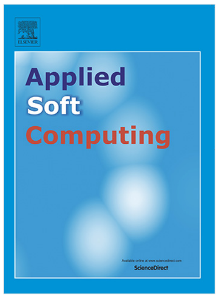
Journal Pre-proof



A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy

Ying Huang, Wei Li, Furong Tian, Xiang Meng

PII: S1568-4946(20)30631-1
DOI: https://doi.org/10.1016/j.asoc.2020.106693
Reference: ASOC 106693

To appear in: *Applied Soft Computing Journal*

Received date : 14 June 2020
Revised date : 21 August 2020
Accepted date : 28 August 2020

Please cite this article as: Y. Huang, W. Li, F. Tian et al., A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy, *Applied Soft Computing Journal* (2020), doi: https://doi.org/10.1016/j.asoc.2020.106693.

A Fitness Landscape Ruggedness Multiobjective Differential Evolution Algorithm with a Reinforcement Learning Strategy

Ying Huang[a], Wei Li[b], Furong Tian[b] and Xiang Meng[b]

[a]*Institute of Mathematical and Computer Sciences, Gannan Normal University, Ganzhou, China*
[b]*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, China*

ARTICLE INFO

Keywords:
Multiobjective
Differential Evolution
Reinforcement Learning
Fitness Landscape
Search Strategy

ABSTRACT

Optimization is the process of finding and comparing feasible solutions and adopting the best one until no better solution can be found. Because solving real-world problems often involves simulations and multiobjective optimization, the results and solutions of these problems are conceptually different from those of single-objective problems. In single-objective optimization problems, the global optimal solution is the solution that yields the optimal value of the objective function. However, for multiobjective optimization problems, the optimal solutions are Pareto-optimal solutions produced by balancing multiple objective functions. The strategic variables calculated in multiobjective problems produce different effects on the mapping imbalance and the search redundancy in the search space. Therefore, this paper proposes a fitness landscape ruggedness multiobjective differential evolution (LRMODE) algorithm with a reinforcement learning strategy. The proposed algorithm analyses the ruggedness of landscapes using information entropy to estimate whether the local landscape has a unimodal or multimodal topology and then combines the outcome with a reinforcement learning strategy to determine the optimal probability distribution of the algorithm's search strategy set. The experimental results show that this novel algorithm can ameliorate the problem of search redundancy and search-space mapping imbalances, effectively improving the convergence of the search algorithm during the optimization process.

1. Introduction

When a problem has more than one objective function that needs to be evaluated at the same time, we call it a multi-objective optimization problem (MOP)[1]. There are large differences between MOPs and single-objective optimization problems. For example, it is usually difficult to determine the objective function in the multi-objective optimization process. When the performance of one target improves, the performance of another target may diminish. Therefore, a compromise method is generally used in multiobjective algorithm solutions to make all the objectives reach the best compromise and obtain a set of approximate optimal solutions. The set of optimal solutions for multiobjective problems is called the Pareto-optimal front (PF). It is much more difficult to solve MOPs than single-objective optimization problems. The process can be visualized from two aspects. One is convergence; the solution should be as close to the real PF of the problem as possible. The other is the distribution of the obtained approximate solution set, which should be evenly distributed along the PFthat is, it should reflect the diversity of the solution distribution.

A multiobjective evolutionary algorithm (MOEA) is a type of global probability optimization heuristic search method formed by imitating the mechanism of biological evolution. MOEAs underwent rapid development in the mid-1990s. An MOEA starts with a set of initial populations and performs evolutionary operations on the population, such as selection, crossover and mutation. Through multiple generations of evolution, better individuals can be continuously obtained, and they gradually approach the Pareto boundary of MOPs. An MOEA can process a group of potential solutions in parallel and it is insensitive to the shape and continuity of the Pareto front of the problem.

Problems with more than three objective functions are called many-objective optimization problems (MaOPs)[2]. Compared with MOPs, MaOPs require more objective functions to be processed, and it is more difficult to evaluate the fitness of the individuals. Moreover, the time complexity of MaOPs increases exponentially as the number of objectives increases, making it more difficult to maintain a uniform distribution of the solution set[3].

*Corresponding author
E-mail address: liwei@jxust.edu.cn (Wei Li).

Fitness landscape theory was proposed by Wright[4] and applied to the optimization dynamics of biological evolution. The study of the fitness landscape is an important topic of evolutionary computation. Influenced by biological evolution, researchers began to study the fitness landscape early in the field of evolutionary computation, with the aim to understand how evolutionary algorithms behave and to improve the solutions of optimization problems. The fitness landscape can reveal the relationship between the search solution space and individual fitness using features of the landscape information, which regards evolutionary optimization as an adaptive random walk a process on a three-dimensional landscape that can be visualized as ridges, canyons, and basins. When the search space is regarded as a landscape, the evolutionary algorithm can be understood as a process of surveying the landscape to locate its highest peak. The fitness landscape is constructed by mapping from a set of genotypes and can be considered as indicating the "height" of the entire genotype. In other words, the fitness involved in the fitness landscape is an orthogonal projection of the value of a genotype attribute. However, the search strategy of an evolutionary algorithm is to decode the genotype into a phenotype with high fitness. The fitness landscape can be considered a useful metaphor to describe the behavior of evolutionary algorithms during the solution process of optimization problems. The fitness landscape is a visual tool that conveys geometric meaning and can be used to evaluate population characteristics as they change over time. These changes are achieved by mapping one genotype to another. Many studies and discussions of the fitness landscape topology have been published, including one in which the topology of the fitness landscape is related to the optimization problem of evolutionary computation. This concept of a problem-oriented fitness landscape has been deeply analyzed, especially for combinatorial optimization problems such as the traveling salesman problem (TSP)[5], image segmentation[6], the graph coloring problem[7], the quadratic assignment problem[8], MAX-SA[9], and the knapsack problem[10]. Although the main focus of such studies is to investigate how to provide the most effective information to solve evolutionary computation problems, the experimental results of some optimization problems also include visualizations of the features of various fitness landscape topologies. For example, in the symmetric TSP problem, the travel cost from point i to point j is the same as that from point j to point i, and the corresponding fitness landscape topology is quite smooth. In comparison, the fitness landscape presented by the asymmetric TSP problem is quite rough. For the quadratic assignment problem, experiments show that a significant neutrality degree of fitness landscape topography can be obtained. For continuous real-valued problems, similar numerical results can be obtained regarding fitness landscape topography among widely used benchmarks[11, 12].

An increasing number of researchers have turned their attention to the performance of evolutionary search algorithms. According to the research on fitness landscape methodology, the fitness landscape of an optimization problem not only presents rich feature information such as local fitness, distance correlation, and landscape ruggedness but can also reflect the topological structure of the optimal solution from different perspectives, such as optimal solution distribution, the number of solutions, and the local unimodal and multimodal topology. Although most studies of fitness landscapes are based on typical classic single-objective optimization problems, many attributes of fitness landscapes have been directly promoted in multiobjective fields[13]. Because MOPs introduce search algorithm analysis and utilization of additional features, this paper uses the fitness landscape fusion method to study the characteristics of multiobjective optimization for a given problem.

Within the scope of evolutionary algorithms, landscape ruggedness is often used to describe the frequency of changes, which appear as undulating ruggedness in local fitness landscapes. For the problem at hand, landscape ruggedness provides a measure of the number and distribution of local optimal solutions. When the fitness values corresponding to adjacent solutions in the search space are significantly different, the fitness landscape topology structure presents as a rough landscape; conversely, the opposite case results in a flat landscape similar to a basin. Therefore, landscape ruggedness is closely related to changes in the fitness values of optimization problems[14].

Reinforcement learning is a model for learning sequential decision tasks in which individual agents are reinforced to optimize their behaviors through repeated trials[15]. In some challenging cases, the actions taken by reinforcement learning agents may not only affect the immediate reward but also have continuous delayed rewards in subsequent situations. Therefore, trial and error optimization and delayed reward are considered to be the two most significant features of reinforcement learning.

Generally, the decision task of reinforcement learning can be expressed as a Markov decision process composed of a set of states, a set of behaviors, a reward function and a transfer function. In reinforcement learning, agents learn through interactions with their dynamic environment[16]. At each time step, the agents acquire the complete state information regarding their environment and take actions, thereby transforming the current environment into a new state. The agents receive a scaled reward as a signal by which they can evaluate the performance of this transformation process. This feedback mechanism requires less information than supervised learning, and the agents can take more targeted actions[17, 18]. Note that during the behavior optimization process of reinforcement learning agents, the trade-off between exploration and utilization of state-behavior strategies is highly important. In reinforcement learning, learners must use what they have learned to maximize the rewards of their current state and must also explore new behavioral strategies to select better actions in the future. To solve the learning problems of reinforcement learning in practical environments, various novel algorithms have been proposed, such as learning methods based on dynamic programming[21, 20], model-free learning methods based on online estimation functions[19], and learning methods using model-based techniques[22]. Most reinforcement learning algorithms are derived from a Q-learning model-free algorithm[17], such as those proposed in [23, 24]. The state-behavior-reward-state-behavior algorithm can be used to solve high-dimensional continuous state space problems when it knows the first few steps after the dominant behavior.

Reinforcement learning has gained widespread popularity over the past few decades, and a number of learning methods have been proposed to establish independent autonomous agents, such as direct strategy search[25], time difference[26], and Monte Carlo[27], that have achieved remarkable success in various practical applications. However, these methods often require large amounts of empirical data and consume excessive amounts of exploration time to accomplish successful learning. In terms of the efficiency aspect of problem solving, reinforcement learning sometimes underperforms, rendering it unsuitable for dealing with many complex problem areas. Therefore, domain expertise and various forms of knowledge are used in many current studies on reinforcement learning to improve learning efficiency. Decomposing tasks into a set of subtasks[28], high-level abstract behavioral learning[29, 30] and value functions abstracted in a given state space[17, 31] are included in the typical research along this direction, allowing the learning experiences of reinforcement learning agents to be promoted more effectively.

