

基于分解的约束多目标进化算法

张鹏懿

广东工业大学数学与统计学院, 广东 广州

收稿日期: 2023年4月21日; 录用日期: 2023年5月23日; 发布日期: 2023年5月30日

摘要

现实世界中为平衡多方利益, 需要进行多目标优化的研究, 然而实际问题中多数问题往往带有约束条件, 富有挑战性。在研究中, 首要解决的问题为如何处理在优化过程中出现的非可行解。我们认为非可行解在优化过程中是重要的, 其包含着种群进化方向的信息, 需在优化过程中保存适当比例的非可行解。为此本文提出了一种基于分解的约束多目标进化算法, 根据进化过程中种群变化来自适应产生权重保留一部分非可行解引导种群收敛。为验证所提算法的性能, 选取了24个测试问题, 4个对比算法, 经数值实验证明了所提算法的有效性。

关键词

约束多目标, 进化算法, 基于分解的, 自适应权重

Decomposition-Based Constrained Multi-Objective Evolutionary Algorithm

Pengyi Zhang

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

Received: Apr. 21st, 2023; accepted: May 23rd, 2023; published: May 30th, 2023

Abstract

In the real world, it is necessary to study multi-objective optimization in order to balance multi-interests. However, most of the practical problems are often challenging with constraints. How to deal with the infeasible solution in the optimization is the most important problem to be solved in the research. We think that the infeasible solution is important in optimization, and it contains the information of population evolution direction, so it is necessary to keep an appropriate proportion of the infeasible solution in optimization. In this paper, a decomposition-based constrained multi-objective evolutionary algorithm is proposed, which adaptively generates weights according

to population changes in evolution and preserves some infeasible solutions to guide population convergence. In order to verify the performance of the proposed algorithm, 24 test problems and 4 comparison algorithms are selected. Numerical experiments show the effectiveness of the proposed algorithm.

Keywords

Constrained Multi-Objective, Evolutionary Algorithm, Decomposition-Based, Adaptive Weight

Copyright © 2023 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

随着科技进步和社会发展,越来越多的问题需要在多个目标之间进行权衡取舍。例如,在工业领域中,我们常常需要考虑成本、质量和生产效率等多个指标,而在城市规划中,则需要平衡经济发展、环境保护和社会公正等多个方面的利益。在这种情况下,传统的单目标优化方法往往无法满足我们的需求。为了解决这个问题,研究者们开始转向多目标优化方法,并将其应用于各种实际问题中。然而,在实际问题中,多目标优化往往受到各种约束条件的限制。

在过去几十年里,研究者们提出了许多不同的约束多目标优化算法[1]。K. Deb.提出了 CDP (Constrained Dominance Principle)约束处理技术[2],因其简单有效的处理约束,成为现今适用范围最广泛的约束处理技术,Peng [3]等人将约束违反程度作为一个新的目标,设计了一种新的基于有向权重的 CHT 来求解 CMOP,其中使用分别分布在可行和可行区域中的两种类型的权重来引导搜索到有希望的区域。Tian 等人[4]提出了一种求解 CMOP 的共同进化框架(CCMO),其中一个种群用来求解原始的 CMOP,即搜索 CPF (constrained Pareto front),而另一个种群忽略约束来寻找 UPF (unconstrained Pareto front)。此外,这两个群体相互协助求解 CMOP。Fan [5]等人提出了 PPS (Push and pull search)框架。在 push 阶段的目标是引导种群穿过不可行区域到达 UPF。在 pull 阶段,采用改进的 ϵ 约束方法来指导种群搜寻 CPF。基于 PPS 框架,将 CMOP 分解为一组简单的子问题,每个子问题对应一个子种群,并将 PPS 框架应用于每个子种群以解决相关的子问题。

然而由于约束条件的存在,其求解难度相较于无约束优化问题大幅增加。现存约束多目标进化算法,大多针对某一类问题表现较好,缺乏一种鲁棒性更强的约束多目标优化算法,这是目前亟需解决的问题。

