

基于多种群分区辅助的用于解决约束多目标优化问题的协同进化算法

范海儒

广东工业大学, 数学与统计学院, 广东 广州

收稿日期: 2024年2月22日; 录用日期: 2024年3月13日; 发布日期: 2024年4月25日

摘要

在解决约束多目标优化问题时, 需要同时考虑多个优化目标以及问题的约束条件, 这给问题的解决带来了很大的挑战, 主要体现在解的可行性、收敛性和多样性三个方面。为了应对这些挑战, 本文提出了一种新的约束多目标进化优化算法, 该算法采用了三个阶段和多个辅助种群的设计。具体来说, 第一阶段中, 算法首先利用不受约束的种群引领其他种群进行搜索, 在第二阶段, 每个辅助种群致力于解决单个约束条件下的多目标优化问题。最后, 本文设计了一种分区辅助策略, 根据不同种群在每个分区的表现来调整辅助种群的参与度。为了评估算法的表现和有效性, 我们将其与当前较为优秀的四个算法在 DASC MOPs 测试基上进行了比较。实验结果表明, 我们的算法具有出色的表现。

关键词

约束多目标, 协同进化, 分区辅助, 进化算法

A Cooperative Evolutionary Algorithm for Solving Constrained Multi-Objective Optimization Problems Based on Multi-Population Partition Assistance

Hairu Fan

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: Feb. 22nd, 2024; accepted: Mar. 13th, 2024; published: Apr. 25th, 2024

Abstract

In tackling constrained multi-objective optimization problems, it is necessary to consider multiple optimization objectives as well as the constraints of the problem, which poses significant challenges to finding solutions. These challenges are mainly reflected in the feasibility, convergence, and diversity of solutions. To address these challenges, this paper proposes a novel constrained multi-objective evolutionary optimization algorithm, which employs a three-stage approach and multiple auxiliary populations. Specifically, in the first stage, the algorithm utilizes an unconstrained population to guide the search of other populations. In the second stage, each auxiliary population focuses on solving the multi-objective optimization problem under a single constraint. Finally, a partition-driven strategy is designed to adjust the participation of auxiliary populations in each partition based on their performance, thereby enhancing the efficiency of the main population in obtaining superior solutions. To evaluate the performance and effectiveness of the algorithm, we compare it with four currently classic and outstanding algorithms on the DASCOPs test suite. The experimental results demonstrate the excellent performance of our algorithm.

Keywords

Constrained Multi-Objective, Co-Evolution, Partition Assistance, Evolutionary Algorithm

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

现实应用中有很多优化问题包含了几个相互冲突的目标和多个复杂的约束条件，如机电力配送系统规划[1]、组合经济排放调度问题[2]、车辆路径规划[3]。这类问题被称为受约束的多目标优化问题。其数学模型可以用下面的公式表达：

$$\begin{aligned} \min_x \quad & f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \\ & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \dots, q \end{aligned} \quad (1)$$

其中， $\mathbf{x} = (x_1, x_2, \dots, x_D) \in \Omega$ 表示的是决策空间维度为 D 维的决策解； $\Omega \subseteq \mathbb{R}^D$ 代表的是决策空间； $f: \Omega \rightarrow \mathbb{R}^M$ 表示的是维度为 M 的目标解；其中还包括了 q 个等式约束 $h_j(\mathbf{x})$ 和 p 个不等式约束 $g_j(\mathbf{x})$ 。

从上述数学模型可以看出，约束多目标优化问题涉及多个优化目标，因此并非具有唯一解，而是通过一组解集展示各个优化目标之间的最优权衡。多目标进化算法，如基于非支配排序[4]、指标[5]、分解[6]等方法，是一种随机启发式算法，能够在一次运行中生成一组解集。受遗传进化的启发，该算法通过不断迭代更新朝向更优的进化方向搜索，最终产生一组较优的解集。然而，由于约束条件的存在，这些条件会改变目标空间的搜索景观，导致不可行解域的出现。在这种情况下，单纯使用多目标进化算法就不再有效，因此需要各种约束处理机制来应对这一挑战。

