

# 控制与决策

Control and Decision

## 基于阈值搜索的多目标人工蜂群算法

赵新秋, 段思雨, 马学敏

引用本文:

赵新秋, 段思雨, 马学敏. 基于阈值搜索的多目标人工蜂群算法[J]. 控制与决策, 2020, 35(8): 1793–1802.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1512>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 基于改进邻域搜索策略的人工蜂群算法

Artificial bee colony algorithm based on improved neighborhood search strategy

控制与决策. 2019, 34(5): 965–972 <https://doi.org/10.13195/j.kzyjc.2017.1506>

### 多搜索策略协同进化的人工蜂群算法

Artificial bee colony algorithm with multi-search strategy cooperative evolutionary

控制与决策. 2018, 33(2): 235–241 <https://doi.org/10.13195/j.kzyjc.2016.1597>

### 一种基于进化知识融合的多目标人工蜂群算法

A multi-objective artificial bee colony based on evolutionary knowledge integrated

控制与决策. 2017, 32(12): 2176–2182 <https://doi.org/10.13195/j.kzyjc.2016.1322>

### 分布式人工蜂群免疫算法求解函数优化问题

Distributed artificial bee colony immune algorithm for the problems of function optimization

控制与决策. 2015(7): 1181–1188 <https://doi.org/10.13195/j.kzyjc.2014.0547>

### Tent混沌人工蜂群与粒子群混合算法

Hybridization algorithm of Tent chaos artificial bee colony and particle swarm optimization

控制与决策. 2015, 30(5): 839–847 <https://doi.org/10.13195/j.kzyjc.2014.0750>

# 基于阈值搜索的多目标人工蜂群算法

赵新秋<sup>1,2†</sup>, 段思雨<sup>1</sup>, 马学敏<sup>1</sup>

(1. 燕山大学 工业计算机控制工程河北省重点实验室, 河北 秦皇岛 066004;

2. 燕山大学 国家冷轧板带装备及工艺工程技术研究中心, 河北 秦皇岛 066004)

**摘要:** 为了平衡人工蜂群算法局部开发能力和全局搜索能力,提高算法收敛速度,提出一种基于阈值搜索的人工蜂群算法. 首先,提出一种混沌镜像初始化方法,保证初始种群的多样性和优越性;然后,利用个体阈值动态调整搜索半径,提高搜索精度和收敛速度,考虑外部档案解的开发次数,合理选择精英解来引导进化. 在11种测试函数上与其他几种算法对比的仿真结果表明,所提出算法具有较好的分布性和收敛性.

**关键词:** 多目标优化; 人工蜂群算法; 镜像初始化; 阈值搜索; Pareto支配关系

中图分类号: TP273

文献标志码: A

## A multi-objective artificial bee colony based on limit search strategy

ZHAO Xin-qiu<sup>1,2†</sup>, DUAN Si-yu<sup>1</sup>, MA Xue-min<sup>1</sup>

(1. Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China; 2. National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Yanshan University, Qinhuangdao 066004, China)

**Abstract:** In order to balance the local exploitation ability and the global exploration ability of artificial bee colony, and accelerate the rate of convergence, a multi-objective artificial bee colony based on the limit search (LSABC) strategy is proposed. Firstly, a chaotic mirroring strategy is proposed in the initialization phase to ensure diversity and superiority. Then, the radius of search is dynamically adjusted while the employed bees and the onlooker bees searching according to the individual limit, which improves the search accuracy and convergence speed. The development times of the external archive are combined to choose elite solution so as to guide evolution reasonably. Finally, experiments are conducted on a set of 11 benchmark functions, and the results show that the proposed algorithm performs well in convergence and distribution.

**Keywords:** multi-objective optimization; artificial bee colony; chaotic mirroring initialization; limit search strategy; Pareto dominance

## 0 引言

随着工程实践的开展和科技水平的进步,在优化过程中往往需要面对多种相互冲突的优化目标,需要多目标算法来获得一组权衡的解,此类问题为多目标优化问题(MOP). 目前,基于仿生学提出的多目标粒子群(MOPSO)算法以及基于分解MOEA/D等一系列经典算法已应用于解决工程实践中的优化问题,并取得了良好的效果.

人工蜂群算法(artificial bee colony)是Karaboba提出的基于仿生学群智能算法,通过模拟蜜蜂觅食的过程,完成一系列优化过程<sup>[1-2]</sup>. 由于其具有结构简单易于实现的优点,已广泛应用于生产调度、路径规划以及电力系统优化中. 起初,人工蜂群算法大范围应

用于单目标优化问题. 文献[3]受差分进化算法的启发提出了IABC算法,通过混沌算子和反向学习的方法对种群进行初始化,同时采用两种不同搜索模式解决单目标问题,获得了良好的收敛性;文献[4]提出一种多搜索策略协同进化的人工蜂群算法,通过动态调节搜索维度并结合不同搜索策略,提高搜索效率的同时,平衡了算法的局部搜索能力和全局搜索能力;文献[5-6]提出的多目标蜂群算法具有里程碑的意义,该算法采用自适应网格的方法对外部档案进行维护,并利用精英蜂引导进化过程,加快了收敛速度并丰富了种群的多样性;文献[7]提出一种基于分解的人工蜂群算法,将分解的思想引入蜂群算法中,前者在一组权重向量的帮助下能够在解决方案中保持良好

收稿日期: 2018-11-05; 修回日期: 2019-04-03.

基金项目: 河北省自然科学基金项目(F2016203249).

责任编辑: 夏元清.

†通讯作者. E-mail: zxq@ysu.edu.cn.

的多样性,后者具有快速的收敛速度,求解标量优化问题时更加高效,并通过一组测试函数验证了所提出的算法能够取得一组分布更好的Pareto解集;文献[8]将种群初始化策略应用到K-means算法,通过实验验证了改进算法对数据聚类的高效性;文献[9]提出了一种基于网格法的人工蜂群算法,并将支配个体数量作为跟随蜂阶段适应度值的评判标准,进一步拓展了算法解决多目标优化问题的能力;文献[10]提出了一种基于快速非支配排序和群体选择策略的蜂群算法,并通过一种新型适合度方法计算跟随蜂阶段的选择概率,结果表明该算法对比其他算法能够在较短时间内收敛到真实前沿;文献[11]提出了一种新型精英策略的人工蜂群算法,并利用个体的拥挤距离来维护外部档案,运用精英策略成功避免了算法早熟收敛的问题;文献[12]提出了一种基于指标的 $\varepsilon$ -MOABC算法,通过质量指标 $I_{\varepsilon+}$ 确定解的质量,解决了不同解在同一帕累托面的选择问题,结果表明算法在处理多目标问题和高维问题上均能取得较好的效果;文献[13]通过ABC算法和基于作业排序的方式解决炼钢过程实际混合车间的调度问题,其中包含相邻解生成方式和两个增强策略,根据实际生产情况进行的实验结果表明,所提出的ABC算法比其他算法效果更优。

