

DOI: 10.20098/j.cnki.1001-5132.2023.0240

基于方向向量采样的大规模多目标进化优化

熊英剑, 史旭华*

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 针对在高维决策空间的大规模多目标优化中难以搜索产生有希望的后代个体, 从而导致常规算法性能恶化的问题, 提出了一种基于方向向量采样的大规模多目标进化算法. 该算法首先选取一些接近理想点的优秀候选个体, 然后根据候选个体分阶段创建方向向量, 最后执行定向采样策略(收敛性相关采样策略和多样性相关采样策略)来生成子代个体, 由此生成的子代个体性能优秀, 且算法收敛加速. 实验结果表明, 与5种具有代表性的大规模多目标进化算法相比, 该算法在多达2000个决策变量的大规模多目标优化测试中, 仍具有较强的寻优能力.

关键词: 高维; 大规模多目标; 方向向量; 定向采样

中图分类号: TP301.6

文献标志码: A

文章编号: 1001-5132 (2024) 02-0001-09

最近在多目标优化领域, 研究者将目光聚焦在决策变量维度高的大规模多目标优化问题上. 常规多目标进化算法^[1-5]存在随着决策变量维度增加, 算法搜索空间呈指数增加、搜索效率快速下降、难以在较大决策空间产生有前途子代解, 从而导致维度诅咒^[6]. 为了解决这些问题, 相关学者提出了诸多大规模多目标进化算法(LSMOEAs). 如第三代协同进化差分进化算法(CCGDE3)^[7], 基于随机分组策略将决策变量分为等长的多个子种群, 然后使用差分进化算法(GDE3)^[8]对每个子种群分别进行优化. 基于分解方法解决大规模多目标问题进化算法(MOEA/D2)^[9], 将一种基于分解的多目标进化算法(MOEA/D)^[10]与协同进化技术相结合, 使算法能够同时对决策空间和目标空间进行分解. 然而这些算法的性能依赖决策变量的分组, 当决策变量分组不当时, 算法性能会出现恶化. 为了精确分组决策变量, 有学者提出了基于决策变量分析的多目标进化算法(MOEA/DVA)^[11], 该算法使用决策变量分析, 将决策变量分为收敛性变量、多样性变量和混合变量, 并分别进行优化. 聚类的大规

模多目标进化算法(LMEA)^[12], 将决策变量分为收敛性变量和多样性变量, 分别采用2种不同策略对变量组进行优化. 但这需要在优化前进行大量的函数评价, 且对一些难以分离决策变量的优化问题, 无法有效提升算法性能. 相较于对决策变量进行分组, 有些算法则利用权值对大规模多目标优化问题进行转化降维, 如基于问题转换框架的大规模多目标进化算法(WOF)^[13], 将决策变量关联权向量进行优化; 基于问题重构策略加速大规模多目标优化(LSMOF)^[14], 在决策空间构建双向参考向量, 然后将其与一组权重变量相关联. 虽然降维能够加速收敛, 但容易陷入局部最优而导致算法多样性较差. 最近提出的一些算法则直接在原始高维空间中通过优秀个体引导搜索优化, 如基于竞争粒子群优化策略的大规模多目标进化算法(LMOCSO)^[15], 提出一种新的粒子更新策略, 劣质粒子向优质粒子学习, 生成有希望的子代来加速对全局最优解的搜索. 自适应子代进化大规模多目标优化(DGEA)^[16], 通过支配解与非支配解构造收敛性和多样性搜索方向来自适应地引导子代的

收稿日期: 2023-02-24.

宁波大学学报(理工版)网址: <http://journaljg.nbu.edu.cn/>

基金项目: 国家自然科学基金(61773225); 宁波市重点研发计划暨“揭榜挂帅”(2023Z067).

第一作者: 熊英剑, 硕士研究生, 主要研究方向: 模式识别与智能系统等. E-mail: xyj2879566278@163.com

*通信作者: 史旭华, 副教授, 主要研究方向: 模式识别与智能系统等. E-mail: shixuhua@nbu.edu.cn

生成;基于定向采样的进化大规模多目标优化(LMOEA-DS)^[17],通过构造双向搜索方向,并沿搜索方向引导子代生成.综上可知,对于大规模多目标优化问题,能够在正确方向搜索,并在有限的计算预算下获得好的子代个体至关重要.

本文受优秀个体引导搜索算法的启发,提出了一种基于方向向量采样的大规模多目标进化算法(LSMOEA-DVS),该算法通过候选个体、边界点和中心点,构建与收敛性、多样性相关的方向向量,然后沿着方向向量定向采样生成子代个体,从而加强算法的寻优能力,提高算法的整体性能.

1 多目标优化问题

多目标优化可以用数学公式表示为^[18]:

$$\begin{cases} \min F(\mathbf{X}) = (f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_m(\mathbf{X})), \\ \text{subject to } \mathbf{X} \in \Omega, \end{cases} \quad (1)$$

式中: $\mathbf{X} = (x_1, x_2, \dots, x_D)$ 是一个来自 D 维的决策空间 Ω 的决策向量; $F(\mathbf{X})$ 是目标函数向量,包含 m 个互相冲突的目标函数.

2 基于方向向量采样的大规模多目标进化算法

本文提出的 LSMOEA-DVS 算法,基于分解的由参考向量引导下一代个体选择的基本框架,在子代个体产生过程中,沿构建方向向量采样生成子代个体,从而加速在高维决策空间中寻找合适的进化方向,加快算法的收敛速度,具体流程如下.

2.1 预处理

预处理目的是从种群的每一代中选择离帕累托前沿(PF)最近,且分布最好的优秀个体充当候选个体,以满足后续采样策略构建方向向量.

2.1.1 目标值转化

由于参考向量的初始点始终是坐标原点,为了方便种群个体的划分,需要对种群个体 $\mathbf{P}_t$ 的目标值进行转化,假设 $\mathbf{F}_t = \{f_{t,1}, f_{t,2}, \dots, f_{t,|p_t|}\}$, t 表示第 t 次迭代.转换公式为:

$$\mathbf{f}'_{t,i} = \mathbf{f}_{t,i} - \mathbf{z}_t^{\min}, i=1,2,\dots,|p_t|, \quad (2)$$

式中: $\mathbf{f}_{t,i}$ 和 $\mathbf{f}'_{t,i}$ 是个体 i 在转换前和后的目标向量; $\mathbf{z}_t^{\min} = (z_{t,1}^{\min}, z_{t,2}^{\min}, \dots, z_{t,m}^{\min})$, 表示当前种群中在每个目标上的最小值,目的是保证转换后种群个体所有目标值与参考向量都在目标空间的第一象限,此时理想点为坐标原点.

2.1.2 种群分配

为了得到多样性好的种群个体,需要将种群个体分配给均匀分布的参考向量,通过种群个体与参考向量在目标空间中的夹角关系进行分配,具体分配公式为:

$$\cos \theta_{t,i,j} = \frac{\mathbf{f}'_{t,i} \cdot \mathbf{V}_{t,j}}{\|\mathbf{f}'_{t,i}\| \cdot \|\mathbf{V}_{t,j}\|}, \quad (3)$$

式中: $\theta_{t,i,j}$ 代表种群个体 $\mathbf{p}_{t,i}$ 转化后的目标向量 $\mathbf{f}'_{t,i}$ 与参考向量 $\mathbf{V}_{t,j}$ 之间的夹角,当且仅当种群个体 $\mathbf{p}_{t,i}$ 与参考向量 $\mathbf{V}_{t,k}$ 夹角最小(余弦值最大)时,将其分配给参考向量 $\mathbf{V}_{t,k}$.

