Journal of Computers Vol. 36 No. 1, February 2025, pp. 59-74
https://doi.org/10.63367/199115992025023601005

A Multi-objective Particle Swarm Optimization using Logistics-tent Chaotic Map with GOBL and Non-inertial Lévy Flight

Yu Lai[1], Lanlan Kang[2,1*], and Wenliang Cao[3]

[1] School of Information Engineering, Jiangxi University of Science and Technology,
Ganzhou 341000, China
lyckyt@163.com

[2] School of Information Engineering, Gannan University of Science and Technology,
Ganzhou 341000, China
victorykll@163.com

[3] School of Electronic Information, Dongguan Polytechnic,
Dongguan, Guangdong 523808, China
caowl22@163.com

Received 21 April 2024; Revised 8 June 2024; Accepted 23 June 2024

Abstract. To address the slow learning of traditional PSO algorithms and alleviate population collapse in the objective space, this paper proposes a multi-objective particle swarm optimization algorithm using a Logistic-tent chaotic map with GOBL and non-inertial Lévy flight (MOPSOLGN). Firstly, the Logistic-tent chaotic map with generalized opposition-based learning initializes individual positions, avoiding blindness and uncertainty in the initial population and improving its distribution. Secondly, a particle flight method combining an individual competition mechanism with k-means clustering divides particles into losers and winners, using a new non-inertial Lévy flight dynamical equation to balance exploration and exploitation and ensure the algorithm can escape local solutions. Thirdly, a differential mutation strategy enables the population to escape collapse and increases the diversity of the optimal solution set. Comparative experiments with state-of-the-art multi-objective algorithms on benchmark functions verify that MOPSOLGN allows individuals to converge to the real Pareto frontier more quickly and with better distribution.

Keywords: evolutionary algorithm, multi-objective optimization, non-inertial Lévy flight dynamical equation, logistics-tent chaotic map

1 Introduction

Multi-objective optimization problems (MOPs) [1-2] involve simultaneously optimizing multiple conflicting objectives, requiring complex decision-making to find the best trade-offs. Unlike single-objective optimization, improving one objective in MOPs often compromises others, making solutions challenging. The trade-off solution set is called the Pareto optimal solution set or Pareto frontier (PF). Due to the NP-hard [3] nature of many practical engineering optimization problems, traditional mathematical programming methods are often infeasible. Evolutionary computation, a population-based heuristic search method that simulates natural selection and evolutionary processes, is well-suited for MOPs due to its random search strategy and applicability. Researchers have proposed various multi-objective evolutionary algorithms (MOEAs) in recent decades, broadly categorized into three types:

(1) MOEA based on domination relation: Identifies non-dominated individuals and eliminates dominated ones using a Pareto-based fitness allocation strategy. Examples include NSGA-II [4], SPEA2 [5], and PESA-II [6].

(2) MOEA based on decomposition: Aggregates sub-objectives into a single target, transforming the problem

* Corresponding Author

into a single-objective optimization. Examples include MOEA/D [7], MOEA/D-M2M [8], and RVEA [9].

(3) Metric-based MOEA: Uses evaluation metrics to assess algorithm quality and guide search and solution selection. Examples include IBEA [10], SMS-EMOA [11], and HypE [12].

Above MOEAs still have the deficiencies of insufficient convergence ability and low computational efficiency, and hence some new evolutionary strategies and mechanisms are constantly proposed or introduced to improve them. For example, Wang et al. [13] introduced a multi-attribute elite individual game mechanism to enhance convergence and diversity. Wang et al. [14] propose a dual-search mode MOEA with an adaptively switching strategy analyzing the correlation between the objective optimization direction and constraint satisfaction direction is designed to determine whether to build the constraint surrogate models to assist the current evolutionary search. Song et al. [15] developed a cooperative evolutionary algorithm with a dual-population approach to balance convergence and feasibility. Lian et al. [22] proposes the Human Evolutionary Optimization Algorithm (HEOA). HEOA divides the global search process into two distinct phases: human exploration and human development. Logistic chaos mapping is employed for initialization. In the human exploration phase, an initial global search is conducted, followed by the human development phase, in which the population is categorized into leaders, explorers, followers, and losers, each utilizing its own distinct search strategy. Additionally, Differential Evolution (DE) has been applied to multimodal MOEA with an improved crowding distance [17]. PSO is used to solve different multi-objective application problems, such as reduce localization error in Wireless Sensor Network [18], continuum robot's developed modeling and control [19], Optimizing Lithium-Ion Battery Modeling [20]. Firefly algorithm (FA) is involved in solving the MOPs based on multiply cooperative strategies (MOFA-MCS) [21]. Despite these advancements, achieving a balance between convergence and diversity in MOEAs remains challenging.

Therefore, building on the comprehensive analysis of prior research, this paper proposes a MOPSOLGN algorithm, which combines dominance relationships, evaluation metrics, and decomposition principles. The innovation mainly includes two aspects:

(1) Logistics-tent chaotic map with GOBL strategy (LTCMGOBL): This paper proposes LTCMGOBL for population initialization, using decomposition and k-means clustering to divide the population into sub-populations. Within each sub-population, individuals are classified as losers or winners through pairwise competition based on dominance relationships and crowding distance mechanisms.

(2) Non-inertial Lévy flight dynamical equation: A new non-inertial Lévy flight dynamical equation guides the flight adjustment of individuals who fail in LTCMGOBL competition. It initially directs towards the winners and Leeroy particles, employing a novel Levy flight mode influenced by the average position vector of all Leeroy particles.

2 Related Works and Motivation

This section provides background knowledge, including the mathematical definition of MOPs and three related works: PSO algorithm, Logistic-tent chaotic map, and Lévy Flight. These were thoroughly investigated to address the slow learning of traditional PSO algorithms and alleviate population collapse in the objective space. The motivation for the algorithm design stems from analyzing these issues' impacts on MOPs.

2.1 MOPs and MOEA

Without loss of generality, a MOP with n decision variables and m objective functions, taking the minimization problem as an example, can be expressed in the form of equation (1).

$$\begin{cases} \min y = F(x) = (f_1(x), f_2(x), ..., f_n(x)) \ . \\ x = (x_1, x_2, ..., x_m) \in X \subset R^m \ . \\ y = (y_1, y_2, ..., y_n) \in Y \subset R^n \ . \end{cases} \tag{1}$$

In Eq.(1), x is called the decision vector, and X is the m- dimensional decision space, y is called the objective vector, and Y is the n- dimensional objective space. The objective function F defines the mapping function and n objectives that need to be optimized simultaneously. If X is a connected and closed region in the space R^m, and the objective function f_i (i =1, 2, ..., n) about x is continuous.

The complexity of MOPs lies in the absence of a single solution that can optimize all objectives. Instead, a Pareto optimal set balancing all objectives is sought, with a preference for better convergence and diversity. However, many practical engineering problems have irregular PFs. Based on existing research, the MOPSOLGN algorithm employs various strategies to maintain high diversity and convergence in the population.

