

一种 MOEA 分布度的逐步评价方法

李密青, 郑金华, 谢炯亮, 杨 平, 李 晶

(湘潭大学信息工程学院, 湖南湘潭 411105)

摘 要: 提出了一种多目标进化算法中解集分布度逐步评价方法. 定义了一种基于角度的坐标, 避免了算法因收敛性不同对分布性评价的影响; 利用了解集均匀分布具有的对称性, 把整个目标空间从大到小划分成不同的对称区域, 逐步进行分布度评价. 实验结果表明, 该方法能精确的评价解集的分布情况.

关键词: 多目标进化算法; 分布度评价; 逐步评价

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2008) 10-1986-06

A Stepwise Diversity Metric for MOEA

LI Mi-qing, ZHENG Jia-hua, XIE Jiong-liang, YANG Ping, LI Jing

(Institute of Information Engineering, Xiangtan University, Xiangtan, Hunan 411105, China)

Abstract: A measurement for stepwise diversity evaluation of solutions in multi objective evolutionary algorithm was proposed. In this method, a coordinate based on angle was defined, which can avoid the influence caused by different extent of convergence in diversity evaluation. Furthermore, considering the symmetry of uniformly distributed solutions, the whole objective space was divided and subdivided into different symmetric regions for evaluation. Experiments show that the method can evaluate diversity of solutions accurately.

Key words: multi objective evolutionary algorithm; diversity metric; stepwise evaluation

1 引言

进化算法(Evolutionary Algorithms, EA)已经被证明是一种解决多目标优化问题(Multiobjective Optimization problem, MOP)的有效方法^[1]. 各国学者提出了很多有效的多目标进化算法(Multi-Objective Evolutionary Algorithm, MOEA)^[2]. 该类算法的特点是不需要人为定义各目标的权重, 而是由算法运行一次找出所有的非支配解^[1], 再由决策者进行权衡选择. 但是, 多目标进化算法并不能保证找到 Pareto 最优解, 而是尽可能的找到一组很好的近似解. 因此, 如何判断一组近似解的优劣就变得非常重要.

在 MOEA 发展前期, 研究者主要是通过对最终边界的观察来评价 MOEA 的好坏, 但这种直观的方法不能给出精确的评价结果, 并且不适合于目标维数为三维以上的测试问题. MOEA 进入第二个发展阶段后, 很多种 MOEA 性能评价方法相继被提出, 归纳起来可以分为两大类^[2]: (1) 一类是用来评价所求解集与真正 Pareto 最优面的趋近程度的评价方法; (2) 一类是用来评价解的

分布性的评价方法. 可见解群体的分布多样性是评价算法性能的重要指标. 目前比较流行的分布度评价方法有: Farhang-Mehr 等提出的基于个体信息熵的评价方法^[3], Deb 等提出的 Spacing metric^[4], 网格评价方法^[5], 基于个体空间的评价方法^[6], Sanaz Mostaghim 等提出的 Sigma 评价方法^[7], 以及 Li 等^[8]用邻域进行分布度评价的方法等等. 但是由于存在参数难以选择, 映射失真等问题, 还没有一种方法能完美的评价分布度情况.

本文提出了在一种新的坐标下进行分布度的评价方法. 利用解集均匀分布具有的对称性, 把整个目标空间从大到小划分成不同的对称区域, 逐步进行分布度评价. 由于整个评价过程是在一种基于角度的坐标下进行的, 这样避免了算法因收敛性不同对分布性评价所造成的影响. 实验表明我们的方法能精确的评价解集的分布情况.

