



融合分区和局部搜索的多模态多目标优化

胡洁, 范勤勤, 王直欢

引用本文:

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.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202010026>

您可能感兴趣的其他文章

响应动态约束条件的多目标货位优化算法研究

Multi-objective location optimization algorithm in response to dynamic constraints
智能系统学报. 2020, 15(5): 925–933 <https://dx.doi.org/10.11992/tis.201906041>

布谷鸟搜索算法研究及其应用进展

Overview of the cuckoo search algorithm and its applications
智能系统学报. 2020, 15(3): 435–444 <https://dx.doi.org/10.11992/tis.201811005>

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>

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>

多特征的光学遥感图像多目标识别算法

Research on multi-feature based multi-target recognition algorithm for optical remote sensing image
智能系统学报. 2016, 11(5): 655–662 <https://dx.doi.org/10.11992/tis.201511011>

微信公众平台



关注微信公众号, 获取更多资讯信息

DOI: 10.11992/tis.202010026

网络出版地址: <https://kns.cnki.net/kcms/detail/23.1538.TP.20210407.1558.007.html>

融合分区和局部搜索的多模态多目标优化

胡洁¹, 范勤勤^{1,2}, 王直欢¹

(1. 上海海事大学 物流研究中心, 上海 201306; 2. 上海交通大学 系统控制与信息处理教育部重点实验室, 上海 200240)

摘要: 为解决多模态多目标优化中种群多样性维持难和所得等价解数量不足问题, 基于分区搜索和局部搜索, 本研究提出一种融合分区和局部搜索的多模态多目标粒子群算法 (multimodal multi-objective particle swarm optimization combining zoning search and local search, ZLS-SMPSO-MM)。在所提算法中, 整个搜索空间被分割成多个子空间以维持种群多样性和降低搜索难度; 然后, 使用已有的自组织多模态多目标粒子群算法在每个子空间搜索等价解和挖掘邻域信息, 并利用局部搜索能力较强的协方差矩阵自适应算法对有潜力的区域进行精细搜索。通过 14 个多模态多目标优化问题测试, 并与其他 5 种知名算法进行比较; 实验结果表明 ZLS-SMPSO-MM 在决策空间能够找到更多的等价解, 且整体性能要好于所比较算法。

关键词: 多模态多目标优化; 分区搜索; 局部搜索; 协方差矩阵自适应策略; 种群多样性; 等价解; 多模态多目标粒子群算法

中图分类号: TP301.6 文献标志码: A 文章编号: 1673-4785(2021)04-0774-11

中文引用格式: 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.

英文引用格式: 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.

Multimodal multi-objective optimization combining zoning and local search

HU Jie¹, FAN Qinqin^{1,2}, WANG Zhihuan¹

(1. Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China; 2. Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai JiaoTong University, Shanghai 200240, China)

Abstract: To maintain population diversity and find a sufficient number of equivalent solutions in multimodal multi-objective optimization, a multimodal multi-objective particle swarm optimization algorithm with zoning and local searches (ZLS-SMPSO-MM) is proposed in this study. In the proposed algorithm, which is based on zoning search and local search, the entire search space is divided into several subspaces to maintain population diversity and reduce search difficulty. Subsequently, an existing self-organizing multimodal multi-objective particle swarm algorithm is used to search equivalent solutions and mine neighborhood information in each subspace, and the covariance matrix adaptation algorithm, which has a better local search ability, is utilized for a refined search in promising regions. Lastly, the performance of ZLS-SMPSO-MM is tested on 14 multimodal multi-objective optimization problems and compared with that of other five state-of-the-art algorithms. Experimental results show that the proposed algorithm can find more equivalent solutions in the decision space and its overall performance is better than that of the compared algorithms.

Keywords: multimodal multi-objective optimization; zoning search; local search; covariance matrix adaptation evolutionary strategy; population diversity; equivalent solutions; multimodal multi-objective particle swarm optimization

收稿日期: 2020-10-23. 网络出版日期: 2021-04-07.

基金项目: 国家重点研发计划项目 (2016YFC0800200); 国家自然科学基金项目 (61603244); 中国博士后科学基金项目 (2018M642017).

通信作者: 范勤勤. E-mail: forever123fan@163.com.

现实生活中的问题往往会涉及多个优化目标, 且它们可能彼此冲突、相互制约, 这类问题被称为多目标优化问题 (multi-objective optimization

problem, MOP)^[1]。而多模态多目标优化问题 (multimodal multi-objective optimization problem, MMOP) 是其中一类较特殊的问题。相比于传统的多目标优化问题, 它在决策空间的多个解可能会有相同的目标值。故多模态多目标优化问题不仅要找到多样性好和逼近性好的近似帕累托前沿 (pareto front, PF), 而且要在决策空间找到尽可能多的等价解^[2]。

由于多模态多目标问题在近几年才受到学者们的重视和研究, 故相比于多目标优化算法的研究, 其成果相对较少。基于 Li^[3] 提出的无参数小生境算法, Yue 等^[4] 在此基础上提出基于环形拓扑结构的粒子群算法 (multi-objective particle swarm optimizer using ring-topology, MO_Ring_PSO_SCD) 来解决多模态多目标问题, 该算法除引入基于索引的环形拓扑结构外, 还在决策空间和目标空间中设计一种新的特殊的拥挤距离来进行粒子选择与更新。结果表明, 该算法能定位和保持大量的等价解; Liang 等^[5] 提出一种自组织多模态多目标粒子群算法 (self-organizing multi-objective particle swarm optimization algorithm, SMPSO-MM)。该算法使用自组织映射网络构建粒子间的邻域关系并进行邻域间信息交流; 另外引入精英策略避免算法陷入停滞。实验结果表明该算法能够定位到更多等价解, 决策空间解的分布也较均匀; Li 等^[6] 提出一种基于适应度排序与强化学习的多模态多目标算法 (differential evolution based on reinforcement learning with fitness ranking, DE-RLFR), 该算法首先使用适应度函数联合排序值确定种群中每个个体的分层状态, 再根据分层状态确定进化方向和变异策略, 最后利用强化学习来引导种群搜索。实验证明该算法可以显著提高决策空间中种群搜索效率, 在解决多模态多目标问题时具有较好表现; Fan 等^[7] 则使用分区概念来提高种群在决策空间中的多样性和降低问题的求解复杂度。研究表明, 该方法能够辅助 MO-Ring-SO-SCD 找到更多等价解。Zhang 等^[8] 提出一种基于聚类的多模态多目标粒子群算法 (cluster based PSO with leader updating mechanism and ring-topology, MMO-CLR-PSO)。该算法使用决策变量聚类方法划分子种群, 利用带有领导粒子更新机制的全局粒子群算法独立地更新每一个子种群的粒子, 并在子种群之间建立环形结构探索未被开发的区域以搜索更多的非支配解。实验结果证明该算法在决策空间与目标空间的解集都能维持较好的多样性; Li-