2.2 Particle Swarm Optimization

PSO, a widely used evolutionary algorithm (EA), is applied to solve many complex optimization problems, including MOPs [2, 16]. In PSO, each individual in the population, represented as a 'particle' without mass or volume, moves in a D-dimensional objective space, where each particle represents a feasible solution. To find the optimal solution for a minimization problem, the positions and velocities of the particles are updated using the formulas in Eq. (2)-(3), as illustrated in Fig. 1.

$$V_i(t+1)=\omega V_i(t)+c_1r_1(t)(p_i(t)-X_i(t)) \\ +c_2r_2(t)(g(t)-X_i(t)) \ . \tag{2}$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \ . \tag{3}$$

Where, ω is the inertia factor, c_1,c_2 are respectively individual learning factors and group learning factors, p_i denotes the current optimal position, and g denotes the global optimal position , r_1,r_2 represent two random factors between 0 and 1.

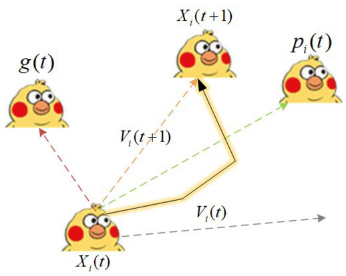

Fig. 1. Illustration of particle motion

PSO algorithms have a strong biological and social background and require fewer parameters, making them adaptable to various types of objective functions and constraints. However, they also have some limitations and drawbacks. In the early stages of evolution, particles are widely distributed, theoretically offering high diversity. Yet, inappropriate strategies for updating velocity and position can quickly lead to a loss of diversity, affecting the balance between exploration and exploitation in later stages of the algorithm. In the later stages of evolution, as particles tend to cluster around high-quality solutions, the algorithm may face a decrease in the diversity of the solution set, limiting its ability to generate a broad set of Pareto-optimal solutions.

In subsequent research, this paper attempts to propose a new strategy from the perspective of maintaining diversity while improving convergence speed, aiming to better avoid the aforementioned drawbacks.

2.3 Logistics-tent Chaotic Map

The Logistic-tent chaotic map [26, 27] is generated by integrating the classical one-dimensional Logistic map and Tent chaotic map. The chaotic map combines the complex chaotic dynamics characteristics of Logistic with the faster iteration speed, more autocorrelation and the characteristics of Tent chaotic map for a large number of sequences. Its mathematical formula is defined as follows:

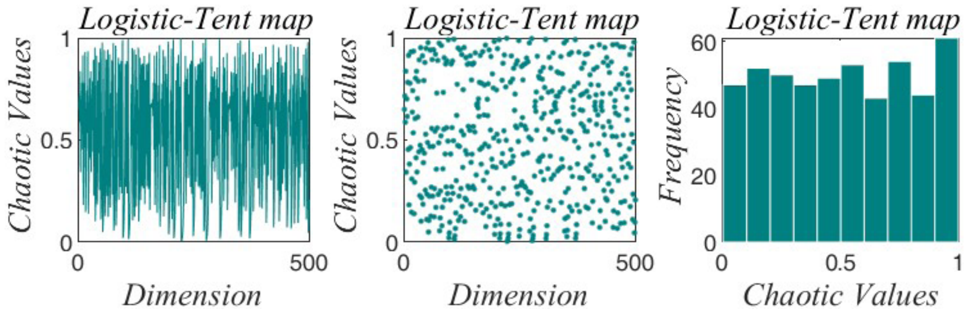

Fig. 2. Logistics-tent chaotic map distribution traversal graph

The mathematical formula for Logistics-tent chaotic map is expressed as in equation (4):

$$X_{i+1} = \begin{cases} \left(rX_i\left(1-X_i\right)+(4-r)X_i/2\right) \bmod 1, X_i < 0.5 \ . \\ \left(rx_i\left(1-X_i\right)+(4-r)\left(1-X_i\right)/2\right) \bmod 1, X_i \geqslant 0.5 \ . \end{cases} \quad r \in \left(0,4\right) \ . \tag{4}$$

As can be seen from Fig. 2 and formula (4), it can show exhibit dynamic behavior, including periodic orbit, chaotic orbit and bifurcation phenomena. Additionally, the chaotic sequence it generates is uniformly distributed in the interval [0,1]. Many practical experiences have demonstrated that using a chaotic map, instead of a conventional uniformly distributed random number generator, can generate better random individuals than a pseudo-random number generator. This gives the chaotic map significant advantages in solving complex MOPs. Therefore, this paper takes advantage of the Logistic-tent chaotic map for population initialization.

2.4 Lévy Flights

Lévy flights [23, 24, 28] are a type of non-Gaussian random process, commonly used to stepwise describe human travel patterns, the foraging trajectories of organisms, and other continuous random stepwise trajectories in mathematical form. They are also among the best strategies in random walk models. Lévy flight trajectories are a kind of Markov random process, where the step lengths follow a heavy-tailed Lévy distribution, as shown in equation (5).

$$L(s) \sim |s|^{-1-\gamma}, 0 < \gamma \leq 2 \ . \tag{5}$$

The random step s of Lévy flights can be obtained from equation (6):

$$s = \frac{a}{|b|^{1/\gamma}}. \tag{6}$$

where, a, b both obey normal distribution, $a \sim N(0,\sigma_a^2)$, $b \sim N(0,\sigma_b^2)$, σ_a and σ_b, satisfy equation (7):

$$\sigma_a = \left[\frac{\Gamma(1+\gamma)\sin(\pi\frac{\gamma}{2})}{\Gamma(\frac{1+\gamma}{2})\gamma * 2^{(\gamma-1)/2}}\right]^{1/\gamma}, \sigma_b = 1. \tag{7}$$

where Γ is the standard Gamma function, by adjusting the γ value, a trade-off can be made between global and local search, resulting in better search performance in the optimization algorithm.

In the MOPSOLGN algorithm, Lévy flight perturbations enhance the search process by enabling both local and global exploration. Small-area local searches near the current optimal solution accelerate convergence, while random long-distance movements allow other particles to explore farther regions, enhancing global search capability and avoiding local optima. Lévy flights introduce random movements with large jumps and sharp directional changes, expanding the search range and increasing population diversity. This approach effectively addresses the challenge of approximating the global Pareto front (PF) in multi-objective optimization problems with multiple local PFs.

3 Proposed MOPSOLGN Algorithm

The main framework of the MOPSOLGN algorithm is outlined in this section. The algorithm comprises four key components: (1) LTCMGOBL for population initialization, (2)-(3) CSS and a non-inertial Lévy flight dynamical equation for guiding populations to explore uncharted spaces, (4) A differential mutation strategy introduced for avoid aggregating of populations in regions that are not globally optimal.

3.1 Logistic-tent Chaotic Map with GOBL Strategies (LTCMGOBL)

The use of chaotic maps for generating random number sequences offers superior randomness and uniformity. Thus, in optimization algorithms, particularly during the population initialization phase, substituting the conventional uniform distribution random number generator with chaos mapping can significantly enhance the initial population quality. This, in turn, boosts the algorithm's search efficiency and solution quality.

