

# 一种新的分布性保持方法

李密青, 郑金华, 伍 军

(湘潭大学 信息工程学院, 湖南 湘潭 411105)

**摘要:** 分布性保持是多目标进化算法主要目标之一. 然而通常维护方法的性能与运行时间存在矛盾. 提出一种基于最小生成树的分布性维护方法. 利用最小生成树中的度数和边长对个体密度进行估计, 使低度数的边界个体和长边长的低密度个体得到了保留. 另外, 一次性选择个体进入下代种群, 避免了每移出一个个体就需要对个体密度进行调整的操作. 通过 5 个测试问题和 4 个方面的测试标准, 与 3 个著名的算法进行比较实验, 结果表明该方法在以较快速度对种群进行维护的同时, 拥有良好的分布性.

**关键词:** 多目标优化; 进化算法; 分布性维护; 最小生成树

**中图分类号:** TP18      **文献标识码:** A

## A novel method for maintaining the diversity in evolutionary multiobjective optimization

LI Mi-qing, ZHENG Jin-hua, WU Jun

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

**Abstract:** Maintaining the diversity of solutions is a crucial part in multi-objective optimization. However, there has to be a trade-off between the diversity and the execution time. A method for maintaining the diversity using a minimum spanning tree is proposed. By estimating the individual density based on the degree and edge of the minimum spanning tree, we preserve the low-degree boundary individuals and the longer-edge-low-density individuals. Moreover, by this one-time selection, the adjustment of individual density after removing each individual can be avoided. Through the extensive comparison study with three other classical methods on four performance metrics in five test problems, it is observed that the proposed method has a good performance in diversity and execution time.

**Key words:** multi-objective; evolutionary algorithms; diversity maintenance; minimum spanning tree

### 1 引言(Introduction)

进化算法(evolutionary algorithm, EA)是一类模拟生物自然选择与自然进化的随机搜索算法, 因其适用于求解高度复杂的非线性问题而得到了非常广泛的应用. 在解决只有单个目标的复杂系统优化问题时, 进化算法的优势得到了充分展现. 然而, 现实世界的很多问题通常由多个目标组成, 解决多目标优化问题(multi-objective optimization problem, MOP)是很困难的, 这些目标往往是互相冲突的, 它不同于单目标优化问题(single objective optimization problem, SOP)那样只有一个最优解, 而是只能在相互冲突的目标间做出折衷, 得到一组非劣解<sup>[1]</sup>.

近年来, 研究者们提出了很多有效的多目标进化

算法(multi-objective evolutionary algorithms, MOEAs), 典型的有 Knowles 等提出的自适应网格算法 AGA<sup>[2]</sup>, Deb 等提出的非劣排序遗传算法 NSGA-II<sup>[3]</sup>, Zitzler 等提出的强 Pareto 进化算法 SPEA2<sup>[4]</sup> 等.

多目标优化的 3 个基本要素为收敛性, 分布性和运行时间. 其中分布性是指解集分布的均匀程度和广泛程度. 如何找寻一种算法能够在较短的时间内求得一个分布性较好的解集是多目标进化算法的一个重要的研究课题. 目前存在的分布性维护方法基本可分为两大类. 一类为用区域对分布性进行保持, 把目标空间划分为很多小区域, 固定每个区域内的个体数或剔除最密集区域内的个体, 如 PESA-II<sup>[5]</sup>, AGA<sup>[2]</sup>, EPS-MOEA<sup>[6]</sup> 等等; 另一类为基于密度的方

法,给每个个体定义一个密度,按个体密度对种群进行维护.基于密度的方法又可以细分为两小类,一类是剔除一些密度较大的个体对种群进行修剪,一类是选择密度较小的个体进入下一代种群.由于前一类方法中个体密度值可具有动态性,较能准确的反映个体的疏密情况,因此绝大多数基于密度方法都属于前一类,如SPEA2<sup>[4]</sup>,MST-MOEA<sup>[7]</sup>等,而后一类的方法由于在选择过程中不需要再对密度信息进行调整,因而具有较快的速度,如NSGA-II<sup>[3]</sup>.

本文提出一种基于最小生成树的种群维护方法,属于第2类中的第2小类.利用最小生成树中的度数和边长对个体密度进行估计,一次性选择个体进入下代种群.通过多个函数实验,测试和讨论了算法的性能.

## 2 基本概念(Basic conception)

最小化与最大化问题可以互相转化,因此,仅以最小化多目标问题为研究对象.多目标问题的一般描述为:

给定决策向量 $X = (x_1, x_2, \dots, x_n)$ ,它满足下列约束:

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

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

设有 $r$ 个优化目标,且这个 $r$ 个优化目标是相互冲突的,优化目标可以表示为:

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

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(也称优越关系). 设 $p$ 和 $q$ 是进化群体中的任意两个不同的个体,称 $p$ 支配 $q$ ,则必须满足下列2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. 此时称 $p$ 为非支配的,  $q$ 为被支配的. 表示为 $p \succ q$ , 其中“ $\succ$ ”是支配关系.

