

文章编号: 2095-2163(2023)06-0019-11

中图分类号: TP301

文献标志码: A

基于社会学习粒子群的大规模多目标优化算法

刘能现

(福州大学 研究生院, 福州 350116)

摘要: 现实世界中存在大量多目标优化问题,其中大规模多目标优化问题是当前研究的热点,然而现有多目标进化算法缺少有效进化算子来处理大规模优化问题。因此,本文提出了一种基于社会学习粒子群的大规模多目标优化算法(A large-scale multi-objective algorithm based on a social learning particle swarm optimization algorithm, LMOSSLPSO)。LMOSSLPSO 首先采用转换的密度估计策略求解每个粒子的适应值;然后基于社会学习粒子群思想,设计了一种有效的粒子进化的方法;最后执行多目标优化算法 RVEA(a reference vector guided evolutionary algorithm)的环境选择操作来选择下一代个体。其中,转换的密度估计策略有利于平衡算法种群收敛性和多样性,新设计的粒子进化的方法有利于提高算法的搜索能力。在9个标准的大规模优化测试问题上,与多个近期提出的多目标优化算法进行对比。实验结果表明,该文所提出的 LMOSSLPSO 算法具有较好的收敛性及分布多样性。

关键词: 社会学习粒子群; 多目标优化; 大规模多目标优化算法

A large-scale multi-objective optimization algorithm based on a social learning particle swarm optimization algorithm

LIU Nengxian

(School of Graduate, Fuzhou University, Fuzhou 350116, China)

[Abstract] There are a lot of multi-objective optimization problems in the real world, among which large-scale multi-objective optimization problem is the current research hotspot. However, it is hard for existing multi-objective evolutionary algorithms to deal with large-scale multi-objective optimization problems due to the lack of effective evolutionary operators. Therefore, this paper proposes a large-scale multi-objective optimization algorithm based on the social learning particle swarm optimization algorithm (LMOSSLPSO). Firstly, LMOSSLPSO employs the shift-based density estimation (SDE) strategy to calculate the fitness value of each particle. Secondly, LMOSSLPSO designs an effective particle evolution operator based on the idea of the social learning particle swarm. Finally, LMOSSLPSO executes the environment selection operation of the RVEA algorithm to select individuals for the next generation. The SDE strategy is beneficial to balance the population convergence and diversity, and the newly designed evolution method is beneficial to improve the search ability of the algorithm. Compared with several recently proposed multi-objective algorithms on 9 large-scale optimization benchmark problems, the experimental results show that the proposed LMOSSLPSO algorithm has better convergence and diversity.

[Key words] social learning particle swarm optimization; multi-objective optimization problem; large-scale multi-objective optimization algorithm

0 引言

多目标优化问题(Multi-Objective Optimization Problems, MOPs)广泛存在于工程实践和科学研究中。如:社区检测^[1]、云工作流调度^[2]和车辆路径问题^[3]等。多目标优化问题通常含有2个及以上的目标,且这些目标之间相互冲突。过去的几十年,关

于多目标优化问题的研究取得了很大发展,研究人员提出了大量的多目标进化算法(Multi-Objective Evolutionary Algorithms, MOEAs),这些算法大致可以分为 Pareto 支配多目标算法(如 NSGA-II),基于分解的多目标算法(如 MOEA/D),基于指标的多目标算法(如 IBEA),及其它不属于前3类的算法(如 MOEA/DD)等等。近年来,大多数关于多目标进化

基金项目: 福建省自然科学基金面上项目(2022J01117)。

作者简介: 刘能现(1983-)男,博士,工程师,主要研究方向:智能计算及其应用。

通讯作者: 刘能现 Email: lylnx@fzu.edu.cn

收稿日期: 2022-12-04

算法的研究主要集中在高维多目标优化问题上,而对于决策变量较多的多目标优化问题关注较少。然而,现实世界中很多多目标优化问题可能有数百甚至数千个决策变量,这类问题被称为大规模多目标优化问题(Large-Scale MOP, LSMOP)^[4]。

通常,大规模 MOP 比决策变量少的 MOP 更难解决,其主要原因是 MOP 的搜索空间与决策变量的数量呈指数关系,即维度灾难问题,使得大多数现有多目标进化算法无法有效地探索搜索空间,并可能会过早地收敛到局部最优值或收敛到太大的区域^[5]。尽管大规模单目标优化问题多年来一直是热门研究课题,但大规模多目标优化的研究仍处于早期阶段。一般来说,现有的求解大规模多目标问题的进化算法可以大致分为 3 类^[6]:

(1) 基于决策变量分组算法

该类算法在求解大规模优化问题时使用分治策略,将决策变量随机或启发式地分成几组,然后交替优化每组决策变量。该策略已广泛用于解决大规模的单目标优化问题,但大规模的多目标优化问题包含多个相互冲突的目标,决策变量之间的相互关系更加复杂,在决策变量分组与优化时,应考虑总体的收敛性和多样性。现有 MOEA 中的决策变量分组技术主要包括随机分组、差分分组和变量分析。如:CCGDE3^[7]算法使用随机分组策略,将决策变量随机划分为大小相同的分组。虽然该算法在一些优化问题上取得了较满意的优化结果,但由于没有考虑变量之间的相互关系,在处理具有复杂变量关系的大规模多目标优化问题时效果较差。Ma 等人^[8]提出了一种基于决策变量分析的大规模多目标优化算法 MOEA/DVA。该算法根据收敛性和多样性把决策变量分成位置相关变量、距离相关变量和混合变量,在解决某些大规模多目标优化问题时具有较好的效果,但需要大量的适应度评估进行变量分析。Zhang 等人^[9]进一步扩展了 MOEA/DVA 算法,提出了 LMEA 算法。该算法根据角度,把决策变量分为收敛性相关决策变量和多样性相关决策变量。

(2) 基于决策空间缩减算法

该类算法用降维的思想,将父代的决策向量维数缩短并用于生成后代,然后将缩短的后代向量恢复到原始决策空间进行函数评估。因此,该类算法只需要找到一个短向量的最优值,而不是在高维决策空间中搜索。目前降维的技术主要包括问题转换、问题重构、随机嵌入、主成分分析等。如:Zille 等人^[10]提出了一种加权优化框架(WOF),该框架

对原优化问题通过变量分组和加权实施转换。其主要思想是采用分组策略将决策变量划分成几组,每组变量关联一个权重,即在同一组内的变量具有相同的权重,从而将大规模决策变量的优化转化为对维度较低权重向量的优化,实现对原搜索空间的降维,但该方法在重构解时较依赖于分组策略。He 等人^[11]提出采用问题重构优化框架 LSMOF,其核心思想是通过问题重构直接跟踪 Pareto 最优解。基于 LSMOF 的思想,Qin 等人^[12]提出了一种大规模多目标优化方法 LMOEA-DS,该方法在搜索方向上直接生成解。

