

## 采用数量级阈值与二维信息排序策略的NSGA-II-DE算法

杨景明, 侯宇浩, 孙浩, 赵志伟

(燕山大学 a. 电气工程学院, b. 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:** 为提高算法NSGA-II-DE解决含有复杂 Pareto 解集优化问题的性能, 分析原NSGA-II中拥挤度计算公式和排挤机制的缺陷, 并以NSGA-II-DE算法为基本框架, 将传统拥挤度排序策略改为包含有角度信息与伪半径的二维信息排序策略. 在拥挤度排挤机制中加入数量级阈值的干预, 提出改进算法2D-Thr. 选取多样性、收敛度和分布度3个评价指标进行量化计算, 并与NSGA-II-DE、原NSGA-II、MACPSO进行比较. 仿真结果表明, 改进算法不仅有效继承了原算法优良的收敛性, 而且提高了 Pareto 前沿的分布度.

**关键词:** 改进的非支配排序遗传算法; 差分进化算法; 二维信息; 极端信息; 数量级阈值

**中图分类号:** TP18

**文献标志码:** A

### Modified NSGA-II-DE with two-dimensional information ordering strategy and magnitude threshold

YANG Jing-ming, HOU Yu-hao, SUN Hao, ZHAO Zhi-wei

(a. School of Electrical Engineering, b. National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Yanshan University, Qinhuangdao 066004, China. Correspondent: HOU Yu-hao, E-mail: hyh8935444@qq.com)

**Abstract:** To improve the performance of NSGA-II-DE solving the optimization problem with the complex Pareto solution set, the defection of the crowding distance formula and crowding distance mechanism in the NSGA-II is analyzed. Taking NSGA-II-DE as the basic frame, the crowding-distance sorting method is changed into the two-dimensional information ordering strategy of including angle and pseudo radius. The intervention of orders of magnitude threshold is joined in the crowding distance mechanism, and the improved algorithm(2D-Thr) is proposed, which is compared with NSGA-II-DE, NSGA-II and MACPSO by using the quantitative calculation of three evaluation indexes: the degree of diversity, convergence and spacing. The simulation results show that the improved algorithm not only inherits the excellent convergence of the original algorithm, but also improves the distribution of the Pareto front.

**Keywords:** NSGA-II; DE; two-dimensional information; extreme information; orders of magnitude threshold

## 0 引言

在科学研究和工程实践中, 越来越多的领域涉及到多目标优化问题. 由于多目标优化问题的复杂性、高度非线性和优化算法的自身局限性, 使得各种多目标优化算法各有优劣<sup>[1]</sup>. 多目标进化算法(MOEA)作为解决多目标优化问题的有效方法之一, 近年来受到越来越多的国内外专家和学者的关注. Knowles等<sup>[2]</sup>提出了 Pareto 存档 PAES 算法, 采用单一父代产生单一子代的进化策略并添加了网格法以维持结果的多样性. Corne等<sup>[3]</sup>提出了 PESA 算法, 采用

显型空间的超级网格划分方法来维持选择的多样性.

Laumanns<sup>[4]</sup>提出了 SPEA2 算法, 利用外部种群存储目前为止发现的非支配解集, 然后采用确定性聚类技术保留精英解. Zhang等<sup>[5]</sup>提出了基于分解的多目标进化算法 MOEA/D, 将传统的数学规划方法与进化算法结合起来.

Deb等<sup>[6]</sup>提出经典的 NSGA-II 算法, 由 NSGA<sup>[7]</sup>算法发展而来, 采用时间复杂度较低的快速非支配排序机制使算法收敛性得到提高, 采用新的排挤机制保证种群分布性. 对 NSGA-II 算法的改进可以分

收稿日期: 2015-08-08; 修回日期: 2016-01-11.

基金项目: 河北省高等学校创新团队领军人才培养计划项目(LJRC013); 河北省科技支撑计划项目(13211817); 国家冷轧板带及装备工程研究中心开放课题项目(2012005).

作者简介: 杨景明(1957—), 男, 教授, 博士生导师, 从事冶金机械综合自动化、先进控制及工程应用等研究; 侯宇浩(1990—), 男, 硕士生, 从事冶金机械综合自动化、多目标决策的研究.

为3类<sup>[8-13]</sup>: 对收敛性的提高、对种群多样性的提升、对时间复杂度的降低. Liu 等<sup>[8]</sup>提出了基于子区域搜索的NSGA-II算法, 将搜索域按角度划分为 $n$  ( $n$ 为种群中个体数量)个子区域. Guo等<sup>[9]</sup>提出了模糊NSGA-II算法. 罗辞勇等<sup>[10]</sup>提出了采用循环拥挤排序策略的NSGA-II算法. Li等<sup>[11]</sup>提出了NSGA-II-DE算法, 以NSGA-II为基本框架, 利用差分进化算法(DE)<sup>[14]</sup>中的交叉和变异算子代替原NSGA-II中的模拟二进制交叉算子, 使得算法比原NSGA-II算法具有更精准的收敛性和更好的多样性.

多数针对Pareto前沿分布性的改进都只是收集所要求解函数的一维信息, 使得求解过程不稳定, 从而导致最优解集的时好时坏. 鉴于此, 本文首先分析拥挤距离计算和拥挤机制的缺陷, 然后提出基于二维信息和数量级阈值干预的循环排序策略, 最后通过仿真数据表明所提出算法的有效性, 并对以后的研究提出展望.

