Journal of Computers Vol. 35 No. 6, December 2024, pp. 29-43
doi: 10.53106/199115992024123506003

Cubic Chaos Preference Multi-Objective Optimization Algorithm with Adaptive Dual-Mode Mutation

Wenwu Liu[1], Lanlan Kang[2,1*], Qian Du[3], Wenliang Cao[4], and Xiuniao Zhao[2]

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

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

[3] School of Information Science and Engineering, Linyi University,
Linyi 27600, China
duqian@lyu.edu.cn

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

Received 30 April 2024; Revised 8 June 2024; Accepted 24 June 2024

Abstract. In the study of preference-based multi-objective optimization algorithms, the performance significantly depends on the preference information provided by the decision-maker. Over-reliance on this preference information can lead the algorithms to become trapped in locally optimal solutions, potentially overlooking high-quality solutions in other regions. Therefore, this paper proposes a Cubic Chaos Preference Multi-Objective Optimization Algorithm with Adaptive Dual-mode Mutation (CPMOP-DM). Firstly, this algorithm utilizes the cubic chaos strategy to initialize population. This strategy possesses better chaos traversal and optimization speed and then, helps enhance the search breadth and global convergence of the optimization algorithm. Secondly, a dynamic focused preference exploration strategy is proposed to enhance the quality and satisfaction of the selected solutions. This strategy can gradually refine the search scope via, constructing dynamically shrinking exploration circles. Compared to traditional preference-based algorithms, experimental results demonstrate the competitiveness of the proposed algorithm. It effectively balances exploration and exploitation searching of the algorithm, thereby enhancing the algorithm's diversity and distribution. It also avoids the local optimum problem caused by over-reliance on preference information.

Keywords: multi-objective optimization strategy, cubic chaos, dynamic, exploration

1 Introduction

Multiple Objective Evolutionary Algorithms (MOEAs) are heuristic optimization algorithms for solving Multi-Objective Optimization Problems (MOPs) [1]. It is widely used to find a set of optimal solutions among conflicting objective functions, forming a solution set called the Pareto frontier.

However, in real-world scenarios, decision-makers are concerned with specific regions rather than all possible solutions. Therefore, incorporating preference information into multi-objective optimization has become essential.

In recent years, researchers have proposed various preference-based multi-objective optimization algorithms. Luo et al. [2] introduced the g-dominance Non-dominated Sorting Genetic Algorithm (NSGA-II) to alleviate dominance selection pressure. Sun et al. [3] proposed HP-NSGA-II, achieving uniform distribution using Chebyshev distance and dynamic region updating. Dai et al. [4] used an adaptive reference region strategy to

* Corresponding Author

prevent solution set concentration. Said et al. [5] developed r-NSGA-II, combining weighted Euclidean distance and preference information for strict partial order relations. Xie et al. [6] proposed a preference decomposition algorithm generating new standard objective vectors through boundary-crossing and hyperplane shifting. Tian et al. [7] leveraged Gaussian processes for preference-based optimization, improving algorithm efficiency. Zhang et al. [8] created a model using adaptive large neighborhood search to resolve objective conflicts, incorporating fuzzy preferences for efficiency and aligned solutions with decision-makers' requirements.

Although these algorithms perform well in incorporating preference information, enhancing search efficiency, and aligning solution sets with decision-makers' preferences, they also have certain drawbacks. The main issue is that these algorithms tend to overly rely on the decision-makers' preference information, which can lead to the solution set becoming trapped in local optima, thereby overlooking potential high-quality solutions in other regions.

For this reason, this paper proposes a novel preference optimization algorithm named CPMOP-DM within the framework of the NSGA-II algorithm [9]. Initially, population uniform distribution is ensured through a novel initialization method, followed by the adoption of strategies for dynamically adjusting the preference region and dual-mode mutation, aiming to enhance the diversity and convergence speed of the population. The main technical achievements and contributions of this algorithm are as follows:

(1) The algorithm introduces a cubic chaotic mapping method to initialize the population, ensuring a more uniform distribution across the entire search space, thus enhancing the global search capability of the algorithm.

(2) A new dynamic focused preference exploration strategy is proposed, which gradually narrows the preference region, guiding individuals to converge towards the decision maker's preferred region, thereby improving the algorithm's convergence speed and diversity.

2 Related Works

This section provide concepts and formulas for multi-objective optimization, g-dominance strategy, and cubic chaotic mapping. These strategies tackle two main challenges in preference-based multi-objective optimization: effectively using decision-maker preferences and addressing uneven population distribution in the objective space.

2.1 Multi-objective Optimization Problems

A multi-objective optimization problem [10] can be generally defined as in equation (1):

$$\begin{aligned} &\min F(x) = \min\{y_1(x), y_2(x), \ldots y_m(x)\}, x \in R^n \,. \\ &s.t. \begin{cases} g_a(x) \le 0, \ \mathrm{a} = 1\ldots, M \\ h_b(x) = 0, \ \mathrm{b} = 1\ldots, N \end{cases} . \end{aligned} \tag{1}$$

Where, y is a decision variable in the n-dimension searching space R^n, $y_m(x)$ is the m-th objective function in the objective space, $g_a(x)$ and $h_b(x)$ are the a inequality constraints and the b equality constraints.

In MOPs, finding a solution that optimizes all objectives simultaneously is often challenging. Hence, the Pareto solution set is used as the optimization outcome [11], which enables selecting the optimal solution from multiple solution sets. This ensures that solutions are evenly distributed in the solution space while considering each objective, allowing the decision maker to choose the best solution based on specific needs.

2.2 G-dominance

Molina et al. [1] proposed a novel strategy called g-dominance, which enhances the flexibility of the dominance relationship between individuals using preference points provided by the decision maker. This strategy introduces preference points to alter the dominance relationship between individuals, resulting in a finer partitioning of the target space. It effectively improves the efficiency of the Pareto dominance algorithm in population selection. The strategy is defined as follows:

$$flag_{g(y)} = \begin{cases} 1, \text{ if } \mathrm{y}_i \le g_i \,\|\, g_i < y_i, \ \forall i = 1,\ldots,\mathrm{m} \\ 0, \text{ other} \end{cases} . \quad \textbf{(2)}$$

Given two objective vectors x and $x^* \in R^m$, xg dominates x^* if and only if it satisfies the condition $Flag_{g(y)}(x) \ge Flag_{g(y)}(x^*)$ and $Flag_{g(y)}(x) = Flag_{g(y)}(x^*)$, but needs to satisfy for any $x_i \le x^*_i (\forall i \in 1,\ldots\mathrm{m})$ there exists at least one i such that $x_i \le x^*_i$.