ang 等^[9] 提出一种多模态多目标差分进化优化算法 (multimodal multi-objective differential evolution optimization algorithm, MMODE)。该算法使用预选择机制将所有个体进行适应度值排序后再根据决策空间中的拥挤距离选择个体进行变异, 同时引入新的边界处理方法将停留在边界的个体重新调整。实验结果证明利用 MMODE 算法获得的 Pareto 解在决策空间和目标空间中都有一个较好的表现; Liu 等^[10] 提出一种基于归档和重组策略的多模态多目标算法 (multimodal multi-objective evolutionary algorithm using two-archive and recombination strategies, TriMOEA-TA&R)。该算法使用决策变量分析技术将种群个体分别归入多样性存档和收敛性存档中, 各自从两个存档中选择父代进行交叉复制。此外, 多样性存档中使用聚类算法及清除策略来保证目标空间多样性。最终将两个存档的解重组以形成最终解集。实验结果证明, 两个存档的分工减少了环境选择的难度, 且算法的整体性能明显优于比较算法; Lin 等^[11] 提出一种决策空间和目标空间双重聚类的多模态多目标进化算法 (multimodal multi-objective evolutionary algorithm with dual clustering in decision and objective spaces, MMOEA/DC)。该算法在决策空间和目标空间均采用聚类算法来保持两个空间的多样性。在决策空间根据邻域关系将父代与子代结合并划分为多个类别, 将这些类中获得的非支配解及其他收敛性好的解对应应在目标空间中的解重新划分为多个类。实验结果表明该算法能够有效定位到全局 Pareto 解和局部 Pareto 解, 并且在两个空间中都有较好的多样性。Zhang 等^[12] 提出一种改进的粒子群算法 (modified particle swarm optimization, MPSO) 求解多模态多目标问题。该算法引入一种基于邻域的动态学习策略代替全局学习策略, 并使用子代竞争机制来提高算法的多样性。实验结果证明该算法在多数测试函数上的表现都优于其他比较算法。Li 等^[13] 提出一种基于高斯采样 (multi objective particle swarm optimization based on gaussian sampling, GS-MOPSO) 的多目标粒子群优化算法以求解多模态多目标问题。该算法在搜索前期利用全局高斯采样方法来进行全局搜索, 在搜索后期则采用局部高斯采样来寻找有潜力解的邻域。此外, GS-MOPSO 算法采用一种新的外部存档策略来储存历史解。实验结果表明, 该算法能够找到更多 Pareto 解。

虽然诸多学者对多模态多目标进化算法进行改进, 但如何保持/提高种群多样性和提高局部搜

索能力来找到更多等价解仍需得到进一步研究。为提高多模态多目标进化算法的性能,本文提出一种融合分区和局部搜索的多模态多目标粒子群算法(multimodal multi-objective particle swarm optimization combining zoning search and local search, ZLS-SMPSO-MM)。在ZLS-SMPSO-MM算法中,整个决策空间首先被划分成多个子空间;然后在各个子空间内,自组织映射网络被用来构建各个子空间的邻域,并使用协方差矩阵自适应算法(covariance matrix adaptation evolutionary strategies, CMA-ES)来增强算法的局部搜索能力;最后对所有子空间得到的解集进行合并选择。为验证所提算法性能,本文选取5种知名多模态多目标进化算法和14个多模态多目标问题进行比较与测试。所得实验结果表明,ZLS-SMPSO-MM不仅能够维持种群多样性以找到更多的等价解,而且能够在较短时间内找到高质量的解。

1 预备知识

1.1 多目标优化问题

不失一般性,多目标优化问题(以最小化问题为例)的数学形式表示如下^[13-14]:

$$\begin{aligned} \min \quad & \mathbf{y} = F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} = (x_1, x_2, \dots, x_D) \in X \subseteq R^D \end{aligned} \quad (1)$$

式中: \mathbf{x} 为 D 维的决策矢量; X 为 D 维的决策空间; $\mathbf{y} = (y_1, y_2, \dots, y_m) \in Y \subseteq R^m$ 为 m 维的目标矢量; Y 为 m 维的目标空间。其他一些相关概念如下^[15]:

定理 1 支配关系: 向量 \mathbf{p} 支配另一个向量 \mathbf{q} (记作 $\mathbf{p} < \mathbf{q}$) 的条件是: 如果 $\forall \theta \in \{1, 2, \dots, \varphi\}$, $p_\theta \leq q_\theta$, 并且 $\mathbf{p} \neq \mathbf{q}$ 。

定理 2 Pareto 最优解集 (pareto set, PS): 一个向量 $\mathbf{x}^* \in X$, 若不存在其他向量 $\mathbf{x} \in X$, 使得 $F(\mathbf{x}) < F(\mathbf{x}^*)$, 就称 \mathbf{x}^* 为 Pareto 解, 所有 Pareto 解构成的集合 (记作 X^*) 被称为 Pareto 解集。

定理 3 Pareto 前沿: 在多目标优化问题中, Pareto 解集对应目标空间中的目标向量被称为 Pareto 前沿, 表示为 $\text{PF} = \{F(\mathbf{x}^*) | \mathbf{x}^* \in X^*\}$ 。

1.2 多模态多目标优化问题及评价指标

多模态多目标优化问题是一种特殊的多目标优化问题, 其主要表现为两种情况: 1) 决策空间中每个解都有多个等价解; 2) 决策空间中部分解有多个等价解^[7]。由于多模态多目标优化问题不仅需要在目标空间获得逼近性及多样性较好的 PF 逼近, 也需要在决策空间中获得足够多等价解。因此, 本文引入帕累托解集近似 (pareto set proximity, PSP)^[4] 和超体积值 (hypervolume,

HV)^[16] 这两种评价指标来评价多模态多目标优化算法的性能。

1) PSP

$$\text{PSP} = \frac{\text{CR}}{\text{IGD}_x} \quad (2)$$

$$\text{IGD}_x(\text{PS}^*, \text{PS}) = \frac{\sum_{b \in \text{PS}} d(b, \text{PS}^*)}{|\text{PS}|} \quad (3)$$

$$\text{CR} = \left(\prod_j \delta_j \right)^{1/2D} \quad (4)$$

$$\delta_j = \begin{cases} 1, & B_j^{\max} = B_j^{\min} \\ 0, & b_j^{\min} \geq B_j^{\max} \text{ 或 } b_j^{\max} \leq B_j^{\min} \\ \frac{\min(b_j^{\max}, B_j^{\min}) - \max(b_j^{\min}, B_j^{\max})}{B_j^{\max} - B_j^{\min}}, & \text{其他} \end{cases} \quad (5)$$

式中: CR (cover rate) 表示所得解集中解个数与真实 PS 中解个数的比值, 也称为解的覆盖率; IGD_x 表示决策空间中的反向世代距离^[14]; b_j^{\max} 和 b_j^{\min} 是算法获得的解集 PS^* 中第 j 个变量的最大值和最小值; B_j^{\max} 和 B_j^{\min} 是 PS 中第 j 个变量的最大值和最小值; $d(b, \text{PS}^*)$ 是 b 与 PS 中最近点的欧几里得距离; $|\text{PS}|$ 表示 PS 中解的数量。PSP 主要评价得到的解集 PS^* 与 PS 之间的相似性。PSP 的值越大, 表示算法得到的等价解越多, 算法的性能就越好。

2) HV

$$\begin{aligned} \text{HV}(\text{PS}^*) = \text{VOL} \left(\bigcup_{\mathbf{x} \in \text{PS}^*} [\mathbf{f}_1(\mathbf{x}), \mathbf{r}_1^*] \times [\mathbf{f}_2(\mathbf{x}), \mathbf{r}_2^*] \times \dots \times \right. \\ \left. [\mathbf{f}_m(\mathbf{x}), \mathbf{r}_m^*] \right) = [(\mathbf{f}_1(\mathbf{x}) - \mathbf{r}_1^*)(\mathbf{f}_1(\mathbf{x}) - \mathbf{r}_1^*)^T + [(\mathbf{f}_2(\mathbf{x}) - \mathbf{r}_2^*)(\mathbf{f}_2(\mathbf{x}) - \mathbf{r}_2^*)^T \dots + [(\mathbf{f}_m(\mathbf{x}) - \mathbf{r}_m^*)(\mathbf{f}_m(\mathbf{x}) - \mathbf{r}_m^*)^T] \end{aligned} \quad (6)$$

