



融合聚类和小生境搜索的多模态多目标优化算法

顾清华, 唐慧, 李学现, 江松

引用本文:

顾清华, 唐慧, 李学现, 江松. 融合聚类和小生境搜索的多模态多目标优化算法[J]. 智能系统学报, 2023, 18(5): 1127–1141.

GU Qinghua, TANG Hui, LI Xuexian, et al. A multimodal multi-objective optimization algorithm with clustering and niching searching[J]. *CAAII Transactions on Intelligent Systems*, 2023, 18(5): 1127–1141.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202204040>

您可能感兴趣的其他文章

融合分区和局部搜索的多模态多目标优化

Multimodal multi-objective optimization combining zoning and local search

智能系统学报. 2021, 16(4): 774–784 <https://dx.doi.org/10.11992/tis.202010026>

目标空间映射策略的高维多目标粒子群优化算法

Multi-objective particle swarm optimization algorithm based on an objective space mapping strategy

智能系统学报. 2021, 16(2): 362–370 <https://dx.doi.org/10.11992/tis.202006042>

基于目标空间分解和连续变异的多目标粒子群算法

Decomposition and continuous mutation-based multi-objective particle swarm optimization

智能系统学报. 2019, 14(3): 464–470 <https://dx.doi.org/10.11992/tis.201711015>

基于NSGA-II的扩展置信规则库激活规则多目标优化方法

NSGA-II-based EBRB rules activation multi-objective optimization

智能系统学报. 2018, 13(3): 422–430 <https://dx.doi.org/10.11992/tis.201710012>

一种求解多模态复杂问题的混合和声差分算法

Hybrid algorithm based on harmony search and differential evolution for solving multi-modal complex problems

智能系统学报. 2018, 13(2): 281–289 <https://dx.doi.org/10.11992/tis.201612030>

膜系统下的一种多目标优化算法

Multi-objective optimization algorithm based on membrane system

智能系统学报. 2017, 12(5): 678–683 <https://dx.doi.org/10.11992/tis.201706013>

DOI: 10.11992/tis.202204040

网络出版地址: <https://kns.cnki.net/kcms2/detail/23.1538.TP.20230427.1729.002.html>

融合聚类和小生境搜索的多模态多目标优化算法

顾清华^{1,2,3}, 唐慧^{1,2}, 李学现^{2,4}, 江松^{1,2,3}

(1. 西安建筑科技大学 资源工程学院, 陕西 西安 710055; 2. 西安市智慧工业感知计算与决策重点实验室, 陕西 西安 710055; 3. 西安优迈智慧矿山研究院有限公司, 陕西 西安 710055; 4. 西安建筑科技大学 管理学院, 陕西 西安 710055)

摘要:针对多模态多目标优化中种群多样性难以维持和所得等价 Pareto 最优解数量不足问题, 提出一种融合聚类和小生境搜索的多模态多目标优化算法 (multimodal multi-objective optimization algorithm with clustering and niching searching, CSSMPIO)。首先利用基于聚类的特殊拥挤距离非支配排序方法 (clustering-based special crowding distance, CSCD) 初始化种群; 引入自适应物种形成策略生成稳定的小生境, 在不同的小生境子空间并行搜索和保持等价 Pareto 最优解; 采用特殊拥挤距离非支配排序策略实现个体选优、精英学习策略避免过早收敛。通过在 14 个多模态多目标函数上进行测试, 并与 7 种新提出的多模态多目标优化算法进行对比实验以及 Wilcoxon 秩和检验发现, CSSMPIO 的总体性能优于对比算法。最后将算法用于基于地图的测试问题, 进一步证明了算法的有效性。

关键词:多模态多目标优化; 鸽群优化算法; 聚类策略; 小生境搜索; 非支配排序; 精英学习策略; 多样性; 地图测试应用

中图分类号: TP301 **文献标志码:** A **文章编号:** 1673-4785(2023)05-1127-15

中文引用格式: 顾清华, 唐慧, 李学现, 等. 融合聚类和小生境搜索的多模态多目标优化算法 [J]. 智能系统学报, 2023, 18(5): 1127-1141.

英文引用格式: GU Qinghua, TANG Hui, LI Xuexian, et al. A multimodal multi-objective optimization algorithm with clustering and niching searching[J]. CAAI transactions on intelligent systems, 2023, 18(5): 1127-1141.

A multimodal multi-objective optimization algorithm with clustering and niching searching

GU Qinghua^{1,2,3}, TANG Hui^{1,2}, LI Xuexian^{2,4}, JIANG Song^{1,2,3}

(1. School of Resources Engineering, Xi'an University of Architecture & Technology, Xi'an 710055, China; 2. Xi'an Key Laboratory for Intelligent Industrial Perception, Calculation and Decision, Xi'an 710055, China; 3. Xi'an Youmai Intelligent Mining Research Institute, Xi'an 710055, China; 4. School of Management, Xi'an University of Architecture & Technology, Xi'an 710055, China)

Abstract: In order to address the problems associated with maintaining population diversity and the insufficient number of equivalent Pareto-optimal solutions for multimodal multi-objective optimization, this study proposes a multimodal multi-objective optimization algorithm combining clustering and niche search (CSSMPIO). In the proposed algorithm, a clustering-based special crowding distance method is designed to initialize the population. Additionally, self-organized speciation is introduced to form stable niches, facilitating parallel searching and maintaining equivalent Pareto-optimal solutions. Subsequently, a non-dominated special crowding distance is introduced to realize individual selection and an elite learning strategy, circumventing premature convergence. The algorithm has been simulated using seven other state-of-the-art algorithms on 14 multimodal multi-objective optimization problems and was tested and analyzed using the Wilcoxon rank sum test. The results reveal that the general performance of CSSMPIO is superior to that of the compared algorithms. Finally, the CSSMPIO algorithm is applied to the map-based test problem, which confirms the effectiveness of the algorithm.

Keywords: multimodal multi-objective optimization; pigeon-inspired optimization algorithm; clustering strategy; niching searching; non-dominant sorting; elite learning strategy; diversity; map-based testing application

收稿日期: 2022-04-24. 网络出版日期: 2023-04-28.

基金项目: 国家自然科学基金项目 (52074205); 陕西省自然科学基金基础研究计划 (2020JC-44).

通信作者: 顾清华. E-mail: qinghuagu@126.com.

多模态多目标优化问题 (multimodal multi-objective optimization problem, MMOP)^[1] 是一类特殊的多目标优化问题 (mmulti-objective optimization

problem, MOP)^[2], 它着重于获取具有相同目标值的帕累托最优解集 (Pareto-optimal sets, PSs)。如何获取并保持决策空间更多的 PSs, 提高决策空间的多样性以及算法的搜索能力一直是多模态多目标优化问题存在的难点。

目前, 多模态多目标问题主要从两个方面来优化: 一是提高算法的搜索能力。最早有学者提出通过改进粒子群算法来提高搜索能力, 如 CLPSO (comprehensive learning particle swarm optimization)^[3] 和 AMPSO (modified particle swarm optimization)^[4] 通过优化粒子的学习方式以提高算法的搜索能力; STMOPSO (star-structured multi-objective particle swarm optimization)^[5] 基于星型拓扑结构提出一种非支配解集分布均匀程度的评价方法, 提高了算法的收敛速度。小生境策略可以提高算法的搜索能力, 如 Qu 等^[6] 提出的 LIPS (distance-based locally informed particle swarm) 利用邻域信息在小生境空间内粒子根据局部最优粒子进行搜索, 提高了算法的精细搜索能力; SS-MOPSO (self-organized speciation based multi-objective particle swarm optimizer)^[7] 基于物种形成策略形成多个稳定的物种, 物种与物种之间互不重叠, 分别搜索并保持 Pareto 最优解; Liang 等^[8] 在 NSGAI 的基础上提出 DN-NSGAI (decision space based niching NSGAI), 在决策空间小生境内分别搜索不同的 Pareto 最优解, 有效防止了过早收敛, 但该算法仅考虑决策空间的拥挤程度; DN-MMOES (two-stage double niched evolution strategy)^[9] 采用两阶段策略, 分别在决策和目标空间中均采用小生境。此外, 还设计了决策密度自适应策略来改善不平衡的决策空间密度, 该算法提高了搜索能力, 但该算法的某些局部最优解可能被全局最优解支配; MO-Ring-PSO-SCD (multi-objective particle swarm optimization algorithm with ring topology)^[10] 和 MO-Ring-CSO-SCD (multi-objective competitive swarm optimization algorithm with ring topology)^[11] 采用环形拓扑结构形成稳定的小生境, 且不需要定义小生境参数, 算法能够搜索到更多的 Pareto 最优解, 能在决策空间和目标空间中良好的分布; SMPSO-MM (multi-objective particle swarm optimizer with a self-organizing mechanism)^[12] 和 MMOPIO (multimodal multi-objective pigeon-inspired optimization)^[13] 都引入自组织映射机制 (self-organizing map, SOM), 将决策空间转化到目标空间, 使粒子在映射空间进行邻域搜索, 有效提高了算法的搜索能力, 但算法引入的 SOM 机制, 映射机理较复杂; MOMMOP (transformation technique based

on multi-objective optimization)^[14] 将 MMOP 转化为具有两个冲突目标的 MOP, 因此 MMOP 的所有最优解都转化成优化问题的 Pareto 最优解; Luo 等^[15] 提出的 MMOEA-CAS (evolutionary algorithm with clustering-based assisted selection strategy) 在决策空间采用添加算子增加多样性, 在目标空间采用删除算子删除最差解, 提高了算法的搜索能力; 近来, 有学者为了保留更多的全局和局部 Pareto 最优解, 提出 MMOMA-DM&CS (multimodal multi-objective memetic algorithm based on a local detection mechanism and a clustering-based selection strategy)^[16] 算法在局部检测机制中采用了基于密度的聚类方法, 帮助收集局部聚类中的解; MMOHEA (efficient two-archive model based multimodal evolutionary algorithm)^[17] 结合竞争粒子群和差分进化算法的策略并行生成子代解, 扩展了具有不同进化需求的决策空间和目标空间, 有效改善了算法搜索能力。

