot)$ denotes the Lebesgue measure.

IGD is the mean distance from reference points on the true Pareto front to the obtained solution set. It primarily measures approximation accuracy, with a smaller value indicating closer agreement with the true front. IGD is defined as:

$$IGD\left(P, z_{ref}\right) = \frac{\sum_{z\in z_{ref}} min_{x\in p} d(x,z)}{\left|z_{ref}\right|} \tag{17}$$

where $d(x,z)$ is the Euclidean distance between solution x and reference point z.

4.3 Comparative results

We evaluated GSG-RLDE against the five comparison algorithms. Tables 7 and 8 report the HV and IGD statistics, respectively. The symbols '+', '-' and '=' indicate that a comparison algorithm performed better than, worse than or similarly to GSG-RLDE, respectively.

Table 7 HV obtained by six algorithms on ZDT, DTLZ, UF and WFG

Func.	MOEA/D	NSGA-II	MOEA/D-DE	MOCGDE	RLMODE	GSG-RLDE
ZDT1	5.5934e-1 (5.48e-2) -	7.0623e-1 (2.98e-3) =	2.4497e-1 (8.89e-2) -	**7.2041e-1 (2.72e-4) +**	7.0753e-1 (2.79e-3) +	7.0522e-1 (2.46e-3)
ZDT2	1.0312e-1 (2.08e-2) -	4.0998e-1 (4.13e-2) =	1.8278e-2 (6.17e-2) -	4.1997e-1 (7.88e-2) -	**4.3231e-1 (4.00e-3) +**	4.2529e-1 (3.09e-3)
ZDT3	5.6797e-1 (6.19e-2) -	5.9823e-1 (2.32e-2) +	2.8468e-1 (7.95e-2) -	**6.1031e-1 (3.49e-2) +**	5.8980e-1 (3.22e-3) =	5.8985e-1 (2.30e-3)
ZDT4	2.5345e-1 (1.28e-1) +	**4.7040e-1 (1.37e-1) +**	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	5.5558e-2 (1.29e-1) +	1.0286e-2 (2.55e-2)
ZDT6	2.8889e-1 (3.37e-2) -	3.1208e-1 (3.37e-2) -	3.4210e-1 (9.18e-2) -	**3.8886e-1 (2.60e-5) +**	3.8724e-1 (2.73e-4) +	3.8697e-1 (2.51e-4)
DTLZ1	**8.4014e-1 (3.27e-3) +**	8.2229e-1 (6.27e-3) =	7.5227e-1 (1.65e-1) +	0.0000e+0 (0.00e+0) -	3.2211e-1 (3.64e-1) -	5.9769e-1 (3.29e-1)
DTLZ2	5.6280e-1 (5.02e-5) +	5.3320e-1 (3.40e-3) -	5.3184e-1 (2.23e-3) -	**5.6326e-1 (6.59e-4) +**	5.2499e-1 (3.94e-3) -	5.3623e-1 (1.95e-3)
DTLZ3	1.7764e-1 (2.14e-1) +	**2.9347e-1 (2.29e-1) +**	2.0236e-1 (2.27e-1) +	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	3.3828e-2 (1.29e-1)
DTLZ4	4.0527e-1 (1.49e-1) =	5.0753e-1 (1.13e-1) =	5.1971e-1 (2.61e-2) -	2.2496e-1 (1.09e-1) -	5.2705e-1 (4.12e-3) -	**5.3547e-1 (3.55e-3)**
DTLZ5	1.8339e-1 (3.60e-5) -	1.9920e-1 (2.08e-4) -	1.9553e-1 (9.61e-5) -	**2.0015e-1 (5.66e-5) +**	1.9896e-1 (2.32e-4) -	1.9971e-1 (1.63e-4)
DTLZ6	1.8331e-1 (1.42e-5) -	1.9962e-1 (1.85e-4) +	1.9594e-1 (2.13e-5) -	**2.0017e-1 (5.64e-5) +**	1.9942e-1 (1.83e-4) =	1.9948e-1 (1.49e-4)
DTLZ7	2.5820e-1	2.6684e-1	2.0222e-1	2.7049e-1	2.6899e-1 (6.04e-	**2.7155e-1**