基于以上分析,本文在以往研究的基础上,提出一种基于阈值搜索的人工蜂群算法(LSABC)。在初始化阶段,通过使用混沌镜像初始化保证获得一组更优的初始化解集,以加快收敛速度;在引领蜂阶段,根据蜜源的开采程度调节搜索范围,平衡算法的局部搜索和全局搜索能力;在跟随蜂阶段,提出一种基于外部档案个体拥挤距离和阈值的选择方式,合理利用外部档案精英个体引导种群进行更新;在外部档案的维护上,将进化过程中产生的非支配解和精英解集进行非支配排序,并且对极端解进行变异操作,合理运用外部档案,将其作为最后的输出结果。针对ZDT系列和DTLZ系列中11个测试函数,通过与其他5种经典算法进行对比表明,LSABC算法在收敛速度及分布性方面有一定提升。

## 1 多目标优化问题

包含多个目标优化问题的数学模型如下:

$$\begin{aligned} \min F_m(x) &= \{f_1(x), f_2(x), \dots, f_m(x)\}. \\ \text{s.t. } g_j(x) &\leq c_j, \quad j = 1, 2, \dots, J; \\ h_k(x) &= 0, \quad k = 1, 2, \dots, K; \\ x_i^{(L)} &\leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n. \end{aligned} \quad (1)$$

其中: $x = (x_1, x_2, \dots, x_n) \in X \subset R_n$ 为 $n$ 维决策向量; $R_n$ 为 $n$ 维决策变量的可行域; $m$ 为优化目标个数; $g(x)$ 和 $h(x)$ 分别为不等式约束和等式约束, $J$ 和 $K$ 分别为不等式约束和等式约束的个数,当 $J$ 和 $K$ 为0时,整个优化问题即为无约束优化问题; $x_i^{(L)}$ 和 $x_i^{(U)}$ 为第 $i$ 维决策空间的上下界。

## 2 标准蜂群算法

人工蜂群算法(artificial of bee colony, ABC)是一种群体智能模型,是通过模拟自然界中蜂群寻找蜜源过程的仿生智能算法。在整个过程中,蜂群根据不同的分工被分为3类:引领蜂(employed foragers, EF)、跟随蜂(onlookers)和侦查蜂(scouts)。3类蜜蜂在整个过程中相互分享信息,促进整个群体进化,完成整个寻优过程。

在使用ABC算法求解问题时,需要对引领蜂位置进行更新,即

$$v_{ij} = x_{ij} + r_{ij}(x_{ij} - x_{kj}). \quad (2)$$

其中: $k \in 1, 2, \dots, \text{NP}/2$ 且 $k \neq i, j \in 1, 2, \dots, D; r_{ij} \in (-1, +1)$ 。在整个搜索过程中,跟随蜂根据引领蜂分享的信息基于蜜源质量使用轮盘赌策略选择一个合适的蜜源进行开采,具体公式如下:

$$p_i = F_i / \sum_{i=1}^{\text{NP}/2} F_i, \quad (3)$$

其中 $F_i$ 为适应度值。以最小化问题为例,其计算公式为

$$F_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0; \\ 1 + |f_i|, & f_i < 0. \end{cases} \quad (4)$$

其中 $f_i$ 为第 $i$ 个解的优化目标函数值。选择优异的蜜源后,跟随蜂按照式(2)对蜜源进行深度开发。引领蜂和跟随蜂完成更新过程,通过对适应度进行贪婪选择,确定是否保留新产生的解。最后,若某个个体循环更新次数达到阈值(limit),但其解的质量还没有得到改善,则放弃该蜜源,引领蜂转化为侦查蜂,对蜜源进行重新初始化,产生一个新个体替代原有个体。

## 3 基于阈值搜索的人工蜂群算法

ABC算法在求解多目标优化问题时,由于同时存在多个互不支配的解,个体通过信息交流搜索新蜜源,整个过程随机性较大。ABC算法具有较强的全局搜索能力,但并没有得到充足的开发以找到最优解,算法局部搜索能力较差,接近最优解时开发效率明显降低,导致算法搜索能力强开发能力弱。在蜜源开发末期,如果继续使用式(2)对其进行开发,则蜜源的质

量基本得不到改善,整体多样性缺失.因此,在初始化阶段如何通过更高效的方法赋予个体更好的进化潜力、在引领蜂和跟随蜂阶段如何根据蜜源自身特点进行进化提高收敛速度、如何通过概率选择精英引导种群进化以改善人工蜂开发能力弱的缺点并获得一组分布性良好的Pareto解集是研究的主要目的.下面将详细介绍整个算法的实现过程.

### 3.1 混沌镜像初始化策略

在利用混沌映射<sup>[14]</sup>初始化种群时,由于混沌映射的随机性和对初始条件的敏感性,可以提取决策空间信息以增加种群的多样性.本文采用正弦公式迭代产生混沌变量,混沌初始化公式如下所示:

$$\begin{aligned} \text{ch}_1 &= \text{rand}(0, 1), \\ \text{ch}_{k+1} &= \sin(\pi \text{ch}_k), \\ x_{ij} &= x_j^{(L)} + \text{ch}_{kj}(x_j^{(U)} - x_j^{(L)}). \end{aligned} \quad (5)$$

其中:  $\text{ch}_1$  为范围在  $0 \sim 1$  之间的随机数,  $k$  为迭代计数器,  $k = 1, 2, \dots, D - 1$ .

在初始化过程中,假设蜜源  $i$  的位置为  $x_i = (x_{i1}, x_{i2}, \dots, x_{id-1}, x_{id})$ , 通过镜像操作后,新的镜像解位置  $\text{mir}_i$  可定义为  $\text{mir}_i = (x_{id}, x_{id-1}, \dots, x_{i2}, x_{i1})$ . 当某一维数值  $x_{ij}$  超出上下界时,使用正则化和反正则化操作产生新的  $x_{ij}$ .

本文将混沌镜像初始化策略应用于初始化过程,具体步骤如下.

step 1: 按照式(6)混沌初始化产生种群  $A$ , 种群个数为  $\text{NP}$  个.

step 2: 将种群  $A$  通过镜像操作产生种群  $B$ .

step 3: 根据  $A$  和  $B$  的适应度值选择  $\text{NP}$  个较优的个体作为初始种群.

### 3.2 局部搜索和精英群体引导策略

在引领蜂和跟随蜂阶段,根据蜜源的开采情况,在其位置邻域附近展开搜索,从而加快算法末期的收敛速度,增强其开发能力.本文提出一种基于蜜源自身阈值和外部种群密度分布的概率选择方式,充分考虑个体的开发情况和分布关系,通过综合考量个体质量以获得具有良好收敛性和分布性的Pareto最优解集.

