Constrained Multi-Objective Optimization Algorithm Based on the Boundary Value Reservation Strategy

Hejun Xuan, Deming Zhou, Yan Ding, Ruiyan Ma, Canyu Zhu, and Yan Feng

***Abstract*—Constrained multi-objective optimization problems are ubiquitous in real life. However, the presence of constraints makes the feasible domain complex, discontinuous and narrow. Consequently, solving multi-objective optimization problems becomes exceptionally challenging. Existing methods struggle to efficiently find the Pareto front (PF) with both good convergence and uniformity. To address this issue effectively, this paper proposed a two-stage constrained multi-objective optimization algorithm based on a three-population evolutionary algorithm (TPEA). In the first stage, the TPEA algorithm is employed to search for approximately constrained Pareto front (CPF). In the second stage, a boundary value retention mechanism was designed to enhance the quality of CPF. Additionally, a local search method with an adaptive perturbation factor is introduced to generate offspring, thereby increasing the probability of generating a boundary point. To select the better individuals for the next generation, a forward-looking environmental selection strategy was proposed. The proposed algorithm is compared with five algorithms. In the experiments, 38 benchmark test functions and three real-world problems are employed. Experimental results demonstrate that the proposed algorithm can obtain the CPFs with better convergence, uniformity, and diversity than five compared algorithms.**

***Index Terms*—constrained multi-objective, two-stage, boundary value reservation, environmental selection, forward looking.**

I. Introduction

MULTI-OBJECTIVE Optimization Problems (MOPs) need to consider multiple objective functions simultaneously in the optimization process. Unlike traditional single-objective optimization problems (SOPs) [1], MOPs involve optimizing a vector of decision variables to achieve the best possible outcomes for each objective function, often under specific constraints or to strike a balance and find compromise solutions across various objectives. These problems are prevalent in real-world scenarios and typically feature one or more equality or inequality constraints, such as structural optimization [2], [3], [4], engineering design [5], [6], [7]. In general, the mathematical definition of CMOPs is described as:

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ s.t. \\ \quad g_i(x) \leq 0, i = 1, 2, \cdots, p; \\ \quad h_i(x) = 0, i = p+1, p+2, \cdots, q \end{cases} \tag{1}$$

where $x = (x_1, x_2, \cdots, x_D)$ is a D-dimensional decision variable within the decision space. The objective function $F(x)$ includes an m-dimensional objective vector. $g(x)$ and $h(x)$ represent p inequality constraints and $q - p$ equality constraints, respectively.

Given that multiple objectives are often in conflict with each other, optimizing one objective may adversely affect other objectives. Therefore, finding a solution that outperforms all other objectives on all objectives is usually unattainable. Thus, the goal of solving a MOP is to find a set of solutions called the Pareto set (PS). These solutions cannot improve one of the objectives without affecting the others. The PS is then mapped to the objective space Pareto Front (PF). This front represents a curve, surface, or hyperplane in the objective space. Constrained multi-objective optimization problems (CMOPs) present unique challenges compared to their unconstrained counterparts. In CMOPs, the goal is to find a set of solutions which cannot be improved in one objective without sacrificing the others. CMOPs are notably more difficult to solve efficiently, as shown in Fig. 1 (the gray area denotes the feasible domain of the MOP). The constraints in Fig. 1(a) lead to a discontinuous feasible domain. Fig. 1(b) demonstrated that the effect of the constraints narrow feasible domain. the discrete feasible domain is shown in Fig. 1(c).

To address these issues, this paper proposed a two-stage constrained multi-objective optimization algorithm based on a boundary value retention strategy. The key contributions of this work are as follows:

- A two-stage CMOEA based on a boundary value preservation strategy was proposed. The algorithm divided the whole evolutionary process into two stages. In the first stage, the TPEA was used to search for the approximate CPF. In the second stage, the boundary value preservation based evolutionary approach was used to improve the accuracy of the CPF.

Manuscript received June 2, 2024; revised December 10, 2024. This work was supported by National Natural Science Foundation of Henan Province (No. 232300420424), Henan Province Key Research and Development Project (No. 241111212200), Science and Technology Department of Henan Province (No.242102211070).

Hejun Xuan is an associate professor at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: xuanhejun0896@xynu.edu.cn)

Deming Zhou is a graduate student at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: zhdm2015@126.com)

Yan Ding is a graduate student at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: 1151812253@qq.com)

Ruiyan Ma is an undergraduate student at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: 690454390@qq.com)

Canyu Zhu is an undergraduate student at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: 417062097@qq.com)

Yan Feng is a professor at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (Corresponding author, email: fengyan_xynu@126.com)

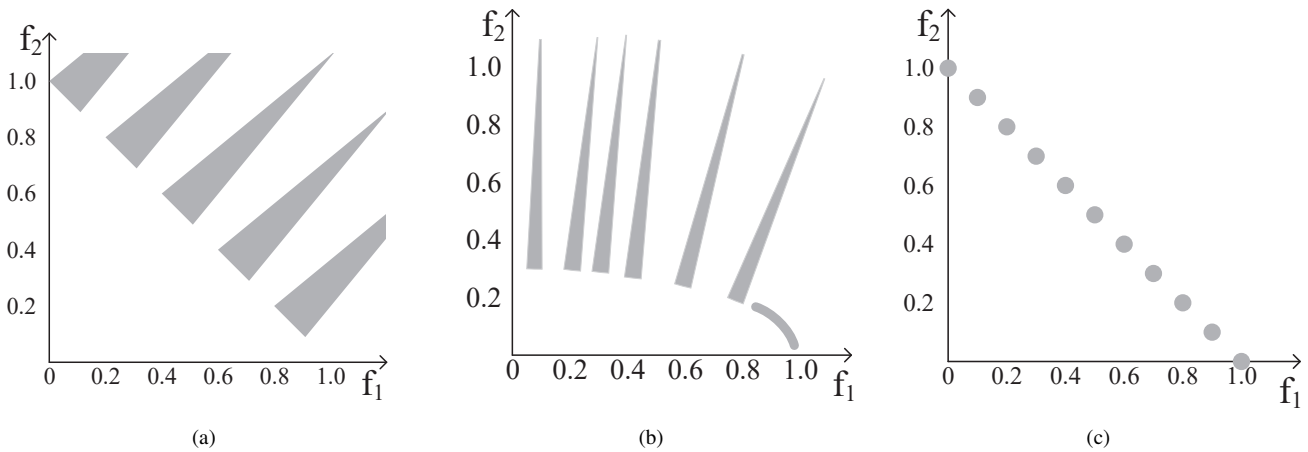


Fig. 1. The challenge of constrained multi-objective problems

- A local search method by modifying the genetic algorithm was proposed. An adaptive boundary value preservation strategy was designed. The strategy adaptively adjusts the boundary threshold of the boundary population based on the current size of the boundary population.
- A forward-looking environmental selection strategy was proposed to improve the original environmental selection strategy. This method uses the nearest-neighbor information to evaluate the crowding degree of an individual and further uses the hyper-volume information to evaluate in the case of the same nearest-neighbor information.

II. Related Works

The evolutionary optimization framework is recognized as one of the popular for solving multi-objective optimization. The evolutionary optimization framework mainly includes the population generation method and environmental selection method.

A. Population Generation Methods

Genetic Algorithm (GA) stands as one of the classical methods for swarm intelligent optimization. Notably, the NSGA-II-CDP algorithm introduced by K. Deb [8]. In addition, some other algorithms are proposed, such as SPEA2 [9], PESA-II [10], C-TAEA [11], c-DPEA [12], and CMEGL [13], all employ GA as the individual generation method.

Differential Evolution (DE) conceived by Rainer Storn and Kenneth Price in 1995[14]. For instance, algorithms like PPS-MOEA-D [15], the Trip algorithm [16], and the ICMA algorithm [17] utilize DE to generate individuals. Furthermore, numerous variants of DE algorithms have emerged, including DE/rand/1 [18], DE/best/1 [19], and DE/current-to-best/1 [20]. Additionally, Liang et al[21] proposed the DE/current-to-exemplar/1 variant strategy. It aims to enhance the probability of generating high-quality individuals around a standardized individual by leveraging individuals with excellent quality and low density.

However, the population generation methods used by these algorithms are almost all based on global search algorithms. Thus, the search accuracy is limited. To solve this problem, this paper modified the GA algorithm and proposed a local search-based population generation strategy. The proposed algorithm can produce more solutions located near the constrained boundary during the evolutionary process.

B. Environmental Selection Strategy

In environmental selection, researchers usually use the crowding degree as the evaluation criterion for environmental selection. Since that the distribution of the decision space has little impact on the performance metrics of the algorithm, researches generally pay more attention to the crowding degree of the objective space. The perimeter of the rectangle enclosed by the point to be evaluated and the two nearest neighbors was first proposed in the literature [22]. The smaller the perimeter indicates the higher the crowding degree of the point. This method begins by performing non-dominated sorting of the population. Next, it sorts the population based on the objective values in a specific dimension. The density value of the first point and the last two points is then set to the maximum value. For the second-to-last point, the density value is determined by the perimeter of the rectangle formed by the nearest previous point and the next point. Finally, the density values are sorted in descending order, and the top NP points are selected for retention (where NP represents the population size). However, this method only makes use of the information related to the point to be measured and its nearest neighbors, and does not consider the problem from the height of the whole population. For example, when an algorithm reaches the late stage of evolution when the distribution of the population is relatively homogeneous, this strategy can only simply retain the first few individuals without considering the overall distribution of the population.

SPEA2 [9] uses the Euclidean distances between each point in the objective space and the rest of the points as a basis for density estimation. Then, some recently proposed algorithms, such as the CCMO [23], c-DPEA [12], CMGEL [13], MCCMO [24]. This method is used as a reference by comparing the distance to the point with its third and fourth distance. However, this information does not reflect the crowding of the point well ensure a uniform distribution of the final population. The density estimation method proposed in PSEAII [10] divided the entire objective space into uniform subregions and randomly selects one point at a time from the most congested subregion to be deleted. Due

to the increasing performance of CMOEA, the distribution of solutions will tend to be uniform in the later stages of evolution.