2.1.3 候选个体选择

在对当前种群个体目标值转化以及将种群个体分配给当前参考向量 $\mathbf{V}_t$ 后,将那些至少分配一个个体的参考向量称为活动参考向量 $\mathbf{V}_{t,a}$,未分配种群个体的参考向量称为非活动参考向量 $\mathbf{V}_{t,ina}$.使用 k -均值聚类,将活动参考向量聚为 k 类,如果活动参考向量数量小于所需要聚类的簇数 k ,则将活动参考向量聚为活动参考向量个数 N_a 类,因此种群个体被分为 $\min(k, N_a)$ 个子种群,然后从每一个子种群中选择一个离理想点距离最近的个体,这样可以得到收敛性和多样性良好的个体用来构建方向向量.这些优秀的候选个体离 PF 最近,因此适合用来构建方向向量,从而引导算法朝正确方向进化.

2.2 定向采样策略

2.2.1 收敛性相关采样策略

随着决策变量维度的增加,算法的搜索空间呈指数剧增,传统的交叉变异生成子代的方式在巨大的搜索空间中效率急剧下降,使得算法的收敛速度缓慢,构造有希望的方向向量引导向 PF 方向搜索,加快算法的收敛速度.本文提出了在决策空间中构造方向向量引导个体朝帕累托最优解集方向生成,该阶段首先通过预处理选出优秀候选个体,通过其在决策空间的位置与决策空间的上下边界点和中心点来构造方向向量.不同于 LSMOF 算法和 LMOEA-DS 算法,本文引入中心点构建三向搜索的方向向量,中心点的加入增加了方向向量与决策空间中帕累托解集相交的机会,从而增大了沿方向向量采样生成的子代与帕累托解集接近的机会,然后通过实验验证引入中心点

构建方向向量对算法性能的作用.

方向向量定义:

$$\begin{aligned} \mathbf{v}_{l,i} &= \mathbf{X}_i - \mathbf{L}, \\ \mathbf{v}_{u,i} &= \mathbf{X}_i - \mathbf{U}, \\ \mathbf{v}_{c,i} &= \mathbf{X}_i - \mathbf{C}, \end{aligned} \quad (4)$$

式中: $\mathbf{L}$ 和 $\mathbf{U}$ 为决策变量空间下边界点和上边界点; $\mathbf{X}_i$ 为预处理选出的候选个体; $\mathbf{C}$ 为中心点, 定义:

$$\mathbf{C} = \mathbf{L} + \frac{\mathbf{U} - \mathbf{L}}{2}. \quad (5)$$

在决策空间中可以生成一组方向向量 $\mathbf{v} = \{\mathbf{v}_{l,1}, \mathbf{v}_{u,1}, \mathbf{v}_{c,1}, \dots, \mathbf{v}_{l,k}, \mathbf{v}_{u,k}, \mathbf{v}_{c,k}\}$. 每个方向向量代表一个搜索方向, 搜索起点是上边界点或下边界点或中心点, 之后在每个搜索方向上随机采样多个解, 这些解就是新生成的子代个体. 该过程的具体计算公式为:

$$\begin{aligned} \mathbf{O}_{l,i,j} &= \mathbf{L} + \gamma_{l,i,j} \cdot \frac{\mathbf{v}_{l,i}}{\|\mathbf{v}_{l,i}\|}, \\ \mathbf{O}_{u,i,k} &= \mathbf{U} + \gamma_{u,i,k} \cdot \frac{\mathbf{v}_{u,i}}{\|\mathbf{v}_{u,i}\|}, \\ \mathbf{O}_{c,i,l} &= \mathbf{C} \pm \gamma_{c,i,l} \cdot \frac{\mathbf{v}_{c,i}}{\|\mathbf{v}_{c,i}\|}, \end{aligned} \quad (6)$$

式中: $\gamma_{l,i,j}$ 和 $\gamma_{u,i,k}$ 为采样系数, 均匀分布在 $[0, \|\mathbf{U} - \mathbf{L}\|]$ 上的随机数; $\gamma_{c,i,l}$ 是均匀分布在 $[0, \|\mathbf{U} - \mathbf{L}\|/2]$ 上的随机数; $\|\cdot\|$ 代表欧几里得距离, 保证沿方向向量 $\mathbf{v}_{l,i}$ 、 $\mathbf{v}_{u,i}$ 、 $\mathbf{v}_{c,i}$ 上生成的个体能够覆盖整个搜索空间; j, k, l 分别代表沿方向向量 $\mathbf{v}_{l,i}$ 、 $\mathbf{v}_{u,i}$ 、 $\mathbf{v}_{c,i}$ 的第 j 次、第 k 次、第 l 次采样.

2.2.2 多样性相关采样策略

收敛性相关采样策略虽然能够产生收敛性较好的个体来加速收敛, 但不能保证产生的个体具有良好的多样性, 因此提出多样性相关采样策略, 使其产生更多收敛性好的个体, 从而提高种群的多样性. 将父代种群 $\mathbf{P}$ 与收敛性相关采样策略生成的后代个体通过预处理选出 k 个收敛性和多样性良好的候选个体 $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k\}$, 依次选择这 k 个优秀个体, 充当构造方向向量的起始点; 然后随机从另外 $(k-1)$ 个候选个体中选择一个充当构造方向向量的终点, 构造过程定义为:

$$\mathbf{v}_i = \mathbf{X}_j - \mathbf{X}_i, \quad (7)$$

式中: $i = \{1, 2, \dots, k\}$, j 属于 $\{1, 2, \dots, k\}$ 中不等于 i

的任意一个数; $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$ 为生成的一组方向向量, 每个方向向量代表一个搜索方向, 搜索起点是 $\mathbf{X}_i$, 在每个方向上随机采样多个解, 这些解就是新生成的子代, 其过程为:

$$\mathbf{Q}_{i,j} = \mathbf{X}_i \pm \delta_{i,j} \cdot \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}, \quad (8)$$

式中: 采样系数 $\delta_{i,j}$ 是均匀分布在 $[0, \|\mathbf{v}_i\|]$ 上的随机数; j 代表沿方向向量 $\mathbf{v}_i$ 第 j 次采样.

在生成子代个体 $\mathbf{Q}$ 后, 将其与父代 $\mathbf{P}$ 和收敛性采样策略生成的子代个体 $\mathbf{O}$, 通过非支配排序选择 N 个非支配个体, 再进行交叉变异, 然后产生子代个体 $\mathbf{S}$ 来进一步提高种群的多样性.

2.2.3 越界处理

由采样系数生成的某些解可能超出决策空间的边界, 因此将超出边界位置的决策变量值重新投影回可行域, 其过程定义为:

$$\begin{aligned} x_{i,d} &= \min(x_{i,d}, U_d), \\ x_{i,d} &= \max(x_{i,d}, L_d), \end{aligned} \quad (9)$$

式中: $x_{i,d}$ 为第 i 个解的第 d 个决策变量值; U_d 和 L_d 分别代表上下边界点的第 d 个决策变量值.

2.3 环境选择

合并父代种群 $\mathbf{P}$, 子代种群 $\mathbf{O}$ 、 $\mathbf{Q}$ 和 $\mathbf{S}$ 进行基于分解的环境选择策略, 该策略首先将种群个体的目标值进行转化, 再将其分配给参考向量 $\mathbf{V}$, 然后从每一个参考向量所分配个体集合中选择在目标空间中离理想点欧几里得距离最近的个体进入下一代. 此外, 在优化的后半部分, 在对种群进行环境选择前, 需要将其非支配排序, 删除当前种群中所有的支配解, 目的是在优化的前半部分尽可能让种群沿均匀分布的参考向量探索整个目标空间, 避免陷入局部最优. 同时在优化的后半部分, 排除一些在具有不规则 PF 几何结构优化问题上, 基于均匀参考向量分解的环境策略, 这会产生一些被支配的种群个体选择进入下一代的隐患.