#### 3.2.1 局部搜索机制

在引领蜂和跟随蜂阶段,通过式(2)的进化方式,原始蜜源每次变化的维度只有一个,在进化过程中信息交换明显缺乏效率.通过引用一个参数  $W$ <sup>[15]</sup> 控制每次进化需要改变的维度,新搜索公式如下所示:

$$v_{iw} = x_{iw} + r_{iw}(x_{iw} - x_{kw}). \quad (6)$$

其中:  $w = \{1, 2, \dots, W\}$ , 参数  $W$  控制个体每次进化过程中变化维度的数目,  $W = \{1, 2, \dots, D\}$ .

此外,在进化初期,由于蜜源  $x_i$  到真实Pareto前沿距离较远,采用较大的搜索半径可以加快收敛速度;当蜜源进化到末期,距离真实Pareto前沿较近,只需较小的搜索半径以提高搜索精度.所以,合理地选择搜索半径可以明显加快收敛速度.本文采用搜索半径随个体阈值变化的策略,即搜索半径随  $x_i.\text{limit}$  的增加而递减,半径公式为

$$r_i = \Phi_m + \cos(\pi x_i.\text{limit}/2\text{limit}). \quad (7)$$

其中:  $\Phi_m$  为搜索半径的基础值,  $x_i.\text{limit}$  为蜜源当前的阈值,  $\text{limit}$  为蜜源开发阈值.

在跟随蜂阶段采用精英个体引导种群进化的方式,蜜蜂位置更新方程为

$$v_{iw} = \text{elite}_{iw} + r_{iw}(\text{elite}_{iw} - x_{iw}), \quad (8)$$

其中  $r_{iw} \in [-(\Phi_m + 1), +(\Phi_m + 1)]$ .

通过上述搜索策略可以使蜜源在整个进化过程中快速地收敛到真实前沿,有能力跳出局部最优,在进化末期,可以有效地提高搜索精度,改善求解质量.

#### 3.2.2 精英蜂引导以及概率选择机制

合理选择精英个体有助于提高种群对局部空间的开发能力,便于个体更快速地收敛到Pareto前沿.本文提出一种精英蜂的概率选择方式,充分考虑外部精英个体的分布情况和开发情况,综合考量精英个体的质量以获得一组分布性和收敛性良好的Pareto最优解集.改进后的概率计算为

$$p_i = \frac{1}{1 + e^{-\text{elite}_i.\text{crowd} \times \text{elite}_i.\text{choice}}}. \quad (9)$$

其中:  $\text{elite}_i.\text{crowd}$  为第  $i$  个个体在外部档案中的拥挤距离,  $\text{elite}_i.\text{choice}$  为外部档案中第  $i$  个精英个体引导进化剩余次数.

对于新进入外部档案中的个体,赋予其进化阈值  $\text{choice}$ ,使刚刚进入外部档案中的解更具有引导种群进化的潜力.外部档案中的解被选择去引导种群进化时,其  $\text{elite}_i.\text{choice}$  值相应减1,直至  $\text{elite}_i.\text{limit}$  值变为1,保持不变.当个体拥挤距离和  $\text{elite}_i.\text{choice}$  值越大时,被选择的概率越大.

在进化初期,外部精英个体较少,精英个体  $\text{elite}_i.\text{choice}$  值越大,引导进化的概率也越大;在进化末期,外部档案中个体  $\text{elite}_i.\text{choice}$  值减小为1,个体拥挤距离越大,被选择的机会也越大.整个进化过程,精英个体的选择概率将被限制在  $[0.5, 1]$ . 这样的精英选择机制能够保证充分利用精英个体的引导作用,并根据精英个体的分布情况选择适当的位置进行开

发,有利于提高种群的收敛性和分布性.

### 3.3 外部档案维护策略

传统的外部档案维护策略未能充分利用进化过程中产生的解,部分优质解由于不能支配原始解而被舍弃.本文采用一种新的外部档案维护策略,首先将初始化非支配的个体加入外部档案,引导种群进化.当外部档案没达到最大时,选择互不支配的解加入外部档案.由于该操作可能导致极端解一直存在于外部档案,造成外部档案中的解不能得到更新,在更新外部档案时,对极端解进行变异操作,以保证极端解已经被完全开发.当存储解的数量超出外部存档的规模时,计算拥挤距离,删除拥挤距离最小的解,直至满足规模要求.将最终的外部存档解集作为最后的输出,决定每一代极端解的变异因子为

$$\begin{aligned} \text{Mu}(t) = & \\ & 0.5 + \text{round}\left(\frac{\text{FEAS} - \text{cur}}{\text{FEAS}}\right) \sin\left(\frac{\pi \cdot \text{cur}}{2\text{FEAS}}\right)^2 + \\ & \text{round}\left(\frac{\text{cur}}{\text{FEAS}}\right) \cos\left(\frac{\pi \cdot \text{cur}}{2\text{FEAS}}\right)^2. \end{aligned} \quad (10)$$

其中:  $\text{cur}$  为当前评价次数,  $\text{FEAS}$  为算法的最大评价次数,  $\text{round}$  为取整函数.

当个体加入到外部档案时,赋予其较小的变异机会;当种群进化到中期时,个体获得的变异机会变大,加速外部档案的收敛速度;到末期时,外部档案个体趋于真实PF,变异机会变小,以保持外部档案的稳定性.具体的变异思路<sup>[16]</sup>如下:

$$V_{ij} = \begin{cases} \text{normrnd}(\mu_1, \sigma_1) \times V_{ij}, & \text{rand} > \text{Mu}(t); \\ (1 + \text{rand}) \times V_{ij} + \text{normrnd}(\mu_1, \sigma_1), & \text{otherwise.} \end{cases} \quad (11)$$

其中  $\text{normrnd}(\mu_1, \sigma_1)$  为服从均值为  $\mu_1$ 、标准差为  $\sigma_1$  的正态分布.通过在外部分档中加入变异操作,可以有效地抑制极端解,更快地收敛到真实前沿PF,并具有较强的能力跳出局部最优.

### 3.4 算法流程

算法的具体流程如下.

step 1: 初始化种群数量和外部档案最大存储个数为  $\text{NP}$ , 引领蜂和跟随蜂各为  $\text{NP}/2$ , 最大评价次数为  $\text{FEAS}$ , 侦查蜂最大淘汰次数为  $\text{limit}$ , 外部档案精英个体最大淘汰次数为  $\text{elite.limit}$ , 通过混沌镜像初始化并计算个体适应度值, 根据支配关系选择最优的  $\text{NP}$  个个体, 并选择非支配个体进入外部档案.

step 2: 按照式(6)对引领蜂位置进行更新,更新后如果支配原始蜜源则被保留,如果互不支配则与外

部档案中的个体比较支配关系确定是否保留.

step 3: 按照式(9)计算外部精英个体的选择概率,通过概率选择是否对其进行深度开发,具体进化公式为式(8),进化完成后精英个体的引导次数减1,并对其后代执行与引领蜂相同的保留策略.

step 4: 当引领蜂和跟随蜂产生的子代个体未能被保留时,淘汰次数加1,当达到个体阈值  $\text{limit}$  时,通过式(5)重新产生一个新蜜源替换旧蜜源位置.

step 5: 在每次迭代完成后,对外部档案进行维护,判断是否达到最大的评价次数  $\text{FEAS}$ . 若达到则结束循环并输出外部档案中的个体作为最终结果;否则,转至 step 2.