In addition, this paper briefly reviewed the constraint processing technology for CMOP. The more mainstream constraint processing method is to treat constraints as additional targets so that CMOP is converted into UMOP. A multi-objective evolutionary optimization algorithm (MOEA) can then be employed to solve the transformed problem. Generally speaking, there are two different ways to handle this. 1) Treat each constraint as an optimization objective, as shown in Eq. (2). 2) Combine all constraints into an optimization objective, as shown in Eq. (4). This article adopted the method of Eq. (4).

$$\min F(x) = (f_1(x), \ldots, f_m(x), CV_1(x), \ldots, CV_q(x)) \quad (2)$$

where

$$CV_i = \begin{cases} \max(0, g_i(x)), i = 1, 2, \cdots, p \\ \max(0, |g_i(x) - \varepsilon|), i = p+1, 2, \cdots, q \end{cases} \quad (3)$$

$$\min F(x) = (f_1(x), f_2(x), \ldots, f_m(x), CV(x)) \quad (4)$$

where

$$CV = \sum_{i=1}^{q} CV_i(x) \quad (5)$$

Although MOEAs have been developed to enhance the search efficiency for solving CMOPs, many of them rely on a combination of constraint-handling techniques (CHT) and environmental selection strategies. However, they often struggle when confronted with complex feasible domains. Specifically, the existing CMOEAs face several challenges: 1) The probability of producing good-quality individuals is low. 2) The retention of good-quality individuals is not ensured in environmental selection. 3) The methodology for assessing population selection for individual crowding in environmental selection is inaccurate.

III. Proposed Algorithm

The precise boundary value search (PBS) method aims to locate an exact boundary close to an approximate boundary. However, it becomes ineffective when the approximate boundary is unknown. Typically CMOEA searches for approximate CPF without performing an exact search. To enhance the accuracy of CMOEAs, integrating PBS and TPEA into a two-phase search framework called BPCMO was proposed. The basic BPCMO process consists of two phases: the TPEA search phase and the PBS search phase. The PBS search phase incorporates three key components: enhanced local search method (Boundary-based Local Search, BLS), adaptive boundary value retention mechanism (BRM), and a forward-looking environmental selection (FES). The BPCMO framework addresses the limitations of traditional CMOEAs by combining the strengths of PBS and TPEA. The integration of BLS, BRM,and FES in the PBS search phase further enhances the precision and effectiveness of the algorithm.

A. Improved Local Search Method

In proposed algorithm, the local search method is proposed. In addition, the crossover process in the genetic algorithm was deleted to just keep the mutation process. In Fig. 2, the process of improved local search strategy is shown. 1) Fig. 2(a) denotes a population characterized by a size of NP and a decision vector with D dimensions, where $NP = 5$, $D = 5$. 2) A matrix $SparseM_{NP\times D}$ is generated randomly. If $SparseM[i][j] = 1$ indicates that the change is made to the jth dimension of the ith individual, similarly, $SparseM[i][j] = 0$ represents the jth dimension of the ith individual is not perturbed. Fig. 2(b) shows an example of the matrix $SparseM$. 3) We use UBL_B and UBL_P to represent upper and lower bounds for each dimension in the archive and individuals in the current population. Suppose that $UBL_B = (0.8758, 0.2241, 0.2348, 0.1382, 0.3558)$ and $UBL_P = (1.0000, 0.2241, 0.1941, 0.1403, 0.3552)$ in current generation. 4) α_{pro} is generated randomly in $(0, 0.5 + gen/(2maxGen))$, where gen and $maxGen$ are current iterations and maximum iterations, respectively. α_{fac} is generated in $(0, \alpha)$, where α is a relatively small constant. 5) ULB is calculated according to Eq. (6). Then, ULB is copied NP times to get a matrix $ULBM_{NP\times D}$. 6) Matrix $PertM$ is calculated by $PertM = ULBM \odot SparseM$. Finally, the offspring are obtained according to $OffSping = PopDec + PerM$, as shown in Fig. 2(c) and Fig. 2(d). The framework of improved local search method BLS is shown in Algorithm 1.

$$ULB = \begin{cases} \alpha_{pro} \times ULB_B, rand() \le \alpha_{pro} \\ \alpha_{fac} \times ULB_P, otherwise \end{cases} \quad (6)$$

Algorithm 1: Improved local search method

Input: parental population $Population$, difference between upper and lower bounds of parental population decisions ULB_P; Difference between the upper and lower bounds of the boundary population decision ULB_B

Output: offspring population $Offspring$

```
Access to parental population decision variables PoPDec;
Randomly generate a sparse matrix with only 1 and 0 SparseM;
Generate a random number rand;
Generate a random number α_pro in (0, 0.5 + gen/(2 × maxGen));
if rand() ≤ α_pro then
    ULB = α_pro × ULB_B;
else
    ULB = α_pro × ULB_P;
end
PertM = SparseM ⊙ ULB;
Offspring = PopDec + PertM;
```

B. Adaptive Boundary Value Retention Mechanism

Adaptive boundary value retention mechanism (BRM) is a technique that utilizes constraint violations as the basis for

0.6720	0.8653	0.8785	0.8967	0.8402
0.6008	0.9740	0.7714	0.9170	0.8391
0.4532	0.9614	0.7989	0.8912	0.8435
0.5710	0.9740	0.7714	0.9165	0.8404
0.3817	0.9264	0.7692	0.9280	0.8722

(a) PopDec

0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	0	0	0	1
0	0	1	0	0

(b) SparseM

0	0	0	0.0006	0
0	0	0.0011	0	0
0	0.0010	0	0	0
0	0	0	0	0.0016
0	0	0.0011	0	0

(c) PertM

0.6720	0.8653	0.8785	**0.8973**	0.8402
0.6008	0.9740	**0.7725**	0.9170	0.8391
0.4532	**0.9624**	0.7989	0.8912	0.8435
0.5710	0.9740	0.7714	0.9165	**0.8420**
0.3817	0.9264	**0.7703**	0.9280	0.8722

(d) OffSping

Fig. 2. The schematic diagram of the BLS

evaluation. The pseudo-code for this mechanism is outlined in Algorithm 2. The key step of this algorithm described as:

Step 1: The BRM takes as input the boundary archive BP, the main population Population, the offspring populations Offspring, and the boundary threshold ε.

Step 2: The boundary archive, the main population, and the offspring populations are merged, and a de-duplication operation is performed on the merged populations.

Step 3: Archive the individuals that are located near the boundary of the constraint.

Step 4: If the archive size is larger than the archive size threshold, update the boundary threshold and perform the archive operation again with the new threshold.

Step 5: Finally, output the archive as well as the boundary threshold.

The distribution of individuals in the archive at each stage in BRM is shown in Fig. 3. Fig. 3(a) represents the distribution of individuals in the archive at the initial stage. Fig. 3(b) represents the result of re-archiving after reaching the archive size threshold and updating the boundary threshold. From Fig. 3(c), it can be seen that the mechanism allows the individuals in the archive to approach the constraint boundary as a whole, which in turn brings the population closer to the PF.

Algorithm 2: Adaptive boundary value retention mechanism

Input: boundary population BP, main population $Population$, offspring population $Offspring$, boundary threshold ϵ, Archival scale factor β, Adaptive scaling factor γ

Output: new boundary population BP; boundary threshold ϵ

Assuming that NP denotes the row number of $Population$, $ComPop$ denotes the concatenation of $Population$ and $Offspring$;
Perform a de-duplication operation on $CmoPop$;
Use Algorithm 3 to find the index $BPIndex$ of the bounded individual;
BP = $ComPop(BPIndex)$
Get the population size of the population BP, N_{BP}
while $N_{BP} > \beta \times NP$ **do**
    $\epsilon = \epsilon/\gamma$;
    Use Algorithm 3 to find the index $BPIndex$ of the bounded individual;
    $BP = BP(BPIndex)$;
end

C. Forward-looking Environmental Selection

Forward-looking environmental selection (FES) is an environmental selection method. As an example, three individuals are selected from five by Proposed environmental selection mechanism was shown in Fig. 4(a). First, the distance of each point from all other points is calculated according to Eq. (7), and the results are saved into a matrix as shown in Fig. 4(b). Subsequently, the point with the smallest distance to its immediate neighbors is determined by sorting. However, in the later stages of evolution, individuals are more evenly distributed in the objective space. So it may happen that many points have equal distances to their nearest neighbors. To address this situation, a method to assess the crowding of these points by comparatively calculating their hyper-volume relative to a reference point was proposed. The smaller the hyper-volume is, the more the solution needs to be removed. As in Fig. 4(c), the fourth column shows the hyper-volume of the points enclosing the polyhedron with the reference point. The maximum value of each dimension plus ε as the value of each dimension at the reference point was defined, where ε is a very small positive value. The calculation of the hyper-volume is shown in Eq. (8). Fig. 4(e) and Fig. 4(f) show the result after the sequential selection and second