Func.	MOEA/D	NSGA-II	MOEA/D-DE	MOCGDE	RLMODE	GSG-RLDE
	(1.40e-3) -	(7.57e-3) -	(2.09e-2) -	(1.99e-2) -	3) -	**(1.45e-3)**
UF1	4.5806e-1 (6.59e-2) -	5.7775e-1 (4.17e-2) -	**6.9448e-1 (3.44e-2) +**	6.4789e-1 (3.45e-2) =	6.4868e-1 (2.43e-2) =	6.3706e-1 (2.35e-2)
UF2	6.3407e-1 (3.05e-2) -	6.7824e-1 (9.62e-3) -	6.9095e-1 (1.90e-2) =	**7.0380e-1 (3.29e-3) +**	6.9969e-1 (1.46e-3) +	6.9733e-1 (3.42e-3)
UF3	3.6904e-1 (4.58e-2) -	4.4487e-1 (5.00e-2) -	**5.5642e-1 (9.41e-2) +**	4.7437e-1 (4.73e-2) -	5.2579e-1 (2.20e-2) =	5.1805e-1 (4.31e-2)
UF4	3.3942e-1 (4.28e-3) -	3.8321e-1 (1.03e-3) -	3.3538e-1 (1.04e-2) -	2.8879e-1 (1.04e-2) -	3.8902e-1 (1.07e-3) -	**3.9000e-1 (1.05e-3)**
UF5	1.5617e-1 (6.19e-2) =	**2.2944e-1 (7.10e-2) +**	1.1674e-1 (8.48e-2) =	0.0000e+0 (0.00e+0) -	1.9712e-1 (8.01e-2) =	1.5946e-1 (7.97e-2)
UF6	2.2747e-1 (5.96e-2) -	2.9201e-1 (7.54e-2) -	2.6697e-1 (7.47e-2) -	1.4527e-1 (6.07e-2) -	3.1533e-1 (8.45e-2) =	**3.4464e-1 (5.96e-2)**
UF7	2.7805e-1 (1.20e-1) -	4.5434e-1 (9.72e-2) -	5.2411e-1 (1.18e-1) -	5.1805e-1 (7.84e-2) =	**5.5265e-1 (2.18e-2) +**	5.4795e-1 (7.80e-3)
UF8	2.9077e-1 (1.01e-1) +	2.6713e-1 (5.41e-2) =	3.1415e-1 (3.31e-2) +	**3.4823e-1 (8.63e-2) +**	2.4764e-1 (5.28e-2) =	2.5093e-1 (5.83e-2)
UF9	4.5484e-1 (3.93e-2) +	3.9911e-1 (8.73e-2) =	5.4722e-1 (5.52e-2) +	**6.3132e-1 (1.08e-1) +**	2.9655e-1 (1.60e-1) -	4.0555e-1 (1.23e-1)
UF10	**1.0617e-1 (2.71e-2) +**	7.0106e-2 (5.14e-2) =	3.9427e-2 (3.17e-2) =	0.0000e+0 (0.00e+0) -	5.5035e-2 (3.84e-2) =	6.1703e-2 (4.85e-2)
WFG1	8.4833e-1 (2.61e-2) +	**9.1691e-1 (1.42e-2) +**	4.1590e-1 (3.55e-2) -	2.1425e-1 (1.42e-2) -	7.3961e-1 (4.41e-2) -	8.0815e-1 (3.59e-2)
WFG2	8.9302e-1 (9.61e-3) -	9.1906e-1 (2.83e-3) -	8.8317e-1 (5.57e-3) -	8.4169e-1 (4.09e-2) -	9.1377e-1 (2.92e-3) -	**9.2269e-1 (1.75e-3)**
WFG3	3.3689e-1 (2.63e-2) -	**3.9714e-1 (3.09e-3) +**	3.5041e-1 (1.52e-2) -	3.2693e-1 (2.78e-2) -	3.7268e-1 (5.74e-3) -	3.8809e-1 (3.48e-3)
WFG4	**5.2841e-1 (4.52e-3) +**	5.1925e-1 (4.72e-3) +	4.7318e-1 (4.99e-3) -	4.4211e-1 (3.19e-2) -	4.8846e-1 (4.21e-3) -	4.9572e-1 (5.92e-3)
WFG5	**5.0085e-1 (3.41e-3) +**	4.9212e-1 (5.70e-3) +	4.6392e-1 (2.79e-3) -	4.2966e-1 (2.63e-2) -	4.8008e-1 (4.54e-3) =	4.7980e-1 (4.53e-3)
WFG6	4.8040e-1 (1.72e-2) -	4.7550e-1 (1.32e-2) -	4.2736e-1 (4.82e-2) -	4.8833e-1 (1.74e-2) =	4.7046e-1 (1.67e-2) -	**4.9723e-1 (1.16e-2)**
WFG7	4.9307e-1 (2.42e-2) =	**5.2188e-1 (4.59e-3) +**	5.0070e-1 (4.04e-3) -	4.3072e-1 (4.72e-2) -	4.9174e-1 (6.58e-3) -	5.0592e-1 (6.26e-3)
WFG8	**4.5337e-1 (5.16e-3) +**	4.4148e-1 (2.99e-3) +	4.0423e-1 (1.19e-2) -	3.6571e-1 (2.20e-2) -	4.1782e-1 (6.03e-3) -	4.3418e-1 (4.00e-3)
WFG9	4.7459e-1 (2.48e-2) -	**4.9929e-1 (5.00e-3) +**	4.8613e-1 (4.51e-3) -	4.1622e-1 (3.48e-2) -	4.7427e-1 (3.63e-2) -	4.9220e-1 (4.07e-3)
+/-/=	11/17/3	12/12/7	6/22/3	9/18/4	6/15/10	

Table 8 IGD obtained by six algorithms on ZDT, DTLZ, UF and WFG

Func.	MOEA/D	NSGA-II	MOEA/D-DE	MOCGDE	RLMODE	GSG-RLDE
ZDT1	1.5271e-1 (6.43e-2) -	1.2496e-2 (2.15e-3) =	4.3997e-1 (1.03e-1) -	**4.0470e-3 (3.71e-4) +**	1.1615e-2 (1.99e-3) +	1.3239e-2 (1.83e-3)
ZDT2	5.2895e-1 (7.79e-2) -	2.9680e-2 (4.15e-2) =	6.9685e-1 (1.96e-1) -	3.5276e-2 (1.05e-1) -	**1.1101e-2 (2.58e-3) +**	1.5853e-2 (2.11e-3)
ZDT3	1.5188e-1 (5.22e-2) -	1.2905e-2 (7.65e-3) +	4.6033e-1 (8.61e-2) -	**1.1496e-2 (2.39e-2) +**	1.4106e-2 (3.36e-3) =	1.5284e-2 (2.18e-3)
ZDT4	4.3134e-1 (1.67e-1) +	**2.5813e-1 (1.41e-1) +**	3.6307e+0 (1.36e+0) -	4.3997e+1 (1.33e+1) -	1.7015e+0 (1.32e+0) =	1.6283e+0 (6.81e-1)
ZDT6	7.5130e-2 (3.15e-2) -	6.0908e-2 (2.86e-2) -	6.3122e-2 (1.29e-1) -	**3.1385e-3 (3.42e-5) +**	4.7656e-3 (2.76e-4) +	5.0341e-3 (2.42e-4)
DTLZ1	**1.9498e-2 (4.57e-**	2.6433e-2	4.8625e-2	3.6169e+1	4.1825e-1	1.2682e-1