## 4 算法性能测试及分析

### 4.1 测试函数及性能指标

本文针对两目标和三目标的优化问题进行测试,分别采用测试函数集为  $\text{ZDT}^{[17]}$  和  $\text{DTLZ}^{[18]}$  系列测试函数,并将综合性指标  $\text{inverted generational distance}(\text{IGD})^{[19]}$  作为本算法的评价指标.  $\text{IGD}$  指标反映了算法的真实前沿与所求解构成的前沿之间的距离.反向世代距离  $\text{IGD}$  越小,表明算法获得解集的分布性和收敛性越好.  $\text{IGD}$  的计算公式如下:

$$\text{IGD} = \frac{1}{\text{num}} \sum_{i=1}^{\text{num}} \text{dis}_i. \quad (12)$$

其中:  $\text{num}$  为理想真实前沿PF上解的个数,  $\text{dis}_i$  为第  $i$  个真实前沿PF上点到所求得解集的最小欧氏距离.

### 4.2 仿真实验结果与分析

为了测试本文算法LSABC的性能,选取5个具有代表性的多目标优化算法进行对比,包括多目标蜂群算法MOABC<sup>[9]</sup>、MOPSO算法<sup>[20]</sup>、非支配排序算法NSGAII<sup>[21]</sup>、基于分解的多目标算法MOEA/D<sup>[22]</sup>和SPEA2+<sup>[23]</sup>.LSABC算法的参数设置为: $\Phi_m = 0.4$ ,  $W = 0.2 \times D$ , 侦查蜂最大淘汰次数  $\text{limit} = 100$ , 精英解搜索阈值  $\text{elite.choice} = 100$ ; 另外5种算法涉及的参数参考原文献.对于双目标问题(ZDT系列函数),6种算法种群规模大小均设为  $\text{NP} = 100$ ,  $\text{archivesize} = 100$ , 最大评价次数  $\text{FEAS} = 3 \times 10^4$ , 6种算法分别独立运行30次;对于三目标问题(DTLZ系列函数),6种算法种群规模大小均设为  $\text{NP} = 150$ ,  $\text{archivesize} = 150$ , 最大评价次数  $\text{FEAS} = 1.5 \times 10^5$ , 分别独立运行20次.以上每种算法均采用Inter(R) Core(TM) i5-7500 CPU, 8G RAM, 实验环境为Matlab R2014a.通过统计并比较6种算法的IGD最优值(best)、最差值(worse)、平均值(mean)和标准差(std),结果如表1所示,其中加粗项为同一测试函数中获得的最优值.

表1 IGD值统计

测试函数		算法					
		LSABC	MOPSO	MOABC	MOEA/D	NSGAI	SPEA2+
ZDT1	worse	<b>3.631e-03</b>	2.345e-02	6.667e-02	5.493e-03	5.351e-03	9.862e-02
	best	<b>3.584e-03</b>	1.313e-02	1.740e-02	5.000e-03	4.608e-03	4.987e-02
	mesn	<b>3.631e-03</b>	1.756e-02	3.059e-01	5.193e-03	4.935e-03	7.702e-02
	std	<b>3.018e-05</b>	2.800e-03	1.235e-01	1.242e-04	1.93 e -04	1.360e-02
ZDT2	worse	<b>3.631e-03</b>	2.345e-02	6.667e-02	5.493e-03	5.351e-03	9.862e-02
	best	<b>3.798e-03</b>	1.081e-02	4.774e-02	3.858e-03	4.562e-03	6.196e-02
	mesn	<b>3.847e-03</b>	1.850e-02	7.721e-02	1.569e-01	4.833e-03	1.150e-01
	std	<b>2.866e-05</b>	3.600e-03	2.444e-01	2.489e-01	1.552e-04	2.302e-02
ZDT3	worse	<b>4.636e-03</b>	4.076e-02	4.855e-02	3.063e-01	1.148e-01	1.145e-01
	best	<b>3.684e-03</b>	1.829e-02	2.410e-02	5.525e-02	4.797e-03	4.540e-02
	mesn	<b>3.922e-03</b>	2.615e-02	3.791e-02	1.614e-01	1.341e-02	8.978e-02
	std	<b>1.918e-04</b>	6.134e-03	6.012e-03	9.940e-02	2.190e-02	1.478e-02
ZDT4	worse	<b>3.887e-03</b>	4.555e+00	7.052e-01	5.591e-01	1.663e-02	1.187e+01
	best	<b>3.672e-03</b>	1.277e+00	1.762e-01	8.120e-03	5.903e-03	1.059e+00
	mesn	<b>3.783e-03</b>	3.747e+00	3.809e-01	2.662e-01	9.474e-03	4.903e+00
	std	<b>5.408e-05</b>	7.420e+00	1.203e-01	1.486e-01	2.400e-03	2.190e+00
ZDT6	worse	2.293e-03	3.340e+00	6.475e-03	<b>2.016e-03</b>	3.271e-03	1.340e-01
	best	1.933e-03	3.810e-03	2.887e-03	<b>1.870e-03</b>	2.337e-03	9.843e-03
	mesn	2.032e-03	1.246e-01	2.411e-03	<b>1.893e-03</b>	2.738e-03	4.321e-02
	std	<b>1.551e-04</b>	6.077e-01	8.354e-04	3.412e-04	2.554e-04	3.382e-02
DTLZ1	worse	<b>1.905e-02</b>	1.494e+01	3.021e-02	2.030e-02	3.363e-01	2.770e+01
	best	1.740e-02	3.322e+00	2.258e-02	<b>1.731e-02</b>	2.391e-02	6.891e-01
	mesn	<b>1.803e-02</b>	9.729e+00	2.516e-02	1.849e-02	2.251e-02	2.211e+00
	std	<b>4.474e-04</b>	2.947e+00	9.340e-02	8.735e-04	1.246e-03	2.057e+00
DTLZ2	worse	5.323e-02	1.014e-01	<b>5.313e-02</b>	5.346e-02	6.726e-02	9.813e-02
	best	<b>4.714e-02</b>	8.532e-02	4.794e-02	4.788e-02	5.811e-02	7.199e-02
	mesn	<b>4.914e-02</b>	9.362e-02	4.993e-02	5.067e-02	6.343e-02	8.060e-02
	std	1.327e-03	4.262e-03	1.952e-02	<b>1.419e-04</b>	2.577e-03	7.078e-03
DTLZ3	worse	<b>5.682e-03</b>	1.759e+02	6.876e-01	6.113e-01	7.544e-02	1.839e+02
	best	<b>4.918e-03</b>	1.182e+02	1.706e-01	5.315e-02	6.109e-02	1.472e-01
	mesn	<b>5.189e-02</b>	1.473e+02	2.760e-01	9.333e-02	6.739e-02	8.153e+02
	std	<b>1.786e-03</b>	1.582e+01	1.114e-01	1.243e-01	3.969e-03	4.869e+01
DTLZ4	worse	9.842e-02	2.789e-01	5.313e-02	<b>6.252e-02</b>	6.638e-02	9.606e-02
	best	5.958e-02	1.470e-01	4.794e-02	<b>5.269e-02</b>	5.937e-02	6.988e-02
	mesn	7.451e-02	2.081e-01	2.516e-02	<b>5.762e-02</b>	6.282e-02	8.279e-02
	std	1.074e-02	3.362e-02	9.340e-02	2.468e-03	<b>1.991e-03</b>	6.853e-03
DTLZ5	worse	<b>3.144e-03</b>	2.220e-02	6.081e-03	2.032e+00	4.182e-03	1.669e-02
	best	<b>2.447e-03</b>	1.684e-02	4.218e-03	8.898e-01	3.397e-03	7.470e-03
	mesn	<b>2.749e-03</b>	1.924e-02	4.962e-03	1.358e+00	3.736e-03	1.106e-02
	std	<b>2.050e-04</b>	1.373e-03	7.345e-04	6.714e-02	1.822e-02	2.534e-03
DTLZ7	worse	<b>6.935e-02</b>	7.179e-02	7.219e-02	2.010e-01	7.967e-01	2.266e-01
	best	<b>5.262e-02</b>	5.940e-02	5.608e-02	1.556e-01	6.716e-02	1.494e-01
	mesn	<b>6.157e-02</b>	6.494e-02	6.375e-02	1.747e-01	1.242e-01	1.806e-01
	std	<b>1.881e-03</b>	3.267e-03	4.978e-02	1.286e-02	1.700e-01	2.353e-02