A novel algorithm known as fitness landscape ruggedness multiobjective differential evolution with a reinforcement learning strategy (LRMODE) is proposed in this paper to address MOPs. The main contributions of the proposed LRMODE are summarized as follows.

- Information entropy is used to analyze the ruggedness of landscapes to estimate the landscape topology of MOPs;
- A reinforcement learning technique is incorporated into LRMODE to determine the optimal probability distribution of the algorithm's search strategy set.
- This paper proposed a novel multiobjective differential evolution algorithm, and the optimization performances of LRMODE are evaluated with ZDT and DTLZ instances. Experimental results indicate that the proposed algorithm can effectively improve the convergence and is highly competitive.

The remainder of this paper is organized as follows. Section 2 introduces the theoretical knowledge regarding multiobjective evolutionary algorithms, fitness landscape ruggedness and reinforcement learning. Our proposed LRMODE algorithm is described in detail in Section 3. Numerical tests and experiments using the LRMODE algorithm are reported in Section 4. Finally, Section 5 provides conclusions and summarizes the directions for future research.

2. Background and Motivation

2.1. Multiobjective optimization problems

Without loss of generality, an MOP can be expressed as follows.

$$Min\ F(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \tag{1}$$

where $x = (x_1, x_2, \cdots, x_d) \in \Omega$ is a d-dimensional decision vector bounded in the decision space Ω, m is the number of objective functions and all objectives conflict with each other in general. A solution x is said to dominate another solution y, noted as $x \prec y$, if and only if $\forall\ i \in 1, 2, \cdots, M$, $f_i(x) \leq f_i(y)$ and $\exists\ j \in 1, 2, \cdots, M$, $f_i(x) < f_i(y)$. If there is no solution x dominated by any other solution y, x is considered to be the Pareto-optimal solution. The set of all Pareto-optimal solutions is called the Pareto-optimal solution set (PS). The goal of addressing the multi-objective optimization problem is to obtain a set of equally distributed Pareto dominance solutions.

2.2. Multiobjective evolutionary algorithms

Research into multiobjective evolutionary algorithms started late and is still undergoing rapid development. The earliest MOP can be traced back to 1772, when Franklin raised the problem of conflicting multiple goals. However, most scholars believe that the MOP was first proposed by the French economist V. Pareto in 1896[32], who summarized many notable problems as MOPs. In 1951, T. C. Koopmans formed an MOP for production activities and proposed the important concept of multiobjective Pareto-optimal solutions (noninferior solutions). In 1968, Z. Johnsen proposed a comprehensive definition of multiobjective problems and substantially advanced the development of MOP solutions. Overall, multiobjective optimization has been under development for almost 70 years, from V. Pareto to Z. Johnsen[33]. In 1984, David Schaffer first proposed combining evolutionary algorithms with MOPs[34]. In 1989, the Pareto theory, which originated in economics, was introduced to multiobjective evolutionary algorithms[35]. Since then, a large number of multiobjective evolutionary algorithms have been proposed, including the first generation of multiobjective evolutionary algorithms MOGA[36], NSGA[37], NPGA[38], the second-generation multiobjective evolution algorithms PAES[39], PESA[40], PESA-II[41], SPEA[42], SPEA2[43], and NSGA-II[44]. Currently, research on multiobjective optimization algorithms is in its third generation. Experts and scholars worldwide have improved the existing algorithms and proposed new algorithm frameworks, such as the multiobjective evolutionary algorithm framework MOEA/D based on decomposition technology[45], the nondominated neighborhood immune algorithm NNIA[46], a multiobjective evolution based on an adaptive differential evolution algorithm[47], and an improved decomposition-based multiobjective evolutionary algorithm[48].

MOEAs can obtain better performance and more satisfactory results when solving optimization problems with fewer objectives. However, when these algorithms are applied to high-dimensional multiobjective problems (with more than three objective functions), their performances degrade to varying degrees, and their space-time complexity increases rapidly. With the rapid development of information technology, the complexity of many problems is increasing, and a large number of high-dimensional optimization problems have emerged. The need for more intensive study of high-dimensional multiobjective evolutionary algorithms is imminent, and many scholars have already started researching many-objective evolutionary algorithms (MaOEAs) based on traditional MOEAs. How to solve high-dimensional MOPs using multiobjective evolutionary algorithms remains a difficult and popular topic in the field of optimization. The existing solutions can be divided into four categories: loose Pareto dominance (MOEA/RP), hybrid strategy, decomposition technology (MOEA/D), and full ranking strategy.

Generally, multiobjective search algorithms are based on simpler single-objective algorithms. Multiple objectives of the original problem are scalarized using specific weight vectors, and single-objective problems are solved using component search methods. Based on this premise, to improve the performance of multiobjective optimization algorithms, the multiobjective problem can be fully characterized by a series of related fitness landscapes. The characteristics of the MOPs are reflected via specific weight vectors. There are abundant solutions for landscape knowledge provided by multiobjective optimization problems, and many representation and analysis methods commonly used with single-objective problems are equally applicable to multiobjective problems. In addition, the methods of single-objective landscape analysis can be directly used to study the multi-objective algorithms, which are composed of various single-objective optimization methods. The goal of this paper is to use comprehensive solution approaches and obtain knowledge regarding fitness landscape features to guide the design of more effective multiobjective algorithms. The related works on fitness landscape representation and analysis methods are discussed in the following content. These methods can be used to visualize the landscape features of both single- and multi-objective optimization problems to obtain useful information regarding multiobjective problem instances.

2.3. Fitness landscape ruggedness

Landscape ruggedness refers to the number and distribution of local optima in a landscape. Entropy measurement is an information theory method proposed by Vassilev et al., and entropy is a measure of the uncertainty of the optimal fitness value of a local fitness landscape[49]. Samples of fitness values on the landscape are obtained by random walk time series. Then, the samples are coded into a certain symbol sequence, and the entropy is measured. The key aspect of the landscape ruggedness problem is this sample entropy; the basic idea is to use the entropy measurement to analyze the ruggedness of the landscape. Entropy was initially often used to study the smoothness, ruggedness, and neutrality of discrete landscape topologies and was later applied to real-valued problems[50]. The ruggedness features of smooth landscapes have almost the same fitness value, while in local fitness landscapes, the differences in the fitness values of neighboring individuals tend to lie along the same direction or slope. In local fitness landscapes and in rough landscapes, the fitness values of neighboring individuals change simultaneously in two directionssimilar to the land-level changes among mountain peaks and canyons. This type of rugged landscape features shapes such as mountain peaks. In a neutral landscape, the current individuals have the same fitness values as their neighbors. Reidys et al. verified that a considerable number of adjacent solutions are neutral and their discrete landscapes are also considered neutral[51]. Neutrality is a landscape feature that is often overlooked, but it has a profound impact on search algorithms and the number and distribution of local optima[52]. During the search process, when a population is evolving through a neutral fitness landscape, the fitness values do not change, which can easily be mistaken for convergence to a local optimum. In this situation, the population seems to be stagnant, but it may simply be that the population is crossing a neutral landscape area. A neutral landscape is not the same as a smooth landscape: features such as plateaus and ridges may still be present in a neutral landscape.

Introducing a time series of random walks traversing the landscape search space is the typical approach used to analyze landscape ruggedness. The fitness values $\{f_t\}_{t=0}^{n}$ corresponding to the time series represent the overall fitness error path and can be used to extract landscape topological feature information. The time series string can be expressed as $S(\varepsilon)=s_1s_2s_3\dots s_n$, with $s_i \in \{\bar{1},0,1\}$, where the combinations of $\bar{1}$s, 0s, and 1s in the set represent nine possible types of path classifications for rough, smooth, and neutral landscapes during the time series of random walks. The encoding function for the time series of a random walk is defined as follows:

$$
\begin{aligned}
s_i = \Psi_{f_i}(i,\varepsilon) &= \bar{1}, if\ f_i - f_{i-1} < -\varepsilon \\
&= 0, if\ |\ f_i - f_{i-1}\ | \leq \varepsilon \\
&= 1, if\ f_i - f_{i-1} > \varepsilon
\end{aligned}
\tag{2}
$$

The accuracy with which the string $S(\varepsilon)$ can be calculated is determined by the real number parameter ε, which can affect changes in the fitness values of neighboring individuals. Simultaneously, the sensitivity of different fitness values on the random walk time series is also determined by ε. Based on the above definition of the encoding function of the random walk time series string, the information entropy of the landscape ruggedness is defined as follows:

$$
H(\varepsilon) := -\sum_{m\neq n} P_{[mn]} log_6 P_{[mn]} \tag{3}
$$

where 6 in the mathematical expression is set as the base of the logarithm because the two rough landscapes features of neutrality (0 0) and smoothness (1 1, -1 -1) are removed in the coding classification of eight time series. $P_{[mn]}$ is defined as follows:

$$
P_{[mn]} = \frac{l_{[mn]}}{l} \tag{4}
$$

where l is the path length of the time series string $S(\varepsilon)$, $l_{[mn]}$ is the length of the subsequence mn in the time series $S(\varepsilon)$, and $P_{[mn]}$ is the relative probability that the subsequence mn appears in the time series string $S(\varepsilon)$. Therefore, $H(\varepsilon)$ is the information entropy measurement of the unequal continuous coding symbol mn.