## 1 NSGA-II-DE 算法缺陷分析

### 1.1 拥挤距离计算公式分析

NSGA-II-DE的拥挤距离计算如下: 基于非支配排序操作, 将个体按照Pareto支配原则分等级, 然后对需要裁剪的某等级内个体进行拥挤距离的计算, 计算公式为

$$I(i)_{\text{dist}} = I(i)_{\text{dist}} + \frac{I[i+1].m - I[i-1].m}{f_m^{\max} - f_m^{\min}} \quad (1)$$

其中:  $m$  为目标函数个数,  $f_m^{\max}$  和  $f_m^{\min}$  分别为目标函数的最大、最小值,  $I = \text{sort}(I, m)$ ,  $s = |I|$ ,  $i = 2 : (s - 1)$ ,  $I[1]_{\text{dist}} = I[\infty]_{\text{dist}} = \infty$ . 由式(1)可见, 拥挤距离实质是计算个体间目标函数值之差, 如果将每一个目标函数值看成一维信息, 则式(1)可以等效为 $m$ 维目标函数信息叠加成一维信息.

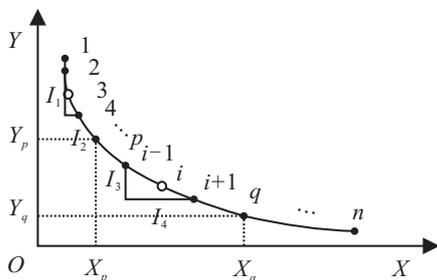


图 1 两种个体拥挤距离比较

图1为两种个体拥挤距离的比较. 由图1可见:  $I_1 = \sqrt{3}$ ,  $I_2 = 1$ ,  $I_3 = I_4 = \sqrt{2}$ , 个体2~个体4与个体 $i-1 \sim i+1$ 的欧氏距离都为2. 设  $f_m^{\max} - f_m^{\min} = 10$ , 根据以上拥挤度计算公式, 个体3的拥挤距离为0.2732, 个体4的拥挤距离为0.2828, 则个体3比个体 $i$ 的拥挤距离值小, 这种情况下个体3比个体 $i$ 更容易被淘汰, 这是导致解集分布不均匀的原因之一.

由图1易得, 个体1~个体 $p$ 在前沿上较为靠近 $Y$ 轴, 个体 $q \sim$ 个体 $n$ 较为靠近 $X$ 轴, 可以规定在个体 $p$ 上方的个体和个体 $q$ 下方的个体为含有极端信息个体, 个体 $p$ 和个体 $q$ 称为极端边界个体. 判断极端边界个体根据实验得到, 在实验中, 若以某个体为分界点, 当此个体以上或以下个体比较稀疏时, 则可确定此个体为极端边界个体. 例如, 规定个体 $p$ 和个体 $q$ 为极端边界个体, 判断个体 $k$ (坐标为 $(X_k, Y_k)$ )是否为含有极端信息个体, 若满足  $Y_k/X_k < Y_q/X_q$  或  $Y_k/X_k < Y_p/X_p$ , 则个体 $k$ 是含有极端信息个体, 否则, 个体 $k$ 是普通个体.

### 1.2 排挤机制分析

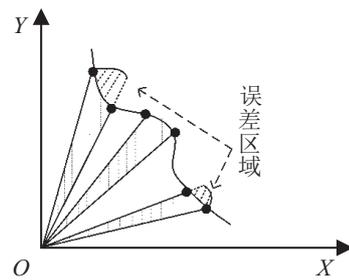
原NSGA-II的排挤机制基于等级与拥挤距离的大小来决定是否保留或淘汰某个体. 首先, 种群经过非支配排序, 个体按照排序后的等级进行分层; 然后, 计算所需裁剪的某等级个体之间的拥挤距离; 最后, 将拥挤距离小的个体删除. 实际上, 原排挤机制是个体的一维信息单次比较, 这种排挤机制有以下的缺点:

- 1) 只评价个体的一维信息有可能会淘汰一些分布度较好且某一维函数值远大于其他维函数值(即包含极端信息)的个体.
- 2) 连续删除拥挤距离较小个体, 没有及时更新受影响个体拥挤距离是单次评价函数信息的一个主要弊端.

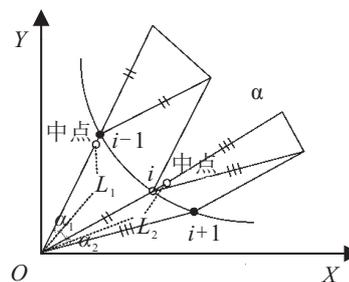
## 2 提出算法

### 2.1 基于二维信息的循环排挤机制

以二维空间为例, 拥挤距离计算公式所得结果是



(a) 个体扇形面积信息



(b) 伪半径与夹角

图 2 二维信息

一维信息,即在坐标轴平面的一条线,它不能有效衡量个体在周围环境的密集程度,那么采用个体与坐标轴圆点围成的扇形面积理论上能有效解决分布问题,但扇形面积并不能适应所有的前沿形状.如图2(a)所示,仅当前沿为上凸时误差较小,所以本文采用伪半径与个体间夹角相乘的形式,这样既可以获得个体所处环境的二维信息,又可以适当减小误差.

每个个体二维信息的构造均基于伪半径  $L$  与两向量间夹角  $\alpha$  相乘的结果,其中  $L$  与  $\alpha$  是两种完全不相关的信息,如图2(b)所示.