Func.	MOEA/D	NSGA-II	MOEA/D-DE	MOCGDE	RLMODE	GSG-RLDE
	4) +	(1.64e-3) =	(7.12e-2) +	(5.28e-2) -	(5.21e-1) -	(1.54e-1)
DTLZ2	**5.0305e-2 (1.61e-6) +**	6.7871e-2 (2.62e-3) -	6.9753e-2 (7.57e-4) -	5.2356e-2 (3.80e-4) +	6.8426e-2 (2.38e-3) -	6.3743e-2 (2.21e-3)
DTLZ3	1.1960e+0 (1.29e+0) +	**4.4183e-1 (4.87e-1) +**	7.8095e+0 (1.36e+1) =	2.5000e+2 (2.66e-5) -	6.5114e+0 (4.66e+0) -	2.6040e+0 (2.42e+0)
DTLZ4	3.8571e-1 (3.02e-1) =	1.2523e-1 (2.23e-1) -	1.3061e-1 (7.94e-2) -	6.4789e-1 (1.94e-1) -	6.7642e-2 (2.24e-3) -	**6.4484e-2 (1.89e-3)**
DTLZ5	3.1115e-2 (7.23e-5) -	5.4996e-3 (2.83e-4) -	1.2161e-2 (1.01e-4) -	**3.9567e-3 (2.95e-5) +**	5.5304e-3 (3.01e-4) -	5.2331e-3 (2.52e-4)
DTLZ6	3.1262e-2 (5.87e-5) -	5.5271e-3 (2.66e-4) +	1.2293e-2 (4.31e-5) -	3.9641e-3 (2.75e-5) +	5.9764e-3 (4.09e-4) =	5.7835e-3 (3.95e-4)
DTLZ7	1.3580e-1 (2.28e-3) -	9.4808e-2 (7.00e-2) =	2.1778e-1 (1.25e-1) -	1.2213e-1 (1.39e-1) -	8.5313e-2 (4.94e-2) =	**7.7680e-2 (4.00e-3)**
UF1	2.8389e-1 (1.16e-1) -	1.1835e-1 (3.66e-2) -	**1.8947e-2 (2.14e-2) +**	5.6540e-2 (2.30e-2) =	5.1739e-2 (1.41e-2) +	6.6904e-2 (1.71e-2)
UF2	1.2939e-1 (6.50e-2) -	4.1655e-2 (1.74e-2) -	3.1467e-2 (2.73e-2) =	1.8783e-2 (4.22e-3) +	**1.8589e-2 (1.30e-3) +**	2.1557e-2 (3.27e-3)
UF3	3.1989e-1 (3.12e-2) -	2.3563e-1 (4.56e-2) -	1.1275e-1 (6.04e-2) +	2.0285e-1 (5.12e-2) -	1.5212e-1 (2.18e-2) =	1.5652e-1 (3.50e-2)
UF4	7.1293e-2 (3.84e-3) -	4.6049e-2 (8.00e-4) -	7.9392e-2 (8.20e-3) -	1.1559e-1 (8.69e-3) -	**4.1926e-2 (6.24e-4) =**	4.1939e-2 (1.08e-3)
UF5	5.0890e-1 (9.76e-2) -	**2.5641e-1 (4.59e-2) +**	4.9048e-1 (1.17e-1) -	2.9832e+0 (5.53e-1) -	3.1814e-1 (1.44e-1) =	3.3977e-1 (9.24e-2)
UF6	4.2478e-1 (1.17e-1) -	2.0598e-1 (1.30e-1) =	2.6849e-1 (1.35e-1) -	3.4937e-1 (1.09e-1) -	1.9779e-1 (1.32e-1) =	**1.6787e-1 (1.01e-1)**
UF7	3.9305e-1 (1.90e-1) -	1.4287e-1 (1.36e-1) -	6.5785e-2 (1.57e-1) -	6.8573e-2 (1.00e-1) =	**2.3083e-2 (1.53e-2)** +	2.7910e-2 (5.70e-3)
UF8	3.5824e-1 (2.63e-1) =	2.8639e-1 (6.91e-2) =	**2.4681e-1 (6.95e-2) +**	3.4136e-1 (1.52e-1) =	2.8678e-1 (8.80e-2) =	3.0033e-1 (8.21e-2)
UF9	3.1950e-1 (4.24e-2) =	3.4965e-1 (8.74e-2) =	2.5770e-1 (4.79e-2) +	**2.0334e-1 (1.28e-1) +**	4.2123e-1 (1.62e-1) -	3.4055e-1 (1.11e-1)
UF10	7.0661e-1 (1.11e-1) -	5.1766e-1 (1.34e-1) =	6.9663e-1 (1.04e-1) -	4.7256e+0 (1.51e+0) -	5.5984e-1 (1.15e-1) -	**4.9746e-1 (1.16e-1)**
WFG1	3.1826e-1 (3.33e-2) +	2.3849e-1 (2.90e-2) +	**1.1834e+0 (1.01e-1) -**	1.6394e+0 (2.25e-2) -	6.7113e-1 (5.90e-2) -	6.3178e-1 (7.93e-2)
WFG2	2.5027e-1 (1.12e-2) -	2.1481e-1 (1.17e-2) -	3.3420e-1 (1.75e-2) -	2.5976e-1 (5.76e-2) -	2.1375e-1 (9.97e-3) -	**2.0509e-1 (1.32e-2)**
WFG3	2.0142e-1 (7.34e-2) -	**9.6697e-2 (1.60e-2) +**	1.5275e-1 (2.52e-2) -	2.1196e-1 (6.53e-2) -	1.4586e-1 (1.99e-2) -	1.0648e-1 (1.18e-2)
WFG4	**2.5040e-1 (5.46e-3) +**	2.6679e-1 (1.14e-2) +	3.6257e-1 (1.17e-2) -	3.0750e-1 (4.11e-2) -	3.0419e-1 (9.02e-3) -	2.9278e-1 (9.97e-3)
WFG5	**2.3770e-1 (4.71e-3) +**	2.7357e-1 (9.91e-3) =	3.1337e-1 (4.39e-3) -	3.0953e-1 (3.90e-2) -	2.7708e-1 (9.29e-3) =	2.7701e-1 (9.09e-3)
WFG6	2.7833e-1 (1.75e-2) =	2.9875e-1 (1.52e-2) -	3.8238e-1 (4.41e-2) -	**2.5806e-1 (2.34e-2) +**	3.1792e-1 (2.15e-2) -	2.8350e-1 (1.36e-2)
WFG7	3.3381e-1 (4.40e-2) -	**2.7324e-1 (1.08e-2) +**	3.3122e-1 (5.33e-3) -	4.4700e-1 (1.29e-1) -	3.0561e-1 (1.27e-2) -	2.8356e-1 (1.11e-2)
WFG8	**3.0885e-1 (1.07e-2) +**	3.6709e-1 (1.30e-2) =	4.0074e-1 (1.52e-2) -	4.6540e-1 (4.71e-2) -	3.9466e-1 (1.29e-2) -	3.6509e-1 (1.01e-2)
WFG9	2.8047e-1 (2.68e-2) =	**2.7041e-1 (9.12e-3) =**	3.0839e-1 (4.10e-3) -	3.4188e-1 (5.60e-2) -	2.8914e-1 (4.26e-2) =	2.7248e-1 (1.08e-2)
+/-/=	8/18/5	9/11/11	5/24/2	9/19/3	6/14/11	