2. 预备知识

2.1. 约束多目标优化模型

不失一般性,由 m 个目标函数, n 维决策变量以及若干约束组成的约束多目标优化问题,其数学模型表达如下:

$$\begin{aligned} \min f(x) &= (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.t.} \quad &\begin{cases} g_j(x) \leq 0, & j = 1, 2, \dots, p \\ h_j(x) = 0, & j = p+1, \dots, q \\ x \in \mathbb{S} \subseteq [Lb, Ub] \end{cases} \end{aligned} \quad (1)$$

其中 $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ 为 n 维决策变量, 决策变量 x 的取值范围由:

$$\mathbb{S} = \{x \mid g_j(x) \leq 0, h_j(x) = 0, j = 1, 2, \dots, q\} \quad (2)$$

$$[Lb, Ub] = \{x = (x_1, x_2, \dots, x_n)^T \mid l_k \leq x_k \leq u_k, k = 1, 2, \dots, n\} \quad (3)$$

共同决定, 称为决策空间。

另外, $g_j(x) \leq 0$ 定义了 p 个不等式约束条件, $h_j(x) = 0$ 定义了 $q - p$ 个等式约束条件。对每一个决策变量 x , 其对第 j 个约束条件的约束违反程度(constraint violation degree)定义为:

$$CV_j(x) = \begin{cases} \max(0, g_j(x)), & j = 1, 2, \dots, p \\ \max(0, |h_j(x) - \delta|), & j = p + 1, p + 2, \dots, q \end{cases} \quad (4)$$

其中 δ 通常为一个非常小的正数。通过对每一个约束条件的约束违反程度相加, 可得其总的约束违反程度为:

$$CV(x) = \sum_{j=1}^q CV_j(x) \quad (5)$$

如果一个决策变量的 CV 值是 0, 那么则可以说明此解为可行解, 反之此解为非可行解。

若有两个可行解 x_1 和 x_2 , 对任一 $i \in \{1, 2, \dots, m\}$ 都有 $f_i(x_1) \leq f_i(x_2)$ 且存在至少一个 $k \in \{1, 2, \dots, m\}$ 有 $f_k(x_1) < f_k(x_2)$, 则称 x_1 支配 x_2 。如果一个解 x 不被其他任何解支配, 则称此解为问题(1)的 Pareto 最优解, 所有 Pareto 最优解的集合构成 PS (Pareto optimal solutions), 对应至目标空间为 Pareto 前沿 PF (Pareto front)。

2.2. MOEA/D-M2M 算法模型

MOEA/D-M2M [6] 是一个基于分解的多目标优化算法, 具有较强的维持种群多样化的能力, 求得的解分布性较为均匀。其中心思想为将问题分解为若干个子问题来同时解决和优化。具体为在目标空间中产生 K 个单位中心权重向量 v^1, v^2, \dots, v^K , 将目标空间分为 K 个子区域 $\Omega_1, \Omega_2, \dots, \Omega_K$, Ω_i 为:

$$\Omega_i = \{u \mid \langle u, v^i \rangle \leq \langle u, v^j \rangle \text{ for any } j = 1, 2, \dots, K\} \quad (6)$$

其中 $\langle u, v^i \rangle$ 为目标向量与中心权重向量 V^i 的角度。基于此类分区, 问题(1)可转化为:

$$\begin{aligned} \min f(x) &= (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.t.} & \begin{cases} g_j(x) \leq 0, & j = 1, 2, \dots, p \\ h_j(x) = 0, & j = p, p + 1, \dots, q \\ x \in \mathbb{S} \subseteq [Lb, Ub] \\ f(x) \in \Omega_K \end{cases} \end{aligned} \quad (7)$$

3. 算法设计

3.1. 多目标优化方法

单目标约束优化问题中, 常使用将约束违反程度 CV 记为另外一个目标函数, 将其转化为一个无约束的双目标优化问题的方法求解。那么借鉴此思路, 在 m 维约束多目标优化问题当中, 可将约束违反程度 CV , 记为第 $m + 1$ 个目标函数, 由此可将问题(1)转化为:

$$\min f(x) = (f_1(x), f_2(x), \dots, f_m(x), CV(x))^T \quad (8)$$

Peng 等人根据此方法设计了一种基于动态权重的约束多目标优化算法[3], 其通过不断调整不可行权重与可行超平面角度, 实现在种群进化过程中向可行超平面收敛。但是由于此种权重调整策略默认随着种群的进化, 个体违反约束程度呈正相关降低, 导致其容易在接近 PF 处出现连续不可行区域中陷入局部最优。本文考虑引入自适应权重产生策略, 根据种群中可行解比例调整可行与不可行权重数量, 着重保留有希望的不可行解, 引导种群跳出局部最优。