假设目标函数个数为  $m$  个,需要进行裁剪操作的某一级个体有  $s$  个,个体的位置(个体从决策空间至目标空间的表现)为  $P_i = (p_{i1}, p_{i2}, \dots, p_{im})$ ,  $p_{i1}$  为个体  $i$  代入目标函数1的值.个体  $i$  所处周围环境(同一非支配等级的相邻两个个体)的两个伪半径为

$$L_{i,1} = (|p_{i-1}| + |p_i|)/2, \quad i = 2, 3, \dots, s - 1; \quad (2)$$

$$L_{i,2} = (|p_i| + |p_{i+1}|)/2, \quad i = 2, 3, \dots, s - 1; \quad (3)$$

$$\alpha_{i,1} = \cos^{-1} \left( \frac{\sum_{k=1}^m p_{i-1,k} p_{i,k}}{\sqrt{\sum_{k=1}^m (p_{i-1,k})^2} \sqrt{\sum_{k=1}^m (p_{i,k})^2}} \right); \quad (4)$$

$$\alpha_{i,2} = \cos^{-1} \left( \frac{\sum_{k=1}^m p_{i,k} p_{i+1,k}}{\sqrt{\sum_{k=1}^m (p_{i,k})^2} \sqrt{\sum_{k=1}^m (p_{i+1,k})^2}} \right). \quad (5)$$

二维信息拥挤度公式为

$$I_i = \alpha_{i,1} L_{i,1} + \alpha_{i,2} L_{i,2}. \quad (6)$$

当目标函数增加,即优化三维多目标优化问题时,依据上述公式只可求得二维信息拥挤度,然而三维多目标优化问题使用三维信息拥挤度比二维更好.对于三维环境下某个体  $i$ ,由式(6)求得  $I_i$ ,由式(1)求得  $I(i)_{\text{dist}}$ ,将  $I_i \times I(i)_{\text{dist}}$  作为个体的三维信息拥挤度.

由以上分析可知:空间距离信息表现形式为个体间距离,它实质上是一维信息,即只包含个体间欧氏距离一种信息;本文提出的二维信息表现形式为两种性质截然不同的信息,即零点与个体间距离、与个体间角度.衡量种群中个体的二维信息比只计算单一空间距离信息全面,对整个种群分布性的估计也更加合理.此外,将二维信息与循环排挤策略结合,可以有效提高分布性计算的准确性.值得一提的是,三维多目标问题的分布性处理采用三维信息,即在二维信息的基础上增加了之前的空间距离信息.

## 2.2 数量级阈值

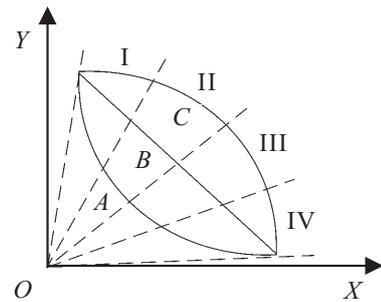
前文提到,一些含有极端信息的个体会因为其拥挤度过小而被删除.然而,真实的情况是它们的分布情况会比一些个体更好,为了避免这种情况的发生,本文提出了数量级阈值.当个体的某一维目标函数值与其他维函数值的比值大于或小于某一数量级时,此个体被定义为包含极端信息的个体,这时的数量级临界值称为数量级阈值.

如图3(a)所示,在二维目标优化问题中,一般情况下理想前沿上个体的分布是均匀的,比较3个PARETO前沿  $A$ 、 $B$ 、 $C$  可以得出以下结论:

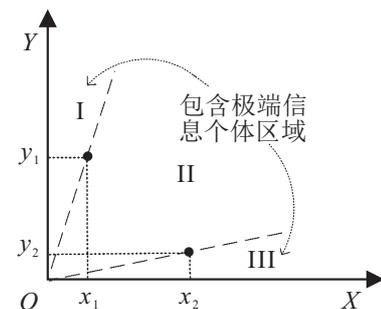
1) 不考虑I~IV区域内算法对不同目标函数搜索能力不同且搜索前沿为理想前沿,其特点是连续而均匀,前沿  $A$  在区域I、IV内的个体数目明显要多于区域II、III,前沿  $B$  在4个区域内相对于  $A$  较均匀,前沿  $C$  在4个区域内最为均匀.这意味着前沿  $A$  在区域I、IV内将要淘汰更多的个体,但这两段前沿  $A$  包含的个体数本应比区域II、III多.

2) I~IV区域内算法对不同目标函数搜索能力不同,由于不存在一种算法可以完美优化各种优化问题,不考虑非支配排序机制与二维信息拥挤度机制搜索能力的固有局限性.在进化后期种群中均为非支配解时,若要增强区域I、IV的搜索能力,则可以适当提高该区域个体二维信息拥挤度值的大小.

3) 在排挤机制的实施过程中,曲线  $A$  最有可能需要加入数量级阈值的干预.