2 相关概念

最小化与最大化问题可以相互转化, 因此, 仅以最小化多目标问题为研究对象. 多目标问题(MOP)的一般

描述为:

给定决策向量 $X = (x_1, x_1, \dots, x_n)$, 它满足下列约束:

$$g_i(X) \geq 0, \quad i = 1, 2, \dots, k \quad (1)$$

$$h_i(X) = 0, \quad i = 1, 2, \dots, l \quad (2)$$

设有 r 个优化目标, 且这 r 个优化目标是相互冲突的, 优化目标可表示为:

$$\vec{f}(X) = (f_1(X), f_2(X), \dots, f_r(X))$$

寻求 $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, 使 $\vec{f}(X^*)$ 在满足约束(1)和(2)的同时达到最小. MOEA 经常用到如下几个基本概念:

定义 1 个体的 Pareto 支配关系(优越关系)

设 p 和 q 是进化群体 Pop 中的任意两个不同的个体, 我们称 p 支配(dominate) q , 则必须满足下列二个条件:

(1) 对所有的子目标, p 不比 q 差, 即 $f_k(p) \leq f_k(q)$ ($k = 1, 2, \dots, r$).

(2) 至少存在一个子目标, 使 p 比 q 好. 即 $\exists l \in \{1, 2, \dots, r\}$ 使 $f_l(p) < f_l(q)$. 其中 r 为子目标的数量, p 支配 q 记为 $p > q$.

定义 2 Pareto 最优解

给定一个多目标优化问题 $\text{Min} \vec{f}(X)$, 称 $X^* \in \Omega$ 是最优解, 若 $\forall X \in \Omega$, 满足下列条件:

或者 $\bigwedge_{i \in I} (f_i(X) = f_i(X^*))$

或者 至少存在一个 $j \in I, I = \{1, 2, \dots, r\}$, 使: $f_j(X) > f_j(X^*)$

其中 Ω 是满足式(1)和式(2)的可行解集, 即:

$$\Omega = \{X \in R^n \mid g_i(X) \geq 0, h_i(X) = 0; \\ (i = 1, 2, \dots, k; j = 1, 2, \dots, l)\}$$

定义 3 Pareto 最优面(前沿)

给定一个多目标优化问题 $\text{Min} \vec{f}(X)$ 和它的最优解集 $\{X^*\}$, 它的 Pareto 最优面定义为:

$$PF^* = \{\vec{f}(X) = (f_1(X), f_2(X), \dots, f_r(X)) \mid X \in \{X^*\}\} \quad (3)$$

3 现有的分布度评价方法分析

Deb^[5]指出在设计 MOEA 性能评价方法时, 应该考虑以下 5 个特征:

(a) 函数值的范围应当在 0 到 1 之间, 因为函数要用于不同代之间的比较, 0 到 1 之间的函数可以更方便比较算法不同代之间的变化.

(b) 期望函数值应当是可知的, 也就是说理论上的非支配集的分布度是可以计算出来的.

(c) 评价曲线应当是随着代数的增加成递增或递减的, 这样更有利于不同集合之间的比较.

(d) 评价函数应适用于任意多个目标, 尽管这不是

绝对需要的, 但这使得算法能在不同维上进行比较.

(e) 函数的计算复杂度不能太高.

Deb 于 2000 年提出了 Spacing metric^[4], 用来评价解集的分布情况, 评价函数如下:

$$\Delta = \sum_{i=1}^{|PF|} \frac{d_i - \bar{d}}{|PF|} \quad (4)$$

PF 代表已知的 Pareto 最优面, d_i 是指解集中非支配边界上两个连续向量的欧几里德距离, \bar{d} 是这些距离的平均值. 值得注意的是这种评价方法比较适用于 2 维的目标空间, 而在高维目标情况效果不理想. 类似地, Schott 提出了一种计算分布度的方法^[9], 其函数定义如下:

$$\Delta' = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (5)$$

式(5)中, $d_i = \min_j (|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$, ($i, j = 1, 2, \dots, n$), \bar{d} 是所有 d_i 的平均值, n 是已知的 Pareto 边界的大小. 由于此方法只考虑最近个体之间的分布关系, 所以它并不能准确的反映整个解集的分布情况.