目前, 约束处理机制主要分为六类。第一类是基于罚函数的方法, 这些方法主要依据约束违反程度来构建惩罚项。将惩罚项融入目标函数中, 将约束多目标优化问题转化为无约束的形式。文献[7]提出了一种新的方法, 将非支配排序遗传算法与无参数惩罚方法相结合, 以持续搜索更优解集。第二类是基于目标与约束分离的方法。文献[4]提出了一个非支配排序方法, 该方法在比较两个不同个体时优先考虑可行解, 若两个个体都是可行解则选择非支配解, 若两个个体都是不可行解则选择约束违反度小的个体。第三类是基于多目标的方法, 将约束条件也视为优化目标处理, 从而将约束多目标优化问题转化为无约束的多目标优化问题。第四类方法是混合方法, 将进化算法与数学规划相结合, 特别是在处理等式约束时, 这种方法可以减少空间搜索的成本。文献[8]提出了一种混合优化技术, 将遗传算法与基于内点方法的局部搜索策略相结合。第五类是基于改变繁殖算子的方法[9], 该类方法专注于设计高效且特定的算子来处理繁殖过程。第六类是将约束多目标优化问题转化为其他问题, 这是处理约束多目标优化问题的主要流行方法。这类方法可以有效利用不同优化问题或不同进化阶段的信息为其余问题和进化阶段提供参考信息, 又可以细分为以下两类:

第一类, 基于将优化问题转化为多个阶段的优化问题的方法, 文献[10]提出了一种双阶段的算法框架, 该框架分为推和拉两个阶段, 在推的阶段, 算法旨在使种群穿过不可行区域到达非约束帕累托前沿; 而拉的阶段, 采用改进的 ε 约束方法来回搜索约束帕累托前沿。第二类方法是将约束多目标优化问题转化为协同优化问题, 核心思想是使用不同的种群处理不同的优化问题, 并相互分享经验。文献[11]提出了一个协同进化框架, 来解决约束多目标优化问题, 其中一个种群搜索约束帕累托前沿, 另一个种群忽略约束条件寻找非约束帕累托前沿, 并且这两个种群相互协助解决约束多目标优化问题。文献[12]设计了双存档进化算法, 其中一个存档以收敛为导向, 另一个存档以多样性为导向, 用于探索未开发的区域并保持多样性。文献[13]研究了约束之间的关系, 根据单个约束的帕累托前沿和它们的共同帕累托前沿的关系确定约束的优先级, 并利用优先级最强的约束对应的种群进行辅助。尽管以上的算法都表现出色, 但文献[11]的两个种群在约束和非约束帕累托前沿有较大分离时, 合作效率较低。文献[13]中, 单纯地根据各个约束条件对应的种群的支配排序来确定约束的优先级可能会忽略掉其他约束条件所能提供的信息。为解决上述问题, 本文提出了一种约束多目标进化优化算法, 使用多个辅助种群在不同分区下提供不同辅助强度。算法采用了三个阶段的进化过程, 每个阶段能够高效地发掘信息。本文所提出的算法与 4 个约束多目标优化算法在 3 个测试基下进行比较, 结果表明, 所提的算法具有较好的表现。

2. 提出的算法

2.1. 算法框架

算法 1 概述了本文提出算法的步骤, 该算法主要用到了 $C + 2$ 个种群, 其中包含 1 个负责解决原问题的种群, 1 个不考虑约束条件的多目标优化问题的种群还有 C 个只考虑一个约束条件的多目标优化问题的种群, 为了方便后文的阅读和表述, 在本文的后续部分, 我们称这些只考虑一个约束条件的多目标优化问题为原问题的子约束问题。每个种群的规模设置为 N , 设置算法的最大函数评估次数为 MaxFEs 次。该算法主要分成三个阶段, 第一个阶段是探索阶段, 其主要目的是利用不考虑约束的种群引领其他种群向前探索。第二个阶段是开发阶段, 在这一阶段, 多个种群之间会进行信息交互, 以实现知识的迁移。第三个阶段是聚焦阶段, 其目标是确保每个辅助种群能够为主种群提供更加有效的辅助信息。在算法 1 的 1~8 行是一些初始化操作, C 代表约束条件的个数, Stage 初始化为 1 表示当前阶段为第一个阶段——探索阶段, 种群 P_1 是不考虑约束的辅助种群, 通过初始化 P_1 并且赋值给其他的 $C + 1$ 个种群。在算法的 9~20 行是三个阶段的内容, 当满足算法的终止条件, 算法返回最终的主种群 P 。