上述提高算法搜索能力的策略可以搜索到更多的 Pareto 最优解, 此外通过提高决策空间的多样性也能解决多模态多目标优化问题。如 TriMOEA-TA&R (multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies)^[18] 引入新的环境选择机制即特殊拥挤距离非支配排序法 (special crowding distance, SCD) 计算决策空间和目标空间的拥挤距离, 建立收敛性和多样性两个档案分别处理空间的变量关系, 最后还对档案中的解进行重组, 实验证明该算法同时保证了决策空间和目标空间的多样性, DN-NSGAI^[8]、MO_Ring_PSO_SCD^[10] 和 MO_Ring_CSO_SCD^[11] 算法也同样引入了 SCD 方法; Yang 等^[19] 提出 EMO-DD (evolutionary multi-objective optimization algorithm with a decomposition strategy in the decision space) 将整个种群分解为若干子种群对收敛程度和分布密度协同优化, 从而筛选出决策空间收敛性和多样性好的解; ZLS-SMPSO-MM (multimodal multi-objective particle swarm optimization combining zoning search and local search)^[20] 在 SOM 机制前对决策空间进行处理, 将决策空间分为多个子种群并用协方差矩阵自适应算法, 缩短了算法的计算时间; MMODE (multimodal multi-objective Differential Evolution optimization algorithm)^[21] 制定决策变量预选档案, 对边界解引入一个边界突变机制, 提高了决策空间的多样性; 此外, 基于聚类的方法也可以改善多样性, 如 MMO-CLRPSO (cluster based PSO with leader updating mechanism

and ring-topology)^[22]引用一种决策变量聚类方法在决策空间形成多个子种群,子种群同时考虑到全局最优粒子和局部最优粒子进行信息交流,在探索与开发阶段实现了较好的平衡;Lin等^[23]提出MMOEA/DC(novel MMOEA with dual clustering in decision and objective spaces)首先在决策空间和目标空间分别采用NCM(neighborhood-based clustering method)聚类方法和HCM(hierarchical clustering method)聚类方法,再选择收敛性好的解进行二次聚类,该算法同时在两空间实现了较好的性能;MMODE-CSCD(differential evolution algorithm based on the clustering technique and an elite selection mechanism)^[24]将差分进化算法与聚类策略结合,采用基于聚类的特殊拥挤距离计算决策空间与目标空间的综合拥挤程度,从而提高种群分布的均匀性;MOEA-GA(multimodal multi-objective evolutionary algorithm using a density-based one-by-one update strategy)^[25]采用谐波平均距离估计决策空间中解的平均密度,只允许粒子与决策空间邻域内的个体进行信息交互,并基于密度逐步更新粒子,实验证明算法提高了决策空间的多样性。

以上算法大多基于粒子群的搜索方式更新粒子的位置,而鸽群优化算法(pigeon-inspired optimization algorithm PIO)^[26]的局部搜索能力强,对求解多模态多目标优化问题有较好的性能。PIO自提出以来,有许多文献对其进行改进研究,例如,利用反向学习策略^[27]、混沌映射策略^[28]初始化种群,引入高斯因子^[29]、非线性变异策略^[30]等。PIO虽然得到了许多改进,但对于多模态多目标优化问题的研究仍不多见。基于此,为解决多模态多目标优化问题,提高种群多样性以及寻找更多的Pareto最优解,本文引入聚类、自适应物种形成策略结合改进的鸽群算法,提出一种融合聚类和小生境搜索的多模态多目标优化算法(multimodal multi-objective pigeon-inspired optimization algorithm with clustering and searching strategy, CSSMPIO)。

1 相关理论

1.1 多模态多目标优化

最小化多目标优化问题^[2]定义如下:

$$\begin{aligned} \min F(\mathbf{x}) &= \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ g_i(\mathbf{x}) &\leq 0, i = 1, 2, \dots, k \\ h_j(\mathbf{x}) &= 0, j = 1, 2, \dots, p \end{aligned} \quad (1)$$

式中: m 表示待优化的目标个数, $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ 是 n 维决策变量, $g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, k$ 为 k 个不等式约束, $h_j(\mathbf{x}) = 0, j = 1, 2, \dots, p$ 为 p 个等式约束。

在多目标优化问题中,根据Pareto解的支配关系可以比较不同解的优劣,若有两个可行解 $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{X}$,对 $\forall i = 1, 2, \dots, m$,都有 $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$,则称 \mathbf{x}_1 支配 \mathbf{x}_2 。如果一个解不受任何其他解的支配,那么它就被称为非支配解,决策空间中非支配解组成的集合称为非支配解集,即PSs, Pareto最优前端(Pareto-optimal front, PF)是目标空间中与Pareto最优解对应的所有向量的集合。

多模态多目标优化^[31]是一种特殊的多目标优化,如果一个多目标优化问题^[2]满足以下条件之一就属于多模态多目标优化问题:

- 1) 存在至少一个局部最优解;
- 2) 存在两个以上全局最优解。

同一问题最优目标值,可能对应着多个不同的解决方案,即PF中的某一点可能对应着决策空间中至少一个局部最优解或两个以上的全局最优解。路径规划问题^[22]就是典型的多模态多目标优化,如图1所示,出行者想要在最短的时间从起点到终点,方案1和方案6、方案2和方案4、方案3和方案5分别对应目标空间的同一点,各自的区别在于有无加油站以及交叉路口的个数,出行者可以根据不同的需求选择适合自己的出行路线,由此可见,仅得到一个Pareto最优解是不能满足出行者的需求的。多模态多目标问题的重点就是在保证目标空间PF优越性的同时,找到决策空间更多的PSs。

1.2 鸽群优化算法

鸽群算法^[26]由地图指南针算子和地标算子两阶段组成,这两个算子处于独立运行阶段,即先在地图指南针算子阶段对鸽子的位置和速度进行初始化并迭代更新,然后在地标算子阶段,每次迭代都会使鸽子数量减少一半,舍弃远离目标的鸽子,对于剩余中心位置的鸽子作为飞行参考方向。

地图指南针算子阶段(Map/compass operator):

$$v_i(t+1) = v_i(t) \cdot e^{(-R_s \cdot t)} + r_{\text{and}} \cdot (x_{\text{gbest}}(t) - x_i(t)), \quad (2)$$

$$i = 1, 2, \dots, N$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

式中: R_s 为地图指南针算子,通常取(0,1); t 为当前迭代次数; x_{gbest} 为全局最佳位置; N 为假设的鸽子数量; r_{and} 为介于(0,1)之间均匀分布的随机数。

地标算子阶段(LandMark operator):

$$N(t+1) = \frac{N(t)}{2} \quad (4)$$

$$x_{\text{center}}(t+1) = \frac{\sum x_i(t+1) \cdot \text{fitness}(x_i(t+1))}{N \cdot \sum \text{fitness}(x_i(t+1))} \quad (5)$$

$$x_i(t+1) = x_i(t) + r_{\text{and}} \cdot (x_{\text{center}}(t+1) - x_i(t)) \quad (6)$$

其中 $\text{fitness}(x)$ 为每只鸽子的适应度值。

2 融合聚类和小生境搜索的优化算法

本文提出融合聚类和小生境搜索的多模态多目标优化算法, 称作 CSSMPIO 算法。主要由基于聚类的特殊拥挤距离排序和自适应物种形成多个子种群两部分组成。

2.1 聚类排序初始化种群

聚类算法将同一非支配层的解划分为多个类, 同一类中的个体将来自同一局部区域。将聚类算法与特殊拥挤距离非支配排序 (special crowding distance, SCD) 方法^[10]结合, 形成基于聚类的特殊拥挤距离排序 (cluster special crowding distance, CSCD)^[24]方法, 可以有效地解决 SCD 不能正确识别邻域解的问题。在 CSCD 中, 个体的邻域解只能从同一类中识别出来。如图 2(a), 把解分成 3 类, 第 1 类(1, 2, 3, 4, 5)第 2 类(6, 7, 8), 第 3 类(9)。

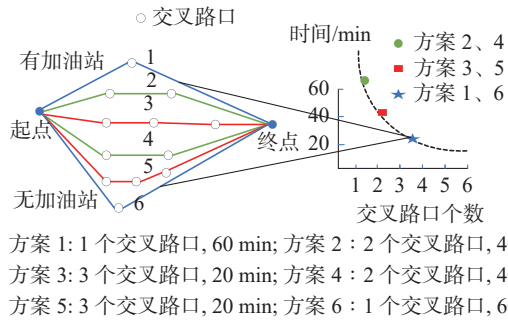


图 1 路径优化问题

Fig. 1 Path planning problem

以解 4 为例, 则基于聚类的特殊拥挤距离值 $C_{SCD(4)}$ 的计算公式如下:

$$C_{D(4,x)} = \frac{|x_{3,1} - x_{5,1}|}{|x_{1,1} - x_{5,1}|} + \frac{|x_{3,2} - x_{5,2}|}{|x_{1,2} - x_{5,2}|} \quad (7)$$

$$C_{D(4,y)} = \frac{|x_{3,1} - x_{5,1}|}{|x_{1,1} - x_{9,1}|} + \frac{|x_{3,2} - x_{5,2}|}{|x_{1,2} - x_{9,2}|} \quad (8)$$

$$\begin{cases} C_{SCD(4)} = \max(C_{D(4,x)}, C_{D(4,y)}), \\ C_{D(4,x)} > C_{D(avg,x)} \text{ 或 } C_{D(4,y)} > C_{D(avg,f)} \\ C_{SCD(4)} = \min(C_{D(4,x)}, C_{D(4,y)}), \\ C_{D(4,x)} \leq C_{D(avg,x)} \text{ 或 } C_{D(4,y)} \leq C_{D(avg,f)} \end{cases} \quad (9)$$

式中: $C_{D(4,x)}$ 为解 4 在决策空间的拥挤距离, $C_{D(4,y)}$ 为解 4 在目标空间的拥挤距离, $C_{D(avg,x)}$ 和 $C_{D(avg,y)}$ 分别为决策空间和目标空间的平均拥挤距离。

基于聚类的特殊拥挤距离初始化种群步骤如下:

输入 种群 P

1) 种群 P 进行非支配排序, 得到多个非支配层次, 定义为 $F_1, F_2, \dots, F_{n_{um}}$ (n_{um} 为层数), $j = 1, 2, \dots, n_{um}$ 表示个体在 F_j 中的非支配排序;

2) 计算 k_j 值: $k_j = |F_j|/n$, $|F_j|$ 表示 F_j 中的个体数, n 为正整数;

3) 根据 k-means 算法将 F_j 分成多个聚类;

4) 利用同一类内的邻域关系计算每个个体的 C_{SCD} 值。

输出 根据种群的非支配排序和 C_{SCD} 值对种群 P 进行排序, 输出一个新的种群 P^* 。

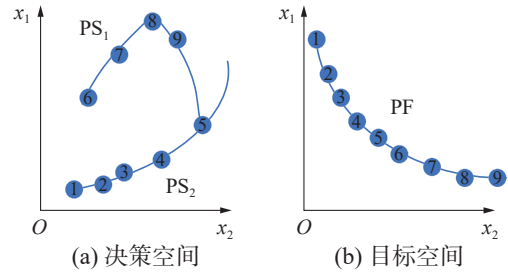


图 2 多模态多目标优化问题

Fig. 2 Multimodal multi-objective optimization problem

2.2 小生境搜索过程

种群经过基于聚类的特殊拥挤距离排序初始化后, 被分成了多个类, 然后采用自适应物种形成策略生成稳定的小生境, 物种形成是根据个体之间相似性将个体划分为多个子种群。如把 S_1, S_2, \dots, S_k 种群划分为多个子种群, 个体 $x^* \in S_i$, 若对于个体 $y \in S_i$, 都有个体 x^* 的非支配等级高于个体 y , 且个体 x^* 到个体 y 的欧氏距离 d_{is} 小于物种半径 R , 则个体 y 向 x^* 学习:

$$d_{is}(x^*, y) = \sqrt{((x^* - y)^2)} \leq R \quad (10)$$

该定义并不意味着如果一个物种 S_i 以非支配个体 x^* 为中心, 则 x^* 的物种半径范围内的所有个体都属于该物种。如图 3 所示, 一个占优势的个体可能会支配多个物种内的个体。