The information entropy method described above determines the range of the local ruggedness landscape based on the random walk strategy. To ensure that the global fitness landscape ruggedness features are more widely expressed and analyzed, the landscape features are reduced to a single scalar value used to represent the ruggedness. This approach simplifies the analysis of problems with different characteristics. When the distribution of the different values is calculated and analyzed, the salient feature of each function is the maximum of $H(\varepsilon)$. The point where the maximum value of $H(\varepsilon)$ appears corresponds to the level of maximum difference in the landscape (e.g., the maximum amount of ruggedness). To represent the global ruggedness landscape features of a test function, we use the following ruggedness measurement function for the ruggedness landscape fitness function:

$$
M_f := \max_{\forall\varepsilon\in[0,\varepsilon^*]} \{H(\varepsilon)\} \tag{5}
$$

2.4. Reinforcement learning

The optimal probability distribution of the search strategy set is obtained from one or more features of the fitness landscape, which is the goal when studying hybrid search strategies. In this paper, we use reinforcement learning to design a hybrid strategy evolutionary algorithm. The hybrid reinforcement learning strategy is a Markov decision process. In evolutionary algorithms, populations are regarded as agents, and fitness landscapes can be considered their environments. Thus, strategies learned from one fitness landscape can be applied to other, similar fitness landscapes. For the instant reward feedback of reinforcement learning, we adopt the convergence speed of the considered evolutionary algorithm. Because evolutionary algorithms are random algorithms, the next state is uncertain and depends on a probability distribution. The probability that the population will move to the next-generation state and the reward value is determined by the current state and strategy, and the current strategy is determined by the current local fitness landscape feature. Generally, a decision task can be expressed as a Markov decision process consisting of a set of states, S, a set of behaviors, A, a reward function, $R(s,a)$, and a transition (transmission) function $P(s^{'}|s,a)$.

Specifically, a state space $S{=}s_1, s_2, \ldots, s_n$ and a set of behaviors $A{=}a_1, a_2, \ldots, a_n$ are given in step t. The reinforcement learning agents obtain an observation point from the current state of the environment; usually, the rewards r_t are included in this observation point. Then, the most appropriate behavior is predicted by evaluating the reward for each possible behavior in set A. Finally, after receiving the reward $R(s,a)$, the agent transitions to a new state $s^{'}$ determined by the probability distribution $P(s^{'}|s,a)$. The strategy $\pi{=}P(a|s)$ determines the distribution of each state. To determine the next action taken by the agent and pick the highest Q value for each state, this greedy strategy is deterministic. When policy π is followed, the value of state-behavior $Q^{*}(s,a)$ can be estimated from the reward obtained by (s,a), and the value of $Q^{*}(s,a)$ is calculated by the Bellman optimization equation (a dynamic programming equation), as follows:

$$Q^{*}(s,a) = R(s,a) + \gamma \sum_{s^{'}} P(s^{'}|s,a) \max_{a^{'}} Q^{*}(s^{'}, a^{'}) \tag{6}$$

where $0{<}\gamma{<}1$ is a discount factor that represents the uncertainty of the coding added to obtain a reward or to limit the boundaries that may otherwise grow indefinitely. Therefore, the goal of reinforcement learning agents is to acquire the search strategy π, and strategies are mapped by agents into appropriate behaviors to maximize the expected rewards obtained in the environment:

$$\bar{\pi}(s) = arg \max_{a} Q(s,a) \tag{7}$$

The above describes the optimal process when calculating $Q^{*}(s,a)$(that is, maximizing the expected return). The agent first calculates $Q^{*}(s,a)$ to achieve the learning goal and then selects the behavior using a greedy strategy.

In multiobjective problems, the ruggedness characteristics of adjacent landscapes can also be considered. In fact, the performance of different types of multiobjective local search algorithms is determined to some extent by the correlations between adjacent landscapes. To make the obtained optimal solution set when solving an MOP coincide with the PF as much as possible, the goals are to avoid wasting search resources on decision variables that have only small impacts on the objective function and to reduce the search redundancy. To accomplish these goals, this paper introduces a measure of the ruggedness of landscapes using information entropy. For an MOP, the quantitative representation and the analysis of the landscape ruggedness features help guide the search strategy to determine the search weight of the objective function; consequently, our approach involves finding more effective search strategies to solve MOPs.

3. Proposed Approach

From one point to another in the search space, at least one objective function improves, and at least one other one becomes worse; this relationship reflects the PF in multiobjective evolutionary algorithms[53]. None of these solutions dominates the others, and all the sets of decision variables at the PF are equal. From a multiobjective perspective, many Pareto-optimal solutions are also the global optimal of single-objective solutions. Generally, for a given convex PF, there is a weight vector for each Pareto-optimal solution, and scalarizing it means computing the global optimal solution for a single-objective problem. Therefore, the local optimal distributions have an effect on multiobjective problems, just as they do on single-objective objects. Generally, however, the different Pareto-optimal solutions are not locally optimal for the same single-objective landscape, which adds an additional consideration to the multiobjective landscape. Therefore, the Pareto-optimal distribution is a different aspect of the landscape in a sense, rather than the local optimal distribution pointing to a single point on the PF.

Because all the decision variables in most multiobjective evolutionary algorithms are treated equally, it is easy to waste search resources on decision variables that have only a small effect on the objective functions. We hope that it is possible to design a multiobjective optimization algorithm that concentrates its search efforts on the decision variables that have a large impact on the objective functions. In addition, the relationships between the decision variables and objective functions in real multiobjective problems are complex. Therefore, a multiobjective search strategy combining landscape ruggedness and reinforcement learning strategies is proposed in this paper. That is, through the ruggedness feature representation and the analysis methods of the fitness landscapes, we seek a more comprehensive representation of the relationship between the decision variables and objective functions in the search space. Then, the search weights that affect the objective function are set to achieve more effective search strategies when solving MOPs. The designed search strategy helps solve MOPs more effectively.

3.1. Landscape ruggedness strategy

Within the scope of evolutionary algorithms, landscape ruggedness is often used to describe the frequency of undulations (ruggedness) in local fitness landscapes. To solve this problem, landscape ruggedness can be thought of as a measure of the number and distribution of locally optimal solutions. The fitness landscape topology appears as a rough landscape when the fitness values corresponding to adjacent solutions in the search space are significantly different and appears as a flat landscape without features similar to a basin when the opposite is true. Therefore, landscape ruggedness is closely related to the degree of changes in the fitness values of the optimization problems.

To simplify the analysis of landscape roughness, the object of the entropy measure analysis used in this section is the set of fitness values corresponding to the three individual vectors in the search space, where the fitness values of each individual lie on the same path as one another. Whether the local topology landscape topology is neutral, smooth, or ruggedness is determined by whether the fitness values have equal error margins. Thus, if the error margin is reduced, the landscape fitness values become more sensitive to the differences, and the landscape tends to be more rugged. Similarly, when the error margin is increased, the difference between the fitness values in the landscape becomes blurred, which allows landscape topologies such as very small steps or very shallow valleys to be considered smooth or neutral. We assume that the fitness value samples are obtained from the fitness landscape using a random walk time series. Based on whether the error margin between two fitness values is the same, the feature coding is classified, and the information function is designed to estimate the entropy of the local landscape ruggedness. The information entropy function is used to analyze how the landscape ruggedness changes with the magnitude of the fitness value difference and then used to guide the search strategy of the difference algorithm and determine the optimal probability distribution in the hybrid mutation strategy, thereby effectively solving various optimization problems with constraints.

The random walk time series selection is approximately implemented as follows. First, an individual is randomly selected from the fitness landscape topology; then, all the neighborhoods of that individual are generated by mutation. One neighborhood is randomly selected and its fitness value is recorded; then, the fitness values of the new neighborhood individuals throughout the entire solution space are repeatedly selected and recorded. The discrete landscape is different from the continuous fitness landscape because random individuals do not represent all possible sets of neighborhoods. Therefore, a population-based random walk strategy is introduced whose purpose is to define walks in the fitness landscape of the search space using aggregated search paths.

For a population P=$\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_\omega\}$, where each $\vec{x}_i$=$\{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$ is a solution on $\mathfrak{R}^n$, we set the dimension D, the constraints $L_i(X) \leq 0$, i=1, …, m and $h_j(X) \leq 0$, j=1, …, p, and the number of random steps, *StepSize*, which is the number of steps given by the range d of the constraint problems:

Step 1: Generate a vector matrix walk to store the random walk steps; then, initialize *cout*=0 and after randomly selecting a starting position, perform *walk*[0] within the constraints.

Step 2: If *cout*<*StepSize*, randomly generate *N_step* in the search space whose dimension is i, satisfying *N_step*∈[0, *StepSize*] and $walk[cout]_i$=$walk[cout]_{i-1}$+*step*. If $walk[cout]_i$ is larger than the maximum constraint boundary of the search space, then $walk[cout]_i$ =$walk[cout]_{i-1}$ -*d*, and *count*=*count* +1. Following this rule, the process is continuously iterated to ensure the randomness and uniform distribution of the individuals in the search space.

Step 3: Calculate the value of the autocorrelation function $\rho(walk)$ according to Formula (1) in combination with the random walk strategy in Step 2, sort them in ascending order, and express them as a sequence, $\rho_1, \rho_2, \ldots, \rho_{step}$. After repeated experiments, we defined the correlation coefficient as a stable value between 0.5 and 1.

Step 4: According to Formulas (2), (3), and (4), calculate the entropy $H^*(\varepsilon)$ of the landscape information for the stability-related individual fitness values under the random walk strategy:

$$H^*(\varepsilon) = \frac{M_f}{\sum_{step}^{d} H(\varepsilon_\rho)} \tag{8}$$

where M_f is the maximum value of ε in the parameter set $H(\varepsilon)$, and $H(\varepsilon_\rho)$ represents the ruggedness entropy measure of the autocorrelation-sensitive parameters under the random walk strategy. The adjustment of the roughness landscape analysis method is based on the sensitive parameters of the time series of the random walk. Finally, the information entropy is obtained by calculating the relative probability of the occurrence of the time series, yielding the probability of effective information entropy.

Step 5: Normalize the processing:

$$\varphi = \frac{H^*(\varepsilon)}{\mu - 1} \tag{9}$$

where μ is the population size in the local tness landscape, φ is the simplified observation feature value of the local fitness landscape, taken as the number of optima in the fitness landscape observation, with $\varphi \in [0,1]$. According to the analysis and representation of the landscape features for the series of CEC 2005 test functions, φ being close to 0 corresponds to a unimodal local fitness landscape; on the other hand, φ being close to 1 corresponds to a multimodal local fitness landscape.