(a) 不同区域前沿



(b) 数量级阈值计算

图3 前沿与数量级阈值

如图3(b)所示,区域I、III是可能包含极端信息个体的区域,要适度多保留这两个区域的个体,可以先确定数量级阈值  $\text{Thr} = [\text{Thr}_1 \ \text{Thr}_2]$  的大小,然后将两个区域中的个体赋予平均夹角,伪半径不变.设本层所含个体数为  $k$ ,区域I、III包含个体数分别为  $k_1$ 、 $k_2$ ,  $x_1$ 、 $y_1$  与  $x_2$ 、 $y_2$  分别为极端边界个体的横纵坐标(即不同目标函数值),基于数量级阈值的夹角公式为

$$\begin{cases} \alpha_i = \frac{\pi}{2k}, i = 2, 3, \dots, k_1 - 1, \frac{y_1}{x_1} > \text{Thr}_2; \\ \alpha_j = \frac{\pi}{2k}, j = k - k_1 + 2, \dots, k_1 - 1, \frac{y_1}{x_1} < \text{Thr}_1. \end{cases} \quad (7)$$

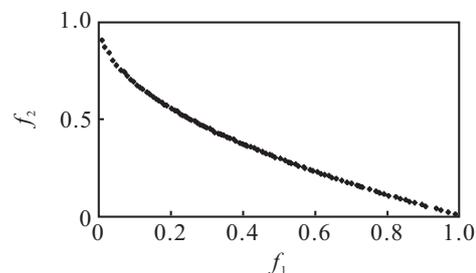
注意到,数量级阈值只对二维多目标问题有效,一般设置为  $[\text{Thr} = [0.01 \ 100]]$ ,若不需要数量级阈值的干预,则可以设置为  $\text{Thr} = [0 \ \text{inf}]$ .这相当于只采用二维信息循环排挤机制,减弱了对极端信息个体的搜索能力,但是某些目标函数并不会使算法产生极端信息个体,故不采用数量级阈值时对某些目标函数不会造成影响.除这些不受影响的目标函数外,对于其他函数可能会导致前沿上包含极端信息个体区域个体较少.此外,数量级阈值因二维信息循环排序策略产生,作用是寻找可能存在的极端信息个体,达到协助二维信息循环排序策略完善前沿的目的.

本文数量级阈值的设置是在进行仿真实验时,动态实时观测出算法对于不同测试函数在包含有极端信息个体的范围内的搜索能力,根据种群中个体的分布,适当地进行数量级阈值的设置,以干预进化过程,提高某些包含极端信息个体区域的个体搜索保留能力.

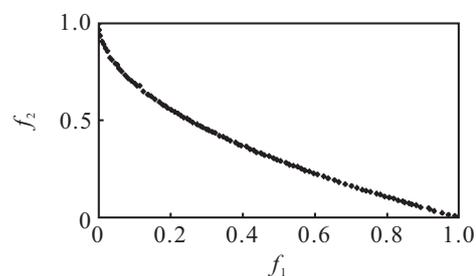
当针对一个全新问题时,数量级阈值的设置可以通过非动态实时观测获得,即进行第1次重复性实验时不采取数量级阈值的干预,得到种群中个体分布的最终结果,根据最终结果的位置决定增强某包含极端信息个体区域的搜索能力.一般情况下,此后的重复性实验采用第1次实验结果所决定的阈值,若算法对某优化问题每次结果差异过大,则取消数量级阈值的干预或者将数量级设置为  $\text{Thr} = [0.00001 \ 100000]$ .这种设置办法合理的原因是:算法在搜索个体时,整个过程在搜索空间内为动态随机搜索,远离 Pareto 前沿时,这些个体在目标空间是邻近的,但在决策空间往往是不邻近的,当这些个体接近 Pareto 前沿时,在决策空间便很有可能是邻近的,因为 Pareto 前沿上的任意一点都可以作为等效单目标优化问题的解.所以,当有数量级阈值的干预时,保留目标空间某区域个体,则在决策空间上与这些个体在目标空间内的相对位置几乎是一一对应的,经过交叉变异再产生的个

体有很大的概率是保留某目标空间区域个体邻近的个体,不会产生劣解.

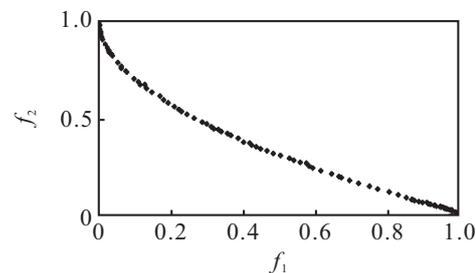
以 ZDT1 函数为例测试数量级阈值,结果如图4所示.由图4(a)可见,当不采用数量级阈值干预,即数量级阈值  $\text{Thr} = [0 \ \text{inf}]$  时,两边含有极端信息个体区域明显稀疏,这是因为二维信息的排挤机制中含有



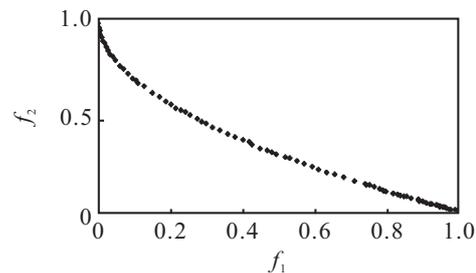
(a)  $[\text{Thr a}] = [0 \ \text{inf} \ 0]$



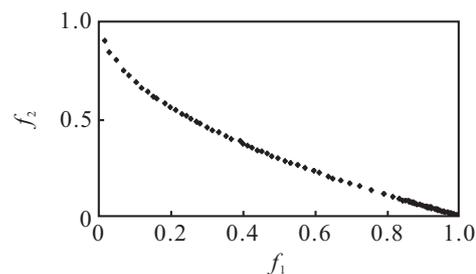
(b)  $[\text{Thr a}] = [0.2 \ 6 \ p_i/200]$