### 4.3 仿真实验结果与分析

由表1可见,无论是ZDT系列函数还是DTLZ系列函数的优化问题,本文算法都能获得一组接近真实前沿且分布性良好的解集,具有良好的稳定性.在双目标问题上,LSABC算法取得了较好的结果.通过图1和图2的Pareto前沿图可以得出,LSABC算法无论在凹凸问题上还是非连续问题上都具有良好的分布性和收敛性.通过计算出来的IGD指标数值以及

Pareto前沿图表明,LSABC算法在收敛到真实PF的同时具有良好的分布性.根据表1提供的标准差(std)和均值(mean)数据可以看出,LSABC算法与其他5种算法对比具有较好的稳定性和收敛性.

在三目标测试函数上,LSABC算法相较于其他5种算法,在DTLZ1、DTLZ5测试函数上具有较好的表现.由图1可见,LSABC算法相较于其他5种算法所获得的前沿面分布性较好.由表1的DTLZ系列测试

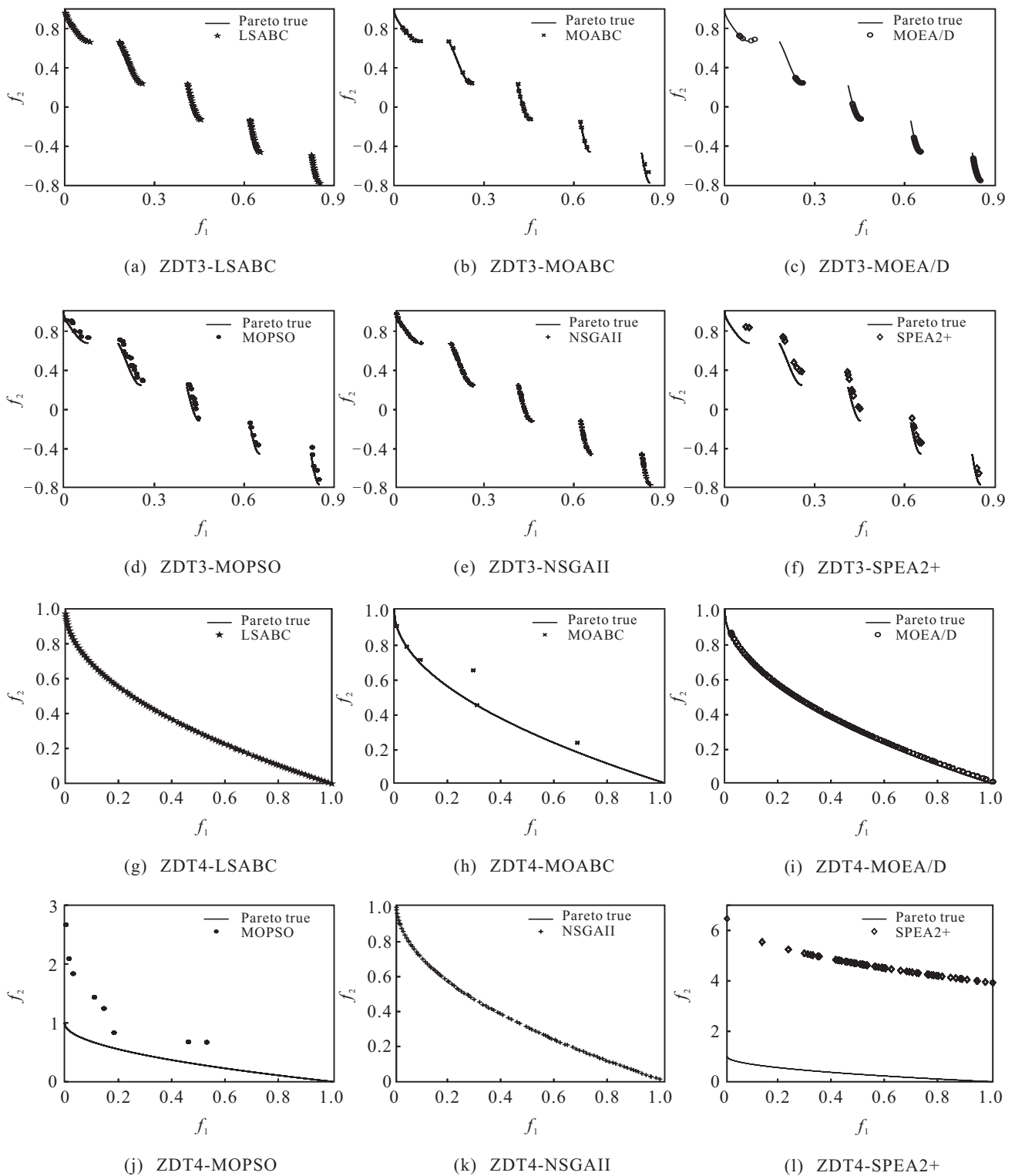


图1 ZDT算法部分测试函数



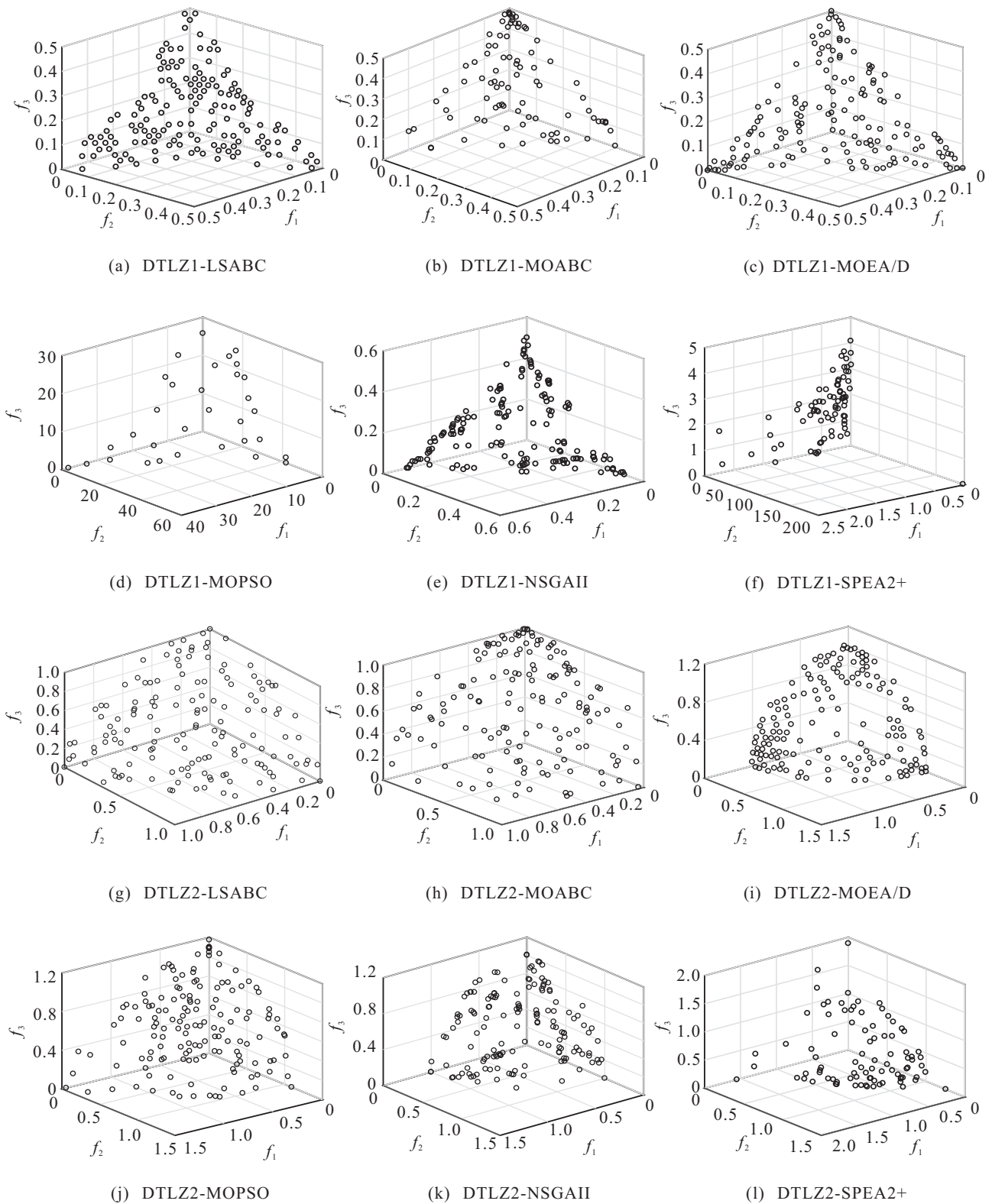


图2 DTLZ算法部分测试函数PF图

函数的标准差(std)可以清晰地看到,LSABC算法具有良好的稳定性,且具有更小的IGD均值(mean).

总体而言,LSABC算法相较于其他算法可以更好地收敛到Pareto前沿.根据表1、图1和图2分析得出,本文算法不仅在处理凹凸优化问题上能够获得较好的结果,在多峰函数的优化过程中,相较于

MOPSO、MOABC、MOEA/D、NSGAI、SPEA2+算法也获得了较优的IGD值,可以在保证收敛性的同时获得良好的分布性.

#### 4.4 初始化策略性能对比实验分析

为了证明本文所提出初始化策略的有效性,设置两组用于多目标算法初始化的对比实验,包括随