Across 31 problems from the ZDT, DTLZ, UF and WFG suites, GSG-RLDE showed strong overall competitiveness. It achieved better or comparable HV and IGD values on most problems relative to MOEA/D-

DE, MOCGDE and RLMODE. Comparisons with NSGA-II and MOEA/D were more problem dependent. Overall, GSG-RLDE remained competitive in solution-set distribution and Pareto-front approximation.

Relative to NSGA-II, GSG-RLDE performed particularly well on ZDT6, DTLZ2, DTLZ5, DTLZ7, UF1-UF4, UF6-UF7, WFG2 and WFG6. These problems commonly contain biased search spaces, irregular or degenerate fronts, or difficult convergence-diversity trade-offs. By perceiving multi-objective evolutionary states, GSG-RLDE adapted its search strategies and control parameters near complex fronts. This mechanism may support more stable optimization on problems with non-uniform search spaces or deceptive convergence, including ZDT6 and WFG6.

We further applied the Friedman test to the HV and IGD results. Table 9 reports the mean ranks. GSG-RLDE attained the best mean rank for both indicators, demonstrating strong aggregate performance across problems.

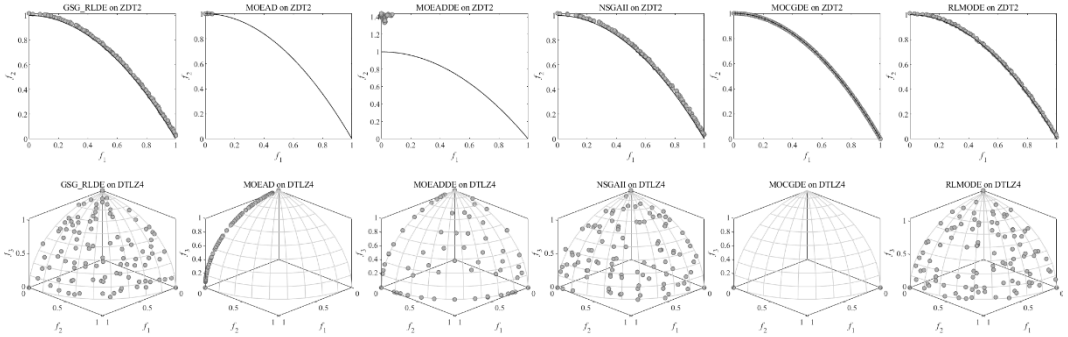
Table 9 Mean Friedman ranks across the four benchmark suites

Algorithm	Mean rank (HV)	Mean rank (IGD)
GSG-RLDE	2.7097	2.6129
MOEA/D	3.8710	4.0645
NSGA-II	2.8710	2.7419
MOEA/D-DE	4.1774	4.4516
MOCGDE	3.9355	4.0968
RLMODE	3.4355	3.0323

Fig. 7 compares the Pareto-front approximations obtained on representative problems. On ZDT2, GSG-RLDE, NSGA-II, MOCGDE and RLMODE closely approximated the true front. MOEA/D and MOEA/D-DE instead produced locally clustered solutions.

UF4 has an irregular, nonlinear Pareto front that requires effective exploration. GSG-RLDE approached the true front while maintaining a relatively continuous and uniform solution distribution, indicating adaptability to complex MOPs.

For the three-objective DTLZ4 and WFG6 problems, complex search spaces and stronger objective coupling accentuated differences in convergence and diversity. MOEA/D-DE did not fully cover the front and left unevenly populated or empty regions. GSG-RLDE provided more stable coverage and distribution uniformity.



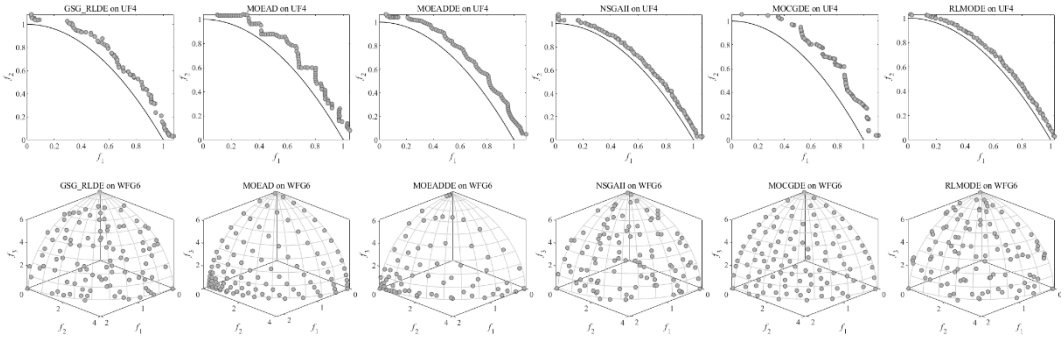


Fig. 7 Pareto fronts corresponding to the mean HV of six algorithms on ZDT2, DTLZ4, UF4 and WFG6

Together, the numerical results, Friedman ranks and Pareto-front distributions establish the overall competitiveness of GSG-RLDE. They also indicate that RL-assisted control can improve the adaptive search capability of DE in multi-objective optimization.

4.4 Time-complexity analysis

The computational cost of GSG-RLDE arises mainly from state acquisition, action selection and parameter generation, environmental selection, and reward calculation. Let NP be the population size, M the number of objectives, A the number of actions and G the maximum number of generations. The corresponding complexities are $O(NP)$ for state acquisition, $O(NP \cdot A)$ for action selection and parameter generation, and $O(NP + A^2)$ for reward calculation.