式中: $\text{VOL}()$ 是勒贝格测度; $\mathbf{r}^* = (\mathbf{r}_1^*, \mathbf{r}_2^*, \dots, \mathbf{r}_m^*)$ 是选择的参考点, 该参考点被目标空间中所有获得的 PF 逼近中的个体所支配。HV 可以用来衡量 PF 的收敛性和多样性。HV 的值越大, 代表算法获得的 PF 逼近越接近真实 PF, 多样性也越好。

2 融合分区和局部搜索的多模态多目标粒子群算法

2.1 分区搜索

对于求解多模态多目标优化问题来说, 提高/维持种群多样性是影响其最后结果的一个重要因素。同时, 降低问题求解复杂度也能辅助算法找到质量更好的解。鉴于文献^[7]中所提方法能够实现以上两个目标, 故所提算法使用分区搜索 (zoning search, ZS) 来求解多模态多目标优化问题。搜索空间分割步骤: 首先从决策空间 X 的 D 维优化问题中随机选取 h ($1 \leq h \leq D$) 个变量, 将

每个变量都分割成 $l(l>1)$ 段, 最终, 整个决策空间被分为 l^h 个子空间。

2.2 自组织多模态多目标粒子群算法

在所提算法中, 选取自组织多模态多目标粒子群算法^[5] (self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems, SMOPSO) 为基本的搜索算法, 其主要步骤如下:

1) 构建自组织映射网络 (self-organizing maps, SOM)。该步骤主要利用 SOM 网络为粒子群算法中的粒子构建邻域关系, 并在根据邻域关系获得的神经元集合中选取合适的引导粒子指导种群进化。SOM 网络构建邻域关系主要有以下几步^[5,17-18]:

①判断获胜神经元。

SOM 网络中输入的数据维度为 D , 将种群的位置信息 $\mathbf{x} = (x_1, x_2, \dots, x_D)$ 输入网络, 隐藏层中的每个神经元与输入层是全连接的结构, 所以神经元的权值矩阵的行秩与输入空间的维度相同, 神经元权值矩阵的列秩则与神经元个数相同。因此, 整个隐藏层的权值矩阵表示为

$$\mathbf{w} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1D} \\ w_{21} & w_{22} & \cdots & w_{2D} \\ \vdots & \vdots & & \vdots \\ w_{\varepsilon 1} & w_{\varepsilon 2} & \cdots & w_{\varepsilon D} \end{bmatrix} \quad (7)$$

其中 ε 代表 SOM 网络中神经元个数。

竞争过程就是找到与输入向量 \mathbf{x} 最佳匹配的权值向量, 等价于找到与向量 \mathbf{x} 欧几里得距离最小的权值向量, 该权值向量所对应的神经元也被称之为获胜神经元。其主要公式为

$$d^2(\mathbf{x}, \mathbf{w}_{u_p}) = (\mathbf{x} - \mathbf{w}_{u_p})^T (\mathbf{x} - \mathbf{w}_{u_p}) \quad (8)$$

式中: $\rho \in (1, \varepsilon)$; u_p 代表获胜神经元编号; $\mathbf{w}_{u_p} = [w_{\rho 1} \ w_{\rho 2} \ \cdots \ w_{\rho D}]^T$ 。

用索引 $I(u_p)$ 来标识与输入向量 \mathbf{x} 最佳匹配的权值向量, 其公式为

$$I(u_p) = \operatorname{argmin} \|\mathbf{x} - \mathbf{w}_{u_p}\| \quad (9)$$

②获取邻域神经元信息并更新权值。

根据竞争过程产生的获胜神经元的索引确定获胜神经元在隐藏层中的位置, 并根据邻域半径确定其邻域神经元。根据已获得的获胜神经元的的信息更新邻域神经元的权值信息, 权值更新为

$$\mathbf{w}^{g+1} = \mathbf{w}^g + \eta^g (\mathbf{x} - \mathbf{w}^g) \quad (10)$$

$$\eta^{g+1} = \eta^g \times \left(1 - \frac{g}{1-G}\right) \quad (11)$$

式中: g 为当前代数; G 为最大迭代代数; η 为学习率。

2) 获得合适的引导粒子。根据 1) 中构建的 SOM 网络, 记录种群内每个粒子对应的获胜神经

元及每个获胜神经元的邻域神经元, 将获胜神经元及其邻域神经元对应的粒子组成引导粒子的备选库, 并采用非支配排序法选择排序在第一位的粒子作为当前的引导粒子。

3) 粒子速度和位置更新。

$$\mathbf{v}_i^{g+1} = w' \times \mathbf{v}_i^g + c_1 r_1 (\mathbf{x}_{\text{pbest}} - \mathbf{x}_i^g) + c_2 r_2 (\mathbf{x}_{\text{nbest}} - \mathbf{x}_i^g) \quad (12)$$

$$\mathbf{x}_i^{g+1} = \mathbf{x}_i^g + \mathbf{v}_i^{g+1} \quad (13)$$

式中: $\mathbf{v} = (v_1, v_2, \dots, v_D)$ 表示粒子速度; w' 为惯性权重; r_1 和 r_2 是在 $(0, 1)$ 区间正态分布间的随机数; c_1 和 c_2 是两个加速因子; $\mathbf{x}_{\text{pbest}}$ 是个体历史最优; $\mathbf{x}_{\text{nbest}}$ 是个体邻域最优。

4) 粒子速度和位置超出临界值修正。在进行粒子速度和位置更新时, 若粒子速度超出临界值, 则将其设置为临界值; 若粒子位置超出临界值, 则将粒子位置设置为临界值并将该点粒子的速度设置为当前速度的相反数。

5) 当 $g \geq G$ 时, 算法停止。否则回到 1)。

2.3 协方差矩阵自适应策略搜索

协方差矩阵自适应策略 (covariance matrix adaptation evolutionary strategies, CMA-ES) 主要步骤如下^[19]:

1) 对种群内所有粒子进行采样。采样即利用正态分布得到粒子的 λ 个新搜索点作为进入下一代进化的候选解, 其采样公式为

$$\mathbf{x}_k^{g+1} = \mathbf{e}^g + \sigma^g N(0, \mathbf{C}^g), \quad (14)$$

$$\mathbf{e}^g = \sum_{1 \leq i \leq \lambda} w_i \mathbf{x}_i^g, \quad (15)$$

$$\sum_{i=1}^{\lambda} w_i = 1, w_1 \geq w_2 \geq w_3 \geq \cdots \geq w_{\lambda} > 0, \quad (16)$$

式中: $\lambda \geq 2$ ^[20]; k 代表第 $g+1$ 代中第 k 个个体; $\mathbf{e}^g \in \mathbf{R}^n$ 是已被挑选入第 g 代种群的所有个体的加权平均位置; $\sigma^g \in \mathbf{R}^+$ 是第 g 代种群进化的步长; $\mathbf{C}^g \in \mathbf{R}^{D \times D}$ 是第 g 代种群进化的协方差矩阵; $N(0, \mathbf{C}^g)$ 是均值为 0, 协方差矩阵为 \mathbf{C}^g 的多元正态分布; w_i 是采样后产生的个体的权值。

