A Constrained Multi-objective Evolutionary Algorithm Based on Early Convergence Followed by Diversity

1st Wenji Li
Shantou University
Guangdong, China
liwj@stu.edu.cn

2nd Yifeng Qiu
Shantou University
Guangdong, China
22yfqiu@stu.edu.cn

3rd Zhaojun Wang
Shantou University
Guangdong, China
17zjwang@stu.edu.cn

4th Qinchang Zhang
Shantou University
Guangdong, China
21qczhang@stu.edu.cn

5th Ruitao Mai
Shantou University
Guangdong, China
21rtmai@stu.edu.cn

6th Biao Xu
Shantou University
Guangdong, China
xubiao@stu.edu.cn

7th Yue Zhang
Shantou University
Guangdong, China
yuezhang@stu.edu.cn

8th Jiafan Zhuang
Shantou University
Guangdong, China
jfzhuang@stu.edu.cn

9th Zhun Fan
Shantou University
Guangdong, China
zfan@stu.edu.cn

2023 5th International Conference on Data-driven Optimization of Complex Systems (DOCS) | 979-8-3503-9352-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/DOCS60977.2023.10294499

***Abstract*—Many real-world optimization problems can be formulated as a kind of constrained multi-objective optimization problems (CMOPs). The main difficulty in solving these problems is to take feasibility, convergence and diversity into account simultaneously. To address this issue, this paper proposes a push and pull search algorithm based on early convergence followed by diversity (PPS-CFD). The proposed algorithm is composed of three different stages, each respectively focusing on convergence, diversity and feasibility. In the first stage, the population rapidly converges to the unconstrained Pareto front (UPF) in M directions, where M is the number of objectives of CMOPs. In the second stage, the population further converges to the UPF and meanwhile its diversity is enhanced. In the last stage, constraints are taken into account to pull the population from the UPF to the constrained Pareto front (CPF). In addition, a search strategy based on objective space division is proposed at the last two stages. Finally, the proposed PPS-CFD is tested on fourteen benchmark problems, compared with other six algorithms, to demonstrate its superiority.**

***Index Terms*—Constrained multi-objective optimization, push and pull search framework, objective space division.**

I. Introduction

Constrained multi-objective optimization problems (CMOPs) widely exist in real-world optimization [1]–[3]. These problems contain several conflicting objectives with a number of constraints, which are hard to solve with traditional ways. Constrained multi-objective evolutionary algorithms (CMOEAs) are effective methods to solve CMOPs. However, the solving procedure might be difficult due to the three challenges [4] listed below:

1) Feasibility challenge. Constraints might create many infeasible regions in the objective space, potentially preventing an algorithm from finding any feasible solutions.
2) Convergence challenge. The infeasible regions might block the population's path to converge towards the CPF.
3) Diversity challenge. The infeasible regions might divide the CPF into some discrete fragments, making an algorithm difficult to find all the fragments simultaneously.

To address these challenges, Fan et al. [5] proposed a push and pull search (PPS) framework, which is able to cross large infeasible regions. Nevertheless, it has no special mechanisms to enhance the performance of diversity and convergence. This paper proposes a push and pull search algorithm based on early convergence followed by diversity (PPS-CFD). The contributions are summarized as follows:

1) A method for quick convergence is proposed. A multi-objective optimization problem (MOP) without constraints is divided into M single-objective optimization problems (SOPs), where M is the number of objectives of the MOP, making the population converge faster toward the UPF along M directions compared to traditional decomposition methods.
2) A method to enhance the diversity is proposed. The objective space of an MOP is divided into multiple regions and some regions without any feasible solutions are removed timely to enhance the search efficiency.
3) By combining the above two mechanisms in the PPS framework, the population can get across infeasible regions and the search performance can be improved

significantly.

The remainder of this work is organized as follows. Section II introduces related work of CMOEAs. Section III describes the proposed PPS-CFD in detail. Section IV gives experimental results and analyzes the advantages of the proposed method. Finally, conclusions are drawn in Section V.

II. Related Work

Among various existing CMOEAs, algorithms based on multi stages or cooperative population have demonstrated their effectiveness in balancing constraints and objectives. As for multi-stage CMOEAs, Fan et al. [5] proposed a push and pull search (PPS) framework, in which the evolutionary procedure is segmented into two stages. In the push stage, the population crosses large infeasible regions and converges toward the UPF by discarding constraints. In the pull stage, a constraint handling technique named improved epsilon method is applied to pull the population from the UPF to the CPF gradually. PPS has provided a novel idea to overcome the search difficulty caused by large infeasible regions, and much work has been done around it. Then, a multi-objective to multi-objective (M2M) decomposition strategy was embedded into PPS [6] to further enhance the diversity. Sun et al. [7] proposed a three-stage CMOEA to solve multi-constraint CMOPs. Initially, the algorithm searches for the UPF without constraints. Then, constraints are added one by one to gradually increase the complexity. Zou et al. [8] proposed a two-stage CMOEA with adaptive adjustment, where a fast global search is performed at the first stage and a mechanism with dynamic resource allocation between exploration and convergence is employed at the last stage.

Regarding multi-population cooperation CMOEAs, Tian et al. [9] designed a weak coevolutionary framework, where an assistant population is evolved with partial objectives or constraints to assist the main population. Liu et al. [10] constructed two other assistant populations to generate escape and expansion forces, which can avoid main population trapped into local feasible regions. Liang et al. [11] use information from cooperative populations to predict the type of a problem, which determines the subsequent evolutionary strategy accordingly.