Farhang Mehr 等于 2003 年提出了一种依靠个体信息的分布度评价方法^[3], 它是采用个体的信息熵来评价解集的分布情况. 但也存在些不足: (1) 信息熵值在很大程度上取决于标准方差 σ 的选取; (2) 对于 Pareto 最优面不连续的情况, 信息熵的方法并不适用. Deb 等在 2002 年提出了一种网格分布度评价方法^[5]. 该方法根据网格中的个体数来评价解集的分布性. 但由于一些参数的难以选择, 如映射的超平面, 网格的大小等, 所以此方法计算出来的分布度值与实际情况并非完全相符. Sanaz Mostaghim 在 2005 年提出了 Sigma 评价方法^[7], 给出了解集在目标空间的覆盖率信息, 但参考半径 d 的大小较难选择, 从而影响了了解的精确性.

4 分布度逐步评价方法

4.1 定义新坐标

我们的评价方法首先定义一种基于个体角度的坐标, 这种坐标只考虑个体的角度分量, 并不考虑个体之间的距离, 这样避免了算法因收敛性不同而对分布性评价造成的影响. 坐标定义如下:

当目标为二维时, 坐标向量为

$$\alpha = (\arcsin \frac{f_2}{\sqrt{f_1^2 + f_2^2}}) \quad (6)$$

当目标为三维时, 坐标向量为:

$$\alpha = (\arcsin \frac{f_2}{\sqrt{f_1^2 + f_2^2}}, \arcsin \frac{f_3}{\sqrt{f_2^2 + f_3^2}}, \arcsin \frac{f_1}{\sqrt{f_3^2 + f_1^2}}) \quad (7)$$

m 维目标空间的个体坐标 α 是 C_m^2 维向量:

$$\alpha = (\arcsin \frac{f_2}{\sqrt{f_1^2 + f_2^2}}, \arcsin \frac{f_3}{\sqrt{f_2^2 + f_3^2}}, \dots, \arcsin \frac{f_m}{\sqrt{f_{m-1}^2 + f_m^2}}, \arcsin \frac{f_1}{\sqrt{f_m^2 + f_1^2}}, \arcsin \frac{f_3}{\sqrt{f_1^2 + f_3^2}}, \dots) \quad (8)$$

4.2 坐标变换

一般的测试问题的解个体并不能覆盖整个目标空间,为了能在目标空间中准确的评价分布情况,需要对其进行变换.我们的方法是先求出解群体中各个目标上最小的值,然后以这些最小的值组成的向量作为新原点,图1显示了坐标变换过程.

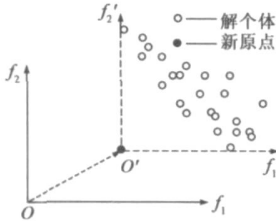


图1 二维空间坐标变换过程

原点 O' 是由解群体 f_1 和 f_2 上最小的值组成,根据点 O' 形成了新的坐标轴 f_1', f_2' , 这样经过坐标变换的解个体就覆盖了整个目标空间.

4.3 分布度逐步评价方法

分布度逐步评价方法利用了均匀分布的解集具有的对称性,先把整个目标空间 σ 划分成两个对称区域 σ_1, σ_2 , 统计这两个区域的个体数目,由数目的多少得出两个区域整体的分布关系,然后对区域 σ_1, σ_2 重复以上操作得到一些更小的区域 $\sigma_3, \sigma_4, \sigma_5, \sigma_6$, 如此进行下去,直到每个区域只有一个个体或者区域大小已经足够小.其具体算法如下:

算法1

参数设置 N : 种群规模, m : 目标空间维数, σ : 区域范围, σ_{size} : 区域大小, σ_{count} : 区域内个体数目, Q : 待评价区域队列

初始值设置 总分布情况 $all_rank = 0$, 总分布区域 $all_area = 0$, 初始维数分量 $j = 0$

Step 1: 对向量 α 第 j 维分量的区域 σ_j , 设置区域初始范围, 初始个体数, 初始区域大小: $\sigma = [0, \pi/2]$, $\sigma_{count} = N$, $\sigma_{size} = \pi/2$.