算法 1: 算法步骤

输入: 种群规模 N ; 最大函数评估次数 MaxFEs ; 权重向量集 W ;
 输出: 种群 P ;
 1 $C \leftarrow$ 约束条件的个数;
 2 $\text{Stage} \leftarrow 1$;
 3 $P_1 \leftarrow$ 随机初始化种群(N);
 4 评估种群 P_1 并初始化函数评估次数 $\text{FEs} \leftarrow N$;
 5 $P \leftarrow P_1$;
 6 for $i \leftarrow 1$ to C do
 7 $SP_i \leftarrow P_1$;
 8 end
 9 While 未满足终止条件:
 10 if $\text{Stage} == 1$ then
 11 $P, P_1, SP_1, SP_2, \dots, SP_C \leftarrow$ 通过算法 2 更新主种群和辅助种群;
 12 $\text{FEs}, \text{Stage} \leftarrow$ 通过算法 2 更新当前函数评估次数以及阶段数;
 13 elseif $\text{Stage} == 2$ then
 14 $P, P_1, SP_1, SP_2, \dots, SP_C \leftarrow$ 通过算法 3 更新主种群和辅助种群;
 15 $\text{FEs}, \text{Stage} \leftarrow$ 通过算法 3 更新当前函数评估次数以及阶段数;
 16 else
 17 $P, P_1, SP_1, SP_2, \dots, SP_C \leftarrow$ 通过算法 5 更新主种群和辅助种群;
 18 $\text{FEs} \leftarrow$ 通过算法 5 更新当前函数评估次数;
 19 end
 20 end
 21 return P ;

2.2. 探索阶段

探索阶段的主要作用是选择利用不考虑约束条件的辅助种群解决最简单的无约束多目标优化问题, 从而大致探索整个可行域并达到潜在的可行域。这样有助于穿越传统的约束多目标优化算法所无法突破的不可行域。需要注意的是, 在这一阶段, 只有种群 P_1 是产生子代的, 并且这些子代会与其他种群共享。这么做的原因是为了节省计算资源, 并且对于其他 C 个辅助种群而言, 在早期阶段产生的子代尚未展现出较好的收敛性, 因此可以认为它们在当前阶段的作用相对较小。算法 2 中的 1~3 行对辅助种群 P_1 进行了父代选择并且产生子代最终通过环境选择更新种群。在 4~10 行中, 将 P_1 子代共享给其余的种群, 最后通过环境选择完成种群更新。12~14 行主要是进行了算法阶段数的更新, 这个更新策略用到了文献[10]的状态转换策略。如公式(2)所示, rz_k 代表最理想点和最不理想点变化率的最大值, rz_k 小于阈值 ϵ 就满足转换条件。其中最理想点和最不理想点的变化率用公式(3)和公式(4)表示。

$$r_k \equiv \max \{ rz_k, rn_k \} \leq \epsilon \quad (2)$$

$$rz_k = \max_{i=1, \dots, m} \left\{ \frac{|z_i^k - z_i^{k-1}|}{\max \{ |z_i^{k-1}|, 1e-6 \}} \right\} \quad (3)$$

$$rn_k = \max_{i=1, \dots, m} \left\{ \frac{|n_i^k - n_i^{k-1}|}{\max\{|n_i^{k-1}|, 1e-6\}} \right\} \quad (4)$$

算法 2: 探索阶段

输入: 种群规模 N ; 最大函数评估次数 MaxFEs ; 当前函数评估次数 FEs ; 约束条件个数 C ; 主种群 P ; 辅助种群 $P_1, SP_1, SP_2, \dots, SP_C$;

输出: 主种群 P ; 辅助种群 SP_1, SP_2, \dots, SP_C ; 当前函数评估次数 FEs ; 当前阶段数 Stage ;

1 $\text{MatingParent}_1 \leftarrow$ 通过二元锦标赛选择从 P_1 中选出 N 个个体作为父代;

2 $\text{Off}_1 \leftarrow$ MatingParent_1 通过 GA 算子产生 N 个子代个体;