2.4 参考向量调整

由于没有对种群个体目标值进行归一化, 可能存在多目标优化问题, 不同的目标被缩放到不同的范围, 这样即使均匀分布的参考向量也不会产生均匀分布的解. 因此, 使用文献[19]中的参考向量自适应方法, 根据目标值调整参考向量. 在一

些具有不规则 PF 几何结构的优化问题中, 均匀分布的参考向量可能会存在一些无效参考向量(没有非支配个体与之相关联), 这会降低算法获得 Pareto 最优解的密度, 同时在这些无效参考向量上浪费计算资源. 因此, 在优化后半部分采用文献[19]中参考向量再生成策略, 对这些无效参考向量进行重新生成, 具体可参阅文献[19].

2.5 LSMOEA-DVS 算法步骤

Step 1 初始化大小为 N 的种群和参考向量 V , 设定最大评估次数为 $F_{E\max}$.

Step 2 先进行预处理, 再按照式(4)~(6)执行收敛性相关采样策略, 最后按照式(9)执行越界处理.

Step 3 先进行预处理, 再按照式(7)和(8)执行多样性相关采样策略, 最后按照式(9)执行越界处理.

Step 4 父子代合并进行环境选择.

Step 5 调整参考向量.

Step 6 判断是否达到最大评估次数, 若满足条件, 输出最终种群, 否则返回 Step 2.

3 实验结果与分析

通过与 5 种代表性算法(LMOEA-DS^[17]、LMOCSSO^[15]、LSMOF^[14]、WOF^[13]和一种用于大规模多目标优化的模糊决策变量框架(FDV)^[20]在 DTLZ^[21]测试函数的 7 个测试问题上进行比较, 评估 LSMOEA-DVS 算法的性能. 这些问题分别具有 500、1 000、2 000 个决策变量, 以及 2、3 个目标维度. 为公平比较, 所有算法均在 PlatEMO^[22]中实现, 并在每个基准问题上使用 AMD Ryzen 7 5800H CPU, 3.20 GHz 处理器和 16GB 内存, 独立运行 20 次.

3.1 性能指标与参数设置

在实验中, 采用反转世代距离(IGD)^[23]作为评估算法的性能指标, 具体公式为:

$$IGD(P^*, P) = \sum_{x \in P^*} d(x, P) / |P^*|, \quad (10)$$

式中: P^* 为均匀分布在 PF 上的参考点集; P 为获得的解集; $d(x, P)$ 为 PF 上的参考点 x 到解集 P 的最小欧几里得距离.

IGD 指标可以衡量算法获得的种群的收敛性和多样性, IGD 指标值越小, 获得的种群越好.

所有算法独立运行 20 次, 初始种群大小设置

为 100. 在目标 3 和目标 2 的 DTLZ 测试函数上算法允许的最大评估次数分别为 20 万次和 6 万次. 使用的遗传算子是模拟二进制交叉(SBX)和多项式变异(PM), 对于 SBX, 分布指数设置为 20, 交叉概率设置为 1; 对于 PM, 分布指数设置为 20, 突变概率设置为 $1/D$, 其中 D 是决策变量的数量. 预处理选择的候选个体数量 $K_{\min}$ 设置为 10, 采样策略沿着每个方向向量随机生成解的数量为 30 个.

所有算法使用的参数设置与原论文中给出的参数设置相同, 对于 WOF 和 FDV, 非支配排序遗传算法(NSGA2)作为其嵌入的优化算法.

3.2 实验结果

3.2.1 消融实验

对决策变量为 500、1 000、2 000 个的 7 个 DTLZ 测试问题进行测试, 以证明 LSMOEA-DVS 算法的有效性. 表 1 给出了运行 20 次的 IGD 指标均值, 括号内为标准差, 阴影表示取得的结果在所有算法中最佳.

表 1 收敛性相关采样策略消融实验结果

测试函数	D	LSMOEA-DVS-two	LSMOEA-DVS
DTLZ1	500	3.195 2e-1(5.34e-1) ^②	2.054 5e-2(2.11e-4)
	1 000	1.175 5e+0(2.44e+0) ^②	2.096 6e-2(1.09e-3)
	2 000	4.141 6e+0(5.78e+0) ^②	2.061 3e-2(2.75e-4)
DTLZ2	500	1.463 9e-1(1.08e-2) ^②	5.583 6e-2(8.15e-4)
	1 000	3.418 7e-1(7.74e-2) ^②	5.560 5e-2(6.29e-4)
	2 000	7.227 9e-1(8.32e-2) ^②	5.573 9e-2(9.06e-4)
DTLZ3	500	1.543 2e+0(5.22e+0) ^②	5.593 7e-2(5.04e-4)
	1 000	1.794 6e+0(3.40e+0) ^②	5.593 6e-2(1.10e-3)
	2 000	1.968 7e+1(6.10e+1) ^②	5.575 9e-2(6.85e-4)
DTLZ4	500	1.008 5e-1(6.72e-2) ^②	5.634 9e-2(5.03e-4)
	1 000	1.696 1e-1(1.50e-1) ^②	5.661 2e-2(1.10e-3)
	2 000	3.482 7e-1(3.34e-1) ^②	5.629 4e-2(6.18e-4)
DTLZ5	500	2.006 8e-2(1.35e-2) ^②	8.582 5e-3(1.18e-3)
	1 000	4.253 6e-2(6.61e-2) ^②	8.359 7e-3(1.02e-3)
	2 000	1.404 2e-1(2.16e-1) ^②	7.732 6e-3(6.13e-4)
DTLZ6	500	7.536 9e-3(1.07e-3) ^③	7.641 0e-3(8.63e-4)
	1 000	7.550 8e-3(1.43e-3) ^③	8.288 5e-3(1.74e-3)
	2 000	7.765 0e-3(1.63e-3) ^③	8.114 9e-3(1.51e-3)
DTLZ7	500	6.016 0e-2(2.31e-3) ^③	6.207 4e-2(3.30e-3)
	1 000	6.148 7e-2(3.52e-3) ^③	6.013 6e-2(2.05e-3)
	2 000	6.087 5e-2(3.59e-3) ^③	6.187 3e-2(3.21e-3)

注: D 表示决策变量维度; M 表示目标维度; ①表示该算法结果优于本文所提的算法; ②表示该算法结果差于本文所提的算法; ③表示该算法结果与本文所提的算法相似. 下表同.

(1)收敛性相关采样策略效果. 在 LSMOEADVS 中, 以上下边界点和中心点 3 个点作为起点, 构造 3 个方向向量进行定向采样. 为检查附加中心点是否有效, 将其与仅以 2 个边界点为起点来构造方向向量的方法(LSMOEADVS-two)比较, 结果见表 1. 从表 1 可见, 与 LSMOEADVS-two 相比, LSMOEADVS 在 21 个测试问题中有 15 个表现更佳, 具体集中在测试函数 DTLZ1~DTLZ5 上, 在测试函数 DTLZ6 和 DTLZ7 上, 两者表现同样出色. 据此可知, LSMOEADVS 算法有效.

(2)多样性相关采样策略的效果. 为评估多样性相关采样策略的性能, 将 LSMOEADVS 算法与仅使用收敛性相关采样策略(LSMOEADVS-c)的变体算法进行比较, 结果见表 2. 从表 2 可见, 在 21 个 DTLZ 测试问题中, LSMOEADVS 算法性能均更佳, 表明该算法大大提高了所获得解集的质量, 证明其是有效的.