3.2. 权重产生策略

Table 1. Weight generation strategy

表 1. 权重产生策略

算法 1. 权重产生策略

输入: 当代可行解 F_t , 不可行解 F'_t ; 预期可行解比例 E_t , 实际可行解比例 E'_t

输出: 权重向量 W_t

Step 1. 初始化产生可行、不可行权重 W'_t 、 W''_t

Step 2. if $F'_t \leq N \cdot (1 - E_t)$ then

Step 3. 在可行超平面上均匀取 F_t 数量个 W'_t

Step 4. $W''_t = F'_t$

Step 5. else

Step 6. 在可行超平面上均匀取 $N \cdot E_t$ 数量个 W'_t

Step 7. 在 F'_t 中通过 Maxmin 方法选出 $N \cdot (1 - E_t)$ 个解作为 W''_t

Step 8. end if

Step 9. $W_t = W'_t \cup W''_t$

首先引入预期可行解比例 E_t , 实际可行解比例 E'_t 。其中 E_t 为种群进化至第 t 代时期望达到的可行解所占比例, E'_t 为第 t 代时可行解占实际种群的比例。

不同于[3]产生固定的可行与不可行权重数量, 此权重产生策略将动态调整可行与不可行权重的数量。为确保进化方向是收敛的, 那么就要求可行权重所占比例要与进化代数 t 呈正相关性, 否则算法无法收敛。权重产生策略如表 1 所示, 其中 W'_t 为在可行超平面中均匀产生的单位可行权重向量。

3.3. Adaptive Weight 算法

AW (Adaptive weight)算法框架基于 MOEA/D-M2M [6], 其算法框架图如表 2 所示:

Table 2. AW algorithm framework

表 2. AW 算法框架

算法 2. AW 算法框架

输入: N : 种群规模; K : 分区数; $MaxFE$: 最大评估次数

输出: 存档集 A

Continued

-
- Step 1. 初始化种群 Q_0 , K 个子区域 $\Omega_i (i=1,2,\dots,K)$ $A = \emptyset$, $t=1$
- Step 2. while 不满足终止条件 do
- Step 3. 使用杂交算子产生子代 O_t
- Step 4. $C_t = O_t \cup Q_t$
- Step 5. 通过**算法 1** 产生权重向量
- Step 6. 通过 ASF 于 C_t 选出下一代个体 Q_{t+1}
- Step 7. 更新存档集 A , $t=t+1$
- Step 8. end while
-

Step 3 中选用 DE/rand/1/bin [7]算子产生后代

Step 6 中对于任一给定的 $m+1$ 维权重向量 $w = (w_1, w_2, \dots, w_{m+1})$, 每一个解都可通过以下:

$$ASF = \max_{i=1}^{m+1} \left(\frac{f_i(x) - z_i}{w_i} \right) \quad (9)$$

公式计算出 ASF [8]数值。其中 $z = (z_1, z_2, \dots, z_m + 1)$ 为参考点, z_i 表示当前种群中所出现的 $f_i(x)$ 最小值。

存档集 A 用以存放种群进化过程中产生的可行解。若存档集 A 中数量少于等于 N , 则直接更新存档集 A , 否则将通过(9)式选取 N 个具有较好收敛性和散布性的可行解来更新存档集 A 。

4. 数值实验

4.1. 参数设置

参数的设定与调整对于算法的运行是有较大影响的, 因此, 对算法进行合理的参数设置是十分有必要的。经过数值分析拟将若干参数设定为: 种群数量 $N=100$, 分区数 $K=\sqrt{N}=10$, 权重数量 = 100, 最大评估次数 $MaxFE=200000$, 最大进化代数 $t_{max} = N/MaxFE = 2000$, DE 算子中 $CR=0.1$ 、 $F=0.8$ 。另外对于预期可行解比例定义如下:

$$E_t = \begin{cases} 0.49 + 0.625 * (t/t_{max}) & t/t_{max} \leq 0.8 \\ 1 & \text{其他} \end{cases} \quad (10)$$