3 $P_1 \leftarrow P_1, P_1 \cup \text{Off}_1$, 不考虑约束条件来评估 P_1 的个体;

4 $P_1 \leftarrow$ 通过环境选择, 更新种群 P_1 ;

5 $P \leftarrow P \cup \text{Off}_1$, 考虑全部约束条件来评估 P 的个体;

6 $P \leftarrow$ 通过环境选择, 更新种群 P ;

7 for $j \leftarrow 1$ to C do

8 $SP_j \leftarrow SP_j \cup \text{Off}_j$, 考虑第 j 个约束条件来评估 SP_j 的个体;

9 $SP_j \leftarrow$ 通过环境选择, 更新种群 SP_j ;

10 end

11 更新函数评估次数 $\text{FEs} \leftarrow \text{FEs} + N$

12 if P_1 满足公式(2)

13 $\text{Stage} \leftarrow \text{Stage} + 1$;

14 end

2.3. 开发阶段

在探索阶段转换到开发阶段后, 由于种群 P_1 的引领, 其余种群也能够到达相对收敛的和可行的区域。探索阶段的主要作用是让除了种群 P_1 之外的种群通过进化从而获得更优的解, 在这个过程中, 由于各个种群用于处理不同的问题, 能够产生不同偏好的子代个体, 因此具有丰富有价值的信息。种群之间能够互相共享自己产生的子代, 为其他种群提供有价值的信息, 从而辅助主种群的解集具有更好的收敛性、可行性和多样性。需要注意的是, 在开发阶段已经停止了辅助种群 P_1 的更新, 而是通过其余的只考虑原问题的子约束问题的辅助种群来进行辅助, 当遇到约束帕累托前沿与非约束帕累托前沿有着较大的差距的优化问题时, 这样的设计能够在一定程度上弥补文献[11]中辅助种群在后期不能有效地提供辅助信息的缺陷。在算法 3 的 1~7 行主要是种群 P, SP_1, \dots, SP_C 产生子代并且将它们地子代都放到子代共享池中, 8~13 行中, 每个种群都将自身与共享子代池合, 随后通过环境选择更新为新的种群。15~17 行是用于更新进化阶段的, 当 SP_1, \dots, SP_C 都处于相对停滞时, 满足转换条件开始进入第三阶段。

2.4. 聚焦辅助阶段

2.4.1. 聚焦辅助策略

当完成了探索阶段后, 处理原问题子约束问题的辅助种群基本上已经探索到属于它们的约束帕累托前沿了。并且, 原问题对应的可行域是它各个子约束问题的可行域的交集, 这些交集的帕累托前沿跟这些子约束问题对应的帕累托前沿会有一些的关系。本文将原问题的帕累托前沿和它子约束问题对应的帕

累托前沿之间的关系简化为两类：1) 子约束问题的帕累托前沿与原问题的帕累托前沿有交集；2) 子约束问题的帕累托前沿与原问题的帕累托前沿没有交集。并且得出以下结论，子约束问题对原问题的辅助效率很大程度上与它和原问题之间的帕累托前沿的相近程度决定。由此本文提出了一个聚焦辅助的策略，用于分配每个处理子约束问题的辅助种群对主种群的辅助强度。

算法 3: 开发阶段

输入：种群规模 N ；最大函数评估次数 MaxFEs ；当前函数评估次数 FEs ；约束条件个数 C ；主种群 P ；辅助种群 SP_1, SP_2, \dots, SP_C ；