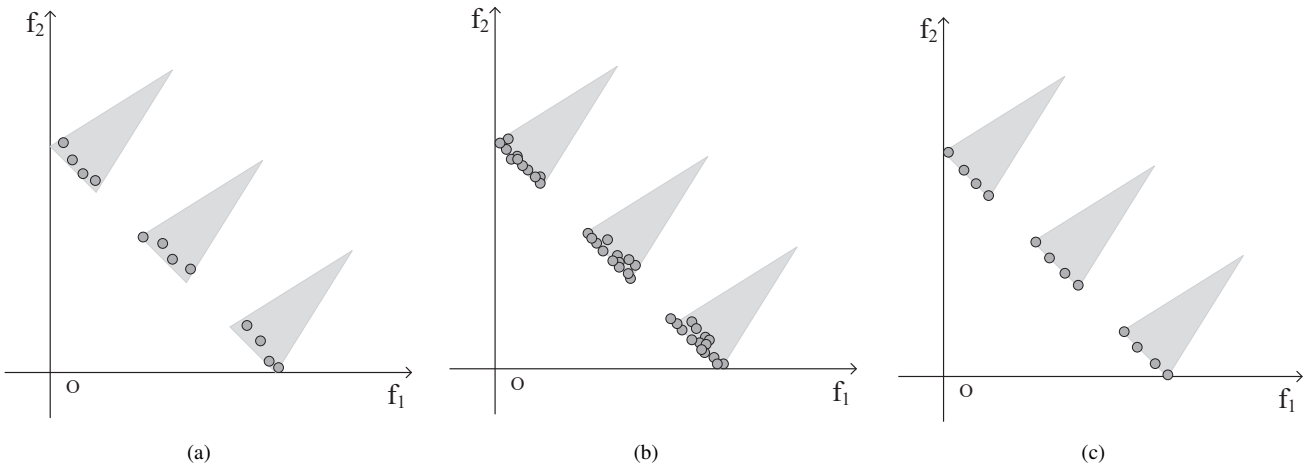


Fig. 3. The schematic diagram of the BRM

Algorithm 3: Boundary Individual Judgment Algorithm

Input: Constraint violation matrix of the population $PopCon$, boundary threshold ϵ

Output: Index of populations located near the boundary $BPIndex$

$BPIndex = \emptyset$. Assuming that row and col denote the row and column number of $PopCon$;
for $i = 1$ *to* row **do**
  for $j = 1$ *to* col **do**
    if $PopCon_{ij} < \epsilon$ **then**
      $BPIndex = [BPIndex, i]$;
    end
  end
end

selection, respectively. Fig. 5 demonstrated the results after the selection of the two environmental selection methods under the same conditions. It can be seen that the results selected by the method proposed in this paper are more evenly distributed.

$$ED_{i,j} = \begin{cases} \sqrt{\sum_{k=1}^{m} (f_{i,k} - f_{j,k})^2}, & i \neq j \\ \infty, & otherwise \end{cases} \quad (7)$$

$$HV_i = \prod_{j=1}^{m} (f_j^{max} + 0.1 - f_{i,j}), i = 1, 2, \ldots, NP; \quad (8)$$

IV. Experimental Setup

To evaluate the effectiveness of BPCMO, we conducted tests using three benchmark test sets. Firstly, the CDTLZ test set was utilized. It consists 10 test functions [20], [21]. C1_DTLZ1, DC1_DTLZ1, DC2_DTLZ1, and DC3_DTLZ1 have 7-dimensional decision variables, while the remaining 6 test functions have 12-dimensional decision variables. Secondly, the MW benchmark test set was employed, which includes 14 test functions with 15-dimensional decision variables [25]. Lastly, we employed the LIRCMOP test set, which comprises 14 test functions with 30-dimensional decision variables. For the purpose of comparison, five state-of-the-art algorithms ware selected, namely PPS [15], CTAEA [11], MTCMO [26], TSTI [27], and C3M [28].

A. Parameter Settings

The parameter settings for the five comparison algorithms are taken from their respective original literature. As for the proposed algorithms in this paper, the following parameter values were used: FirstStageFEs = 40,000, α = 0.9, ϵ = 0.01, and $\beta = 0.4$, $\gamma = 10$. To ensure a fair comparison, the population size (NP) was set to 100 for all problems in this study, and the maximum number of evaluations (maxFEs) was set to 70,000.

To evaluate the significance of the difference between the two methods, a Wilcoxon test with a significance level of 0.05 was conducted. The results of the comparison are indicated as follows: "+" or "-" denotes that the comparative algorithm performed better or worse than BPCMO, respectively. "=" signifies that BPCMO and the comparative algorithm performed similarly.

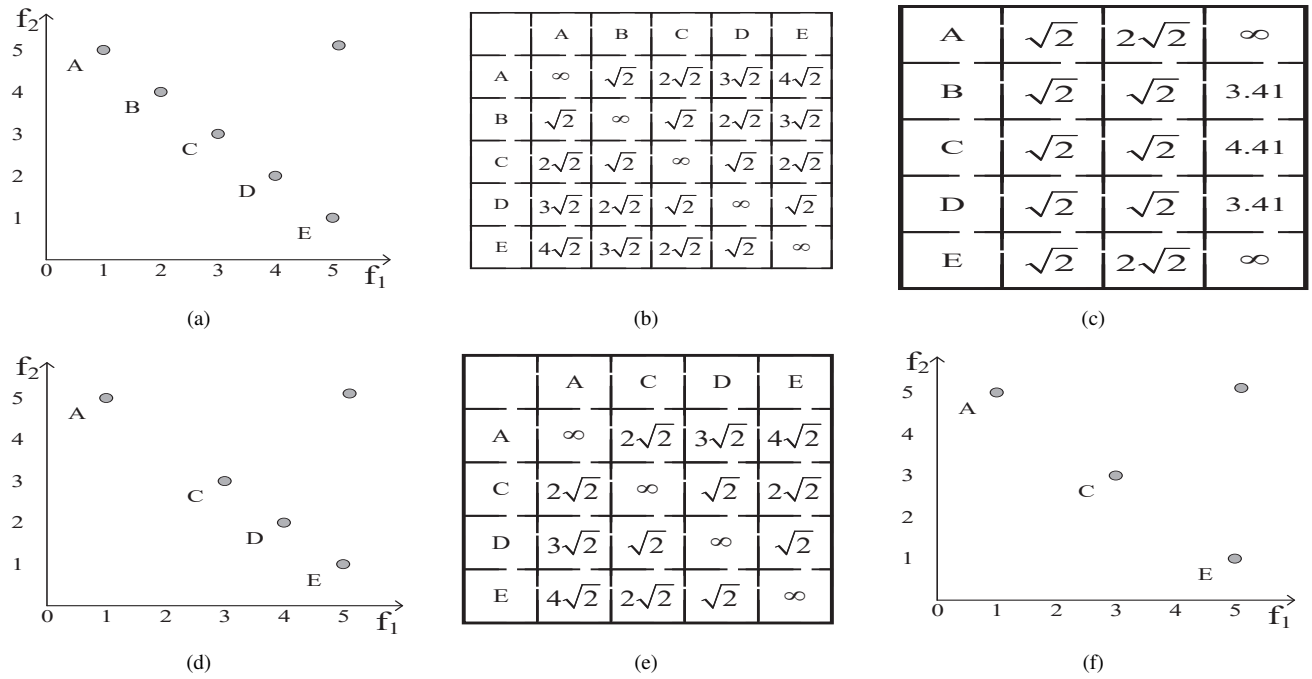
B. Performance Metrics

In this paper, three performance metrics are utilized: Inverse Generation Distance (IGD) [29], hypervolume (HV) [30], and IGDp [31]. IGD serves as a measure of the convergence of the proposed PF and true PF. The value of IGD lower is, the better performance of the algorithm is. HV is another widely adopted metric in multi-objective optimization. It primarily evaluates the convergence and diversity of the proposed PF. A higher HV value signifies better performance of the proposed PF. IGDp is used to assess the distribution of a set of solutions in the target space, checking that these solutions cover the entire objective space and are evenly distributed in the objective space.

V. Results and Discussion

A. Results on Benchmark

Table I to Table III show the results of the IGD, HV, and IGDp metrics for five comparison algorithms on 38 benchmarks, respectively. These test functions generally possess



	A	B	C	D	E
A	∞	$\sqrt{2}$	$2\sqrt{2}$	$3\sqrt{2}$	$4\sqrt{2}$
B	$\sqrt{2}$	∞	$\sqrt{2}$	$2\sqrt{2}$	$3\sqrt{2}$
C	$2\sqrt{2}$	$\sqrt{2}$	∞	$\sqrt{2}$	$2\sqrt{2}$
D	$3\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	∞	$\sqrt{2}$
E	$4\sqrt{2}$	$3\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	∞

A	$\sqrt{2}$	$2\sqrt{2}$	∞
B	$\sqrt{2}$	$\sqrt{2}$	3.41
C	$\sqrt{2}$	$\sqrt{2}$	4.41
D	$\sqrt{2}$	$\sqrt{2}$	3.41
E	$\sqrt{2}$	$2\sqrt{2}$	∞

	A	C	D	E
A	∞	$2\sqrt{2}$	$3\sqrt{2}$	$4\sqrt{2}$
C	$2\sqrt{2}$	∞	$\sqrt{2}$	$2\sqrt{2}$
D	$3\sqrt{2}$	$\sqrt{2}$	∞	$\sqrt{2}$
E	$4\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	∞