Step 2: 当 $\sigma_{size} \geq \eta$ 并且 $\sigma_{count} > 1$ 时转 Step3, 否则转 Step7, 其中 $\eta = \pi/2/2 \times \sqrt[m-1]{N}$.

Step 3: 划分 σ 为两个相等的区域, $\sigma' = [\sigma_{left}, \sigma_{left} + \sigma_{size}/2]$, $\sigma'' = [\sigma_{left} + \sigma_{size}/2, \sigma_{right}]$, $\sigma'_{size} = \sigma''_{size} = \sigma_{size}/2$, 其中 $\sigma_{left}, \sigma_{right}$ 分别为 σ 的左右边界.

Step 4: 统计区域 σ', σ'' 内个体的数目: $\sigma'_{count} = Stat(\sigma')$, $\sigma''_{count} = Stat(\sigma'')$.

Step 5: 评价 σ 分布情况 $all_rank = all_rank + (\min(\sigma'_{count}, \sigma''_{count}) / \max(\sigma'_{count}, \sigma''_{count})) \times \sigma_{size}$, $all_area = all_area + \sigma_{size}$.

Step 6: 插入 σ', σ'' 到队列 Q 中.

Step 7: 如果 $j = C_m^2 - 1$, 结束算法. 否则如果 Q 为空, $j = j + 1$, 转 Step1; 否则返回并删除 Q 的第一个元素 σ^* , $\sigma = \sigma^*$, $\sigma_{count} = Stat(\sigma)$, 转 Step2.

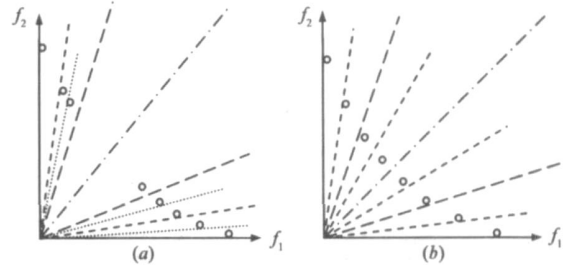


图2 二维空间中分布度评价实例

最终解集的分布度为总分布情况比总分布区域: $final_diversity = all_rank / all_area$, 下面给出几个用此算法来求分布度的简单例子.

图2显示了二维空间中解集分布情况的两个实例. 图2(a)中解集的分布不太均匀, 我们的

算法对目标空间进行了8次划分, 使每个区域至多有一个个体, 其分布度计算如下:

$$final_diversity = [(\frac{3}{5} \times \frac{\pi}{2} + \frac{1}{4} \times \frac{\pi}{4} + \frac{0}{3} \times \frac{\pi}{4} + \frac{2}{2} \times \frac{\pi}{8} + \frac{1}{2} \times \frac{\pi}{8} + \frac{1}{1} \times \frac{\pi}{16} + \frac{1}{1} \times \frac{\pi}{16} + \frac{1}{1} \times \frac{\pi}{16}) / (\frac{\pi}{2} + \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{8} + \frac{\pi}{8} + \frac{\pi}{16} + \frac{\pi}{16} + \frac{\pi}{16})] = 0.513043$$

图2(b)中个体的分布基本均匀, 我们的算法对目标空间进行了7次划分, 使每个区域至多有一个个体, 其分布度计算如下:

$$final_diversity = [(\frac{4}{4} \times \frac{\pi}{2} + \frac{2}{2} \times \frac{\pi}{4} + \frac{2}{2} \times \frac{\pi}{4} + \frac{1}{1} \times \frac{\pi}{8} + \frac{1}{1} \times \frac{\pi}{8} + \frac{1}{1} \times \frac{\pi}{8} + \frac{1}{1} \times \frac{\pi}{8}) / (\frac{\pi}{2} + \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{8} + \frac{\pi}{8} + \frac{\pi}{8} + \frac{\pi}{8})] = 1.000000$$