机初始化和混沌初始化法. 对于两目标问题(ZDT函数), 总评价次数FEAS = 1000, 独立运行30次; 对于三目标问题(DTLZ函数), 总评价次数FEAS = 10000, 独立运行20次, 参数设置同4.2节. IGD平均值如表2所示, 其中加粗字体为各算法在同一测试问题上获得的最优值, “+”、“=”、“-”表示同一测试问题中IGD平均值在3种初始化方法中分别处于首位、次位、末位的统计结果.

表2 不同初始化方法IGD值统计

测试函数	初始化方法		
	混沌镜像	混沌	随机
ZDT1	1.803e-01	<b>1.691e-01</b>	1.971e-01
ZDT2	<b>1.932e-01</b>	2.054e-01	2.485e-01
ZDT3	<b>2.013e-01</b>	2.191e-01	2.493e-01
ZDT4	<b>7.359e-01</b>	8.335e-01	8.358e-01
ZDT6	<b>2.612e-01</b>	3.435e-01	6.402e-01
DTLZ1	<b>3.805e+00</b>	4.277e+00	3.911e+00
DTLZ2	5.469e-01	<b>5.234e-01</b>	5.670e-01
DTLZ3	8.979e+02	1.959e+02	<b>8.205e+01</b>
DTLZ4	<b>2.382e-01</b>	6.626e-01	3.270e-01
DTLZ5	4.549e-01	4.127e-01	<b>1.204e-02</b>
DTLZ7	5.830e-01	7.494e-02	<b>7.402e-02</b>
+ / = / -	6 / 2 / 3	2 / 1 / 2	3 / 2 / 6

由表2可见, 本文提出的混沌镜像初始化方式对比其他2种初始化方法, 在进化初期可以获得一组接近前沿且分布良好的解集, 使得算法在优化过程中具有优势.