Fig. 4. The results of environmental selection strategies

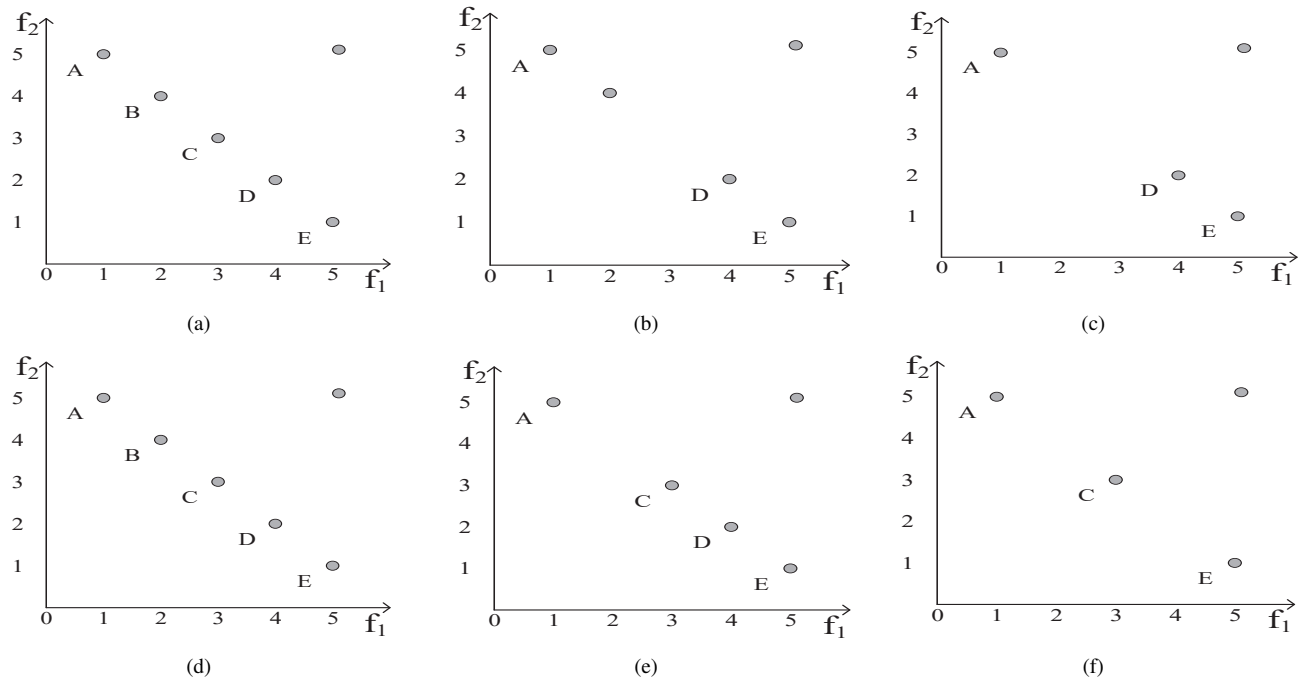
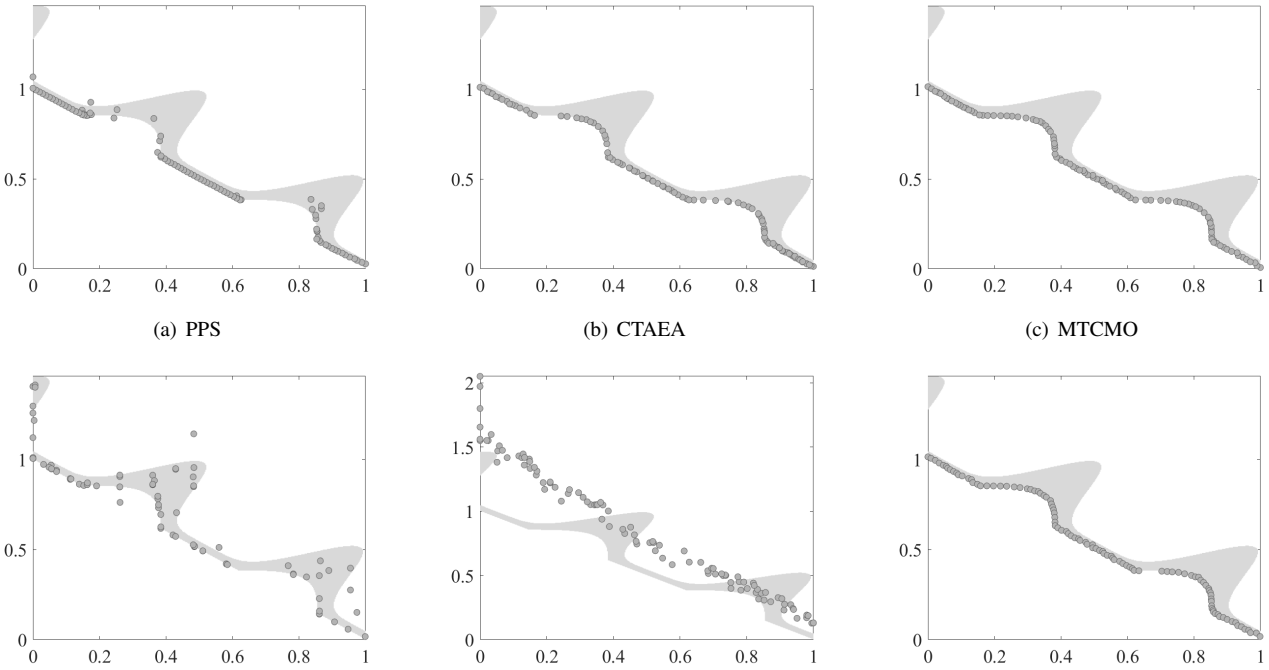


Fig. 5. The comparison results of two environmental selection strategies

large feasible regions but also contain numerous infeasible obstacles. As a result, algorithms must maintain diversity in the early stages while enhancing convergence in the later stages. The proposed BPCMO algorithm demonstrates the best performance on the MW test set, followed by TSTI and MTCMO. However, when facing two 3-objective type I problems, MW4 and MW14, the performance of BPCMO is not optimal. This indicates that there is still potential for improving the algorithm's effectiveness in handling such problem instances.

Fig 6 and 7 show the simulation experiments of the six algorithms on test problem MW3. The results show that the algorithms can basically search for the approximate front when the population evolves to the 25% stage. At the 75% stage, the algorithm proposed in this paper can obtain the best optimization result among all the algorithms.

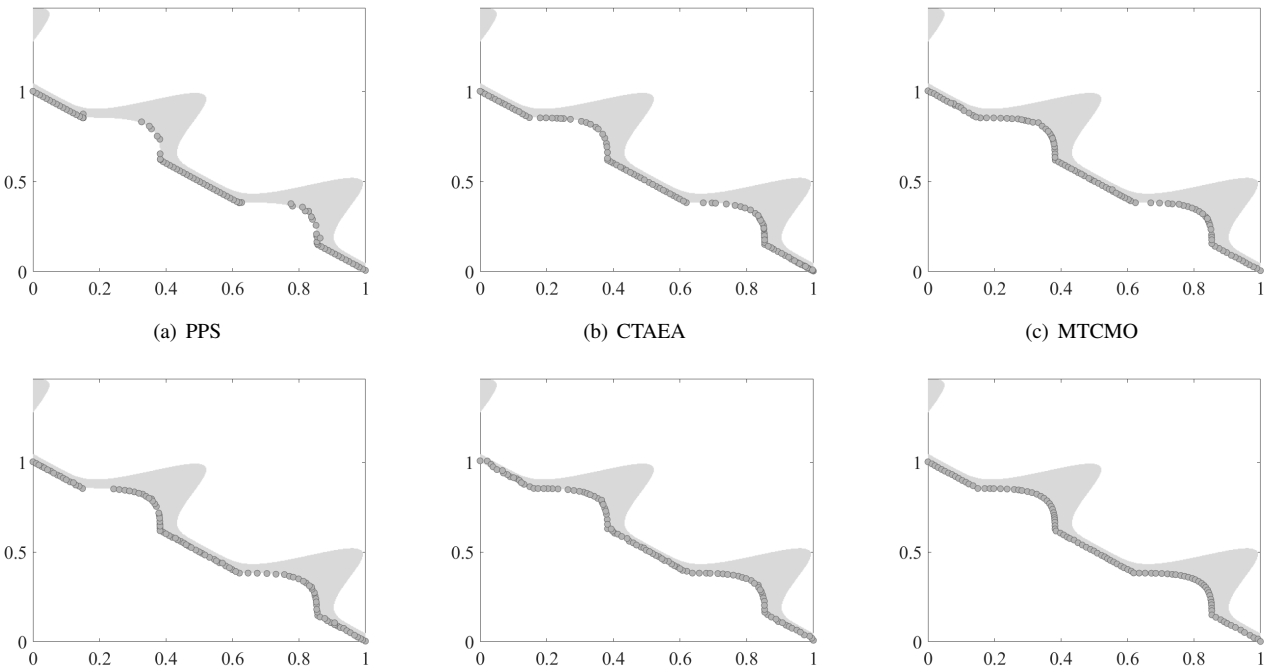
Upon analyzing the detailed comparison results, it becomes apparent that PPS and C3M are excellent in solving the LIRCMO problem. Specifically, PPS achieves optimal values for LIRCMO1 to LIRCMO6, while C3M obtains optimal values for LIRCMO7 to LIRCMO12. On the other hand, CTAEA, MTCMO, and TSTI demonstrate proficiency in addressing the DTLZ test set problem. CTAEA attains optimal values for C1_DTLZ3, DC2_DTLZ1, DC3_DTLZ1 and DC3_DTLZ3. MTCMO and TSTI achieve optimal values for DC1_DTLZ1 and DC1_DTLZ3, as well as C1_DTLZ1,



(a) PPS (b) CTAEA (c) MTCMO

(d) TSTI (e) C3M (f) BPCMO

Fig. 6. Population distribution of the six algorithms when the number of evaluations reaches 25%



(a) PPS (b) CTAEA (c) MTCMO

(d) TSTI (e) C3M (f) BPCMO

Fig. 7. Population distribution of the six algorithms when the number of evaluations reaches 75%