输出：主种群 P ；辅助种群 SP_1, SP_2, \dots, SP_C ；当前函数评估次数 FEs ；当前阶段数 Stage ；

1 $\text{MatingParent} \leftarrow$ 通过二元锦标赛选择从 P 中选出 N 个个体作为父代；

2 $\text{Off} \leftarrow \text{MatingParent}$ 通过 GA 算子产生 N 个子代个体；

3 for $j \leftarrow 1$ to C do

4 $\text{MatingParent}_j \leftarrow$ 通过二元锦标赛从 SP_j 选择出 N 个父代个体；

5 $\text{OffSpring}_j \leftarrow$ 通过 DE 算子产生 N 个子代个体；

6 $\text{Off} \leftarrow \text{Off} \cup \text{OffSpring}_j$

7 end

8 $P \leftarrow P \cup \text{Off}$ ，考虑全部约束条件来评估 P 的个体；

9 $P \leftarrow$ 通过环境选择，更新种群 P ；

10 for $j \leftarrow 1$ to C do

11 $SP_j \leftarrow SP_j \cup \text{Off}$ ，考虑第 j 个约束条件来评估 SP_j 的个体；

12 $SP_j \leftarrow$ 通过环境选择，更新种群 SP_j ；

13 end

14 更新函数评估次数 $\text{Fes} \leftarrow \text{FEs} + (C+1) \times N$ ；

15 if SP_1, \dots, SP_C 均满足公式(2)

16 $\text{Stage} \leftarrow \text{Stage} + 1$ ；

17 end

首先，将目标空间均匀地分成 K 份，计算并统计每个辅助种群在每个分区的可行解的个数，得到矩阵(5)，其中 $a_{c,k}$ 代表第 c 个辅助种群在第 k 个分区的可行解的个数。需要注意的是，这里的可行指的是针对原问题的可行，而非针对子约束问题的可行。

$$\begin{pmatrix} a_{11} & \dots & a_{1K} \\ \vdots & \ddots & \vdots \\ a_{C1} & \dots & a_{CK} \end{pmatrix} \quad (5)$$

随后通过公式(6)计算第 c 个辅助种群在每个分区下的可行解比例，并用矩阵(7)表示每个辅助种群在每个分区对应的可行解比例。

$$\text{rate}_{ck} = \frac{a_{ck}}{\sum_{i=1}^K a_{ci}} \quad (6)$$

$$\text{Rate} = \begin{pmatrix} \text{rate}_{11} & \dots & \text{rate}_{1K} \\ \vdots & \ddots & \vdots \\ \text{rate}_{C1} & \dots & \text{rate}_{CK} \end{pmatrix} \quad (7)$$

针对可能发生某些子约束问题对应的帕累托前沿在某些分区重合从而导致资源浪费的情况, 本文认为应该将这些分区交给最擅长在这些分区处理优化问题的辅助种群, 由此根据公式(8)计算每个辅助种群在分区 k 下的辅助资源数。并用矩阵(9)表示每个辅助种群在每个分区的辅助资源, res_{ck} 本质上就是第 c 个辅助种群在分区 k 中能容纳的最大个体数。

$$res_{ck} = \begin{cases} rate_{ck} \times N & \text{if } rate_{ck} = \max_{1 \leq c \leq C} (rate_{ck}) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$resource = \begin{pmatrix} res_{11} & \dots & res_{1K} \\ \vdots & \ddots & \vdots \\ res_{C1} & \dots & res_{CK} \end{pmatrix} \quad (9)$$

随后设计新的环境选择策略。核心思想是将种群的个体都分配到它所擅长的区间, 从而发挥每个辅助种群的辅助优势。算法 4 中 1~3 行主要是计算每个候选集的适应值和所属的分区, 方便从每个分区筛选出对应资源数的优秀个体更新到下一代。4~12 行是对每个分区进行遍历, 假如在当前分区中候选解的个数小于其对应的资源数, 那么就将该区的全部个体合并到 *Population_new* 中, 否则的话就选择最优的对应的前资源数个个体。13~17 主要是当 *Population_new* 个体数不满 N 个时, 便从剩余的候选解中选出最优的 *leave* 个个体加入到 *Population_new* 中。

算法 4: 环境选择策略

输入: 种群规模 N ; 候选解集 *Population*; c ; *resource*; 分区权重 W ; 权重个数 K ;

输出: 更新后种群 *Population_new*;

1 *Fitness* \leftarrow 针对第 c 个约束条件, 通过非支配排序计算 *Population* 每个个体的适应值;

2 *Region* \leftarrow 通过欧氏距离计算每个个体所在的区域;

3 *Population_new* $\leftarrow \emptyset$;

4 for $i = 1:K$ do

5 $n \leftarrow$ 第 c 个辅助种群在第 i 个分区的资源数 *resource*[c][i];

6 *feasible_number* \leftarrow *Population* 在第 i 个分区中针对第 c 个约束条件的可行解个数;