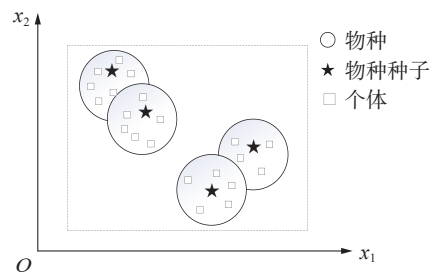


图 3 物种在二维区域的分布

Fig. 3 Distribution of species in a two-dimensional region

自适应物种形成中, 物种种子经过特殊拥挤距离排序选择后, 直接根据物种半径确定其物种内所包含的个体。若个体同时在两个物种种子的物种半径内, 则由非支配等级更高的物种种子支配该个体。其主要步骤如下:

1) 种群进行基于聚类的非支配特殊拥挤距离升序排序, 输出新种群, 存储到 spop 中;

2) 将 NP_{spop} 作为 spop 中集合的数目, spop 中的第一个个体记为物种种子 $x_{\text{species seed}(j)}$;

3) 获得物种种子 $x_{\text{species seed}(j)}$ 后, 迭代更新形成互不重叠的物种;

4) 当 $NP_{\text{spop}} \neq 0$ 时, 执行以下操作:

计算 $x_{\text{species seed}(j)}$ 与 $s_{\text{pop}(j)}$ 的欧氏距离; 若 $d_{is} \leq R$, 则个体 i 位于该物种; 同时清除 spop 中已经分配给第 j 个物种种子的个体, spop 得到更新。

输出 更新后的多个子种群。

2.3 改进的位置更新策略

为平衡个体的探索与开发过程, 受多目标鸽群启发算法 (multi-objective pigeon-inspired optimization algorithm, MPIO)^[32] 的影响, 结合粒子群优化算法, 将地图指南针算子和地标算子结合起来, 个体在每个子种群的位置更新如图 4。

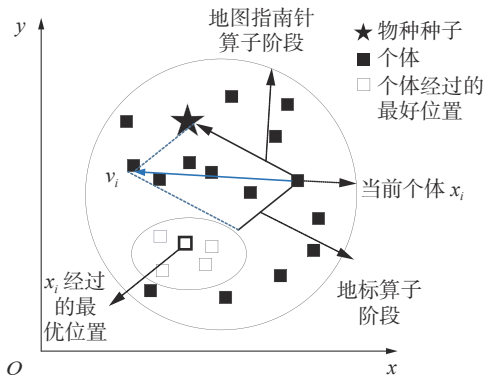


图 4 单个物种中个体位置更新方式

Fig. 4 Individual position updating in a single species

个体的速度更新公式如下:

$$v_i(t+1) = v_i(t) \cdot i_{wt} + \left(1 - \log\left(\frac{t}{T}\right)\right) \cdot r_{\text{and}} \cdot (x_{\text{species seed},i}(t) - x_i(t)) + \log\left(\frac{t}{T}\right) \cdot r_{\text{and}} \cdot (x_{\text{center}}(t) - x_i(t)) \quad (11)$$

$$i_{wt} = w_{\text{max}} + (w_{\text{min}} - w_{\text{max}}) \cdot \frac{T-t}{T} + w_{\text{min}} \quad (12)$$

式中: w_{max} 与 w_{min} 分别表示惯性权重的最大最小值; $x_{\text{species seed}}$ 为物种种子; r_{and} 为 0~1 之间的随机数; x_{center} 表示鸽群中心坐标位置; T 为最大迭代次数; i_{wt} 为非线性惯性权重。

为了保留更多的解, 对最优解执行精英学习策略, 避免早熟收敛, 公式如下:

$$x_i(t+1) = x_{\text{species seed},i}(t) + (u_{b(i)} - l_{b(i)}) \cdot \text{Guass}(0, p_r^2) \quad (13)$$

$$p_r = p_r0 - \frac{(p_r0(1) - p_r0(2)) \cdot 2}{T} \quad (14)$$

式中: u_b 、 l_b 分别为变量的上下界; $\text{Guass}(0, p_r^2)$ 为均值为 0, 标准差为 p_r 的高斯分布, p_r0 值从 0.2 线性

下降到 0.05。进化初期, 较大的变异范围有利于算法的全局搜索, 随着迭代的进行, 算法需要在局部范围找到更精确的解, 因此小的变异范围更适合。

2.4 CSSMPIO 算法流程

输入 种群规模 P_{opsize} ; 最大评估次数 M_{axGen} ; 鸽群参数: w_{max} 、 w_{min} ; 变异算子 pr0 ; 外部存档集 P_{achive} ;

1) 初始化个体的速度和位置, 记为 POA , 并评估 POA ;

2) 种群 $P = \text{POA}$;

3) 当迭代 $t \leq M_{\text{axGen}}$ 时, 执行以下操作:

种群 P 执行 2.2 节操作, 输出新种群 P^* ; 对新种群 P^* 执行步骤 2.3 节操作;

4) 更新个体的位置和速度, 步骤如下:

选出 p_{best} 和 $x_{\text{species seed}(j)}$; p_{best} 为种群 P 中的最优个体, $x_{\text{species seed}(j)}$ 物种中的最优个体; 对 p_{best} 进行非支配排序, 依据式 (5) 将其转换成 x_{center} ; 依据式 (11)、(3) 更新个体的速度和位置, 式 (13) 对子代执行精英学习策略, 更新 P^* ; 对更新后的 P^* 进行评估; 将 P^* 放入 POA 中, 对 POA 中的个体进行排序;

5) 当前迭代次数满足 $t > M_{\text{axGen}}$ 时, 结束循环, 否则返回 3)。

输出 PS , PF 。

在 CSSMPIO 中, 聚类排序首先对粒子进行初始化, 然后采用自适应物种形成策略在决策空间生成稳定的小生境子种群, 鸽群算法的局部搜索能力强, 因此可以增强子种群的局部搜索能力, 在决策空间中搜索到大量的 PSs, 最后再引入精英学习策略, 对子种群中的最优解进行变异, 避免早熟收敛。

2.5 CSSMPIO 时间复杂度分析

CSSMPIO 算法的主要时间成本来源于基于聚类的特殊拥挤距离排序初始化种群和自适应物种形成策略两部分组成。设种群规模为 N , 目标维数为 M 。基于聚类的特殊拥挤距离排序初始化种群时间复杂度为 $O(N)$ 。自适应物种形成策略中假设有 N 个个体被排序并存储在 spop 中, 其大小为 NP_{spop} , for 循环 NP_{spop} 次, 以查看该个体是否在当前种子的半径 R 内。最好的情况是, 所有的个体都在第一个种子的物种半径中, 因此 for 循环只执行了 N 次。在最坏的情况下, 当所有 N 个个体都是 N 个独立物种的种子时, for 循环执行 $N(N+1)/2$ 次。表明该过程的复杂度最高为 $O(N^2)$ 。并且自适应物种形成过程中, 已分配给某些物种的个体被从种群中移除, 种群中剩余的个

体不再计算与之前物种种子的欧氏距离。因此,其复杂度仅依赖于 N , 计算次数在物种形成过程中迅速减少。因此, CSSMPIO 最坏的时间复杂度为 $O(MN^3)$ 。

3 实验结果与分析

3.1 评价指标

多模态多目标优化问题既要求目标空间的分布性能良好, 又要保证找到决策空间更多的等价 PSs, 因此本文采用了以下评价指标对所提算法进行评价。

1) PSP (Pareto set proximity)^[31]:

$$P_{PS} = \frac{R_c}{x_{IGD}} \quad (15)$$

式中: R_c (cover rate, CR) 表示算法求得的 PSs 对真实 PSs 的覆盖率; x_{IGD} 为求得的 PSs 与真实 PSs 之间的欧氏距离 IGDx, 因此 PSP 反映的是真实 PSs 与获得的 PSs 之间的相似性, PSP 越大算法性能越好。

2) IGDx^[31]:

$$x_{IGD}(O, P^*) = \frac{\sum_{v \in P^*} d_{(v, O)}}{|P^*|} \quad (16)$$

式中: $|P^*|$ 表示参考点的数目, O 表示一组已得到的解, $d_{(v, O)}$ 表示 v 和 O 中所有点之间最小的欧氏距离。如果参考解集能很好地表示真实 PSs, 那么 IGDx 就能较好地衡量决策空间的收敛性和多样性。IGDx 的值越小, 说明算法求得的 PSs 和实际 PSs 越接近。

3.2 实验设置

为进一步验证所提算法的性能, 将 CSSMPIO 算法与自适应物种形成粒子群优化算法 SS_MOPSO^[7]、基于决策空间小生境算法 DN_NSGAII^[8]、基于环形拓扑结构的多模态多目标粒子群优化算法 MO-Ring-PSO-SCD(MRPS)^[10]、自组织多模态多目标粒子群优化算法 SMOPSO-MM^[12]、自组织多模态多目标鸽群优化算法 MMOPIO^[13]、基于双存档模型的高效混合进化多模态多目标优化算法 MMOHEA^[17] 算法和多目标优化算法 MOEAD^[33] 进行对比实验研究。所有算法的最大评估次数为 80 000 次, 种群规模为 800, 算法在每个测试函数上运行 20 次, 算法参数设置如表 1。

选取 14 个多模态多目标问题^[31] 进行测试。其中, 8 个 MMF 函数^[10]、3 个 SYM-PART 函数^[34]、3 个 OMNI 函数^[35]。所有优化问题的目标个数均为 2, MMF 函数与 SYM-PART 函数的空间维度为 2, OMNI 函数的空间维度分别为 3、4、5。一般

来说, 在每个维度上 PSs 重叠的测试函数比没有重叠的测试函数更复杂, 此外, 当一个函数的 PSs 数目较多时, 它也会变得更加复杂。MMF₁ 和 MMF₂ 是相对简单的测试函数, MMF₃ 比 MMF₁ 和 MMF₂ 复杂。MMF₄ 和 MMF₈ 与表中其他测试函数相比, PF 是凹的, 它们有 4 个 PSs, 每个维度都不重叠。MMF₆ 有 4 个 PSs, 它们在每个维度上重叠, MMF₇ 的 PSs 呈不规则曲线。SYM-PART₂ 测试函数是由 SYM-PART₁ 旋转生成的, 因此比 SYM-PART₁ 更复杂。在 SYM-PART 函数中, PSs 的数量都是 9 个。Omni-test 中 PSs 的数量最多, 分别为 27、72、360, 而且每个维数都有重叠。因此, Omni-test 是 14 个测试函数中最复杂的。测试系数特征如表 2。