(c)  $[\text{Thr a}] = [0.1 \ 3 \ p_i/100]$



(d)  $[\text{Thr a}] = [0.2 \ 6 \ p_i/100]$



(e)  $[\text{Thr a}] = [0.1 \ \text{inf} \ p_i/100]$

图4 不同数量级阈值结果对比

个体向量的角度,当个体越接近 $f_2$ 轴或 $f_1$ 轴时,其角度过小,使得不能保留期望个数的个体.图4(b)在平均角度不变的情况下,将数量级阈值的每个元素扩大为2倍,相当于选择性增加了区域的个体搜索保留能力,但个体分布依旧较均匀,没有出现理论上区域内个体密集、区域外个体稀疏,产生这种情况的原因是扩大区域中个体夹角与设置的平均夹角几乎相同,不会对扩大区域的个体造成影响.图4(c)提高仿真中的平均角度,将 $\pi/200$ 增大到 $\pi/100$ ,其他值不变,个体在数量级阈值范围内的个体很紧密、范围之外很稀疏,表明平均角度可以提高某区域的搜索能力和个体保留能力.图4(d)保持增大的平均角度 $\pi/100$ 不变,数量级阈值范围增加一倍,在该范围内个体密集、范围外个体稀疏.

### 2.3 算法流程

本文所提出算法基于NSGA-II-DE的框架,采用含有数量级阈值的二维信息循环排挤机制(2D-Thr)代替原拥挤距离排挤机制,算法流程如下.

Step 1: 判断优化问题维数,若为二维,则按第1个目标函数值从小到大排序层级内个体,若为三维,则对每个目标函数值都要从小到大排序层级内个体,获得个体的位置信息.

Step 2: 函数值最大与函数值最小个体的拥挤度为无穷大,从第2个个体开始利用个体的位置信息根据式(2)~(6)计算其伪半径与两向量间的夹角,得出个体的二维信息拥挤度.若是三维多目标问题,则继续计算出个体的三维信息拥挤度.

Step 3: 采用设置的Thr,并根据式(7)计算结果于预算法进程.

Step 4: 比较个体的信息拥挤度,将拥挤度最小的个体删除,判断所删除个体相邻两个个体是否为边界个体.若不是,则重新计算该删除个体前后两个个体的信息拥挤度;若是上边界个体,则重新计算该删除个体下一个体的信息拥挤度;若是下边界个体,则重新计算该删除个体上一个体的信息拥挤度,然后将信息拥挤度最小的个体删除,直至种群中个体数目满足要求为止.

## 3 仿真分析

### 3.1 测试函数和程序参数设置

选取8个测试函数,其中3个测试函数为ZDT系列函数ZDT1、ZDT2、ZDT3<sup>[6]</sup>,其真实前沿分别为凹型、凸型、非连续型,另外3个为CEC2009进化大赛的测试函数UF2、UF4、UF7,其真实前沿分别为凹型、凸型、直线型.Zhang等<sup>[11]</sup>认为,UF函数具有复杂Pareto解集的形状,该特点对于研究算法的性能很有

帮助.以上测试函数的搜索空间维数均为30.此外,2个不同类型三维多目标问题为具有连续前沿面的DTLZ4和非连续前沿面的DTLZ7<sup>[15]</sup>,搜索空间维数分别为12和22.

程序的参数设置为:ZDT系列测试函数和DTLZ系列测试函数的最大迭代次数为500,UF系列测试函数的最大迭代次数为3000,二维目标问题时个体数统一为100( $k=100$ ).为方便比较,三维目标问题的个体数设置为150,采用多项式变异,变异概率为 $1/n$ (测试函数要求 $n$ 为30),变异参数为20,差分缩放系数为0.5,差分交叉系数为1.因为交叉算子采用差分算子代替,所以采用的联赛选择元度为3,联赛规模为50,表1为数量级阈值和平均角度的设置.注意到,ZDT3的Pareto前沿存在于第1和第4象限,运算时,ZDT3的平均角度为 $\pi/k$ ( $k=100$ ),数量级阈值的选择依据算法在每个测试函数不同位置内搜索能力的不同获得.

表1 数量级阈值设置

测试函数	2D-Thr		
	Thr <sub>1</sub>	Thr <sub>2</sub>	$\alpha$
ZDT1	0.1	3	$\pi/2k$
ZDT2	0.1	6	$\pi/2k$
ZDT3	0	7	$\pi/k$
UF2	0.3	8	$\pi/2k$
UF4	0.00001	100000	$\pi/2k$
UF7	0.00001	100000	$\pi/2k$
DTLZ4	0	inf	—
DTLZ7	0	inf	—

### 3.2 仿真结果对比

图5为4个算法2D-Thr、NSGA-II-DE、NSGA-II、MACPSO<sup>[16]</sup>在8个测试函数下的表现.由图5可见:在ZDT系列函数上,2D-Thr所得个体分布均匀程度与MACPSO相差不多且稍优于NSGA-II和NSGA-II-DE,4个算法所得结果均无明显断档;在UF系列函数上,2D-Thr总体表现相对较好,NSGA-II、NSGA-II-DE、MACPSO均存在较大断档,并且在含有极端信息个体区域保留个体较少;在处理连续和非连续三维多目标问题方面,MACPSO在处理DTLZ7时收敛度表现较好,除此之外均是2D-Thr表现较好.