(3) 基于新搜索策略算法

该类算法通过设计新颖搜索策略直接在原搜索空间生成后代,来求解大规模多目标优化问题。这类算法只需设计相对简单的操作就能在不同的大规模多目标优化问题取得较好的性能。新的搜索策略主要包括新的生成后代的操作算子、概率模型等。如:Tian 等人^[4]提出一种基于竞争学习粒子群的大规模多目标优化算法 LMOCSO。在该算法中,作者提出一种考虑速度和加速度的新策略。其中,失败粒子向获胜粒子学习,使失败粒子能更有效地向更好的位置移动。Cheng 等人^[13]提出了一种基于概率模型的大规模多目标优化算法 IM-MOEA,该算法通过构建基于高斯过程的逆模型,将解从目标空间映射到决策空间,并使用逆模型对目标空间进行采样来生成后代。

尽管现有的大规模多目标优化算法都有良好的性能,但每一类算法也有其自身的不足。如基于决策变量分组的方法,在求解复杂景观的多目标优化问题会遇到困难。具体来说,基于决策变量分组的 MOEA,可能会将两个相互关联的变量分到不同的组,因此在分别优化这两个变量时,算法很可能陷入局部最优。对于基于决策变量分析的 MOEA,虽然能够检测变量间的相互关系,但该操作需要大量的函数评估。基于决策空间缩减的算法,可以快速找到大规模多目标优化问题的一些局部最优解,但即使消耗更多的函数评估,其也可能无法找到全局最优解,这是因为转换后的问题很可能会丢失原始大规模多目标优化问题的全局最优解^[6]。基于新搜索策略的算法一般需要较多函数评估次数才能获得问题解^[12],但新搜索策略一般独立于选择策略,其可以比较容易地嵌入到许多 MOEA 中,以提高其在处理大规模多目标优化问题上的性能。

受基于竞争机制粒子群的大规模多目标优化算法的启发,本文基于社会学习的粒子群算法,提出了

一种处理大规模多目标优化问题的算法 LMOSLPSO。通过实验对比发现,该算法比其它几个大规模多目标进化算法有更好的性能表现。

1 相关概念

1.1 多目标优化问题

不失一般性,求最小值问题的多目标优化问题可描述如下:

$$\begin{aligned} \min \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{s.t.} \quad & g_p(\mathbf{x}) \leq 0 \quad (p = 1, 2, \dots, k) \\ & h_q(\mathbf{x}) = 0 \quad (q = 1, 2, \dots, l) \end{aligned} \quad (1)$$

其中, $\mathbf{x} = (x_1, x_2, \dots, x_D)$ 表示决策空间的 D 维向量; m 为目标个数; $f_i(x)$; $i = 1, 2, \dots, m$ 是 m 维目标空间中的目标函数; g_p 和 h_q 分别表示 k 个不等式约束和 l 个等式约束。

给定两个可行解 x_a 和 x_b , x_a 支配 x_b , 当且仅当对于 $\forall i, f_i(x_a) \leq f_i(x_b)$ 和 $\exists j, f_j(x_a) < f_j(x_b), i, j \in \{1, 2, \dots, m\}$ 。如果没有其他解支配 x^* , 那么 x^* 称为帕累托最优解。所有的帕累托最优解构成帕累托最优解集 (Pareto optimal Set, PS), 其目标值构成帕累托前沿 (Pareto Front, PF)。

带 2 个最小化目标的优化问题如图 1 所示。其中, x_3 支配 x_1 和 x_2 , x_1 和 x_2 之间互不支配, 蓝色曲线上的点为帕累托最优解, 蓝色曲线为帕累托前沿。

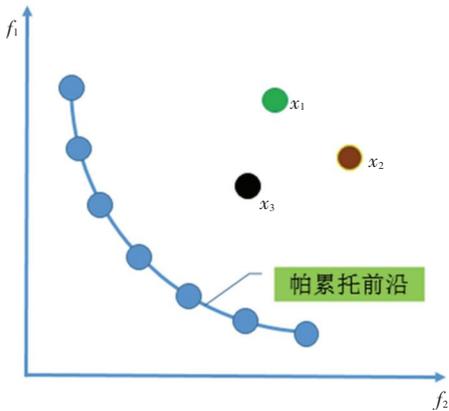


图 1 带 2 个最小化目标的优化问题

Fig. 1 Optimization problems with two minimization objectives

1.2 基于社会学习的粒子群算法

为求解单目标优化问题, Kennedy 等于 1995 年提出粒子群优化算法 (PSO), 该算法在求解决策变量规模较小的问题时有较好的效果。但是, 随着问题规模的增大, PSO 算法在搜索过程中较难平衡算法收敛性以及种群多样性, 而且粒子在算法搜索的后期容易早熟收敛, 陷入局部极值。为此, Cheng 等人^[14]在 2015 年提出了一种改进的粒子群算法来求

解大规模单目标优化问题, 并将其称为基于社会学习的粒子群算法 (SLPSO)。

在 SLPSO 算法中, 首先按适应值大小对种群中的粒子进行排序, 然后种群中除最优粒子外的其它粒子以一定的概率向其它较好的粒子及种群的平均位置学习。具体学习方式由公式 (2)、(3) 定义。

$$v_{i,j}(g+1) = r_{1,j}(g) \times v_{i,j}(g) + r_{2,j}(g) \times (I_{i,j}(g) + r_{3,j}(g) \times \varepsilon \times C_{i,j}(g)) \quad (2)$$

$$x_{i,j}(g+1) = x_{i,j}(g) + v_{i,j}(g+1) \quad (3)$$

其中:

$$I_{i,j}(g) = x_{k,j}(g) - x_{i,j}(g) \quad (4)$$

$$C_{i,j}(g) = \bar{x}_j(g) - x_{i,j}(g) \quad (5)$$

式中: v, x 和 g 分别表示粒子的速度、位置和进化代数, $r_{1,j}, r_{2,j}$ 和 $r_{3,j}$ 均为区间 $[0, 1]$ 内的随机数, $j = 1, 2, \dots, D, \varepsilon = \beta \times \frac{D}{M}$ 为社会影响因子, 用来调节算法的多样性和收敛性, 其中 $\beta = 0.01, M = 100, k$ 表示

粒子 i 在 j 维上学习粒子 (示范粒子), k 选择过程如图 2 所示, $\bar{x}_j(g) = \frac{\sum_{i=1}^{ps} x_{i,j}(g)}{ps}$ 为当前种群在第 j 维上的平均值, ps 为种群大小。