Referring to the landscape ruggedness expression and analysis methods above, if the fitness landscape changes only slightly when the weight vector is changed only slightly, then the multiobjective problem can be described as a smooth landscape. In contrast, the rough multiobjective problem occurs when small changes made to the weight vector result in large changes in the fitness landscape features. In the process of solving the optimal solution set of MOPs, the difficulty of finding other relative optimal solutions depends on the smoothness of the fitness landscape features. This concept can be expressed more intuitively as follows: smoothness means that when a relatively optimal solution exists on a specific landscape, another relatively optimal solution can be found in the nearby landscape. This spatial landscape characteristic influences the optimization algorithm to always try to find more feature information of the multiobjective problem from a relatively optimal solution.

3.2. Reinforcement learning strategy

Reinforcement learning is a Markov decision process. The following design methods can be adopted for reinforcement learning strategies in MOPs: In the multiobjective algorithm, the population is considered as an agent, and the fitness landscape is considered to be the agent's environment. The strategies learned from each fitness landscape can be applied to similar fitness landscapes. The instant reward feedback of reinforcement learning is defined as the speed of convergence of each evolutionary algorithm. Because evolutionary algorithms are random algorithms whose next state is uncertain, they depend upon a certain probability distribution. The reward value and the probability of the population moving from the current state to a next-generation state depend on the current state and strategy, and the current strategy depends on the roughness characteristics of the current local fitness landscape. In the current local fitness landscape, the specific strategy is as follows: If a faster convergence rate can be obtained when the population adopts a certain strategy, then when the population faces a similar or the same local fitness landscape in the future, the probability of using that same strategy should be increased. Conversely, if the convergence speed slows because the population adopts a certain strategy, then when the population faces that same or a similar local fitness landscape in the future, the probability of using that same strategy should be reduced.

Based on the above reinforcement learning strategy, the probability distribution of the combined fitness landscape features and search mutation strategy is updated according to each individual's fitness function and the reward-based feedback mechanism. Each individual in a subpopulation of a multiobjective problem can be classified as feasible or infeasible. The difference is that the feasible individuals belong to the PF solution set during the evolution process. An adaptive reward is obtained for a feasible individual in the reinforcement learning strategy, while infeasible individuals are appropriately punished according to the following formula:

$$\delta f_{i,l,k} = \begin{cases} 0.5 + \left(\frac{rf_{i,l,k}}{\sum_{i=1}^{m} rf_{i,l,k}}\right), \textit{i-th individual is feasible}; \\ 0.5 - \left(\frac{\frac{1}{pf_{i,l,k}}}{\frac{1}{\sum_{i=1}^{m} pf_{i,l,k}}}\right), \textit{i-th individual is infeasible}. \end{cases} \tag{10}$$

where $\delta\ f_{i,l,k}$ is the fitness value reward of the *i-th* individual generated by the *l-th* mutation strategy in the *K-th* subpopulation, $rf_{i,l,k}$ represents the fitness value of feasible individual $x_{i,k}$ before reward, $pf_{i,l,k}$ represents the fitness value of feasible individual $x_{i,k}$ before punishment, and m represents the number of individuals generated in the *i-th* mutation strategy.

3.3. Mutation search strategy

According to the characteristics of MOPs, the search strategy set of the LRMODE algorithm uses four improved mutation operators, namely, DE/rand/1/bin, DE/best/2/bin, DE/current-to-rand/1/bin, and DE/current-to-best/1/bin.

The DE/rand/1/ bin mutation strategy randomly selects three different individual vectors from a subpopulation and uses the boundary conditions of the decision space to form new individuals: it has strong randomness, a large range of individual values, a greater guarantee of population diversity, and can easily find the global optimal value. With the mutation operator DE/best/2/bin, two different randomly selected vectors are combined during the mutation process of the current optimal individual vector. The current vector is involved in this mutation strategy, and the current optimal individual vector is used as a reference standard to guide other individual vectors. Therefore, the DE/best/2/bin operator has the characteristics of small individual vector values and low population diversity and is often used to improve the accuracy and convergence speed of the algorithm. The DE/current-to-rand/1/bin and DE/current-to-best/1/bin mutation strategies no longer simply select randomly from the current search space vector; instead, they select the current population vector based on random walks and autocorrelation. These vectors not only maintain population diversity but also correlate with the fitness distance of the fitness topological structure. Therefore, the DE/current-to-rand/1/bin and DE/current-to-best/1/bin m
utation search strategies possess random diversity and do not easily fall into a local optimum when solving problems.

The mutation search operator based on the traditional differential evolution algorithm can be expressed as S={S_{Drb}, S_{Dbb}}. The mathematical expressions of S_{Drb} and S_{Dbb} are specifically defined as follows:

$$v^{g}_{i,j,k} = \begin{cases} x^{(g)}_{r1,j,k} + F \cdot |x^{(g)}_{r2,j,k} - x^{(g)}_{r3,j,k}|, & if \quad x^{(g)}_{r0,j,k} < l_j \\ x^{(g)}_{r1,j,k} - F \cdot |x^{(g)}_{r2,j,k} - x^{(g)}_{r3,j,k}|, & if \quad x^{(g)}_{r0,j,k} > u_j, \\ x^{(g)}_{r1,j,k} + F \cdot |x^{(g)}_{r2,j,k} - x^{(g)}_{r3,j,k}|, & otherwise \end{cases} \tag{11}$$

where l_j and u_j are respectively the upper and lower boundaries of each dimension of the trust space and $x^{(g)}_{r1,j,k}$,$x^{(g)}_{r2,j,k}$, and $x^{(g)}_{r3,j,k}$ are different individuals selected randomly from the Kth subpopulation.

S_{Dbb}: The DE/best/2/bin operator is used to implement mutation, which is calculated as follows:

$$v^{(g)}_{i,j,k} = x^{(g)}_{best,j,k} + F \cdot (x^{(g)}_{r1,j,k} - x^{(g)}_{r2,j,k}) + F \cdot (x^{(g)}_{r3,j,k} - x^{(g)}_{r4,j,k}) \tag{12}$$

where $x^{(g)}_{best,j,k}$ is the optimal individual vector of the Kth subpopulation in the decision space $x^{(g)}_{r1,j,k}$, $x^{(g)}_{r2,j,k}$,$x^{(g)}_{r3,j,k}$, and $x^{(g)}_{r4,j,k}$ are different vectors randomly selected from the Kth subpopulation.

First, the rugged features of the information entropy value are extracted according to the relationship between the ruggedness and the optimal value distribution in the fitness landscape. Then, the optimal probability distribution is determined from the characteristics of the rugged landscape to the algorithm search strategy set, and a differential hybrid strategy based on the rugged landscape is designed. To facilitate the visual description, the search strategy based on the fitness landscape ruggedness proposed in this paper preferentially observes the local fitness landscape. When the landscape presents or approaches a multimodal distribution, the improved DE/rand/1/bin mutation strategy S_{Drb} is used to maintain the population diversity and avoid falling into a local optimum. When the landscape presents or approaches a unimodal distribution, the improved DE/best/2/bin mutation strategy S_{Dbb} is applied to improve the convergence speed and accuracy of the algorithm.

In this way, the search mutation strategy based on the fitness landscape can be expressed as S={s_{Dcrb}, s_{Dcbb}}. The mathematical expressions of s_{Dcrb} and s_{Dcbb} are described as follows:

s_{Dcbb}: The DE/current-to-best/1/bin operator is used for mutation, and the current individuals randomly walk through the fitness landscape and acquire autocorrelation features. The *w-th* optimal individual in the search space is temporarily stored. If the population diversity decreases in a later stage of the evolution process, the temporarily stored optimal individual is used as a reference standard to prevent premature convergence. The definition of s_{Dcbb} is as follows:

$$v^{g}_{i,j,k} = \begin{cases} x^{(g)}_{i,j,k} + F \cdot |x^{(w)}_{best,j,k} - x^{(g)}_{i,j,k}| + F \cdot |x^{(g)}_{r1,j,k} - x^{(g)}_{i,j,k}|, & if \quad x^{(g)}_{i,j,k} < l_j \\ x^{(g)}_{i,j,k} - F \cdot |x^{(w)}_{best,j,k} - x^{(g)}_{i,j,k}| - F \cdot |x^{(g)}_{r1,j,k} - x^{(g)}_{i,j,k}|, & if \quad x^{(g)}_{i,j,k} > u_j, \\ x^{(g)}_{i,j,k} + F \cdot |x^{(w)}_{best,j,k} - x^{(g)}_{i,j,k}| + F \cdot |x^{(g)}_{r1,j,k} - x^{(g)}_{i,j,k}|, & otherwise \end{cases} \tag{13}$$

where l_j and u_j are respectively the upper and lower boundaries of each dimension in the trust space, $x^{(g)}_{r1,j,k}$, $x^{(g)}_{r2,j,k}$ and $x^{(g)}_{r3,j,k}$ are different individuals randomly selected from the Kth subpopulation, and $x^{(w)}_{best,j,k}$ is the *w-th* optimal individual randomly selected from the search space.

s_{Drb}: The DE/current-to-rand/1/bin operator is used to implement mutation. This mutation strategy is described as follows:

$$v^g_{i,j,k} = \begin{cases} x^{(g)}_{i,j,k} + F \cdot |x^{(g)}_{r1,j,k} - x^{(g)}_{i,j,k}| + F \cdot |x^{(g)}_{r2,j,k} - x^{(g)}_{r3,j,k}|, if \quad x^{(g)}_{i,j,k} < l_j \\ x^{(g)}_{i,j,k} - F \cdot |x^{(g)}_{r1,j,k} - x^{(g)}_{i,j,k}| - F \cdot |x^{(g)}_{r2,j,k} - x^{(g)}_{r3,j,k}|, if \quad x^{(g)}_{i,j,k} > u_j, \\ x^{(g)}_{i,j,k} + F \cdot |x^{(g)}_{r1,j,k} - x^{(g)}_{i,j,k}| + F \cdot |x^{(g)}_{r2,j,k} - x^{(g)}_{r3,j,k}|, otherwise \end{cases} \tag{14}$$

where l_j and u_j are respectively the upper and lower boundaries of each dimension of the search space and $x^{(g)}_{r1,j,k}$, $x^{(g)}_{r2,j,k}$ and $x^{(g)}_{r3,j,k}$ are different individuals randomly selected from the Kth subpopulation.