表 2 多样性相关采样策略消融实验结果

测试函数	D	LSMOEA-DVS-c	LSMOEA-DVS
DTLZ1	500	2.088 0e-1(9.32e-4) ^②	2.054 5e-2(2.11e-4)
	1 000	2.090 0e-1(1.07e-3) ^②	2.096 6e-2(1.09e-3)
	2 000	2.100 3e-1(4.18e-3) ^②	2.061 3e-2(2.75e-4)
DTLZ2	500	5.611 2e-1(1.57e-3) ^②	5.583 6e-2(8.15e-4)
	1 000	5.618 1e-1(4.64e-4) ^②	5.560 5e-2(6.29e-4)
	2 000	5.620 1e-1(1.30e-4) ^②	5.573 9e-2(9.06e-4)
DTLZ3	500	5.637 1e-1(1.90e-3) ^②	5.593 7e-2(5.04e-4)
	1 000	5.664 5e-1(1.17e-2) ^②	5.593 6e-2(1.10e-3)
	2 000	5.713 9e-1(1.47e-2) ^②	5.575 9e-2(6.85e-4)
DTLZ4	500	9.459 2e-1(3.74e-8) ^②	5.634 9e-2(5.03e-4)
	1 000	9.459 2e-1(3.41e-8) ^②	5.661 2e-2(1.10e-3)
	2 000	9.459 2e-1(8.56e-8) ^②	5.629 4e-2(6.18e-4)
DTLZ5	500	4.321 9e-1(1.11e-3) ^②	8.582 5e-3(1.18e-3)
	1 000	4.324 1e-1(7.01e-4) ^②	8.359 7e-3(1.02e-3)
	2 000	4.325 0e-1(3.15e-4) ^②	7.732 6e-3(6.13e-4)
DTLZ6	500	1.773 7e+1(9.72e+0) ^②	7.641 0e-3(8.63e-4)
	1 000	3.499 1e+1(2.29e+1) ^②	8.288 5e-3(1.74e-3)
	2 000	6.098 0e+1(2.69e+1) ^②	8.114 9e-3(1.51e-3)
DTLZ7	500	1.347 3e+0(3.42e-1) ^②	6.207 4e-2(3.30e-3)
	1 000	1.416 9e+0(2.73e-1) ^②	6.013 6e-2(2.05e-3)
	2 000	1.538 1e+0(1.23e-4) ^②	6.187 3e-2(3.21e-3)

3.2.2 五种大规模多目标算法的对比实验

将 LSMOEADVS 算法与其他 5 种有代表性的大规模多目标算法在 DTLZ 测试函数上进行比较,

结果见表 3. 从表 3 可见, 在测试函数 DTLZ1~DTLZ7 上 LSMOEADVS 算法在 42 个测试问题中有 36 个测试结果优于其他算法. 对 LMOEA-DS、LMOCSSO、LSMOF、WOF 和 FDV 算法, LSMOEADVS 算法在 42 个测试问题中分别有 39、42、36、42、41 个测试问题胜出. LSMOEADVS 算法在测试函数 DTLZ1~DTLZ4 和 DTLZ7 得到的结果均为最佳, 特别在 2 目标的 DTLZ 问题上, LSMOEADVS 算法在测试函数 DTLZ1~DTLZ7 上均取得最佳结果. LMOEA-DS 算法仅在测试函数 DTLZ6 和 DTLZ7 上具有良好性能. 与其他几种算法相比, LSMOF 算法是除 LSMOEADVS 算法外, 性能较好的一种算法. 然而, 由于算法本身缺乏多样性维护, 其没有取得最佳结果, 其他几种算法也未能取得较好结果.

为了更加直观地说明 LMOEA-DVS 算法的优越性, 给出了 6 种算法在 DTLZ 测试问题上获得的最终解集的分布图(图 1), 其中灰色实心圆点代表一个解, 灰色网格线代表真实 PF.

从图 1 可见, LSMOEADVS 算法真正收敛到真实 PF 上, 与 IGD 值给出的结论类似; LSMOF 算法是对比算法中表现最好的一种算法, 也能收敛到真实 PF 上, 但是获得的解集分布比较密集, 没有广泛分布在 PF 上, 表明像 LSMOF 这类基于问题转换的算法容易陷入局部最优而导致解集的多样性变差. LMOEA-DS 算法也能收敛到真实 PF 上, 但是由于该算法是基于分解和支配互补的环境选择, 且通过一个阈值判定环境选择是基于分解还是支配, 因此仍有一定概率存在支配个体保留下来的可能性. 其他几种算法则没有能够收敛到真实 PF 上.

图 2 为各种算法在 1 000 个决策变量的 2 目标 DTLZ7 上获得的解集. 从图 2 可知, 相比于 3 目标, 所有算法在 2 目标上均表现较佳, 其中 LSMOEADVS 算法仍表现最好, 能够实现解集在真正 PF 上全覆盖. 而基于问题转换的算法, 如 LSMOF 算法和 WOF 算法, 只找到一半的真实 PF; LMOEA-DS 算法虽能找到全部的真实 PF, 却保留了不连续区域的支配解, 可知其并不能处理不连续 PF 的多目标优化问题; LMOCSSO 算法同样不能处理不规则 PF 的优化问题, 且仍然需要更多的适应度函数来