表 1 算法的参数设置

Table 1 Parameters setting of algorithms

算法	参数
CSSMPIO	$R = 1/120V_R; k = 10; c_1 = c_2 = 2.05$
MMOHEA ^[17]	$c_r = 0.9; F = 0.5$
MRPS ^[10]	$c_1 = c_2 = 2.05; w = 0.729\ 8$
MMOPIO ^[13]	$\lambda = 7; R_s = 0.3$ $c_1 = c_2 = 2.05; w = 0.729\ 8$
SMOPSO-MM ^[12]	拓扑结构=1×100; $\eta_0 = 0.7$; $\sigma_0 = \frac{\sqrt{1^2 + 100^2}}{2}$
SS_MOPSO ^[7]	$R = 1/120V_R; w = 0.729\ 8; c_1 = c_2 = 2.05$
DN_NSGAII ^[8]	$C_F = 0.5; \eta_c = \eta_m = 20; p_c = 1; p_m = 1/n$
MOEAD ^[33]	$p_m = 0.5; \eta_m = 20$

表 2 测试函数特征

Table 2 The characteristics of test problems

函数名称	PSs 的数量	每个维度的重叠
MMF ₁	2	×
MMF ₂	2	×
MMF ₃	2	√
MMF ₄	4	×
MMF ₅	4	×
MMF ₆	4	√
MMF ₇	2	×
MMF ₈	4	×
SYM-PART	9	√
Omni-test ₁	27	√
Omni-test ₂	72	√
Omni-test ₃	360	√

3.2.1 种群规模 popsize 分析

在本节中, 分析种群规模 popsize 对算法性能

PSP 的影响, 由于 MMF_1 、 MMF_3 、 MMF_4 和 MMF_6 这 4 个测试函数的 PSs 的数量以及维度重叠是不同的, 因此对比算法性能在该 4 个测试函数上随种群规模的变化情况, 并将种群规模设置为 200、400、600、800、1000, 其他参数设置如表 1。

实验结果如图 5 所示, 可以看出, MOEAD 作为多目标优化算法, 种群规模的改变对算法性能没有多大的影响。虽然 DN-NSGAI 的 PSP 值较低, 但算法性能随种群规模的增大而增大。MMOHEA 仅在 MMF_3 上算法性能随种群规模有所改变, 这

是因为 MMF_3 的测试函数复杂, 且 PSs 的曲线更加平滑。 MMF_3 测试函数中, 除 SS-MOPSO 算法, 算法性能都随种群规模的增大而增大, 在种群规模小于 600 时, SS-MOPSO 性能最好, 但当种群规模高于 600 时, 算法性能几乎不变。除以上算法, 其他算法性能都随种群规模的增大而变化, 由于 MMOPIO 和 SMPSO-MM 采用 SOM 机制来获得更好的解集, 所以在种群规模小于 800 时, MMOPIO 和 SMPSO-MM 性能优于 CSSMPPIO, MRPS 算法性能变化明显, 且一直低于 CSSMPPIO。

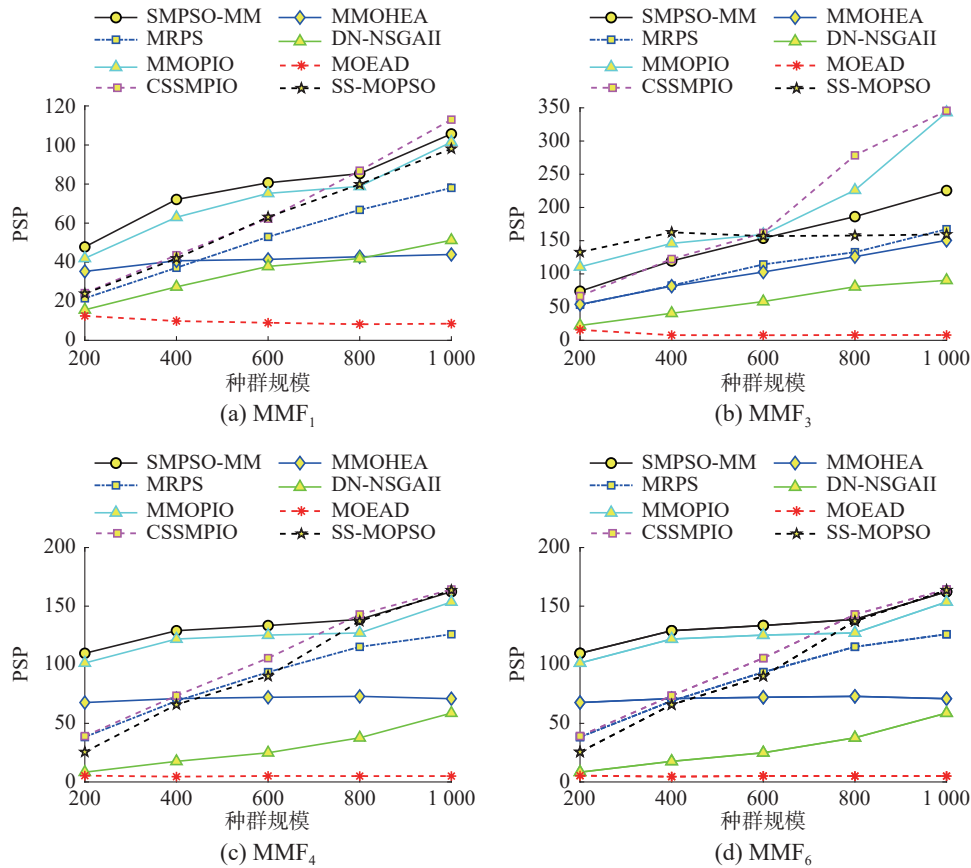


图 5 不同种群规模的 PSP 值

Fig. 5 PSP values with different population sizes

综合来看 CSSMPPIO 的性能随种群规模的变化更优于对比算法, 并且所有算法性能在不同的测试函数上都受种群规模的影响, 在计算资源有限的情况下, 所提算法的种群规模仍设置为 800。

3.2.2 物种半径 R 分析

自适应物种形成策略生成小生境子空间, 其中物种半径确定子种群的大小, 物种半径的大小对算法的性能有一定的影响, 一般情况下, 物种半径设置为决策变量范围的 $1/20 \sim 1/10$, 因此本小节验证在不同测试函数上算法性能随物种半径变化情况。本实验选定 MMF_2 、 MMF_5 、SYM-PART₁ 和 Omni-test₁ 测试函数分析不同物种半径对 CSS-

MPIO 性能的影响, 10 次实验的平均 PSP 值如图 6, 其他参数设置同表 1。图中显示, 测试函数的 PSP 值都随物种半径的增大而减小, 这是因为随着物种半径的增大, 决策空间中形成的物种数量减少。此外, 与 SYM-PART₁ 和 Omni-test₁ 相比, 随着半径的增加, MMF_2 、SYM-PART₁ 和 Omni-test₁ 的 PSP 值显示出较小的变化。产生这种现象的原因是测试函数决策变量变化范围小, 因此半径的变化很小, 这对算法的性能影响很小。因此将物种半径设置为决策变量范围的 $1/20$ 时, CSSMPPIO 可以找到决策空间更多 Pareto 解集, 提高决策空间的种群多样性。

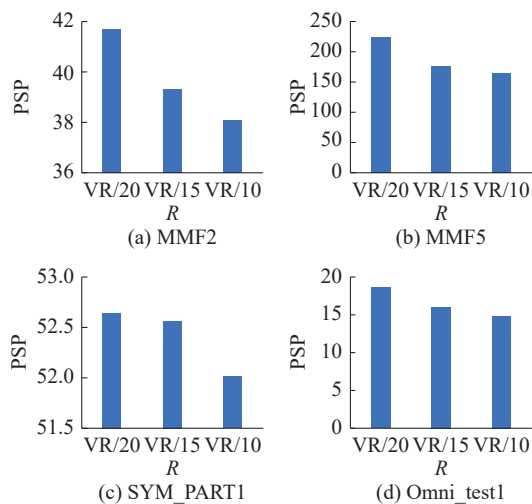


图 6 不同物种半径的 PSP 值

Fig. 6 PSP values with different radius

表 3 CSSMPIO 使用不同参数 n 获得的 PSP 值Table 3 PSP values obtained by CSSMPIO with different n values

测试问题	$n = 5$	$n = 10$	$n = 15$	$n = 20$
MMF ₁	79.97±2.39	83.19±2.24	82.68±2.87	77.75±3.29
MMF ₂	274.14±63.78	275.76±37.53	282.44±51.37	268.65±57.27
MMF ₃	298.12±84.37	305.08±29.13	272.72±36.95	273.78±43.89
MMF ₄	128.64±6.05	132.78±7.88	116.78±10.28	110.12±11.95
MMF ₅	39.22±1.35	43.11±1.35	40.99±0.83	34.49±1.11
MMF ₆	38.77±1.85	41.60±1.77	40.16±1.73	39.80±1.09
MMF ₇	110.29±7.81	113.60±5.99	102.62±8.65	110.38±6.87
MMF ₈	60.59±2.45	57.55±3.21	53.24±5.22	57.62±2.32
SYM-PART ₁	42.71±3.73	52.01±3.58	42.85±1.97	42.29±3.53
Omni-test ₁	10.39±1.24	11.73±1.25	5.64±2.73	8.10±2.60

3.3 实验结果与对比

3.3.1 PSP 指标对比实验和分析

CSSMPIO 与其他比较算法所得的 PSP 值的平均值与标准方差的统计结果如表 4 所示, 其中最佳结果被突出显示。同时, 采用 Wilcoxon 秩和检验^[36]表明 CSSMPIO 的结果与其他比较算法在 0.05 显著水平上的差异性, “+”“-”和“=”分别表示比较算法与 CSSMPIO 相比体现出明显优越、明显差、没有显著差异的性能。从表 4 可以看出, CSSMPIO 取得的 PSP 最优值数量为 8 个, SMP SO_MM 取得的最优值数量为 3 个, SS_MOPSO 求得的最优值数量为 3 个。CSSMPIO 在 14 个测试函数上的 PSP 结果显著优于 MOEAD 和 DN_NSGAII。MOEAD 是典型的多目标算法, 对于求解多模态多目标测试函数的效果最

3.2.3 聚类数目 k 分析

在所提算法中, 初始化种群中聚类数目 k 值的确定也会影响算法的性能。如 2.1 小节所示, $k = |F_j|/n$, $|F_j|$ 表示 F_j 中的个体数, n 为正整数, 可以看出, 参数 n 间接决定了 k 的值, 当 n 较大时, k 越小。本小节验证不同 n 值^[24] 对算法性能的影响, 参数设置如表 1, 实验结果如表 3。从表 3 可以看出, 大多数测试函数在 $n=10$ 时, 算法性能最好, 尤其是表现在复杂函数中。即使 $n=10$ 时 MMF₂ 和 MMF₈ 函数的 PSP 值不是最高的, 但与表现最好的也相差不大。由于一个解和它的邻域解应来自同一局部区域, 所以参数 n 越小越好, 但如果同一类中解的个数过少, 则会影响边界解的 CSCD 值。综合考虑, 取参数 $n=10$ 。