2.3 Cubic Chaotic Mapping Initialization

In the traditional NSGA-II algorithm [9], randomly generated initial populations often lead to a non-uniform distribution across the solution space, potentially affecting algorithm performance. Among various chaotic mapping sequence methods, CCM is regarded as one of the more effective methods. Compared to traditional random initialization, CCM utilizes chaotic mapping's characteristics to achieve a more uniform distribution of the initial population across the search space by introducing nonlinearity and randomness. This initialization method not only expands the coverage of potential solution spaces but also mitigates population concentration issues in local regions, thereby enhancing the likelihood of discovering global optimal solutions. From Fig. 1, a better understanding of the advantages of CCM can be obtained.

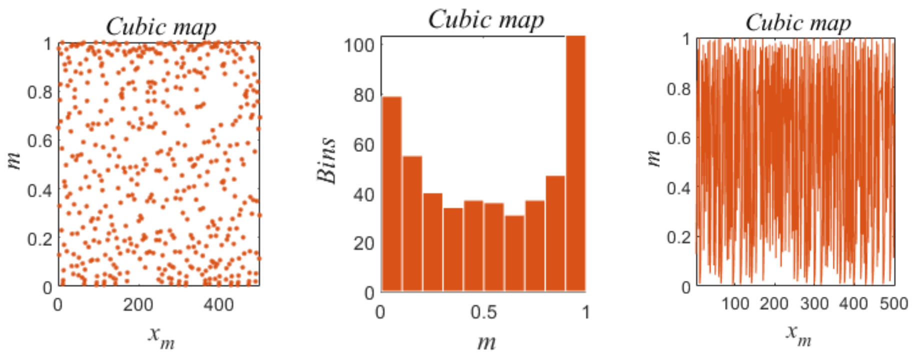


Fig. 1. Cubic chaotic map distribution traversal graph

Fig. 1 is a visualization of CCM, where *m* represents the chaotic value, x_m represents the dimension, and *Bins* represents the frequency. It can be observed from the figure that CCM exhibits better characteristics of diversity, stability, and randomness. In this paper, the population positions are initialized using the cubic map method [12], defined as follows:

$$y(k+1) = 4y(k)^3 - 3y(k), \ \ |y(k)| \le 1 \ \ k = 0,1,2\ldots N . \quad \textbf{(3)}$$

$$x_k = M_p + (1 + y_k) * \frac{N_p - M_p}{2} . \quad \textbf{(4)}$$

The initial population is generated using equation (3). Then, the population positions are mapped into the solution space according to equation (4) for initialization, where M_p represents the lower limit of the solution space and N_p represents the upper limit. The main execution steps of CCM initialization are described in Algorithm 1.

Algorithm 1. CCM

```
Inputs: populations pop;
Output: populations after initialization;
Initialize populations using CCM;
Generate complementary population mpop
of pop;
  If mpop < 0
    Constrain mpop to be between 0 and 1;
  end
Calculate the Pareto rank for each individual from {pop
∪ mpop};
Select half of the high-quality individuals as the ini-
tial population.
```

3 Cubic Chaos Preference Multi-Objective Optimization Algorithm with Adaptive Dual-Mode Mutation

This section proposes a Cubic Chaos Preference Multi-Objective Optimization Algorithm with Adaptive Dual-mode Mutation (CPMOP-DM). The algorithm leverages the CCM strategy for population initialization, capitalizing on its superior chaos traversal and optimization speed. Additionally, introducing a dynamic focused preference exploration (DFPE) strategy aimed at refining the search scope and aligning the exploration solution set with preference boundaries. During the evolution of the algorithm, a dual-mode mutation enhanced (DM) strategy is also employed, combining differential mutation with crossover mutation. By establishing regions using g-dominance, the optimization range of the particle population can be constrained, thereby guiding the population to iterate more effectively toward the optimization direction. The main execution steps of CPMOP-DM are described in Algorithm 4.

3.1 Dynamic Focused Preference Exploration Strategy

In MOPs, integrating preference information [13] is vital. However, the conventional g-dominance approach delineates preference and non-preference regions solely based on decision maker input [14], leading to subjective influence on algorithm diversity selection.

This section introduces a new strategy termed DFPE. Initially, the preference point and region are determined based on decision maker input. Then, a large circular exploration area centered at the preference point is established, from which high-quality individuals are selected iteratively. As iterations progress, the exploration area gradually contracts, with high-quality individuals from within it selected for the subsequent iteration's exploration area. Eventually, the exploration area becomes tangent to the preference region boundary, forming a focusing circle. The main steps of DFPE are outlined in Algorithm 2.

In Fig. 2, initially, the exploration circle's preference region is large enough to include the desired quality individuals, i.e., the objective vectors. As iterations proceed, the preference region shrinks until it matches the preference region's radius, as shown in Fig. 3. Initially, target vectors are distributed across non-preference and preference regions, while in the final stage, they concentrate in the new preference regions. This occurs because quality individuals are selectively retained as the exploration circle shrinks.

The key to this strategy is dynamically adjusting preference regions during algorithm execution to balance exploration and exploitation. By continuously resizing preference regions, this dynamic mechanism ensures the algorithm maintains sufficient diversity, increasing the chance of finding higher quality and more diverse solutions.