Environmental selection dominates the cost. Non-dominated sorting requires $O(M \cdot NP^2)$, whereas crowding-distance calculation requires $O(M \cdot NP\log NP)$. Because the action count A is constant, non-dominated sorting determines the leading term. Over G generations, the total time complexity of GSG-RLDE is $O(G \cdot M \cdot NP^2)$. Thus, its leading complexity is of the same order as that of most MOEAs based on non-dominated sorting.

5 Engineering application

As unmanned aerial vehicle (UAV) applications expand, path planning increasingly determines mission efficiency, flight safety and system reliability. Multi-objective UAV path planning must jointly consider route length, safety, energy consumption and constraint satisfaction. Evolutionary algorithms have consequently been applied to constrained and multi-objective path-planning problems. Zhang et al. (2024) formulated multi-UAV path planning as a constrained MOP to improve feasibility and distribution uniformity. Xu et al. (2024) used multi-objective evolution to coordinate conflicting objectives such as route length and flight altitude. These studies support the use of evolutionary algorithms for UAV path-planning trade-offs. We therefore apply GSG-RLDE to this problem to assess its effectiveness in an engineering setting.

5.1 Multi-objective UAV path-planning model

UAV path planning seeks an executable trajectory from a start point to a destination while satisfying environmental constraints and manoeuvrability requirements. In complex three-dimensional environments, a shorter route improves mission efficiency but may increase exposure to terrain or no-fly zones. Improving safety generally requires additional detours. Route efficiency and flight safety therefore form two competing objectives.

Let a UAV trajectory comprise the ordered waypoint set $Q = (q_1, q_2, \ldots, q_M)$, where $q_i = (x_i, y_i, z_i)$, and adjacent waypoints define a flight segment. We evaluate each trajectory using route length, terrain safety, no-fly-zone avoidance and path smoothness.

Route length is the total distance travelled from the start point to the destination, calculated as the sum of Euclidean distances between adjacent waypoints:

$$J_{len} = \sum_{i=1}^{M-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{18}$$

A smaller J_{len} indicates a shorter route, reducing unnecessary detours and improving mission efficiency.

The terrain-safety cost measures the deviation between flight altitude and the prescribed safe-altitude range:

$$J_{crash} = \left(\sum_{i=1}^{M} C_i \right) \cdot J_{\infty} + \sum_{i=1,\, h_i \geq z_{min}}^{M} \left| h_i - \frac{z_{max} + z_{min}}{2} \right| \tag{19}$$

where $[z_{min}, z_{max}]$ is the safe-altitude range, h_i is the altitude at the current waypoint, C_i indicates violation of the minimum-altitude constraint, and J_{∞} is the collision penalty. A large penalty is imposed when the route falls below the minimum safe altitude or intersects the terrain.

The no-fly-zone threat cost measures spatial conflict between a route and restricted airspace. Suppose the environment contains k no-fly zones, and $d_{k,j}$ is the minimum distance from the centre of zone k to route segment j. The cost is then:

$$J_{ban} = \sum_{k=1}^{K} \sum_{j=1}^{M-1} \phi_k(d_{k,j}) \tag{20}$$

Here, $\phi_k(\cdot)$ is a distance-based penalty function. A large collision penalty applies inside a no-fly zone. Within the safety buffer, the penalty increases as distance decreases, and becomes zero beyond the prescribed clearance. The function incorporates UAV size, buffer distance and the corresponding penalty coefficients.

The smoothness cost limits abrupt changes in trajectory direction. From changes in climb angle and horizontal turning angle between adjacent segments, we define:

$$J_{turn} = \sum_{i=1}^{M-2} \psi(\Delta\theta_i) + \chi(\Delta\tau_i) \tag{21}$$

where $\Delta\theta_i$ and $\Delta\tau_i$ are changes in climb and turning angles between adjacent segments, and $\psi(\cdot)$ and $\chi(\cdot)$ are their penalty functions. The penalty is zero within the allowable limit and increases beyond the maximum climb or turning angle. This cost reduces sharp turns and abrupt climbs, improving trajectory feasibility.

Using these costs, we formulate UAV path planning as a bi-objective problem that balances flight efficiency and safety:

$$minf(x) = \begin{cases} f_1(x) = J_{len} + J_{turn} \\ f_2(x) = J_{crash} + J_{ban} \end{cases} \tag{22}$$

Objective f_1 combines route length and smoothness to assess flight efficiency and trajectory feasibility. Objective f_2 combines terrain safety and no-fly-zone threat to assess flight safety. Jointly optimizing these competing objectives yields Pareto-optimal candidate routes with different efficiency-safety trade-offs, supporting mission-specific decisions.

5.2 Experimental settings and results

We applied GSG-RLDE to multi-objective UAV path planning to evaluate its effectiveness in an engineering optimization problem. All experiments were conducted in MATLAB R2022a, with multi-objective optimization implemented in PlatEMO. We first describe the solution encoding and parameter settings, then compare algorithmic performance in the path-planning scenarios.

Candidate routes were represented using waypoint-based real-valued encoding. A UAV flies from a fixed start point to a destination through K adjustable intermediate control points, giving each individual dimension $D = 3K$. The decision vector is:

$$X = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_K, y_K, z_K) \tag{23}$$

Table 10 lists the experimental settings. The start and destination coordinates were (1000, 12000, 0) and (9500, 1000, 0), respectively, and the altitude constraint was [50, 300]. The maximum horizontal turning and climb angles were both 45°. Each algorithm used a population of 100, a maximum of 30000 evaluations and 30 independent runs

Table 10 Experimental settings

Parameter	Value
Start coordinates	(1000, 12000, 0)
Destination coordinates	(9500, 1000, 0)
Altitude constraint	[50, 300]
Maximum horizontal turning angle	45°
Maximum climb angle	45°
Population size	100
Weight coefficients (w_{len}, w_{crash}, w_{ban} and w_{turn})	(1, 2, 5, 10)
Function evaluations	30,000
Independent runs	30

GSG-RLDE was compared with NSGA-II, MOEA/D, MOEA/D-DE, MOCGDE and RLMODE. Performance was assessed using HV and pure diversity (PD) (PD; Wang et al. 2017). HV measures convergence and coverage of the non-dominated set, whereas PD characterizes its distributional diversity in objective space. Larger HV generally indicates better overall solution quality, and larger PD indicates a broader range of trade-off solutions.