差。DN_NSGAII 在所有函数上的性能表现都较差, 这是因为 DN_NSGAII 只考虑了决策空间的拥挤距离, 而忽略了目标空间。MMOHEA 算法在 Omni-test 测试函数上的性能较好, 虽然在其他测试函数上 PSP 性能较差, 但是算法运行所需的时间较短, 并且算法性能比较稳定。而 MRPS 性能略低的原因在于仅考虑了环形拓扑邻域结构, 没有进行非支配特殊拥挤距离排序, 在决策空间的解分布不够均匀。SMP SO_MM 和 MMOPIO 采用 SOM 机制来建立粒子的邻域, 使其更适合求解较复杂的问题。在 SS_MOPSO 中, 不同物种重叠粒子的分配仅取决于物种种子的非支配等级, 粒子会向非支配等级更高的物种种子学习, 这会影响种群的多样性, 得到的 PSs 是不完整、不均匀的。

表 4 所有算法得到的 PSP 平均值与标准方差
Table 4 Mean and standard variance of PSP obtained by all algorithms

测试问题	mean±std							
	DN_NSGAII	MOEAD	MRPS	SMPSO_MM	MMOPIO	SS_MOPSO	MMOHEA	CSSMPIO
MMF ₁	41.88±2.80 (-)	8.02±4.89 (-)	67.05±3.12 (-)	85.34±2.34 (=)	73.83±2.78 (-)	83.11±3.27 (-)	42.78±1.43 (-)	86.86±4.15
MMF ₂	61.55±33.31 (-)	4.62±3.24 (-)	107.53±11.92 (-)	152.69±15.35 (-)	167.78±45.24 (-)	163.28±18.06 (-)	116.75±16.14 (-)	277.15±34.30
MMF ₃	75.11±28.88 (-)	7.00±3.79 (-)	128.63±17.06 (-)	186.24±15.19 (-)	226.06±32.79 (-)	189.94±12.67 (-)	125.96±16.67 (-)	305.44±24.93
MMF ₄	38.45±8.59 (-)	1.93±0.97 (-)	114.51±3.29 (-)	132.78±3.82 (-)	120.15±3.91 (-)	139.40±3.49 (+)	73.03±3.99 (-)	135.90±5.95
MMF ₅	14.57±1.47 (-)	3.35±1.35 (-)	33.11±1.35 (-)	40.12±1.33 (-)	30.24±1.46 (-)	40.11±1.19 (-)	22.21±0.41 (-)	43.62±1.50
MMF ₆	18.89±1.37 (-)	4.97±2.37 (-)	36.44±1.59 (-)	42.42±1.08 (=)	36.23±1.99 (-)	42.19±1.29 (=)	24.16±0.59 (-)	41.74±1.39
MMF ₇	95.55±17.79 (-)	7.49±4.78 (-)	108.77±3.89 (-)	141.13±5.14 (+)	136.49±4.78 (+)	123.73±6.27 (+)	78.05±2.87 (-)	119.82±6.11
MMF ₈	18.01±5.89 (-)	0.25±0.12 (-)	47.64±2.22 (-)	62.53±2.34 (+)	54.78±7.12 (-)	59.34±22.39 (=)	33.78±3.40 (-)	58.38±2.76
SYM-PART ₁	0.44±0.59 (-)	0.01±0.01 (-)	21.95±1.67 (-)	30.37±2.77 (-)	59.97±0.24 (+)	34.50±1.30 (-)	31.22±1.48 (-)	52.01±3.58
SYM-PART ₂	3.07±4.00 (-)	0.01±0.01 (-)	18.73±0.94 (-)	16.88±1.33 (-)	43.18±3.16 (-)	30.77±1.19 (-)	31.82±0.70 (-)	47.08±3.79
SYM-PART ₃	5.57±9.97 (-)	0.01±0.01 (-)	9.37±2.19 (-)	15.78±2.80 (-)	41.15±1.78 (-)	28.17±2.45 (-)	11.34±2.34 (-)	56.94±2.78
Omni-test ₁	1.16±0.23 (-)	0.03±0.01 (-)	8.04±1.06 (-)	7.69±1.48 (-)	1.15±0.08 (-)	11.75±1.10 (-)	12.09±0.54 (=)	12.15±2.23
Omni-test ₂	0.44±0.08 (-)	0.03±0.01 (-)	0.98±0.07 (=)	0.95±0.07 (=)	0.95±0.04 (-)	1.46±0.05 (+)	1.01±0.00 (=)	0.99±0.07
Omni-test ₃	0.28±0.09 (-)	0.02±0.01 (-)	0.51±0.02 (+)	0.49±0.02 (+)	0.50±0.01 (+)	0.58±0.02 (+)	0.47±0.01 (=)	0.46±0.02
+/-/-	0/0/14	0/0/14	1/1/12	3/3/8	3/0/11	4/2/8	0/3/11	—

此外, CSSMPIO 对 MMF₂、MMF₃、MMF₄、SYM-PART 与 Omni-test₁ 测试函数的性能显著优越, 尤其是对 SYM-PART 和 MMF₈ 测试函数, C_R 值达到 1.0, 这意味着求得的 PSs 全部覆盖了真实 PSs。CSSMPIO 与 SMPSO_MM 在 MMF₁、MMF₅ 和 MMF₆ 上的结果相近, 这是因为算法在决策空间表现相似, 而 CSSMPIO 在 MMF₇ 表现较差的原因是 MMF₇ 的 PSs 的分布呈不规则曲线。由于鸽群算法的局部搜索能力强, CSSMPIO 与 MMOPIO 都适于求解 SYM-PART 测试函数, 并且对于多个 PSs 的 Omni-test 测试函数 CSSMPIO 也表现较好。

表 5 反映了 CSSMPIO 与对比算法的 HV 值, HV 反映了目标空间的收敛性和多样性^[31]。从表 5 可以看出, CSSMPIO 在 MMF₅、SYM-PART₂ 和 SYM-PART₃ 测试函数上的 HV 最高; DN_NSGAII

和 MOEAD 的 HV 大都表现较好, 原因是这两个算法都是多目标算法, 考虑了目标空间而未考虑决策空间。虽然 CSSMPIO 与其他算法相比没有表现出最佳 HV, 但从统计结果来看, 结果相差不大。综合来讲 CSSMPIO 不仅在决策空间有较好的收敛性和多样性, 而且也考虑到在目标空间的分布。

3.3.2 IGDx 指标对比实验和分析

CSSMPIO 与其他比较算法所得的 IGDx 值的平均值和标准方差见表 6。同时, Wilcoxon 得到的统计分析结果也在表 6 中显示。从表中可以看出, 相比于 MOEAD 和 DN_NSGAII, CSSMPIO 在 14 个测试函数上都表现出了较好的性能。CSSMPIO 在 MMF₁、MMF₅ 和 MMF₆ 测试函数上的 IGDx 值与 SMPSO_MM 算法相比没有显著差异。CSSMPIO 在规则曲线测试函数 MMF₂ 和 MMF₃ 上的 IGDx 值较小, 说明在这两个测试函数上获得的

PSs 与真实 PSs 值的欧氏距离较小, 算法求得的解集和参考解集越接近。在 MMF_4 上, SMPSO_MM 取得了较好的效果, 由于 MMF_7 和 MMF_8 的解分布为不规则的曲线, 而 SOM 映射网络适于求解此类问题, 所以 SMPSO_MM 和 MMOPIO 求得的解集和参考解集更接近。在更复杂、空间维度更

高的测试函数 SYM-PART 与 Omni-test 上的性能指标表示, CSSMPIO 却取得了更好的结果, 这意味着更接近真正的 PSs。并且 MMF_1 、 MMF_4 、 MMF_5 、 MMF_7 和 MMF_8 是除 MMF_6 之外在每个维度上没有重叠的简单测试函数, 从结果中可以看出, CSSMPIO 对求解复杂的测试函数效果更优越一些。

表 5 所有算法得到的 HV 平均值与标准方差

Table 5 The mean and standard variance of HV obtained by all algorithms

测试问题	mean±std							
	DN_NSGAII	MOEAD	MRPS	SMPSO_MM	MMOPIO	SS_MOPSO	MMOHEA	CSSMPIO
MMF_1	$3.66\pm 1.30\times 10^{-3}$	$3.67\pm 4.79\times 10^{-4}$	$3.66\pm 4.63\times 10^{-4}$	$3.67\pm 3.99\times 10^{-4}$	$3.66\pm 2.84\times 10^{-4}$	$3.66\pm 4.55\times 10^{-4}$	$3.66\pm 5.40\times 10^{-4}$	$3.66\pm 1.31\times 10^{-4}$
MMF_2	$3.66\pm 6.20\times 10^{-3}$	$3.67\pm 5.34\times 10^{-4}$	$3.65\pm 8.10\times 10^{-3}$	$3.66\pm 4.16\times 10^{-3}$	$3.65\pm 2.07\times 10^{-3}$	$3.66\pm 4.20\times 10^{-3}$	$3.65\pm 4.80\times 10^{-3}$	$3.66\pm 4.02\times 10^{-4}$
MMF_3	$3.66\pm 4.90\times 10^{-3}$	$3.67\pm 7.72\times 10^{-4}$	$3.65\pm 5.10\times 10^{-3}$	$3.66\pm 3.34\times 10^{-3}$	$3.66\pm 1.41\times 10^{-3}$	$3.66\pm 3.28\times 10^{-3}$	$3.66\pm 2.25\times 10^{-3}$	$3.66\pm 1.96\times 10^{-3}$
MMF_4	$3.33\pm 3.60\times 10^{-4}$	$3.33\pm 1.63\times 10^{-4}$	$3.33\pm 1.20\times 10^{-3}$	$3.33\pm 8.88\times 10^{-4}$	$3.33\pm 3.88\times 10^{-4}$	$3.33\pm 8.45\times 10^{-4}$	$3.33\pm 1.34\times 10^{-4}$	$3.33\pm 3.14\times 10^{-4}$
MMF_5	$3.67\pm 2.70\times 10^{-3}$	$3.67\pm 6.81\times 10^{-4}$	$3.66\pm 3.52\times 10^{-4}$	$3.66\pm 4.89\times 10^{-4}$	$3.66\pm 6.98\times 10^{-5}$	$3.67\pm 3.03\times 10^{-4}$	$3.66\pm 3.72\times 10^{-4}$	$3.67\pm 3.10\times 10^{-4}$
MMF_6	$3.66\pm 1.10\times 10^{-3}$	$3.67\pm 5.53\times 10^{-4}$	$3.66\pm 3.31\times 10^{-4}$	$3.66\pm 3.74\times 10^{-4}$	$3.66\pm 3.19\times 10^{-4}$	$3.66\pm 3.19\times 10^{-4}$	$3.66\pm 3.24\times 10^{-4}$	$3.66\pm 3.24\times 10^{-4}$
MMF_7	$3.66\pm 7.10\times 10^{-4}$	$3.67\pm 2.39\times 10^{-4}$	$3.67\pm 2.38\times 10^{-4}$	$3.66\pm 3.00\times 10^{-4}$	$3.66\pm 6.24\times 10^{-4}$	$3.66\pm 2.88\times 10^{-4}$	$3.66\pm 3.61\times 10^{-4}$	$3.66\pm 4.16\times 10^{-4}$
MMF_8	$3.21\pm 1.30\times 10^{-3}$	$3.21\pm 1.44\times 10^{-4}$	$3.21\pm 1.50\times 10^{-4}$	$3.21\pm 1.58\times 10^{-4}$	$3.21\pm 4.14\times 10^{-5}$	$3.21\pm 7.30\times 10^{-4}$	$3.21\pm 3.99\times 10^{-4}$	$3.21\pm 7.30\times 10^{-4}$
$SYM-PART_1$	$1.32\pm 3.09\times 10^{-4}$	$1.32\pm 8.35\times 10^{-5}$	$1.30\pm 1.90\times 10^{-4}$	$1.27\pm 4.86\times 10^{-3}$	$1.32\pm 8.20\times 10^{-5}$	$1.32\pm 8.30\times 10^{-4}$	$1.31\pm 7.72\times 10^{-4}$	$1.32\pm 2.29\times 10^{-4}$
$SYM-PART_2$	$1.32\pm 4.92\times 10^{-4}$	$1.32\pm 6.85\times 10^{-4}$	$1.29\pm 2.20\times 10^{-3}$	$1.27\pm 4.39\times 10^{-3}$	$1.32\pm 5.16\times 10^{-4}$	$1.31\pm 1.12\times 10^{-3}$	$1.31\pm 1.02\times 10^{-3}$	$1.32\pm 4.26\times 10^{-4}$
$SYM-PART_3$	$1.32\pm 5.00\times 10^{-4}$	$1.32\pm 1.40\times 10^{-3}$	$1.29\pm 2.50\times 10^{-3}$	$1.25\pm 6.48\times 10^{-3}$	$1.31\pm 2.04\times 10^{-3}$	$1.31\pm 2.04\times 10^{-3}$	$1.30\pm 1.88\times 10^{-3}$	$1.26\pm 6.60\times 10^{-3}$
Omni-test ₁	$62.06\pm 3.98\times 10^{-4}$	$62.06\pm 4.80\times 10^{-3}$	$61.97\pm 1.24\times 10^{-2}$	$61.95\pm 1.56\times 10^{-2}$	$61.99\pm 5.33\times 10^{-3}$	$62.02\pm 5.33\times 10^{-3}$	$61.01\pm 1.81\times 10^{-2}$	$62.06\pm 2.16\times 10^{-4}$
Omni-test ₂	$77.54\pm 2.10\times 10^{-3}$	$77.53\pm 9.80\times 10^{-3}$	$77.03\pm 8.52\times 10^{-2}$	$76.84\pm 1.09\times 10^{-1}$	$77.49\pm 1.35\times 10^{-2}$	$77.35\pm 2.53\times 10^{-2}$	$77.01\pm 1.34\times 10^{-1}$	$77.39\pm 4.98\times 10^{-2}$
Omni-test ₃	$94.59\pm 8.80\times 10^{-3}$	$94.55\pm 2.42\times 10^{-2}$	$92.88\pm 2.72\times 10^{-1}$	$92.29\pm 3.06\times 10^{-1}$	$94.31\pm 6.65\times 10^{-1}$	$93.86\pm 1.08\times 10^{-1}$	$92.61\pm 2.41\times 10^{-1}$	$93.78\pm 2.11\times 10^{-1}$