The above-mentioned CMOEAs have similar ideas, i.e. using the information with simplified constraints or no constraints to help solve the original CMOPs. Information from high quality infeasible solutions is employed to enhance the convergence toward CPFs.

III. Proposed Method

The proposed PPS-CFD includes three stages as illustrated in Algorithm 1 and Fig. 1. The first stage disregards constraints and makes the population converge rapidly toward the UPFs in M directions by decomposition technique. The second stage continues to dismiss constraints and the original MOP is optimized to further improve the convergence and diversity. In the last stage, constraints are taken into account to guide the population toward the CPFs.

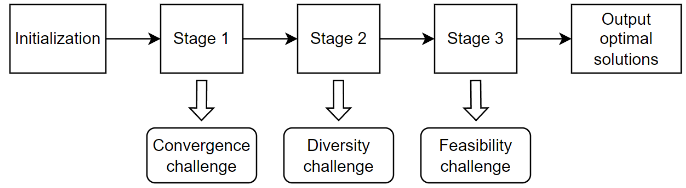

Fig. 1. Procedure of PPS-CFD. There are three different search stages which focus on convergence, diversity and feasibility, respectively.

Algorithm 1: Procedure of PPS-CFD

Input: population size N, subpopulation size $subN$, objective dimension M

Output: population P

$P \leftarrow$ randomly generate N individuals;
// stage 1
$P \leftarrow SolveDecomposedProblem(P, subN)$;
// stage 2
$flag \leftarrow 0$; // ignoring constraints
$P \leftarrow RegionBasedEvolution(P, N, flag)$;
// stage 3
$flag \leftarrow 1$; // considering constraints
$P \leftarrow RegionBasedEvolution(P, N, flag)$;

A. Stage One

This stage is dedicated to bolstering the convergence of the proposed PPS-CFD, with its pseudocode outlined in Algorithm 2. We assume that the early stages of evolution necessitate a rapid convergence of the population towards the UPFs, disregarding the distribution and feasibility of the population. Optimizing multiple objectives simultaneously might pose a challenge for achieving convergence. To mitigate this, we decompose the original M-objective problem into M single-objective subproblems. For each objective, a one-hot weight vector is constructed. For instance, when $M = 2$, weight vectors are assigned as (1,0) and (0,1). Each subproblem employs the penalty-based boundary intersection (PBI) function [12] as its objective function, defined as follows:

$$\begin{cases} min\ g^{pbi}(\mathbf{x}, \mathbf{W}, \mathbf{z}) = d_1 + \theta d_2 \\ d_1 = \frac{\|(\mathbf{z} - \mathbf{F}(\mathbf{x}))^T \mathbf{W}\|}{\|\mathbf{W}\|} \\ d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z} - d_1 \mathbf{W})\| \end{cases} \quad (1)$$

where $\mathbf{x}$ is the decision vector, $\mathbf{W}$ is the weight vector, $\mathbf{z}$ is the reference point and θ is the penalty factor.

For each subproblem, we select $subN$ solutions from population P that have the minimum PBI values, which then form the initial population $subP$. Following this, a single-objective evolutionary algorithm (SOEA) optimizes $subP$ until stability is achieved, meaning no further individual replacements occur. We have chosen NL-SHADE-LBC [13] as the optimizer for this process due to its ability to adaptively adjust the parameters of the differential evolution (DE) operator. Importantly, depending on the specific characteristics of the subproblem,

Algorithm 2: SolveDecomposedProblem

Input: population P, subpopulation size $subN$
Output: population P
$offspring \leftarrow \emptyset$;
for $i \leftarrow 1$ *to* M **do**
  $W \leftarrow$ construct weight vector for the ith objective;
  $pbi \leftarrow$ calculate PBI value of P by equation (1);
  $subP \leftarrow$ select $subN$ solution from P with minimum pbi;
  repeat
    $subP \leftarrow$ optimize $subP$ with SOEA for 1 generation;
  until $subP$ *is stable*;
  $best \leftarrow$ select the best solution from $subP$;
  $offspring \leftarrow offspring \cup best$
end
$offspring \leftarrow$ add normally distributed disturbance to $offspring$;
$P \leftarrow$ select N best solution from $P \cup offspring$ by non-dominated sorting;

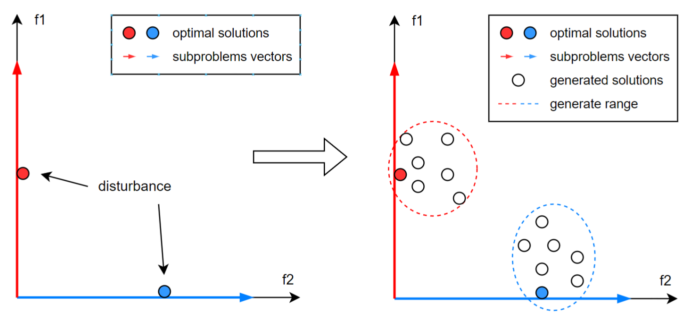

Fig. 2. The effect of disturbance where $M = 2$. The red dot and the blue dot stand for single-objective optimal solutions. With disturbance added, new solutions is generated in nearby area shown as red circle and blue circle.

a different SOEA could be used to potentially yield superior performance.

While the above-mentioned decomposition approach can hasten convergence, it may inadvertently lead to local convergence. To counteract this, we construct a new population derived from these optimal $subP$s to amplify diversity. Initially, we select an optimal solution from each $subP$ respectively. Subsequently, we introduce a normally distributed disturbance to these selected solutions, thereby generating additional solutions in close proximity. The effect of this disturbance, where $M = 2$, is illustrated in Fig. 2. Ultimately, from both the initial population and the generated solutions, we select N solutions to constitute the new population for the second stage.