TABLE I
IGD STATISTICS FOR FIVE ALGORITHMS IN THE 38 BENCHMARK

Problem	PPS	CTAEA	MTCMO	TSTI	C3M	BPCMO
C1DTLZ1	2.5846e-2 (5.93e-4) −	2.3128e-2 (2.55e-4) −	1.9947e-2 (1.81e-4) +	2.0036e-2 (1.68e-4) ≈	2.2625e-2 (2.97e-3) −	2.0142e-2 (2.73e-4)
C1DTLZ3	1.6730e+0 (3.23e+0) −	4.5460e-1 (1.50e+0) +	5.8946e+0 (3.58e+0) −	5.8918e+0 (3.58e+0) −	1.2251e+0 (2.41e+0) ≈	6.8906e-1 (1.54e+0)
C2DTLZ2	5.5501e-2 (2.36e-3) −	5.6835e-2 (1.35e-3) −	4.2579e-2 (5.91e-4) ≈	4.2264e-2 (4.10e-4) +	4.8569e-2 (1.11e-3) −	4.2633e-2 (5.55e-4)
C3DTLZ4	1.9385e-1 (1.10e-1) −	1.1189e-1 (2.58e-3) −	9.5259e-2 (1.41e-3) −	9.6015e-2 (1.37e-3) −	1.1309e-1 (2.68e-3) −	9.2000e-2 (1.14e-3)
DC1DTLZ1	4.4630e-2 (5.73e-2) −	1.5169e-2 (2.36e-4) −	1.1443e-2 (8.01e-5) +	1.1513e-2 (1.87e-4) +	1.5937e-1 (1.74e-1) −	1.1776e-2 (2.14e-4)
DC1DTLZ3	4.3601e-1 (6.73e-1) −	4.2689e-2 (1.37e-3) +	3.3801e-2 (3.41e-4) +	4.4833e-2 (5.86e-2) +	1.9356e+0 (2.16e+0) −	1.3186e-1 (1.14e-1)
DC2DTLZ1	5.2325e-2 (5.43e-2) −	2.3138e-2 (2.38e-4) +	1.2448e-1 (6.56e-2) −	8.3822e-2 (7.33e-2) ≈	7.0421e-2 (6.89e-2) −	2.5105e-2 (1.93e-3)
DC2DTLZ3	4.7852e-1 (2.02e-1) ≈	3.8811e-1 (2.32e-1) ≈	5.6460e-1 (2.33e-3) ≈	5.6041e-1 (0.00e+0) ≈	3.4437e-1 (2.64e-1) ≈	5.8206e-1 (6.75e-2)
DC3DTLZ1	9.9755e-1 (1.41e+0) −	9.2394e-3 (3.19e-4) +	1.1245e-1 (1.06e-1) ≈	2.0074e-1 (1.27e-1) −	1.4225e+0 (1.11e+0) −	2.5167e-2 (6.95e-2)
DC3DTLZ3	3.1024e+0 (3.07e+0) −	4.3881e-2 (1.64e-2) +	1.2683e+0 (3.89e-1) −	1.8513e+0 (5.79e-1) −	3.3250e+0 (2.83e+0) −	6.1274e-1 (1.75e-1)
MW1	3.1609e-2 (1.06e-1) −	2.2057e-3 (1.01e-3) −	6.0839e-3 (2.24e-2) ≈	1.4582e-2 (2.64e-2) ≈	1.2380e-1 (1.69e-1) −	1.6311e-3 (2.35e-5)
MW2	1.8990e-1 (1.16e-1) −	1.5615e-2 (9.17e-3) −	2.1391e-2 (8.52e-3) −	2.1322e-2 (9.75e-3) −	9.1838e-2 (4.36e-2) −	4.1204e-3 (1.30e-3)
MW3	6.1517e-3 (5.39e-4) −	5.3001e-3 (4.36e-4) −	5.0945e-3 (5.25e-4) −	5.4703e-3 (4.39e-4) −	6.1054e-3 (3.66e-4) −	4.4992e-3 (2.83e-4)
MW4	6.1414e-2 (2.24e-3) −	4.6558e-2 (4.98e-4) −	4.0994e-2 (4.21e-4) ≈	4.0721e-2 (3.81e-4) ≈	6.8394e-2 (3.51e-2) −	4.0855e-2 (3.09e-4)
MW5	3.6102e-1 (3.59e-1) −	1.4556e-2 (4.21e-3) −	5.7509e-2 (1.91e-1) −	1.2958e-1 (2.48e-1) −	3.3520e-1 (3.57e-1) −	1.3412e-3 (6.56e-4)
MW6	5.7904e-1 (3.60e-1) −	1.2720e-2 (8.46e-3) −	1.7978e-2 (1.29e-2) −	5.1505e-2 (1.14e-1) −	3.3871e-1 (2.15e-1) −	3.0547e-3 (1.34e-3)
MW7	5.5529e-3 (4.66e-4) −	7.2720e-3 (7.21e-4) −	4.6341e-3 (3.62e-4) −	6.7413e-3 (8.42e-3) −	6.2939e-3 (8.31e-4) −	4.0193e-3 (1.61e-4)
MW8	1.6057e-1 (5.50e-2) −	5.3614e-2 (2.10e-3) −	4.5428e-2 (3.05e-3) −	5.3716e-2 (2.18e-2) −	1.1103e-1 (3.63e-2) −	4.2312e-2 (5.37e-4)
MW9	2.1166e-1 (2.79e-1) −	9.0664e-3 (1.11e-3) −	1.4849e-2 (3.50e-2) −	3.7572e-2 (1.28e-1) −	4.2045e-1 (3.24e-1) −	3.9742e-3 (4.59e-5)
MW10	3.3785e-1 (2.30e-1) −	1.4474e-2 (1.07e-2) −	3.5362e-2 (3.52e-2) −	5.2767e-2 (4.68e-2) −	3.7560e-1 (1.74e-1) −	6.8599e-3 (4.93e-3)
MW11	7.3802e-3 (3.84e-4) −	1.5400e-2 (2.65e-3) −	6.0384e-3 (1.48e-4) −	6.1561e-3 (1.35e-4) −	7.5027e-3 (4.37e-4) −	5.8613e-3 (1.39e-4)
MW12	1.9281e-1 (2.90e-1) −	7.9689e-3 (6.60e-4) −	3.0353e-2 (1.40e-1) −	2.8417e-2 (9.08e-2) −	5.1904e-1 (2.96e-1) −	4.5222e-3 (8.22e-5)
MW13	4.2916e-1 (3.08e-1) −	4.7709e-2 (2.65e-2) −	7.9225e-2 (3.73e-2) −	1.5458e-1 (6.83e-2) −	2.3007e-1 (1.12e-1) −	3.0816e-2 (5.20e-2)
MW14	1.7419e-1 (5.13e-2) −	1.1125e-1 (3.77e-3) −	9.7785e-2 (2.25e-3) ≈	9.7482e-2 (2.01e-3) ≈	2.3852e-1 (6.99e-2) −	9.7378e-2 (2.36e-3)
LIRCMOP1	4.7749e-2 (4.21e-2) +	3.1610e-1 (9.24e-2) −	1.5473e-1 (1.98e-2) ≈	2.2099e-1 (2.21e-2) −	2.0685e-1 (7.73e-2) −	1.3851e-1 (3.56e-2)
LIRCMOP2	4.3748e-2 (4.26e-2) +	2.4876e-1 (8.50e-2) −	1.3740e-1 (1.48e-2) +	1.9229e-1 (1.61e-2) −	1.8012e-1 (6.71e-2) ≈	1.8094e-1 (2.93e-2)
LIRCMOP3	7.5838e-2 (4.91e-2) +	3.8825e-1 (1.27e-1) −	1.5776e-1 (2.71e-2) −	2.3091e-1 (2.68e-2) −	1.7742e-1 (8.56e-2) −	1.0507e-1 (2.29e-2)
LIRCMOP4	9.2386e-2 (6.92e-2) ≈	3.1467e-1 (6.19e-2) −	1.6105e-1 (2.10e-2) −	2.2425e-1 (2.57e-2) −	1.5533e-1 (8.72e-2) ≈	1.1688e-1 (2.13e-2)
LIRCMOP5	4.9255e-2 (2.17e-1) +	1.2008e+0 (1.71e-1) −	1.0738e+0 (3.28e-1) −	1.1281e+0 (2.69e-1) −	5.2444e-1 (5.58e-1) +	1.0561e+0 (3.51e-1)
LIRCMOP6	4.7612e-2 (3.88e-2) +	1.3462e+0 (7.05e-4) −	1.2371e+0 (2.81e-1) −	1.2114e+0 (3.09e-1) −	2.7722e-1 (4.52e-1) +	1.1032e+0 (3.59e-1)
LIRCMOP7	1.3864e-1 (2.41e-2) ≈	3.9344e-1 (5.31e-1) −	1.2170e-1 (2.76e-2) ≈	2.4108e-1 (3.93e-1) ≈	2.7757e-2 (4.28e-2) +	1.3414e-1 (4.69e-2)
LIRCMOP8	1.9970e-1 (6.76e-2) −	8.8659e-1 (6.82e-1) −	2.0990e-1 (5.01e-2) −	5.3207e-1 (5.86e-1) −	2.8246e-2 (4.04e-2) +	1.6012e-1 (9.41e-2)
LIRCMOP9	4.1962e-1 (8.51e-2) ≈	5.8932e-1 (7.02e-2) −	8.9272e-1 (1.35e-1) −	5.9186e-1 (9.64e-2) −	3.3649e-1 (1.05e-1) +	4.4888e-1 (1.02e-1)
LIRCMOP10	3.1685e-1 (1.34e-1) ≈	4.1622e-1 (6.38e-2) −	8.2989e-1 (1.58e-1) −	7.9116e-1 (1.57e-1) −	1.4427e-1 (8.35e-2) +	3.9801e-1 (1.83e-1)
LIRCMOP11	2.9391e-1 (1.16e-1) −	2.4011e-1 (9.20e-2) −	6.7627e-1 (1.85e-1) −	6.2916e-1 (1.51e-1) −	9.6605e-2 (7.86e-2) +	2.0100e-1 (1.17e-1)
LIRCMOP12	1.9212e-1 (9.53e-2) +	2.7355e-1 (1.61e-1) ≈	5.4271e-1 (1.48e-1) −	4.1799e-1 (1.12e-1) −	1.3405e-1 (7.92e-2) +	2.2282e-1 (6.27e-2)
LIRCMOP13	1.3213e-1 (4.78e-3) −	1.0963e-1 (3.06e-3) ≈	1.3161e+0 (1.87e-3) −	1.3151e+0 (1.31e-3) −	1.7621e-1 (4.95e-2) −	1.0899e-1 (1.62e-3)
LIRCMOP14	1.2245e-1 (7.88e-3) −	1.1103e-1 (1.25e-3) −	1.2730e+0 (1.87e-3) −	1.2328e+0 (2.15e-1) −	1.8016e-1 (1.63e-1) −	9.2895e-2 (7.70e-4)
$+/-/\approx$	6/27/5	5/30/3	4/26/8	3/28/7	8/26/4	

respectively. However, when considering the overall results, the algorithm proposed in this paper outperforms the others on all 38 test functions.