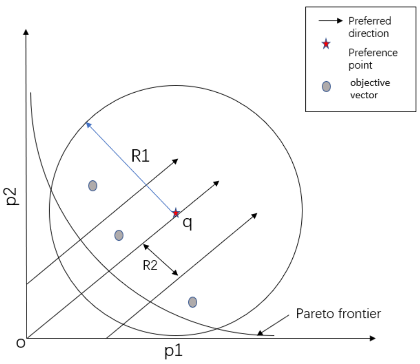


Fig. 2. Exploring the preference circle

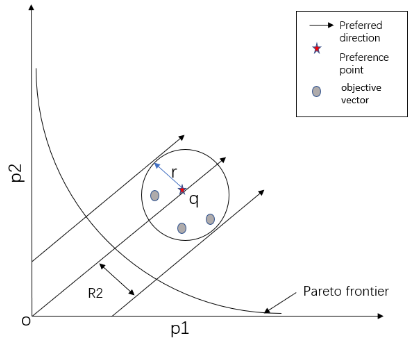


Fig. 3. Focused preference circle

Algorithm 2. DFPE

```
Inputs: populations, Initial preference area domain, prefer-
ence points;
Output: High-quality populations;
Initialize temp1 and temp0 as empty arrays
    Calculate the distance to the center point for
    each individual in the combined population
    for i = 1 : m
     if the individual's distance is less than the radius of
    the temporary circle.
        Putting individuals into temp1
     else
        Putting individuals into temp2
     end
    end
    Calculating the size of temp1 and temp2
    if  temp1 >=pop
        Non-dominated sorting to select high-quality individ-
    uals to be placed in the preference region
    else
        Search for quality individuals from non-preferred ar-
    eas to place in preferred areas until population size is
    reached
```

3.2 Dual-Mode Enhanced Mutation

This paper proposes an innovative DM to enhance the quality and diversity of solutions for MOPs. This strategy employs binary crossover mutation [15] to generate a new population, followed by differential mutation [16]. Binary crossover mutation simulates gene crossover and mutation in biological evolution, enabling extensive exploration of new solutions across the search space, thus generating a diverse population. where $b1$ and b_2 are two parents, C_1 and C_2 are two offspring, V is the distribution factor, a is a random number between [0, 1], and m is a user-defined parameter. The larger the value of m , the greater the probability that the offspring individuals will approach the parents.

$$C_1 = \frac{1}{2}(b_2 + b_1) - \frac{1}{2} * V * (b_2 - b_1) \ . \tag{5}$$

$$C_2 = \frac{1}{2}(b_2 + b_1) + \frac{1}{2} * V * (b_2 - b_1) \ . \tag{6}$$

$$V = \begin{cases} (2 * a)^{\frac{1}{m+1}} & \text{a} \le 0.5 \\ \left(\frac{1}{2 * (1-a)}\right)^{\frac{1}{m+1}} & \text{a} > 0.5 \end{cases} \ . \tag{7}$$

Differential mutation is employed as a secondary mutation to effectively adjust the search direction of the population. This strategy enhances search efficiency and global convergence by computing differences between individuals in the population. where q_1 and q_2 are weight coefficients, f_1 and f_2 are coefficients of Gaussian distribution functions using random numbers generated from a Gaussian distribution with mean 0 and variance 1, Y^* denotes the position of the optimal individual, Y_{rand} represents the position of the random individual, and $Y(t)$ denotes the position of the current individual.

$$Y(t+1) = q_1 * f_1 * (Y^* - Y(t)) + q_2 * f_2 * (Y_{rand} - Y(t)) \ . \tag{8}$$

Overall, the DM convergence speed and solution quality of the population while preserving diversity by leveraging the strengths of both binary crossover and differential mutation. This strategy presents a novel and effective approach for addressing multi-objective optimization problems. The main execution steps of DM are described in Algorithm 3.

Algorithm 3. DM

```
Inputs: populations pop, iterations, crossover probability pc,
mutation probability pm, decision variables x_num;
Output: mutated population;
First, the population undergoes bi nary crossover muta-
tion;
 If rand(1) < pc
   perform binary crossover;
 end
 If rand(1) < pm
   Perform polynomial mutation;
 end
 Then, the population undergoes differential mutation.

 For i = 1 : pop
   for j = 1 : x_num
     if off(i, j) != x_max(j)
        off(i, j) = x_max(j)
     end
   end
 end
```

```
Algorithm 4. The main steps of CPMOP-DM
Inputs: populations pop, iterations it, preference points, radius of preference, crossover probability, mutation probability, decision variables Preference convergence indicator y;
Output: pareto optimal solution set;
   CCM;
    For i = 1 : it
      DM;
      The mutated population is then combined with the initial population;
      DFPE;
      Calculate the dominance ratio of the population;
      if ratio >= y
        g-dominance sort refer to reference [2];
      else
        non-dominated sort refer to reference[14]
      end
    end
```

4 Experiments

This section introduces the evaluation metrics, experimental parameter settings, and provides a detailed analysis of the experimental results. The performance of the CPMOP-DM algorithm is compared with two preference algorithms, g-NSGAII and r-NSGAII, using ZDT [17] and DTLZ [18] test functions. These functions are commonly used for algorithm performance testing. All experiments were conducted on the PlatEMO v4.4 platform within MATLAB R2021a [19].

4.1 Performance Metrics

In this paper, Generational Distance (GD) [20], Spread (SP) [21], and Inverted Generational Distance (IGD) [22] are selected as evaluation metrics due to their ability to quantify solution quality, diversity, and convergence, respectively.

Firstly, GD is a commonly used metric for assessing algorithm convergence, measuring the distance between the algorithm's solution set and the true Pareto frontier. Smaller GD values indicate better convergence. The equation for GD is:

$$GD = \frac{\sqrt{\sum_{i=1}^{N} (dis_i^2)}}{N} . \tag{9}$$

where dis_i is the minimum Euclidean distance from individual i in the population to the true Pareto frontier. This index is used to measure the proximity between the optimized solution set and the true Pareto front. Smaller values signify superior convergence performance, indicating closer proximity to the true Pareto front.

When addressing a multi-objective optimization problem, the resulting solution typically consists of a set of non-dominated optimal solutions. In the domain of preference-based multi-objective research, the optimal solutions of interest are those distributed around the preference point region. The evolution of a single optimal solution to a point may not offer additional alternative solutions for decision-making but could potentially complicate the decision-making process. In addition to considering GD as an evaluation metric, SP and IGD are also included. A smaller SP value indicates better diversity of the algorithm. The definition of SP is illustrated in the following equation:

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d - d_i)^2} \quad . \tag{10}$$

where d_i represents the minimum Euclidean distance between the i non-dominated solution of the solution set and other solution individuals; d denotes the average value of d_i among the optimized non-dominated solution set individuals.