图3显示了三维空间中解集分布情况的一个实例, 按向量中每一维分量分别求解集的分布值, 并且区域大小下限为 $\eta = (\pi/2)/2 \times \sqrt[m-1]{N} = \pi/8.944$, 其总分布度如下:

$$final_diversity = [(\frac{3}{3} \times \frac{\pi}{2} + \frac{1}{2} \times \frac{\pi}{4} + \frac{1}{2} \times \frac{\pi}{4} + \frac{1}{1} \times \frac{\pi}{8} + \frac{0}{2} \times \frac{\pi}{8}) / (\frac{2}{2} \times \frac{\pi}{2} + \frac{1}{1} \times \frac{\pi}{4} + \frac{1}{1} \times \frac{\pi}{4})]$$

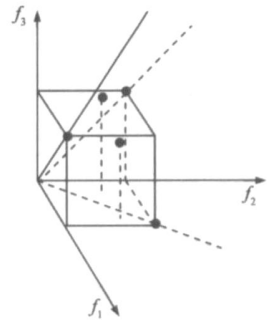


图3 三维空间中分布度评价实例

$$\begin{aligned} &\times \frac{\pi}{4} + \frac{1}{2} \times \frac{\pi}{8}) + (\frac{3}{3} \times \frac{\pi}{2} + \frac{1}{2} \times \frac{\pi}{4} \\ &+ \frac{1}{2} \times \frac{\pi}{4} + \frac{1}{1} \times \frac{\pi}{8} + \frac{0}{2} \times \frac{\pi}{8})] / [(\frac{\pi}{2} + \frac{\pi}{4} \\ &+ \frac{\pi}{4} + \frac{\pi}{8} + \frac{\pi}{8}) + (\frac{\pi}{2} + \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{8}) \\ &+ (\frac{\pi}{2} + \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{8} + \frac{\pi}{8})] = 0.660920 \end{aligned}$$

5 实验讨论与分析

为了测试我们的分布度评价方法的性能,本文选取了几个分布情况具有典型代表的 MOEA: NSGA-II^[4], G-NSGA-II^[6], SPEA2^[10] 分别在目标数 2, 3, 4 和 6 的情况下进行实验,在实验中,三种算法都采用实数编码,交叉概率 0.9,变异概率 0.001,二维测试函数的规模为 100,评价次数为 20000;三维测试函数的规模为 300,评价次数为 90000,四维测试函数的规模为 500,评价次数为 250000,四维测试函数的规模为 800,评价次数为

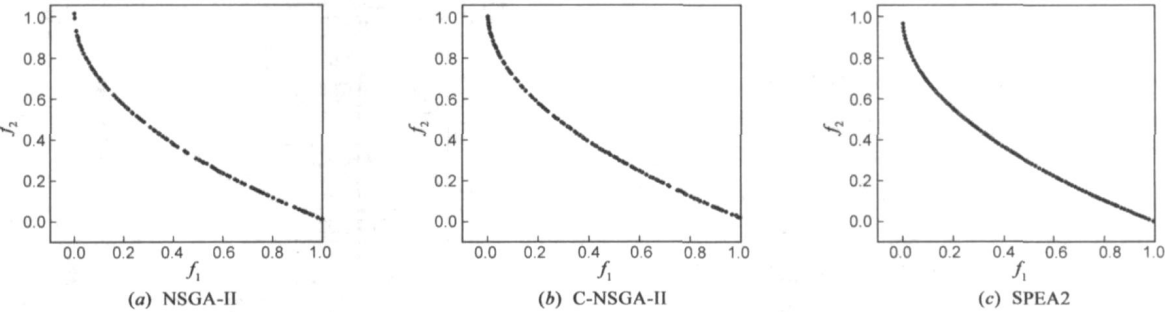


图4 3种算法在ZDT1上的最终解集分布情况

表 1 二维测试问题的解集分布度

算法	SCH	ZDT1	ZDT2	ZDT4
NSGA-II	0.5765354	0.7328515	0.7460961	0.5155786
G-NSGA-II	0.6248696	0.6781463	0.8118234	0.6363249
SPEA2	0.8167192	0.8487792	0.8742296	0.7339431