B. Test Results on Real Problems

To assess the effectiveness of the proposed algorithm on real-world problems, this paper employed CMOPs derived from three real-world scenarios: the simply supported I-beam design problem [32], front rail design [33], and the high-speed train collision energy management problem [2]. Among these, the front rail design aims to minimize impact forces while maximizing energy absorption. Its solution is constrained by three constraints pertaining to cross-section height and wall thickness.

In particular, the front rail design aims to minimize impact forces and maximize energy absorption, and its solution is limited by three constraints related to cross-section height and wall thickness. The process integration problem aims to minimize process equipment procurement costs and operating conditions, and its solution is limited by nine inequality constraints.

The experimental results (shown in Table IV)demonstrate that BPCMO achieves optimal solutions in all three practical problems. Particularly, it outperforms the other five algorithms significantly in the I-beam support problem and the energy management problem, while slightly surpassing them in the front rail design problem.

VI. CONCLUSION

This paper introduced a novel BPCMO algorithm aimed at enhancing the performance of CMOEA by refining the population generation method with an environmental selection approach. In BPCMO, the existing CMOEA is initially employed to approximate the CPF. In essence, BPCMO focuses on generating, retaining, and selecting high-quality and well-distributed PS. Experiments results demonstrated that the proposed algorithm outperforms others. However, BPCMO has some limitations:

1) The constraint Pareto front might not precisely align with the boundary of the feasible region, as the CPF is typically a subset of the concatenation of the UPF and the boundary of the feasible region. Hence, the current algorithm may not effectively handle problems with overlap between the UPF and the CPF.

2) Although the proposed offspring generation algorithm shortens the search step, the step size remains fixed. For instance, in the MW5 test problem, the ϵ value remained constant throughout the evolution, indicating a lack of generation of more accurate elite individuals. Consequently, the algorithm's applicability might be limited. These limitations were identified during testing, but the exact test problem remains unspecified.

REFERENCES

[1] H. Long, S. Liu, T. Chen, H. Tan, J. Wei, C. Zhang, and W. Chen, "Optimal reactive power dispatch based on multi-strategy improved

TABLE II
HV STATISTICS FOR FIVE ALGORITHMS IN THE 38 BENCHMARK

Problem	PPS	CTAEA	MTCMO	TSTI	C3M	BPCMO
C1DTLZ1	8.1665e-1 (4.49e-3) −	8.3596e-1 (2.46e-3) ≈	8.3868e-1 (4.52e-3) +	8.4194e-1 (6.03e-4) +	8.1699e-1 (1.41e-2) −	8.3244e-1 (8.01e-3)
C1DTLZ3	3.9320e-1 (2.05e-1) +	4.2866e-1 (1.78e-1) +	1.4863e-1 (2.51e-1) −	1.4848e-1 (2.50e-1) −	2.5573e-1 (2.16e-1) ≈	2.5904e-1 (1.92e-1)
C2DTLZ2	4.9727e-1 (3.17e-3) −	5.0660e-1 (1.72e-3) −	5.1629e-1 (1.20e-3) −	5.1651e-1 (1.77e-3) −	4.9261e-1 (3.00e-3) −	5.1882e-1 (1.16e-3)
C3DTLZ4	7.5400e-1 (3.22e-2) −	7.8483e-1 (1.19e-3) −	7.8968e-1 (1.04e-3) −	7.8942e-1 (8.65e-4) −	7.7805e-1 (1.64e-3) −	7.9406e-1 (7.99e-4)
DC1DTLZ1	5.5408e-1 (7.13e-2) −	6.2695e-1 (9.93e-4) −	6.3204e-1 (1.03e-3) +	6.3104e-1 (1.65e-3) +	3.5125e-1 (2.69e-1) −	6.3013e-1 (1.82e-3)
DC1DTLZ3	2.8527e-1 (1.42e-1) ≈	4.6064e-1 (2.68e-3) +	4.7351e-1 (1.21e-3) +	4.6592e-1 (2.23e-2) +	6.6153e-2 (1.22e-1) −	3.1717e-1 (1.34e-1)
DC2DTLZ1	7.4183e-1 (1.44e-1) −	8.3812e-1 (4.80e-4) +	5.7888e-1 (1.64e-1) −	6.8109e-1 (1.84e-1) ≈	7.0902e-1 (1.75e-1) −	8.2632e-1 (5.80e-3)
DC2DTLZ3	1.0544e-1 (1.98e-1) ≈	1.8984e-1 (2.38e-1) ≈	1.3124e-2 (1.25e-3) ≈	1.3523e-2 (0.00e+0) ≈	2.4640e-1 (2.62e-1) ≈	1.1067e-2 (5.51e-3)
DC3DTLZ1	1.5519e-1 (2.05e-1) −	5.2243e-1 (3.15e-3) +	2.8200e-1 (2.25e-1) ≈	1.5121e-1 (1.72e-1) −	5.2055e-2 (1.58e-1) −	4.9528e-1 (1.11e-1)
DC3DTLZ3	2.3010e-2 (7.31e-2) ≈	3.3306e-1 (2.32e-2) +	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	4.6243e-2 (9.95e-2) +	0.0000e+0 (0.00e+0)
MW1	4.6132e-1 (8.17e-2) −	4.8842e-1 (2.09e-3) −	4.8544e-1 (2.08e-2) −	4.7492e-1 (2.60e-2) ≈	3.8167e-1 (1.35e-1) −	4.8965e-1 (1.61e-4)
MW2	3.5250e-1 (1.23e-1) −	5.6204e-1 (1.55e-2) −	5.5225e-1 (1.31e-2) −	5.5235e-1 (1.48e-2) −	4.5659e-1 (5.36e-2) −	5.8178e-1 (2.18e-3)
MW3	5.4290e-1 (8.61e-4) −	5.4435e-1 (5.65e-4) ≈	5.4389e-1 (9.54e-4) −	5.4447e-1 (5.09e-4) ≈	5.4161e-1 (7.73e-4) −	5.4452e-1 (6.56e-4)
MW4	8.0256e-1 (6.81e-3) −	8.3817e-1 (3.55e-4) −	8.4168e-1 (4.51e-4) +	8.4176e-1 (4.39e-4) +	8.0114e-1 (3.67e-2) −	8.4070e-1 (4.68e-4)
MW5	2.0362e-1 (1.07e-1) −	3.1550e-1 (2.78e-3) −	3.0584e-1 (6.07e-2) −	2.7628e-1 (7.89e-2) −	2.0230e-1 (1.06e-1) −	3.2390e-1 (3.11e-4)
MW6	9.7351e-2 (9.36e-2) −	3.1008e-1 (1.18e-2) −	3.0595e-1 (1.66e-2) −	2.9076e-1 (4.11e-2) −	1.6787e-1 (6.53e-2) −	3.2787e-1 (2.42e-3)
MW7	4.1155e-1 (3.63e-4) −	4.0891e-1 (7.63e-4) −	4.1156e-1 (8.06e-4) −	4.1140e-1 (1.52e-3) −	4.1094e-1 (5.16e-4) −	4.1268e-1 (3.83e-4)
MW8	3.0683e-1 (9.13e-2) −	5.2606e-1 (9.54e-3) −	5.3465e-1 (1.24e-2) −	5.1206e-1 (4.35e-2) −	3.9807e-1 (6.19e-2) −	5.5378e-1 (5.78e-4)
MW9	2.4817e-1 (1.62e-1) −	3.9033e-1 (2.35e-3) −	3.8803e-1 (2.79e-2) −	3.7265e-1 (7.79e-2) −	1.4611e-1 (1.79e-1) −	4.0200e-1 (6.26e-4)
MW10	2.5623e-1 (1.07e-1) −	4.3890e-1 (1.18e-2) −	4.2094e-1 (2.47e-2) −	4.0798e-1 (3.10e-2) −	2.3540e-1 (7.93e-2) −	4.4890e-1 (7.72e-3)
MW11	4.4736e-1 (1.45e-4) −	4.4161e-1 (1.39e-3) −	4.4767e-1 (2.10e-4) −	4.4738e-1 (2.74e-4) −	4.4622e-1 (3.80e-4) −	4.4814e-1 (2.35e-4)
MW12	4.4013e-1 (2.43e-1) −	6.0069e-1 (6.56e-4) −	5.8476e-1 (1.10e-1) −	5.8699e-1 (6.76e-2) −	1.7246e-1 (2.42e-1) −	6.0578e-1 (2.61e-4)
MW13	2.6687e-1 (9.68e-2) −	4.5630e-1 (1.13e-2) −	4.4123e-1 (1.98e-2) −	4.0566e-1 (3.28e-2) −	3.4972e-1 (3.78e-2) −	4.6405e-1 (3.09e-2)
MW14	4.3294e-1 (1.46e-2) −	4.6595e-1 (2.95e-3) −	4.7294e-1 (2.47e-3) +	4.7338e-1 (1.63e-3) +	4.1908e-1 (2.11e-2) −	4.6873e-1 (2.41e-3)
LIRCMOP1	2.1491e-1 (2.44e-2) +	1.1212e-1 (2.59e-2) −	1.5851e-1 (7.49e-3) −	1.3771e-1 (9.74e-3) −	1.3996e-1 (2.78e-2) −	1.6544e-1 (1.56e-2)
LIRCMOP2	3.3969e-1 (2.42e-2) +	2.4274e-1 (3.79e-2) ≈	2.8513e-1 (1.10e-2) +	2.6252e-1 (1.01e-2) ≈	2.5627e-1 (3.72e-2) ≈	2.5975e-1 (1.30e-2)
LIRCMOP3	1.7761e-1 (1.97e-2) +	9.2154e-2 (1.79e-2) −	1.4526e-1 (1.06e-2) −	1.2081e-1 (9.30e-3) −	1.4200e-1 (2.58e-2) −	1.6367e-1 (1.02e-2)
LIRCMOP4	2.7528e-1 (3.06e-2) ≈	1.8508e-1 (2.27e-2) −	2.4674e-1 (9.47e-3) −	2.2094e-1 (1.31e-2) −	2.5022e-1 (3.51e-2) ≈	2.6542e-1 (1.06e-2)
LIRCMOP5	2.7941e-1 (5.28e-2) +	5.2607e-3 (2.88e-2) ≈	2.2698e-2 (5.21e-2) ≈	1.5070e-2 (4.65e-2) ≈	1.5776e-1 (1.34e-1) +	2.6805e-2 (6.21e-2)
LIRCMOP6	1.8375e-1 (1.18e-2) +	0.0000e+0 (0.00e+0) −	1.1440e-2 (2.98e-2) ≈	1.3776e-2 (3.15e-2) ≈	1.4036e-1 (7.14e-2) +	2.1627e-2 (3.45e-2)
LIRCMOP7	2.4181e-1 (6.92e-3) ≈	1.9809e-1 (8.55e-2) −	2.4801e-1 (8.26e-3) ≈	2.2683e-1 (6.23e-2) ≈	2.8514e-1 (1.75e-2) +	2.4622e-1 (1.36e-2)
LIRCMOP8	2.3173e-1 (1.57e-2) −	1.1826e-1 (1.04e-1) −	2.3027e-1 (8.86e-3) −	1.7747e-1 (9.04e-2) −	2.8479e-1 (1.65e-2) +	2.4312e-1 (1.93e-2)
LIRCMOP9	4.0670e-1 (5.07e-2) ≈	2.9155e-1 (3.87e-2) −	1.7128e-1 (7.70e-2) −	3.0577e-1 (7.25e-2) −	4.1223e-1 (3.56e-2) ≈	3.9837e-1 (5.69e-2)
LIRCMOP10	5.4308e-1 (7.33e-2) +	4.7426e-1 (3.92e-2) ≈	1.2819e-1 (1.11e-1) −	1.5510e-1 (9.45e-2) −	6.4084e-1 (4.10e-2) +	4.4739e-1 (1.43e-1)
LIRCMOP11	5.0439e-1 (8.17e-2) −	5.9013e-1 (4.45e-2) ≈	2.7961e-1 (1.07e-1) −	2.9664e-1 (8.30e-2) −	6.3707e-1 (5.22e-2) +	5.7026e-1 (8.18e-2)
LIRCMOP12	5.2333e-1 (4.98e-2) +	4.8828e-1 (5.59e-2) ≈	4.0435e-1 (5.87e-2) −	4.1223e-1 (5.09e-2) −	5.5345e-1 (4.31e-2) +	5.0607e-1 (3.49e-2)
LIRCMOP13	5.0038e-1 (8.89e-3) −	5.4597e-1 (1.77e-3) +	1.2854e-4 (1.23e-4) −	1.0793e-4 (1.23e-4) −	4.4701e-1 (5.20e-2) −	5.2526e-1 (2.04e-3)
LIRCMOP14	5.2266e-1 (1.31e-2) −	5.4569e-1 (1.02e-3) −	4.5353e-4 (2.66e-4) −	1.8999e-2 (1.01e-1) −	4.6996e-1 (9.19e-2) −	5.5896e-1 (9.73e-4)
$+/-/\approx$	8/24/6	6/24/8/	6/26/6	5/24/9	8/25/5	