B. Stage Two

Following the first stage, the population has already converged as close as possible to the UPFs. Stage 2 then furthers this convergence and enhances diversity through multi-objective optimization, with constraints discarded. The pseudocode is exhibited in Algorithm 3, where the input parameter $flag$ is set to 0, signifying the discarding of constraints.

Algorithm 3: RegionBasedEvolution

Input: population P, population size N, constraint status flag $flag$
Output: population P
$W \leftarrow$ generate N weight vector uniformly;
$Region \leftarrow$ divided objective space according to W;
$gen \leftarrow 0$;
$DR \leftarrow floor(N/10)$; `// detection range for` $Algorithm5$
while *termination criterion is not fulfilled* **do**
  $Region \leftarrow$ associate P with region;
  for $i \leftarrow 1$ *to* $|Region|$ **do**
    $parent \leftarrow$ select parent from $Region(i)$ randomly;
    $offspring \leftarrow$ generate an offspring with $parent$ by DE and PM;
    $P \leftarrow P \cup offspring$;
  end
  $P \leftarrow EnvironmentalSelection(P, N, Region, flag)$;
  $gen \leftarrow gen + 1$;
  if *HV of P is almost unchanged* **then**
    if $flag == 0$ **then**
      break;
    else
      $Region \leftarrow UpdateRegion(P, N, Region, DR)$;
      $DR \leftarrow floor(DR/2)$;
      $DR \leftarrow max(DR, M + 1)$;
    end
  end
end

Initially, the objective space is divided into N regions, each associated with one of the N uniformly generated weight vectors. And all solutions are associated with regions based on their distance to respective weight vectors. The parent solutions are randomly selected from each region. If a region does not have enough solutions, neighboring regions and then the entire regions are considered. Following this, a combination of the DE operator and polynomial mutation (PM) is utilized for reproduction. Subsequently, an environmental selection strategy is suggested to select N solutions for the next generation. An adaptive stage-switch mechanism is developed. Specifically, when the HV [14] value of the population experiences negligible changes, the evolution of stage 2 is deemed complete.

Algorithm 4 presents the environmental selection strategy referenced in Algorithm 3. We design a novel environmental selection approach using the Constrained Dominance Principle (CDP) [15] and region division. The primary aim of this approach is to maintain the minimum number of solutions

Algorithm 4: EnvironmentalSelection

```
Input: population P, population size N, objective
       space region Region, constraint status flag
       flag
Output: population P
1  avg ← floor(N/|Region|);
2  worstSet ← ∅;
3  for i ← 1 to |Region| do
4      amount ← get the amount of solutions in
         Region(i);
5      if amount > avg then
6          if flag == 0 then
7              worst ← select amount − avg worst
                 solutions from Region(i) by
                 non-dominated sorting;
8          else
9              worst ← select amount − avg worst
                 solutions from Region(i) by constrained
                 non-dominated sorting;
10         end
11         worstSet ← worstSet ∪ worst;
12     end
13 end
14 if flag == 0 then
15     worstSet ← select |P| − N worst solutions from
         worstSet by non-dominated sorting;
16 else
17     worstSet ← select |P| − N worst solutions from
         worstSet by constrained non-dominated sorting;
18 end
19 P ← P − worstSet;
```

in each region. The minimum quantity, denoted as avg, is defined as the smallest number of solutions in a region when the population is uniformly distributed across all regions. If a region contains more than avg solutions, the excess worst solution, as determined by the CDP method, is added to $worstSet$. Once all regions have been assessed, the worst solution in $worstSet$ is eliminated until only N solutions remain in the population.

C. Stage Three

Upon completion of the second stage, the population nearly converges to the UPFs with a robust diversity. Stage 3 aims to draw the population from the UPFs to the CPFs, taking constraints into account. The pseudocode is exhibited in Algorithm 3 in a similar manner, where the input parameter $flag$ is set to 1 to denote the consideration of constraints. The key differences between stages 2 and 3 can be itemized as follows:

1) Regarding environmental selection, non-dominated sorting is applied in stage 2 (refer to Line 7 and 15 in Algorithm 4), whereas constrained non-dominated sorting is applied in stage 3 (refer to Line 9 and 17 in Algorithm 4).
2) Pertaining to operations when the population is relatively stable, the current stage is considered complete in stage 2 (refer to Line 16 in Algorithm 3), while regions are updated in stage 3 (refer to Line 18 in Algorithm 3).

Algorithm 5: UpdateRegion

```
Input: population P, population size N, objective
       space region Region, detection range DR
Output: objective space region Region
best ← get feasible and non-dominated solution;
if |best| < N/2 then
    best ← get floor(N/2) best solution by
      constrained non-dominated sorting;
end
del ← ∅;
for i ← 1 to |Region| do
    B ← get the nearest DR region to Region(i);
    if B ∩ best == ∅ then
        del ← del ∪ Region(i);
    end
end
Region ← Region − del;
Region ← associate P with Region;
```

The region update mentioned earlier is intended to eliminate regions that are unlikely to contain any solutions on the CPFs, thus concentrating computing resources on promising regions. The pseudocode for this process is illustrated in Algorithm 5. Initially, if more than half solutions in the population are feasible and non-dominated, these solutions are stored in a set denoted as $best$. If not, constrained non-dominated sorting is applied to select half of the population to form the set $best$. Subsequently, all regions that do not contain any solutions from the $best$ set within their closest DR regions (including the regions themselves) is identified. These regions are likely devoid of any CPFs and are thus eliminated by deleting their respective weight vectors.