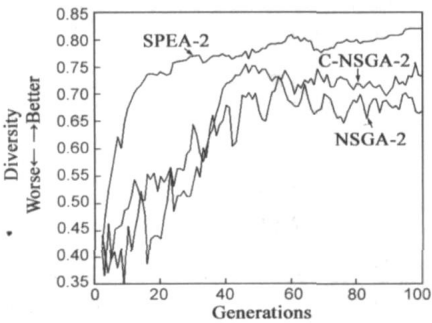


图5 三种算法对测试问题ZDT1的分布曲线

图5是用我们的方法对三种算法在运行过程中得到的解集进行分布情况评价得出的分布度曲线,从图中可以看出,SPEA2的解集随代数的增加分布度逐渐提高,分布情况逐渐均匀,并且曲线振幅很小,说明分布优秀的解丢失概率不大;G-NSGA-II的解集和NSGA-II的解集在40代左右以前分布度上升的比较快,而到了

640000. 算法的运行代数评价为评价个体的数目除以种群规模. 每个算法对各个测试函数独立运行 20 次,结果取平均值.

5.1 二维测试问题

我们选取了 4 个二维测试函数 SCH, ZDT1, ZDT2, ZDT4. SCH 被认为是在 MOP 领域具有代表性的测试函数, ZDT 是由 Zitzler 等提出的一系列测试函数^[1], 这些函数是较具有代表性的二维测试函数.

首先我们给出三种算法对 ZDT1 的最终解集分布图,如图 4 所示,然后采用本文提出的评价方法对三种算法进行评价,给出分布度曲线与运行代数之间的关系,如图 5 所示. 图 4 反映了三种算法对测试问题 ZDT1 的解集分布情况,从图中可以看出, SPEA2 的解集分布情况最好,个体的分布基本均匀, G-NSGA-II 的解集分布情况要稍好于 NSGA-II 的解集分布情况.

40 代以后,分布度提升很小,只是原地摆动,并且曲线振幅较大,其中 NSGA-II 的解集要比 G-NSGA-II 的解集振荡要强烈,说明在进化过程中分布优秀的解丢失更加严重. 表 1 为 2 维测试问题的分布度评价结果.

5.2 二维以上测试问题

下面我们对二维以上的测试问题进行分布度评价, DTLZ 系列是 Deb 提出的一种规模可变的测试函数^[11], 在这里我们取目标空间分别为 3, 4 和 6 维. 首先我们给出三种算法对 DTLZ1 在三维目标上的最终解集分布图,如图 6 所示,然后采用本文提出的评价方法对三种算法进行评价,给出分布度曲线与运行代数之间的关系,如图 7 所示.

图 6 反映了三种算法对测试问题 DTLZ1 在三维上的解集分布情况,从图中可以看出, NSGA-II 的解集收敛性最好(收敛到 $x + y + z = 1$ 平面), G-NSGA-II 次之(收敛到平面 $x + y + z = 2$), SPEA2 最差(收敛到平面 $x + y + z = 6$); 而分布性则正好相反, SPEA2 的解集分布情况最好,个体的分布基本均匀, G-NSGA-II 的解集分布情况要好于 NSGA-II 的解集分布情况. 图 7 是用我们的方法对三种算法在运行过程中得到的解集进行分布情