采样完成后, 重新计算 \mathbf{e}^{g+1} 并设置每个个体的权值, 使质量更好的个体有更大概率进行下一步操作。

2) 协方差矩阵更新。使用秩- μ 更新模式来计算连续进化代之间的变异步长, 并调整协方差矩阵。协方差矩阵的更新分为两个部分, 第 1 部分是对进化路径进行更新, 公式为

$$\mathbf{P}_c^{g+1} = h_{\sigma} \sqrt{c_c(2-c_c)} \mu_w (\mathbf{e}^{g+1} - \mathbf{e}^g) / \sigma^g + (1-c_c) \mathbf{P}_c^g \quad (17)$$

$$h_{\sigma}(c) = \begin{cases} 1, & c \geq 0 \\ 0, & c < 0 \end{cases} \quad (18)$$

$$\mu_w = \frac{1}{\sum_i w_i^2} \quad (19)$$

式中: h_σ 为 Heaviside 函数; c_c 为权值系数; μ_w 为方差有效选择质量^[20]。第2部分是对协方差矩阵进行更新, 公式为

$$\mathbf{C}^{g+1} = (1 - c_\gamma)\mathbf{C}^g + c_\gamma \mathbf{P}_c^{g+1} (\mathbf{P}_c^{g+1})^T \quad (20)$$

其中 c_γ 是协方差矩阵的学习率。

3) 全局步长控制。全局步长控制进化路径更新与2)的进化路径相似, 公式为

$$\mathbf{P}_\sigma^{g+1} = \sqrt{c_\sigma(2 - c_\sigma)} \mu_{\text{eff}} (\mathbf{e}^{g+1} - \mathbf{e}^g) / \sigma^g + (1 - c_\sigma) \mathbf{P}_\sigma^{(g)} \quad (21)$$

式中: c_σ 为进化路径的学习率; μ_{eff} 为方差有效选择质量, 计算方式与 μ_w 一致^[20]。

全局步长更新公式为

$$\sigma^{g+1} = (\sigma^g)^{\frac{1}{d_\sigma} \left(\frac{\|\mathbf{P}_\sigma^{g+1}\|}{\|\mathbf{P}_\sigma^{(g)}\|} - 1 \right)} \quad (22)$$

其中 d_σ 是接近阻尼1的系数。

4) 停止准则。当算法消耗评价次数大于CMA-ES算法最大评价次数时, 算法停止。否则, 回到2)。

2.4 融合分区和局部搜索的多模态多目标进化算法

基于2.1~2.3节, 融合分区和局部搜索的多模态多目标算法的具体实现步骤如下:

输入 种群规模 NP ; 算法最大评价次数 A ; 分区数量 z_n ; 粒子群参数 w 、 r_1 、 r_2 、 c_1 、 c_2 ; 子空间外部存档容量 Q ; CMA-ES 算法参数 c_γ 、 c_c 、 c_σ 、局部搜索步长 σ 、单个个体进行CMA-ES搜索消耗的最大评价次数 A_1 ; 单个个体进行CMA-ES搜索消耗的累计评价次数 A_2 ; CMA-ES 算法加入时消耗的算法总评价次数 A_3 ; 算法累计评价次数 A_4 。

输出 \mathbf{PS}^* 和 \mathbf{PF}^* 。

1) 初始化种群 $\mathbf{x} = (x_1, x_2, \dots, x_{NP})$, 子空间外部存档 \mathbf{Q} , $A_1=12$, $A_2=0$, $A_3=0$, $A_4=0$; CMA-ES 算法其他参数与文献[19]保持一致。

2) 根据2.1节对决策空间进行分割, 得到4个子空间。

3) 在各个子空间中使用2.2节的多模态多目标进化算法进行搜索。

4) 当 $A_3=0$ 时, 分别对各个子空间中得到的非支配解使用2.3节的局部搜索算法。当单个粒子消耗的累计评价次数大于 A_1 时, 该粒子的局部搜索过程结束。下一粒子继续进行局部搜索。直到 NP 个粒子完成局部搜索, 或者 $A_4 \geq A$ 时局部搜索结束。

5) 每个参与局部搜索的粒子与其产生的子代

均使用基于特殊拥挤距离的非支配排序法^[4]进行排序, 排在第一位的粒子取代原本参与局部搜索的粒子进入粒子群, 其余非支配解均加入外部存档。当外部存档内的非支配解个数大于外部存档容量 Q 时, 继续使用基于特殊拥挤距离的非支配排序法进行排序, 将前 Q 个解保留。若外部存档内非支配解个数小于 Q , 则全部保留。

6) 循环3)~5), 直到 $A_4 \geq A_1$ 时跳出循环。

7) 将每个子空间内最后一代粒子群与子空间外部存档合并, 使用基于特殊拥挤距离的非支配排序法选出每个子空间的非支配解。将4个子空间的解集合并后使用环境选择得到最终解集 $\mathbf{PS}^* = \text{selection}(\mathbf{PS}_1^* \cup \mathbf{PS}_2^* \cup \mathbf{PS}_3^* \cup \mathbf{PS}_4^*)$ 和近似帕累托前沿 $\mathbf{PF}^* = \text{selection}(\mathbf{PF}_1^* \cup \mathbf{PF}_2^* \cup \mathbf{PF}_3^* \cup \mathbf{PF}_4^*)$ 。

3 实验结果与分析

3.1 测试函数

为验证ZLS-SMPSO-MM的性能, 本文选取14个多模态多目标优化问题对其进行测试。其中, 8个MMF问题^[4]、3个SYM-PART问题^[21]、3个OMNI问题^[22], 详情见表1。所有优化问题的目标个数均为2, MMF问题与SYM-PART问题的空间维度为2, OMNI问题的空间维度分别为3、4、5。PSs的数量表示决策空间中的不同PS对应目标空间同一个PF的数量, PSs的数量越大, 问题的求解难度也会相应增大。最后一列表示的是PSs在决策空间的每一维度是否重叠, 一般情况下, PSs在决策空间的重叠会增加问题的复杂度。

表1 多模态多目标优化问题的相关特征

Table 1 Features of multimodal multi-objective optimization problems

问题	目标个数	空间维度	PSs的数量	是否重叠
MMF ₁	2	2	2	否
MMF ₂	2	2	2	否
MMF ₃	2	2	2	是
MMF ₄	2	2	4	否
MMF ₅	2	2	4	否
MMF ₆	2	2	4	是
MMF ₇	2	2	2	否
MMF ₈	2	2	4	否
SYM-I	2	2	9	是
SYM-II	2	2	9	是
SYM-III	2	2	9	是
OMNI-TEST ₁	2	3	27	是
OMNI-TEST ₂	2	4	72	是
OMNI-TEST ₃	2	5	360	是

3.2 实验设置

为验证 ZLS-SMPSO-MM 的性能,本文选取 5 种多模态多目标进化算法来进行比较。其分别是 DN-NSGA-II^[2]、Omni-optimizer^[22]、MO-Ring-PSO-SCD^[4]、SMPSO-MM^[5]、Zoning-MO-Ring-SCD-PSO^[7]。同时,所有比较算法的参数设置与原文献一致。本实验中所有测试函数均为 2 目标问题,所有测试函数的种群数量均设置为 800,最大评价次数为 80 000,单个粒子使用 CMA-ES 算法的评价次数设置为 12。外部存档容量设置为 800。每种算法在测试函数上均独立运行 20 次。为保证实验结果分析的可靠性,采用 Wilcoxon^[23] 和 Friedman^[24] 非参数检验方法对实验结果进行统计分析,显著性水平设置为 5%。其中“+”、“-”分别表示所提算法优于、劣于相比较算法;“≈”表示所提算法与相比较的算法性能相近。