表 6 所有算法得到的 IGDx 平均值与标准方差

Table 6 Mean and standard variance of IGDx obtained by all algorithms

测试问题	mean±std							
	DN_NSGAII	MOEAD	MRPS	SMPSO_MM	MMOPIO	SSMOPSO	MMOHEA	CSSMPIO
MMF_1	0.023 7±	0.150 3±	0.014 9±	0.011 7±	0.012 9±	0.011 9±	0.023 3±	0.011 5±
	0.003 3(-)	0.068 5(-)	0.000 7(-)	0.000 3(=)	0.000 3(-)	0.000 5(-)	1.430 3(-)	0.000 5
MMF_2	0.022 3±	0.264 5±	0.009 2±	0.006 5±	0.005 6±	0.006 1±	0.008 5±	0.004 3±
	0.015 0(-)	0.116 3(-)	0.001 0(-)	0.000 7(-)	0.000 6(-)	0.000 6(-)	0.001 2(-)	0.000 6
MMF_3	0.016 6±	0.137 8±	0.007 2±	0.005 3±	0.005 6±	0.005 9±	0.007 9±	0.004 9±
	0.007 1(-)	0.046 7(-)	0.001 0(-)	0.000 5(-)	0.000 1(-)	0.001 0(-)	0.001 1(-)	0.000 7
MMF_4	0.026 8±	0.426 2±	0.008 7±	0.005 3±	0.008 3±	0.007 6±	0.013 6±	0.007 1±
	0.006 3(-)	0.123 5(-)	0.000 3(-)	0.000 5(+)	0.000 3(-)	0.000 3(-)	0.000 7(-)	0.000 3
MMF_5	0.068 9±	0.298 5±	0.030 2±	0.024 6±	0.033 1±	0.024 8±	0.044 8±	0.025 0±
	0.007 1(-)	0.126 3(-)	0.001 3(-)	0.000 9(=)	0.001 6(-)	0.000 7(=)	0.000 7(-)	0.000 8
MMF_6	0.010 9±	0.204 5±	0.027 5±	0.023 8±	0.028 6±	0.023 5±	0.041 2±	0.023 9±
	0.002 4(-)	0.070 8(-)	0.001 2(-)	0.000 6(=)	0.001 6(-)	0.000 8(+)	0.001 0(-)	0.000 8
MMF_7	0.064 3±	0.152 2±	0.009 2±	0.007 1±	0.007 2±	0.008 1±	0.012 8±	0.008 1±
	0.035 2(-)	0.066 2(-)	0.000 3(-)	0.000 3(+)	0.000 3(+)	0.000 4(=)	0.000 4(-)	0.000 4

续表 6

测试问题	mean±std							
	DN_NSGAII	MOEAD	MRPS	SMP SO_MM	MMOPIO	SSMOPSO	MMOHEA	CSSMPIO
MMF ₈	3.025 6±	2.711 4±	0.021 0±	0.016 0±	0.017 4±	0.016 7±	0.029 8±	0.017 4±
	1.073 0(-)	0.586 2(-)	0.001 0(-)	0.000 6(+)	0.000 9(=)	0.000 6(+)	0.003 4(-)	0.000 8
SYM-PART ₁	1.051 3±	12.687 7±	0.045 8±	0.032 3±	0.016 7±	0.029 0±	0.032 1±	0.019 3±
	0.830 0(-)	2.730 0(-)	0.003 5(-)	0.000 3(-)	0.240 2(+)	0.006 1(-)	0.001 5(-)	0.001 3
SYM-PART ₂	0.953 8±	12.788 9±	0.053 3±	0.059 7±	0.023 1±	0.032 6±	0.031 4±	0.021 3±
	0.075 6(-)	2.480 0(-)	0.002 8(-)	0.000 5(-)	0.001 5(-)	0.010 5(-)	0.000 6(-)	0.001 7
SYM-PART ₃	0.955 6±	10.369 7±	0.109 4±	0.065 1±	0.024 1±	0.035 7±	0.084 2±	0.012 3±
	0.756 3(-)	2.490 0(-)	0.024 1(-)	0.013 3(-)	0.008 9(-)	0.004 2(-)	0.016 7(-)	0.000 7
Omni-test ₁	0.889 9±	3.418 3±	1.262 3±	0.134 5±	0.870 6±	0.081 0±	0.082 4±	0.078 6±
	0.165 2(-)	0.413 7(-)	0.021 2(-)	0.031 6(-)	0.061 8(-)	0.027 4(-)	0.003 8(-)	0.018 1
Omni-test ₂	2.197 6±	3.976 3±	1.013 2±	1.047 3±	0.998 7±	0.788 7±	0.989 9±	0.998 3±
	0.292 3(-)	0.483 2(-)	0.065 1(-)	0.076 2(-)	0.033 9(=)	0.068 9(+)	0.100 4(+)	0.103 8
Omni-test ₃	3.163 0±	4.696 4±	1.942 3±	2.028 8±	1.968 4±	1.722 4±	2.118 5±	2.124 4±
	0.512 4(-)	0.455 1(-)	0.086 2(+)	0.081 5(+)	0.050 3(+)	0.064 9(+)	0.011 8(+)	0.090 9
+/-/-	0/0/14	0/0/14	1/0/13	4/3/7	3/2/9	4/2/8	2/0/12	—

3.3.3 决策空间 PSs 分布分析

图 7~9 比较了算法在 MMF₁、MMF₃ 和 SYM-PART₁ 测试函数上得到的 PSs, 图中的水平坐标和垂直坐标分别表示对应决策变量的边界范围。从图中可以看出, MOEAD 求得的 PSs 最差。在 MMF₁ 测试函数上, 不仅 3.3.1 和 3.3.2 小节部分

CSSMPIO 求得 PSP 值和 IGDx 值更优越, 而且从图 7 可以看出 CSSMPIO 求得的 PSs 分布也更均匀。对于性能表现较好的 MMF₃ 和 SYM-PART₁ 测试函数来说, 图 8、图 9 更直观地体现了 CSSMPIO 找到更多的 Pareto 最优解, 且解的分布也更均匀。