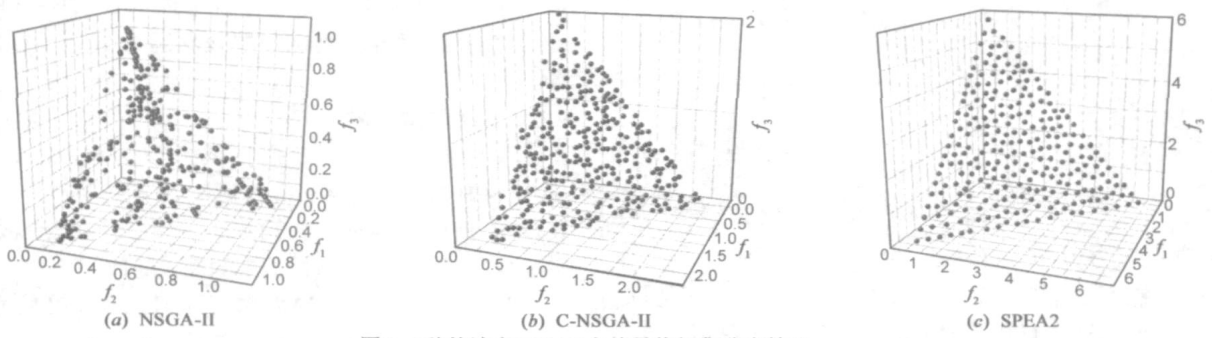


图6 3种算法在DTLZ1上的最终解集分布情况

们发现除了个别点分布较远以外, 解集的整体分布已经接近于图 6 (a) 中最终解集了, 这与图 7 中 NSGA-II 的分布度曲线相符. 表 2, 3 分别为用我们的评价方法对三种算法关于 DTLZ1 和 DTLZ2 在不同目标数上的分布度结果, 可以看出, 在不同的目标维数下 SPEA2 的分布性都是最好, G-NSGA-II 次之, NSGA-II 最差, 以上的实验结果表明我们的算法能正确评价二维以上测试问题的解集分布情况.

表 2 DTLZ1 在不同目标数上的分布度

目标维数	NSGA-II	G-NSGA-II	SPEA2
3	0.6708820	0.7529785	0.8702486
4	0.6892342	0.7778562	0.8973282
6	0.7079258	0.8199692	0.9123541

表 3 DTLZ2 在不同目标数上的分布度

目标维数	NSGA-II	G-NSGA-II	SPEA2
3	0.6572729	0.7573931	0.8618195
4	0.6725545	0.7873339	0.8960841
6	0.7222249	0.7985543	0.9163254

6 结论

一个有效的多目标进化算法, 除了要有较好的收敛程度外, 还必须保持较好的分布度. 因此, 收敛度与分布度都是衡量算法的重要指标. 本文主要针对分布度评价方法进行研究, 分析了国际上流行的多目标进化算法分布度评价方法的特点和不足, 提出了分布度逐步评价方法, 该方法利用了解集均匀分布具有的对称性, 把整个目标空间从大到小划分成不同的对称区域, 逐步进行分布度评价. 由于分布关系权值是由区域决定的, 这样影响量(权值)从大到小逐渐变化, 相应分布粒度从粗到细逐步变得精确. 为了证明方法的有效性, 我们选择了目前流行的几种 MOEAs 对 6 个测试函数在 2, 3, 4 和 6 维目标空间下进行实验. 通过实验证明, 我们的方法能精确的评价解集的分布情况.

需要指出的是, 在我们的算法中, 个体在新的坐标下的维数是在直角坐标下维数的组合数($m' = C_m^2$), 这样当目标空间维数较高时, 算法的计算量较大, 其时间

况评价得出的分布度曲线, 从图中可以看出, SPEA2 的分布度值最好, G-NSGA-II 的分布度值次之, NSGA-II 的分布度值最差, 这说明我们的评价方法避免了算法因收敛度不同对分布度评价造成的影响. 另外, 三种算法得到的解集在进化前期都存在不同程度的退化现象. SPEA2 的解集的分布度曲线在 50 代后开始稳步上升, 分布度曲线比较平滑, 说明分布优秀的解在进化过程中丢失概率不大; G-NSGA-II 的解集和 NSGA-II 的解集在进化的前期也有严重的退化现象, 分布度曲线在 60 代左右才开始上升, 而在 100 代左右以后, 分布度提升幅度变缓, 曲线基本处于原地摆动状态, 并且在整个进化过程中曲线振幅较大, 说明分布优秀的解丢失情况比较严重.