表3 与5种大规模多目标算法的对比实验结果

测试函数	M	D	LMOEADS	LMOCSSO	LSMOF	WOF	FDV	LSMOEA-DVS
DTLZ1	2	500	$1.793\ 3\text{e}+0(2.82\text{e}+0)^{②}$	$3.389\ 6\text{e}+3(8.77\text{e}+2)^{②}$	$2.566\ 7\text{e}-3(3.42\text{e}-4)^{②}$	$2.690\ 9\text{e}+3(5.47\text{e}+2)^{②}$	$5.369\ 1\text{e}+1(9.75\text{e}+0)^{②}$	$1.937\ 3\text{e}-3(3.03\text{e}-5)$
		1 000	$1.173\ 3\text{e}+1(2.26\text{e}+1)^{②}$	$6.640\ 2\text{e}+3(1.73\text{e}+3)^{②}$	$2.488\ 6\text{e}-3(3.07\text{e}-4)^{②}$	$6.390\ 7\text{e}+3(1.44\text{e}+3)^{②}$	$4.496\ 3\text{e}+2(4.21\text{e}+1)^{②}$	$1.941\ 2\text{e}-3(3.43\text{e}-5)$
		2 000	$2.328\ 3\text{e}+1(2.47\text{e}+1)^{②}$	$1.355\ 7\text{e}+4(3.90\text{e}+3)^{②}$	$2.382\ 1\text{e}-3(2.58\text{e}-4)^{②}$	$1.193\ 3\text{e}+4(3.59\text{e}+3)^{②}$	$1.818\ 0\text{e}+3(8.52\text{e}+1)^{②}$	$1.945\ 2\text{e}-3(2.69\text{e}-5)$
	3	500	$2.079\ 4\text{e}+0(3.93\text{e}+0)^{②}$	$1.814\ 7\text{e}+3(3.16\text{e}+2)^{②}$	$2.962\ 9\text{e}-2(1.98\text{e}-3)^{②}$	$2.301\ 3\text{e}+3(4.73\text{e}+2)^{②}$	$2.816\ 8\text{e}-2(1.25\text{e}-3)^{②}$	$2.054\ 5\text{e}-2(2.11\text{e}-4)$
		1 000	$8.311\ 7\text{e}+0(1.52\text{e}+1)^{②}$	$3.453\ 6\text{e}+3(8.15\text{e}+2)^{②}$	$2.957\ 7\text{e}-2(1.60\text{e}-3)^{②}$	$4.258\ 2\text{e}+3(1.03\text{e}+3)^{②}$	$6.645\ 5\text{e}+1(6.69\text{e}+0)^{②}$	$2.096\ 6\text{e}-2(1.09\text{e}-3)$
		2 000	$8.261\ 6\text{e}+0(8.56\text{e}+0)^{②}$	$7.056\ 3\text{e}+3(1.62\text{e}+3)^{②}$	$2.978\ 6\text{e}-2(1.92\text{e}-3)^{②}$	$9.156\ 8\text{e}+3(2.78\text{e}+3)^{②}$	$7.487\ 4\text{e}+2(6.06\text{e}+1)^{②}$	$2.061\ 3\text{e}-2(2.75\text{e}-4)$
DTLZ2	2	500	$4.447\ 2\text{e}-1(3.06\text{e}-2)^{②}$	$9.765\ 5\text{e}-1(7.86\text{e}-2)^{②}$	$5.450\ 2\text{e}-3(2.41\text{e}-4)^{②}$	$3.234\ 5\text{e}-1(1.24\text{e}-1)^{②}$	$1.980\ 2\text{e}-1(4.76\text{e}-2)^{②}$	$4.211\ 7\text{e}-3(8.54\text{e}-5)$
		1 000	$1.085\ 5\text{e}+0(1.04\text{e}-1)^{②}$	$2.512\ 2\text{e}+0(2.70\text{e}-1)^{②}$	$5.216\ 9\text{e}-3(1.64\text{e}-4)^{②}$	$1.839\ 1\text{e}+0(9.54\text{e}-1)^{②}$	$7.573\ 3\text{e}+0(6.08\text{e}-1)^{②}$	$4.194\ 6\text{e}-3(4.10\text{e}-5)$
		2 000	$2.169\ 8\text{e}+0(2.23\text{e}-1)^{②}$	$5.691\ 3\text{e}+0(5.34\text{e}-1)^{②}$	$5.332\ 8\text{e}-3(1.99\text{e}-4)^{②}$	$9.464\ 4\text{e}+0(4.99\text{e}+0)^{②}$	$4.328\ 0\text{e}+1(2.42\text{e}+0)^{②}$	$4.188\ 1\text{e}-3(3.59\text{e}-5)$
	3	500	$2.663\ 1\text{e}-1(2.07\text{e}-2)^{②}$	$6.148\ 1\text{e}-1(8.34\text{e}-2)^{②}$	$1.291\ 4\text{e}-1(2.57\text{e}-1)^{②}$	$1.753\ 2\text{e}-1(4.06\text{e}-2)^{②}$	$6.897\ 6\text{e}-2(2.34\text{e}-3)^{②}$	$5.583\ 6\text{e}-2(8.15\text{e}-4)$
		1 000	$6.197\ 6\text{e}-1(1.29\text{e}-1)^{②}$	$2.238\ 6\text{e}+0(2.29\text{e}-1)^{②}$	$4.926\ 4\text{e}+0(1.26\text{e}+1)^{②}$	$7.122\ 0\text{e}-1(3.80\text{e}-1)^{②}$	$4.223\ 8\text{e}-1(8.65\text{e}-2)^{②}$	$5.560\ 5\text{e}-2(6.29\text{e}-4)$
		2 000	$1.394\ 7\text{e}+0(5.04\text{e}-1)^{②}$	$6.005\ 6\text{e}+0(5.01\text{e}-1)^{②}$	$7.149\ 2\text{e}-2(2.83\text{e}-3)^{②}$	$3.587\ 9\text{e}+0(2.07\text{e}+0)^{②}$	$1.647\ 2\text{e}+1(1.12\text{e}+0)^{②}$	$5.573\ 9\text{e}-2(9.06\text{e}-4)$
DTLZ3	2	500	$8.460\ 9\text{e}+0(1.20\text{e}+1)^{②}$	$1.014\ 0\text{e}+4(2.19\text{e}+3)^{②}$	$5.419\ 6\text{e}-3(4.87\text{e}-4)^{②}$	$7.413\ 6\text{e}+3(1.22\text{e}+3)^{②}$	$1.453\ 3\text{e}+2(2.46\text{e}+1)^{②}$	$4.260\ 5\text{e}-3(2.07\text{e}-4)$
		1 000	$1.002\ 2\text{e}+1(1.34\text{e}+1)^{②}$	$1.755\ 5\text{e}+4(5.22\text{e}+3)^{②}$	$5.419\ 2\text{e}-3(4.23\text{e}-4)^{②}$	$1.722\ 9\text{e}+4(3.35\text{e}+3)^{②}$	$1.292\ 3\text{e}+3(8.36\text{e}+1)^{②}$	$4.215\ 0\text{e}-3(6.82\text{e}-5)$
		2 000	$6.283\ 1\text{e}+1(8.55\text{e}+1)^{②}$	$3.700\ 6\text{e}+4(1.09\text{e}+4)^{②}$	$5.595\ 1\text{e}-3(4.15\text{e}-4)^{②}$	$3.701\ 0\text{e}+4(6.66\text{e}+3)^{②}$	$5.197\ 3\text{e}+3(2.46\text{e}+2)^{②}$	$4.202\ 1\text{e}-3(5.20\text{e}-5)$
	3	500	$4.318\ 6\text{e}+0(7.60\text{e}+0)^{②}$	$5.702\ 5\text{e}+3(1.79\text{e}+3)^{②}$	$7.090\ 4\text{e}-2(3.80\text{e}-3)^{②}$	$6.458\ 3\text{e}+3(1.33\text{e}+3)^{②}$	$7.066\ 0\text{e}-2(2.69\text{e}-3)^{②}$	$5.593\ 7\text{e}-2(5.04\text{e}-4)$
		1 000	$3.097\ 7\text{e}+1(8.44\text{e}+1)^{②}$	$1.184\ 3\text{e}+4(2.15\text{e}+3)^{②}$	$7.249\ 6\text{e}-2(4.16\text{e}-3)^{②}$	$1.360\ 1\text{e}+4(2.79\text{e}+3)^{②}$	$1.887\ 9\text{e}+2(2.52\text{e}+1)^{②}$	$5.593\ 6\text{e}-2(1.10\text{e}-3)$
		2 000	$1.908\ 6\text{e}+1(2.03\text{e}+1)^{②}$	$2.228\ 3\text{e}+4(5.80\text{e}+3)^{②}$	$7.168\ 1\text{e}-2(3.20\text{e}-3)^{②}$	$2.965\ 3\text{e}+4(6.40\text{e}+3)^{②}$	$2.304\ 3\text{e}+3(1.94\text{e}+2)^{②}$	$5.575\ 9\text{e}-2(6.85\text{e}-4)$