This paper enhances the Logistic-tent chaotic map with a generalized opposition-based learning initialization strategy, yielding notable improvements over the traditional opposition-based learning approach. Let's delve into the generating opposition-based learning strategy [30]:

For initial solutions, a corresponding reverse solution is generated for each one as follows:

$$OP_i = K(X_{\min}^d + X_{\max}^d) - X_i \ . \tag{8}$$

Where OP_i is the opposite individual of X_i , K is a random value, and $K \in U(0,1)$, $X_{\min}^d$, $X_{\max}^d$ are the lower and upper boundary values of the D-dimensional search space, respectively.

In this paper, I have imposed restrictions on the opposition process, because the generation method of OP_i might result in the existence of negative solutions. When OP_i is less than 0, I have implemented certain constraints.

$$OP_i = -OP_i * rand; \quad if \ \mathrm{OP}_i < 0 \ . \tag{9}$$

Where $rand \in U(0,1)$.

The primary steps of the LTCMGOBL strategies are outlined in Algorithm 1.

```
Algorithm 1. LTCMGOBL
Output: initial population NP;
1. Initialize population P using Logistic-Tent chaotic map;
2. Generate opposition population OP of P;
3.   If OP < 0 then
4.      modify the OP ;
5.   end
6. Elite select half individual from {P ∪ OP} as NP;
```

3.2 A competitive Selection Mechanism within Subpopulations (CSS)

Multi-population evolutionary strategies can effectively accelerate convergence speed, but they also increase the risk of the population getting trapped in local optima. To address the issue of particles potentially congregating in non-global optimal regions, this paper first combines the non-dominance relationship with the optimal particle selection strategy in PSO, and employs the K-means clustering algorithm to divide them into subpopulations (the clustering effect is shown in Fig. 3). The competition mechanism is defined by comparing the non-dominance level and crowding distance between two individuals, with individuals within each subpopulation competing in pairs. Ultimately, the losers are guided by the winners and Leeroy. The number of subpopulations is determined by Equation (10).

$$subpop\ k_i = \arg\min_j \left\| x_i - \mu_j \right\|^2 ,\ (i = 1,2,\cdots,N)(j = 1,2,...,k)\ . \tag{10}$$

Where μ_i is the cluster centroids and is the arithmetic mean of the coordinates of all points within that cluster, μ_1, μ_2, ..., $\mu_K \in R^n$.

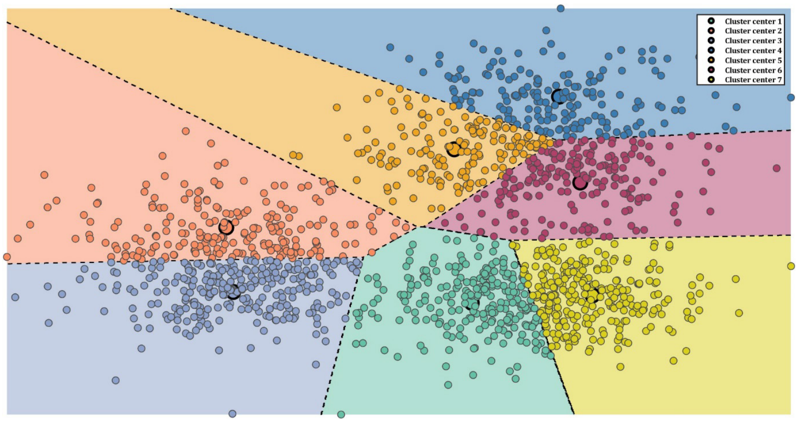

Fig. 3. K-means clustering effect figure

The main steps of CSS are list in Algorithm 2.

Algorithm 2. CSS

Input: initial population *NP*, the number of clusters *K*;
Output: Leeroy individual *Le*; loser and winner individuals X_l, X_w ;
1. Cluster center initialization with k-means++ algorithm;
2. For *i* = 1 to *K* do
3. Pairwise competition in the *i* subpopulation to obtain loser and winner individuals;
4. Randomly select Le_i in the highest non-dominated rank sequence of the current subpopulation;
5. End

3.3 Non-inertial Lévy Flight Dynamical Equation

In PSO, due to the tendency of particles to cluster around high-quality solutions, the algorithm may encounter a decrease in the diversity of the solution set, affecting the balance between subsequent exploration and exploitation. Therefore, developing a new dynamical equation is crucial, allowing for steady evolution while ensuring diversity. To this end, three innovative strategies are proposed. First, using the difference between the mean swarm positions instead of the initial velocity improves convergence rate and precision of the solution, only the flight direction of losers identified by CSS is updated, effectively increasing population diversity. Second, losers are guided by two types of particles: the winners of the competition and a Leeroy particle, ensuring smooth progression of population optimization. Third, a Leeroy Lévy flight mechanism is devised to adjust the flight direction of particles, reducing the probability of the population falling into local optima. The formula for this adjustment is given in equation (11)-(15):

$$V_{l,i}(t+1) = s\cdot(u_k(t)-u_k(t-1)) + c_1(X_{w,i}(t)-X_{l,i}(t)) + c_2(Le_{k,i}(t)-X_{l,i}(t))\ . \tag{11}$$

$$X_{l,i}(t+1) = X_{l,i}(t) + L_{k,i}(t+1) + V_{l,i}(t+1)\ . \tag{12}$$

Where, the parameters involved are described as follows:

(1). $V_{l,i}(t)$ denote the velocity vectors of the loser, $X_{w,i}(t)$ represents the position vector of the winner. $s\in(0.1,0.2)$ denotes a differential coefficient, $u_k(t)$ and $u_k(t-1)$ are the mean swarm positions of the tth and $(t-1)$th generation in the k^{th} subpopulation. k is the clustering group index, $Le_{k,i}(t)$ denotes the Leeroy particle.

(2). $L_{k,i}(t+1)$ denotes the Leeroy Lévy flight in the k^{th} subpopulation. It is determined according to equation (13).

$$L_k(t+1) = \begin{cases} L^{(X_{l,i}(t)-\overline{Le_k(t)})} & \text{if } rand > j\ . \\ rand\cdot(X_{r1}^k - X_{r2}^k) & \text{else}\ . \end{cases} \tag{13}$$

Where L represents walk steps of Lévy flight, and $r1$, $r2$ are two random integers. X_{r1}^k and X_{r2}^k denote two random particles in the k^{th} subpopulation. $\overline{P_k(t)}$ is the average position vector of all Leeroy particles. $j = 1-i_k / NP$, where i_k is the number of losers in the k^{th} subpopulation, $rand\in(0,1)$.

(3). Parameters c_1,c_2 retain the same definitions as in equation (2). Adaptive parameters are far better for flight than static parameters, so all parameters adopt adaptive strategies. The formulas are expressed as in equations (14).

$$\begin{cases} c_1 = 1 + \dfrac{it_{\max} - it - 1}{it_{\max}}\ . \\ c_2 = 1 + \dfrac{it+1}{it_{\max}}\ . \end{cases} \tag{14}$$

c_1 and c_2 represent "own experience" and "social experience", respectively. c_1 gradually decreases as it increases, and c_2 behaves oppositely. This analysis suggests that a larger c_1 favors local searching guided by own experience, is more beneficial in the early stage of the algorithm's evolution. Conversely, a larger c_2 favoring global search guided by social experience, is more advantageous in the later stage.