3.3 结果对比

ZLS-SMPSO-MM 与其他 5 种比较算法所得 PSP 值的平均值和标准方差见表 2。同时,由 Wil-

coxon 得到的统计分析结果也在表 2 中显示。从表 2 可以看出,所提算法在 14 个测试函数上所得结果都要显著好于 DN-NSGA-II、Omni-optimizer、MO-Ring-PSO-SCD、ZS-MO-Ring-SCD-PSO。相比于 SMPSO-MM,除在 MMF₃ 函数上取得相似的结果,ZLS-SMPSO-MM 在其他测试函数上都能得到更好的结果。同时,这 6 种算法得到的 PSP 值排序结果如下表 3,在该表中,利用 Friedman 非参数检验方法对实验结果进行统计分析时,将所有算法得到的测试函数的 PSP 值均取其相反数,故排序值越小,算法性能越佳。根据表 3 结果可得,ZLS-SMPSO-MM 在所有比较算法中性能最佳。这是因为使用多模态多目标进化算法进行搜索前先使用分区策略划分了决策空间。在各个互不重叠的子空间区域使用相同规模的粒子群分别进行空间搜索,提升种群多样性的同时也降低了各个子空间的搜索难度;每个子空间内都引入局部搜索算法,在每次全局搜索后的潜力区域用局部搜索能力强的 CMA-ES 算法进行精细搜索,因此能够找到更多高质量的解。

表 2 不同算法得到的 PSP 平均值与标准方差

Table 2 PSP average and standard deviation values obtained by different algorithms

测试函数	ZLS-SMPSO-MM	DN-NSGA-II	Omni-optimizer	MO-Ring-SCD-PSO	SMPSO-MM	ZS-MO-Ring-SCD-PSO
MMF ₁	1.126 4×10² (6.52)	4.209 1×10 ¹ (4.12)+	4.803 2×10 ¹ (4.94)+	6.712 4×10 ¹ (3.45)+	8.636 7×10 ¹ (4.27)+	9.215 8×10 ¹ (5.40)+
MMF ₂	1.904 3×10² (1.45×10 ¹)	6.942 9×10 ¹ (2.30×10 ¹)	7.061 5×10 ¹ (3.63×10 ¹)	1.083 5×10 ² (1.12×10 ¹)	1.520 1×10 ² (1.05×10 ¹)	1.348 7×10 ² (1.61×10 ¹)
MMF ₃	1.848 2×10 ² (2.18×10 ¹)	7.401 8×10 ¹ (3.53×10 ¹)	7.631 2×10 ¹ (1.89×10 ¹)	1.360 4×10 ² (1.97×10 ¹)	1.863 8×10² (1.80×10 ¹)≈	1.308 1×10 ² (1.64×10 ¹)
MMF ₄	2.456 6×10² (1.11)	4.368 4×10 ¹ (8.32)+	3.738 0×10 ¹ (7.87)+	1.158 0×10 ² (4.06)+	1.373 9×10 ² (5.93)+	1.930 6×10 ² (5.81)+
MMF ₅	4.975 5×10¹ (2.61)	1.449 5×10 ¹ (1.64)+	1.494 8×10 ¹ (1.31)+	3.353 7×10 ¹ (1.10)+	4.029 7×10 ¹ (1.51)+	4.384 0×10 ¹ (2.42)+
MMF ₆	6.346 0×10¹ (2.82)	1.775 3×10 ¹ (1.57)+	1.738 9×10 ¹ (1.32)+	3.658 3×10 ¹ (1.62)+	4.189 2×10 ¹ (1.11)+	5.591 9×10 ¹ (2.24)+
MMF ₇	2.212 3×10² (1.10)	9.387 1×10 ¹ (1.34×10 ¹)	9.921 3×10 ¹ (1.69×10 ¹)	1.104 3×10 ² (2.24)+	1.442 5×10 ² (5.30)+	1.797 3×10 ² (7.96)+
MMF ₈	9.219 8×10¹ (9.26)	1.705 7×10 ¹ (6.22)+	1.626 7×10 ¹ (5.81)+	4.651 7×10 ¹ (2.09)+	6.319 0×10 ¹ (2.46)+	6.482 2×10 ¹ (2.60)+
SYM-I	4.608 0×10¹ (6.61)	4.418 5×10 ⁻¹ (1.98×10 ⁻¹)	6.362 2×10 ⁻¹ (8.98×10 ⁻¹)	2.158 3×10 ¹ (1.08)+	3.150 9×10 ¹ (3.16)+	2.096 9×10 ¹ (1.44)+
SYM-II	4.260 9×10¹ (3.65)	4.135 7 (4.95)+	1.923 3 (2.28)+	1.826 2×10 ¹ (1.27)+	1.733 4×10 ¹ (9.63×10 ⁻¹)	1.987 8×10 ¹ (1.11)+
SYM-III	4.244 6×10¹ (8.28)	8.323 5 (1.18×10 ¹)	1.227 7 (2.04)+	9.299 8 (5.04)+	1.614 5×10 ¹ (2.67)+	1.084 4×10 ¹ ((2.13×10 ⁻²))+

续表 2

测试函数	ZLS-SMPSO-MM	DN-NSGA-II	Omni-optimizer	MO-Ring-SCD-PSO	SMPSO-MM	ZS-MO-Ring-SCD-PSO
Omni-I	1.653 3 $\times 10^1$ (1.24)	1.062 4 (1.36×10^{-1}) ⁺	9.026 9 $\times 10^{-1}$ (1.33×10^{-1}) ⁺	7.671 2 (1.42) ⁺	7.701 5 (1.46) ⁺	9.082 1 (1.09) ⁺
Omni-II	1.295 3 (1.18×10^{-2})	4.493 8 $\times 10^{-1}$ (1.03×10^{-1}) ⁺	4.018 0 $\times 10^{-1}$ (8.04×10^{-2}) ⁺	9.377 1 $\times 10^{-1}$ (4.98×10^{-2}) ⁺	9.536 9 $\times 10^{-1}$ (1.02×10^{-1}) ⁺	1.012 0 (7.30×10^{-2}) ⁺
Omni-III	5.294 0 $\times 10^{-1}$ (1.69×10^{-2})	2.645 1 $\times 10^{-1}$ (7.38×10^{-2}) ⁺	2.551 8 $\times 10^{-1}$ (6.75×10^{-2}) ⁺	4.968 7 $\times 10^{-1}$ (1.98×10^{-2}) ⁺	4.855 6 $\times 10^{-1}$ (1.29×10^{-2}) ⁺	5.021 4 $\times 10^{-1}$ (2.13×10^{-2}) ⁺
+/-/≈		14/0/0	14/0/0	14/0/0	13/0/1	14/0/0

表 3 所有比较算法在 PSP 上的性能排序

Table 3 Performance rankings of all compared algorithms on PSP

算法	排序
ZLS-SMPSO-MM	1.4
ZS-MO-Ring-SCD-PSO	2.533 33
SMPSO-MM	2.733 33
MO-Ring-PSO-SCD	3.666 67
DN-NSGA-II	5.133 33
Omni-optimizer	5.533 33