3.4. The LRMODE algorithm

The specific operation of the LRMODE algorithm is as follows:

Step 1: The set of mutation search strategies can be expressed as S_m={s_{Drb},s_{Dbb},s_{Dcrb},s_{Dcbb}} , where m represents the *i-th* mutation strategy. In addition, S_i (i =1, 2., ⋯, m) is also used to indicate the probability of each mutation strategy being selected. The average probability of the search mutation operator β_l is calculated as follows:

$$\beta_l = \frac{\sum_{i=1}^{m} S_m}{m} \tag{15}$$

Step 2: The probability ξ_l of using the operator in the hybrid mutation strategy is calculated as follows:

$$\xi_l = \frac{\beta_l}{\sum_{i=1}^{m} \beta_i} \tag{16}$$

Step 3: The search mutation strategy probability distribution is generated as follows:

$$\pi = \beta_i^{1/S_{FL}(i)} \tag{17}$$

where $S_{FL}(i)$ refers to the fitness landscape ruggedness strategy, $S_{FL}(i) = \{\varpi_{FDC}, \varpi_{r_{FD}}, \varpi_{\rho}, \varpi_{H^*(\varepsilon)}, \dots\}$.

Step 4: Update the optimal probability distribution of the search mutation strategy in the combined fitness landscape using the reward and punishment mechanism in the reinforcement learning strategy.

Step 5: Complete the reorganization, crossover, and selection stages according to the standard differential evolution algorithm.

Algorithm 1 :The general framework of the proposed LRMODE

```
population ← Initialization: NP, F, CR and Maximum generation G_max.
Calculate the fitness value, fit(i)=F(X_i) // i = 1,2, ⋯, M
while termination condition is not met do
    Perform Landscape ruggedness strategy;
    Perform Mixed search strategy;
    Perform Reinforcement Learning strategy;
    for i ← 1toNP do
        Perform Mutation operator;
        Perform Crossover operator;
        Perform Selection operator;
    end for
end while
Output the best nondominated solutions
```

At this point, the multi-objective optimization algorithm (LRMODE) based on the combined fitness landscape theory is complete. This algorithm first extracts the fitness value of the correlation features of the multiobjective problem and then classifies it based on the value of the random walk autocorrelation function. Finally, the landscape ruggedness information entropy is used to quantify the feature analysis and form a combined landscape strategy. To determine

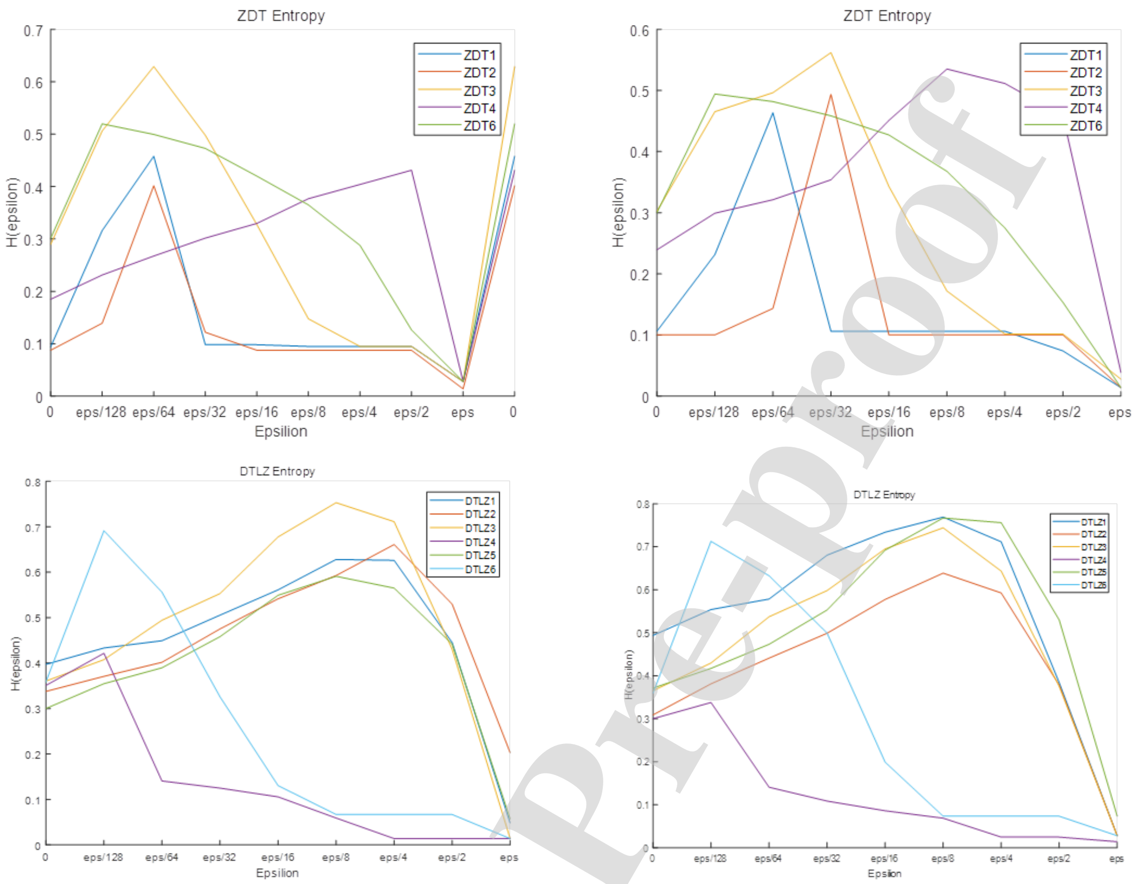


Figure 1: $H(\varepsilon)$ of various ε on ZDT and DTLZ functions.

the probability distribution of the multiobjective landscape features mapped to the algorithm search strategy set, the reinforcement learning strategy is introduced. Thus, a multiobjective differential evolution algorithm combining the fitness landscape and the hybrid search strategy is designed. Algorithm 1 presents the general framework of LRMODE.

4. Experimental Study

To verify the performance of the multiobjective differential evolution algorithm based on the integration of fitness landscape ruggedness and reinforcement learning strategies, this section evaluates the performance of the classic multi-objective optimization algorithm through experiments. The test function selects a set of two-objective problems that has been widely adopted in the multi-objective field: the ZDT[54] and DTLZ[44] series of test functions. The ZDT series was compiled by Zitzler et al. in 2000. The ZDT series includes five problems, ZDT1 - ZDT4 and ZDT6. In these widely used multi-objective test problems, the Pareto fronts are characteristically bumpy, with multiple peaks. The experimental parameters of the functions were reported in [44].

4.1. Representation and analysis of ruggedness landscape

To determine the potential relationship between the characteristics of rough landscapes and the algorithm search strategy, we analyzed the ZDT[54] and DTLZ[44] series using the fitness landscape ruggedness characteristics. First, the entropy value of the landscape ruggedness information is calculated. Then, the relationship between the topological fitness features and the optimal solution is further analyzed. During the initialization, the random walk was set to 10,000 steps, and 30 independent random walks were performed for each test function. These random walk steps were used

Table 1
M_f for 10-Dimensional Functions.

ZDT	M_f
ZDT1	0.46
ZDT2	0.40
ZDT3	0.63
ZDT4	0.43
ZDT6	0.52

Table 2
M_f for 30-Dimensional Functions.

ZDT	M_f
ZDT1	0.46
ZDT2	0.49
ZDT3	0.56
ZDT4	0.54
ZDT6	0.49

Table 3
M_f for 2-Objectives Functions.

ZDT	M_f
ZDT1	0.63
ZDT2	0.66
ZDT3	0.75
ZDT4	0.42
ZDT5	0.59
ZDT6	0.69

Table 4
M_f for 3-Objectives Functions.

ZDT	M_f
ZDT1	0.77
ZDT2	0.64
ZDT3	0.74
ZDT4	0.34
ZDT5	0.77
ZDT6	0.71

to calculate the information stability measure (ε^*) of each test function. The symbol ε^* denotes the minimum value of all ε and is used as the upper limit of the sensitive range. Below this value, landscape features become smooth. In each random walk, $H(\varepsilon)$ is calculated to increase the value of ε. The value of $H(\varepsilon)$ is critical for characterizing the ruggedness of the fitness landscape. To compare the test functions across a greater variety of different range fitness values, in this section, based on the specified test function, we defined nine parameter values (0, $\varepsilon^*/128$, $\varepsilon^*/64$, $\varepsilon^*/32$, $\varepsilon^*/16$, $\varepsilon^*/8$, $\varepsilon^*/4$, $\varepsilon^*/2$ and ε^*). For these 9 parameter values, the selected $\bar{H}(\varepsilon)$ is the average of 30 random walks.

Figure 1 shows an illustration of the change in information entropy obtained on the test functions of the ZDT and DTLZ series with different parameters. The left and right sides of the figure show the effects of the entropy trajectory of the rough landscape information in different dimensions for the random walk time series. The two results of the 10 and 30 dimensions of the function ZDT and the individual dimensions of the function DTLZ are calculated according to the different values of the target number K. Based on fitness landscape ruggedness theory, the changes in information entropy reflect the degree of ruggedness of the local fitness landscape. Thus, the information entropy value can be used to guide the algorithm to adopt a suitable search strategy. The results of the information entropy analysis of these multiobjective functions are similar to the results of a single-objective problem. The $H(\varepsilon)$ value is in the range [0,1] and serves as an estimate of the fitness landscape topological features in the search space based on a random walk. Through information entropy analysis, the local landscape ruggedness characteristics are obtained. After parameter adjustment by trial and error, we found that the closer $\bar{H}(\varepsilon)$ is to 1, the rougher the landscape of the test function is. This conclusion helps in designing the landscape strategy for the optimization problem. For multiobjective problems, representing and analyzing the landscape ruggedness information entropy can be used to help design search strategies. The landscape features are reduced to a single scalar value that represents the ruggednessthat is, the maximum value of the information entropy. As shown in Tables 1 to 4 below, these quantitative results represent the maximum difference level in the landscape and the global ruggedness of the landscape features. This knowledge is expected to aid further analysis of problems with different features.