3.4 Differential Mutation

Mutation is a widely used technique to avoid algorithms becoming trapped in local optima. In this context, the paper integrates differential mutation into the proposed algorithm to bolster its capability for global search, drawing inspiration from the classical Differential Evolution (DE) strategy as discussed in references [25, 29]. The process is encapsulated in equation (15).

$$V_i^t = X_{r1}^t + F\cdot(X_{r2}^t - X_{r3}^t) + F\cdot(X_{r4}^t - X_{r5}^t)\ . \tag{15}$$

$$U_{i,j}^{t} = \begin{cases} V_{i,j}^{t} & if \quad rand \le CR \;\; or \;\; i = j_{rand} \; . \\ X_{i,j}^{t} & others \; . \end{cases} \tag{16}$$

Where $F\in[0,2]$ is the scaling factor, $i = \{1,2,..., N_p\}$, N_p is the population size, V_i^t is the mutant vector, and $r1,r2,r3,r4,r5\in\{1,2,..., N_p\}\backslash\{i\}$ are randomly generated, $r1\neq r2\neq r3\neq r4\neq r5$, CR is randomly chosen from the set $\{1,2,..., N\}$, which guarantees that U_i^t has at least one component from V_i^t.

3.5 Pseudo-code for the MOPSOLGN

Algorithm 3 (MOPSOLGN) combines the four strategies of 3.1-3.4 effectively, and the main steps of the MOPSOLGN algorithm are listed below.

```
Algorithm 3. MOPSOLGN
Input:
    A MOP and a stopping criterion;
    Population size N;
    K-means clustered subgroups K;
    Boundary of X and V;
Output:
    Final solution set;
1. LTCMGOBL;
2. Compute the objective function values;
3. While stopping criterion is not satisfied do
4.     CSS;
5.     For i = 1:K do
6.      Update V^k(t) and X^k(t) of particles according to non-in-
ertial Lévy flight dynamical equation;
7.         non-dominated sort NP refer to reference [4];
8.         differential mutation NP generates MP;
9.        combine population (NP + MP) to CP and repeat step 2;
10.        elitism selects population CP to restore the initial
population size N;
11.     End
12. End
```

4 Experimental Study

This section is divided into three parts: The first part details the experimental setup, including benchmark test functions, main parameter settings, and performance metrics. The second part presents the experimental results and comparative analysis of six algorithms, including MOPSOLGN, across seven benchmark functions. The third part analyzes the validity of the three main strategies of MOPSOLGN. All experiments were conducted using MATLAB 2023a.

4.1 Experiment Settings

The experimental study uses two types of MOP test functions, ZDT [31] and DTLZ [32], to verify the algorithm's performance. Table 1 summarizes these functions' characteristics, including the number of decision variables, objectives, and real PF samples.

Two performance metrics, IGD [33] and SP [33], are adopted to evaluate the algorithms' effectiveness. IGD (inverted generational distance) is a comprehensive index measuring both diversity and convergence of the population, defined as follows in equation (17):

$$IGD(P^*, P) = \frac{\sum_{x^* \in P^*} \min_{x \in P} d(x^*, x)}{|P^*|} . \quad \textbf{(17)}$$

Equation (17) defines IGD, which calculates the average distance from each reference point on the Pareto approximate frontier (P^*) to the nearest solution in the solution set (P). Here, $d(x^*,x)$ represents the Euclidean distance between points x^* and x, and $|P^*|$ is the number of points in the set P^*. A smaller IGD value indicates better algorithm performance. SP, defined in equation (18), stands for Spatial Indicator and measures the population's diversity.

$$SP(P) = \sqrt{\frac{1}{|P|-1} \sum_{i=1}^{|P|} (\bar{d} - d_i)^2} . \quad \textbf{(18)}$$

In SP, d_i calculates the distance between the two closest solutions using the Manhattan distance between x_i and x_j , not the Euclidean distance. It refers to the standard deviation of the minimum distance from each solution to other solutions, serving as an important indicator of the spacing between adjacent solutions. A smaller SP value indicates a more uniform solution set.

Table 1. Test function properties

Test function	Decision variables	Objectives
ZDT1		
ZDT2	30	2
ZDT3		
ZDT4	10	2
ZDT6		
DTLZ2	12	3
DTLZ7	22	3

4.2 Experiment Comparison and Results Analysis

The performance of MOPSOLGN was compared with six advanced MOEAs: MOEA/D, MOPSOCD, MOPSO, NSGA-II, SMPSO, and SPEA2. Each algorithm was run independently 30 times across seven benchmark functions (Table 1), with 100 generations per run. All algorithms except MOPSOLGN used the standard parameter settings from the MOEAs platform PlatEMO [34], while MOPSOLGN's specific configurations are detailed in Table 2.

Table 2. Parameter settings of MOPSOLGN

Algorithm	Parameter settings
MOPSOLGN	NP = 300, gen = 100, P_c = 0.7, P_m = 1/NP, n1 = 2, n 2= 5, k = 7

Table 3 and Table 4 present the average IGD and SP values (with standard deviations in parentheses) for the algorithms, highlighting superior results in bold. MOPSOLGN outperformed the others on the ZDT1, ZDT2, ZDT4, ZDT6, and DTLZ7 benchmarks, achieving the lowest IGD values. Table 4 shows that MOPSOLGN demonstrated better dispersion, indicating a more uniform distribution of non-dominated solutions along the PF. However, MOPSOLGN does not perform well on ZDT3 and DTLZ2, but its overall performance is still better.

Table 3. Mean/ (standard deviation) of IGD metrics on multi-objective test functions

Func.	MOPSOLGN	MOEA/D	MOPSOCD	MOPSO	SMPSO	SPEA2
ZDT1	**1.42e-3 (3.78e-5)**	1.5355e-2 (1.18e-2)	1.4305e-3 (2.70e-5)	1.5427e-1 (5.35e-2)	1.8003e-3 (2.46e-4)	3.0110e-3 (3.28e-4)
ZDT2	**1.41e-3 (4.12e-5)**	5.4705e-2 (8.45e-2)	1.4465e-3 (2.56e-5)	7.4951e-1 (4.07e-1)	1.6469e-3 (4.83e-5)	3.3329e-3 (5.15e-4)
ZDT3	3.78e-3 (4.33e-4)	2.7080e-2 (2.39e-2)	**1.8178e-3 (1.04e-4)**	1.6388e-1 (5.22e-2)	3.4831e-3 (4.45e-3)	2.6396e-3 (2.77e-4)
ZDT4	**2.13e-3 (6.42e-4)**	2.5171e-2 (1.64e-2)	8.9246e+0 (4.01e+0)	8.0752e+0 (4.87e+0)	3.0705e+0 (1.57e+0)	9.3013e-3 (8.90e-3)
ZDT6	**1.07e-3 (3.44e-5)**	6.4863e-3 (2.19e-3)	1.0991e-3 (1.91e-5)	2.3840e-3 (4.10e-4)	1.2962e-3 (5.28e-5)	7.5006e-3 (4.84e-3)
DTLZ2	3.64e-2 (2.78e-4)	**2.9067e-2 (1.50e-4)**	5.1975e-2 (2.55e-3)	3.7010e-2 (7.44e-4)	4.2036e-2 (1.39e-3)	3.0959e-2 (2.51e-4)
DTLZ7	**3.52e-2 (2.34e-3)**	1.2984e-1 (1.82e-1)	4.4327e-2 (1.85e-3)	4.8773e-1 (1.48e-1)	1.9211e-1 (1.35e-1)	3.5579e-2 (8.46e-4)