### 3.3 评价指标

算法性能的好坏主要由收敛性和多样性两方面决定,选取文献[5]提出的iGD(反向世代距离)衡量算法的收敛性.假设 $Q^*$ 在目标空间内沿着Pareto前沿均匀分布,Pareto近似前沿 $Q$ 与 $Q^*$ 的距离定义为iGD,有

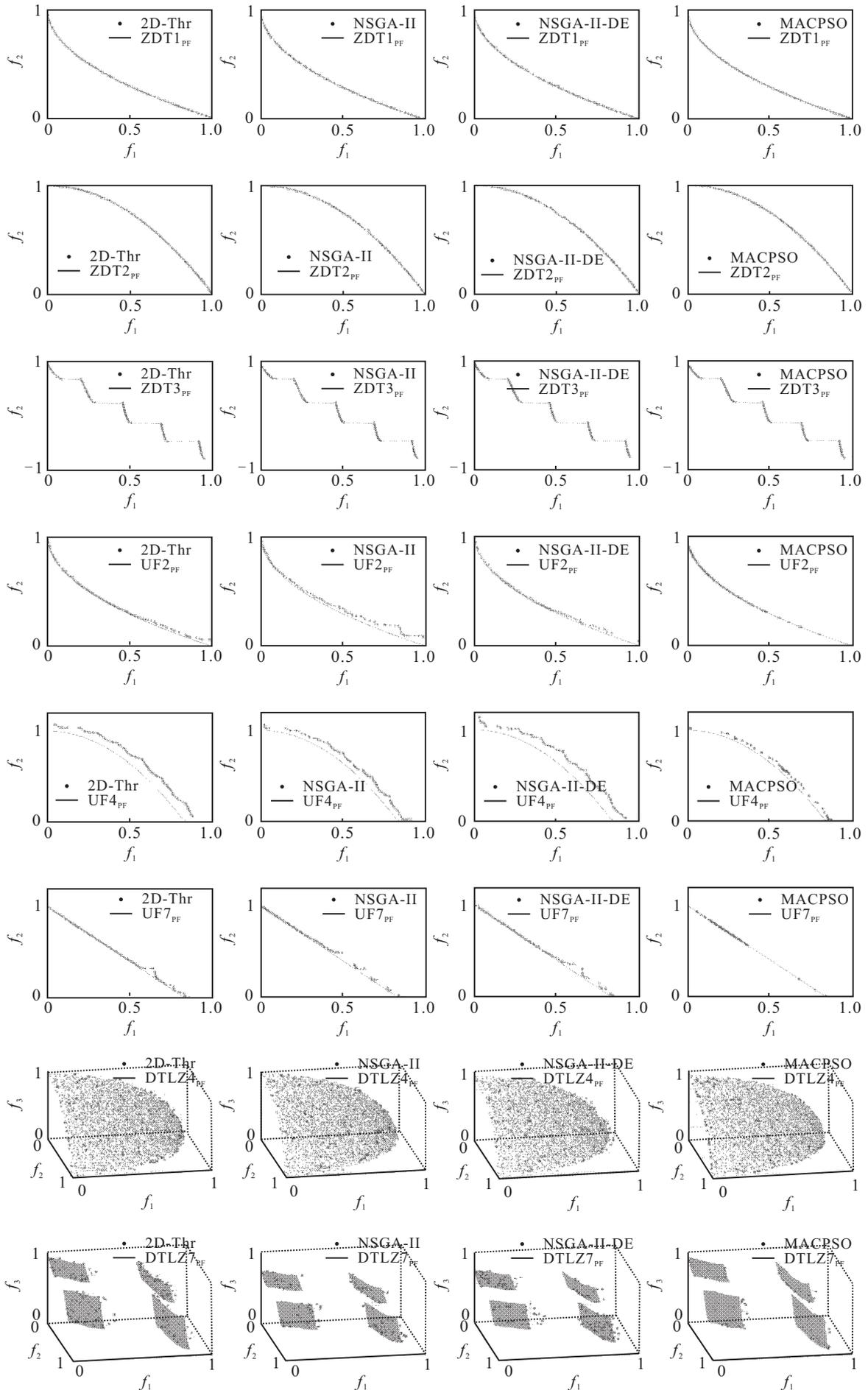


图5 仿真结果对比

$$iGD(Q, Q^*) = \frac{\sum_{v \in Q^*} d(v, Q)}{|Q^*|}, \quad (8)$$

其中  $d(v, Q)$  为点  $v \sim Q$  的最小欧氏距离.  $iGD$  值越小代表算法有越好的收敛性能. 实验中, 在二维真实前沿均匀选取 500 个点作为  $Q^*$ , DTLZ4 和 DTLZ7 三维真实前沿分别取 4 106 和 9 409 个点作为  $Q^*$ . 此外, 采用文献 [6] 提出的  $\Delta$  (多样性) 衡量二维空间内个体的分布情况, 有

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}}, \quad (9)$$

其中  $d_f$  和  $d_l$  是所获得非支配解集中极端解与边界解之间的欧几里得距离. 假设非支配前沿上有  $N$  个最优解,  $\bar{d}$  为所有  $d_i$  的平均值,  $i = 1, 2, \dots, N-1$ , 若有  $N$  个解, 则必存在  $N-1$  个连续的欧几里得距离. 由于  $\Delta$  不能作为三维多目标优化问题评价指标, 衡量三维空间内种群分布性采用文献 [17] 提出的 Spacing 评价方法.