A smaller IGD value indicates a more uniform distribution of the solution set generated by the algorithm, approximating the real Pareto front and ensuring uniform distribution. Therefore, IGD and SP are crucial indicators for assessing the overall performance of the algorithm.

$$IGD(P,Q) = \frac{\sum_{v \in p} d(v,Q)}{|P|} \quad . \tag{11}$$

Here, P represents the set of points uniformly distributed on the true Pareto surface, and $|P|$ denotes the number of individuals in this set. Q represents the set of optimal Pareto-optimal solutions obtained by the algorithm. $d(v, Q)$ signifies the minimum Euclidean distance from individual v to population Q.

4.2 General Parameters

Population sizes are set to N=100 for the ZDT series test functions, with a maximum of 200 iterations. For the DTLZ series test functions, population sizes are set to N=200, with a maximum of 300 iterations.

Reproduction operator: The r-NSGAII algorithm in this paper adopts average weight allocation for each target and sets the non-dominance threshold δ to 0.1. The R-value in the preference vector bootstrap and the exploration circle are set to 0.1 and 1.0, respectively. To maintain population diversity and prevent excessive preference region adjustment, the preference convergence index γ is set to 0.5. Simulated binary crossover and polynomial mutation are chosen as genetic operators, with probabilities of 0.99 and 0.1, respectively, to balance convergence and diversity.

Number of evaluations and preference point coordinate settings: This paper sets different parameters for different test questions and dimensions, and the specific values are shown in Table 1. All the algorithms are run independently for 30 times.

4.3 Experimental Results

To comprehensively validate the competitiveness of the algorithms proposed in this paper, their performance is compared with that of the benchmark algorithms g-NSGAII and r-NSGAII on both 2D and 3D test functions. Table 2 to Table 4 present the results of each algorithm, with the algorithms exhibiting optimal performance highlighted by horizontal lines beneath the data.

Table 2 reveals that the r-NSGAII algorithm excels in specific scenarios. It achieves optimal GD values on the ZDT1 frontier, the infeasible and feasible domains of the ZDT3 frontier, and the DTLZ4 frontier. Although r-NSGAII generally has better GD values for ZDT3, this is due to the complex multi-modal nature of ZDT3. r-NSGAII use of reference points and regularization helps maintain diversity and avoid local optima, covering the entire Pareto front. However, the proposed algorithm consistently performs best across most experiments, demonstrating adaptability and robustness. By dynamically adjusting search strategies and preserving population diversity, it consistently finds high-quality solutions. Therefore, while r-NSGAII excels in specific cases, the proposed algorithm shows superior overall balance and stability.

Table 3 and Table 4 show that the g-NSGAII and r-NSGAII algorithms perform significantly worse than the proposed algorithm on IGD and SP metrics. The proposed algorithm demonstrates superior diversity and distribution across several test functions. On the IGD metric, it achieves closer proximity to the true Pareto frontier, indicating its advantage in maintaining population diversity. On the SP metric, its solution set is more uniformly distributed, reflecting its superiority in solution set distribution. These results highlight the effectiveness and advantages of the proposed algorithm in multi-objective optimization problems.

Fig. 4(a) indicates poor convergence of the r-NSGAII algorithm, resulting in limited coverage of the preference solution set and fewer alternative solutions. In contrast, both the paper algorithm and g-NSGAII demonstrate more balanced convergence. However, g-NSGAII covers too many Pareto fronts, adding decision-making pressure. The algorithm in paper effectively controls the preferred solution set range, offering the decision-maker necessary non-dominated solutions to better meet their needs.

Fig. 4(b) and Fig. 4(d) show that both the paper algorithm and r-NSGAII exhibit similar convergence for the preference solution sets. However, the algorithm in paper offers a wider range of solutions aligning with the decision-maker's preferences. In contrast, g-NSGAII generates too many preference solution sets, complicating decision-making and reducing practicality.

Fig. 4(c) illustrates distinct performances of the three algorithms. While g-NSGAII suffers from excessive influence of preference points leading to poor convergence, r-NSGAII shows good convergence but with an overly concentrated preference range. In contrast, the proposed algorithm excels in the ZDT2 problem by achieving both good convergence and accommodating decision maker's preferences, offering diverse and personalized solution selections. This showcases the proposed algorithm's robust global search capability and its ability to cater to decision maker's requirements.

In the case of the three-dimensional test function DTLZ2, Fig. 5 depicts the preference solution sets generated by the three algorithms. As evident from Fig. 5(a) and Fig. 5(b), all algorithms converge to the Pareto frontier. But, g-NSGAII and r-NSGAII result in clustered solution sets, constraining decision makers' options. Conversely, the algorithm in paper presents a well-distributed solution set, offering decision makers more choices.

Examining the remaining graphs in Fig. 5, it is clear that the g-NSGAII algorithm shows poor convergence and overly dispersed solution sets. The r-NSGAII algorithm converges to the Pareto front but suffers from over-aggregation. In contrast, the proposed algorithm effectively handles 3D problems, achieving balanced convergence across feasible and infeasible domains.

Table 1. Parameter setting of test functions

Test function	Parameter settings	Preference point		
		On the frontier	Infeasible region	Feasible region
ZDT1	m=2;n=30	(0.5,0.3)	(0.1,0.2)	(0.8,0.8)
ZDT2	m=2;n=30	(0.6,0.64)	(0.2,0.4)	(0.9,0.9)
ZDT3	m=2;n=30	(0.24,0.28)	(0.2,0.2)	(0.5,0.6)
ZDT6	m=2;n=10	(0.6,0.64)	(0.3,0.2)	(0.7,0.8)
DTLZ2	m=3;n=10	(0.5,0.7,0.5)	(0.2,0.3,0.4)	(0.7,0.7,0.8)
DTLZ4	m=3;n=10	(0.3,0.4,0.5)	(0.6,0.6,0.8)	(0.5,0.5,0.7)

Table 2. Results of GD calculations for each of the three algorithms run 30 times on the test function