Table 4. Mean/ (standard deviation) of SP metrics on multi-objective test functions

Func.	MOPSOLGN	MOEA/D	MOPSOCD	MOPSO	SMPSO	SPEA2
ZDT1	**1.26e-3 (2.30e-5)**	3.4391e-3 (1.35e-3)	1.6256e-3 (7.23e-5)	3.5983e-3 (3.50e-4)	2.1309e-3 (1.60e-4)	1.3555e-3 (7.74e-5)
ZDT2	**1.37e-3 (1.56e-5)**	6.3868e-3 (4.43e-3)	1.6436-3 (7.03e-5)	3.6047e-3 (3.08e-4)	2.1839e-3 (1.27e-4)	1.9452e-3 (5.38e-4)
ZDT3	5.87e-3 (6.28e-4)	8.6641e-3 (2.43e-3)	2.4406e-3 (1.77e-4)	5.2423e-3 (1.38e-3)	3.3319e-3 (1.65e-3)	**1.5848e-3 (1.34e-4)**
ZDT4	2.64e-3 (7.33e-4)	3.9818e-3 (1.38e-3)	2.5216e-1 (3.83e-1)	**1.8521e-3 (4.45e-3)**	4.4978e-2 (5.04e-2)	3.8863e-3 (4.51e-3)
ZDT6	**1.76e-3 (1.39e-4)**	2.4619e-3 (6.81e-4)	1.0439e-1 (1.08e-1)	2.2391e-2 (2.50e-2)	2.4312e-2 (7.43e-2)	2.7423e-3 (1.25e-3)
DTLZ2	2.78e-2 (5.41e-4)	2.8618e-2 (3.96e-4)	3.1479e-2 (1.57e-3)	3.0480e-2 (1.20e-3)	3.0019e-2 (3.14e-3)	**1.3167e-2 (7.00e-4)**
DTLZ7	**1.42e-2 (2.76e-4)**	1.0042e-1 (2.13e-2)	3.8190e-2 (3.05e-3)	2.4902e-2 (7.83e-3)	3.1482e-2 (1.07e-2)	1.6209e-2 (1.01e-3)

To furtherly detail the convergence and distribution of the nondominated solution sets obtained by the comparison algorithms, Fig. 4 to Fig. 8 respectively display the nondominated solution sets and PFs when the 6 algorithms solve the ZDT1, ZDT2, ZDT4, ZDT6, and DTLZ7 benchmark functions.

Fig. 4 and Fig. 5 show that the MOPSOLGN and MOPSOCD algorithms fit the PF most closely. Table 3 and Table 4 indicate that MOPSOCD's optimization performance is slightly inferior to MOPSOLGN.