#### 4.5 不同策略性能对比实验分析

为了证明改进策略对算法的提升作用, 设计4种自身对比实验. LSABC-I为采用一般种群进化公式的蜂群算法; LSABC-II为采用删除外部档案极端解变异策略的蜂群算法; LSABC-III为原始跟随蜂概率选择机制的算法. 上述算法除了在某些策略上不同外, 其余各项实验参数均与4.1节数据一致, 实验结果如表3所示. 由表3可见, LSABC算法在11个测试函数中取得了10个IGD最优平均值以及6个最优标准差值, 表明所提出算法对于不同的测试函数, 在提高优化性能的同时也具备较好的稳定性; 在算法性能上, 与LSABC-I相比, IGD值在11个测试函数上全部优于LSABC-I算法, 并且在标准差上8个测试函数优于LSABC-I算法, 表明阈值搜索公式能够增强算法局部寻优能力, 在种群进化后能够提高收敛精度, 优于原始进化方式; 与LSABC-II算法相比, LSABC算法的IGD平均值和标准差在11个测试函数全部胜出, 一定程度上表明对外部极端解变异操作可以增强种群跳出局部最优的能力, 避免陷入局部最优; 与LSABC-III算法相比, LSABC算法的IGD平均值

在10个测试函数上胜出, 标准差在7个测试函数上胜出, 表明改进的跟随蜂概率机制在提高种群多样性方面起到了积极的作用. 上述实验说明, LSABC算法不仅能够平衡算法的局部和全局寻优能力, 也能够提高算法的多样性.

表3 不同策略IGD值统计

测试函数		算法			
		LSABC	LSABC-I	LSABC-II	LSABC-III
ZDT1	Mean	<b>3.631e-03</b>	3.748e-02	3.747e-03	3.702e-03
	Std	<b>3.018e-05</b>	3.121e-05	3.9784e-05	3.050e-05
ZDT2	mesn	3.847e-03	3.854e-03	3.849e-03	<b>3.837e-03</b>
	std	<b>2.866e-05</b>	2.360e-04	2.444e-01	3.124 e-05
ZDT3	mesn	<b>3.922e-03</b>	4.425e-03	4.422e-03	4.745e-03
	std	<b>1.918e-04</b>	1.949e-04	7.086e-04	1.962e-04
ZDT4	mesn	<b>3.783e-03</b>	4.157e-03	8.874e-02	2.602e-02
	std	<b>5.408e-05</b>	6.123e-05	5.666e-03	3.499e-02
ZDT6	mesn	<b>2.032e-03</b>	1.246e-01	2.395e-03	2.364e-03
	std	1.551e-04	1.349e-04	1.349e-04	<b>1.263e-04</b>
DTLZ1	mesn	<b>1.803e-02</b>	2.035e-02	2.516e-02	2.223e-02
	std	<b>4.474e-04</b>	4.9001e-04	9.340e-02	4.416e-04
DTLZ2	mesn	<b>4.914e-02</b>	6.012e-02	5.063e-02	4.950e-02
	std	1.327e-03	4.262e-03	1.242e-03	<b>7.280e-04</b>
DTLZ3	mesn	<b>5.189e-02</b>	1.092e+00	2.760e-01	9.174e-02
	std	1.786e-03	<b>9.530e-04</b>	1.114e-01	1.162e-03
DTLZ4	mesn	<b>7.451e-02</b>	1.523e-01	2.124e-01	2.037e-01
	std	1.074e-02	3.344e-02	9.340e-02	4.311e-02
DTLZ5	mesn	<b>2.749e-03</b>	2.841e-03	4.962e-03	2.870e-03
	std	2.050e-04	<b>1.348e-04</b>	7.345e-04	1.503e-04
DTLZ7	mesn	<b>6.157e-02</b>	6.360e-02	6.625e-02	6.193e-02
	std	<b>1.881e-03</b>	3.709e-03	4.978e-02	3.949e-03

#### 4.6 算法收敛性分析

图2为LSABC算法与其余5种对比算法的IGD指标下降趋势. 为了能够直观地看出变化趋势, 选取IGD的对数作为纵坐标, ZDT和DTLZ系列函数所记录的数据分别为30次和20次独立实验所得的平均值. ZDT系列函数每1000次评价记录一次数据; DTLZ系列函数每5000次评价记录一次数据. 由图2可见, LSABC算法评价次数达到10000次时, 所有函数基本收敛到Pareto前沿, 在所有测试函数上都可以快速地收敛到前沿且具有较好的稳定性; 对比其他算法, 除了ZDT6外, ZDT系列函数均取得最小的IGD值. 在DTLZ系列中, DTLZ2-3、DTLZ5和DTLZ7相较于其他5种对比算法, 均取得了较好的收敛速度和IGD平均值. 在DTLZ4测试函数上, 相较于其他5种对比算法, 在收敛速度上差于NSGAI和