The above analysis method using information entropy depends on the local ruggedness landscape based on a random walk strategy. Through this analysis, the ruggedness features of the global fitness landscape are reduced to and expressed as a single scalar value that represents the ruggedness. The ruggedness characteristics of neighborhood landscapes can also be considered in multiobjective problems. In fact, the performances of different types of multiobjective local search algorithms are partly determined by the correlations between adjacent landscapes. To obtain a solution set as close as possible to the optimal solution set or to coincide with the PF when solving MOPs, we should avoid wasting search resources on decision variables that have little impact on the objective functions, and redundant searches should be reduced. Quantitative analysis and representation of landscape ruggedness features can be used to guide search strategies in MOPs to determine the search weights of the objective functions. Using this approach, more effective search strategies can be designed when solving MOPs.

4.2. Comparative analysis of algorithms

We compared the experimental results of our proposed LRMODE algorithm with those of five classic algorithms, namely, NSGA-II[54], SPEA2[42], GDE3[54], MOEA/D[45] and DBEA[55]. To more comprehensively evaluate the convergence of the LRMODE algorithm and the diversity and uniformity of the solution space distribution, we adopted the generational distance (GD)[56], hypervolume (HV)[42] and the inverted generational distance (IGD) proposed by Deb et al. and Zitzler et al. as metrics for the performance analysis[44, 57].

This section focuses on comparing the performance results of the LRMODE algorithm with those of classic multi-objective optimization algorithms on the ZDT and DTLZ series of test functions. The ZDT results are listed in Tables 5-9, and the DTLZ results are listed in Tables 10-14. The IGD, GD and HV performance indicators are calculated for each test function. For each indicator, we adopt four measurements: best, worst, mean and standard deviation (Sed). The LRMODE algorithm proposed in this chapter was applied to the 10 multiobjective test functions and compared with the results of five classic algorithms: NSGA-II, SPEA2, GDE3, MOEA/D and DBEA. As Tables 7, 8, 9 and 14 show, on ZDT3, ZDT4, ZDT6 and DTLZ7, the result obtained by the LRMODE algorithm for the HV indicator is larger than those obtained by the other algorithms. This indicates that the convergence and solution distribution of the LRMODE algorithm are generally better. Tables 5 and 10 show that on ZDT1 and DTLZ1, the LRMODE algorithm result for GD is smaller than those of the other algorithms. Table 13 shows that on DTLZ2, the LRMODE algorithm

Table 5
Computational result of comparing algorithms on ZDT1.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	4.47E-03	3.86E-03	3.68E-03	8.00E-03	3.87E-03	**3.88E-03**
	Worst	5.03E-03	4.04E-03	**3.75E-03**	1.61E-02	3.91E-03	4.37E-03
	Mean	4.71E-03	3.94E-03	3.72E-03	1.11E-02	3.88E-03	**3.71E-03**
	Std.	1.52E-04	5.00E-05	1.90E-05	2.05E-03	**8.00E-06**	1.09E-04
GD	Best	1.07E-04	1.01E-04	8.20E-05	6.96E-04	**3.80E-05**	4.20E-05
	Worst	2.65E-04	2.15E-04	1.74E-04	1.57E-03	1.04E-04	**5.80E-05**
	Mean	1.81E-04	1.73E-04	1.15E-04	1.05E-03	5.60E-05	**4.40E-05**
	Std.	4.70E-05	2.20E-05	2.70E-05	2.34E-04	1.80E-05	**3.00E-06**
HV	Best	6.61E-01	6.62E-01	6.62E-01	6.53E-01	6.62E-01	**6.62E-01**
	Worst	6.60E-01	6.61E-01	**6.62E-01**	6.40E-01	6.61E-01	6.61E-01
	Mean	6.60E-01	6.62E-01	6.62E-01	6.48E-01	6.61E-01	**6.62E-01**
	Std.	**2.37E-04**	1.18E-04	1.70E-05	3.26E-03	5.70E-05	1.47E-04

Table 6
Computational result of comparing algorithms on ZDT2.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	4.45E-03	3.84E-03	3.80E-03	7.35E-03	3.81E-03	**3.80E-03**
	Worst	5.01E-03	4.00E-03	3.91E-03	1.90E-01	**3.82E-03**	4.02E-03
	Mean	4.69E-03	3.91E-03	3.85E-03	2.44E-02	**3.81E-03**	3.83E-03
	Std.	1.35E-04	4.40E-05	2.60E-05	3.83E-02	**3.00E-06**	5.70E-05
GD	Best	8.00E-05	4.70E-05	4.20E-05	5.29E-04	4.60E-05	**4.20E-05**
	Worst	1.90E-04	1.43E-04	**5.20E-05**	4.86E-03	5.50E-05	1.37E-04
	Mean	1.25E-04	7.80E-05	5.60E-05	1.74E-03	5.00E-05	**5.00E-05**
	Std.	2.70E-05	2.50E-05	3.00E-06	9.32E-04	**2.00E-06**	2.00E-05
HV	Best	3.28E-01	3.29E-01	3.29E-01	3.20E-01	3.28E-01	**3.28E-01**
	Worst	**3.27E-01**	3.28E-01	3.29E-01	1.54E-01	3.28E-01	3.28E-01
	Mean	3.28E-01	3.28E-01	3.29E-01	3.00E-01	3.28E-01	**3.28E-01**
	Std.	1.63E-04	8.60E-05	2.10E-05	3.42E-02	3.40E-05	**9.10E-05**

Table 7
Computational result of comparing algorithms on ZDT3.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	3.08E-03	2.94E-03	**2.67E-03**	7.21E-03	6.44E-03	3.61E-03
	Worst	3.59E-03	3.29E-03	**2.79E-03**	3.54E-02	6.61E-03	3.97E-03
	Mean	3.32E-03	3.11E-03	**2.72E-03**	1.06E-02	6.54E-03	3.75E-03
	Std.	1.27E-04	9.20E-05	**3.50E-05**	5.82E-03	4.40E-05	1.06E-04
GD	Best	1.97E-04	1.96E-04	1.72E-04	6.28E-04	2.14E-04	**1.71E-04**
	Worst	2.25E-04	2.37E-04	**2.07E-04**	1.31E-03	2.42E-04	2.18E-04
	Mean	2.11E-04	2.17E-04	1.91E-04	9.34E-04	2.25E-04	**1.91E-04**
	Std.	1.00E-05	1.00E-05	8.00E-06	1.55E-04	**7.00E-06**	1.30E-05
HV	Best	5.16E-01	5.16E-01	5.16E-01	5.08E-01	5.14E-01	**5.16E-01**
	Worst	5.15E-01	5.16E-01	5.16E-01	4.96E-01	5.14E-01	**5.16E-01**
	Mean	5.15E-01	5.16E-01	5.16E-01	5.02E-01	5.14E-01	**5.16E-01**
	Std.	1.00E-04	5.00E-05	1.30E-05	2.79E-03	5.60E-05	**1.68E-04**

Table 8
Computational result of comparing algorithms on ZDT4.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	2.55E-01	1.25E-01	**3.69E-03**	2.56E-01	2.55E-01	3.88E-03
	Worst	1.27E+00	1.27E+00	**3.78E-03**	1.43E+00	3.07E+00	3.89E-01
	Mean	4.88E-01	6.38E-01	**3.74E-03**	8.39E-01	1.43E+00	8.55E-02
	Std.	2.14E-01	2.88E-01	**2.20E-05**	3.52E-01	8.04E-01	1.22E-01
GD	Best	2.59E-02	1.27E-02	8.30E-05	2.49E-02	2.49E-02	**4.10E-05**
	Worst	1.41E-01	1.47E-01	**1.67E-04**	1.54E-01	3.14E-01	3.76E-02
	Mean	5.18E-02	7.16E-02	1.28E-04	8.67E-02	1.43E-01	**8.11E-03**
	Std.	2.41E-02	3.39E-02	**2.70E-05**	3.86E-02	8.31E-02	1.20E-02
HV	Best	3.40E-01	4.95E-01	6.62E-01	3.39E-01	3.41E-01	**6.62E-01**
	Worst	0.00E+00	0.00E+00	6.62E-01	0.00E+00	0.00E+00	**2.09E-01**
	Mean	1.65E-01	1.13E-01	6.62E-01	6.27E-02	3.34E-02	**5.58E-01**
	Std.	9.69E-02	1.40E-01	1.70E-05	9.23E-02	8.43E-02	**1.48E-01**

Table 9
Computational result of comparing algorithms on ZDT6.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	3.44E-03	2.95E-03	2.95E-03	5.10E-02	**2.60E-03**	2.98E-03
	Worst	4.82E-03	3.73E-03	3.47E-03	1.23E-01	**2.64E-03**	3.11E-03
	Mean	3.99E-03	3.32E-03	3.24E-03	9.00E-02	**2.62E-03**	3.04E-03
	Std.	3.58E-04	1.79E-04	**1.24E-04**	1.75E-02	8.00E-06	3.70E-05
GD	Best	**4.97E-04**	5.06E-04	5.08E-04	5.78E-03	5.25E-04	5.26E-04
	Worst	5.91E-04	5.63E-04	5.69E-04	1.30E-02	5.37E-04	**5.35E-04**
	Mean	5.47E-04	5.37E-04	5.36E-04	8.94E-03	5.33E-04	**5.32E-04**
	Std.	2.10E-05	1.80E-05	1.70E-05	1.87E-03	**3.00E-06**	5.00E-06
HV	Best	4.00E-01	4.01E-01	4.01E-01	3.34E-01	4.01E-01	**4.01E-01**
	Worst	3.99E-01	4.00E-01	4.01E-01	2.75E-01	4.01E-01	**4.01E-01**
	Mean	4.00E-01	4.00E-01	4.01E-01	3.05E-01	4.01E-01	**4.01E-01**
	Std.	1.81E-04	1.20E-04	1.70E-05	1.55E-02	1.00E-06	**2.21E-04**

results for both IGD and HV are smaller than those of the other algorithms. Finally, Table 14 shows that on DTLZ4, the LRMODE algorithm result for HV is smaller than those of the other algorithms. We can conclude that overall, the LRMODE algorithm outperforms the other five algorithms. The IGD values of LRMODE on the test functions ZDT3, ZDT4, ZDT6, DTLZ3, DTLZ4, and DTLZ7 are all larger than those of the other algorithms, which shows that the LRMODE algorithm does not have good convergence performance; however, its solution distribution is still better