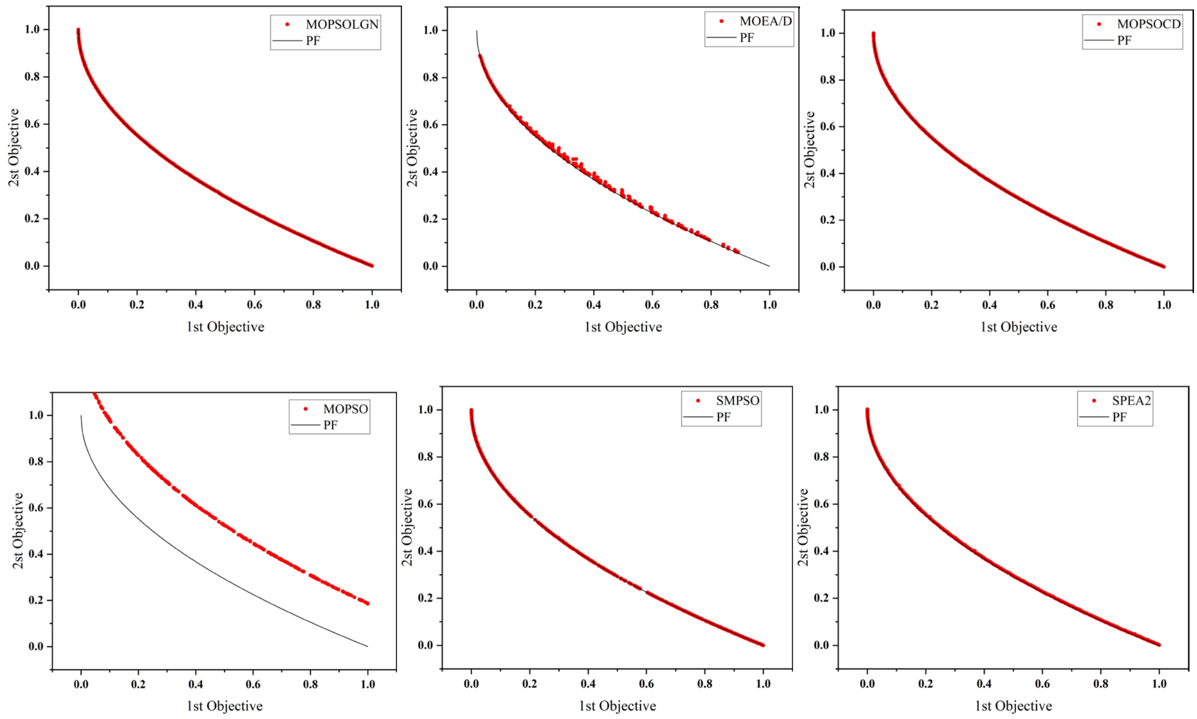

Fig. 4. True and obtained PF by six compared MOEAs on ZDT1

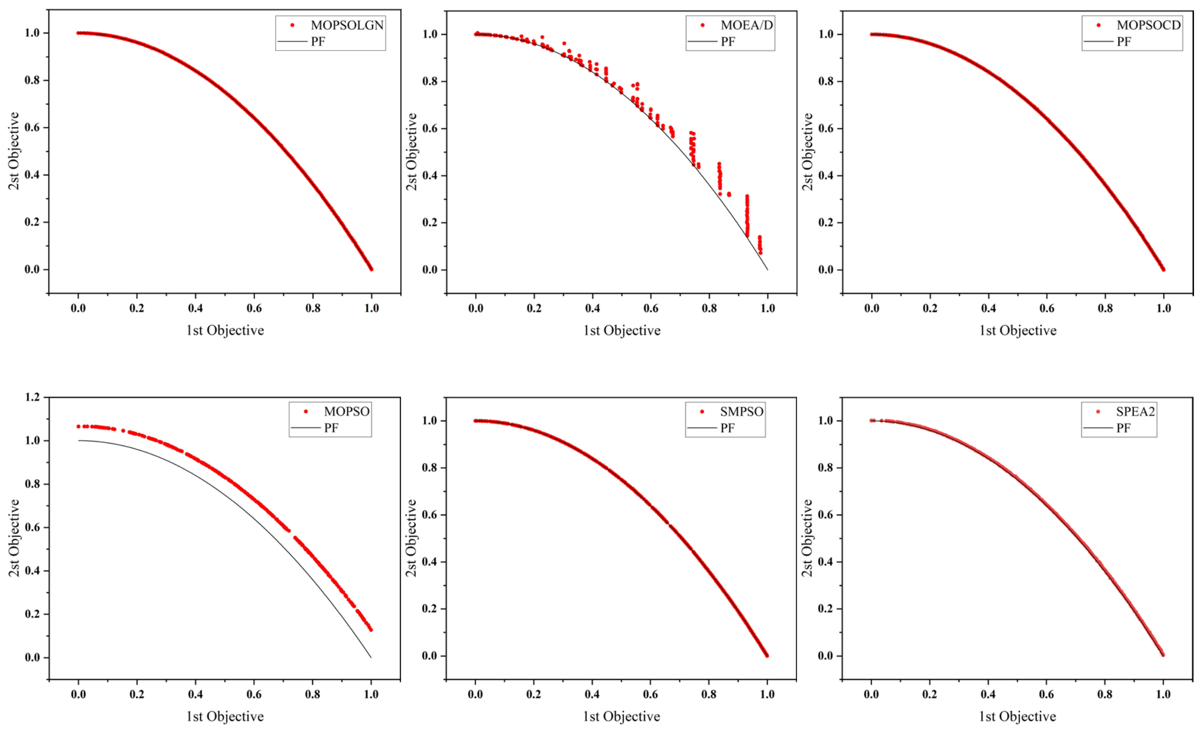

Fig. 5. True and obtained PF by six compared MOEAs on ZDT2

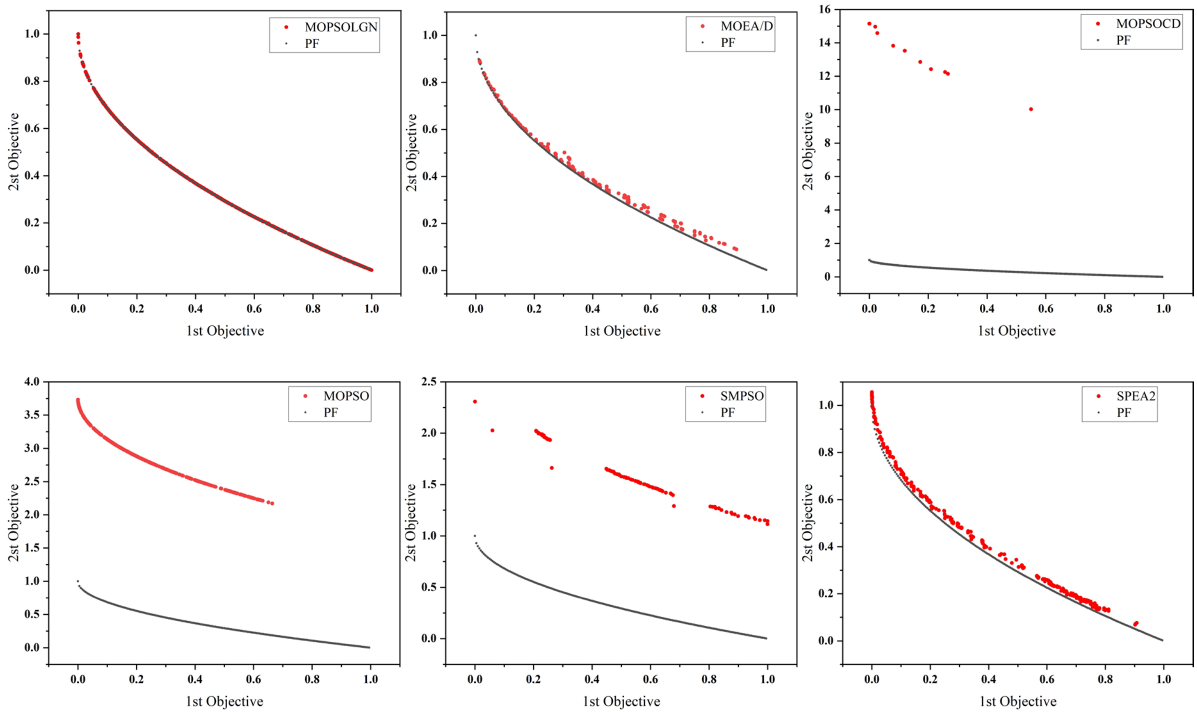

Fig. 6. True and obtained PF by six compared MOEAs on ZDT4

This study focuses on addressing the slow learning issue of traditional PSO algorithms and mitigating population collapse in the objective space. These issues are effectively resolved, as shown in Fig. 6, where MOPSOLGN outperforms other algorithms in convergence and diversity.