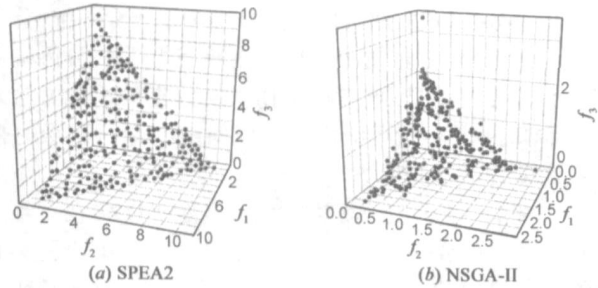


图8 SPEA2, NSGA-II对DTLZ1在100代时解集分布情况

图 8 (a) 为 SPEA2 对 DTLZ1 在 100 代时解集的分布情况, 对比图 6 (a), (b), 我们发现 SPEA2 在 100 代时的解集分布情况已经好于 NSGA-II 和 G-NSGA-II 最终解集的分布情况了, 这与我们图 7 中的分布度曲线一致. 图 8 (b) 为 NSGA-II 对 DTLZ1 在 100 代时解集分布情况, 我

复杂度较高。另外, 考虑到算法利用解集分布的对称性进行评价, 这样不适用于那些真实 Pareto 面并不对称的非连续问题。因此, 如何对非连续问题进行分布度评价是我们将来的研究工作。

参考文献:

- [1] Kalyanmoy Deb. Multi-Objective Optimization using Evolutionary Algorithms[M]. UK: John Wiley & Sons, Chichester, 2001. 1- 7.
- [2] 郑金华. 多目标进化算法及其应用[M]. 北京: 科学出版社, 2007. 20- 50.
- [3] Ali Farhang Mehr and Shapour Azarm. An information theoretic entropy metric for assessing multi objective optimization solution set quality. ASME[J]. Journal of Mechanical Design, 2003, 125(4): 655- 663.
- [4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182- 197.
- [5] Kalyanmoy Deb, Sachin Jain. Running Performance Metrics for Evolutionary Multi-Objective Optimization[R]. Kanpur India: KanGAL Report No. 2002004, 2002.
- [6] Kalyanmoy Deb, Manikant Mohan and Shikhar Mishra. A Fast Multiobjective Evolutionary Algorithm for Finding Well Spread Pareto-Optimal Solutions[R]. Kanpur India: KanGAL Report No. 2003002, 2003.
- [7] Sanaz Mostaghim, Jürgen Teich. The role of ϵ dominance in multi objective particle swarm optimization methods[A]. Proceedings of IEEE Congress on Evolutionary Computation (CEC 2003) [C]. Canberra, Australia: IEEE Press, 2003. 1764 - 1771.
- [8] Miqing Li, Jinhua Zheng, Guixia Xiao. Uniformity assessment for evolutionary multi objective optimization[A]. Proceedings of IEEE Congress on Evolutionary Computation (CEC' 2008) [C]. Hongkong: IEEE Press, 2008. 625- 632.
- [9] Schoot J R. Fault tolerant design using single and multicriteria genetic algorithms optimization[D]. Cambridge: Massachusetts Institute of Technology, 1995.
- [10] Eckart Zitzler, Marco Laumanns, Lothar Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm [R]. Zurich, Switzerland: TIK-Report No. 103, 2001.
- [11] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization [A]. Evolutionary Multiobjective Optimization: Theoretical Advances and Applications[C]. London: Springer Verlag, 2005. 105- 145.

作者简介:



李密青 男, 1981 年 10 月生于湖南常德, 湘潭大学信息工程学院硕士研究生. 主要研究方向为多目标进化算法.
E-mail: limit1008@126.com



郑金华 男, 1963 年 12 月生于湖南邵阳. 教授, 博士生导师, CCF 高级会员. 主要研究领域为进化算法, 智能科学等.
E-mail: jhzheng@xtu.edu.cn