4.2. 测试问题集

本实验中选用 C_DTLZ 系列[9]以及 MW 系列[10]测试集。

通过对比无约束优化问题 UPF (unconstrained Pareto front)和带约束优化问题 CPF (constrained Pareto front), 可发现两类 PF 并不完全相同, 大致可分为以下四种情况如图 1 所示:

- Type I: CPF 与 UPF 完全相同;
- Type II: CPF 完全是 UPF 的一部分;
- Type III: CPF 一部分与 UPF 相同;
- Type IV: CPF 与 UPF 完全不同。

根据上述条件, 可将两类测试集中的问题进行分类, 如表 3 所示:

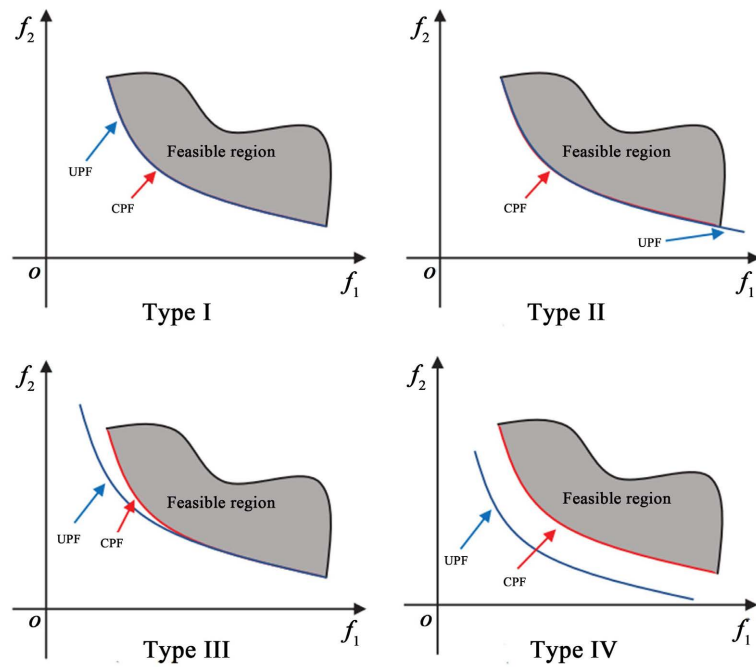


Figure 1. Classification of CPF and UPF

图 1. CPF 与 UPF 分类

Table 3. CPF classification of test problems

表 3. 测试问题 CPF 分类

Type	Problem
I	MW2
	MW4
	MW14
	C1_DTLZ1
	C1_DTLZ3
	DC1_DTLZ1
	DC1_DTLZ3
II	MW1
	MW5
	MW6
	MW8
	C2_DTLZ2
	DC1_DTLZ1
	DC1_DTLZ3
	DC3_DTLZ1
DC3_DTLZ3	

Continued

	MW3
III	MW7
	MW10
	MW13
	MW9
IV	MW11
	MW12
	C3_DTLZ4

4.3. 性能指标

4.3.1. IGD 指标

IGD 指标[11]评估的是从真实 Pareto 最优集中的点到实际求得的非劣解集合的平均欧氏距离。它用一组足够多的点来衡量种群的收敛性与散布性。一般来讲 IGD 值越小, 所得到的种群更接近 PF, 且具有更好的散布性。给定 P^* 为均匀散布在 PF 上的点的集合, P 为算法所求得的最优解集所包含的点, 那么 IGD 公式可定义为:

$$\text{IGD}(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (11)$$

4.3.2. HV 指标

HV 指标[12]评估的是算法实际求得的非劣解集合中的解与参考点之间形成空间的超体积, 同样能够同时衡量种群的收敛性与散布性。一般来讲 HV 值越大, 所得到的种群则更接近 PF, 且具有更好的散布性。给定参考点 Z , X 为算法实际所求得的非劣解集, x 为 X 中的个体, 那么 HV 公式可定义为:

$$\text{HV}(X, P) = \bigcup_{x \in X} v(x, P) \quad (12)$$

由于 HV 指标在计算过程中并未使用真实 PF 数据, 所以其可适用于一些 PF 未知问题的算法性能检验当中, 而 IGD 指标则无法做到。

4.4. 实验结果

本次试验中, 我们使用了 DW [3]、CCMO [4]、PPS [5]、IDBEA [13]作为 AW 的对比算法, 并给出了对应的对比实验结果。因与 DW 拥有较为类似的算法机制, 所以须重点关注权重产生策略不同之处所带来的算法性能的影响。为尽可能消除偶然误差, 每个算法将在 Platemo [14]平台上对测试问题运行 30 轮, 并取 IGD、HV 结果平均值。

Table 4. The algorithm runs average IGD results in 30 rounds of test problems

表 4. 算法在测试问题 30 轮运行平均 IGD 结果

Type	Problem	M	D	AW	DW	CCMO	PPS	IDBEA
I	MW2	2	15	1.15E-01	1.32E-01	1.43E-01	3.59E-01	5.59E-01
	MW4	3	15	6.08E-02	4.92E-02	5.53E-02	5.77E-02	6.60E-01

Continued

	MW14	3	15	6.08E-02	2.15E-01	1.03E-01	1.29E-01	2.71E+00
	C1_DTLZ1	3	7	2.34E-02	2.53E-02	3.31E-02	2.59E-02	4.20E-01
I	C1_DTLZ3	3	12	4.86E+00	5.63E+00	6.50E-02	7.69E-02	8.01E+00
	DC2_DTLZ1	3	7	2.80E-02	5.12E-02	5.14E-02	2.85E-02	2.82E-01
	DC2_DTLZ3	3	12	4.86E-01	5.68E-01	6.95E-02	5.73E-01	1.20E+00
	MW1	2	15	2.60E-03	2.04E-02	2.93E-03	2.66E-03	5.93E-01
	MW5	2	15	6.63E-03	5.21E-01	4.18E-03	7.43E-01	7.38E-01
	MW6	2	15	7.37E-02	7.68E-01	5.28E-01	3.18E-01	7.84E-01
	MW8	3	15	8.08E-02	1.57E-01	1.96E-01	9.74E-02	9.12E-01
II	C2_DTLZ2	3	12	5.40E-02	5.73E-02	8.81E-01	5.70E-02	6.82E-01
	DC1_DTLZ1	3	7	1.36E-02	2.36E-02	3.68E-02	2.13E-02	3.94E-02
	DC1_DTLZ3	3	12	5.23E-02	5.91E-02	2.25E-01	2.48E-01	1.37E-01
	DC3_DTLZ1	3	7	1.24E-01	9.89E-02	8.52E-03	2.06E-01	4.67E-01
	DC3_DTLZ3	3	12	2.10E+00	2.06E+00	6.74E-01	1.97E+00	3.01E+00
	MW3	2	15	4.46E-03	9.76E-02	5.35E-03	6.47E-03	4.50E-01
	MW7	2	15	5.01E-03	7.23E-03	3.98E-03	5.73E-03	5.59E-01
III	MW10	2	15	1.36E-01	4.81E-01	7.42E-01	2.90E-01	6.15E-01
	MW13	2	15	2.63E-01	6.69E-01	6.55E-01	4.76E-01	1.39E+00
	MW9	2	15	8.64E-03	8.68E-03	1.50E-02	1.44E-02	4.93E-01
	MW11	2	15	7.35E-03	2.35E-02	6.39E-03	7.54E-03	1.16E+00
IV	MW12	2	15	5.03E-03	7.55E-03	7.09E-02	6.81E-03	6.10E-01
	C3_DTLZ4	3	12	1.28E-01	1.45E-01	1.10E-01	1.30E-01	3.64E-01

Table 5. The algorithm runs average HV results in 30 rounds of test problems**表 5.** 算法在测试问题 30 轮运行平均 HV 结果