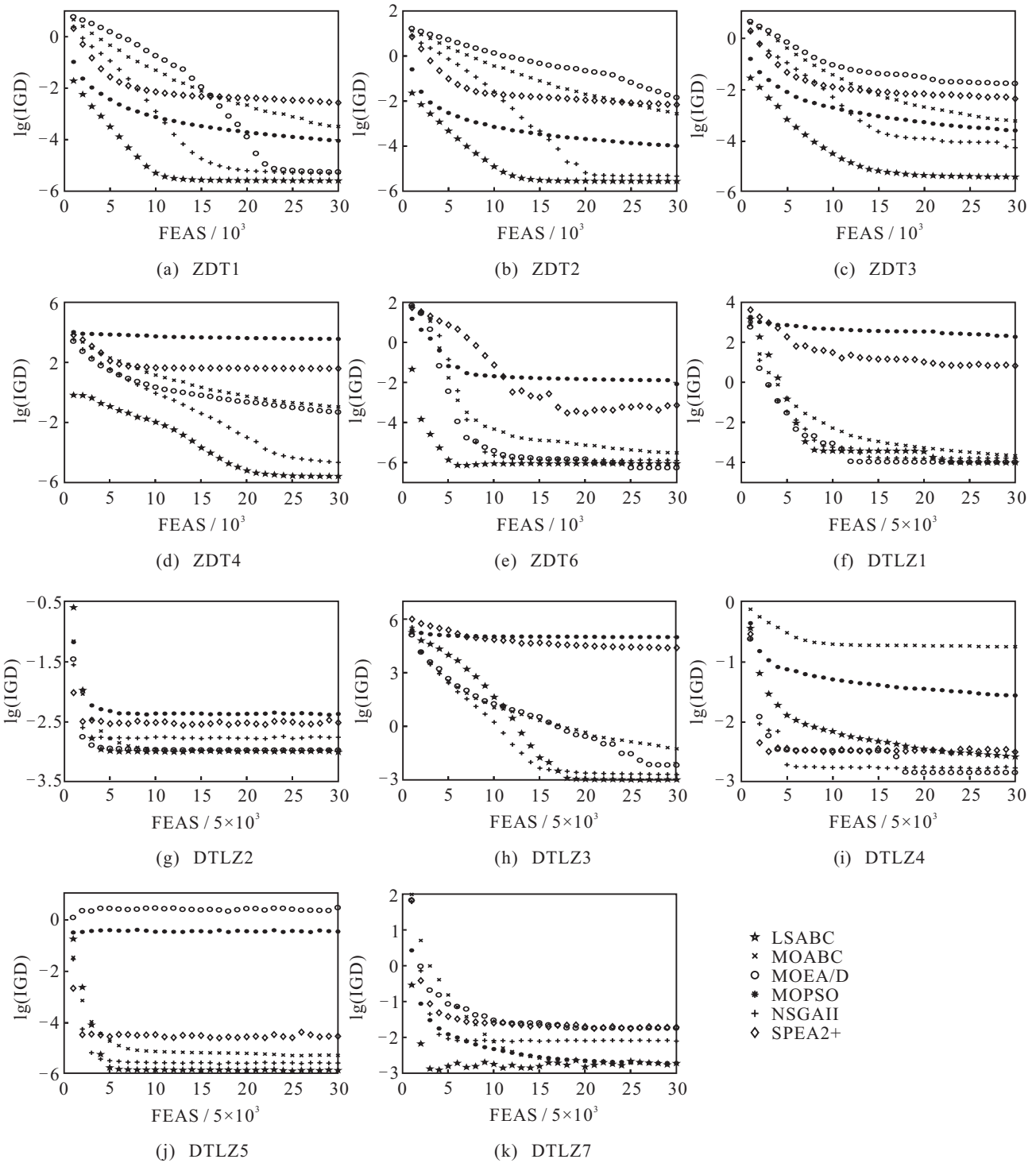


图3 各算法IGD收敛图

MOEA/D和SPEA2+算法. 结合表1和图2可见, 本文提出的LSABC算法在11个测试函数中, 均取得了较好的结果, 由此说明LSABC算法在平衡收敛速度和求解多样性的同时获得了良好的性能.

### 5 结论

为了加快种群的收敛进程并平衡算法的全局搜索能力和局部开发能力, 本文提出了一种基于阈值搜索的多目标蜂群算法. 在初始化阶段, 提出了一种新

的混沌镜像初始化方式, 使种群在进化初期便能够获得一组分布性良好且接近Pareto前沿的解集, 一定程度上加快了收敛速度. 在引领蜂和跟随蜂阶段, 通过考虑个体阈值的方式选择合适的搜索半径, 加快了算法前期搜索速度以及后期搜索精度. 在跟随蜂阶段, 考虑外部档案中精英解的分布状况对种群进行进化引导, 合理地利用外部档案解的信息, 提高了整个算法的分布性. 通过一种新的变异方式防止外部档案

中的极端解引导,使种群具有较强的能力跳出局部最优.最后,将进化过程中的互不支配解加入到外部档案中进行非支配排序,避免了多样性丢失问题的发生,使得所获得的真实前沿PF分布更加均匀.通过与5种算法在同一环境下的仿真结果对比,表明了本文所提出算法在分布性和收敛性上都有所提升.因此,LSABC算法是一种整体性能更优且鲁棒性更强的多目标优化方法.

#### 参考文献(References)

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Turkey: Computers Engineering Department, Erciyes University, 2005.
- [2] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm[J]. Applied Mathematics & Computation, 2009, 214(1): 108-132.
- [3] Gao W Y, Liu S Y. Improved artificial bee colony algorithm for global optimization[J]. Information Processing Letters, 2011, 111(17): 871-882.
- [4] 王志刚, 尚旭东, 夏慧明, 等. 多搜索策略协同进化的人工蜂群算法[J]. 控制与决策, 2018, 33(2): 235-241. (Wang Z G, Shang X D, Xia H M, et al. Artificial bee colony algorithm with multi-search strategy cooperative evolutionary[J]. Control and Decision, 2018, 33(2): 235-241.)
- [5] Hedayatzadeh R, Hasanizadeh B, Akbari R, et al. A multi-objective artificial bee colony for optimizing multi-objective problems[C]. International Conference on Advanced Computer Theory and Engineering. Chengdu: IEEE, 2010, 5: 277-281.
- [6] Akbari R, Hedayatzadeh R, Ziarati K, et al. A multi-objective artificial bee colony algorithm[J]. Swarm & Evolutionary Computation, 2012, 2(1): 39-52.
- [7] Xiang Y, Zhou Y, Tang L, et al. A decomposition-based many-objective artificial bee colony algorithm[J]. IEEE Transactions on Cybernetics, 2017, 99: 1-14.
- [8] Kumar A, Kumar D, Jarial S K. A novel hybrid  $K$ -means and artificial bee colony algorithm approach for data clustering[J]. Decision Science Letters, 2018, 7(1): 65-76.
- [9] Akbari R, Hedayatzadeh R, Hassanizadeh B, et al. A multi-objective artificial bee colony algorithm[J]. Swarm & Evolutionary Computation, 2012, 2(1): 39-52.
- [10] Huo Y, Zhuang Y, Gu J, et al. Elite-guided multi-objective artificial bee colony algorithm.[J]. Applied Soft Computing, 2015, 32(C): 199-210.
- [11] Xiang Y, Zhou Y, Liu H. An elitism based multi-objective artificial bee colony algorithm[J]. European Journal of Operational Research, 2015, 245(1): 168-193.
- [12] Luo J, Liu Q, Yang Y, et al. An artificial bee colony algorithm for multi-objective optimisation[J]. Applied Soft Computing, 2017, 50: 235-251.
- [13] Pan Q K, Wang L, Mao K, et al. An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process[J]. IEEE Transactions on Automation Science & Engineering, 2013, 10(2): 307-322.
- [14] Alatas B. Chaotic bee colony algorithms for global numerical optimization[J]. Expert Systems with Applications, 2010, 37(8): 5682-5687.
- [15] Yi X, Zhou Y, Tang L, et al. A decomposition-based many-objective artificial bee colony algorithm[J]. IEEE Transactions on Cybernetics, 2019, 19(1): 287-300.
- [16] 杨黎明, 侯新培, 崔慧慧, 等. 基于融合多策略改进的多目标粒子群优化算法[J]. 控制与决策, 2018, 33(2): 226-234. (Yang J M, Hou X P, Cui H H, et al. Improved multi-objective particle swarm optimization algorithm based on integrating multiply strategies[J]. Control and Decision, 2018, 33(2): 226-234.)
- [17] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.
- [18] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Proceedings of the Congress on Evolutionary Computation. Honolulu: IEEE, 2002: 825-830.
- [19] Ishibuchi H, Masuda H, Tanigaki Y, et al. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems[C]. IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making. Orlando: IEEE, 2014: 170-177.
- [20] Coello C C, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization[C]. Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002: 1051-1056.
- [21] 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.
- [22] Zhang Q, Li H. A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [23] Kim M, Hiroyasu T, Miki M, et al. SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2[J]. Parallel Problem Solving from Nature-PPSN VIII, 2004, 3242(4): 742-751.

#### 作者简介

赵新秋(1969—),女,副教授,从事冶金机械综合自动化、智能控制及工程应用等研究, E-mail: zxq@ysu.edu.cn;  
段思雨(1995—),男,硕士生,从事多目标优化算法的研究, E-mail: dsy001ysu@163.com;  
马学敏(1994—),女,硕士生,从事动态多目标优化算法的研究, E-mail: mxm0217@163.com.

(责任编辑: 郑晓蕾)