Tables 11 and 12 report statistics from 30 independent runs in the simple and complex scenarios. In the simple scenario, GSG-RLDE achieved the highest mean HV, exceeding NSGA-II, MOEA/D, MOEA/D-DE, MOCGDE and RLMODE by 1.24%, 8.60%, 1.94%, 91.59% and 1.44%, respectively. In the complex scenario,

its mean HV remained highest, with corresponding improvements of 0.97%, 16.40%, 9.83%, 83.23% and 2.29%. GSG-RLDE therefore produced high-quality non-dominated sets in both scenarios.

GSG-RLDE also had a lower HV standard deviation than every comparison algorithm in both scenarios, indicating greater run-to-run stability. Its PD values showed that high solution quality was accompanied by substantial distributional diversity. By contrast, the markedly lower mean PD of MOEA/D suggests that its solutions were concentrated in local regions.

Table 11 Results of six algorithms in the simple multi-objective scenario

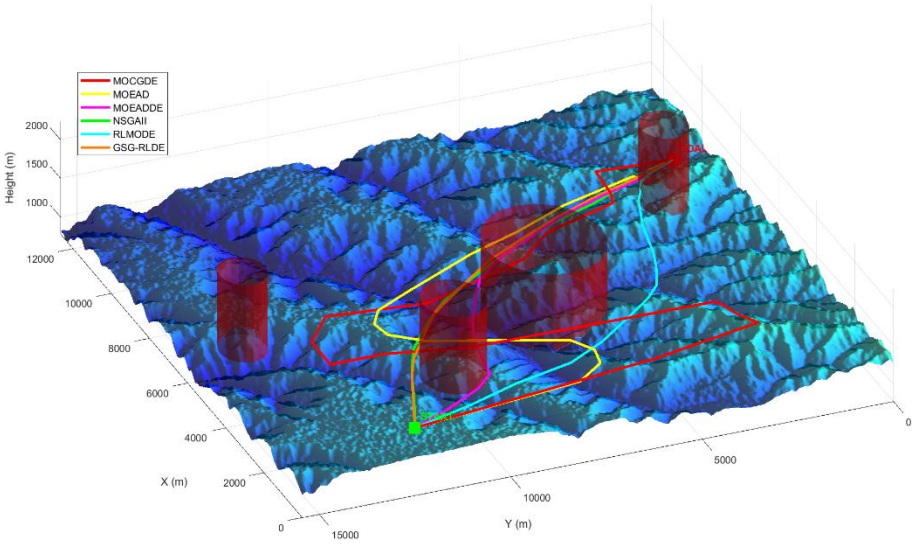
Algorithm	Mean HV	HV standard deviation	Mean PD	PD standard deviation
GSG-RLDE	0.8179	0.0083	8.5019E+08	1.4626E+09
NSGA-II	0.8079	0.0206	1.5997E+08	3.5292E+08
MOEA/D	0.7531	0.0442	4.2465	22.029
MOEA/D-DE	0.8023	0.0231	4.1728E+06	8.5049E+06
MOCGDE	0.4269	0.0985	1.8342E+07	5.0058E+07
RLMODE	0.8063	0.0242	6.3222E+08	9.7125E+08

Table 12 Results of six algorithms in the complex multi-objective scenario

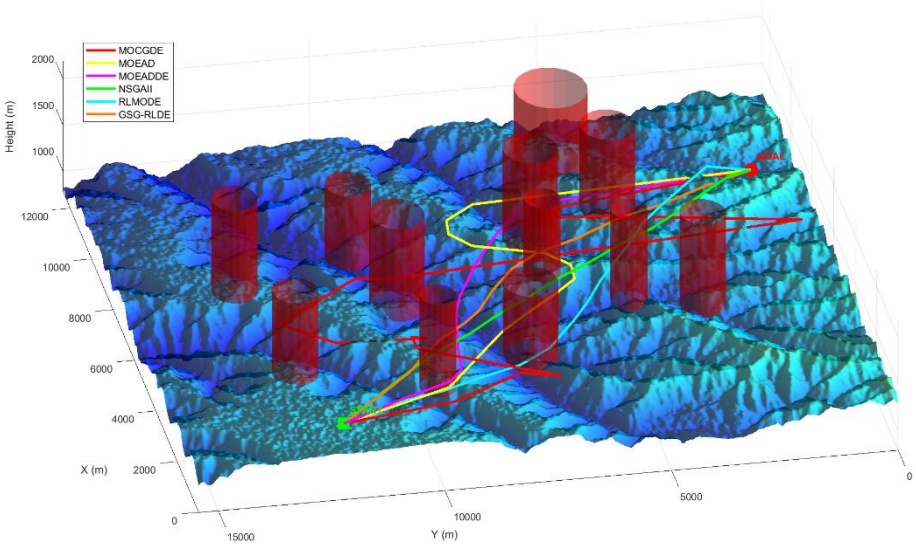
Algorithm	Mean HV	HV standard deviation	Mean PD	PD standard deviation
GSG-RLDE	0.8260	0.0076	4.7126E+08	5.0449E+08
NSGA-II	0.8181	0.0185	3.019E+08	3.3985E+08
MOEA/D	0.7096	0.1093	3342.7	7546.3
MOEA/D-DE	0.7521	0.0516	3.2343E+07	6.5592E+07
MOCGDE	0.4508	0.0881	2.5237E+07	5.93E+07
RLMODE	0.8075	0.0259	4.083E+08	5.1857E+08

Figs. 8 and 9 show the three-dimensional routes, horizontal projections and vertical altitude profiles. In both terrains, GSG-RLDE generated relatively short routes with effective obstacle avoidance. Its smooth altitude profiles avoided abrupt climbs and descents, which may reduce energy consumption and improve flight stability.