Test function	Preference point	g-NSGAII		r-NSGAII		CPMOP-DM	
		Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
ZDT1	On the frontier (0.5,0.3)	2.46E-05	1.18E-05	<u>1.82E-05</u>	9.03E-06	5.26E-05	9.00E-06
	Infeasible region (0.1,0.2)	3.06E-04	1.41E-04	2.23E-04	1.16E-04	<u>7.01E-05</u>	9.00E-06
	Feasible region (0.8,0.8)	8.33E-05	1.68E-05	5.70E-05	4.70E-05	<u>4.51E-05</u>	1.20E-05
ZDT2	On the frontier (0.6,0.64)	5.22E-04	3.42E-04	4.32E-04	1.27E-04	<u>1.15E-04</u>	1.40E-05
	Infeasible region (0.2,0.4)	2.31E-04	2.86E-05	1.29E-04	1.43E-04	<u>7.05E-05</u>	1.60E-05
	Feasible region (0.9,0.9)	1.73E-04	2.15E-05	2.75E-04	1.07E-04	<u>1.17E-04</u>	1.60E-05

ZDT3	On the frontier (0.24,0.28)	7.58E-05	4.12E-05	5.85E-05	2.44E-05	3.16E-04	1.10E-05
	Infeasible region (0.2,0.2)	4.04E-05	9.68E-06	2.50E-05	2.17E-05	1.30E-04	3.62E-03
	Feasible region (0.5,0.6)	5.47E-05	4.47E-06	2.96E-05	3.55E-05	5.73E-04	3.01E-03
ZDT6	On the frontier (0.6,0.64)	6.36E-02	4.97E-02	5.67E-02	9.81E-03	1.10E-04	1.70E-05
	Infeasible region (0.3,0.2)	2.13E-02	1.67E-02	1.90E-02	3.22E-03	2.18E-04	1.46E-04
	Feasible region (0.7,0.8)	2.78E-02	2.33E-02	2.56E-02	3.19E-03	2.73E-04	7.35E-03
DTLZ2	On the frontier (0.5,0.7,0.5)	5.86E-04	2.52E-04	4.19E-04	2.36E-04	3.23E-04	1.60E-05
	Infeasible region (0.2,0.3,0.4)	3.09E-03	2.29E-03	5.69E-03	5.60E-04	2.90E-04	1.80E-05
	Feasible region (0.7,0.7,0,8)	5.32E-04	4.66E-05	3.89E-04	3.43E-04	3.23E-04	1.35E-04
DTLZ4	On the frontier (0.3,0.4,0.5)	4.15E-04	5.72E-05	2.36E-04	2.53E-04	5.73E-04	2.05E-04
	Infeasible region (0.6,0.6,0,8)	7.24E-04	1.46E-04	4.35E-04	4.08E-04	3.26E-04	1.15E-04
	Feasible region (0.5,0.5,0.7)	4.48E-04	8.08E-06	2.28E-04	3.11E-04	6.13E-04	3.35E-04

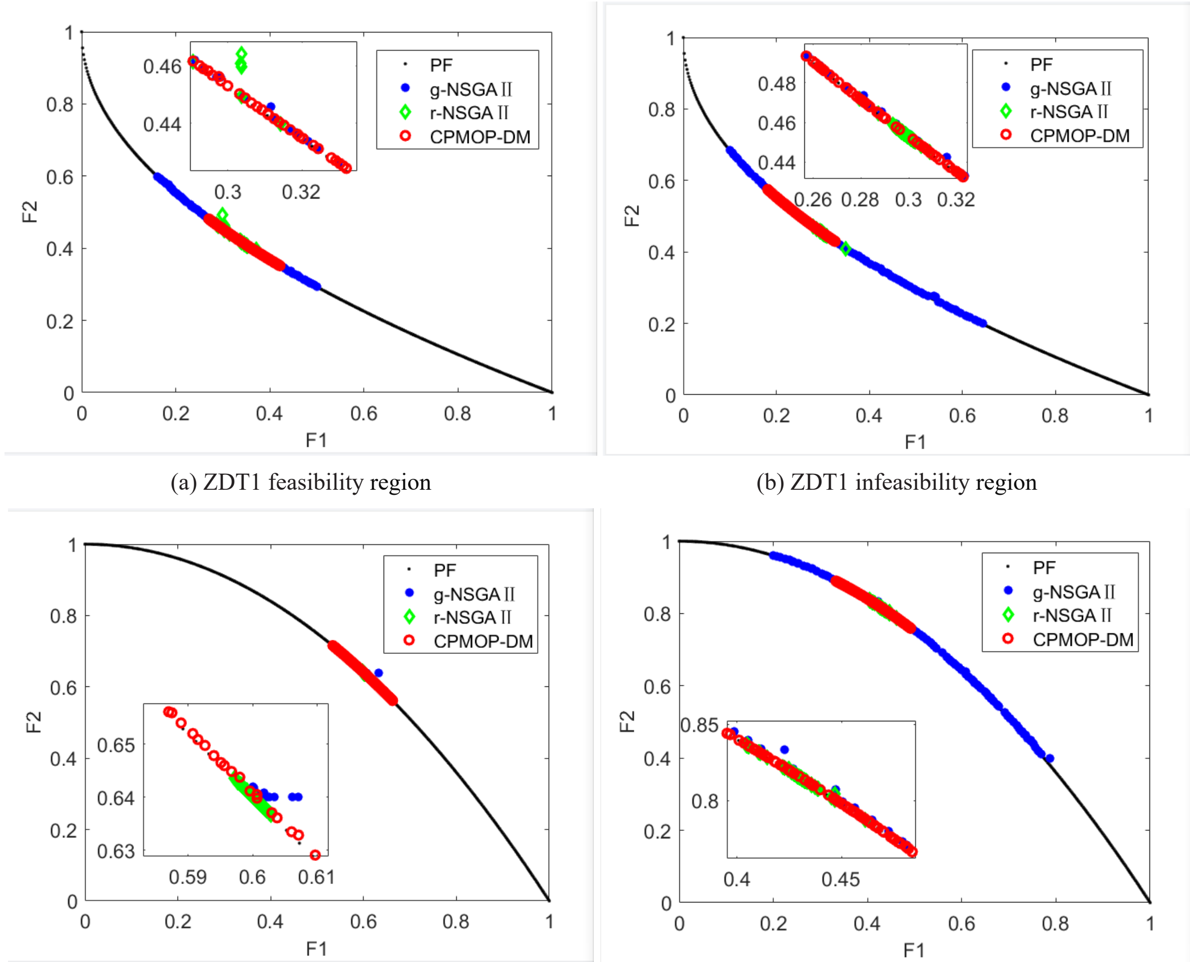
Table 3. Results of 30 IGD calculations for each of the three algorithms run on the test function