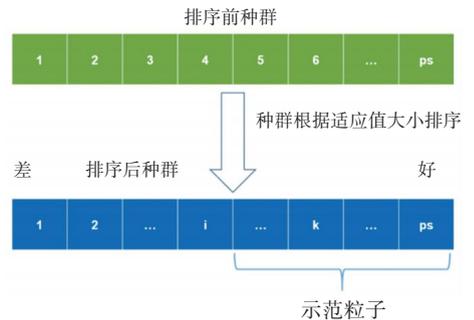


图 2 种群排序和选择示范粒子 k

Fig. 2 Population sorting and selection of demonstration particle k

2 基于社会学习粒子群的大规模多目标算法

2.1 算法框架

LMOSLPSO 算法采用了一个简洁的框架, 如算法 1 所示。算法首先执行初始化操作, 然后进行迭代进化循环操作, 最后输出最终非支配解。其主循环主要包括基于社会学习的粒子群进化和环境选择操作两部分。其中社会学习粒子群进化用来提高收敛性, 环境选择操作用来维持多样性。值得说明的是, LMOSLPSO 在进化过程中使用了与经典多目标优化算法 RVEA^[15] 相同的环境选择策略。LMOSLPSO 算法的框架如图 3 所示。

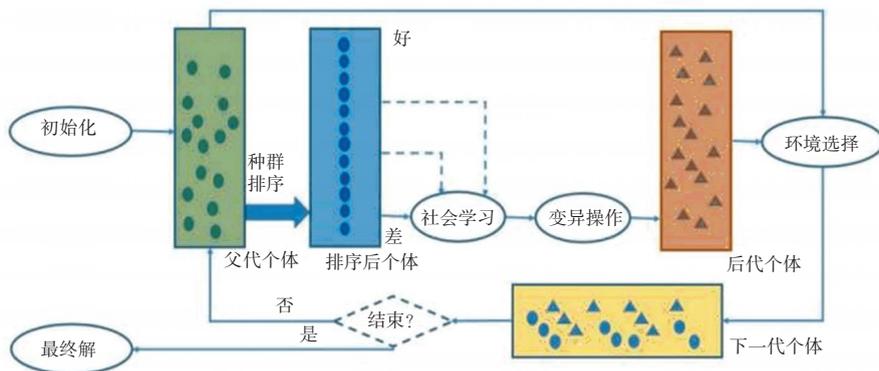


图3 LMOslPSO 算法的框架图

Fig. 3 The framework of LMOslPSO algorithm

算法1 LMOslPSO 框架

初始化:种群大小 ps , 问题维数 D , 最大函数评估次数 FES_{max} , 函数评估次数 FES , 随机初始化种群 P ;

while ($FES < FES_{max}$) do

$P' \leftarrow$ 基于社会学习的粒子群进化 (P);

$P \leftarrow$ 环境选择 (P, P');

end

输出 P 中的所有非支配个体

2.2 基于社会学习的粒子群进化

基于社会学习的粒子群进化主要包括粒子的适应值计算、种群 P 的排序及学习概率设置、粒子的速度和位置更新,以及所有粒子的多项式变异操作等。基于社会学习的粒子群进化操作的伪代码如算法2所示。

算法2 基于社会学习的粒子群进化 (P)

输入 当前种群 P ;

$P' \leftarrow \emptyset$;

利用公式(6)计算种群 P 中每个粒子的适应值;

对种群 P 按适应值从大到小(好到差)进行排序;

按公式(7)计算种群 P 中每个粒子学习概率;

for 除最优粒子外的每个粒子 x_i in P do

if ($lp_i > rand$) then //学习率大于随机数

$\{k, l\} \leftarrow$ 从 $[1, i]$ 中随机选择两个粒子;

使用公式(8)更新粒子速度 v_i ;

使用公式(3)更新粒子位置 x_i ;

end if

end for

对所有粒子执行变异操作并放入后代种群 P'

输出 后代种群 P'

首先,进行粒子的适应值计算,与单目标优化问题不同,因多目标优化问题含有多个目标且各个目标之间相互冲突,无法采用单个目标值直接进行粒子优劣的比较,为此需要一个能够评价粒子优劣的方法。本文采用转换的密度估计策略(Shift-based Density Estimation, SDE)^[16]来求解每个粒子的适应值。SDE策略能够从多样性和收敛性两个方面进行粒子的质量评价,并已被多个MOEAs所采用^[4]。因此,在LMOslPSO算法中,本文也采用了基于SDE的策略来评估种群中每个个体的收敛性和多样性。具体来说,用最小的基于SDE的距离公式(6)^[16]来定义粒子 x 的适应值。

$$Fitness(x) = \min_{q \in P \setminus \{x\}} \sqrt{\sum_{i=1}^m (\max\{0, f_i(q) - f_i(x)\})^2} \quad (6)$$

按社会学习粒子群的思想,对种群 P 中的粒子按适应值从大到小进行排序,并按公式(7)计算每个粒子的学习率 lp_i , 即为粒子排序值 i 除以种群数 ps , 表示越差的粒子越有机会向好的粒子学习。

$$lp_i = i/ps \quad (7)$$

接着对每个粒子按学习概率进行速度更新和位置更新。具体来说,随机产生0到1之间随机数 $rand$, 当学习率大于随机数 $rand$ 时,从种群中随机选择比粒子 i 具有更好适应值的两个粒子 k, l , 并按公式(8)更新粒子速度和按公式(3)更新粒子位置。本文的速度更新公式(8)与公式(2)不同,由于在多目标优化中,使用相同的种群平均位置引导种群会使种群失去多样性,因此本文利用了两个适应值较好的粒子来引导种群进化。

$$v_i(g+1) = r_1(g) \times v_i(g) + r_2(g) \times (x_k(g) - x_i(g)) + r_3(g) \times (x_l(g) - x_i(g)) \quad (8)$$

最后,为进一步提高LMOslPSO算法性能,避

免算法陷入局部极值,对种群中的每个粒子执行多项式变异操作^[17]。在基于社会学习的粒子群进化结束后返回到算法1,执行多目标环境选择操作。

3 实验结果与分析

3.1 对比算法及测试函数

为验证 LMOSLPSO 算法的效果,将其算法与多个经典的多目标算法进行对比。其中包括:LMEA^[9]、IM-MOEA^[13]、MMOPSO^[18]和 LMOCSO^[4]。

实验采用大规模多目标测试集^[19]作为测试问题,即 LSMOP1-LSMOP9。LSMOP 测试集是在大规模单目标问题基础上设计的,该测试集的问题具有较好的可扩展性和普遍性。在这9个 LSMOP 问题中,在 Pareto 集上存在线性变量连接(如 LSMOP1-LSMOP4)和非线性变量连接(如 LSMOP5-LSMOP),以及线性 Pareto 前沿(如 LSMOP1-LSMOP4)、凹 Pareto 前沿(如 LSMOP5-LSMOP8),和非连续 Pareto 前沿(如 LSMOP9)。

3.2 评价指标

为了比较不同算法的性能,本文采用反转世代距离(Inverted Generational Distance, *IGD*)作为性能指标,来评价不同算法的实验结果。假设目标空间中最优 Pareto 前沿面上均匀分布的点集合为 P^* ,算法所求得的非支配解集为 P ,则 *IGD* 的计算公式定义为

$$IGD(P, P^*) = \frac{\sum_{x \in P^*} \min_{y \in P} dist(x, y)}{|P^*|} \quad (9)$$