Table 10
Computational result of comparing algorithms on DTLZ1.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	4.11E-03	3.73E-03	2.81E-03	7.66E-03	3.61E-03	**3.59E-03**
	Worst	4.84E-03	4.52E-03	4.64E-03	1.55E+00	**3.61E-03**	3.91E-03
	Mean	4.49E-03	3.76E-03	3.74E-03	4.27E-01	3.61E-03	**3.60E-03**
	Std.	1.81E-04	4.75E-04	4.72E-04	4.91E-01	**1.00E-06**	9.40E-05
GD	Best	3.79E-04	3.60E-04	3.60E-04	1.12E-03	4.16E-04	**3.36E-04**
	Worst	4.65E-04	1.59E+00	4.90E-04	7.83E-01	4.17E-04	**4.16E-04**
	Mean	4.23E-04	1.07E-01	4.24E-04	3.01E-01	4.26E-04	**4.16E-04**
	Std.	2.40E-05	3.98E-01	3.60E-05	2.66E-01	**0.00E+00**	9.00E-06
HV	Best	4.95E-01	4.95E-01	4.95E-01	4.87E-01	4.95E-01	**4.95E-01**
	Worst	4.92E-01	4.94E-01	4.95E-01	0.00E+00	4.95E-01	**4.95E-01**
	Mean	4.93E-01	4.95E-01	4.95E-01	2.38E-01	4.95E-01	**4.95E-01**
	Std.	4.75E-04	2.38E-04	3.00E-05	1.98E-01	**0.00E+00**	2.28E-04

Table 11
Computational result of comparing algorithms on DTLZ2.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	4.08E-03	3.82E-03	3.70E-03	4.11E-03	4.13E-03	**3.60E-03**
	Worst	5.17E-03	4.27E-03	4.13E-03	4.46E-03	4.16E-03	**4.12E-03**
	Mean	4.69E-03	4.04E-03	4.93E-03	4.20E-03	4.15E-03	**4.14E-03**
	Std.	3.01E-04	1.62E-04	1.34E-04	8.90E-05	**1.00E-05**	6.30E-05
GD	Best	8.92E-04	1.04E-03	**9.51E-04**	1.16E-03	1.21E-03	1.21E-03
	Worst	1.10E-03	1.11E-03	1.02E-03	1.23E-03	1.22E-03	**1.02E-03**
	Mean	1.01E-03	1.07E-03	**9.78E-04**	1.21E-03	1.21E-03	1.21E-03
	Std.	5.30E-05	1.90E-05	1.80E-05	1.40E-05	3.00E-06	**2.00E-06**
HV	Best	2.10E-01	2.10E-01	2.11E-01	2.10E-01	2.10E-01	2.10E-01
	Worst	2.09E-01	2.10E-01	2.11E-01	2.09E-01	2.10E-01	**2.11E-01**
	Mean	2.09E-01	2.10E-01	2.10E-01	2.10E-01	2.10E-01	**2.10E-01**
	Std.	1.80E-04	7.40E-05	2.60E-05	1.82E-04	1.00E-06	**8.00E-05**

Table 12
Computational result of comparing algorithms on DTLZ3.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	4.41E-03	3.81E-03	**3.39E-03**	9.35E+00	4.20E-03	4.25E-03
	Worst	6.36E-03	6.13E-03	**4.37E-03**	5.65E+01	4.69E-03	6.61E-03
	Mean	5.02E-03	4.60E-03	**3.80E-03**	2.27E+01	4.30E-03	4.88E-03
	Std.	5.36E-04	5.91E-04	2.22E-04	9.49E+00	**1.17E-04**	6.97E-04
GD	Best	**8.43E-04**	9.35E-04	9.40E-04	2.95E+00	1.19E-03	1.13E-03
	Worst	1.09E-03	1.12E-03	**1.02E-03**	2.29E+01	1.26E-03	1.30E-03
	Mean	9.92E-04	1.05E-03	**9.84E-04**	1.17E+01	1.21E-03	1.22E-03
	Std.	6.30E-05	5.00E-05	1.80E-05	4.23E+00	**1.60E-05**	3.90E-05
HV	Best	2.10E-01	2.10E-01	2.11E-01	0.00E+00	2.10E-01	**2.19E-01**
	Worst	2.04E-01	2.05E-01	**2.11E-01**	0.00E+00	2.09E-01	2.03E-01
	Mean	2.08E-01	2.08E-01	**2.11E-01**	0.00E+00	2.10E-01	2.07E-01
	Std.	1.33E-03	1.36E-03	2.10E-05	0.00E+00	2.59E-04	**1.71E-03**

than those of the comparison algorithms. Therefore, the multiobjective differential evolution algorithm based on the combination of fitness landscape and hybrid search strategy proposed in this paper improves the solutions for the ZDT and DTLZ problem sets, especially in terms of solution convergence, search space mapping distribution and reduction in search redundancy.

Table 13
Computational result of comparing algorithms on DTLZ4.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	4.10E-03	3.76E-03	**3.49E-03**	4.13E-03	4.14E-03	4.12E-03
	Worst	7.54E-01	7.54E-01	**4.08E-03**	7.54E-01	7.54E-01	7.54E-01
	Mean	1.55E-01	5.41E-02	**3.74E-03**	1.17E-01	5.42E-02	1.54E-01
	Std.	3.00E-01	1.87E-01	**1.68E-04**	2.68E-01	1.87E-01	3.00E-01
GD	Best	3.00E-06	3.00E-06	9.61E-04	3.00E-06	**3.00E-06**	3.00E-06
	Worst	1.08E-03	1.10E-03	**1.02E-03**	1.24E-03	1.21E-03	1.22E-03
	Mean	**7.92E-04**	9.84E-04	9.88E-04	1.03E-03	1.13E-03	9.72E-04
	Std.	4.00E-04	2.64E-04	**1.60E-05**	4.30E-04	3.02E-04	4.84E-04
HV	Best	2.10E-01	2.10E-01	2.11E-01	2.10E-01	2.10E-01	**2.10E-01**
	Worst	0.00E+00	0.00E+00	2.11E-01	0.00E+00	0.00E+00	**0.00E+00**
	Mean	1.68E-01	1.96E-01	**2.11E-01**	1.78E-01	1.96E-01	1.68E-01
	Std.	8.38E-02	5.24E-02	2.10E-05	7.50E-02	5.24E-02	**8.40E-02**

Table 14
Computational result of comparing algorithms on DTLZ7.

Indicators		NSGAII	SPEA2	GDE3	DBEA	MOEA/D	LRMODE
IGD	Best	3.31E-03	3.09E-03	**2.90E-03**	6.23E-03	4.76E-03	3.46E-03
	Worst	4.30E-03	3.82E-03	**3.22E-03**	1.03E-02	3.73E-01	3.72E-01
	Mean	3.80E-03	3.43E-03	**3.09E-03**	8.02E-03	2.94E-02	7.77E-02
	Std.	2.32E-04	1.79E-04	**1.10E-04**	1.12E-03	9.17E-02	1.47E-01
GD	Best	4.74E-04	4.89E-04	4.76E-04	7.70E-04	**2.90E-04**	2.92E-04
	Worst	7.83E-04	5.78E-04	5.03E-04	1.29E-03	5.41E-04	**5.02E-04**
	Mean	5.37E-04	5.18E-04	4.90E-04	9.42E-04	5.20E-04	**4.90E-04**
	Std.	9.60E-05	2.10E-05	**7.00E-06**	1.46E-04	6.20E-05	1.02E-04
HV	Best	3.30E-01	3.30E-01	3.30E-01	3.24E-01	3.29E-01	**3.30E-01**
	Worst	3.29E-01	3.30E-01	3.30E-01	3.17E-01	2.17E-01	**3.37E-01**
	Mean	3.29E-01	3.30E-01	3.30E-01	3.21E-01	3.21E-01	**3.31E-01**
	Std.	1.13E-04	6.20E-05	1.20E-05	1.83E-03	2.78E-02	**4.50E-02**

5. Conclusion

A multiobjective differential evolution algorithm (LRMODE) that combines landscape roughness and a reinforcement learning strategy is proposed in this paper. In this algorithm, the unimodal or multimodal topology of the local landscape is evaluated with a landscape fitness value corresponding to the information entropy of landscape ruggedness. The optimal probability distribution of the algorithm search strategy set is determined by integrating reinforcement learning. This search strategy set is then used to guide the search weights of search strategies and to design and implement a multiobjective differential evolution algorithm that combines landscape ruggedness and reinforcement learning strategies. The experimental data analysis results show that the LRMODE algorithm is better than other algorithms at solving MOPs involving search redundancy and an imbalanced search space. Finally, the optimization process effectively improves LRMODE's convergence performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ying Huang: Conceptualization, Writing - Original Draft, **Wei Li:** Methodology, Writing - Review, Editing. **Furong Tian:** Writing - Review, Editing. **Xiang Meng:** Software, Data Curation.

Acknowledge

This work was supported by the National Natural Science Foundation of China (Grant No. 61903089), the Natural Science Foundation of Jiangxi Province of China (Grant Nos. 20202BABL202020 and 20202BAB202014) and the Science Foundation of Jiangxi University of Science and Technology (Grant No. JXXJBS18059).