Test function	Preference point	g-NSGAII		r-NSGAII		CPMOP-DM	
		Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
ZDT1	On the frontier (0.5,0.3)	3.13E-01	3.50E-04	1.56E-01	2.21E-01	8.87E-04	9.30E-05
	Infeasible region (0.1,0.2)	7.76E-02	8.01E-03	4.28E-02	4.92E-02	9.52E-04	9.70E-05
	Feasible region (0.8,0.8)	1.73E-01	2.53E-04	8.66E-02	1.22E-01	9.32E-04	8.10E-05
ZDT2	On the frontier (0.6,0.64)	3.68E-01	6.24E-03	1.87E-01	2.56E-01	1.10E-03	1.48E-04
	Infeasible region (0.2,0.4)	7.15E-02	1.12E-03	3.63E-02	4.97E-02	9.40E-04	1.75E-04
	Feasible region (0.9,0.9)	6.83E-02	2.15E-04	3.42E-02	4.81E-02	1.11E-03	1.73E-04
ZDT3	On the frontier (0.24,0.28)	4.80E-01	2.02E-04	2.40E-01	3.39E-01	2.61E-03	1.34E-04
	Infeasible region (0.2,0.2)	3.01E-01	1.77E-04	1.50E-01	2.12E-01	7.16E-03	3.45E-02
	Feasible region (0.5,0.6)	1.85E-01	1.93E-04	9.26E-02	1.30E-01	8.10E-03	2.99E-01
ZDT6	On the frontier (0.6,0.64)	2.45E-01	4.11E-03	1.24E-01	1.70E-01	9.32E-04	2.24E-04
	Infeasible region (0.3,0.2)	8.96E-02	3.04E-02	6.00E-02	4.19E-02	1.34E-03	1.11E-03
	Feasible region (0.7,0.8)	2.85E-01	8.00E-02	1.82E-01	1.45E-01	6.61E-03	7.39E-02

DTLZ2	On the frontier (0.5,0.7,0.5)	5.42E-01	1.42E-02	2.78E-01	3.73E-01	4.28E-03	2.01E-04
	Infeasible region (0.2,0.3,0.4)	1.89E-01	2.80E-03	9.59E-02	1.31E-01	3.83E-03	2.36E-04
	Feasible region (0.7,0.7,0,8)	2.43E-01	1.65E-03	1.22E-01	1.71E-01	4.26E-03	2.36E-04
DTLZ4	On the frontier (0.3,0.4,0.5)	5.29E-01	2.56E-02	2.77E-01	3.55E-01	6.26E-03	1.29E-04
	Infeasible region (0.6,0.6,0,8)	2.87E-01	1.53E-03	1.44E-01	2.01E-01	5.76E-03	2.49E-04
	Feasible region (0.5,0.5,0.7)	3.83E-01	1.68E-04	1.91E-01	2.70E-01	4.98E-03	2.33E-04

Table 4. Results of SP calculations for each of the three algorithms run 30 times on the test function

Test function	Preference point	g-NSGAII		r-NSGAII		CPMOP-DM	
		Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
ZDT1	On the frontier (0.5,0.3)	9.97E-01	1.56E-03	4.99E-01	7.03E-01	7.62E-04	1.79E-04
	Infeasible region (0.1,0.2)	7.28E-01	2.30E-02	3.75E-01	4.98E-01	7.57E-04	1.28E-04
	Feasible region (0.8,0.8)	8.69E-01	1.26E-02	4.41E-01	6.05E-01	6.94E-04	1.35E-04
ZDT2	On the frontier (0.6,0.64)	1.00E+00	1.12E-02	5.08E-01	7.03E-01	6.75E-04	9.30E-05
	Infeasible region (0.2,0.4)	7.36E-01	2.69E-02	3.81E-01	5.01E-01	6.87E-04	8.50E-05
	Feasible region (0.9,0.9)	6.74E-01	4.72E-02	3.61E-01	4.43E-01	6.81E-04	1.02E-04
ZDT3	On the frontier (0.24,0.28)	9.99E-01	4.94E-05	4.99E-01	7.06E-01	6.70E-04	1.14E-04
	Infeasible region (0.2,0.2)	9.20E-01	1.85E-02	4.69E-01	6.37E-01	5.27E-04	1.39E-04
	Feasible region (0.5,0.6)	8.20E-01	1.63E-02	4.18E-01	5.68E-01	7.33E-04	1.35E-04
ZDT6	On the frontier (0.6,0.64)	1.23E+00	3.89E-01	8.09E-01	5.94E-01	7.21E-04	8.60E-05
	Infeasible region (0.3,0.2)	9.69E-01	3.51E-01	6.60E-01	4.36E-01	5.64E-04	3.37E-04
	Feasible region (0.7,0.8)	9.81E-01	1.67E-02	4.98E-01	6.81E-01	1.41E-03	3.79E-03
DTLZ2	On the frontier (0.5,0.7,0.5)	1.00E+00	1.62E-02	5.10E-01	6.99E-01	3.64E-03	2.63E-04
	Infeasible region (0.2,0.3,0.4)	6.01E-01	2.45E-02	3.12E-01	4.07E-01	3.61E-03	2.70E-04
	Feasible region (0.7,0.7,0,8)	6.58E-01	6.49E-03	3.32E-01	4.61E-01	3.64E-03	2.70E-04
DTLZ4	On the frontier (0.3,0.4,0.5)	9.93E-01	2.89E-03	4.98E-01	7.00E-01	2.74E-03	2.30E-04
	Infeasible region (0.6,0.6,0,8)	5.58E-01	2.34E-01	3.96E-01	2.29E-01	3.96E-03	2.00E-04
	Feasible region (0.5,0.5,0.7)	7.98E-01	1.34E-02	4.06E-01	5.55E-01	3.81E-03	1.70E-04