MOEA/D generated feasible routes, but they were more tortuous, with visible horizontal detours and larger vertical fluctuations. This behaviour indicates sensitivity to local feasible regions under complex constraints.

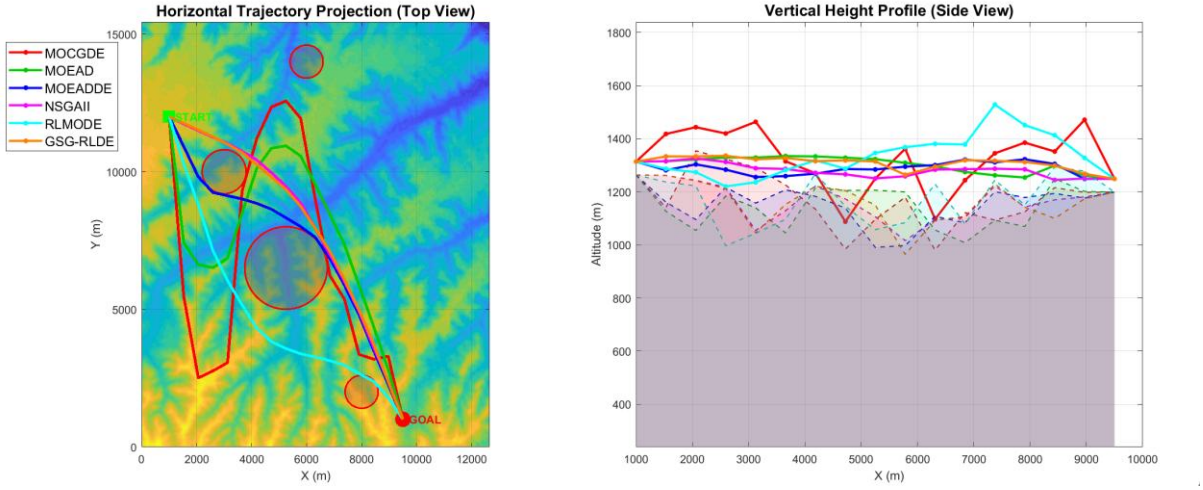


(a) Simple scenario

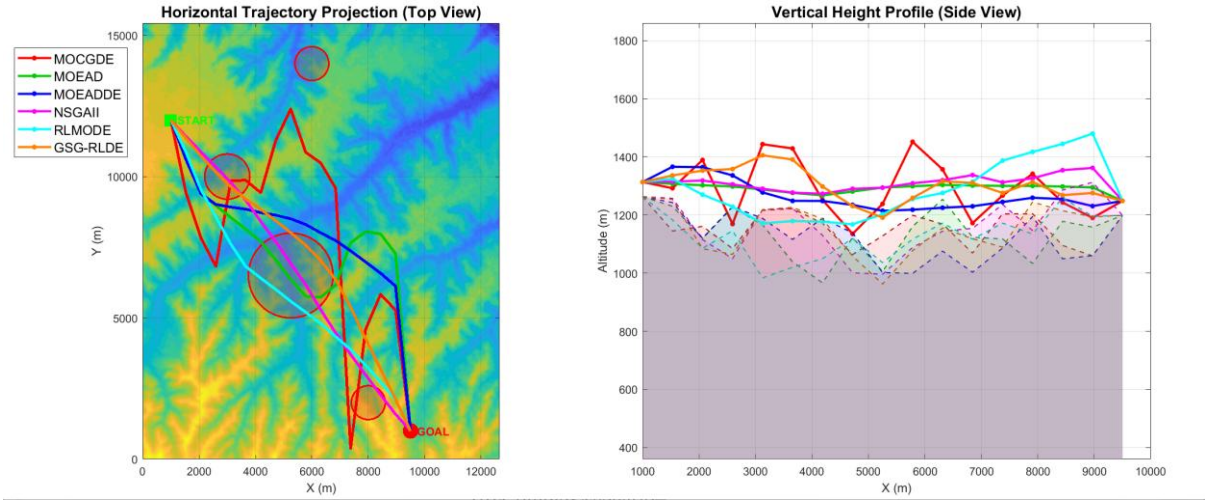


(b) Complex scenario

Fig. 8 Three-dimensional routes generated by the six algorithms



(a) Simple scenario



(b) Complex scenario

Fig. 9 Horizontal and vertical route profiles generated by the six algorithms

The quantitative indicators and route visualizations jointly show that GSG-RLDE provides competitive convergence and diversity in multi-objective UAV path planning, with potential for engineering applications.

6 Conclusion

We proposed GSG-RLDE, an RL-assisted DE algorithm that integrates a multi-objective global state, shared Q-learning control with individual action execution, and action-conditioned successful-parameter memory. Dominance, distribution and intergenerational trend information identify the current search stage, while joint strategy-parameter actions adapt the search behaviour.

Across the ZDT, DTLZ, UF and WFG suites, GSG-RLDE showed competitive convergence, solution diversity and global search performance. In multi-objective UAV path planning, it generated short, smooth and safe routes, with stable HV and PD results. These findings support its use beyond standard benchmarks for constrained engineering optimization.

Overall, global-state feedback and action-conditioned parameter reuse improved the adaptive search of DE across the evaluated MOPs. Future work should examine richer state representations, dynamic environments and cooperative multi-UAV planning, and should evaluate the method on higher-dimensional objective spaces.