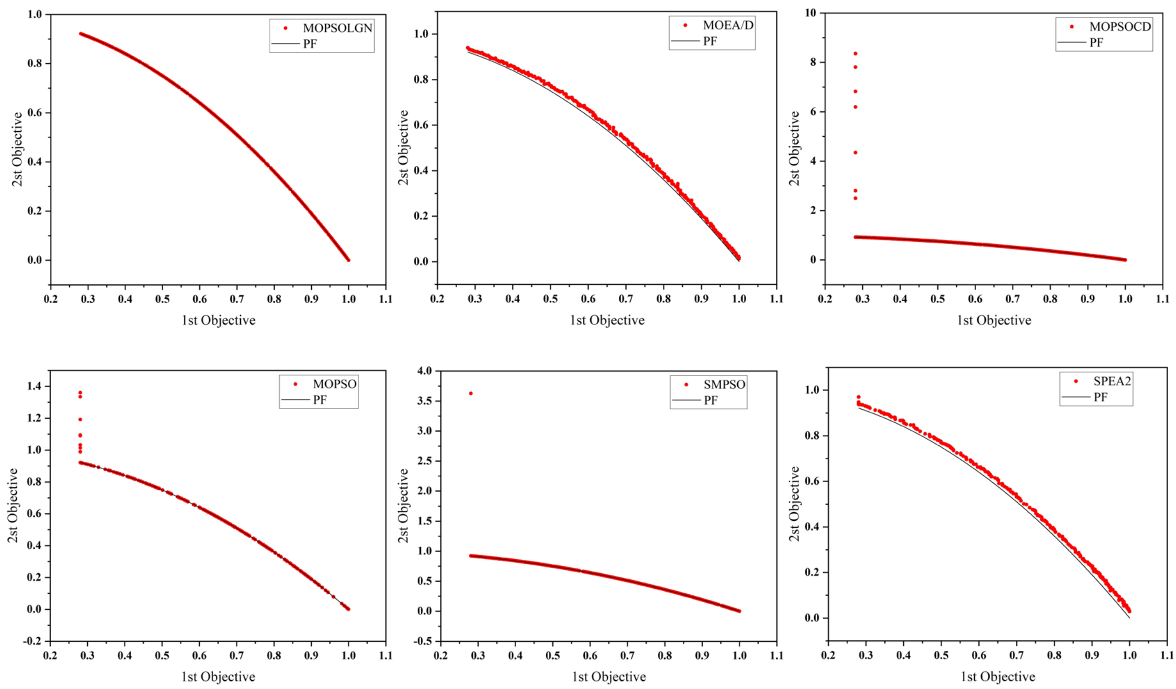

Fig. 7. True and obtained PF by six compared MOEAs on ZDT6

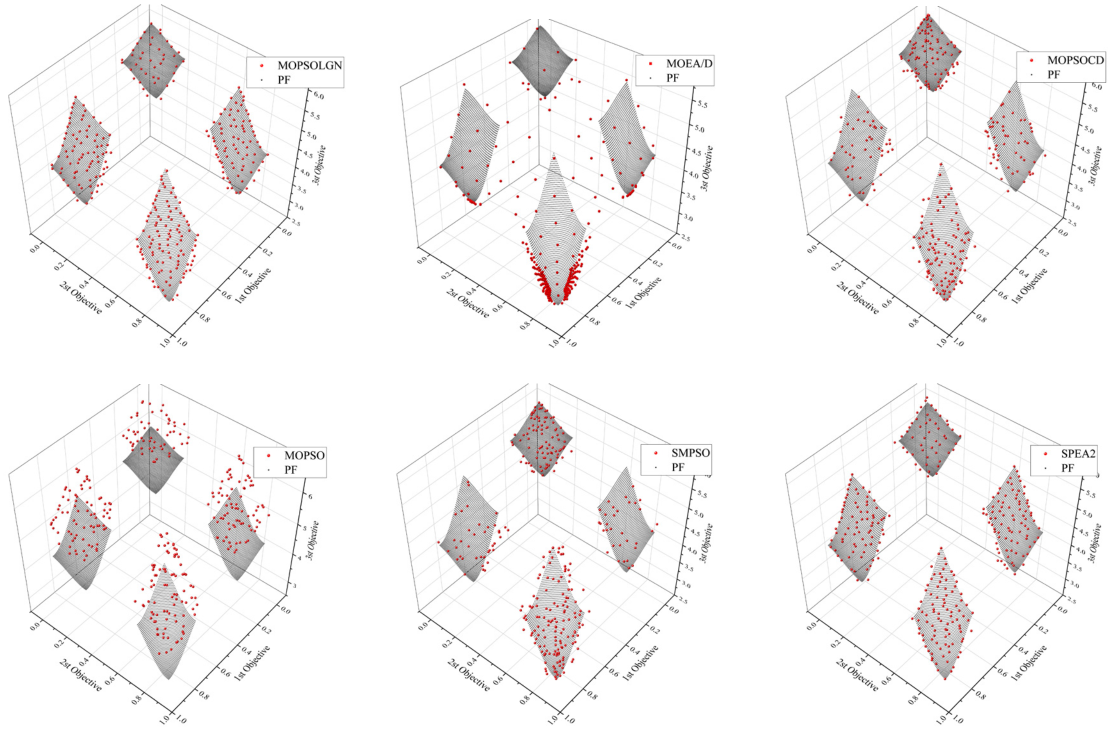

Fig. 8. True and obtained PF by six compared MOEAs on DTLZ7

However, Table 3 and Table 4, along with Fig. 7 and Fig. 8, indicate that MOPSOLGN performs poorly on segmented-type test functions. Despite this, it shows superior convergence and diversity in three-objective optimization problems compared to most other algorithms. These experimental results align with the anticipated outcomes discussed in related work.

4.3 Strategy Validity Analysis

Three main strategies of MOPSOLGN are analyzed in this subsection, i.e., Logistic-tent chaotic mapping with GOBL strategy (LTCMGOBL), non-inertial Lévy flight dynamical equation and differential mutation (DM). In order to facilitate comparison, three variants of MOPSOLGN are set, that is 1) removes LTCMGOBL. 2) removes non-inertial Lévy flight dynamical equation, and replaced by equation (2)-(3). 3). removes DM.

Taking ZDT4 as an example, as shown in Fig. 9, the comparative experiments highlight the significant impact of missing strategies on the optimization process. The absence of the LTCMGOBL strategy negatively affects the exploration speed and the diversity of the global search, leading to blindness and uncertainty in the optimization process. This ultimately prevents the uniformity of traversal and the diversity of the population from reaching the expected outcomes. Without the non-inertial Lévy flight dynamical equation, the population may converge prematurely because the particles' flight paths are not effectively guided, thereby limiting the overall exploration and exploitation capabilities of the algorithm and increasing the likelihood of becoming trapped in local optima. The algorithm performs best on the ZDT4 test function, underscoring the importance of DM. It helps the population avoid the overly restrictive constraints of k-means clustering, preventing population collapse.

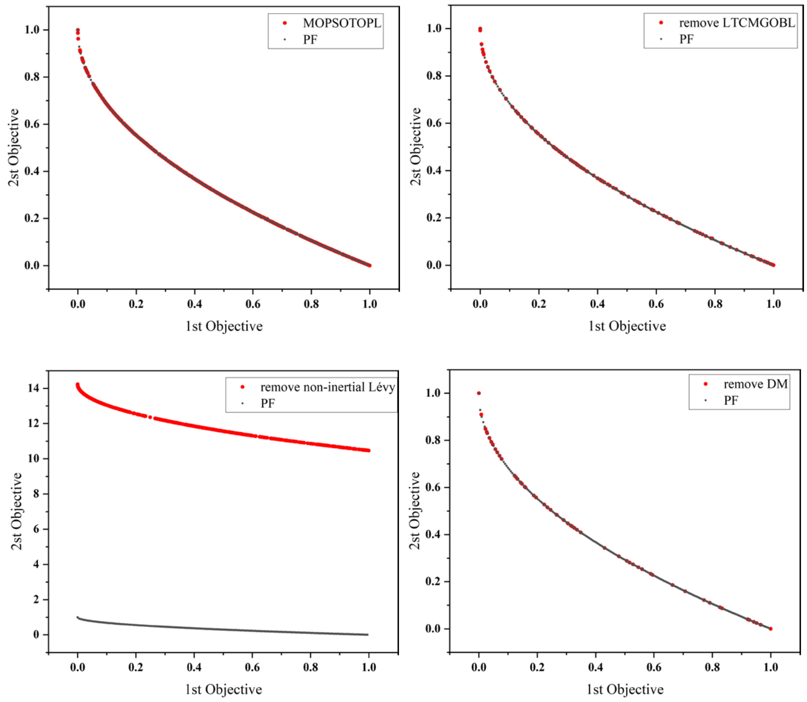

Fig. 9. Comparison trend graphs between MOPSOLGN and its three variants on ZDT4

5 Conclusions

In this paper, a MOPSOLGN algorithm was proposed, which integrates four strategies: LTCMGOBL, CSS, a non-inertial Lévy flight dynamic equation, and DM. Initially, LTCMGOBL improves population uniformity, iteration speed, and solution diversity. CSS and the non-inertial Lévy flight dynamic equation then address issues of population collapse, evolutionary stagnation, and local optimization. Finally, DM helps particles escape local optima. Comparative experiments with benchmark test functions show that MOPSOLGN excels in IGD and SP metrics, outperforming five advanced algorithms. This confirms that the combined strategies significantly enhance the algorithm's performance.

However, MOPSOLGN still has a small probability of population collapse in the early optimization stages, affecting efficiency. This may be due to the interaction between the Leeroy particle force magnitude and mutation in the non-inertial Lévy flight process. Further study is needed to effectively combine these methods for MOPs. Additionally, the performance on segmented-type test problems requires improvement.