It is necessary to note that this update procedure only occur in the case of the population is nearly stable. Moreover, if a region contains solutions on the CPFs, it may suggest that its neighboring region also houses non-dominated solutions on the CPFs. To bolster the search capability at the CPF's edge, we establish a detection range, DR, to preserve these neighboring regions. Initially, DR is set to $N/10$ and is halved with each region update. As DR diminishes, regions lacking any solutions on the CPFs will be progressively eliminated, thereby concentrating computing resources in the most promising regions.

IV. Experimental Study

A. Experimental Setup

The proposed PPS-CFD is compared with six other CMOEAs, namely NSGA-II [15], C-MOEA/D [16], MOEA-DD [17], PPS [5], MTCMO [18], and CCMO [9]. These algorithms are tested on the LIR-CMOP [19] benchmark suite,

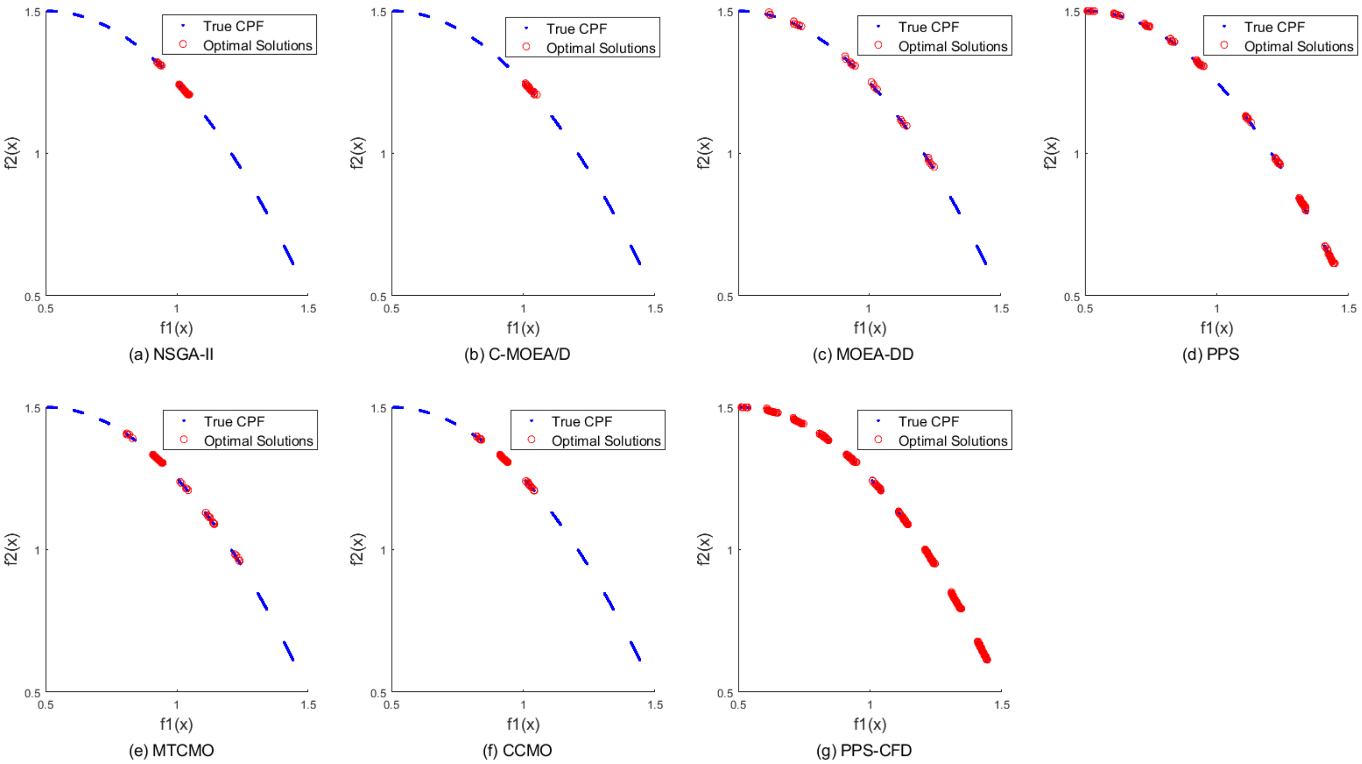

Fig. 3. Results achieved by each algorithms on LIRCMOP3. Blue solid lines represent CPF, red dots represent feasible and non-dominated solutions, and white areas represent infeasible regions.

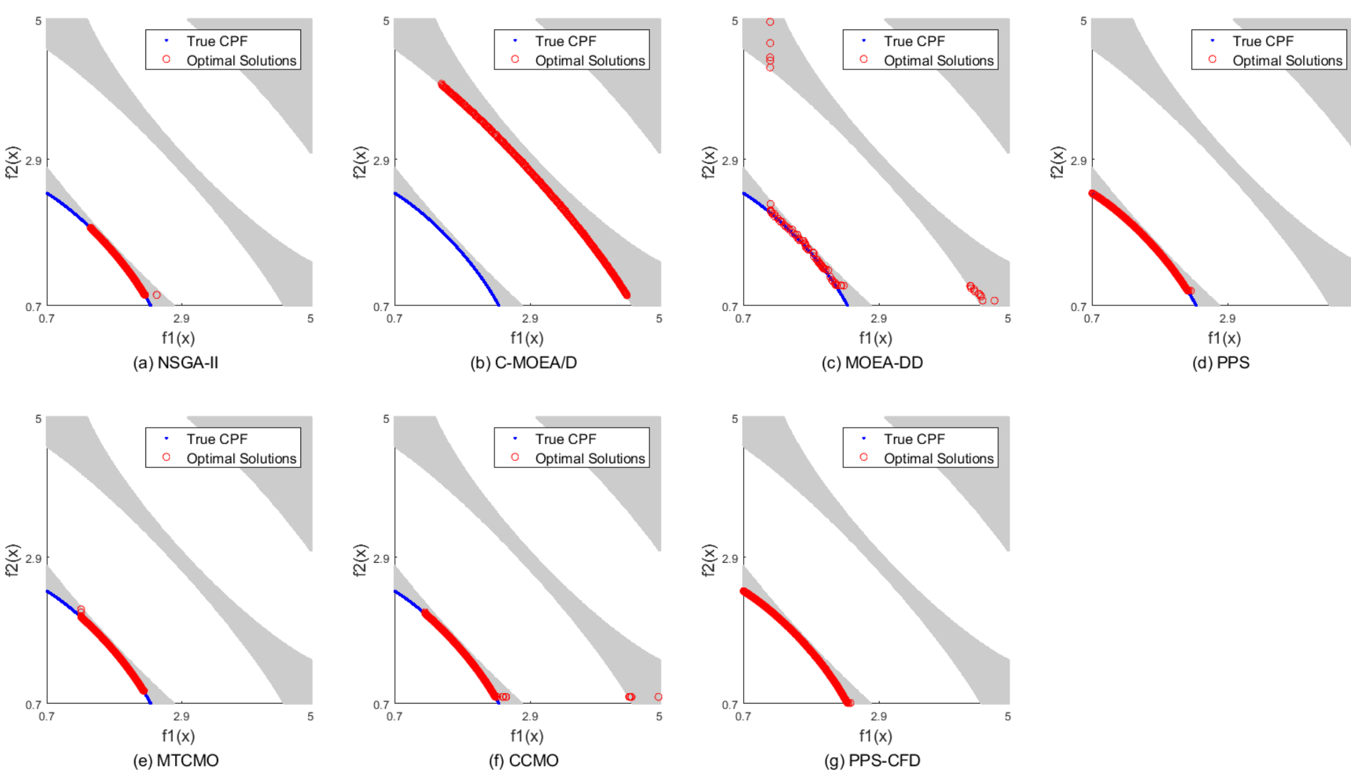

Fig. 4. Results achieved by each algorithms on LIRCMOP7. Blue solid lines represent CPF, red dots represent feasible non-dominated solutions, white areas represent infeasible regions, and gray areas represent feasible regions.

TABLE I

IGD RESULTS OF PPS-CFD AND THE OTHER SIX CMOEAS ON LIR-CMOP1-LIR-CMOP14. THE TERMS '+', '-' AND '=' INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE OR STATISTICALLY SIMILAR TO THE RESULTS OBTAINED BY PPS-CFD, RESPECTIVELY. THE BEST RESULT ON EACH PROBLEM IS HIGHLIGHTED IN GRAY.

Problem	M	D		NSGA-II	C-MOEA/D	MOEA-DD	PPS	MTCMO	CCMO	PPS-CFD
LIRCMOP1	2	30	mean	2.5984E-1	2.7302E-1	1.5160E-1	7.7559E-3	9.1523E-2	1.6552E-1	**3.22E-03**
			(std)	(2.24E-02) -	(3.19E-02) -	(8.20E-02) -	(4.13E-03) -	(1.77E-02) -	(4.08E-02) -	**(9.75E-04)**
LIRCMOP2	2	30	mean	2.2314E-1	2.3326E-1	1.1438E-1	5.1129E-3	7.8331E-2	1.8079E-1	**2.40E-03**
			(std)	(2.01E-02) -	(2.25E-02) -	(3.22E-02) -	(7.16E-04) -	(1.15E-02) -	(3.45E-02) -	**(5.09E-04)**
LIRCMOP3	2	30	mean	2.7717E-1	2.9934E-1	1.2348E-1	4.3940E-3	9.7608E-2	1.9989E-1	**2.65E-03**
			(std)	(4.07E-02) -	(4.02E-02) -	(3.79E-02) -	(5.97E-04) -	(2.10E-02) -	(4.71E-02) -	**(1.81E-03)**
LIRCMOP4	2	30	mean	2.4605E-1	2.7176E-1	1.3256E-1	3.7818E-3	1.1648E-1	2.1699E-1	**2.06E-03**
			(std)	(2.91E-02) -	(3.16E-02) -	(2.40E-02) -	(4.61E-04) -	(1.67E-02) -	(3.67E-02) -	**(4.06E-04)**
LIRCMOP5	2	30	mean	1.2149E+0	1.2190E+0	1.2060E+0	2.3271E-3	7.3889E-1	2.7224E-1	**2.08E-03**
			(std)	(6.45E-03) -	(5.76E-03) -	(1.14E-02) -	(1.31E-04) -	(4.53E-01) -	(4.94E-02) -	**(5.49E-05)**
LIRCMOP6	2	30	mean	1.3447E+0	1.3449E+0	1.3531E+0	2.6975E-3	8.2768E-1	2.6607E-1	**2.21E-03**
			(std)	(7.16E-05) -	(1.32E-04) -	(2.37E-03) -	(1.61E-04) -	(4.66E-01) -	(6.50E-02) -	**(5.96E-05)**
LIRCMOP7	2	30	mean	3.3287E-1	8.1164E-1	7.1079E-1	1.4360E-2	1.1114E-1	1.0154E-1	**2.97E-03**
			(std)	(5.38E-01) -	(7.73E-01) -	(7.62E-01) -	(2.82E-02) -	(2.84E-02) -	(2.72E-02) -	**(8.44E-05)**
LIRCMOP8	2	30	mean	8.5152E-1	1.4588E+0	9.0607E-1	6.3636E-2	1.5714E-1	1.5304E-1	**2.93E-03**
			(std)	(7.38E-01) -	(5.10E-01) -	(7.08E-01) -	(1.30E-01) -	(3.71E-02) -	(4.30E-02) -	**(1.47E-04)**
LIRCMOP9	2	30	mean	9.4682E-1	7.7600E-1	6.6036E-1	3.5062E-1	6.4571E-1	4.2881E-1	**6.79E-02**
			(std)	(8.42E-02) -	(1.61E-01) -	(1.61E-01) -	(8.78E-02) -	(1.39E-01) -	(1.39E-01) -	**(1.06E-01)**
LIRCMOP10	2	30	mean	8.6221E-1	4.8466E-1	4.0393E-1	5.8543E-2	5.3437E-1	1.3216E-1	**2.48E-03**
			(std)	(5.37E-02) -	(2.33E-01) -	(5.00E-02) -	(9.72E-02) -	(2.82E-01) -	(4.85E-02) -	**(6.75E-05)**
LIRCMOP11	2	30	mean	7.3212E-1	7.7799E-1	6.1257E-1	1.9927E-1	3.6320E-1	5.4289E-2	**2.41E-03**
			(std)	(8.20E-02) -	(1.16E-01) -	(1.44E-01) -	(1.21E-01) -	(1.39E-01) -	(2.80E-02) -	**(6.99E-05)**
LIRCMOP12	2	30	mean	6.8136E-1	5.0015E-1	2.6502E-1	1.2325E-1	3.0299E-1	2.0574E-1	**3.08E-03**
			(std)	(1.49E-01) -	(1.38E-01) -	(3.65E-02) -	(1.69E-02) -	(8.62E-02) -	(7.24E-02) -	**(1.59E-04)**
LIRCMOP13	3	30	mean	1.3084E+0	1.2983E+0	1.3250E+0	7.1860E-2	1.3039E+0	**5.2736E-2**	9.59E-02
			(std)	(8.08E-04) -	(1.44E-04) -	(3.00E-03) -	(1.61E-03) +	(5.24E-04) -	**(3.54E-04)** +	(2.63E-03)
LIRCMOP14	3	30	mean	1.2647E+0	1.2542E+0	1.2813E+0	6.7209E-2	1.2598E+0	**5.4786E-2**	7.49E-02
			(std)	(9.34E-04) -	(1.67E-04) -	(2.70E-03) -	(1.26E-03) +	(4.81E-04) -	**(4.39E-04)** +	(1.60E-03)
Wilcoxon's rank sum test(+/-/=)				0/14/0	0/14/0	0/14/0	2/12/0	0/14/0	2/12/0	

known for its large infeasible regions that present considerable challenges to CMOEAs. For comparative purposes, we select IGD [20] and HV [14] as performance indicators. All experiments are conducted using PlatEMO [21].

The experimental parameter settings are as follows:

1) The population size N is set to 300, and the maximum evaluation is set to 300,000. Each algorithm is run 30 times independently.
2) For the proposed PPS-CFD, the subpopulation size is set to 100. The parameters for the embedded NL-SHADE-LBC [13] are set refer to the recommendations in the original paper.
3) For the test problems and the other algorithms compared, the parameters are also set according to their original papers.

B. *Experimental Results*

Table I presents the IGD results of PPS-CFD and the other six CMOEAs on LIR-CMOP1-14, while Table II depicts the corresponding HV results. Each metric is represented by both its mean value and standard deviation, with the best mean values highlighted in gray with a bold font. Moreover, we employ Wilcoxon's rank-sum test to identify significant differences.

The experimental results indicate that PPS-CFD significantly outperforms the other six comparison CMOEAs, achieving the optimal outcomes on most of the test problems. Specifically, PPS-CFD demonstrates substantial advantages on test problems with extremely narrow feasible regions (LIR-CMOP1-4), achieving the best results across all metrics. For test problems in which the CPFs are obstructed by numerous infeasible regions (LIRCMOP5-12), PPS-CFD also secures the majority of the top results, falling short only once when compared with PPS.

The superiority of PPS-CFD can be ascribed to the following factors:

1) The early stages of evolution disregard constraints, allowing the population to traverse substantial infeasible regions and converge towards the UPFs.
2) Single-objective optimization for decomposed subproblems in stage 1 effectively accelerates the early convergence.
3) The proposed search strategy, based on region division, ensures good diversity. Coupled with the proposed region update mechanism, computing resources are focused on potential regions containing Pareto optimal solutions, significantly enhancing search efficiency.

To further demonstrate the superiority of PPS-CFD, we showcase the optimal solutions obtained by each algorithm on LIR-CMOP3 and LIR-CMOP7, respectively shown in Fig. 3 and Fig. 4. LIR-CMOP3 is a problem characterized

TABLE II
HV RESULTS OF PPS-CFD AND THE OTHER SIX CMOEAS ON LIR-CMOP1-LIR-CMOP14. THE TERMS '+', '-' AND '=' INDICATE THAT THE RESULT IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE OR STATISTICALLY SIMILAR TO THE RESULTS OBTAINED BY PPS-CFD, RESPECTIVELY. THE BEST RESULT ON EACH PROBLEM IS HIGHLIGHTED IN GRAY.

Problem	M	D		NSGA-II	C-MOEA/D	MOEA-DD	PPS	MTCMO	CCMO	PPS-CFD
LIRCMOP1	2	30	mean	1.2449E-1	1.2055E-1	1.6238E-1	2.3685E-1	1.8326E-1	1.5538E-1	**2.39E-01**
			(std)	(7.02E-03) -	(1.08E-02) -	(2.74E-02) -	(1.34E-03) -	(8.69E-03) -	(1.62E-02) -	**(1.79E-04)**
LIRCMOP2	2	30	mean	2.4317E-1	2.3897E-1	2.9733E-1	3.6024E-1	3.1486E-1	2.6399E-1	**3.62E-01**
			(std)	(8.74E-03) -	(1.04E-02) -	(1.55E-02) -	(3.94E-04) -	(6.70E-03) -	(1.81E-02) -	**(2.01E-04)**
LIRCMOP3	2	30	mean	1.0926E-1	1.0346E-1	1.5599E-1	2.0592E-1	1.6637E-1	1.3156E-1	**2.09E-01**
			(std)	(1.05E-02) -	(1.23E-02) -	(1.27E-02) -	(4.78E-04) -	(7.62E-03) -	(1.55E-02) -	**(3.24E-04)**
LIRCMOP4	2	30	mean	2.1099E-1	2.0032E-1	2.5903E-1	3.1512E-1	2.6551E-1	2.2244E-1	**3.17E-01**
			(std)	(1.31E-02) -	(1.38E-02) -	(1.08E-02) -	(3.95E-04) -	(8.60E-03) -	(1.61E-02) -	**(3.22E-04)**
LIRCMOP5	2	30	mean	0.0000E+0	0.0000E+0	0.0000E+0	2.9350E-1	7.7917E-2	1.6403E-1	**2.94E-01**
			(std) -	(0.00E+00) -	(0.00E+00) -	(0.00E+00) -	(5.65E-05) =	(7.56E-02) -	(1.89E-02) -	**(2.73E-05)**
LIRCMOP6	2	30	mean	0.0000E+0	0.0000E+0	0.0000E+0	**1.9883E-1**	5.4545E-2	1.2387E-1	1.99E-01
			(std)	(0.00E+00) -	(0.00E+00) -	(0.00E+00) -	**(4.52E-05)** +	(4.98E-02) -	(1.26E-02) -	(3.24E-05)
LIRCMOP7	2	30	mean	2.1381E-1	1.3759E-1	1.5461E-1	2.9104E-1	2.5087E-1	2.5363E-1	**2.96E-01**
			(std)	(8.57E-02) -	(1.23E-01) -	(1.21E-01) -	(1.17E-02) -	(8.41E-03) -	(8.58E-03) -	**(9.82E-05)**
LIRCMOP8	2	30	mean	1.2864E-1	3.1766E-2	1.1921E-1	2.8068E-1	2.4125E-1	2.4125E-1	**2.96E-01**
			(std)	(1.15E-01) -	(7.56E-02) -	(1.10E-01) -	(2.64E-02) =	(9.12E-03) -	(1.01E-02) -	**(1.18E-04)**
LIRCMOP9	2	30	mean	1.2776E-1	2.1115E-1	2.5219E-1	4.6107E-1	2.7916E-1	3.9664E-1	**5.47E-01**
			(std)	(3.19E-02) -	(8.36E-02) -	(1.05E-01) -	(3.14E-02) -	(8.21E-02) -	(6.32E-02) -	**(3.15E-02)**
LIRCMOP10	2	30	mean	9.8546E-2	3.8833E-1	4.8709E-1	6.8173E-1	3.4458E-1	6.3452E-1	**7.08E-01**
			(std)	(3.03E-02) -	(1.69E-01) -	(2.43E-02) -	(4.77E-02) =	(2.16E-01) -	(2.49E-02) -	**(7.94E-05)**
LIRCMOP11	2	30	mean	2.3351E-1	2.4701E-1	3.2214E-1	5.7022E-1	4.7464E-1	6.7094E-1	**6.94E-01**
			(std)	(4.35E-02) -	(7.79E-02) -	(1.17E-01) -	(7.84E-02) -	(1.01E-01) -	(1.04E-02) -	**(5.31E-05)**
LIRCMOP12	2	30	mean	3.0449E-1	4.2912E-1	5.0194E-1	5.6749E-1	4.8191E-1	5.2030E-1	**6.20E-01**
			(std)	(8.86E-02) -	(5.85E-02) -	(1.86E-02) -	(7.51E-03) -	(3.92E-02) -	(3.88E-02) -	**(3.25E-05)**
LIRCMOP13	3	30	mean	2.6536E-4	5.5845E-4	1.1586E-4	5.5527E-1	3.9237E-4	**5.7731E-1**	5.34E-01
			(std)	(1.85E-04) -	(2.21E-05) -	(1.27E-04) -	(2.64E-03) +	(1.38E-04) -	**(4.55E-04)** +	(2.02E-03)
LIRCMOP14	3	30	mean	1.0623E-3	1.5981E-3	3.7219E-4	5.6485E-1	1.4204E-3	**5.7569E-1**	5.54E-01
			(std)	(3.82E-04) -	(2.56E-05) -	(3.59E-04) -	(2.14E-03) +	(2.09E-04) -	**(4.67E-04)** +	(1.61E-03)
Wilcoxon's rank sum test(+/-/=)				0/14/0	0/14/0	0/14/0	3/8/3	0/14/0	2/12/0	

by extremely narrow feasible regions and a discrete CPF, presenting a considerable challenge for convergence, diversity, and feasibility. As shown in Fig. 3, only PPS-CFD manages to converge to all segments of the CPF and covers all fragments uniformly. LIR-CMOP7, on the other hand, is a problem in which the CPF is obstructed by a large number of infeasible regions, demanding a strong ability from CMOEAs to traverse infeasible areas. As illustrated in Fig. 4, all solutions from C-MOEA/D and some solutions from MOEA-DD and CCMO are stuck far away from the CPF. Among the other four CMOEAs, PPS-CFD exhibits the best diversity, covering the entire CPF uniformly.

V. CONCLUSION

This paper introduces a three-stage CMOEA, dubbed PPS-CFD. In the first stage, constraints are disregarded and the original MOP is decomposed into several SOPs. This accelerates the convergence towards the UPFs in specific directions. In the second stage, constraints continue to be disregarded as the original MOP is processed for further convergence and diversity enhancement. An evolutionary strategy centered around region division is implemented to ensure robust diversity. During the third stage, constraints are considered to transition the population from the UPFs toward the CPFs. Additionally, a region update mechanism is proposed to eliminate regions devoid of the CPFs and to concentrate computational resources on promising regions. Compared to the original PPS, our proposed PPS-CFD demonstrates superior performance in both convergence and diversity. Experimental comparisons with six other CMOEAs on a benchmark test suite further solidify the superiority of PPS-CFD.

ACKNOWLEDGMENT

This research was supported in part by the National Key R&D Program of China (Nos. 2021ZD0111501, 2021ZD0111502), the Key Laboratory of Intelligent Manufacturing Technology (Shantou University), the Ministry of Education, the Science and Technology Planning Project of Guangdong Province of China (grant number 180917144960530), the Project of Educational Commission of Guangdong Province of China (grant number 2017KZDXM032), the State Key Lab of Digital Manufacturing Equipment & Technology (grant number DMETKF2019020), the National Natural Science Foundation of China (Nos. 62176147, 62201336, 51907112 and U2066212), the STU Scientific Research Foundation for Talents (Nos. NTF21001, NTF21039, NTF22030 and NTF20009), the Science and Technology Planning Project of Guangdong Province of China (Nos. 2019A050520001, 2021A0505030072, 2022A1515110660), the Science and Technology Special Funds Project of Guangdong Province of China (Nos. STKJ2021176, STKJ2021019), the Guangdong Special Support Program for Outstanding Talents (grant number 2021JC06X549), the Jiangxi "Double Thousand Plan" Project (grant number JXSQ20210019), the Natural Science Foundation of Guangdong Province of China (Nos. 2019A1515011671, 2021A1515011709), and the Li Ka Shing Foundation Cross Research Project (grant number 2020LKSFG02D).

REFERENCES

[1] W. Li, Z. Wang, R. Mai, P. Ren, Q. Zhang, Y. Zhou, N. Xu, J. Zhuang, B. Xin, L. Gao *et al.*, "Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey," *Visual Intelligence*, vol. 1, no. 1, p. 2, 2023.

[2] Z. Fan, Z. Zheng, B. Xu, W. Li, Y. Zhang, and Z. Hao, "Performance optimization of hard rock tunnel boring machine using multi-objective evolutionary algorithm," *Computers & Industrial Engineering*, vol. 169, p. 108251, 2022.

[3] Z. Fan, Y. You, X. Cai, H. Zheng, G. Zhu, W. Li, A. Garg, K. Deb, and E. Goodman, "Analysis and multi-objective optimization of a kind of teaching manipulator," *Swarm and Evolutionary Computation*, vol. 50, p. 100554, 2019.

[4] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit," *Evolutionary Computation*, vol. 28, no. 3, pp. 339–378, 09 2020.

[5] 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.

[6] Z. Fan, Z. Wang, W. Li, Y. Yuan, Y. You, Z. Yang, F. Sun, and J. Ruan, "Push and pull search embedded in an m2m framework for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 54, p. 100651, 2020.

[7] R. Sun, J. Zou, Y. Liu, S. Yang, and J. Zheng, "A multi-stage algorithm for solving multi-objective optimization problems with multi-constraints," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022.

[8] J. Zou, J. Luo, Y. Liu, S. Yang, and J. Zheng, "A flexible two-stage constrained multi-objective evolutionary algorithm based on automatic regulation," *Information Sciences*, vol. 634, pp. 227–243, 2023.

[9] 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.

[10] Z.-Z. Liu, F. Wu, J. Liu, Y. Qin, and K. Li, "Constrained multiobjective optimization with escape and expansion forces," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2023.

[11] 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.

[12] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[13] V. Stanovov, S. Akhmedova, and E. Semenkin, "Nl-shade-lbc algorithm with linear parameter adaptation bias change for cec 2022 numerical optimization," in *2022 IEEE Congress on Evolutionary Computation (CEC)*, 2022, pp. 01–08.

[14] 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.

[15] 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.

[16] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[17] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[18] 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.

[19] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions," *Soft Computing*, vol. 23, pp. 12 491–12 510, 2019.

[20] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[21] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.