其中, $dist(x, y)$ 表示点 x 到点 y 之间的欧氏距离。因此,反转世代距离 *IGD* 是从最优前沿面 P^* 中的每个点到支配解集 P 的平均最小距离,其测量的是支配解集 P 的收敛性和均匀性。*IGD* 值越小,说明对应算法的综合性能越好。同时,采用显著性水平为 0.05 的威尔科克森符号秩检验对不同对比算法进行比较。

3.3 实验设置

本文实验采用 Tian 等人^[20]提出的 PlatEMO 平台,所有算法采用 Matlab 语言编程。实验所用的电脑配置:中央处理器采用 Inter(R) Core(TM) i5-4590 CPU @ 3.30 GHz,内存 8.00 GB,操作系统 Windows 7。

相关算法的参数参考文献^[4]设置,所有对比算法种群大小设置相同;对于2个目标的问题种群大小设置为300,3个目标的问题种群大小设置为496。对于 LSMOP1~LSMOP9 问题,每个变量组的

子分量数量 n_k 设置为5,目标数 M 设置为2和3,决策变量维数 D 的大小从100到500不等。最大函数评估次数作为所有对比算法的终止条件,设置为 $15\,000 \times D$ 。

3.4 实验结果

不同算法在2维多目标问题 LSMOP1-LSMOP9 函数上 *IGD* 值的平均值和标准差详见表1,最好的 *IGD* 值用蓝色粗体字体标出。从表1中可见,本文提出的 LMOSLPSO 算法对比其他4种方法获得了更好的优化结果。所提出的 LMOSLPSO 算法在27个测试问题中获得了17个最优平均 *IGD* 值。对比算法 LMEA、IMMOEA、MMOPSO 和 LMOCSO 分别在0、5、1和4个测试问题上获得了最优平均 *IGD* 值。具体来说,与 LMEA 相比,本文提出的 LMOSLPSO 算法在27个测试问题中的22(5)个测试问题上表现更好(相似)。与 IMMOEA 相比,LMOSLPSO 算法在27个测试问题中的17(4)个测试问题上表现更好(相似)。与 MMOPSO 相比,LMOSLPSO 算法在27个测试问题中的21(1)个测试问题上表现更好(相似)。与 LMOCSO 相比,LMOSLPSO 算法在27个测试问题中的19(3)个测试问题上表现更好(相似)。LMOSLPSO 算法具有更好的性能主要是因为其使用了基于社会学习的粒子群进化操作。

表2给出了所有对比多目标算法在3维多目标问题 LSMOP1~LSMOP9 函数上的 *IGD* 值的平均值和标准差,最好的 *IGD* 值用蓝色字体标出。从表2中我们同样可以清楚地看出,本文提出的 LMOSLPSO 算法比4种对比方法获得了更好的优化结果。所提出的 LMOSLPSO 算法在27个测试问题中的13个上获得了最优平均 *IGD* 值,对比算法 LMEA、IMMOEA、MMOPSO 和 LMOCSO 分别在0、4、1和9个测试问题上获得了最优平均 *IGD* 值。具体来说,与 LMEA 相比,本文提出的 LMOSLPSO 算法在27个测试问题中的25(1)个测试问题上表现更好(相似)。与 IMMOEA 相比,LMOSLPSO 算法在27个测试问题中的21(3)个测试问题上表现更好(相似)。与 MMOPSO 相比,LMOSLPSO 算法在27个测试问题中的22(5)个测试问题上表现更好(相似)。与 LMOCSO 相比,LMOSLPSO 算法在27个测试问题中的11(10)个测试问题上表现更好(相似)。在3维多目标问题中,对比算法 LMEA、IMMOEA 和 MMOPSO 的对比性能有所下降,对比算法 LMOCSO 的对比性能有所提高,但 LMOCSO 的性能还是比本文提出的 LMOSLPSO 算法的性能差。

表1 不同算法在2维LSMOP1-LSMOP9上得到IGD的平均值和标准差比较

Tab. 1 Comparison of the mean and standard deviation of IGD obtained by different algorithms on 2-dimensional lsmop1-lsmop9

问题	维数	LMEA	IMMOEA	MMOPSO	LMOCSO	LMOSLPSO
LSMOP1	100	1.885 1e-1 (2.38e-1) -	5.674 3e-3 (2.83e-3) -	1.623 0e-2 (7.27e-3) -	1.205 0e-3 (3.45e-6) -	1.189 0e-3 (1.79e-6)
LSMOP1	200	1.779 0e-1 (2.57e-1) -	6.612 7e-3 (4.28e-4) -	5.462 7e-3 (3.70e-3) -	1.201 9e-3 (8.20e-6) -	1.189 1e-3 (1.03e-6)
LSMOP1	500	1.135 4e-1 (1.53e-1) -	1.156 5e-2 (2.38e-4) -	8.357 6e-3 (1.59e-3) -	1.198 3e-3 (2.20e-6) -	1.190 3e-3 (8.19e-7)
LSMOP2	100	1.448 6e-1 (5.09e-2) -	8.816 1e-3 (3.66e-4) -	3.030 0e-2 (1.41e-2) -	5.787 5e-3 (4.57e-3) -	5.462 5e-3 (1.42e-3)
LSMOP2	200	1.010 4e-1 (3.49e-2) -	7.762 3e-3 (2.15e-4) -	3.379 4e-2 (1.62e-2) -	1.417 0e-2 (1.48e-2) -	2.904 8e-3 (1.24e-3)
LSMOP2	500	5.720 1e-2 (8.12e-3) -	6.782 3e-3 (1.38e-4) -	3.571 1e-2 (9.94e-4) -	1.465 6e-2 (7.36e-3) -	3.486 8e-3 (3.19e-3)
LSMOP3	100	3.369 4e+0 (3.70e+0)	5.279 4e-1 (3.70e-2)	1.794 4e+0 (8.32e-1) -	6.293 9e-1 (1.16e-1) -	4.324 0e-1 (2.46e-1)
LSMOP3	200	3.376 3e+0 (4.55e+0)	5.179 7e-1 (2.99e-2)	2.483 7e+0 (1.02e+0)	7.016 6e-1 (2.18e-2) -	4.716 3e-1 (2.01e-1)
LSMOP3	500	3.419 6e+0 (5.34e+0) -	5.134 9e-1 (2.11e-2)+	1.922 1e+0 (3.68e-1) -	7.071 0e-1 (4.00e-5) -	6.079 5e-1 (1.45e-1)
LSMOP4	100	1.183 0e-1 (4.39e-2) -	1.280 1e-2 (7.98e-4) -	3.313 5e-2 (2.45e-3) -	8.114 5e-3 (2.71e-3) -	2.056 0e-3 (6.67e-4)
LSMOP4	200	6.374 1e-2 (2.42e-2) -	8.483 2e-3 (3.11e-4) -	2.791 8e-2 (7.43e-3) -	2.348 3e-3 (1.13e-3) -	1.424 2e-3 (1.25e-4)
LSMOP4	500	3.798 1e-2 (1.24e-2) -	6.939 2e-3 (1.99e-4) -	2.741 0e-2 (4.23e-3) -	1.316 7e-3 (3.22e-5) -	1.265 7e-3 (3.26e-5)
LSMOP5	100	3.941 0e-1 (2.53e-1) -	4.947 5e-2 (2.25e-2) -	3.774 4e-3 (8.12e-4) -	3.690 6e-1 (1.01e-1) -	1.338 9e-3 (6.88e-6)
LSMOP5	200	4.598 9e-1 (2.39e-1) -	3.636 0e-2 (1.52e-2) -	4.512 1e-3 (1.40e-3) -	1.605 4e-1 (1.73e-1) -	1.343 9e-3 (5.14e-6)
LSMOP5	500	3.580 2e-1 (1.59e-1) -	2.688 7e-2 (1.54e-2) -	5.274 1e-3 (1.45e-3) -	1.430 5e-3 (2.11e-5) -	1.359 8e-3 (9.93e-6)
LSMOP6	100	8.201 8e-1 (1.90e-1) -	3.607 8e-1 (7.22e-2)	4.395 4e-1 (3.72e-2) +	7.172 3e-1 (9.69e-2)	7.421 9e-1 (2.67e-5)
LSMOP6	200	6.875 2e-1 (1.73e-1) =	3.496 5e-1 (8.64e-2) +	3.971 9e-1 (3.70e-2) +	2.872 1e-1 (3.29e-2) +	6.978 9e-1 (1.35e-1)
LSMOP6	500	6.121 9e-1 (1.79e-1) =	2.732 9e-1 (7.18e-2) +	3.215 5e-1 (4.16e-2) +	5.553 4e-1 (1.41e-1) +	7.025 7e-1 (1.04e-1)
LSMOP7	100	1.619 1e+0 (1.29e+0) -	8.294 2e-1 (1.65e-1) +	1.596 9e+0 (5.75e-1) -	1.270 5e+0 (2.42e-1) -	1.048 6e+0 (3.38e-1)