Type	Problem	M	D	AW	DW	CCMO	PPS	IDBEA
	MW2	2	15	3.87E-01	4.14E-01	3.90E-01	1.87E-01	1.17E-01
	MW4	3	15	8.12E-01	8.30E-01	8.19E-01	8.17E-01	1.63E-01
	MW14	3	15	4.89E-01	4.38E-01	4.64E-01	4.47E-01	2.43E-02
I	C1_DTLZ1	3	7	8.30E-01	8.20E-01	7.97E-01	8.21E-01	9.43E-02
	C1_DTLZ3	3	12	1.79E-01	1.65E-01	5.26E-01	5.15E-01	0.00E+00
	DC2_DTLZ1	3	7	8.21E-01	7.63E-01	8.24E-01	8.09E-01	1.24E-01
	DC2_DTLZ3	3	12	1.03E-02	1.29E-02	5.16E-01	1.18E-02	0.00E+00
	MW1	2	15	4.89E-01	4.72E-01	4.89E-01	4.88E-01	4.13E-02
	MW5	2	15	3.15E-01	1.61E-01	3.22E-01	9.09E-02	0.00E+00
II	MW6	2	15	2.43E-01	6.53E-02	1.01E-01	1.57E-01	3.09E-02
	MW8	3	15	4.44E-01	3.26E-01	2.56E-01	4.18E-01	5.37E-02

Continued

	C2_DTLZ2	3	12	4.92E-01	5.09E-01	4.89E-01	4.99E-01	8.98E-02
	DC1_DTLZ1	3	7	6.17E-01	6.17E-01	6.17E-01	6.02E-01	5.65E-01
II	DC1_DTLZ3	3	12	4.59E-01	4.31E-01	2.35E-01	3.53E-01	3.71E-01
	DC3_DTLZ1	3	7	2.64E-01	3.29E-01	5.22E-01	1.79E-01	4.13E-02
	DC3_DTLZ3	3	12	0.00E+00	0.00E+00	1.73E-01	0.00E+00	0.00E+00
	MW3	2	15	5.44E-01	4.89E-01	5.43E-01	5.43E-01	1.67E-01
	MW7	2	15	4.12E-01	4.11E-01	4.13E-01	4.12E-01	1.72E-01
III	MW10	2	15	3.36E-01	1.87E-01	7.17E-02	2.85E-01	1.14E-01
	MW13	2	15	2.57E-01	2.09E-01	2.14E-01	2.23E-01	1.81E-01
	MW9	2	15	3.88E-01	3.92E-01	3.84E-01	3.75E-01	1.05E-01
	MW11	2	15	4.48E-01	4.43E-01	4.47E-01	4.48E-01	9.48E-02
IV	MW12	2	15	6.03E-01	6.01E-01	5.46E-01	6.03E-01	1.35E-01
	C3_DTLZ4	3	12	7.77E-01	7.80E-01	7.79E-01	7.70E-01	5.92E-01

表 4、表 5 可以看出与 DW 相比, AW 算法几乎在任何测试问题上的表现都要更好; 与其他算法相比较, 在 MW14、MW6、MW8、DC1_DTLZ3、MW10、MW13、MW9 中 AW 的性能也有明显的优势。尤其以 MW10、MW8 最为显著。