ZLS-SMPSO-MM 与其他 5 种比较算法所得 HV 值的平均值和标准方差见表 4。同时,由 Wilcoxon 得到的统计分析结果也在表 4 中显示。从

表 4 中可以看出, ZLS-SMPSO-MM 只在 MM7、SYM-I、SYM-II、SYM-III 和 OMNI-I 这 5 个测试函数中得到较好的结果, Omni-optimizer 则在剩余的 9 个测试函数中占优势,但它们之间的数值差异并不是很显著。这 6 种比较算法得到的 HV 值排序结果见表 5, ZLS-SMPSO-MM 排在第二位。主要原因是 ZLS-SMPSO-MM 中采取的分区策略和局部搜索只能改变算法在决策空间的性能,对目标空间的影响十分有限。目标空间中仍旧采取了之前的选择策略,因此性能提升效果不明显,但相对于除 Omni-optimizer 外的其他算法,所提算法在目标空间的表现还是要优于其他 4 种算法。因此,若是想提升算法在目标空间的性能就必须要在目标空间中使用合适的多目标处理技术。

表 4 不同算法得到的 HV 平均值与标准方差

Table 4 HV average and standard deviation values obtained by different algorithms

测试函数	ZLS-SMPSO-MM	DN-NSGA-II	Omni-optimizer	MO-Ring-SCD-PSO	SMPSO-MM	ZS-MO-Ring-SCD-PSO
MMF ₁	3.665 2 (3.29×10^{-4})	3.665 1 (3.57×10^{-4}) [≈]	3.665 4 (1.10×10^{-3}) ⁻	3.664 5 (3.87×10^{-4}) ⁺	3.665 1 (3.44×10^{-4}) [≈]	3.664 8 (5.00×10^{-4}) ⁺
MMF ₂	3.658 9 (2.00×10^{-3})	3.665 3 (7.09×10^{-4}) ⁻	3.665 7 (8.01×10^{-5}) ⁻	3.649 7 (6.31×10^{-3}) ⁺	3.657 2 (2.58×10^{-3}) ⁺	3.652 8 (5.20×10^{-3}) ⁺
MMF ₃	3.657 9 (2.24×10^{-3})	3.663 4 (9.03×10^{-3}) ⁻	3.665 7 (3.58-05) ⁻	3.653 0 (6.48×10^{-3}) ⁺	3.659 1 (2.39×10^{-3}) ⁻	3.649 7 (6.67×10^{-3}) ⁺
MMF ₄	3.330 8 (1.30×10^{-3})	3.331 9 (2.44×10^{-4}) ⁻	3.332 3 (2.93 $\times 10^{-5}$) ⁻	3.330 4 (8.13×10^{-4}) [≈]	3.331 3 (9.04×10^{-4}) [≈]	3.330 9 (9.14×10^{-4}) [≈]
MMF ₅	3.665 3 (4.36×10^{-4})	3.665 2 (2.42×10^{-4}) [≈]	3.665 7 (2.37 $\times 10^{-5}$) ⁻	3.664 6 (4.77×10^{-4}) ⁺	3.665 1 (3.64×10^{-4}) [≈]	3.665 0 (5.17×10^{-4}) [≈]
MMF ₆	3.665 4 (3.76×10^{-4})	3.665 2 (2.88×10^{-4}) [≈]	3.665 7 (2.64 $\times 10^{-5}$) ⁻	3.664 7 (3.64×10^{-4}) ⁺	3.665 0 (4.03×10^{-4}) ⁺	3.665 0 (3.90×10^{-4}) ⁺
MMF ₇	3.665 9 (2.68 $\times 10^{-4}$)	3.664 1 (6.57×10^{-4}) ⁺	3.665 4 (5.96×10^{-4}) ⁺	3.664 9 (1.41×10^{-3}) ⁺	3.665 4 (1.97×10^{-4}) ⁺	3.665 5 (3.66×10^{-4}) ⁺
MMF ₈	3.213 5 (3.51×10^{-4})	3.212 2 (9.03×10^{-4}) ⁺	3.213 6 (3.14 $\times 10^{-4}$) [≈]	3.210 6 (1.41×10^{-3}) ⁺	3.212 4 (8.45×10^{-4}) ⁺	3.211 9 (1.18×10^{-3}) ⁺
SYM-I	1.326 9 (2.50 $\times 10^{-4}$)	1.322 2 (3.12×10^{-4}) ⁺	1.322 7 (3.51×10^{-4}) ⁺	1.298 4 (1.74×10^{-3}) ⁺	1.309 4 (1.60×10^{-3}) ⁺	1.299 0 (1.93×10^{-3}) ⁺
SYM-II	1.325 3 (1.97 $\times 10^{-4}$)	1.318 7 (3.90×10^{-4}) ⁺	1.320 7 (1.65×10^{-4}) ⁺	1.291 0 (2.91×10^{-3}) ⁺	1.286 8 (2.87×10^{-3}) ⁺	1.295 9 (2.03×10^{-3}) ⁺

续表4

测试函数	ZLS-SMPSO-MM	DN-NSGA-II	Omni-optimizer	MO-Ring-SCD-PSO	SMPSO-MM	ZS-MO-Ring-SCD-PSO
SYM-III	1.321 6	1.319 2	1.320 9	1.287 5	1.278 3	1.288 8
	(5.10×10⁻⁴)	(3.77×10 ⁻⁴) ⁺	(3.13×10 ⁻⁴) ⁺	(2.70×10 ⁻³) ⁺	(3.89×10 ⁻³) ⁺	(2.63×10 ⁻³) ⁺
Omni-I	6.209 9×10¹	6.205 7×10 ¹	6.205 9×10 ¹	6.197 0×10 ¹	6.197 7×10 ¹	6.198 4×10 ¹
	(7.60×10⁻³)	(5.03×10 ⁻⁴) ⁺	(2.31×10 ⁻⁴) ⁺	(1.34×10 ⁻²) ⁺	(1.18×10 ⁻²) ⁺	(9.82×10 ⁻³) ⁺
Omni-II	7.727 6×10 ¹	7.754 2×10 ¹	7.754 7×10¹	7.701 1×10 ¹	7.699 0×10 ¹	7.705 6×10 ¹
	(4.07×10 ⁻²)	(1.83×10 ⁻³) ⁻	(7.17×10⁻³)⁻	(8.13×10 ⁻¹) ⁺	(1.01×10 ⁻¹) ⁺	(5.68×10 ⁻²) ⁺
Omni-III	9.359 1×10 ¹	9.459 0×10 ¹	9.460 0×10¹	9.285 8×10 ¹	9.262 6×10 ¹	9.280 1×10 ¹
	(1.92×10 ⁻¹)	(7.15×10 ⁻³) ⁻	(4.16×10⁻³)⁻	(2.74×10 ⁻¹) ⁺	(2.30×10 ⁻¹) ⁺	(2.12×10 ⁻¹) ⁺
+/-/~		6/5/3	5/8/1	13/1/0	10/1/3	12/0/2

表5 所有比较算法在HV上的性能排序

Table 5 Performance rankings of all compared algorithms on HV

算法	排序
Omni-optimizer	1.6
ZLS-SMPSO-MM	2.6
DN-NSGA-II	2.666 67
SMPSO-MM	4.066 67
ZS-MO-Ring-SCD-PSO	4.266 67
MO-Ring-PSO-SCD	5.8