### 3.4 结果分析

将所提出算法与其余 3 种算法作量化比较, 以展示出算法间的区别, 各项指标越小越好, 结果如

表 2~表 4 所示. 由表 2 和表 3 可见, 2D-Thr 算法在个体分布性方面优于其他 3 种算法, 其中 ZDT3 略微提升, 有以下两方面原因: 1) 拥挤距离排挤机制对于 ZDT3 在目标空间内搜索能力较强, 几乎不需要采用数量级阈值策略提高搜索局部含有极端信息区域的能力; 2) ZDT3 的真实前沿存在于坐标轴的第 1 和第 4 象限, 若以 0 点为圆点,  $Y$  轴正方向的角度为 0,  $Y$  轴负方向的角度为  $\pi$ , 则 ZDT3 的真实前沿角度范围为  $[0, \pi]$ , 且 ZDT3 函数不为连续函数, 中间间断部分个体夹角会导致 2D-Thr 的平均角度增加, 使得数量级阈值的平均角度不准确. 若将数量级阈值设置成某范围, 则会导致某范围内个体密集、范围外个体稀疏. 在 ZDT1/2 两个简单测试函数方面, MACPSO 优于 2D-Thr, 然而对于复杂测试函数 UF2/4/7, 2D-Thr 全面优于 MACPSO, 主要原因是 MACPSO 侧重于函数收敛性, 对于易早熟收敛的复杂函数而言, 2D-Thr 性能更均衡. 表 4 中 2D-Thr 有一定优势, 在 UF2 和 UF7 函数上, 2D-Thr 的收敛度有明显提升, 在 ZDT3 和 UF4 函数上无优势, 但收敛度均属于一个数量级且很小. ZDT3 和 UF4 函数上的收敛度均值从小到大依次为 NSGA-II、2D-Thr、NSGA-II-DE, 即 NSGA-II-DE

表 2 二维空间种群多样性比较

测试函数	2D-Thr			NSGA2			NSGA2-DE			MACPSO		
	$\Delta_{max}$	$\Delta_{mean}$	$\Delta_{min}$									
ZDT1	3.77e-01	3.17e-01	2.67e-01	6.88e-01	6.01e-01	5.37e-01	6.81e-01	5.95e-01	4.55e-01	1.58e-01	<b>9.56e-02</b>	7.75e-01
ZDT2	2.31e-01	2.08e-01	1.71e-01	7.44e-01	6.29e-01	5.72e-01	6.90e-01	6.16e-01	5.44e-01	8.46e-02	<b>8.20e-02</b>	7.41e-02
ZDT3	6.38e-01	<b>6.14e-01</b>	5.39e-01	7.01e-01	6.16e-01	5.41e-01	7.09e-01	6.27e-01	5.55e-01	6.61e-01	6.56e-01	6.53e-01
UF2	6.83e-01	<b>5.00e-01</b>	3.83e-01	7.45e-01	6.35e-01	5.31e-01	7.26e-01	6.32e-01	5.33e-01	6.57e-01	6.05e-01	5.69e-01
UF4	5.79e-01	<b>3.66e-01</b>	2.63e-01	6.99e-01	6.19e-01	5.39e-01	7.41e-01	6.35e-01	5.24e-01	6.83e-01	5.95e-01	5.34e-01
UF7	1.11e+00	<b>5.49e-01</b>	2.53e-01	7.67e-01	6.69e-01	5.71e-01	6.88e-01	6.34e-01	5.28e-01	1.02e+00	8.28e-01	7.26e-01

表 3 三维空间种群分布度比较

测试函数	2D-Thr			NSGA2			NSGA2-DE			MACPSO		
	SPmax	SPmean	SPmin	SPmax	SPmean	SPmin	SPmax	SPmean	SPmin	SPmax	SPmean	SPmin
DTLZ4	5.41e-02	<b>4.02e-02</b>	4.07e-02	5.65e-02	5.09e-02	4.42e-02	5.59e-02	4.79e-02	4.39e-02	5.54e-02	4.71e-02	4.05e-02
DTLZ7	5.70e-02	<b>4.68e-02</b>	2.56e-02	6.86e-02	5.19e-02	4.90e-02	6.59e-02	5.23e-02	1.53e-02	7.34e-02	6.06e-02	4.90e-02