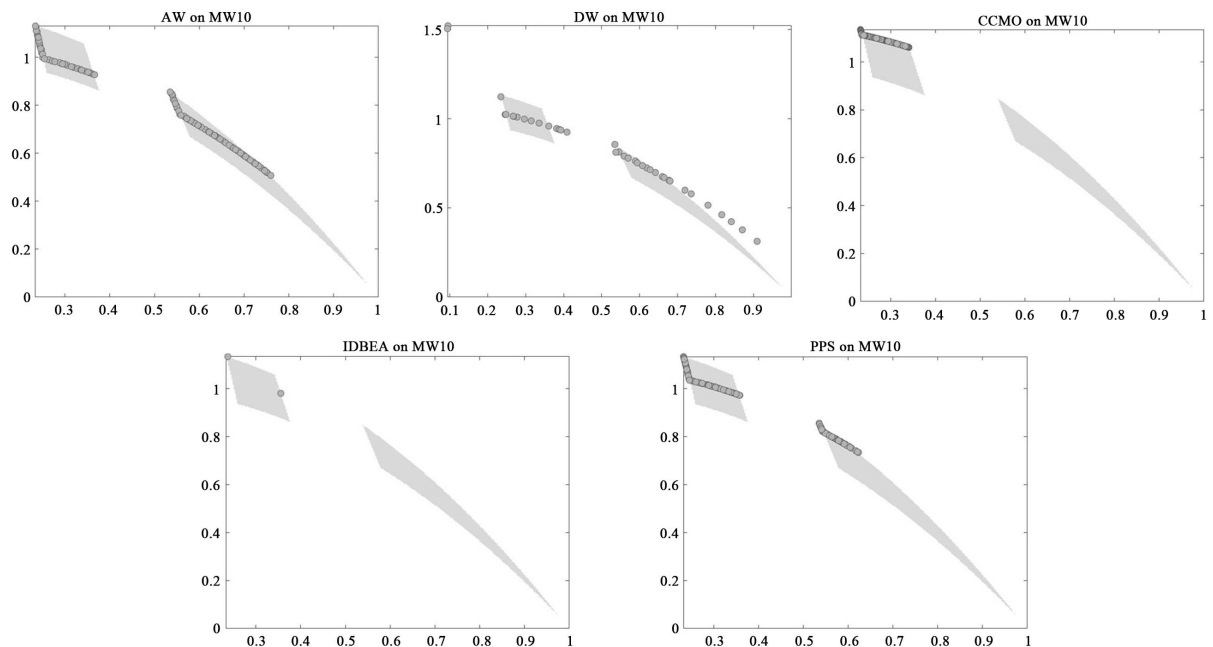


Figure 2. Optimal solutions of each algorithm in two-dimensional problem MW10

图 2. 各算法在二维问题 MW10 中所求的最优解

从图 2 可以看出在二维测试问题 MW10 中可以看出 AW 算法最终求得的非劣解相较于对比算法展现出更好的收敛性和散布性。由于 AW 算前期产生了较多不可行权重, 保留了较多有希望的不可行解帮助

算法在进化前期阶段快速跳出局部可行域, 更快进入 PF 可行域中, 有效的提升算法的收敛性能。同时由于采用 M2M [6] 框架权重在各个子区域中均匀产生, 种群进化过程中保留了更多分布均匀的解, 有效提升了各个阶段种群的散布性。

从图 3 可以看出在三维测试问题 MW8 中 AW 算法相较于其他对比算法也具有更佳的收敛性。由于在自适应权重产生策略, 在种群进化后期可行权重数量增加引导种群向 PF 收敛, 相较于 DW [3] 的固定可行权重数量可展现出更好的收敛性能。