aquila optimization algorithm," *IAENG International Journal of Computer Science*, vol. 49, no. 4, pp. 1249–1267, 2022.

[2] H. Zhang, Y. Peng, L. Hou, G. Tian, and Z. Li, "A hybrid multi-objective optimization approach for energy-absorbing structures in train collisions," *Information Sciences*, vol. 481, pp. 491–506, 2019.

[3] A. Mendo, J. Outes-Carnero, Y. Ng-Molina, and J. Ramiro-Moreno, "Multi-agent reinforcement learning with common policy for antenna tilt optimization," *IAENG International Journal of Computer Science*, vol. 51, no. 3, pp. 883–889, 2024.

[4] S. Tian, C. Dai, X. Wang, R. Wu, K. Bi, and T. Wang, "Robust optimization of vehicle scheduling for the secondary distribution of refined oil products under an active distribution model," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp. 1560–1573, 2024.

[5] F. Y. CHENG and X. S. LI, "Generalized center method for multiobjective engineering optimization," *Engineering Optimization*, vol. 31, no. 5, pp. 641–661, 1999.

[6] R. Zhou, Y. Zhang, X. Sun, H. Liu, , and Y. Cai, "Mswoa: Multi-strategy whale optimization algorithm for engineering applications," *Engineering Letters*, vol. 32, no. 8, pp. 1603–1615, 2024.

[7] C. Zhu, Z. Wang, X. Ma, and X. Yang, "Route optimization of rail transit travel chain under bounded rationality," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 1050–1057, 2023.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[9] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.

[10] D. W. Corne, N. R. Jerram, J. D. Knowles, and 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, pp. 283–290.

[11] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[12] M. Ming, A. Trivedi, R. Wang, D. Srinivasan, and T. Zhang, "A dual-population-based evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 739–753, 2021.

[13] K. Qiao, J. Liang, Z. Liu, K. Yu, C. Yue, and B. Qu, "Evolutionary multitasking with global and local auxiliary tasks for constrained multi-objective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 10, pp. 1951–1964, 2023.

[14] R. Storn and K. Price, "Differential evolution c a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, p. 341C359, 1997.

[15] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.

[16] F. Ming, W. Gong, L. Wang, and C. Lu, "A tri-population based co-evolutionary framework for constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 70, p. 101055, 2022.

[17] J. Yuan, H.-L. Liu, Y.-S. Ong, and Z. He, "Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 379–391, 2022.

[18] S.-C. Liu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A multiobjective framework for many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13 654–13 668, 2022.

[19] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution c an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[20] J. Liang, K. Qiao, K. Yu, B. Qu, C. Yue, W. Guo, and L. Wang, "Utilizing the relationship between unconstrained and constrained pareto fronts for constrained multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 53, no. 6, pp. 3873–3886, 2023.

[21] J. Liang, K. Qiao, C. Yue, K. Yu, B. Qu, R. Xu, Z. Li, and Y. Hu, "A clustering-based differential evolution algorithm for solving multimodal multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 60, p. 100788, 2021.