DTLZ4	2	500	$4.093\ 8\text{e}-1(1.59\text{e}-1)^{②}$	$4.499\ 5\text{e}+0(1.93\text{e}+0)^{②}$	$5.165\ 8\text{e}-3(1.87\text{e}-4)^{②}$	$4.573\ 9\text{e}-1(3.06\text{e}-1)^{②}$	$3.125\ 0\text{e}-1(1.42\text{e}-1)^{②}$	$4.321\ 7\text{e}-3(2.27\text{e}-4)$
		1 000	$9.575\ 7\text{e}-1(2.65\text{e}-1)^{②}$	$1.077\ 1\text{e}+1(2.86\text{e}+0)^{②}$	$5.052\ 0\text{e}-3(1.77\text{e}-4)^{②}$	$2.312\ 8\text{e}+0(1.52\text{e}+0)^{②}$	$8.628\ 9\text{e}+0(8.81\text{e}-1)^{②}$	$4.282\ 8\text{e}-3(1.32\text{e}-4)$
		2 000	$2.031\ 3\text{e}+0(4.65\text{e}-1)^{②}$	$3.120\ 6\text{e}+1(1.68\text{e}+1)^{②}$	$5.132\ 5\text{e}-3(1.35\text{e}-4)^{②}$	$5.812\ 3\text{e}+0(3.08\text{e}+0)^{②}$	$4.949\ 5\text{e}+1(2.86\text{e}+0)^{②}$	$4.265\ 7\text{e}-3(2.04\text{e}-4)$
	3	500	$2.237\ 8\text{e}-1(1.54\text{e}-1)^{②}$	$1.528\ 0\text{e}+0(1.39\text{e}+0)^{②}$	$7.816\ 2\text{e}-1(1.08\text{e}+0)^{②}$	$1.673\ 6\text{e}-1(3.52\text{e}-2)^{②}$	$7.041\ 9\text{e}-2(2.56\text{e}-3)^{②}$	$5.634\ 9\text{e}-2(5.03\text{e}-4)$
		1 000	$2.975\ 0\text{e}-1(3.06\text{e}-1)^{②}$	$6.249\ 1\text{e}+0(3.39\text{e}+0)^{②}$	$7.090\ 5\text{e}+0(1.20\text{e}+1)^{②}$	$6.329\ 5\text{e}-1(2.26\text{e}-1)^{②}$	$7.402\ 9\text{e}-2(3.76\text{e}-3)^{②}$	$5.661\ 2\text{e}-2(1.10\text{e}-3)$
		2 000	$4.422\ 4\text{e}-1(4.41\text{e}-1)^{②}$	$1.805\ 0\text{e}+1(6.51\text{e}+0)^{②}$	$3.800\ 9\text{e}+1(5.82\text{e}+1)^{②}$	$2.663\ 5\text{e}+0(1.15\text{e}+0)^{②}$	$9.920\ 5\text{e}+0(1.09\text{e}+0)^{②}$	$5.629\ 4\text{e}-2(6.18\text{e}-4)$
DTLZ5	2	500	$4.553\ 7\text{e}-1(3.41\text{e}-2)^{②}$	$9.661\ 0\text{e}-1(1.09\text{e}-1)^{②}$	$5.402\ 2\text{e}-3(2.61\text{e}-4)^{②}$	$3.729\ 6\text{e}-1(2.30\text{e}-1)^{②}$	$2.136\ 4\text{e}-1(4.36\text{e}-2)^{②}$	$4.278\ 2\text{e}-3(1.99\text{e}-4)$
		1 000	$1.062\ 9\text{e}+0(9.32\text{e}-2)^{②}$	$2.548\ 0\text{e}+0(2.45\text{e}-1)^{②}$	$5.276\ 3\text{e}-3(2.07\text{e}-4)^{②}$	$1.928\ 3\text{e}+0(9.31\text{e}-1)^{②}$	$7.629\ 9\text{e}+0(6.11\text{e}-1)^{②}$	$4.195\ 1\text{e}-3(4.58\text{e}-5)$
		2 000	$2.074\ 5\text{e}+0(1.75\text{e}-1)^{②}$	$5.674\ 6\text{e}+0(6.38\text{e}-1)^{②}$	$5.165\ 3\text{e}-3(1.84\text{e}-4)^{②}$	$7.733\ 2\text{e}+0(3.42\text{e}+0)^{②}$	$4.375\ 7\text{e}+1(2.48\text{e}+0)^{②}$	$4.204\ 6\text{e}-3(4.68\text{e}-5)$
	3	500	$2.442\ 2\text{e}-1(1.77\text{e}-2)^{②}$	$6.376\ 5\text{e}-1(6.81\text{e}-2)^{②}$	$6.545\ 1\text{e}-3(4.00\text{e}-4)^{①}$	$1.298\ 3\text{e}-1(6.26\text{e}-2)^{②}$	$5.848\ 7\text{e}-3(4.19\text{e}-4)^{①}$	$8.582\ 5\text{e}-3(1.18\text{e}-3)$
		1 000	$7.571\ 4\text{e}-1(6.05\text{e}-2)^{②}$	$2.091\ 1\text{e}+0(2.17\text{e}-1)^{②}$	$6.530\ 4\text{e}-3(4.45\text{e}-4)^{①}$	$7.593\ 0\text{e}-1(4.03\text{e}-1)^{②}$	$3.536\ 0\text{e}-1(1.19\text{e}-1)^{②}$	$8.359\ 7\text{e}-3(1.02\text{e}-3)$
		2 000	$1.668\ 1\text{e}+0(1.96\text{e}-1)^{②}$	$5.795\ 9\text{e}+0(5.10\text{e}-1)^{②}$	$6.279\ 0\text{e}-3(4.96\text{e}-4)^{①}$	$3.694\ 9\text{e}+0(2.02\text{e}+0)^{②}$	$1.745\ 2\text{e}+1(1.32\text{e}+0)^{②}$	$7.732\ 6\text{e}-3(6.13\text{e}-4)$
DTLZ6	2	500	$6.459\ 2\text{e}-3(3.93\text{e}-4)^{②}$	$1.441\ 7\text{e}+2(2.44\text{e}+1)^{②}$	$5.956\ 7\text{e}-3(3.38\text{e}-4)^{②}$	$6.273\ 2\text{e}+1(1.91\text{e}+1)^{②}$	$1.468\ 4\text{e}+2(1.12\text{e}+1)^{②}$	$4.207\ 0\text{e}-3(4.92\text{e}-5)$
		1 000	$6.364\ 6\text{e}-3(3.64\text{e}-4)^{②}$	$3.743\ 0\text{e}+2(4.16\text{e}+1)^{②}$	$5.903\ 4\text{e}-3(3.48\text{e}-4)^{②}$	$1.592\ 9\text{e}+2(6.59\text{e}+1)^{②}$	$4.557\ 2\text{e}+2(1.26\text{e}+1)^{②}$	$4.205\ 7\text{e}-3(4.77\text{e}-5)$
		2 000	$8.504\ 1\text{e}-2(2.57\text{e}-1)^{②}$	$8.531\ 0\text{e}+2(5.04\text{e}+1)^{②}$	$5.985\ 8\text{e}-3(3.34\text{e}-4)^{②}$	$3.980\ 8\text{e}+2(1.39\text{e}+2)^{②}$	$1.149\ 4\text{e}+3(2.52\text{e}+1)^{②}$	$4.190\ 8\text{e}-3(4.53\text{e}-5)$
	3	500	$5.879\ 2\text{e}-3(5.86\text{e}-4)^{①}$	$6.408\ 8\text{e}+1(1.03\text{e}+1)^{②}$	$6.161\ 6\text{e}-3(3.01\text{e}-4)^{②}$	$2.357\ 0\text{e}+1(1.54\text{e}+1)^{②}$	$4.846\ 6\text{e}+1(7.25\text{e}+0)^{②}$	$7.641\ 0\text{e}-3(8.63\text{e}-4)$
		1 000	$5.589\ 4\text{e}-3(3.12\text{e}-4)^{①}$	$1.785\ 6\text{e}+2(4.29\text{e}+1)^{②}$	$6.246\ 8\text{e}-3(2.55\text{e}-4)^{②}$	$7.691\ 9\text{e}+1(4.11\text{e}+1)^{②}$	$2.971\ 0\text{e}+2(1.20\text{e}+1)^{②}$	$8.288\ 5\text{e}-3(1.74\text{e}-3)$
		2 000	$5.515\ 6\text{e}-3(2.33\text{e}-4)^{①}$	$5.021\ 6\text{e}+2(1.52\text{e}+2)^{②}$	$6.038\ 4\text{e}-3(3.73\text{e}-4)^{②}$	$1.718\ 0\text{e}+2(1.14\text{e}+2)^{②}$	$9.174\ 7\text{e}+2(2.78\text{e}+1)^{②}$	$8.114\ 9\text{e}-3(1.51\text{e}-3)$