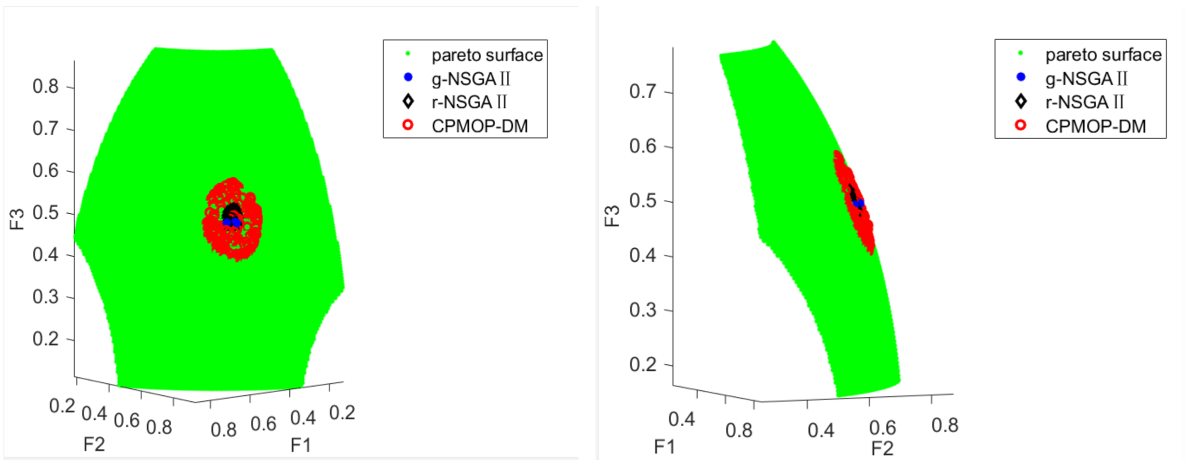
(a) ZDT1 feasibility region

(b) ZDT1 infeasibility region

(c) ZDT2 frontier

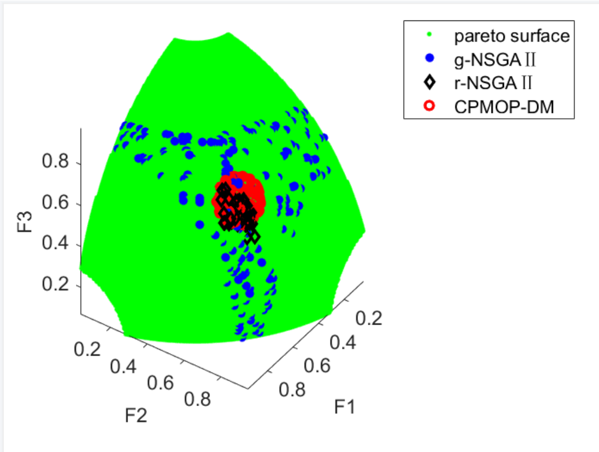
(d) ZDT2 infeasibility region

Fig. 4. The preference solutions from running the 3 algorithms on the ZDT test function

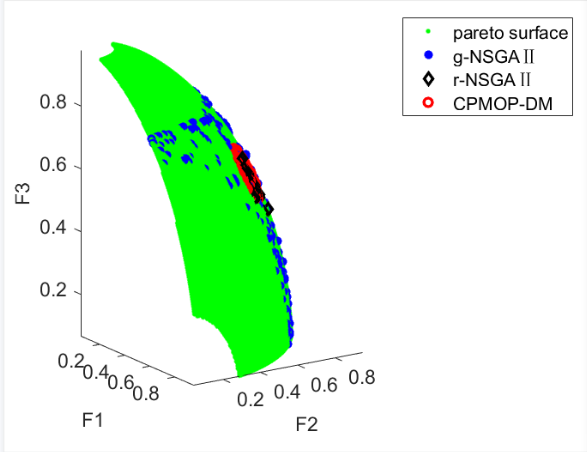


(a) Front view with preference point on the leading edge

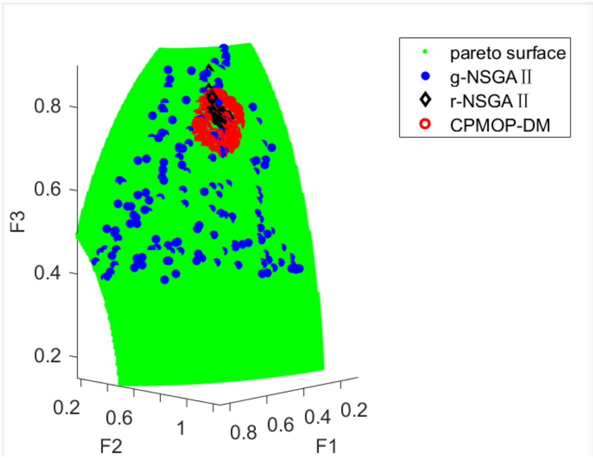
(b) Side view with preference point at the leading edge



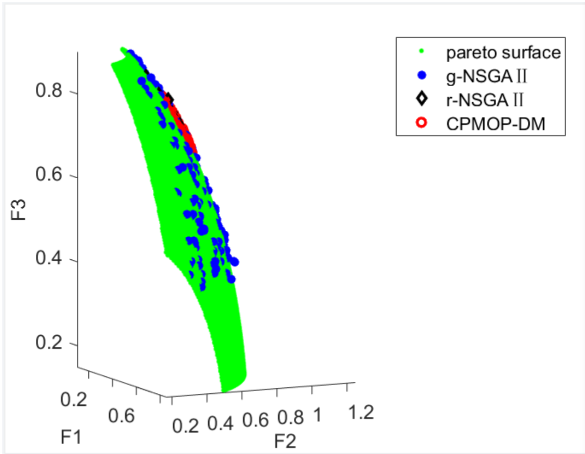
(c) Orthographic view of a preference point in a feasible region



(d) Side view of a preference point in a feasible region



(e) Orthographic view of a preference point in an infeasible region



(f) Side view of a preference point on an infeasible region

Fig. 5. The set of preference solutions obtained by running the 3 algorithms on the DTLZ2 test function

5 Conclusion

This paper proposes the Cubic Chaos Preference Multi-Objective Optimization Algorithm with Adaptive Dual-mode Mutation (CPMOP-DM). The population is initialized using CCM for uniform distribution across the search space. A DFPE balances exploration and exploitation while adjusting the preference region dynamically, ensuring solution set diversity. The DM combines binary crossover mutation and differential mutation to guide the search towards global optima and thoroughly explore the search space. Compared to traditional preference-based multi-objective optimization algorithms, CPMOP-DM excels in most cases on IGD, SP, and GD metrics, demonstrating superior diversity, convergence, and distribution.