7 if $n \geq feasible_number$

8 将属于分区 i 的 *feasible_number* 个个体添加到 *Population_new*;

9 else

10 将属于分区 i 的适应值排名前 n 名的个体添加到 *Population_new*;

11 end

12 end

13 $N_new \leftarrow$ *Population_new* 中的个体数;

13 if $N_new < N$

14 $leave \leftarrow N - N_new$;

15 *diff* \leftarrow *Population* - *Population_new*;

16 从 *diff* 中选出适应值排名前 $leave$ 名的个体添加到 *Population_new* 中;

17 end

18 return *Population_new*;

2.4.2. 聚焦辅助阶段算法流程

聚焦辅助阶段的目的是让每个辅助种群根据每个分区所分配的辅助资源来进一步为主种群提供有价

值信息。算法 5 的第 1 行主要是计算每个辅助种群的个体相对原问题而言的可行解的比例，如果一个辅助种群中不存在可行解，说明它的帕累托前沿与原问题的帕累托前沿没有交集，因此在后期不能很好地提供辅助作用，因此便不考虑让其产生子代以此节省计算资源和减少信息干扰。第 3 行主要是根据公式 (8) 得出每个辅助种群在各个分区的辅助资源数。3~12 行是让主种群和辅助种群产生子代并且都合并添加到共享子代池当中。13~18 行是更新主种群。并且利用新设计的环境选择策略更新辅助种群。

算法 5: 聚焦辅助阶段

输入: 种群规模 N ; 最大函数评估次数 MaxFEs ; 当前函数评估次数 FEs ; 约束条件个数 C ; 主种群 P ; 辅助种群 SP_1, SP_2, \dots, SP_C ; 权重向量集 W ;

输出: 主种群 P ; 辅助种群 SP_1, SP_2, \dots, SP_C ; 当前函数评估次数 FEs ;

```

1 Feasible_Rate  $\leftarrow$  计算每个辅助种群的可行解个数;  $F \leftarrow 0$ ;
2 resource  $\leftarrow$  通过聚焦辅助策略分配每个种群在每个分区的计算资源;
3 MatingParent  $\leftarrow$  通过二元锦标赛选择从  $P$  中选出  $N$  个个体作为父代;
4 Off  $\leftarrow$  MatingParent 通过 GA 算子产生  $N$  个子代个体;
5 for j  $\leftarrow$  1 to C do
6   if Feasible_Rate[j] > 0
7      $F \leftarrow F + 1$ ;
8     MatingParentj  $\leftarrow$  通过二元锦标赛从  $SP_j$  选择出  $N$  个父代个体;
9     OffSpringj  $\leftarrow$  通过 DE 算子产生  $N$  个子代个体;
10    Off  $\leftarrow$  Off  $\cup$  OffSpringj;
11  end
12 end
13 P  $\leftarrow$  P  $\cup$  Off, 考虑全部约束条件来评估  $P$  的个体;
14 P  $\leftarrow$  通过环境选择, 更新种群  $P$ ;
15 for j  $\leftarrow$  1 to C do
16    $SP_j \leftarrow SP_j \cup$  Off, 考虑第  $j$  个约束条件来评估  $SP_j$  的个体;
17    $SP_j \leftarrow$  算法 4 进行环境选择, 更新种群  $SP_j$ ;
18 end
19 更新函数评估次数  $\text{FEs} \leftarrow \text{FEs} + (F + 1) \times N$ ;

```

3. 实验分析

本文主要选取广泛用于测试约束多目标优化算法性能的 DASC MOP 测试基[14], 该测试基需要平衡约束多目标优化问题中常见的可行性、收敛性、多样性三大难题, 包含有三大类的约束条件, 分别是控制收敛性的条件、控制可行性的条件和控制多样性的条件。本文使用的评价指标是 IGD 指标[15], 该指标主要反映了一个算法找到的帕累托前沿的收敛性和多样性。

为了检验所提算法的有效性, 本文与四个前沿的算法[10] [11] [12] [13]进行比较, 种群大小设置为 100, 最大评估次数设置为 300,000 次。所有测试问题都运行 30 轮。需要注意的是, 对比的算法的参数都参考对应文献的设置。