续表 1

问题	维数	LMEA	IMMOEA	MMOPSO	LMOCSO	LMOSLPSO	
LSMOP7	200	1.281 0e+0 (2.82e-1) =	1.128 1e+0 (2.47e-1) =	2.095 7e+0 (7.22e-1) -	1.201 9e+0 (4.03e-1) =	1.284 3e+0 (4.58e-1)	
LSMOP7	500	1.386 6e+0 (3.32e-1) -	1.324 1e+0 (2.49e-1) -	1.365 0e+0 (3.88e-1) -	7.319 0e-1 (5.00e-1) =	9.225 7e-1 (6.25e-1)	
LSMOP8	100	8.808 7e-2 (2.96e-2) -	2.249 2e-2 (3.17e-3) -	4.215 6e-2 (7.32e-3) -	3.470 3e-1 (2.12e-3) -	6.073 4e-3 (3.21e-3)	
LSMOP8	200	6.186 3e-2 (8.53e-3) -	2.859 6e-2 (4.39e-3) -	4.733 3e-2 (3.26e-3) -	1.530 7e-1 (1.71e-1) -	5.913 2e-3 (2.83e-3)	
LSMOP8	500	4.188 5e-2 (5.80e-3) -	2.352 2e-2 (3.24e-3) -	2.663 6e-2 (1.44e-3) -	2.298 6e-3 (8.21e-4) -	1.851 1e-3 (3.70e-4)	
LSMOP9	100	6.852 6e-1 (3.47e-1) =	6.131 1e-1 (2.86e-1) +	4.720 7e-2 (4.84e-2) +	7.594 0e-2 (1.03e-1) +	6.144 7e-1 (3.10e-1)	
LSMOP9	200	6.482 9e-1 (2.60e-1) =	5.694 5e-1 (3.56e-1) =	3.022 6e-1 (4.33e-2) +	6.283 6e-2 (1.31e-1) +	5.804 0e-1 (2.86e-1)	
LSMOP9	500	5.797 7e-1 (2.09e-1) -	4.787 0e-1 (2.99e-1) -	2.041 9e-1 (4.12e-3) =	2.502 7e-3 (2.17e-5) =	2.157 4e-1 (2.85e-1)	
		+/-/=	0/22/5	6/17/4	5/21/1	5/19/3	-/-/-

注：“-”、“=”和“+”分别表示所提出的 LMOSLPSO 算法对比算法“较好”、“相似”和“较差”。

表 2 不同算法在 3 维 LSMOP1~LSMOP9 上得到的 IGD 平均值和标准差比较

Tab. 2 Comparison of IGD mean and standard deviation obtained by different algorithms on 3-d lsmop1~lsmop9

问题	维数	LMEA	IMMOEA	MMOPSO	LMOCSO	LMOSLPSO
LSMOP1	100	6.606 2e-2 (6.34e-2) -	1.073 7e-1 (5.55e-3) -	1.152 3e-1 (1.35e-2) -	1.781 8e-2 (3.28e-4) =	1.769 2e-2 (1.39e-4)
LSMOP1	200	9.948 8e-2 (1.07e-1) -	1.371 1e-1 (7.97e-3) -	1.293 8e-1 (1.05e-2) -	1.831 2e-2 (4.91e-4) -	1.769 5e-2 (2.42e-4)
LSMOP1	500	6.628 7e-2 (5.44e-2) -	1.623 1e-1 (1.18e-2) -	1.679 9e-1 (9.27e-3) -	1.882 1e-2 (6.01e-4) -	1.790 7e-2 (2.68e-4)
LSMOP2	100	7.754 7e-2 (4.78e-2) -	9.740 2e-2 (2.60e-3) -	1.530 2e-1 (4.73e-3) -	3.659 9e-2 (1.28e-3) +	3.841 6e-2 (1.37e-3)
LSMOP2	200	9.347 0e-2 (6.63e-2) -	7.951 6e-2 (1.86e-3) -	1.074 2e-1 (3.00e-3) -	3.374 7e-2 (1.06e-3) =	3.421 1e-2 (2.13e-3)
LSMOP2	500	8.410 0e-2 (9.56e-2) -	4.988 0e-2 (9.16e-4) -	5.705 9e-2 (1.09e-3) -	2.420 3e-2 (1.32e-3) -	2.284 0e-2 (1.07e-3)
LSMOP3	100	8.511 1e-1 (8.78e-2) -	4.774 5e-1 (2.07e-2) -	4.618 4e-1 (1.41e-1) -	3.316 2e-1 (3.42e-2) -	3.123 8e-1 (3.34e-2)
LSMOP3	200	8.772 9e-1 (9.35e-2) -	5.352 5e-1 (2.41e-2) -	3.347 8e-1 (9.33e-2) =	3.989 1e-1 (2.93e-2) -	3.774 2e-1 (3.72e-2)
LSMOP3	500	8.713 0e-1 (6.23e-2) -	6.167 3e-1 (4.42e-2) -	5.316 7e-1 (6.59e-2) -	6.116 9e-1 (1.90e-1) -	4.047 0e-1 (2.64e-2)
LSMOP4	100	1.589 5e-1 (6.75e-2) -	1.739 4e-1 (1.21e-2) -	1.331 3e-1 (1.16e-2) -	4.334 0e-2 (1.06e-2) =	3.975 7e-2 (4.70e-3)