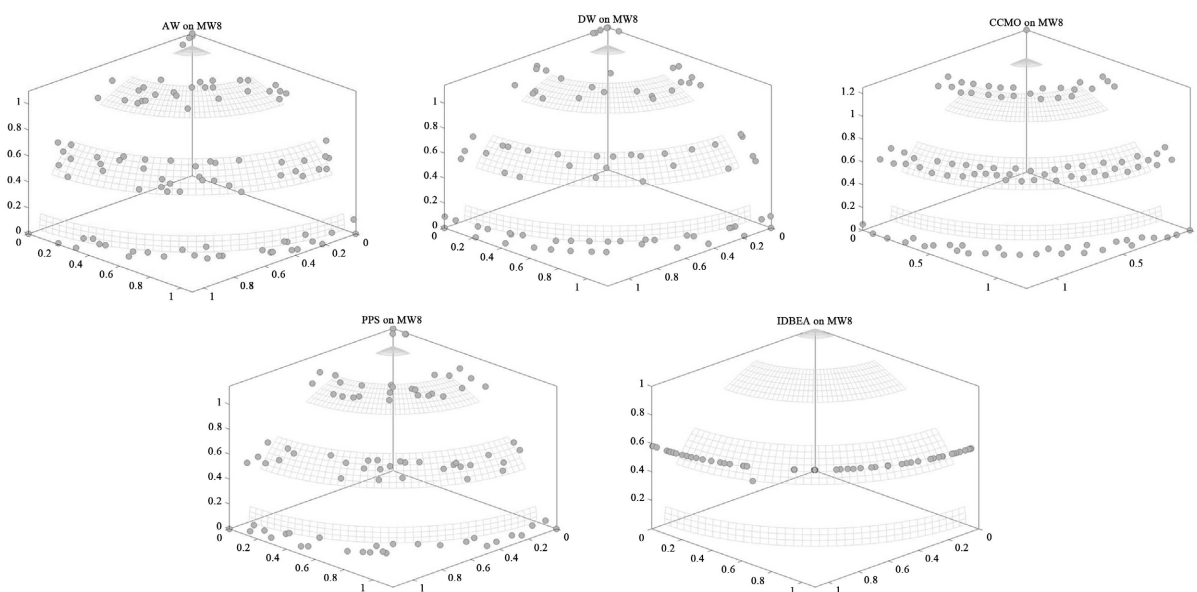


Figure 3. Optimal solutions of each algorithm in three-dimensional problem MW8

图 3. 各算法在三维问题 MW8 中所求的最优解

5. 总结

在本文中, 我们提出了不同的权重产生策略以处理多目标约束优化问题, 并将此整合为新的命名为 AW 算法。在数值实验中, AW 算法在测试问题中展现出了较好的性能, 说明了算法的有效性和适应性。

未来的工作中, 可能会继续针对权重产生策略、参数敏感性分析等进行改进, 并将其应用拓展至更高维度问题中。

参考文献

- [1] Liang, J., et al. (2023) A Survey on Evolutionary Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **27**, 201-221. <https://doi.org/10.1109/TEVC.2022.3155533>
- [2] 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>
- [3] Peng, C., Liu, H.-L. and Gu, F. (2017) An Evolutionary Algorithm with Directed Weights for Constrained Multi-Objective Optimization. *Applied Soft Computing*, **60**, 613-622. <https://doi.org/10.1016/j.asoc.2017.06.053>
- [4] Tian, Y., Zhang, T., Xiao, J., Zhang, X. and Jin, Y. (2020) A Coevolutionary Framework for Constrained Multi-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, **25**, 102-116. <https://doi.org/10.1109/TEVC.2020.3004012>
- [5] Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K. and Goodman, E.D. (2017) 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>

-
- [6] Liu, H., Gu, F. and Zhang, Q. (2014) Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems. *IEEE Transactions on Evolutionary Computation*, **18**, 450-455. <https://doi.org/10.1109/TEVC.2013.2281533>
- [7] Jia, G., Wang, Y., Cai, Z. and Jin, Y. (2013) An Improved $(\mu + \lambda)$ -Constrained Differential Evolution for Constrained Optimization. *Information Sciences*, **222**, 302-322. <https://doi.org/10.1016/j.ins.2012.01.017>
- [8] Wierzbicki, A.P. (1980) The Use of Reference Objectives in Multiobjective Optimization. *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter*, West Germany, 20-24 August 1979, 468-486. https://doi.org/10.1007/978-3-642-48782-8_32
- [9] Jain, H. and Deb, K. (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Non-Dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, **18**, 602-622. <https://doi.org/10.1109/TEVC.2013.2281534>
- [10] Ma, Z. and Wang, Y. (2019) Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons. *IEEE Transactions on Evolutionary Computation*, **23**, 972-986. <https://doi.org/10.1109/TEVC.2019.2896967>
- [11] Zhang, Q., Zhou, A. and Jin, Y. (2008) Rm-meda: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. *IEEE Transactions on Evolutionary Computation*, **12**, 41-63. <https://doi.org/10.1109/TEVC.2007.894202>
- [12] Zitzler, E. and Thiele, L. (1998) Multiobjective Optimization Using Evolutionary Algorithms: A Comparative Case Study. *Parallel Problem Solving from Nature, PPSN V: 5th International Conference*, Amsterdam, 27-30 September 1998, 292-301. <https://doi.org/10.1007/BFb0056872>
- [13] Asafuddoula, M., Ray, T. and Sarker, R. (2015) A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization. *IEEE Transactions on Evolutionary Computation*, **19**, 445-460. <https://doi.org/10.1109/TEVC.2014.2339823>
- [14] Tian, Y., Cheng, R., Zhang, X.Y. and Jin, Y.C. (2017) PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Computational Intelligence Magazine*, **12**, 73-87. <https://doi.org/10.1109/MCI.2017.2742868>