3.1. 实验结果

算法在测试问题上的 IGD 指标平均值，如表 1 所示。

Table 1. The average value of the IGD metric of the algorithm on the test problems
表 1. 算法在测试问题上的 IGD 指标平均值

问题	约束个数	CCMO	C-TAEA	PPS	C3M	PACMO
DASCMP1	11	7.0611e-1 (3.80e-2)-	1.7669e-1 (1.21e-2)-	1.8406e-1 (2.33e-1)-	4.4179e-3 (4.32e-3)=	3.0924e-3 (2.60e-4)
DASCMP2	11	2.3945e-1 (2.50e-2)-	37.6141e-2 (2.51e-2)-	5.1617e-3 (1.43e-4)-	4.1369e-3 (7.75e-5)+	4.3704e-3 (9.65e-5)
DASCMP3	11	3.1565e-1 (4.62e-2)-	1.3624e-1 (2.58e-2)-	2.8881e-1 (1.06e-1)-	6.8422e-2 (8.34e-2)-	1.9093e-2 (4.39e-3)
DASCMP4	11	1.2432e-3 (5.34e-4)+	1.0040e-2 (2.16e-3)-	1.5161e-1 (6.82e-2)-	1.4273e-3 (2.93e-4)+	2.2025e-3 (2.04e-3)
DASCMP5	11	2.7024e-3 (4.90e-5)+	7.0255e-3 (4.44e-4)-	4.2054e-3 (3.16e-4)+	1.2886e-2 (5.24e-2)=	6.9428e-3 (1.86e-2)
DASCMP6	11	2.2403e-2 (8.69e-3)=	2.3315e-2 (2.41e-3)=	2.2225e-1 (2.90e-1)-	1.0107e-1 (2.18e-1)=	8.3968e-2 (1.43e-1)
DASCMP7	7	3.0968e-2 (6.46e-4)-	3.8140e-2 (8.94e-4)-	5.2107e-2 (7.90e-3)-	4.1895e-2 (5.52e-3)-	2.8156e-2 (4.65e-4)
DASCMP8	7	4.0353e-2 (1.00e-3)-	6.0485e-2 (1.36e-2)-	6.4678e-2 (7.87e-3)-	5.4690e-2 (1.10e-2)-	3.5761e-2 (8.62e-4)
DASCMP9	7	3.4014e-1 (8.53e-2)-	2.0814e-1 (7.02e-2)-	1.0823e-1 (9.42e-2)-	4.0824e-2 (1.06e-3)-	3.5435e-2 (5.84e-4)
+/-/=		2/6/1	0/8/1	1/8/0	2/4/3	

3.2. 实验结果分析

从表 1 可以看出，本文提出的算法 PACMO 在 DASCMP 测试基上取得了最好的实验效果，证明了它的有效性，尤其是在 DASCMP3 和 DASCMP7-DASCMP9 中遥遥领先于对比的算法，并且说明该算法还在维度为 3 维的目标空间中也有着很好的表现。其次，算法在 DASCMP4 和 DASCMP5 中稍稍逊色于 CCMO，这两个问题中的约束条件个数有 11 个，故本算法比 CCMO 需要消耗更多的资源在辅助种群中。总的而言，本文提出的算法能够较高效地利用到辅助种群提供的信息。

4. 结束语

本文提出了一种约束多目标进化优化算法，采用了三个阶段和多个辅助种群的设计。其核心思想是在第一阶段利用无约束的种群引领其他种群搜索潜在的可行区域，在第二阶段通过协同进化方式在辅助种群和主种群之间传递和共享信息，在第三阶段则采用了基于分区的辅助策略，以更好地让辅助种群为主种群提供可行信息。实验结果表明，本文提出的算法表现良好。但是，在计算资源有限的前提下，优化问题的约束条件越多，所需的辅助种群也就越多，如此平分到每个种群的资源也就随之减少，可能会导致效果略显不足，在 DASCMP1-DASCMP6 中，测试问题的约束条件多达 11 个，显然本文提出算法在 DASCMP2、DASCMP4、DASCMP5 中表现较差，然而在 DASCMP7、DASCMP8、DASCMP9 中测试问题中的约束条件个数为 7 个，本文所提出的算法效果显著优于其余的算法。因此，在未来的研究中，将进一步改进和优化算法，特别关注处理约束较多情况的问题。