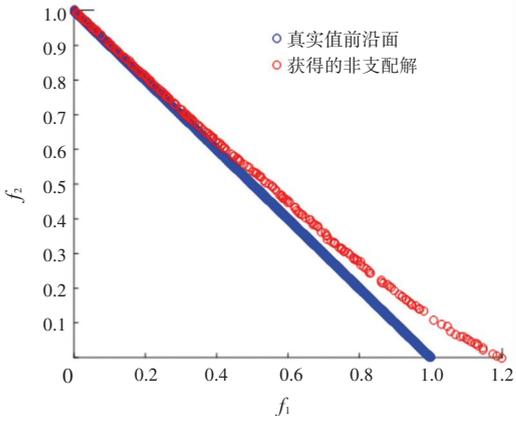
续表 2

问题	维数	LMEA	IMMOEA	MMOPSO	LMOCSO	LMOSLPSO
LSMOP4	200	9.366 2e-2 (3.55e-2) -	1.396 3e-1 (7.48e-3) -	1.559 2e-1 (1.88e-2) -	3.895 5e-2 (5.32e-3) =	3.808 4e-2 (4.96e-3)
LSMOP4	500	6.295 7e-2 (5.25e-2) -	9.168 6e-2 (2.71e-3) -	1.639 2e-1 (5.78e-3) -	3.724 2e-2 (6.88e-3) -	2.926 5e-2 (2.98e-3)
LSMOP5	100	1.651 9e+0 (1.73e+0) -	1.451 2e-1 (3.77e-2) -	5.798 1e-2 (1.91e-2) -	2.225 1e-2 (5.45e-5) -	2.220 4e-2 (5.54e-5)
LSMOP5	200	3.297 5e+0 (3.13e+0) -	1.494 4e-1 (2.50e-2) -	7.001 5e-2 (1.10e-2) -	2.232 0e-2 (6.03e-5) -	2.220 4e-2 (4.82e-5)
LSMOP5	500	5.371 7e+0 (4.23e+0) -	1.875 5e-1 (1.33e-2) -	2.278 0e-1 (6.28e-3) -	2.237 4e-2 (1.35e-4) -	2.221 8e-2 (4.68e-5)
LSMOP6	100	1.216 3e+2 (3.36e+2) -	5.451 2e-1 (8.18e-2) +	7.978 0e-1 (1.58e-1) -	6.494 4e-1 (9.60e-2) =	6.565 8e-1 (9.56e-2)
LSMOP6	200	3.625 8e+2 (8.36e+2) -	7.599 5e-1 (9.42e-2) +	1.241 3e+0 (1.23e-1) -	9.947 3e-1 (1.24e-1) =	9.804 9e-1 (1.84e-1)
LSMOP6	500	2.331 4e+3 (4.66e+3) -	1.134 2e+0 (2.23e-1) =	1.087 5e+0 (2.51e-1) =	1.058 2e+0 (4.01e-1) =	1.193 0e+0 (3.33e-1)
LSMOP7	100	2.259 1e+0 (5.28e-1) -	6.111 8e-1 (4.59e-2) -	6.393 9e-1 (1.81e-1) -	5.468 6e-1 (2.06e-1) =	5.540 5e-1 (2.43e-1)
LSMOP7	200	1.601 9e+0 (2.26e-1) -	6.061 4e-1 (3.62e-2) =	8.107 4e-1 (2.48e-1) =	7.923 5e-1 (2.22e-1) =	7.450 8e-1 (2.53e-1)
LSMOP7	500	9.345 4e-1 (7.70e-2) +	5.971 9e-1 (4.45e-2) +	8.419 6e-1 (1.43e-1) =	9.315 8e-1 (7.86e-2) =	9.459 3e-1 (2.24e-7)
LSMOP8	100	1.623 0e-1 (7.37e-2) -	1.418 3e-1 (6.52e-3) -	1.348 5e-1 (1.51e-2) -	3.837 5e-2 (4.47e-3) +	4.258 7e-2 (7.20e-3)
LSMOP8	200	1.447 7e-1 (1.97e-2) -	1.420 4e-1 (6.96e-3) -	1.223 0e-1 (4.04e-3) -	4.238 3e-2 (4.18e-3) +	4.749 7e-2 (4.39e-3)
LSMOP8	500	1.099 0e-1 (1.35e-2) -	1.165 7e-1 (1.21e-2) -	7.702 4e-2 (2.11e-3) -	4.451 7e-2 (2.37e-3) -	4.036 5e-2 (4.51e-3)
LSMOP9	100	5.796 0e-1 (1.77e-1) =	5.850 6e-1 (1.18e-3) =	9.836 1e-1 (1.27e-1) -	2.306 8e-1 (2.84e-1) +	6.916 3e-1 (3.74e-1)
LSMOP9	200	6.384 0e-1 (2.85e-1) -	5.797 3e-1 (2.64e-3) -	6.584 2e-1 (2.94e-2) -	1.929 6e-1 (1.48e-1) +	2.921 3e-1 (1.20e-1)
LSMOP9	500	6.308 3e-1 (3.82e-1) -	5.651 6e-1 (5.29e-3) -	3.965 5e-1 (1.40e-2) =	1.117 6e-1 (1.01e-1) +	3.205 6e-1 (1.62e-1)
	+/-/=	1/25/1	3/21/3	0/22/5	6/11/10	-/-/-

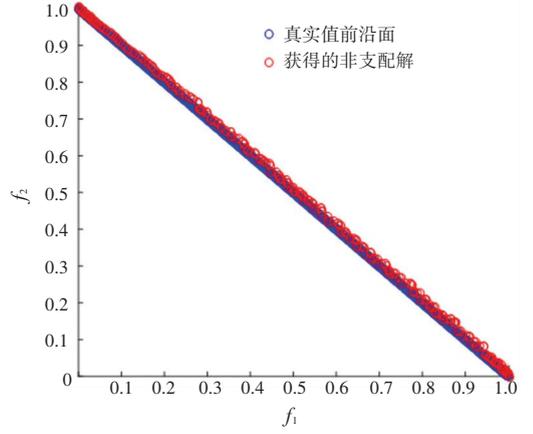
注：“-”、“=”和“+”分别表示所提出的 LMOSLPSO 算法对比算法“较好”、“相似”和“较差”。