The algorithm proposed in this study shows significant advantages in addressing multi-objective optimization problems. However, there are still some limitations. Particularly, its performance is not satisfactory when dealing with the ZDT3 test function. ZDT3 function exhibits nonlinearity and multi-modality, posing higher demands on the algorithm. Therefore, improving its performance in handling similar problems is an important direction for future research.

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. GJJ218505.The work of Wenliang Cao was supported the Dongguan Science and Technology of Social Development Program under Grant No. 20221800905842, and the project for the innovative platform in industry and education integration for higher vocational college of guangdong province-the innovative platform in industry and education integration of testing technology for intelligent terminal product under Grant No. 2020CJPT014, and the Special for key fields of colleges and universities in Guangdong Province under Grant NO. 2024ZDZX1092.

References

[1] Q.H. Gu, J.L. Luo, X.X. Li, Evolutionary Algorithm Based on Niche for Multi-Objective Optimization, Computer Engineering and Applications 59(1)(2023) 126-139.
[2] W.J. Luo, L. Shi, X. Lin, C.A. Coello Coello, The ĝ-dominance Relation for Preference-Based Evolutionary Multi-Objective Optimization, IEEE Congress on Evolutionary Computation (2019) 2418-2425.
[3] G. Sun, J.J. Wu, H. Chen, J. Li, S.Y. Xu, Hidden Preference-based Multi-objective Evolutionary Algorithm Based on Chebyshev Distance, Computer Science 49(6)(2022) 297-304.
[4] Y.B. Dai, H.T. Chen, A Preference Multi-objective Particle Swarm Optimization Algorithm by Hybrid Guidance, Control Engineering of China 26(3)(2019) 549-554.
[5] L.B. Said, S. Bechikh, K. Ghedira, The r-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making, IEEE Transactions on Evolutionary Computation 14(5)(2015) 801-818.
[6] Q.W. Xie, L.L. He, Many-object optimization algorithm with preference based on decomposition, Intelligent Computer and Applications 10(9)(2020) 81-85.
[7] H. Tian, L. Ke, Preference-Based Multi-objective Optimization with Gaussian Process, IEEE International Conference on Systems, Man, and Cybernetics (SMC) (2023)1148-1153.
[8] Y.M. Zhang, B. Atasoy, R.R. Negenborn, Preference-Based Multi-Objective Optimization for Synchromodal Transport Using Adaptive Large Neighborhood Search, Transportation Research Record 2676(3)(2022) 71-87.
[9] X.Y. Wen, Q.B. Song, Y.J. Qian, D.P. Qiao, H.Q. Wang, Y.Y. Zhang, H. Li, Effective Improved NSGA-II Algorithm for Multi-Objective Integrated Process Planning and Scheduling, Mathematics 11(16)(2023) 3523.
[10] W.Y. Long, H.C. Dong, P. Wang, Y. Huang, J.L. Li, X.B. Yang, C.B. Fu, A constrained multi-objective optimization algorithm using an efficient global diversity strategy, Complex & Intelligent Systems 9(2)(2023) 1455-1478.
[11] P.P. Nie, G.C. Wang, Y. Wang, Necessary and Sufficient Conditions for Pareto Optimal Solution of Backward Stochastic System With Application, IEEE Transactions on Automatic Control 68(11)(2023) 6696-6710.
[12] C. Yan, R.X. Wang, R. Sheng, Improved honey badger algorithm for dual population collaborative evolution, Application Research of Computers 41(3)(2023) 736-745.
[13] Y.B. Shen, A.D. Lei, X.Y. Ying, G. Yang, Y.Y. Wang, Y.Z. Luo, Static multi-objective control of active beam string structures based on improved genetic algorithm, Spatial Structures 28(1)(2022) 37-45.
[14] P.P. Zhao, L.P. Wang, Z.L. Fang, X.T. Pan, Q.C. Qiu, Preference-based multi-objective evolutionary algorithm with linear combination scalarizing function and reference point adjustment, Applied Soft Computing 153(2024) 111296.
[15] H.Y. Sun, H.X. Yang, S.Y. Tao, Optimization of the Number, Hub Height and Layout of Offshore Wind Turbines, Journal of Marine Science and Engineering 11(8)(2023) 1566.
[16] H. Huang, G. Liu, H.Z. Li, Research on differential evolution algorithm with multiple mutation strategies, Changjiang Information & Communications 36(4)(2023) 51-54.
[17] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation 8(2)(2000) 173-195.
[18] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, Evolutionary Multiobjective Optimization: Theoretical Advances and Applications (2005) 105-145.
[19] Y. Tian, W.J. Zhu, X.Y. Zhang, Y.C. Jin, A practical tutorial on solving optimization problems via PlatEMO, Neurocomputing 518(2023) 190-205.
[20] J.H. Zhang, Z.F. Du, J. Zou, S.X. Yang, A weight vector generation method based on normal distribution for preference-based multi-objective optimization, Swarm and Evolutionary Computation 77(2023) 101250.

[21] H.N. Liang, Y.J. Wang, W.X. Lin, W.F. Chen, Preferential multi-objective optimization algorithm based on two-stage hybrid guidance, Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition) 34(5) (2022) 836-848.
[22] X. Feng, Z.Y. Ren, A.Q. Pan, J.C. Hong, Y.H. Tong, A multi-preference-based constrained multi-objective optimization algorithm, Swarm and Evolutionary Computation 83(2023) 101389.