综合各算法的PSP值和HV值实验结果,所提ZLS-SMPSO-MM算法获得的帕累托解集能够较好地逼近真实帕累托前沿,相比于其他算法具有更好的收敛性和多样性。

3.4 算法分析

3.4.1 分区搜索与局部搜索对算法的影响

为验证分区搜索与局部搜索的有效性,该部分实验分别选取没有分区搜索和局部搜索的SMPSO-MM、有分区搜索但没有局部搜索的SMPSO-MM(命名为ZS-SMPSO-MM)来和所提算法进行性能比较。同时,所有算法参数设置与3.2节一致;并挑选3.2节14个测试函数进行实验,所得结果见表6。

从ZLS-SMPSO-MM和ZS-SMPSO-MM的结果来看,除MMF₁、MMF₄、MMF₆、MMF₅、MMF₇这5个测试函数外,ZLS-SMPSO-MM在其余测试函数上得到的PSP值均明显优于ZS-SMPSO-MM,尤其是在较复杂的SYM函数与OMNI函数上。这说明局部搜索策略能够有效辅助多模态多目标进化算法找到更高质量和更多的等效解。从ZLS-SMPSO-MM与SMPSO-MM的结果来看,除MMF₃测试函数外,所提算法能够在所有测试函数上取得更佳效果。这表明分区搜索和局部搜索能够有效帮助SMPSO-MM算法找

到更多的等效解。主要是由于分区搜索能够维持算法种群多样性和降低问题求解难度,而局部搜索能够提高搜索精度。另外,从ZS-SMPSO-MM与SMPSO-MM的结果来看,除MMF₃、SYM-I、SYM-II这3个测试函数外,其余测试函数的PSP值均显著优于SMPSO-MM,从实验结果可知,分区搜索能够帮助SMPSO-MM算法定位到更多的等效解,该结论与文献[7]一致。

表6 SMPSO-MM及其变种得到的PSP平均值与标准方差

Table 6 PSP average and standard deviation values obtained by SMPSO-MM and its variant algorithms

测试函数	ZLS-SMPSO-MM	ZS-SMPSO-MM	SMPSO-MM
MMF ₁	1.126 4×10 ²	1.214 1×10¹	8.636 7×10 ¹
	(6.52)	(4.90)⁻	(4.27) ⁺
MMF ₂	1.904 3×10²	1.699 2×10 ²	1.520 1×10 ²
	(1.45×10¹)	(1.63×10 ¹) ⁺	(1.05×10 ¹) ⁺
MMF ₃	1.848 2×10 ²	1.774 4×10 ²	1.863 8×10²
	(2.18×10 ¹)	(1.42×10 ¹) [≈]	(1.80×10¹)[≈]
MMF ₄	2.456 6×10 ²	2.539 5×10²	1.373 9×10 ²
	(1.11)	(1.01×10¹)⁻	(5.93) ⁺
MMF ₅	4.975 5×10¹	4.962 5×10 ¹	4.029 7×10 ¹
	(2.61)	(2.14) [≈]	(1.51) ⁺
MMF ₆	6.346 0×10 ¹	7.297 7×10¹	4.189 2×10 ¹
	(2.82)	(2.82)⁻	(1.11) ⁺
MMF ₇	2.212 3×10 ²	2.529 2×10²	1.442 5×10 ²
	(1.10)	(1.10×10¹)⁻	(5.30) ⁺
MMF ₈	9.219 8×10¹	7.598 2×10 ¹	6.319 0×10 ¹
	(9.26)	(4.39) ⁺	(2.46) ⁺
SYM-I	4.608 0×10¹	2.208 0×10 ¹	3.150 9×10 ¹
	(6.61)	(1.66) ⁺	(3.16) ⁺
SYM-II	4.260 9×10¹	1.689 5×10 ¹	1.733 4×10 ¹
	(3.65)	(1.78) ⁺	(9.63×10 ⁻¹) ⁺
SYM-III	4.244 6×10¹	2.002 6×10 ¹	1.614 5×10 ¹
	(8.28)	(1.67) ⁺	(2.67) ⁺

续表 6

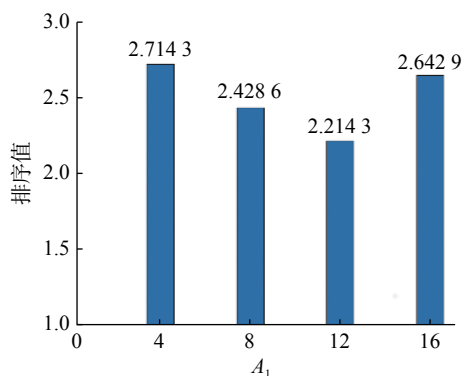
测试函数	ZLS-SMPSO-MM	ZS-SMPSO-MM	SMPSO-MM
Omni-I	1.653×10^1 (1.24)	9.485 6 (9.65×10^{-1}) ⁺	7.701 5 (1.46) ⁺
Omni-II	1.295×10^{-2} (1.18×10^{-2})	1.019 5 (6.76×10^{-2}) ⁺	9.536×10^{-1} (1.02×10^{-1}) ⁺
Omni-III	5.294×10^{-1} (1.69×10^{-2})	4.994×10^{-1} (2.15×10^{-2}) ⁺	4.855×10^{-1} (1.29×10^{-2}) ⁺
+/-/~		8/4/2	13/1/0

由于测试函数的帕累托解在整个决策空间并非均匀分布,因此均衡的分区策略会浪费一些计算资源,如测试函数 MMF₃, SMPSO-MM 加入分区搜索后,其 PSP 值反而下降。但总体来说,同时使用分区搜索与局部搜索能够提高算法求解多模态多目标问题的能力。

3.4.2 局部搜索评价次数 A_1 敏感值分析

由于局部搜索会消耗一定的计算量,局部搜索过多消耗计算资源可能使算法全局求解能力下降,而过少消耗计算资源则可能导致局部搜索算法能力发挥不足。故本部分对单个粒子进行局部搜索使用的评价次数进行分析,并用 PSP 性能指标来对实验结果进行评价。选取 3.1 节的 14 个测试函数进行实验,并将单个粒子进行局部搜索使用的评价次数 A_1 设定为 4、8、12、16,其他参数设定同 3.2 节。

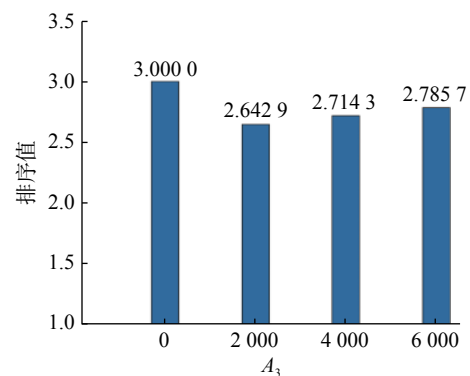
所得结果使用 Friedman 测试来进行性能排序,见图 1。从图 1 中可以看出,一开始随着局部搜索评价次数的增加,算法的性能也会得到相应的提升,但随着评价次数增加到一定的程度,算法的性能反而会降低。这是因为合适的局部搜索评价次数能平衡局部搜索能力与全局搜索能力。另外,根据图 1 的结果显示,当 $A_1 = 12$ 时,算法的性能表现最佳,因此在所提算法中将 A_1 设置为 12。