图 4 和图 5 分别给出了所有对比多目标算法在 2 维和 3 维大规模多目标优化问题 LSMOP4 上获得的非支配解。由图中可以看出,对比算法 LMEA、IMMOEA 和 MMOPSO 获得的非支配解均匀性和收敛性较差。从图 4 可以看出,在 2 维多目标优化问

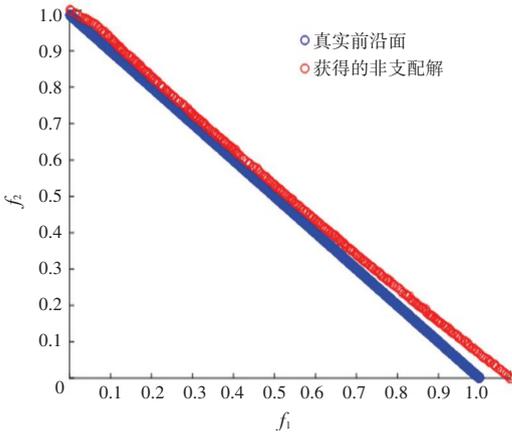
题 LSMOP4 上,对比算法 LMOCSO 和提出算法 LMOSLPSO 均匀性和多样性都较好;从图 5 可以看出,在 3 维多目标优化问题 LSMOP4 上,本文算法 LMOSLPSO 在均匀性上比算法 LMOCSO 更佳。



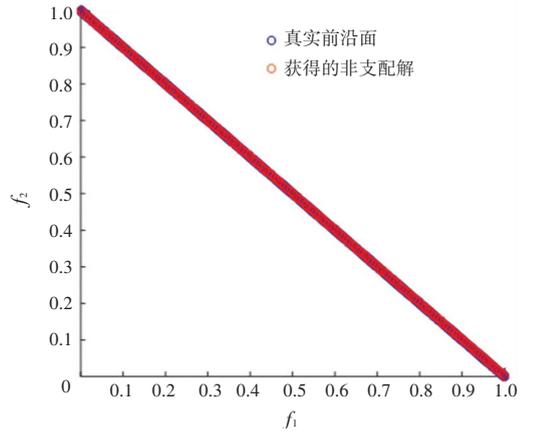
(a) LMEA



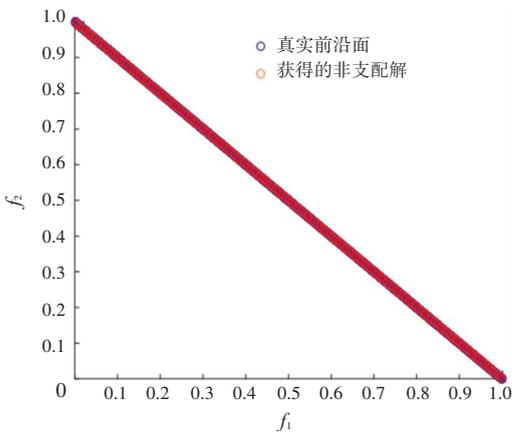
(b) IMMOEA



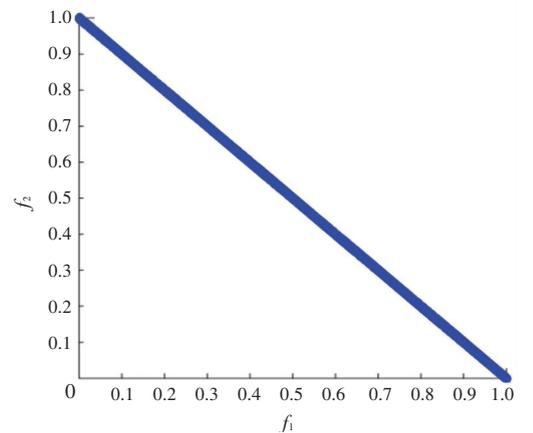
(c) MMOPSO



(d) LMOCSO



(e) LMOSLPSO



(f) 真实前沿面

图 4 不同算法在双目标 LSMOP4 函数上获得的非支配解

Fig. 4 The non-dominated solution obtained by different algorithms on the bi-objective LSMOP4 function