表 4 算法收敛度比较

测试函数	2D-Thr			NSGA2			NSGA2-DE			MACPSO		
	iGDmax	iGDmean	iGDmin	iGDmax	iGDmean	iGDmin	iGDmax	iGDmean	iGDmin	iGDmax	iGDmean	iGDmin
ZDT1	7.19e-03	<b>5.68e-03</b>	5.03e-03	6.41e-03	5.78e-03	5.25e-03	6.85e-03	6.22e-03	5.39e-03	6.04e-03	5.68e-03	4.92e-03
ZDT2	8.50e-03	5.41e-03	4.89e-03	7.53e-03	5.91e-03	5.29e-03	7.66e-03	6.31e-03	5.64e-03	4.64e-03	<b>4.53e-03</b>	4.46e-03
ZDT3	7.73e-03	7.37e-03	6.67e-03	6.93e-03	6.38e-03	5.75e-03	9.89e-03	7.78e-03	6.43e-03	6.42e-03	<b>6.08e-03</b>	5.92e-03
UF2	2.75e-02	<b>2.15e-02</b>	1.78e-02	5.17e-02	3.69e-02	3.05e-02	3.29e-02	2.44e-02	1.78e-02	5.54e-02	3.28e-02	1.99e-02
UF4	3.42e-02	6.20e-02	5.77e-02	5.32e-02	5.09e-02	4.95e-02	9.28e-02	7.71e-02	6.10e-02	4.93e-02	<b>4.21e-02</b>	3.74e-02
UF7	1.11e+00	<b>1.56e-02</b>	1.01e-02	5.21e-02	4.32e-02	2.60e-02	2.69e-02	2.22e-02	1.85e-02	5.00e-01	1.70e-01	6.23e-02
DTLZ4	4.75e-01	<b>3.41e-01</b>	3.13e-01	5.51e-01	3.49e-01	3.18e-01	5.33e-01	3.49e-01	3.15e-01	4.76e-01	3.47e-01	3.13e-01
DTLZ7	4.09e-01	1.42e-01	6.46e-02	1.05e+00	2.74e-01	7.76e-02	5.04e-01	2.08e-01	8.10e-02	3.52e-01	<b>1.23e-01</b>	6.56e-02

的差分算子对于这两个测试函数的收敛性没有提升的作用,从而导致以NSGA-II-DE为基础框架的2D-Thr也对这两个函数的收敛性起不到提升作用。MACPSO对于简单函数的收敛性略好于2D-Thr,然而在UF4函数上,MACPSO因为降低了种群分布性,从而提高了整个种群的收敛性。

#### 4 结 论

本文对NSGA-II的拥挤度计算公式和排挤机制进行了深入研究,分析了不足之处,提出了以NSGA-II-DE为基础框架的改进算法2D-Thr。仿真结果表明,相比于传统的NSGA-II、NSGA-II-DE和新算法MACPSO,2D-Thr具有明显的优势,但是在某些函数方面并无提高,这也是值得继续深入研究的问题。

#### 参考文献(References)

- [1] Giagkiozis I, Fleming P J. Methods for multi-objective optimization: An analysis[J]. *Information Sciences*, 2015, 293(1): 338-350.
- [2] Knowles J, Corne D. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation[C]. *Proc of the 1999 Congress on Evolutionary Computation CEC*. Washington DC: IEEE Press, 1999, 1: 98-105.
- [3] Corne D W, Knowles J D, Oates M J. The Pareto envelope-based selection algorithm for multiobjective optimization[C]. *Parallel Problem Solving from Nature PPSN VI*. Berlin: Springer-Heidelberg, 2000: 839-848.
- [4] Laumanns M. SPEA2: Improving the strength Pareto evolutionary algorithm[C]. *Evolutionary Methods for Design Optimisation and Control*. CIMNE, 2002, 3242: 95-100.
- [5] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. *IEEE Trans on Evolutionary Computation*, 2007, 11(6): 712-731.
- [6] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(2): 182-197.
- [7] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms[J]. *IEEE Trans on Evolutionary Computation*, 1994, 2(3): 221-248.
- [8] Liu H L, Gu F. A improved NSGA-II algorithm based on sub-regional search[C]. *IEEE Congress on Evolutionary Computation*. New Orleans: IEEE, 2011: 1906-1911.
- [9] Guo D, Wang J, Huang J, et al. Chaotic-NSGA-II: An effective algorithm to solve multi-objective optimization problems[C]. *Int Conf on Intelligent Computing and Integrated Systems(ICISS)*. Guilin: IEEE, 2010: 20-23.
- [10] 罗辞勇, 陈民铀, 张聪誉. 采用循环拥挤排序策略的改进NSGA-II算法[J]. *控制与决策*, 2010, 25(2): 227-231. (Luo C Y, Chen M Y, Zhang C Y. Improved NSGA-II algorithm with circular crowded sorting[J]. *Control and Decision*, 2010, 25(2): 227-231.)
- [11] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. *IEEE Trans on Evolutionary Computation*, 2009, 13(2): 284-302.
- [12] Sindhya K, Sinha A, Deb K, et al. Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems[C]. *IEEE Congress on Evolutionary Computation*. Trondheim: IEEE, 2009: 2919-2926.
- [13] D'Souza R G L, Sekaran K C, Kandasamy A. Improved NSGA-II based on a novel ranking scheme[J]. *J of Computing*, 2010, 2(2): 91-95.
- [14] Storn R, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces[J]. *J of Global Optimization*, 1997, 11(4): 341-359.
- [15] Mezura-Montes E, Velázquez-Reyes J, Coello Coello C A. A comparative study of differential evolution variants for global optimization[C]. *Proc of the 8th Annual Conf on Genetic And Evolutionary Computation*. New York: ACM, 2006: 485-492.
- [16] 杨景明, 马明明, 车海军, 等. 多目标自适应混沌粒子群优化算法[J]. *控制与决策*, 2015, 30(12): 2168-2174. (Yang J M, Ma M M, Che H J, et al. Multi-objective adaptive chaotic particle swarm optimization algorithm[J]. *Control and Decision*, 2015, 30(12): 2168-2174.)
- [17] Schott J R. *Fault tolerant design using single and multicriteria genetic algorithm optimization*[D]. Cambridge: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

(责任编辑: 郑晓蕾)