图 1 A_1 对所提算法的影响Fig. 1 Impact of A_1 on the proposed algorithm

3.4.3 局部搜索评价次数 A_3 敏感值分析

使用局部搜索的时间对算法性能会产生影响。直观来看,过早使用局部搜索会因为所寻得的区域潜力不足而浪费计算资源,过晚使用则可能无法发挥局部搜索的真正作用。因此,本节对局部搜索的使用时间进行分析,并用 PSP 性能指标来对实验结果进行评价。选取 3.1 节的 14 个测试函数进行实验。将局部搜索的使用时间设置为算法已消耗的评价次数。使用时间分为 $A_3 = 0$ 、 $A_3 = 2\,000$ 、 $A_3 = 4\,000$ 和 $A_3 = 6\,000$ 。除该参数外其余实验参数设置同 3.2 节。

所得结果使用 Friedman 测试来进行性能排序,见图 2。从图 2 中可以看出,在种群进化前期加入局部搜索能够更加有效帮助算法找到更多的等效解。其主要原因是局部搜索能够在较短时间内辅助多模态多目标进化算法找到高质量的解。当然,这完全取决于算法能否在进化前期就能找到有潜力的搜索区域。另外,图 2 表示当算法已消耗评价次数不小于 2 000 时,其能使所提算法性能达到最佳,因此在所提算法中设置 $A_3 = 2\,000$ 。

图 2 A_3 对所提算法的影响Fig. 2 Impact of A_3 on the proposed algorithm

3.4.4 种群规模敏感值分析

为分析种群规模对算法的影响,本实验选取第 3.1 节的 14 个测试函数;使用 PSP 作为算法性能评价指标;并将种群规模分别设定为 400、800、1 200、1 600,其他参数设定同第 3.2 节。最大评价次数均为 80 000。

所得结果使用 Friedman 测试来进行性能排序,见图 3。从图 3 中可以看出,随着种群规模的增加,算法的性能逐渐降低。其主要原因是,虽然大的种群规模可以提高各个分区的种群多样性,但在计算资源有限的情况下,搜索代数会减

小, 这将直接影响算法的搜索效率。同时, 当算法种群规模在 400 时, 算法的性能最佳。但与第 3.2 节中比较算法的种群规模保持一致, 所提算法的种群规模仍设置为 800。

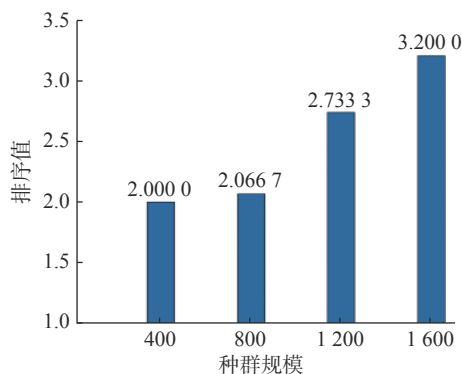


图3 种群规模对所提算法的影响

Fig. 3 Impact of population size on the proposed algorithm

4 结束语

为提高多模态多目标进化算法在搜索过程中的种群多样性和搜索等价解的能力, 本文提出一种融合分区和局部搜索的多模态多目标粒子群算法 ZLS-SMPSO-MM。在该算法中, 首先将整个搜索空间分成多个子空间, 然后使用多模态多目标粒子群算法对各个子空间进行搜索, 并使用局部搜索来提高算法的搜索效率, 最后将各个子空间所得解集进行合并选择。为验证 ZLS-SMPSO-MM 算法的性能, 选取 14 个多模态多目标优化问题, 并将其与 DN-NSGA-II、Omni-optimizer、MO-Ring-PSO-SCD、SMPSO-MM、ZS-MO-Ring-SCD-PSO 等多模态多目标算法进行比较。实验结果表明, 分区搜索和局部搜索能够有效帮助 SMPSO-MM 找到更多的等价解。

参考文献:

- [1] BRANKE J. Multiobjective optimization[M]. Berlin: Springer, 2008.
- [2] LIANG J J, YUE C T, QU B Y. Multimodal multi-objective optimization: a preliminary study[C]//Proceedings of 2016 IEEE Congress on Evolutionary Computation. Vancouver, Canada: IEEE, 2016: 2454–2461.
- [3] LI Xiaodong. Niching without niching parameters: particle swarm optimization using a ring topology[J]. *IEEE transactions on evolutionary computation*, 2010, 14(4): 150–169.
- [4] YUE Caitong, QU Boyang, LIANG Jing. A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems[J]. *IEEE transactions on evolutionary computation*, 2018, 22(5): 805–817.
- [5] LIANG Jing, GUO Qianqian, YUE Caitong, et al. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems[C]//Proceedings of the 9th International Conference on Swarm Intelligence. Shanghai, China: Springer, 2018: 550–560.
- [6] LI Zhihui, SHI Li, YUE Caitong, et al. Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems[J]. *Swarm and evolutionary computation*, 2019, 49: 234–244.
- [7] FAN Qing, YAN Xuefeng. Solving multimodal multiobjective problems through zoning search[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2019.
- [8] 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.
- [9] LIANG Jing, XU Weiwei, YUE Caitong, et al. Multimodal multiobjective optimization with differential evolution[J]. *Swarm and evolutionary computation*, 2019, 44: 1028–1059.
- [10] LIU Yiping, YEN 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.
- [11] 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.
- [12] 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.
- [13] LI Guosen, YAN Li, QU Boyang. Multi-objective particle swarm optimization based on Gaussian sampling[J]. *IEEE access*, 2020, 8: 209717–209737.
- [14] DEB K. Multi-objective evolutionary algorithms: introducing bias among pareto-optimal solutions[M]//GHOSH A, TSUTSUI S. *Advances in Evolutionary Computing*. Berlin: Springer, 2003.
- [15] DEB K. Multi-objective genetic algorithms: problem difficulties and construction of test problems[J]. *Evolutionary computation*, 1999, 7(3): 205–230.
- [16] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach[J]. *IEEE transactions on evolutionary*

- computation, 1999, 3(4): 257–271.
- [17] KOHONEN T. The self-organizing map[J]. Neurocomputing, 1998, 21(1/2/3): 1–6.
- [18] ZHANG Hu, ZHOU Aimin, SONG Shenmin, et al. A self-organizing multiobjective evolutionary algorithm[J]. IEEE transactions on evolutionary computation, 2016, 20(5): 792–806.
- [19] HANSEN N, OSTERMEIER A. Completely derandomized self-adaptation in evolution strategies[J]. Evolutionary computation, 2001, 9(2): 159–195.
- [20] 李焕哲. 两阶段搜索的多峰优化算法研究 [D]. 武汉: 武汉大学, 2017.
- LI Huanzhe. Research on multimodal optimization algorithm with two-stage search[D]. Wuhan: Wuhan University, 2017.
- [21] 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. Matsushima, Japan: Springer, 2007: 36–50.
- [22] DEB K, TIWARI S. Omni-optimizer: a procedure for single and multi-objective optimization[C]//Proceedings of the 3rd International Conference on Evolutionary Multi-criterion Optimization, Third International Conference. Guanajuato, Mexico: Springer, 2006: 47–61.
- [23] WILCOXON F. Individual comparisons by ranking methods[J]. Biometrics bulletin, 1945, 1(6): 80–83.
- [24] FRIEDMAN M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance[J]. Journal of the American statistical association, 1937, 32(200): 675–701.

作者简介:



胡洁, 硕士研究生, 主要研究方向为多模态多目标优化。



范勤勤, 副教授, 博士生导师, 主要研究方向为多目标优化、机器学习、进化计算。发表学术论文 40 余篇。



王直欢, 高级工程师, 主要研究方向为大数据、进化计算、智能信息处理。