References

[1] Zheng, J.H.. Multi-objective evolutionary algorithm and its application. Beijing: Science Press 2007;12–38.
[2] Gong, M.G., Liu, C., Jiao, L.C., Cheng, G.. Hybrid immune algorithm with lamarckian local search for multi-objective optimization. Memetic Computing 2010;2(1):47–67.
[3] Islam, M.A., Anderson, D.T., Petry, F., Elmore, P.. An efficient evolutionary algorithm to optimize the Choquet integral. International Journal of Intelligent Systems 2019;34(3):366–385.
[4] Wright, S.. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. 1932;1.
[5] Stadler, P.F., Schnabl, W.. The landscape of the traveling salesman problem. Physics Letters A 1992;161(4):337–344.
[6] Wang, F., Zhu, H.Q.,Li, W.,Li, K.S.. A hybrid convolution network for serial number recognition on banknotes. Information Sciences 2020;512:952–963.
[7] Bouziri, H., Mellouli, K., Talbi, E.. The k-coloring fitness landscape. Journal of Combinatorial Optimization 2011;21(3):306–329.
[8] Daolio, F., Verel, S., Ochoa, G., Tomassini, M.. Local optima networks of the quadratic assignment problem. IEEE Congress on Evolutionary Computation 2010;1–8.
[9] Prugel-Bennett, A., Tayarani-Najaran, M.. Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimization problem. IEEE transactions on evolutionary computation 2011;16(3):319–338.
[10] Tavares, J., Pereira, F.B., Costa, E.. Multidimensional knapsack problem: A fitness landscape analysis. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 2008;38(3):604–616.
[11] CaamañO, P., Bellas, F., Becerra, J.A., Duro, R.J.. Evolutionary algorithm characterization in real parameter optimization problems. Applied Soft Computing 2013;13(4):1902–1921.
[12] Pérez-Cañedo, B., Verdegay, J. L., Miranda Pérez, R.. An epsilon-constraint method for fully fuzzy multiobjective linear programming. International Journal of Intelligent Systems 2020;35(4):600–624.
[13] Mohamadghasemi, A., Hadi-Vencheh, A., Hosseinzadeh Lotfi, F.. The multiobjective stochastic CRITIC–TOPSIS approach for solving the shipboard crane selection problem. International Journal of Intelligent Systems 2020.
[14] Devarapalli, R., Bhattacharyya, B., Sinha, N.K.. An intelligent EGWO-SCA-CS algorithm for PSS parameter tuning under system uncertainties. International Journal of Intelligent Systems 2020.
[15] Sutton, R.S.. Reinforcement learning: Past, present and future. Asia-Pacific Conference on Simulated Evolution and Learning 1998;195–197.
[16] Peng, X.D., Liu, L.. Information measures for q-rung orthopair fuzzy sets. International Journal of Intelligent Systems 2019;34(8):1795–1834.
[17] Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.. Learning from data. 2012;4.
[18] Jiang, N., Xu, D., Zhou, J., Yan, H.Y., Wan, T., Zheng, J.Q.. Toward optimal participant decisions with voting-based incentive model for crowd sensing. Information Sciences 2020;512:1–17.
[19] Wang, Feng and Li, Yixuan and Zhou, Aimin and Tang, Ke Wang, F.,Li, Y.X.,Zhou, A.M.,Tang, K.. An estimation of distribution algorithm for mixed-variable newsvendor problems. IEEE Transactions on Evolutionary Computation 2019.
[20] Fei, L.G., Deng, Y.. A new divergence measure for basic probability assignment and its applications in extremely uncertain environments. International Journal of Intelligent Systems 2019;34(4):584–600.
[21] Huang, Y., Li, W.,Ouyang, C.T., Chen, Y.. A self-feedback strategy differential evolution with fitness landscape analysis. Soft Computing 2018;22(23):7773–7785.
[22] Moore, R.R., Cohn, A.K.. Method and apparatus for tying suture to bone. 1993;US Patent 5,250,055.
[23] Greenwald, G., Nosek, B.A., Banaji, M.R.. Understanding and using the implicit association test: I. an improved scoring algorithm. Journal of personality and social psychology 2003;85(2):197.
[24] Lauer, M., Riedmiller, M.. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In Proceedings of the Seventeenth International Conference on Machine Learning 2000.
[25] Wang, F., Zhang, H.,Li, K.S.,Lin, Z.Y.,Yang, J., Shen, L.. A hybrid particle swarm optimization algorithm using adaptive learning strategy. Information Sciences 2018;436:162–177.
[26] Li, W.,meng, X.,Huang, Y.,Fu, Z.H.. Multipopulation Cooperative Particle Swarm Optimization with A Mixed Mutation Strategy. Information Sciences 2020;529:179–196.
[27] Guestrin, C., Lagoudakis, M., Parr, R.. Coordinated reinforcement learning. ICML 2002;2:227–234.
[28] Fitch, W.T., Hauser, M.D., Chomsky, N.. The evolution of the language faculty: clarifications and implications. Cognition 2005;97(2):179–210.
[29] Busoniu, L., Schutter, B.D., Babuska, R.. Multiagent reinforcement learning with adaptive state focus. BNAIC 2005;35–42.
[30] Jiang, N., Chen, J., Zhou, R.G., Wu, C.X., Chen, H.L., Zheng, J.Q., et al. Pan: Pipeline assisted neural networks model for data-to-text generation in social internet of things. Information Sciences 2020;.
[31] Garg, H., Rani, D.. Some results on information measures for complex intuitionistic fuzzy sets. International Journal of Intelligent Systems 2019;34(10):2319–2363.
[32] Wang, F.,Li, Y.X.,Zhang, H.,Hu, T.,Shen, X.L.. An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. Swarm and Evolutionary Computation 2019;49:220–233.
[33] Jian-wei, Y.. A summary of researches on multi-objective evolutionary algorithm. Natural Science Journal of Hainan University 2005;4.

[34] Schaffer, J.D.. Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of the first international conference on genetic algorithms and their applications 1985.
[35] Goldberg, D.E., Korb, B., Deb, K.. Messy genetic algorithms: Motivation, analysis, and first results. Complex systems 1989;3(5):493–530.
[36] Murata, T., Ishibuchi, H.. Moga: multi-objective genetic algorithms. IEEE international conference on evolutionary computation 1995;1:289–294.
[37] Srinivas, N., Deb, K.. Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary computation 1994;2(3):221–248.
[38] Horn, J., Nafpliotis, N., Goldberg, D.E.. A niched pareto genetic algorithm for multiobjective optimization. IEEE world congress on computational intelligence 1994;82–87.
[39] Knowles, J., Corne, D.. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 1999;1:98–105.
[40] Corne, D.W., Knowles, J.D., Oates, M.J.. The pareto envelope-based selection algorithm for multiobjective optimization. International conference on parallel problem solving from nature 2000;839–848.
[41] Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.. PESA-II: Region-based selection in evolutionary multiobjective optimization. Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation 2001;283–290.
[42] Zitzler, E., Thiele, L.. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE transactions on Evolutionary Computation 1999;3(4):257–271.
[43] Zitzler, E., Laumanns, M., Thiele, L.. Spea2: Improving the strength pareto evolutionary algorithm. TIK-report 2001;103.
[44] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 2002;6(2):182–197.
[45] Zhang, Q.F., Li, H.. Meyarivan, moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on evolutionary computation 2007;11(6):712–731.
[46] Gong, M.G., Jiao, L.C., Du, H.F., Bo, L.F.. Multiobjective immune algorithm with nondominated neighbor-based selection. Evolutionary computation 2008;16(2):225–255.
[47] Bi, X.J., Xiao, J.. Classification-based self-adaptive differential evolution with fast and reliable convergence performance. Soft Computing 2011;15(8):1581–1599.
[48] Wang, F., Zhang, H., Li, Y.X., Zhao, Y.Y., Rao, Q.. External archive matching strategy for MOEA/D. Soft Computing 2018;22(23):7833–7846.
[49] Vassilev, V.K., Fogarty, T.C., Miller, J.F.. Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application. Advances in evolutionary computing 2003;3–44.
[50] Liu, P., Liu, W.Q.. Multiple-attribute group decision-making method of linguistic q-rung orthopair fuzzy power Muirhead mean operators based on entropy weight. International Journal of Intelligent Systems 2019;34(8):1755–1794.
[51] Reidys, C.M., Stadler, P.F.. Neutrality in fitness landscapes. Applied Mathematics and Computation 2001;117(2-3):321–350.
[52] Borenstein, Y., Poli, R.. Fitness distributions and ga hardness. International Conference on Parallel Problem Solving from Nature 2004;11–20.
[53] Yee, A.K., Ray, A.K., Rangaiah, G.P.. Multiobjective optimization of an industrial styrene reactor. Computers & Chemical Engineering 2003;27(1):111–130.
[54] Zitzler, E., Deb, K., Thiele, L.. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation 2000;8(2):173–195.
[55] Asafuddoula, M., Ray, T., Sarker, R.. A decomposition-based evolutionary algorithm for many objective optimization. IEEE Transactions on Evolutionary Computation 2014;19(3):445–460.
[56] Tang, L., Wang, H., Li, G., Xu, F.X.. Adaptive heuristic search algorithm for discrete variables based multi-objective optimization. Structural and Multidisciplinary Optimization 2013;48(4):821–836.
[57] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da, F., Viviane, G.. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on evolutionary computation 2003;7(2):117–132.

Highlights

- A fitness landscape ruggedness multi-objective differential evolution (LRMODE) algorithm with a reinforcement learning strategy is proposed.
- Analyses the ruggedness of landscapes using information entropy to estimate whether the local landscape has a unimodal or multimodal topology.
- Combines fitness landscape ruggedness with a reinforcement learning strategy to determine the optimal probability distribution of the algorithm's search strategy set.
- The novel proposed LRMODE can ameliorate the problem of search redundancy and search-space mapping imbalances effectively improving the convergence of the search algorithm during the optimization process.

Ying Huang: Conceptualization, Writing - Original Draft.

Wei Li: Methodology, Writing - Review & Editing.

Furong Tian: Writing - Review & Editing.

Xiang Meng: Software, Data Curation.

Ying Huang, Wei Li, Furong Tian and Xiang Meng
Email: liwei@jxust.edu.cn

Corresponding author: Wei Li at School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

***Declaration of Interest Statement**

Declaration of competing interest

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: