2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Parameter-Adaptive Paired Offspring Generation for Constrained Large-Scale Multiobjective Optimization Algorithm

Haiyue Zhu
the State Key Laboratory of Synthetical Automation for Process Industries
Northeastern University
Shenyang, China
2829677592@qq.com

Qingda Chen
the State Key Laboratory of Synthetical Automation for Process Industries
Northeastern University
Shenyang, China
cqd0309@126.com

Jinliang Ding
the State Key Laboratory of Synthetical Automation for Process Industries
Northeastern University
Shenyang, China
jlding@mail.neu.edu.cn

Xingyi Zhang
School of Artificial Intelligence
Anhui University
Hefei, China
xyzhanghust@gmail.com

Hongfeng Wang
the State Key Laboratory of Automation for Process Industries
Northeastern University
Shenyang, China
wanghongfeng@ise.neu.edu.cn

***Abstract*—There are significant challenges in designing optimization algorithms for constrained large-scale multiobjective optimization problems due to numerous decision variables and constraints. For example, the decision space size exponentially grows with the number of decision variables, and constraints restrict the feasible range, increasing the complexity of the search space. To solve these problems, this paper presents a constrained large-scale multiobjective optimization algorithm based on adaptive paired offspring generation (aPOCEA). Specifically, an adaptive parameter adjustment strategy is proposed to determine the number of solutions in each subpopulation and balance the exploration and exploitation ability of the algorithm, enhancing the convergence speed of aPOCEA. Meanwhile, we propose a parent selection strategy to select high-quality parent solutions, increasing the probability of generating high-quality offspring solutions. Experimental results on ten benchmarks, each with two to three objectives, multiple constraints, and hundreds of decision variables, demonstrate that aPOCEA outperforms other representative optimization algorithms.**

Keywords—Constrained large-scale multiobjective optimization, adaptive parameter adjustment strategy, parent selection strategy.

I. INTRODUCTION

Many decision variables need to be simultaneously optimized in many real-world optimization problems (e.g., automotive design problems [1], supply chain network problems [2]) to achieve multiple conflicting objectives subject to constraints. The above problems are known as constrained large-scale multiobjective optimization problems (CLMOPs), and their mathematical formulation is given as follows:

$$\begin{aligned} &\underset{x\in\Omega}{Min}\, F(x) = (f_1(x), f_2(x), ..., f_M(x))^T \\ &s.t. g_j(x) \le 0, j = 1, ..., J \end{aligned} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, ..., x_D)$ is a decision vector consisting of D (usually $D \geq 100$) decision variables, Ω represents the decision space, M is the number of objectives, and J is the total number of inequality and equality constraints. The final goal of solving CLMOPs is to obtain a set of Pareto-optimal solutions that satisfy the constraints and have good convergence and diversity, usually called the Pareto optimal set (POS). The image of the POS in the objective space is called the Pareto optimal front (POF).

Scholars have proposed large-scale multiobjective evolutionary algorithms (LMOEAs) in the past decade, which can be roughly categorized into four types. The first type is LMOEAs based on cooperative co-evolution, which optimizes decision variables by dividing them into multiple groups and optimizing them separately. A representative algorithm in this category is the third-generation collaborative coevolutionary differential evolution algorithm designed in [3]. The second type is LMOEAs based on decision variable analysis, which uses a mechanism for analyzing decision variables and obtaining the optimal grouping of decision variables. A representative algorithm of this type has been proposed by Ma *et al.* [4]. The third type is LMOEAs based on problem transformation, which transforms the original large-scale problem into a small-scale problem through a problem transformation function. For example, the weight optimization framework proposed by Zille *et al.* [5] optimizes the weight vector instead of optimizing the decision variables, transforming the original large-scale problem into a small-scale problem. The last type is a search method based on a learning strategy, which uses the learning mechanism among particles in the original decision space to improve the optimization ability of the algorithm. The most representative algorithm in this type is large-scale multiobjective optimization based on a competitive swarm optimizer (LMOCSO), proposed by Tian *et al.* [6].

Indeed, a few scholars have proposed constrained large-scale multiobjective evolutionary algorithms (CLMOEAs). Regis *et al.* [1] propose a non-population-based algorithm to handle large-scale problems with ordinal discrete variables and multiple black-box constraints. Zhang *et al.* [2] develop an auxiliary population-based co-evolutionary algorithm to solve the constrained large-scale multiobjective supply chain network problem. He *et al.* [7] introduce a paired offspring generation-based EA (POCEA) to highlight the importance of generating valuable feasible or functional infeasible solutions through offspring generation.

To improve the search efficiency of the algorithm and handle infeasible solutions effectively, this paper presents a CLMOEA based on adaptive paired offspring generation (aPOCEA). The specific contributions of this paper are summarized as follows:

1) This paper presents an adaptive parameter adjustment strategy to balance the exploration and exploitation of the algorithm and ensure the stability and flexibility of the aPOCEA, increasing the convergence speed of the aPOCEA.

2) We propose a parent selection strategy that uses the reference vector-guided EA (RVEA) selection strategy [8] and the tournament selection strategy to improve the quality of the parent solutions, significantly increasing the probability of generating high-quality offspring solutions.

3) We conduct a comprehensive empirical study on a set of CLMOPs to evaluate the performance of the aPOCEA. The experimental results show that aPOCEA outperforms other representative multiobjective evolutionary algorithms (MOEAs), verifying the superiority of the aPOCEA.

The remainder of this paper is structured as follows. Section II provides an overview of the fundamental concepts in constrained large-scale multiobjective optimization and reviews existing research on CHTs. Section III presents a detailed description of the proposed aPOCEA. Section IV outlines the experimental setup and presents the experimental results of aPOCEA compared to three representative CMOEAs on benchmark problems. Finally, we conclude in Section V.

II. RELATED WORK

This section overviews the basic concepts in constrained large-scale multiobjective optimization. Subsequently, we discuss the existing CHTs solving CLMOPs.

A. Basic Concept

Definition 1*: Pareto Dominance*

The Pareto dominance is used to evaluate feasible solutions in multiobjective optimization problems. If $\mathbf{x}_1$ and $\mathbf{x}_2$ are two solutions, $\mathbf{x}_1$ dominates $\mathbf{x}_2$ ($\mathbf{x}_1 \prec \mathbf{x}_2$) if the objective values of $\mathbf{x}_1$ are not greater than the objective values of $\mathbf{x}_2$ for all objectives, and the objective values of $\mathbf{x}_1$ are strictly less than the objective values of $\mathbf{x}_2$ for at least one objective. A solution $\mathbf{x}^*$ is Pareto optimal if no other solution can dominate it.

Definition 2*: Constraint Violation (CV)*

The *CV* is widely used to evaluate the extent to which a solution $\mathbf{x}$ violates constraints, and its mathematical description is as follows:

$$CV(\mathbf{x}) = \sum_{j=1}^{J} \max(0,\ g_j(\mathbf{x})) \quad (2)$$

If the *CV* value of $\mathbf{x}$ is greater than 0, then $\mathbf{x}$ is an infeasible solution; if the *CV* value of $\mathbf{x}$ is equal to 0, then $\mathbf{x}$ is a feasible solution.

B. Constraint handling techniques

According to [9], many CHTs have been proposed to solve CMOPs, such as penalty function strategies, objective and constraint separation methods, methods for transforming multiobjective, techniques for transforming CMOPs, hybrid methods, and methods of altering the reproduction operators. Next, we will discuss these technologies in the following subsection.

1) *Penalty function strategies*: These strategies add the penalty term to the objective function to convert the constrained problem into an unconstrained optimization problem [10]. However, determining the penalty factor can be difficult because it may affect algorithm performance [11]. To surpass this difficulty, researchers have proposed adaptive penalty functions [12] that adjust the penalty factor based on feedback from the search process.

2) *Separation of objectives and Constraints:* The basic idea of the goal and constraint separation method is to help the population converge by comparing the goal and constraint separately, which mainly includes the constrained domination principle (CDP) [13], ε constraint [14], and stochastic ranking (SR) [15]. The CDP selected individuals according to their feasibility and Pareto dominance relationship. The ε constrained relaxes constraints using the parameter ε. The *CV* value of a solution less than ε is considered feasible. The SR method compares two solutions based on their objective values using the probability *pf*, and based on their *CV* values with the probability 1-*pf*.

3) *Method for transforming multiobjective:* To solve CMOPs, constraints can be treated as objective functions and transformed into multiobjective optimization problems (MOPs), which could be solved by MOEAs [16]. Peng *et al.* [17] developed a CHT based on directed weights to handle CMOPs by considering *CV* value as an objective.

4) *The method of converting CMOPs:* To solve CMOPs more efficiently, some scholars have attempted to transform CMOPs into other problems (e.g., two-stage optimization problems) and have achieved good results. For example, Wang *et al.* [18] proposed a collaborative differential evolution framework with *m* subpopulations, each of which optimizes a constrained objective.

5) *Hybrid Approaches:* A hybrid method for solving CMOPs integrates several CHTs into different evolutionary stages or subpopulations. Wang *et al.* [19] proposed an adaptive trade-off model based on various search scenarios. If there are no feasible solutions, a multiobjective method handles constraints. A penalty function selects the next generation if the population has feasible and infeasible solutions. If the population only has feasible solutions, the comparison is based on objective values.

6) *Methods of Altering Reproduction Operators:* These methods focus on the design of reproduction operators. Yu *et al.* [20] designed a mutation mechanism for infeasible and feasible solutions. He *et al.* [7] proposed the POCEA, highlighting the importance of offspring generation for finding promising feasible or practical infeasible offspring solutions.

III. PROPOSED *A*POCEA

This section first introduces the basic framework of the proposed aPOCEA, followed by its components in detail.

A. The basic framework of aPOCEA

The pseudo-code for the basic framework of the aPOCEA is shown in Algorithm 1. For a given CLMOP, a set of *N* random solutions is generated to form the initial population *P*. Then, the population evolves iteratively until the termination criteria are met. In each generation, the aPOCEA firstly uses the parent selection strategy to obtain the parent population

Algorithm 1: Basic Framework of a*POCEA*

Input: N (population size), t_{max} (maximum number of generations), V'= {$\mathbf{v}'_1$, ..., $\mathbf{v}'_L$} and V= {v_1, ..., v_L} (two sets of uniformly distributed reference vectors).
Output: P (final population).
P = Initialization(N);
while termination criterion is not fulfilled **do**
  P'= Parent Selection Operator (P);
  Q = Adaptive Paired Offspring Generation Strategy (P', V');
  P = Environmental Selection Operator ($P \cup Q$, V);
end while

and subsequently uses the adaptive pairing offspring strategy to generate the offspring population Q. During this period, the aPOCEA introduces the generation method of uniformly distributed reference vectors in [21] and [22] to generate unit reference vectors and uses them to guide offspring generation in different subregions. Afterward, the aPOCEA uses the environmental selection operator to select promising solutions from the combination of P and Q to update the population. It is worth noting that the size of the reference vector set $\mathbf{V'}$ is $1/K$ of $|\mathbf{V}|$, where K is the neighborhood size.

B. Parent Selection Operator

The reference vector-guided selection strategy proposed in [8] can effectively ensure the convergence and diversity of solutions in subregions, promoting the discovery of several convergent and diverse solutions in global search. Therefore, to ensure the quality of parent solutions, the selection strategy introduced in [8] is used to select solutions with good convergence and diversity to form a preliminary parent population. The pseudo-code of the reference vector-guided selection strategy is shown in Algorithm 2. In this strategy, the detailed formula for calculating the angle penalized distance (*APD*) value is as follows:

$$APD(\theta_{i,j},\mathbf{f}_i') = \left(1+\left(\frac{t}{t_{\max}}\right)^{\alpha}\bullet\frac{M\cdot\theta_{i,j}}{\min\limits_{i\in\{1,\ldots,|V|\},i\neq j}\langle\mathbf{v}_i,\mathbf{v}_j\rangle}\right)\|\mathbf{f}_i'\| \quad (3)$$

where M is the number of objectives, and V is the set of reference vectors. t is the current iterations, and t_{max} is the predefined maximum number of iterations. α is a user-defined parameter, $\mathbf{f}_i$' refers to the ith normalized objective vector of the candidate solution $\mathbf{x}_i$ and $\theta_{i,j}$ indicates the angle between $\mathbf{f}_i$' and $\mathbf{v}_i$.

To effectively utilize the high-quality parent solutions, the parent selection strategy replicates selected solutions and subsequently uses tournament selection to choose feasible solutions with good convergence. Specifically, solutions are sorted using the Pareto dominance relationship based on objective values. For solutions on the same Pareto level, their *CV* values are compared, with the solution having a smaller value of *CV* being selected as a parent solution for the next generation. This ensures that parent solutions have good objective function values and satisfy constraints as much as possible.

C. Adaptive pairwise offspring generations Strategy

During the evolutionary process, if the algorithm overly prioritizes global search ability, it may lead to poor population diversity. Conversely, suppose the algorithm emphasizes its local search ability too much, it may result in a significant gap between the population and the true POF. Balancing the global

Algorithm 2 RVEA Selection

Input: P= {$\mathbf{p}_1$, ..., $\mathbf{p}_N$} (population) and $\mathbf{V}$= {$\mathbf{v}_1$, ..., $\mathbf{v}_W$} (uniform reference vector set).
Output: Parent P', APD.
P'=Φ;
$\mathbf{z}$ = Calculate the ideal point of P;
for i = 1: N **do**
  $\mathbf{f}_i' = \mathbf{f}_i - \mathbf{z}$;
  for j = 1: W **do**
    $\cos\theta_{i,j} = \frac{f_i'\cdot v_j}{\|f_i'\|}$;
  end for
  $\boldsymbol{k} = argmax_{i\in\{1,\ldots,W\}} \cos\theta_{i,j}$;
  $\boldsymbol{S_k} = \boldsymbol{S_k} \cup \{\boldsymbol{p_k}\}$;
end for
for j = 1: W **do**
  for i = 1: $|S_j|$ **do**
    $d_{i,j} = APD(\theta_{i,j}, \mathbf{f}_i')$;
  end for
  $\boldsymbol{k} = argmin_{i\in\{1,\ldots,|S_j|\}} \boldsymbol{d_{i,j}}$;
  $\boldsymbol{P'} = \boldsymbol{P'} \cup \{\boldsymbol{p_k}\}$;
end for

and local search ability of the algorithm is, therefore, a considerable challenge. The paired offspring generation strategy in POCEA balances local and global searches by constructing different subpopulations and taking advantage of infeasible but well-converged solutions, addressing this challenge. The main procedure is as follows:

1) *Solution Association:* To better adapt to irregular POF, avoid premature convergence, increase population diversity, and search for optimal solutions in different directions, the solution association strategy introduced in POCEA divides the population into multiple subpopulations. Fig. 1 shows an example of solution association, where each reference vector is associated with two candidate solutions. For example, the solution sets {$\mathbf{x}_1$, $\mathbf{x}_2$}, {$\mathbf{x}_4$, $\mathbf{x}_6$}, and {$\mathbf{x}_5$, $\mathbf{x}_8$} are associated with $\mathbf{v}_1$', $\mathbf{v}_2$', and $\mathbf{v}_3$', respectively.

2) *Construction of subpopulations:* A large angle between the candidate solution and the reference vector suggests no Pareto-optimal solutions between neighboring reference vectors. In this case, some well-converged candidate solutions from the current population will be merged into the subpopulation for better convergence. In addition, if some well-converged but infeasible solutions exist in the subpopulation, their *CV* value can be relaxed to cross the infeasible region. Based on these two situations, the subpopulation construction strategy introduced in POCEA is presented in Algorithm 3.

3) *Paired Competition:* The paired competition strategy in POCEA considers both convergence degree and *CV* value. If the *CV* values of both solutions are less than ε, they are considered feasible solutions, and the Euclidean distance is further compared to select the solution with a smaller Euclidean distance as the winning solution. If the *CV* values of both solutions exceed ε, then the solution with a smaller *CV* is selected as the winning solution. This method can improve population convergence and select solutions near the feasible region in the objective space, gradually moving the population toward the feasible region.

4) *Offspring generation:* After the competition, solutions can use the particle swarm update strategy in [23] to learn from winners and losers and generate offspring. Genetic operators like simulated binary crossover and polynomial mutation [24]

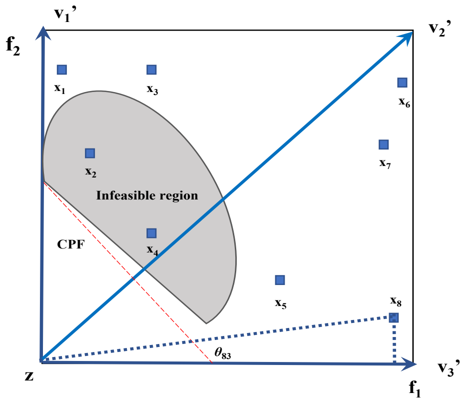

Fig. 1 Example of solution association, $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_8$ are eight candidate solutions, {$\mathbf{v}_1'$, $\mathbf{v}_2'$, $\mathbf{v}_3'$} is a set of uniformly distributed reference vectors, where θ_{83} denotes the angle between the candidate solution $\mathbf{x}_8$ and the reference vector $\mathbf{v}_3'$.

can be used to generate offspring. This strategy is effective for high-dimensional problems as the preference of the winners provides a specific direction.

However, the paired offspring generation strategy in POCEA has some limitations. For example, it requires a manual setting of parameters, which impacts the performance and stability of the algorithm. Additionally, the quality and quantity of parent solutions affect the offspring generation method. There needs to be high-quality parent solutions to ensure the quality and distribution of offspring solutions. To address the above problems, we propose targeted strategies. We introduce a parent selection strategy in sector *B* for the second limitation. We propose a paired offspring generation strategy using parameter adaptation to address the first limitation, as outlined in Algorithm 4. First, the adaptive paired offspring generation dynamically changes the neighborhood size according to the ratio of the current iteration number to the maximum iteration number, and the distribution and quality of solutions, to adaptively balance the global and local search abilities of the algorithm. Second, the *CV* values of all candidate solutions in the current population are calculated, along with the percentage of feasible solutions, and these candidate solutions are associated with different reference vectors to construct a subgroup for each reference vector. Finally, candidate solutions in the same subpopulation are paired, and offspring are generated. Next, we will briefly analyze the parameter *K* and provide a detailed introduction to the proposed parameter adaptation strategy.

a) Parameter analysis: The parameter *K* in the algorithm represents solutions associated with each reference vector. It determines the selection range of solutions in each subpopulation, influencing the trade-off between local and global search capabilities. The larger the value of *K*, the stronger the global search ability and the weaker the local search ability, and vice versa. Specifically, If the value of *K* is larger, then more solutions are selected in each subpopulation, which will have more chances to generate different new solutions, increasing the ability of the algorithm to find the optimal solution in the whole search space. Conversely, if *K* is small, solutions within each subpopulation are more concentrated and have sufficient pressure to converge toward optimal solutions, increasing the ability of the algorithm to search for optimal solutions in local regions.

b) Adaptive strategy for parameter: Based on the parameter analysis, we adopt two adaptive strategies, one based on the ratio of the current number of iterations to the maximum number of iterations and the other based on the distribution and quality of the solutions in the current population. In detail, an enormous *K* value is needed in the initial stage to enhance global search capability, explore more solution space, and avoid falling into local optimal solutions. In contrast, a smaller *K* value is needed in the later stage to enhance local search capability, improve solution accuracy, and find better solutions. At the same time, the adaptive strategy adjusts the *K* value dynamically based on population diversity and solution quality, enabling different search capabilities in different situations. Specifically, a smaller *K* value is needed to enhance local search capability and promote convergence of the population to the true POF when the solutions in the population are more dispersed and high-quality; conversely, a larger *K* value is needed to enhance global search capability and explore more solutions. Based on the above principles, the adaptive strategy introduces the *APD* in RVEA to characterize the quality of solutions in the current population. The smaller the *APD*, the better the quality of solutions and the better their convergence and diversity. Therefore, the convergence and diversity of solutions in the current population are poor when the *APD* value of solutions is large. At this time, the *K* value should remain small to enhance the global search capability of the algorithm and explore more solutions; vice versa.

Algorithm 3 Subpopulation Construction

Input: $V'=\{\mathbf{v}'_1, ..., \mathbf{v}'_L\}$ (unit reference vector set), *P* (current population), *K* (neighborhood size), $\mathbf{d}=(d_1, ..., d_L)$ (Euclidean distance), $\vartheta=(\theta_1,...,\theta_L)$ (angle sets associated to each reference vector).

Output: S (subpopulation sets, Ɛ (*CV* tolerance).

S'=Select *K* solutions with the minimal **d**;
For i=1: L **do**
  if $\frac{1}{K}\sum_{k=1}^{K}\theta'_{k,j} \geq \frac{\pi}{2L}$ **then**
    $S_i=S_i \cup S'$
    CV' = Constraint_Violation (S_i);
    $\varepsilon_i=\max(CV')$;
  else
    CV' = Constraint_Violation (S_i);
    $\varepsilon_i=\min(CV')rf+(1-rf)\frac{1}{k}\sum_{k=1}^{K}CV'_k$;
  end if
end for
$\boldsymbol{\varepsilon}=(\varepsilon_1,\varepsilon_2,...,\varepsilon_L)$;
S= $\{S_1, ..., S_L\}$;

D. Environment selection operator

The reference vector-guided selection strategy proposed in [12] manages convergence and diversity within subspaces of the objective space and handles optimization problems with complex POF characteristics. The environmental selection operator also uses this strategy to select solutions with good convergence and diversity from the mixed parent population *P* and offspring population *Q* as the next-generation population.

IV. Experimentation

In this section, we perform experiments on the PlatEMO [25] platform utilizing MATLAB language to compare our algorithms with three representative MOEAs, including CMOEA/D [26], LMOCSO [6], and POCEA [7], as introduced in Table I. The study examines the performance of algorithms on ten benchmark problems ranging from CF1 to

Algorithm 4 Adaptive Paired Offspring Generation

Input: $V'=\{\mathbf{v}'_1, ..., \mathbf{v}'_L\}$ (reference vector set), P (Parent population), t_{max} (maximum number of generations)
Output: Q (offspring population)
$t = FE/N$ /*FE refers to the number of function evaluations, and N refers to the population size. * /
if $t < 1/2 * t_{max}$ **then**
  K=20;
else
  K=3;
end if
APD=RVEA Selection (P, V);
if $APD \leq 0.5$ **then**
  K=3;
else
  K=20;
end if
$Q=\Phi$;
$rf = \frac{|CV \leq 0|}{|P|}$;/*The rate of feasible solutions*/
d, ϑ, S, F =Solution Association (V', P, K);
S, Ɛ=Subpopulation Construction (rf, K, S, ϑ);
For j=1: K
  $\mathbf{p}_1$, $\mathbf{p}_2$=Randomly select two solutions from S_i;
  $\mathbf{P}_w$, $\mathbf{P}_l$=Competition ($\mathbf{p}_1$, $\mathbf{p}_2$);
  $\mathbf{q}$=Offspring Generation ($\mathbf{P}_w$, $\mathbf{P}_l$);
  $Q=Q \cup \mathbf{q}$;
end for

CF10 [27].

A. Experimental Settings

We used the recommended parameter settings to ensure fair comparisons and implemented all the compared algorithms in PlatEMO [25]. The termination condition for our experiments was the maximum number of function evaluations (*FEs*).

1) *Parameter Settings:* The population size was set to 100 for parents and offspring, and the number of decision variables varied between 100 and 200. The maximum number of *FEs* was 3000 times the number of decision variables (D) for each test instance. For the CMOEA/D algorithm, the neighborhood size (T) was set to N/10, where N is the population size. The probability of choosing parents locally (δ) was set to 0.9, and the maximum number of solutions replaced by each offspring (*nr*) was set to 3. In POCEA, the parameter K was set to 5.

2) *Performance Indicators:* The experiments utilize the inverted generational distance (*IGD*) [28] indicator to evaluate the performance of each CMOEA. The mathematical formula for this metric is as follows:

$$IGD(P^*, \Omega) = \frac{\sum_{x \in P^*} dis(\mathbf{x}, \Omega)}{|P^*|} \tag{4}$$

where P^* represents a set of evenly distributed reference points on the Constrained Pareto Front, Ω denotes the set of obtained feasible and non-dominated solutions, and dis ($\mathbf{x}$, Ω) represents the minimum Euclidean distance between $\mathbf{x}$ and points in Ω. It should be noted that a smaller *IGD* value indicates better performance of the tested algorithm.

In the following experiments, each algorithm is independently run 30 times on each test problem. The results obtained by two CMOEAs are compared using the Wilcoxon rank-sum test [29] at a significance level of 0.05. Symbols "+", "-", and "=" indicate that the compared algorithm is significantly better than, substantially worse than, or statistically tied by the last MOEA.

B. Algorithm Comparison

This section compares aPOCEA with three algorithms (i.e., CMOEA/D, LMOCSO, and POCEA). Table I presents comparison results of the above four algorithms regarding *IGD* values.

These comparison results of the *IGD* values show that the proposed aPOCEA performs excellently in solving ten benchmark problems with different decision dimensions. More specifically, CMOEA/D outperforms aPOCEA on CF1, CF5, CF7, CF9 (D=200), and CF2 (D=100, 200). Conversely, aPOCEA performs better than CMOEA/D on the remaining test instances. The varying performance of the two algorithms can be attributed to their different search strategies and parameter settings. CMOEA/D decomposes the multiobjective problem into single-objective sub-problems and solves them simultaneously, effectively handling complex Pareto fronts and non-convex shapes. On the other hand, aPOCEA employs a parameter adaptive strategy to balance global and local search capabilities, maintaining diversity and convergence. LMOCSO outperforms aPOCEA on CF8 with D=100 and 200 and CF9 with D=100. Conversely, aPOCEA performs better than LMOCSO in other instances. The reason behind these differences in performance could be attributed to the specific characteristics of each test instance. LMOCSO may be better suited for this problem due to its powerful search capability. On the other hand, the parameter adaptive strategy of the aPOCEA may be more effective in other instances where maintaining diversity and convergence is critical for achieving good performance. aPOCEA and POCEA have similar performance on CF2, CF6 (D=100), CF3, CF9 (D=100, 200), and CF10 (D=200) test instances. However, aPOCEA outperforms POCEA in the other test instances. This may be because aPOCEA incorporates a parental selection strategy to enhance the probability of generating promising offspring solutions. Additionally, aPOCEA uses a more adaptive parameter setting mechanism, which allows it to better adapt to the specific characteristics of each test instance.

V. Conclusion

This paper presents an improved MOEA based on POCEA for solving CLMOPs. aPOCEA introduces two critical improvements compared to the original algorithm. Firstly, it replaces the fixed K value with a parameter adaptive strategy, reducing the difficulty and sensitivity of parameter selection and enhancing the robustness and universality of the algorithm. Secondly, aPOCEA incorporates a parental selection strategy to enhance the likelihood of generating promising offspring solutions by selecting high-quality parents, guiding the population towards convergence with the POF, and improving algorithm convergence speed. However, it is worth noting that aPOCEA may not perform as effectively as different algorithms when the dimension of decision variables exceeds 200. Therefore, in the future, we will focus on developing strategies to handle high-dimensional decision variables in CLMOPs.

Acknowledgment

This work was supported in part by the National Key R&D Plan Project under Grant 2022YFB3304700, the National Natural Science Foundation of China under Grants 61988101 and 62203101, the 111 Project 2.0 under Grant B08015, the Liaoning Province Central Leading Local Science, Technology Development Special Project under Grant

TABLE I. STATISTICS OF *IGD* RESULTS ACHIEVED BY CMOEA/D, LMOCSO, AND POCEA ON 30 TEST INSTANCES. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED, AND *NAN* INDICATES THAT NO FEASIBLE SOLUTION IS FOUND.

Problem	D	CMOEA/D	LMOCSO	POCEA	aPOCEA
CF1	100	1.59e-1(7.93e-2) -	1.61e-1(9.30e-3) -	1.97e-1(7.73e-3) -	1.19-1(2.18e-2)
	200	5.05e-2(1.49e-2) +	1.74e-1(7.36e-2) -	2.13e-1(7.73e-3) -	1.23e-1(2.24e-2)
CF2	100	3.33e-1(1.65e-1) +	6.19e-1(1.23e-1) =	5.61e-1(8.24e-2) =	6.39e-1(2.73e-1)
	200	1.70e-1(1.10e-1) +	8.17e-1(4.29e-1) =	8.02e-1(1.29e-1) -	7.71e-1(6.36e-1)
CF3	100	1.67e+0(1.02e+0) -	1.22e+0(5.72e-2) -	5.52e-1(1.02e-1) -	3.91e-1(1.17e-1)
	200	3.86e-1(2.34e-1) =	3.47e+0(2.05e+0) -	4.43e-1(1.85e-1) =	4.19e-1(1.87e-1)
CF4	100	3.99e-1(1.37e-1) =	5.82e+0(3.12e+0) -	3.28e+0(5.90e+0) -	3.58e-1(6.58e-2)
	200	3.76e-1(1.07e-1) =	4.05e+1(3.0e+1) -	1.97e+1(2.19e+0) -	3.63e-1(5.88e-2)
CF5	100	3.94e+1(4.17e+1) -	4.17e+1(1.20e+1) -	2.1e+1(3.21e+0) -	3.58e+0(1.42e+0)
	200	5.43e-1(9.88e-2) +	8.03e+1(8.21e+0) -	4.98e+1(3.57e+1) -	7.30e+0(1.94e+0)
CF6	100	5.08e+0(3.43e+0) -	1.68e+0(1.28e+0) -	5.21e-1(1.18e-1) =	4.70e-1(1.09e-1)
	200	5.97e-1(8.86e-2) -	1.73e+1(1.58e+1) -	2.72e+0(1.89e+0) -	4.81e-1(1.23e-1)
CF7	100	4.89e+1(3.56e+1) -	5.55e+1(1.42e+1) -	4.45e+1(1.33e+1) -	5.17e+0(1.58e+0)
	200	4.43e-1(9.87e-2) +	1.43e+2(1.26e+1) -	1.28e+2(9.12e+1) -	1.19e+1(2.33e+0)
CF8	100	*NAN*(*NAN*) -	2.53e-1(9.20e-2) +	1.12e+0(3.15e-1) -	5.65e-1(8.25e-2)
	200	*NAN*(*NAN*) -	3.13e-1(5.92e-2) +	8.90e-1(2.93e-1) -	5.41e-1(3.98e-2)
CF9	100	7.52e-1(2.44e-1) -	1.70e-1(2.68e-2) +	6.24e-1(1.57e-1) =	6.62e-1(1.87e-1)
	200	3.99e-1(3.44e-1) +	9.33e-1(6.28e-1) =	6.10e-1(1.51e-1) =	6.47e-1(1.40e-1)
CF10	100	*NAN*(*NAN*) -	5.04e-1(5.95e-1) =	*NAN*(*NAN*) -	5.14e-1(0.0e+0)
	200	*NAN*(*NAN*) -	5.33e-1(3.92e-2) =	6.47e-1(8.98e-2) =	5.94e-1(0.0e+0)
+/-/=		6/11/3	3/12/5	0/14/6	

2022JH6/100100055 and the China Postdoctoral Science Foundation under Grant 2023T160086.

REFERENCES

[1] R. G. Regis. High-dimensional constrained discrete multiobjective optimization using surrogates [C] // Machine Learning, Optimization, and Data Science: 6th International Conference, LOD 2020, Siena, Italy, 2020, pp: 203-214.

[2] X. Zhang, Z. Ma, B. Ding, W. Fang, and P. Qian. A coevolutionary algorithm based on the auxiliary population for a constrained large-scale multiobjective supply chain network [J]. Math. Biosci. Eng, 2022, pp: 271-286.

[3] L. M. Antonio and C. A. C. Coello, “Use of cooperative coevolution for solving large scale multiobjective optimization problems,” in Proc. IEEE Congr. Evol. Comput. (CEC), 2013, pp. 2758-2765.

[4] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, and L. Jiao, *et al.*, “A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large scale variables,” IEEE Trans. Evol. Comput., 2016, pp. 275–298.

[5] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, “A framework for large-scale multiobjective optimization based on problem transformation,” IEEE Trans. Evol. Comput., 2018, pp. 260–275.

[6] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, “Efficient Large-Scale Multiobjective Optimization Based on a Competitive Swarm Optimizer,” IEEE Trans. Cybern, 2020, pp. 3696-3708.

[7] C. He, R. Cheng, Y. Tian, X. Zhang, K. C. Tan, and Y. Jin. Paired offspring generation for constrained large-scale multiobjective optimization [J]. IEEE Trans. Evol. Comput, 2020, 25(3): 448-462.

[8] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” IEEE Trans. Evol. Comput., 2016, pp. 773–791.

[9] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, and C. Yue, *et al.*, “A survey on evolutionary constrained multiobjective optimization,” IEEE Trans. Evol. Comput., 2022, pp. 201-221.

[10] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, and Q. Zhang, *et al.*, “Push and pull search for solving constrained multiobjective optimization problems,” Swarm Evol. Comput., 2019, pp. 665–679.

[11] Z. Ma and Y. Wang, “Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons,” IEEE Trans. Evol. Comput., 2019, pp. 972–986.

[12] G. Anescu, "An Adaptive Penalty Function Method for Constrained Continuous Optimization in Population-Based Meta-Heuristic Optimization Methods," 2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 2017, pp. 434-441.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., 2002, pp. 182–197.

[14] T. Takahama and S. Sakai., “Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites,” in Proc. IEEE Congr. Evol. Comput., 2006, pp. 1–8.

[15] T. P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” IEEE Trans. Evol. Comput., vol. 4, no. 3, pp. 284–294, Sep. 2000.

[16] E. Mezura-Montes and C. A. Coello Coello, “Constraint-handling in nature-inspired numerical optimization: Past, present and future,” Swarm Evol. Comput., 2011, pp. 173–194.

[17] C. Peng, H. Liu, and F. Gu, “An evolutionary algorithm with directed weights for constrained multiobjective optimization,” Appl. Soft Comput., 2017, pp. 613–622.

[18] J. Wang, G. Liang, and J. Zhang, “Cooperative differential evolution framework for constrained multiobjective optimization,” IEEE Trans. Cyber., 2019, pp. 2060–2072.

[19] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, “An adaptive tradeoff model for constrained evolutionary optimization,” IEEE Trans. Evol. Comput., 2008, pp. 80–92.

[20] X. Yu, X. Yu, Y. Lu, G. G. Yen, and M. Cai, “Differential evolution mutation operators for constrained multiobjective optimization,” Appl. Soft Comput., 2018, pp. 452–466.

[21] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” SIAM J. Optim., 1998, pp. 631–657.

[22] C. He, L. Pan, H. Xu, Y. Tian, and X. Zhang, “An improved reference point sampling method on Pareto optimal front,” in Proc. IEEE Congr. Evol. Comput. (CEC), Vancouver, BC, Canada, 2016, pp. 5230–5237.

[23] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, “Efficient large-scale multiobjective optimization based on a competitive swarm optimizer,” IEEE Trans. Cybern., 2020, pp. 3696–3708.

[24] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” Complex Syst., 1995, pp. 115–148.

[25] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB platform for evolutionary multiobjective optimization [educational forum],” IEEE Comput. Intell. Mag., 2017, pp. 73–87.

[26] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, “A survey of multiobjective evolutionary algorithms based on decomposition,” IEEE Trans. Evol. Comput., 2017, pp. 440–462.

[27] C. He, R. Cheng and D. Yazdani, "Adaptive Offspring Generation for Evolutionary Large-Scale Multiobjective Optimization," IEEE Trans. Syst. Man. Cybern, 2022, pp. 786-798.

[28] C. A. Coello Coello and N. C. Cortes, “Solving multiobjective optimization problems using an artificial immune system,” Genet. Program. Evolvable Mach., 2005, pp. 163–190.

[29] Z. Ma and Y. Wang, “Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons,” IEEE Trans. Evol. Comput., 2019, pp. 972–986.