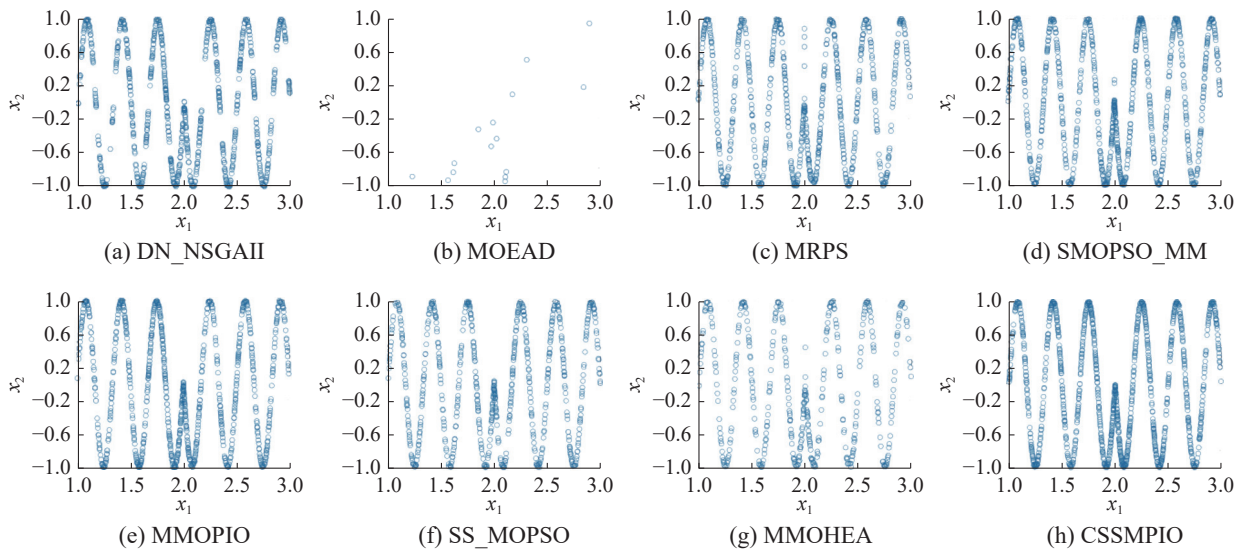
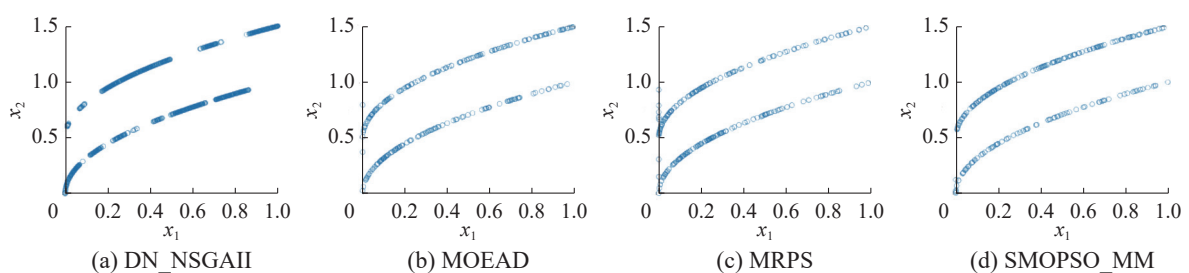
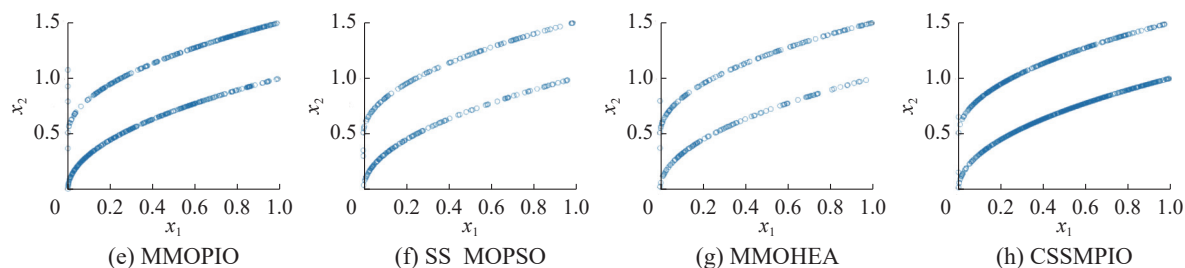
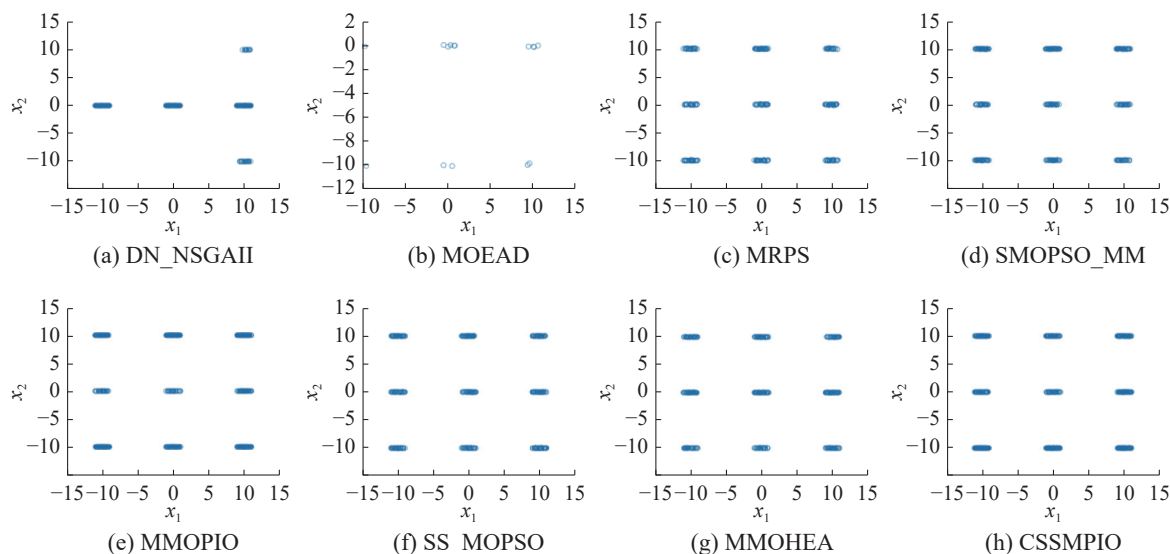


图 7 不同算法在 MMF₁ 上得到的 PSs

Fig. 7 PSs obtained on MMF₁ with different algorithms



图 8 不同算法在 MMF_3 上得到的 PSsFig. 8 PSs obtained on MMF_3 with different algorithms图 9 不同算法在 $SYM-PART_1$ 上得到的 PSsFig. 9 PSs obtained on $SYM-PART_1$ with different algorithms

为了更准确地说明这一点,图 10 表示 CSSMPIO 在其他测试函数上得到的决策空间 Pareto 解集的分布和真实 Pareto 解集分布的图像,“o”代表的是获得的 PS,“+”代表的是真实 PS。从图中可以看出, CSSMPIO 求得的 PSs 大多可以覆盖于真实的 PSs,且 PSs 的分布是均匀的。尤其在复杂

函数 $SYM-PART_2$ 上取得了更优的效果。综合各算法的 PSP 值和 IGDx 值以及 Pareto 解集的分布结果,所提 CSSMPIO 获得的 PSs 能够较好地覆盖真实 PSs,相比于其他算法具有更好的收敛性和多样性,在决策空间和目标空间上实现更好的平衡。

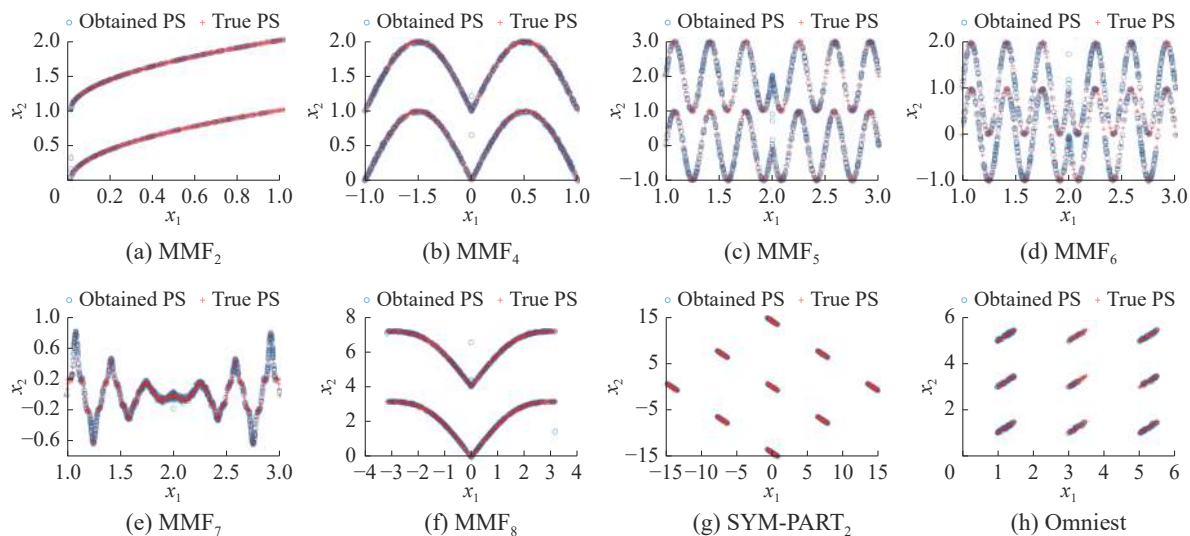


图 10 CSSMPIO 在测试函数上得到的决策空间 PSs 分布

Fig. 10 The PSs distribution of decision space obtained by the CSSMPIO algorithm in test functions

3.4 算法运行时间对比分析

图 11 可视化了不同算法在测试问题上计算一次的平均运行时间, 运行时间刻度设置为 \log_{10} 。从图 11 可以看出, MMOHEA 的时间成本最低, 这是因为算法采用了有效支配排序策略, 但是该算法的计算结果较差。MOEAD 的运行时间是不稳定的; CSSMPIO 采用了小生境搜索策略, 算法运行时间相较 MRPS、DN_NSGAII 较长; 相较于 SMPSO_MM、MMOPIO 来说, CSSMPIO 算法的运行时间相差不多, 但是算法的求解结果更好, 综合来看, CSSMPIO 算法的效率较高。

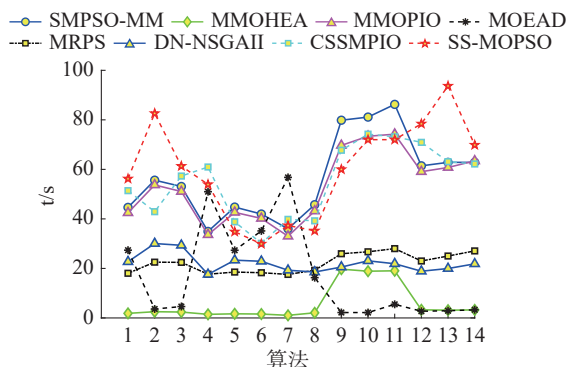


图 11 比较算法在 14 个测试函数上的运行时间

Fig. 11 Runtime of different comparison algorithms on 14 test functions

4 基于地图测试问题的应用

为验证 CSSMPIO 的性能, 采用基于地图的测试问题^[37], 测试问题是从实际地图生成的, 如图 12(a)所示, 涉及 6 所小学(红色圆圈)、3 所中学(绿色圆圈)、13 家便利店(蓝色圆圈)和 3 个火车站(黑色圆圈), 利用决策空间(即地图中的点)中解 x 的欧氏距离定义了 4 个最小化目标: $f_1(x)$: 距离最近的小学(红色圆圈); $f_2(x)$: 距离最近的中学(绿色圆圈); $f_3(x)$: 距离最近的便利店(蓝色圆圈); $f_4(x)$: 距离最近的火车站(黑色圆圈)。

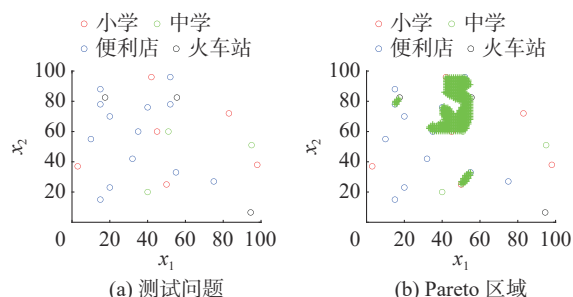


图 12 基于地图的测试问题及 Pareto 区域

Fig. 12 The map-based test problems and the Pareto regions

图 12(b)中的绿色部分表示该问题的帕累托最优区域, 并通过检查 101×101 个点($x_1 = 0, 1, \dots$,

$100; x_2 = 0, 1, \dots, 100$)中每个点的 Pareto 最优性来描述。

为了证明 CSSMPIO 的有效性, 将 CSSMPIO 与 MRPS 在基于地图的问题上进行了比较, 参数设置如表 1。结果如图 13 所示, CSSMPIO 可以覆盖所有 Pareto 区域, 而 MRPS 获得的 Pareto 区域是不完整的。这表明 CSSMPIO 的性能优于 MRPS, 证明了 CSSMPIO 的有效性。