续表

测试函数	M	D	LMOEADS	LMOCSO	LSMOF	WOF	FDV	LSMOEA-DVS
DTLZ7	2	500	$7.727\ 6\text{e-}3(4.66\text{e-}4)^{(2)}$	$5.351\ 2\text{e+}0(3.33\text{e-}1)^{(2)}$	$4.429\ 1\text{e-}1(4.25\text{e-}5)^{(2)}$	$4.004\ 5\text{e-}1(1.35\text{e-}1)^{(2)}$	$6.234\ 0\text{e-}1(4.77\text{e-}2)^{(2)}$	$4.866\ 8\text{e-}3(1.64\text{e-}4)$
	1 000		$7.824\ 8\text{e-}3(5.48\text{e-}4)^{(2)}$	$6.261\ 2\text{e+}0(2.01\text{e-}1)^{(2)}$	$4.428\ 7\text{e-}1(9.95\text{e-}5)^{(2)}$	$3.354\ 2\text{e-}1(1.94\text{e-}1)^{(2)}$	$2.154\ 2\text{e+}0(1.81\text{e-}1)^{(2)}$	$4.849\ 3\text{e-}3(8.69\text{e-}5)$
	2 000		$8.026\ 8\text{e-}3(7.00\text{e-}4)^{(2)}$	$6.907\ 8\text{e+}0(1.56\text{e-}1)^{(2)}$	$4.630\ 2\text{e-}1(4.25\text{e-}2)^{(2)}$	$3.781\ 9\text{e-}1(1.60\text{e-}1)^{(2)}$	$3.605\ 6\text{e+}0(1.82\text{e-}1)^{(2)}$	$4.864\ 3\text{e-}3(1.36\text{e-}4)$
	3	500	$7.782\ 7\text{e-}2(4.21\text{e-}3)^{(2)}$	$4.711\ 7\text{e+}0(5.85\text{e-}1)^{(2)}$	$7.992\ 6\text{e-}1(8.14\text{e-}4)^{(2)}$	$5.598\ 8\text{e-}1(3.40\text{e-}1)^{(2)}$	$7.722\ 7\text{e-}2(4.37\text{e-}3)^{(2)}$	$6.207\ 4\text{e-}2(3.30\text{e-}3)$
	1 000		$7.805\ 9\text{e-}2(2.80\text{e-}3)^{(2)}$	$7.107\ 1\text{e+}0(5.91\text{e-}1)^{(2)}$	$8.850\ 4\text{e-}1(2.29\text{e-}1)^{(2)}$	$7.270\ 5\text{e-}1(2.22\text{e-}1)^{(2)}$	$5.244\ 4\text{e-}1(5.49\text{e-}2)^{(2)}$	$6.013\ 6\text{e-}2(2.05\text{e-}3)$
	2 000		$8.146\ 9\text{e-}2(3.74\text{e-}3)^{(2)}$	$8.981\ 3\text{e+}0(5.56\text{e-}1)^{(2)}$	$1.119\ 2\text{e+}0(3.62\text{e-}1)^{(2)}$	$6.897\ 9\text{e-}1(2.66\text{e-}1)^{(2)}$	$2.501\ 3\text{e+}0(1.51\text{e-}1)^{(2)}$	$6.187\ 3\text{e-}2(3.21\text{e-}3)$
①/②/③			3/39/0	0/42/0	6/36/0	0/42/0	1/41/0	

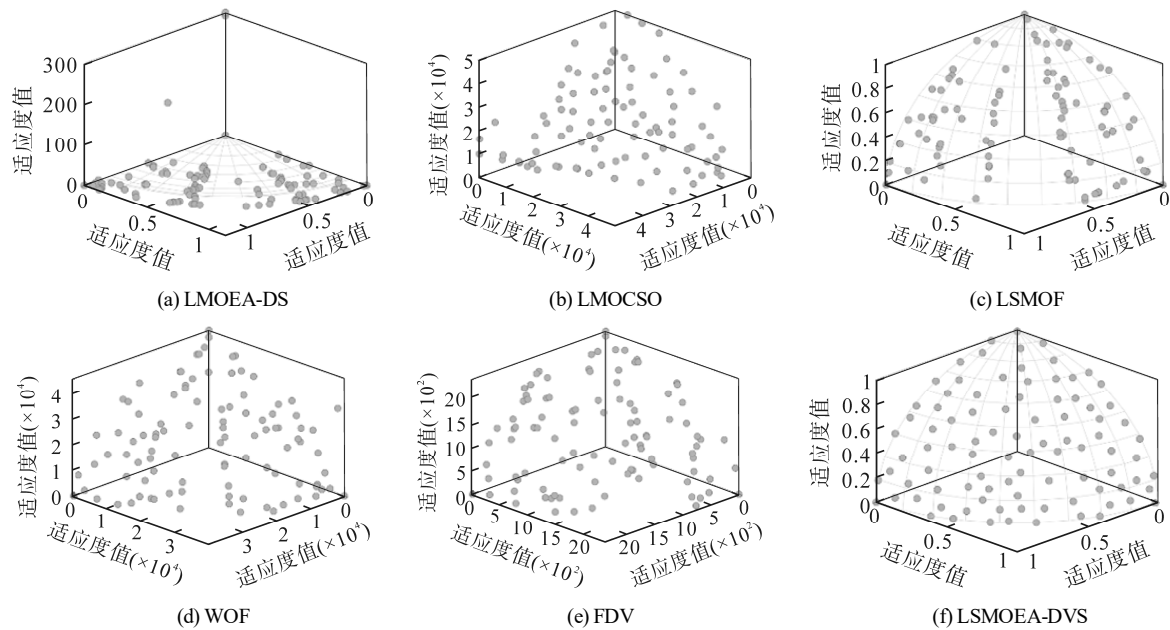


图1 各算法在2 000个决策变量的3目标DTLZ3获得的解集

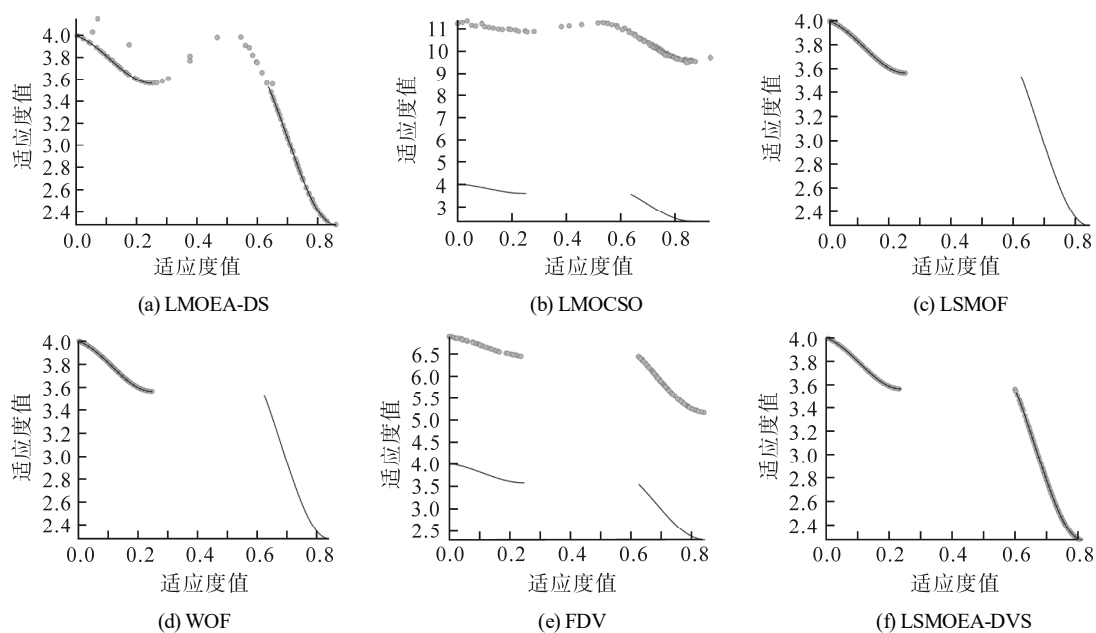
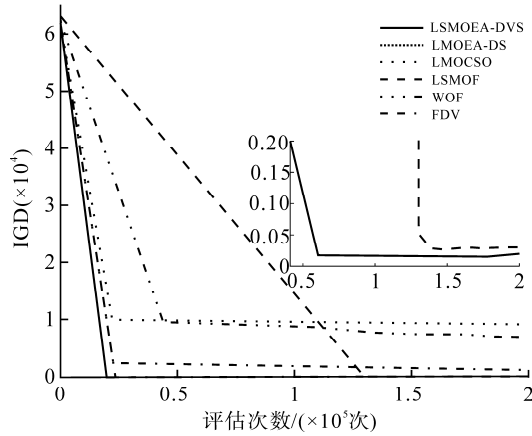