References

Abbass HA, Sarker RA, Newton CS (2001) PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol 2, pp 971–978. https://doi.org/10.1109/CEC.2001.934295

Al-Kaabi M, Al Hasheme J, Al-Bahrani L (2022) Improved differential evolution algorithm to solve multi-objective optimal power flow problem. Archives of Electrical Engineering 71(3):641-657. https://doi.org/10.24425/aee.2022.141676

Ali A, Abbas G, Keerio MU et al (2023) Solution of constrained mixed-integer multi-objective optimal power flow problem considering the hybrid multi-objective evolutionary algorithm. IET Generation, Transmission & Distribution 17(1):66-90. https://doi.org/10.1049/gtd2.12664

Anosri S, Panagant N, Champasak P et al (2023) A comparative study of state-of-the-art metaheuristics for solving many-objective optimization problems of fixed-wing unmanned aerial vehicle conceptual design. Archives of Computational Methods in Engineering 30(6):3657-3671. https://doi.org/10.1007/s11831-023-09914-z

Bhuyan HK, Ravi V, Yadav MS (2022) Multi-objective optimization-based privacy in data mining. Cluster Computing 25(6):4275-4287. https://doi.org/10.1007/s10586-022-03667-3

Brest J, Greiner S, Bošković B et al (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation 10(6):646-657. https://doi.org/10.1109/TEVC.2006.872133

Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, New York. https://doi.org/10.1007/978-0-387-36797-2

Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182-197. https://doi.org/10.1109/4235.996017

Falcon-Cardona JG, Coello Coello CA (2020) Indicator-based multi-objective evolutionary algorithms: a comprehensive survey. ACM Computing Surveys 53(2):1-35. https://doi.org/10.1145/3376916

Han Y, Peng H, Mei C et al (2023) Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning. Knowledge-Based Systems 277:110801. https://doi.org/10.1016/j.knosys.2023.110801

Hua Y, Liu Q, Hao K et al (2021) A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts. IEEE/CAA Journal of Automatica Sinica 8(2):303-318. https://doi.org/10.1109/JAS.2021.1003817

Huynh TN, Do DTT, Lee J (2021) Q-learning-based parameter control in differential evolution for structural optimization. Applied Soft Computing 107:107464. https://doi.org/10.1016/j.asoc.2021.107464

Li H, Zhang Q (2006) A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Parallel Problem Solving from Nature-PPSN IX. Springer, Berlin, pp 583-592. https://doi.org/10.1007/11844297_59

Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13(2):284-302. https://doi.org/10.1109/TEVC.2008.925798

Liang P, Chen Y, Sun Y et al (2024) An information entropy-driven evolutionary algorithm based on reinforcement learning for many-objective optimization. Expert Systems with Applications 238:122164. https://doi.org/10.1016/j.eswa.2023.122164

Liu Y, Lu H, Cheng S et al (2019) An adaptive online parameter control algorithm for particle swarm optimization based on reinforcement learning. In: 2019 IEEE Congress on Evolutionary Computation. IEEE, Piscataway, pp 815-822. https://doi.org/10.1109/CEC.2019.8790035

Lowe R, Wu Y, Tamar A et al (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, vol 30. Curran Associates, Red Hook, pp 6379-6390

Ma H, Shen S, Yu M et al (2019) Multi-population techniques in nature-inspired optimization algorithms: a comprehensive survey. Swarm and Evolutionary Computation 44:365-387. https://doi.org/10.1016/j.swevo.2018.04.011

Polo-Mendoza R, Martinez-Arguelles G, Penabaena-Niebles R (2023) A multi-objective optimization based on genetic algorithms for the sustainable design of warm mix asphalt (WMA). International Journal of Pavement Engineering 24(2):2074417. https://doi.org/10.1080/10298436.2022.2074417

Qu BY, Suganthan PN (2010) Multiobjective evolutionary algorithms based on the summation of normalized objectives and diversified selection. Information Sciences 180(17):3170-3181. https://doi.org/10.1016/j.ins.2010.05.013

Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4):341-359. https://doi.org/10.1023/A:1008202821328

Sun J, Liu X, Bäck T et al (2021) Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. IEEE Transactions on Evolutionary Computation 25(4):666-680. https://doi.org/10.1109/TEVC.2021.3060811

Tian Y, Cheng R, Zhang X et al (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. IEEE Computational Intelligence Magazine 12(4):73-87. https://doi.org/10.1109/MCI.2017.2742868

Tian Y, Chen H, Ma H et al (2022) Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization. IEEE/CAA Journal of Automatica Sinica 9(10):1801-1817. https://doi.org/10.1109/JAS.2022.105875

Varadarajan M, Swarup KS (2008) Solving multi-objective optimal power flow using differential evolution. IET Generation, Transmission & Distribution 2(5):720-730. https://doi.org/10.1049/iet-gtd:20070457

Wang H, Jin Y, Yao X (2017) Diversity assessment in many-objective optimization. IEEE Transactions on Cybernetics 47(6):1510-1522. https://doi.org/10.1109/TCYB.2016.2550502

Wang L, Pan X, Shen X et al (2021) Balancing convergence and diversity in resource allocation strategy for decomposition-based multi-objective evolutionary algorithm. Applied Soft Computing 101:106968. https://doi.org/10.1016/j.asoc.2020.106968

Xu X, Xie C, Luo Z et al (2024) A multi-objective evolutionary algorithm based on dimension exploration and discrepancy evolution for UAV path planning problem. Information Sciences 657:119977. https://doi.org/10.1016/j.ins.2023.119977

Xue F, Sanderson AC, Graves RJ (2003) Pareto-based multiobjective differential evolution. In: Proceedings of the 2003 Congress on Evolutionary Computation. IEEE, Piscataway, pp 862-869. https://doi.org/10.1109/CEC.2003.1299757

Yao F, Yang W, Zhang M (2010) Application of multi-objective differential evolution to load distribution in hot strip rolling. Control Theory & Applications 27(7):897-902 (in Chinese)

Yu X, Hu Z, Luo W et al (2024) Reinforcement learning-based multi-objective differential evolution algorithm for feature selection. Information Sciences 661:120185. https://doi.org/10.1016/j.ins.2024.120185

Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation 13(5):945-958. https://doi.org/10.1109/TEVC.2009.2014613

Zhang J, Xing L (2017) A survey of multiobjective evolutionary algorithms. In: 2017 IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing. IEEE, Piscataway, pp 93-100. https://doi.org/10.1109/CSE-EUC.2017.27

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6):712-731. https://doi.org/10.1109/TEVC.2007.892759

Zhang W, Peng C, Yuan Y et al (2024) A novel multi-objective evolutionary algorithm with a two-fold constraint-handling mechanism for multiple UAV path planning. Expert Systems with Applications 238:121862. https://doi.org/10.1016/j.eswa.2023.121862