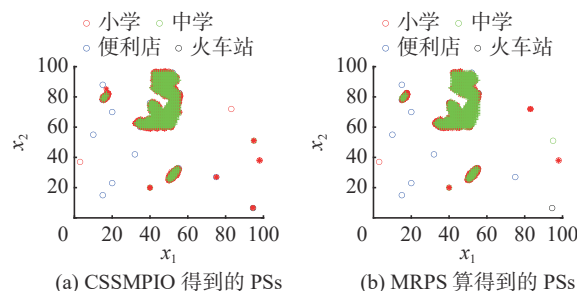


图 13 两种算法得到的解

Fig. 13 Obtained solutions by two algorithms

5 结束语

针对多模态多目标优化中种群多样性难以维持以及等价 Pareto 最优解数量不足问题, 本文提出一种融合聚类和小生境搜索的多模态多目标优化算法(CSSMPIO)。CSSMPIO 采用基于聚类的特殊拥挤距离排序(CSCD)克服了非支配特殊拥挤排序(SCD)不能正确识别邻域的优点, 并且考虑到了决策空间和目标空间的拥挤距离, 个体在自适应物种形成策略中形成的小生境子种群中利用改进的鸽群优化算法并行搜索等价 Pareto 最优解, 最后执行精英学习策略, 从而在决策空间和目标空间中获得一个均匀分布的群体。

综合分析 CSSMPIO 算法与其他 7 种多模态多目标算法在 14 个测试函数上的结果发现, CSSMPIO 算法能够找到更多的 PSs, 提高了决策空间的多样性以及算法搜索能力。另外求得的解集接近于真实解集, 具有较好的收敛性和多样性, 在决策空间和目标空间上实现更好的平衡。

但是本文所提算法是连续优化算法, 不适于求解离散优化问题, 在未来的工作中, 设计出离散优化多模态算法将是研究的方向。

参考文献:

- [1] QU B, SUGANTHAN P N. Novel multimodal problems and differential evolution with ensemble of restricted tournament selection[C]//Proceedings of the IEEE Congress on Evolutionary Computation. Barcelona: IEEE,

- 2010: 1–7.
- [2] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE transactions on evolutionary computation*, 2002, 6(2): 182–197.
- [3] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *IEEE transactions on evolutionary computation*, 2006, 10(3): 281–295.
- [4] ZHANG Xuwei, LIU Hao, TU Liangping. A modified particle swarm optimization for multimodal multi-objective optimization[J]. *Engineering applications of artificial intelligence*, 2020, 95: 103905.
- [5] 高海军, 潘大志. 星型结构的多目标粒子群算法求解多模态多目标问题 [J]. *计算机工程与科学*, 2020, 42(8): 1472–1481.
- GAO Haijun, PAN Dazhi. A multi-objective particle swarm optimization algorithm with star structure to solve the multi-modal multi-objective problem[J]. *Computer engineering and science*, 2020, 42(8): 1472–1481.
- [6] QU B Y, SUGANTHAN P N, DAS S. A distance-based locally informed particle swarm model for multimodal optimization[J]. *IEEE transactions on evolutionary computation*, 2013, 17(3): 387–402.
- [7] QU Boyang, LI Chao, LIANG Jing, et al. A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems[J]. *Applied soft computing*, 2020, 86: 105886.
- [8] LIANG Jing, YUE Caitong, QU Boyang. Multimodal multi-objective optimization: a preliminary study[C]// 2016 IEEE Congress on Evolutionary Computation. Vancouver: IEEE, 2016: 2454–2461.
- [9] ZHANG Kai, SHEN Chaonan, YEN G G, et al. Two-stage double niched evolution strategy for multimodal multiobjective optimization[J]. *IEEE transactions on evolutionary computation*, 2021, 25(4): 754–768.
- [10] YUE Caitong, QU Boyang, LIANG Jing. A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems[J]. *IEEE transactions on evolutionary computation*, 2018, 22(5): 805–817.
- [11] WANG Ying, YANG Zhile, GUO Yuanjun, et al. A novel multi-objective competitive swarm optimization algorithm for multi-modal multi objective problems[C]// 2019 IEEE Congress on Evolutionary Computation. Wellington: IEEE, 2019: 271–278.
- [12] LIANG Jing, GUO Qianqian, YUE Caitong, et al. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems[C]// International Conference on Swarm Intelligence. Cham: Springer, 2018: 550–560.
- [13] HU Yi, WANG Jie, LIANG Jing, et al. A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm[J]. *Science China information sciences*, 2019, 62(7): 70206.
- [14] WANG Yong, LI Hanxiong, YEN G G, et al. MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems[J]. *IEEE transactions on cybernetics*, 2015, 45(4): 830–843.
- [15] LUO Naili, LIN Wu, HUANG Peizhi, et al. An evolutionary algorithm with clustering-based assisted selection strategy for multimodal multiobjective optimization[J]. *Complexity*, 2021, 2021: 1–13.
- [16] LUO Naili, YE Yulong, LIN Wu, et al. A novel multimodal multiobjective memetic algorithm with a local detection mechanism and a clustering-based selection strategy[J]. *Memetic computing*, 2023, 15(1): 31–43.
- [17] HU Yi, WANG Jie, LIANG Jing, et al. A two-archive model based evolutionary algorithm for multimodal multi-objective optimization problems[J]. *Applied soft computing*, 2022, 119: 108606.
- [18] LIU Yiping, YEN G G, GONG Dunwei. A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies[J]. *IEEE transactions on evolutionary computation*, 2019, 23(4): 660–674.
- [19] YANG Qite, WANG Zhenkun, LUO Jianping, et al. Balancing performance between the decision space and the objective space in multimodal multiobjective optimization[J]. *Memetic computing*, 2021, 13(1): 31–47.
- [20] 胡洁, 范勤勤, 王直欢. 融合分区和局部搜索的多模态多目标优化 [J]. *智能系统学报*, 2021, 16(4): 774–784.
- HU Jie, FAN Qinqin, WANG Zhihuan. Multimodal multi-objective optimization combining zoning and local search[J]. *CAAI transactions on intelligent systems*, 2021, 16(4): 774–784.
- [21] LIANG Jing, XU Weiwei, YUE Caitong, et al. Multimodal multiobjective optimization with differential evolution[J]. *Swarm and evolutionary computation*, 2019, 44: 1028–1059.
- [22] ZHANG Weizheng, LI Guoqing, ZHANG Weiwei, et al. A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization[J]. *Swarm and evolutionary computation*, 2019, 50: 100569.
- [23] LIN Qiuzhen, LIN Wu, ZHU Zexuan, et al. Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces[J]. *IEEE transactions on evolutionary computation*, 2021, 25(1): 130–144.
- [24] LIANG Jing, QIAO Kangjia, YUE Caitong, et al. A clus-

- tering-based differential evolution algorithm for solving multimodal multi-objective optimization problems[J]. *Swarm and evolutionary computation*, 2021, 60: 100788.
- [25] SHI Ruizhi, LIN Wu, LIN Qiuzhen, et al. Multimodal multi-objective optimization using A density-based one-by-one update strategy[C]//2019 IEEE Congress on Evolutionary Computation. Wellington: IEEE, 2019: 295–301.
- [26] DUAN Haibin, QIAO Peixin. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning[J]. *International journal of intelligent computing and cybernetics*, 2014, 7(1): 24–37.
- [27] 陶国娇, 李智. 带认知因子的交叉鸽群算法 [J]. 四川大学学报(自然科学版), 2018, 55(2): 295–300.
- TAO Guojiao, LI Zhi. A crossed pigeon-inspired optimization algorithm with cognitive factor[J]. *Journal of Sichuan university (natural science edition)*, 2018, 55(2): 295–300.
- [28] 尹德鑫, 张达敏, 蔡朋宸, 等. 基于鸽群算法的 Fuch 混沌蝗虫算法 [J]. 计算机应用研究, 2021, 38(7): 2013–2017.
- YIN Dexin, ZHANG Damin, CAI Pengchen, et al. Fuch chaotic grasshopper algorithm based on pigeon swarm algorithm[J]. *Application research of computers*, 2021, 38(7): 2013–2017.
- [29] LIU Hanmin, YAN Xuesong, WU Qinghua. An improved pigeon-inspired optimisation algorithm and its application in parameter inversion[J]. *Symmetry*, 2019, 11(10): 1291.
- [30] 马龙, 卢才武, 顾清华, 等. 引入改进鸽群搜索算子的粒子群优化算法 [J]. 模式识别与人工智能, 2018, 31(10): 909–920.
- MA Long, LU Caiwu, GU Qinghua, et al. Particle swarm optimization with search operator of improved pigeon-inspired algorithm[J]. *Pattern recognition and artificial intelligence*, 2018, 31(10): 909–920.
- [31] 岳彩通, 梁静, 瞿博阳, 等. 多模态多目标优化综述 [J]. 控制与决策, 2021, 36(11): 2577–2588.
- YUE Caitong, LIANG Jing, QU Boyang, et al. A survey on multimodal multiobjective optimization[J]. *Control and decision*, 2021, 36(11): 2577–2588.
- [32] QIU Huaxin, DUAN Haibin. Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design[J]. *Science China technological sciences*, 2015, 58(11): 1915–1923.
- [33] ZHANG Qingfu, LI Hui. MOEA/D: a multiobjective evolutionary algorithm based on decomposition[J]. *IEEE transactions on evolutionary computation*, 2007, 11(6): 712–731.
- [34] RUDOLPH G, NAUJOKS B, PREUSS M. Capabilities of EMOA to detect and preserve equivalent Pareto subsets [C]//Proceedings of the 4th international conference on Evolutionary multi-criterion optimization. New York: ACM, 2007: 36–50.
- [35] DEB K, TIWARI S. Omni-optimizer: a procedure for single and multi-objective optimization[C]//Proceedings of the Third international conference on Evolutionary Multi-Criterion Optimization. New York: ACM, 2005: 47–61.
- [36] DERRAC J, GARCÍA S, MOLINA D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. *Swarm and evolutionary computation*, 2011, 1(1): 3–18.
- [37] ISHIBUCHI H, AKEDO N, NOJIMA Y. A many-objective test problem for visually examining diversity maintenance behavior in a decision space[C]//Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. New York: ACM, 2011: 649–656.

作者简介:



顾清华, 教授, 博士生导师, 主要研究方向为多目标优化、车辆调度和复杂系统建模与仿真。以第一或通信作者发表学术论文 95 篇。



唐慧, 硕士研究生, 主要研究方向为多模态多目标优化、露天矿车铲配置优化。



李学现, 博士研究生, 主要研究方向为群智能优化算法在采矿系统工程中的应用。