参考文献

- [1] Rivas-Davalos, F. and Irving, M. (2005) An Approach Based on the Strength Pareto Evolutionary Algorithm 2 for Power Distribution System Planning. In: Coello Coello, C.A., Hernández Aguirre, A. and Zitzler, E., eds., *Evolutionary Multi-Criterion Optimization*, Springer, Berlin, Heidelberg, 707-720. https://doi.org/10.1007/978-3-540-31880-4_49
- [2] Basu, M. (2011) Economic Environmental Dispatch Using Multi-Objective Differential Evolution. *Applied Soft Computing*, **11**, 2845-2853. <https://doi.org/10.1016/j.asoc.2010.11.014>
- [3] Jozefowicz, N., Semet, F. and Talbi, E.G. (2008) Multi-Objective Vehicle Routing Problems. *European Journal of Operational Research*, **189**, 293-309. <https://doi.org/10.1016/j.ejor.2007.05.055>
- [4] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182-197. <https://doi.org/10.1109/4235.996017>
- [5] Liu, Z.-Z., Wang, Y. and Wang, B.-C. (2021) Indicator-Based Constrained Multiobjective Evolutionary Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **51**, 5414-5426. <https://doi.org/10.1109/TSMC.2019.2954491>
- [6] Li, K., Chen, R., Fu, G. and Yao, X. (2019) Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **23**, 303-315. <https://doi.org/10.1109/TEVC.2018.2855411>
- [7] Jadaan, O.A., Rao, C.R. and Rajamani, L. (2009) Solving Constrained Multi-Objective Optimization Problems Using Non-Dominated Ranked Genetic Algorithm. *Proceedings of 2009 Third Asia International Conference on Modelling & Simulation*, Bandung, Indonesia, 25-29 May 2009, 113-118. <https://doi.org/10.1109/AMS.2009.38>
- [8] Kelner, V., Capitanescu, F., Léonard, O. and Wehenkel, L. (2008) A Hybrid Optimization Technique Coupling an Evolutionary and a Local Search Algorithm. *Journal of Computational and Applied Mathematics*, **215**, 448-456. <https://doi.org/10.1016/j.cam.2006.03.048>
- [9] Xu, B., Duan, W., Zhang, H. and Li, Z. (2020) Differential Evolution with Infeasible-Guiding Mutation Operators for Constrained Multi-Objective Optimization. *Applied Intelligence*, **50**, 4459-4481. <https://doi.org/10.1007/s10489-020-01733-0>
- [10] Fan, Z., Li, W.J., Cai, X.Y. and Li, H. (2019) Push and Pull Search for Solving Constrained Multi-Objective Optimization Problems. *Swarm and Evolutionary Computation*, **44**, 665-679. <https://doi.org/10.1016/j.swevo.2018.08.017>
- [11] Tian, Y., Zhang, T., Xiao, J.H., Zhang, X.Y. and Jin, Y.C. (2021) A Coevolutionary Framework for Constrained Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, **25**, 102-116. <https://doi.org/10.1109/TEVC.2020.3004012>
- [12] Li, K., Chen, R., Fu, G. and Yao, X. (2019) Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **23**, 303-315. <https://doi.org/10.1109/TEVC.2018.2855411>
- [13] Zou, J., Sun, R.Q., Liu, Y., et al. (2024) A Multipopulation Evolutionary Algorithm Using New Cooperative Mechanism for Solving Multiobjective Problems with Multiconstraint. *IEEE Transactions on Evolutionary Computation*, **28**, 267-280. <https://doi.org/10.1109/TEVC.2023.3260306>
- [14] Fan, Z., Li, W., Cai, X., Li, H., Zhang, Q., Deb, K. and Goodman, E. (2019) Difficulty Adjustable and Scalable Constrained Multi-Objective Test Problem Toolkit. *Evolutionary Computation*, **28**, 1-28. https://doi.org/10.1162/evco_a_00259
- [15] Bosman, P.A.N. and Thierens, D. (2003) The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, **7**, 174-188. <https://doi.org/10.1109/TEVC.2003.810761>