Future work will address high-dimensional, large-scale, and dynamic optimization problems, which are critical for real-world applications such as resource scheduling and edge computing.

Acknowledgement

The work of LanLan Kang was supported in part by the Natural Science Foundation of China under Grant No. 62166019, and the Natural Science Foundation of Jiangxi Province under Grant No. 20232BAB202024, and the Science and Technology Research Program of Jiangxi Provincial Department of Education under Grant No. GJJ2403701. The work of Wenliang Cao was supported the Dongguan Science and Technology of Social Development Program under Grant No. 20221800905842, and the Special projects in Key Fields of Colleges and Universities in Guangdong Province under Grant No. 2024ZDZX1092.

References

[1] R. Tavakkoli-Moghaddam, A.H. Akbari, M. Tanhaeean, R. Moghdani, F. Gholian-Jouybari, M. Hajiaghaei-Keshteli, Multi-objective boxing match algorithm for multi-objective optimization problems, Expert Systems with Applications, 239(2024) 122394.

[2] O. Schütze, A.E. Rodríguez-Fernandez, C. Segura, C. Hernández, Finding the Set of Nearly Optimal Solutions of a Multiobjective Optimization Problem, IEEE Transactions on Evolutionary Computation 29(1)(2025) 145-157.

[3] J.-R. Jiang, C.-W. Chu, Classifying and Benchmarking Quantum Annealing Algorithms Based on Quadratic Unconstrained Binary Optimization for Solving NP-hard Problems, IEEE Access 11(2023) 104165-104178.

[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6(2)(2002) 182-197.

[5] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, ETH Zurich, 103(2001) 1-22.

[6] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, PESA-II: region-based selection in evolutionary multiobjective optimization, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, 2001.

[7] Q.-F. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11(6)(2007) 712-731.

[8] H.-L. Liu, F.-Q. Gu, Q.-F. Zhang, Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems, IEEE Transactions on Evolutionary Computation 18(3)(2014) 450-455.

[9] R. Cheng, Y.-C. Jin, M. Olhofer, B. Sendhoff, A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 20(5)(2016) 773-791.

[10] E. Zitzler, S. Künzli, Indicator-Based Selection in Multiobjective Search, in: Proc. 8th International Conference on Parallel Problem Solving from Nature - PPSN VIII, 2004.

[11] N. Beume, B. Naujoks, M.T.M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181(3)(2007) 1653-1669.

[12] J. Bader, E. Zitzler, HypE: An algorithm for Fast Hypervolume-Based Many-objective, Evolutionary Computation 19(1)(2011) 45-76.

[13] X. Wang, W. Ji, G. Zhou, J. Yang, Multi-objective Optimization Algorithm Based on Multi-index Elite Individual Game Mechanism, Journal of System Simulation 35(3)(2023) 494-514.

[14] Z.-S. Song, H.-D. Wang, B. Xue, M.-J. Zhang, Y.-C. Jin, Balancing Objective Optimization and Constraint Satisfaction in Expensive Constrained Evolutionary MultiObjective Optimization, IEEE Transactions on Evolutionary Computation 28(5)(2024) 1286-1300.

[15] S.-Q. Song, K. Zhang, L. Zhang, N. Wu, A dual-population algorithm based on self-adaptive epsilon method for constrained multi-objective optimization, Information Sciences 665(2024) 119906.

[16] G. Sahu, K.C. Ray, PSO Optimized Hybrid Deep Learning Model for Detection and Localization of Myocardial Infarction. IEEE Sensors Journal 24(5)(2024) 6643-6654.

[17] C.-T. Yue, P.N. Suganthan, J. Liang, B.-Y. Qu, K.-J. Yu, Differential evolution using improved crowding distance for multimodal multiobjective optimization, Swarm and Evolutionary Computation 62(2021) 100849.

[18] P. Tripathy, P.M. Khilar, PSO based Amorphous algorithm to reduce localization error in Wireless Sensor Network, Pervasive and Mobile Computing 100(2024) 101890.

[19] E.A. Aner, M.I. Awad, O.M. Shehata, Performance evaluation of PSO-PID and PSO-FLC for continuum robot's developed modeling and control, Scientific reports 14(1)(2024) 733.

[20] M. Camas-Náfate, A. Coronado-Mendoza, C. Vargas-Salgado, J. Águila-León, D. Alfonso-Solar, Optimizing Lithium-Ion Battery Modeling: A Comparative Analysis of PSO and GWO Algorithms, Energies 17(4)(2024) 822.

[21] C.-W. Xie, F.-L. Zhang, J.-B. Lu, C. Xiao, G.-L. Long, Multi-Objective Firefly Algorithm Based on Multiply Cooperative Strategies, Acta Electroncia Sinica, 47(11)(2019) 2359-2367.

[22] J.-B. Lian, G.-H. Hui, Human Evolutionary Optimization Algorithm, Expert Systems with Applications 241(2024) 122638.

[23] D. Boyer, G. Mercado-Vásquez, S.N. Majumdar, G. Schehr, Optimizing the random search of a finite-lived target by a Lévy flight, Physical Review E 109(2)(2024) L022103.

[24] S. Dipierro, G. Giacomin, E. Valdinoci, The Lévy flight foraging hypothesis: comparison between stationary distributions and anomalous diffusion, Journal of Physics A: Mathematical and Theoretical 56(48)(2023) 485601.

[25] M.-J. Duan, C. Yu, S.-P. Wang, B. Li, A differential evolution algorithm with a superior-inferior mutation scheme, Soft Computing - A Fusion of Foundations, Methodologies & Applications 27(23)(2023) 17657-17686.

[26] Z.-Y. Hua, Z.-H. Zhu, S. Yi, Z. Zhang, H.-J. Huang, Cross-plane colour image encryption using a two-dimensional logistic tent modular map, Information Sciences 546(2021) 1063-1083.

[27] S. Patel, T. Veeramalai, Image Encryption Using a Spectrally Efficient Halton Logistics Tent (HaLT) Map and DNA Encoding for Secured Image Communication, Entropy 24(6)(2022) 803.

[28] J. Li, Q. An, H. Lei, Q. Deng, G.-G. Wang, Survey of Lévy Flight-Based Metaheuristics for Optimization, Mathematics 10(15)(2022) 2785.

[29] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous

Spaces, Journal of Global Optimization 11(4)(1997) 341-359.
[30] L.-L. Kang, R.-S. Chen. W.-L. Cao, Y.-C. Chen, non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications, Applied Soft Computing 88(2020) 106038.
[31] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation 8(2)(2000) 173-195.
[32] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, In: A. Abraham, L. Jain, R. Goldberg (eds.), Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing, Springer, London, 2005 (105-145).
[33] L.-P. Wang, Y. Ren, Q.-C. Qiu, F.-Y. Qiu, Survey on Performance Indicators for Multi-Objective Evolutionary Algorithms, Chinese Journal of Computers 44(8)(2021) 1590-1619.
[34] Y. Tian, R. Cheng, X.-Y. Zhang, Y.-C. Jin, PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization 12(4)(2017) 73-87.