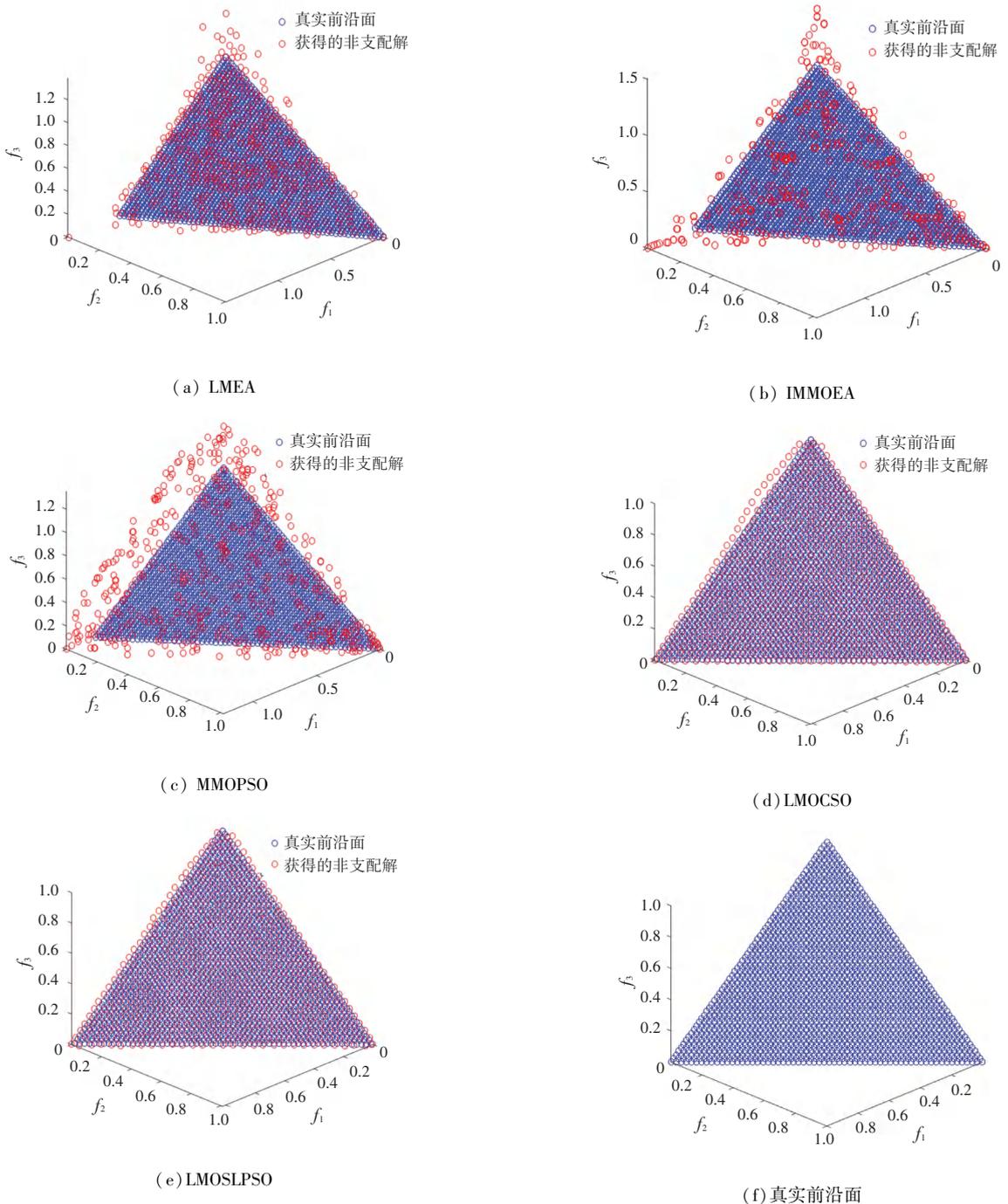


图5 不同算法在3目标LSMOP4函数上获得的非支配解

Fig. 5 The non-dominated solution obtained by different algorithms on the 3-objective LSMOP4 function

综上所述,从表1、表2以及图4、图5可以得出,本文提出的LMOslPSO算法比其它算法在求解大规模多目标优化问题上有较好的性能表现。

4 结束语

本文基于社会学习粒子群算法思想,提出了一种用于求解大规模多目标优化问题的多目标粒子群算法(LMOslPSO)。该算法利用转换的密度估计策略求解每个粒子的适应值,然后基于适应值排序设

计了一种有效的粒子速度更新方法。该方法能有效引导种群进化,使所提出的LMOslPSO算法获得了综合性能较优的优化结果。实验结果表明,LMOslPSO算法比对其他算法有较好的性能表现,该算法是处理大规模优化问题的有效算法。

参考文献

- [1] LIU C, LIU J, JIANG Z. A multiobjective evolutionary algorithm based on similarity for community detection from signed social

- networks[J]. IEEE Transactions on Cybernetics, 2014, 44(12): 2274–2287.
- [2] CHEN Z G, ZHAN Z H, LIN Y, et al. Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach[J]. IEEE Transactions on Cybernetics, 2019, 49(8): 2912–2926.
- [3] DUAN J, HE Z, YEN G G. Robust multiobjective optimization for vehicle routing problem with time windows [J]. IEEE Transactions on Cybernetics, 2021.
- [4] TIAN Y, ZHENG X, ZHANG X, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer[J]. IEEE Transactions on Cybernetics, 2020, 50(8): 3696–3708.
- [5] OLDEWAGE E T, ENGELBRECHT A P, CLEGHORN C W. The merits of velocity clamping particle swarm optimisation in high dimensional spaces [J]. 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 – Proceedings, 2018: 1–8.
- [6] TIAN Y, SI L, ZHANG X, et al. Evolutionary large-scale multi-objective optimization: A survey[J]. ACM Computing Surveys, 2022, 54(8): 1–34.
- [7] ANTONIO L M, COELLO C A C. Use of cooperative coevolution for solving large scale multiobjective optimization problems [J]. 2013 IEEE Congress on Evolutionary Computation, CEC 2013, 2013: 2758–2765.
- [8] MA X, LIU F, QI Y, et al. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(2): 275–298.
- [9] ZHANG X, TIAN Y, CHENG R, et al. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2018, 22(1): 97–112.
- [10] ZILLE H, ISHIBUCHI H, MOSTAGHIM S, et al. A framework for large-scale multiobjective optimization based on problem transformation [J]. IEEE Transactions on Evolutionary Computation, 2018, 22(2): 260–275.
- [11] HE C, LI L, TIAN Y, et al. Accelerating large-scale multi-objective optimization via problem reformulation [J]. IEEE Transactions on Evolutionary Computation, 2019, PP (19324779): 949–961.
- [12] QIN S, SUN C, JIN Y, et al. Large-scale evolutionary multiobjective optimization assisted by directed sampling [J]. IEEE Transactions on Evolutionary Computation, 2021, 25(4): 724–738.
- [13] CHENG R, JIN Y, NARUKAWA K, et al. A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling [J]. IEEE Transactions on Evolutionary Computation, 2015, 19(6): 838–856.
- [14] CHENG R, JIN Y. A social learning particle swarm optimization algorithm for scalable optimization [J]. Information Sciences, Elsevier, 2015, 291: 43–60.
- [15] CHENG R, JIN Y, OLHOFER M, et al. A reference vector guided evolutionary algorithm for many-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 773–791.
- [16] LI M, YANG S, LIU X. Shift-based density estimation for pareto-based algorithms in many-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 348–365.
- [17] DEB K, GOYAL M. A combined genetic adaptive search (GeneAS) for engineering design [J]. Computer Science and Informatics, 1996, 26(1): 30–45.
- [18] LIN Q, LI J, DU Z, et al. A novel multi-objective particle swarm optimization with multiple search strategies [J]. European Journal of Operational Research, 2015, 247(3): 732–744.
- [19] CHENG R, JIN Y, OLHOFER M, et al. Test problems for large-scale multiobjective and many-objective optimization [J]. IEEE transactions on cybernetics, 2017, 47(12): 4108–4121.
- [20] TIAN Y, CHENG R, ZHANG X, et al. PlatEMO: A matlab platform for evolutionary multi-objective optimization [J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73–87.

(上接第18页)

- [11] KWAPISZ J R, WEISS G M, MOORE S A. Cell Phone based biometric identification [C]// IEEE Fourth International Conference on Biometrics, Los Angeles, 2010: 1–7.
- [12] 张丽娜. 基于加速度传感器的步态特征身份认证 [D]. 沈阳: 沈阳工业大学, 2014.
- [13] SEO J, MA H, SAHA T K. On Savitzky-Golay filtering for online condition monitoring of transformer on load tap changer [J]. IEEE Transactions on Power Delivery, 2017, 33(4): 1689–1698.
- [14] 张云华. 统计学中四分位数的计算 [J]. 中国高新技术企业, 2009(20): 173–174.
- [15] 段小虎, 蒋刚, 留沧海. 基于手机加速度传感器的人体步态识别研究 [J]. 传感器与微系统, 2021, 40(1): 30–33, 38.
- [16] 汪涛, 汪泓章, 夏懿, 等. 基于卷积神经网络与注意力模型的人体步态识别 [J]. 传感技术学报, 2019, 32(7): 1027–1033.
- [17] 孙宇航, 周建钦, 张学锋. 基于加速度传感器的人体运动模式识别 [J]. 计算机系统应用, 2020, 29(6): 196–203.
- [18] ZHOU Z H, FENG J. Deep forest [J]. National Science Review, 2019, 6(1): 74–86.
- [19] BETHAPUDI P, REDDY E S, VARMA K V. Classification of breast cancer using Gini index based fuzzy supervised learning in quest decision tree algorithm [J]. International Journal of Computer Applications, 2015, 111(14): 50–57.