[22] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization

TABLE III
IGDP STATISTICS FOR FIVE ALGORITHMS IN THE 38 BENCHMARK

Problem	PPS	CTAEA	MTCMO	TSTI	C3M	BPCMO
C1DTLZ1	1.9175e-2 (4.91e-4) −	1.6305e-2 (1.90e-4) −	1.4169e-2 (1.70e-4) +	1.4065e-2 (1.24e-4) +	1.8453e-2 (2.39e-3) −	1.5300e-2 (5.39e-4)
C1DTLZ3	1.6498e+0 (3.24e+0) −	4.2505e-1 (1.51e+0) +	5.8867e+0 (3.60e+0) −	5.8840e+0 (3.59e+0) −	1.2081e+0 (2.42e+0) ≈	6.7194e-1 (1.54e+0)
C2DTLZ2	2.5091e-2 (7.63e-4) −	2.3631e-2 (7.49e-4) −	1.8963e-2 (6.12e-4) ≈	1.8685e-2 (4.59e-4) ≈	3.0212e-2 (1.09e-3) −	1.8849e-2 (5.53e-4)
C3DTLZ4	1.0069e-1 (3.86e-2) −	6.1262e-2 (1.87e-3) −	5.5940e-2 (1.80e-3) −	5.6354e-2 (1.39e-3) −	7.4351e-2 (2.54e-3) −	4.8831e-2 (1.24e-3)
DC1DTLZ1	3.1221e-2 (4.53e-2) −	1.0687e-2 (2.47e-4) −	8.2489e-3 (2.34e-4) +	8.5360e-3 (5.26e-4) +	1.5417e-1 (1.76e-1) −	9.0822e-3 (5.05e-4)
DC1DTLZ3	3.6734e-1 (6.77e-1) −	1.6339e-2 (9.15e-4) +	1.3262e-2 (6.21e-4) +	2.1394e-2 (3.46e-2) +	1.9239e+0 (2.16e+0) −	1.1780e-1 (1.19e-1)
DC2DTLZ1	4.7522e-2 (5.65e-2) −	1.6332e-2 (2.23e-4) +	1.2314e-1 (6.78e-2) −	8.0857e-2 (7.59e-2) ≈	6.8135e-2 (7.03e-2) −	2.1123e-2 (2.56e-3)
DC2DTLZ3	4.7249e-1 (2.13e-1) ≈	3.7159e-1 (2.50e-1) ≈	5.6395e-1 (2.25e-3) ≈	5.5996e-1 (0.00e+0) ≈	3.3434e-1 (2.73e-1) ≈	5.7897e-1 (5.90e-2)
DC3DTLZ1	9.8209e-1 (1.42e+0) −	6.4476e-3 (2.48e-4) ≈	1.1178e-1 (1.07e-1) ≈	1.9175e-1 (1.23e-1) −	1.4212e+0 (1.11e+0) −	2.3742e-2 (6.94e-2)
DC3DTLZ3	3.0720e+0 (3.09e+0) −	1.9655e-2 (8.49e-3) +	1.2682e+0 (3.89e-1) −	1.8513e+0 (5.79e-1) −	3.2996e+0 (2.85e+0) −	6.1042e-1 (1.73e-1)
MW1	2.2562e-2 (7.03e-2) −	1.5683e-3 (6.53e-4) −	4.0760e-3 (1.45e-2) −	9.5558e-3 (1.71e-2) ≈	8.7191e-2 (1.15e-1) −	1.2539e-3 (6.02e-5)
MW2	1.8979e-1 (1.16e-1) −	1.4602e-2 (9.41e-3) −	2.0669e-2 (8.81e-3) −	2.0577e-2 (1.01e-2) −	9.1796e-2 (4.36e-2) −	3.0251e-3 (1.31e-3)
MW3	3.6924e-3 (4.43e-4) −	2.8328e-3 (2.99e-4) ≈	3.2189e-3 (5.88e-4) −	2.8888e-3 (3.40e-4) ≈	4.5643e-3 (4.49e-4) −	2.7943e-3 (3.78e-4)
MW4	5.0741e-2 (2.60e-3) −	3.2571e-2 (3.41e-4) −	2.9188e-2 (3.76e-4) +	2.8963e-2 (3.42e-4) +	6.0388e-2 (2.28e-2) −	3.0097e-2 (4.45e-4)
MW5	1.7054e-1 (1.54e-1) −	1.3925e-2 (4.01e-3) −	2.6019e-2 (8.53e-2) −	6.6679e-2 (1.11e-1) −	1.6426e-1 (1.55e-1) −	1.2950e-3 (5.59e-4)
MW6	5.4907e-1 (3.26e-1) −	1.2264e-2 (8.86e-3) −	1.7770e-2 (1.31e-2) −	4.9155e-2 (1.06e-1) −	3.2262e-1 (2.03e-1) −	1.5949e-3 (1.70e-3)
MW7	2.7594e-3 (1.80e-4) −	3.9326e-3 (3.18e-4) −	2.3217e-3 (3.83e-4) −	2.5173e-3 (9.98e-4) −	2.8187e-3 (2.34e-4) −	1.6947e-3 (1.74e-4)
MW8	1.5586e-1 (5.69e-2) −	2.8360e-2 (5.32e-3) −	2.8294e-2 (6.81e-3) −	4.1325e-2 (2.50e-2) −	1.0651e-1 (3.77e-2) −	1.8159e-2 (5.26e-4)
MW9	1.9768e-1 (2.58e-1) −	7.1040e-3 (1.07e-3) −	1.2739e-2 (3.29e-2) −	3.3273e-2 (1.18e-1) −	3.8863e-1 (2.99e-1) −	2.1406e-3 (4.03e-5)
MW10	2.9331e-1 (1.86e-1) −	1.2759e-2 (1.04e-2) −	3.3449e-2 (3.27e-2) −	4.9644e-2 (4.33e-2) −	3.2897e-1 (1.39e-1) −	5.8713e-3 (5.19e-3)
MW11	3.6996e-3 (2.48e-4) −	8.3401e-3 (1.20e-3) −	2.6984e-3 (1.40e-4) −	2.7448e-3 (1.33e-4) −	4.5399e-3 (3.19e-4) −	2.0646e-3 (8.75e-5)
MW12	1.7720e-1 (2.76e-1) −	5.7950e-3 (5.05e-4) −	2.7437e-2 (1.34e-1) −	1.8781e-2 (6.02e-2) −	4.8794e-1 (2.87e-1) −	2.3699e-3 (5.41e-5)
MW13	4.1618e-1 (3.11e-1) −	2.5666e-2 (1.44e-2) −	4.7788e-2 (2.70e-2) −	1.0871e-1 (5.87e-2) −	2.2057e-1 (1.16e-1) −	2.2372e-2 (5.08e-2)
MW14	1.0384e-1 (2.61e-2) −	6.2679e-2 (3.66e-3) +	6.5268e-2 (3.72e-3) ≈	6.4443e-2 (2.19e-3) +	1.4797e-1 (4.11e-2) −	6.6595e-2 (2.54e-3)
LIRCMOP1	3.7539e-2 (3.83e-2) +	2.5177e-1 (6.86e-2) −	1.3430e-1 (1.62e-2) −	1.8741e-1 (1.59e-2) −	1.7348e-1 (6.28e-2) −	1.2398e-1 (1.80e-2)
LIRCMOP2	2.6799e-2 (2.27e-2) +	1.5853e-1 (5.81e-2) −	7.9418e-2 (1.15e-2) +	1.1319e-1 (1.10e-2) ≈	1.1899e-1 (4.92e-2) ≈	1.1303e-1 (1.87e-2)
LIRCMOP3	5.6497e-2 (4.57e-2) +	2.9593e-1 (7.40e-2) −	1.3750e-1 (2.17e-2) −	1.9568e-1 (2.02e-2) −	1.4753e-1 (6.92e-2) −	9.3223e-2 (1.93e-2)
LIRCMOP4	5.5467e-2 (4.12e-2) ≈	2.0554e-1 (4.31e-2) −	9.4360e-2 (1.39e-2) −	1.3718e-1 (2.03e-2) −	9.5190e-2 (5.70e-2) ≈	6.5689e-2 (1.38e-2)
LIRCMOP5	4.7897e-2 (2.16e-1) +	1.1983e+0 (1.84e-1) −	1.0595e+0 (3.60e-1) −	1.1179e+0 (2.98e-1) −	5.1151e-1 (5.64e-1) +	1.0436e+0 (3.76e-1)
LIRCMOP6	4.5365e-2 (3.77e-2) +	1.3462e+0 (7.05e-4) −	1.2161e+0 (3.35e-1) −	1.1844e+0 (3.67e-1) −	2.5862e-1 (4.50e-1) +	1.0448e+0 (4.35e-1)
LIRCMOP7	1.1699e-1 (1.73e-2) ≈	3.6347e-1 (5.36e-1) −	1.0209e-1 (2.06e-2) ≈	2.1939e-1 (3.98e-1) ≈	2.4447e-2 (3.70e-2) +	1.0733e-1 (3.51e-2)
LIRCMOP8	1.7107e-1 (5.67e-2) −	8.4481e-1 (7.08e-1) −	1.7552e-1 (4.06e-2) −	4.9717e-1 (6.03e-1) −	2.4295e-2 (3.40e-2) +	1.2940e-1 (7.26e-2)
LIRCMOP9	2.5496e-1 (8.97e-2) ≈	4.1887e-1 (9.31e-2) −	8.1103e-1 (1.86e-1) −	4.0247e-1 (1.57e-1) −	2.1350e-1 (4.90e-2) +	2.7667e-1 (1.03e-1)
LIRCMOP10	2.4264e-1 (1.01e-1) +	3.4933e-1 (4.92e-2) −	7.8374e-1 (1.77e-1) −	7.4171e-1 (1.80e-1) −	1.1153e-1 (6.50e-2) +	3.4734e-1 (1.73e-1)
LIRCMOP11	2.3544e-1 (1.03e-1) −	2.0361e-1 (7.76e-2) −	6.4640e-1 (2.18e-1) −	5.9679e-1 (1.76e-1) −	7.5301e-2 (6.32e-2) +	1.6048e-1 (9.43e-2)
LIRCMOP12	1.5325e-1 (7.14e-2) +	2.5299e-1 (1.39e-1) ≈	4.8218e-1 (1.44e-1) −	3.3328e-1 (1.25e-1) −	1.1074e-1 (6.33e-2) +	1.8303e-1 (4.95e-2)
LIRCMOP13	8.7868e-2 (8.67e-3) −	4.7408e-2 (1.28e-3) +	1.3145e+0 (1.70e-3) −	1.3137e+0 (1.20e-3) −	1.5719e-1 (5.40e-2) −	7.3111e-2 (2.48e-3)
LIRCMOP14	6.8648e-2 (1.18e-2) −	4.9044e-2 (9.00e-4) −	1.2713e+0 (1.81e-3) −	1.2295e+0 (2.24e-1) −	1.4967e-1 (1.58e-1) −	4.1517e-2 (1.27e-3)
+/ − / ≈	7/27/4	6/28/4	5/28/5	5/26/7	8/26/4	

TABLE IV
HV STATISTICS FOR FIVE ALGORITHMS IN THREE REAL PROBLEMS

Problem	PPS	CTAEA	MTCMO	TSTI	C3M	BPCMO
Supported I-beam Design	5.4676e-1 (7.66e-3) −	5.5002e-1 (5.83e-3) −	5.5425e-1 (3.85e-3) −	5.5335e-1 (5.25e-3) −	5.5627e-1 (2.03e-3) −	6.1015e-1 (1.20e-1)
Front Rail Design	4.0481e-2 (1.15e-5) −	4.0344e-2 (8.41e-5) −	4.0518e-2 (2.74e-6) −	4.0518e-2 (2.87e-6) −	4.0513e-2 (2.50e-6) −	4.0523e-2 (9.20e-7)
Energy Manage	3.1623e-2 (2.32e-5) −	3.1647e-2 (4.26e-5) −	3.1757e-2 (9.81e-7) −	3.1757e-2 (9.90e-7) −	3.1760e-2 (7.55e-7) −	3.8005e-2 (7.10e-4)
+/ − / ≈	0/3/0	0/3/0	0/3/0	0/3/0	0/3/0	

problems," in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 1179–1186.

[23] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, 2021.

[24] J. Zou, R. Sun, Y. Liu, Y. Hu, S. Yang, J. Zheng, and K. Li, "A multipopulation evolutionary algorithm using new cooperative mechanism for solving multiobjective problems with multiconstraint," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 267–280, 2024.

[25] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 972–986, 2019.

[26] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, C. Yue, H. Lin, and K. C. Tan, "Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 642–656, 2023.

[27] J. Dong, W. Gong, F. Ming, and L. Wang, "A two-stage evolutionary algorithm based on three indicators for constrained multi-objective optimization," *Expert Systems with Applications*, vol. 195, p. 116499, 2022.

[28] R. Sun, J. Zou, Y. Liu, S. Yang, and J. Zheng, "A multistage algorithm for solving multiobjective optimization problems with multi-constraints," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1207–1219, 2023.

[29] Carlos A. Coello Coello and Nareli Cruz Corts, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.

[30] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[31] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, March 29–April 1, 2015. Proceedings, Part II 8*. Springer, 2015, pp. 110–125.

[32] H.-Z. Huang, Y.-K. Gu, and X. Du, "An interactive fuzzy multi-objective optimization method for engineering design," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 5, pp. 451–460, 2006.

[33] Lulu Fan, Tatsuo Yoshino, Tao Xu, Ye Lin, and Huan Liu, "A novel hybrid algorithm for solving multiobjective optimization problems with engineering applications," *Mathematical Problems in Engineering*, vol. 2018, p. 5316379, 2018.