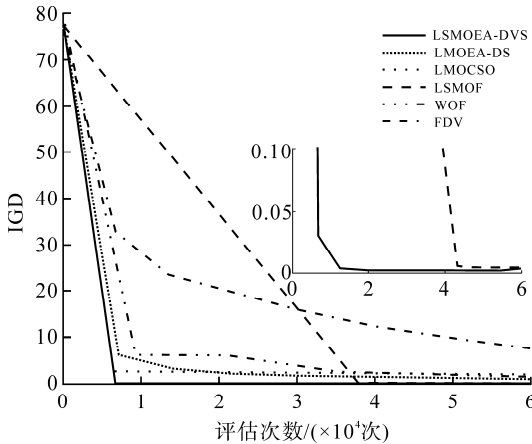
图2 各算法在1 000个决策变量的2目标DTLZ7获得的解集

评估其是否收敛到真实 PF 上, FDV 算法则能够处理不规则 PF 优化问题, 但也需要更多的适应度函数评估其是否收敛到真实 PF 上.

为了更直观地说明 LSMOEADVS 算法的收敛速度, 图 3 给出了所有算法的收敛曲线. 从图 3 可知, LSMOEADVS 算法收敛速度最快, 能够在 3 万次适应度函数评估前收敛到一个较低的水准. LSMOF 算法虽然收敛速度最慢, 但在所有对比算法中能够获得更佳性能. 反观其他几种算法都过早陷入了局部最优, 导致算法后期收敛缓慢, 难以取得较好结果.



(a) 3 目标 2000 个决策变量 DTLZ1



(b) 2 目标 1000 个决策变量 DTLZ5

图 3 各算法在 DTLZ1 和 DTLZ5 上的收敛曲线

4 结语

针对大规模多目标优化中高维的决策变量空间导致算法难以生成有希望的子代个体, 从而使算法性能恶化问题, 提出了一种基于方向向量采样的大规模多目标优化算法(LSMOEADVS), 该

算法通过在决策空间中构造方向向量进行定向采样, 使产生的子代个体更加优异, 从而提高算法的性能. 研究表明, 从 IGD 指标看, LSMOEADVS 算法相比于其他几种大规模多目标优化算法具有更好的表现.

参考文献:

- [1] 孙刚, 伍江江, 陈浩, 等. 一种基于切比雪夫距离的隐式偏好多目标进化算法[J]. 计算机科学, 2022, 49(6): 297-304.
- [2] 谢承旺, 余伟伟, 郭华, 等. DAV-MOEA: 一种采用动态角度向量支配关系的高维多目标进化算法[J]. 计算机学报, 2022, 45(2):317-333.
- [3] 顾清华, 骆家乐, 李学现. 基于小生境的多目标进化算法[J]. 计算机工程与应用, 2023, 59(1):126-139.
- [4] PENG C D, LIU H L, GOODMAN E D, et al. A two-phase framework of locating the reference point for decomposition-based constrained multi-objective evolutionary algorithms[J]. Knowledge-based Systems, 2022, 239:107933.
- [5] GAN X J, SUN J, GONG D W, et al. An adaptive reference vector-based interval multiobjective evolutionary algorithm[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(5):1235-1249.
- [6] PARSONS L, HAQUE E, LIU H. Subspace clustering for high dimensional data[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1):90-105.
- [7] ANTONIO L M, COELLO C A C. Use of cooperative coevolution for solving large scale multiobjective optimization problems[C]//2013 IEEE Congress on Evolutionary Computation. Cancun, Mexico. IEEE, 2013: 2758-2765.
- [8] KUKKONEN S, LAMPINEN J. GDE3: the third evolution step of generalized differential evolution[C]//2005 IEEE Congress on Evolutionary Computation. Edinburgh, UK. IEEE, 2005:443-450.
- [9] MIGUEL A L, COELLO C A. Decomposition-based approach for solving large scale multi-objective problems [C]//HANDL J, HART E, LEWIS P, et al. International conference on parallel problem solving from nature. Cham: Springer, 2016:525-534.
- [10] ZHANG Q F, LI H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [11] MA X L, LIU F, QI Y T, 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.
- [12] ZHANG X Y, 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.
- [13] 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.
- [14] HE C, LI L H, TIAN Y, et al. Accelerating large-scale multiobjective optimization via problem reformulation[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(6):949-961.
- [15] TIAN Y, ZHENG X T, ZHANG X Y, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer[J]. IEEE Transactions on Cybernetics, 2020, 50(8):3696-3708.
- [16] HE C, CHENG R, YAZDANI D. Adaptive offspring generation for evolutionary large-scale multiobjective optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(2):786-798.
- [17] QIN S F, SUN C L, JIN Y C, et al. Large-scale evolutionary multiobjective optimization assisted by directed sampling[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(4):724-738.
- [18] GIAGKIOZIS I, PURSHOUSE R C, FLEMING P J. An overview of population-based algorithms for multi-objective optimisation[J]. International Journal of Systems Science, 2015, 46(9):1572-1599.
- [19] CHENG R, JIN Y C, OLHOFFER M, et al. A reference vector guided evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5):773-791.
- [20] YANG X, ZOU J, YANG S X, et al. A fuzzy decision variables framework for large-scale multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(3):445-459.
- [21] DEB K, THIELE L, LAUMANN S M, et al. Scalable test problems for evolutionary multiobjective optimization [M]//Advanced Information and Knowledge Processing. London: Springer-Verlag, 2005:105-145.
- [22] TIAN Y, CHENG R, ZHANG X Y, et al. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization educational forum[J]. IEEE Computational Intelligence Magazine, 2017, 12(4):73-87.
- [23] SHANG K, ISHIBUCHI H, HE L J, et al. A survey on the hypervolume indicator in evolutionary multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(1):1-20.

Large-scale multi-objective evolutionary optimization based on direction vector sampling

XIONG Yingjian, SHI Xuhua

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: To tackle the difficulty to search for promising offspring individuals in large-scale multi-objective optimization of high-dimensional decision space, which leads to the deterioration of the performance of conventional algorithms, a large-scale multi-objective evolutionary algorithm based on direction vector sampling is proposed. The algorithm first selects some good candidate individuals close to the ideal point, then creates direction vectors in stages, and executes directional sampling strategies, that is, convergence-related sampling strategy and diversity-related sampling strategy to generate offspring individuals. In this way, the offspring individuals with better performance are generated to accelerate the convergence of the algorithm. Experimental results show that compared with five representative large-scale multi-objective evolutionary algorithms, the proposed algorithm has retained strong optimizing ability on large-scale multi-objective optimization test problems with as many as 2 000 decision variables.

Key words: high-dimensional; large-scale multi-objective; direction vector; directional sampling

(责任编辑 史小丽)