**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  $\wedge (f_i(X) = f_i(X^*))$ ,

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 $\Omega$ 是满足式(1)和式(2)的可行解集, 即

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

**定义 3** Pareto最优边界(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

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

**定义 4** 非支配集(也称非劣解). 设有解集 $P$ ,  $P$ 中的个体 $q$ 不被任何其他个体支配, 则 $q$ 是 $P$ 中的非支配个体;  $P$ 的非支配个体构成的子集称为 $P$ 的非支配集NDset, 即 $\text{NDset} = \{q | q \in P \text{ 且 } \nexists p \in P, \text{ 使 } p \succ q\}$ .

## 3 新的种群维护方法(New diversity maintenance method)

PESA-II<sup>[5]</sup>等第1类方法用区域对种群进行维护, 虽然拥有较快的速度, 但解集分布性很大程度上依赖于区域尺寸的设置. 第2类第1小类中一些方法虽然能较好的维护种群分布, 但通常具有较高的时间复杂度, 如SPEA2<sup>[4]</sup>用第 $k$ 小距离对种群进行维护, 时间复杂度为 $O(MN^2 \log N)$ ; MST-MOEA<sup>[7]</sup>定义一种密度估计指标树聚集距离对种群进行维护, 时间复杂度可达 $O(N^3)$ . 另外, NSGA-II<sup>[3]</sup>利用聚集距离一次性选择个体进入下代种群, 属于第2类中第2小类, 算法虽然拥有较低的时间复杂度 $O(MN \log N)$ , 但种群分布结果并不十分理想. 本文提出一种与NSGA-II相同类型的种群维护方法, 利用最小生成树的度数和边长对个体进行密度估计, 一次性选择较小密度的个体进入下代种群, 其具体流程如下:

### 算法 1

**参数设置**  $Q$ : 待维护的非支配集,  $N$ : 种群规模.

**步骤 1** 计算 $Q$ 中所有个体之间的欧氏距离, 并根据欧氏距离用Prim算法生成一棵最小生成树 $T$ , 记录下每条边 $E_{ij}$ , 其中 $i, j$ 为与其连接的两个个体; 同时对任意个体 $i$ , 记录下其度数 $d_i$ , 以及与其连接的最短边 $E_{i, \min}$ , 并记与 $i$ 构成 $E_{i, \min}$ 的另一个个体为 $i_{\min}$ .

**步骤 2** 统计最小生成树中度数为1(即 $d_i = 1$ )的个体数目, 记为COUNT.

**步骤 3** 若 $\text{COUNT} = N$ , 加度数为1的个体到下代种群, 算法结束; 否则若 $\text{COUNT} < N$ , 转步骤6.

**步骤 4** 对度数为1的个体, 按连接其边的边长(即 $|E_{i, \min}|$ )从大到小的顺序进行排序.

**步骤 5** 选择前  $N$  个度数为 1 的个体进入下代种群, 算法结束.

**步骤 6** 加度数为 1 的个体到下代种群, 并对  $T$  中所有边按边长从大到小的顺序进行排序, 记为  $E^1, E^2, \dots, E^k, \dots, E^{|Q|-1}$ , 赋  $k$  初始值为 1.

**步骤 7** 对  $E_{i,j}^k$  中的  $i$ : 若  $i$  已进入下代种群或者  $i$  未进入下代种群但  $|E_{i,j}^k| > |E_{i,\min}|$  并且  $i_{\min}$  已进入下代种群, 转步骤 9; 否则加  $i$  到下代种群.

**步骤 8**  $COUNT = COUNT + 1$ , 若  $COUNT = N$ , 算法结束.

**步骤 9** 对  $E_{i,j}^k$  中的  $j$ : 若  $j$  已进入下代种群或者  $j$  未进入下代种群但  $|E_{i,j}^k| > |E_{j,\min}|$  并且  $j_{\min}$  已进入下代种群,  $k = k + 1$  转步骤 7; 否则加  $j$  到下代种群.

**步骤 10**  $COUNT = COUNT + 1$ , 若  $COUNT = N$ , 算法结束; 否则  $k = k + 1$ , 转步骤 7.

需要注意的是, 度数为 1 的个体只有一条边与其相连, 因此步骤 4 即是对  $E_{i,\min}$  按长度进行排序. 下面通过两个例子说明以上维护步骤.

图 1(a)~(d) 为二目标下种群维护实例, 图中非支配集  $|Q| = 7$ , 种群规模  $N = 4$ ,  $A, B, C, D, E, F, G$  为待维护的非支配个体.

步骤 1 生成一棵最小生成树(如图 1(a));

步骤 2 统计出度数为 1 个体数目;

步骤 3 比较  $COUNT$  与  $N$ , 由于  $COUNT = 2 < N = 4$ , 转步骤 6;

步骤 6 首先个体  $A, G$  进入下代种群(如图 1(b)), 然后对最小生成树中边进行排序得  $E_{E,F}, E_{C,D}, E_{A,B}, E_{D,E}, E_{F,G}, E_{B,C}$ ;

步骤 7 对边  $E_{E,F}$  的个体  $E$ , 有  $D = E_{\min}$ , 由于  $|E_{E,F}| > |E_{E,\min}| = |E_{D,E}|$  且  $E$  和  $D$  并没有加入到下代种群, 因此  $E$  进入下代种群(如图 1(c)), 进步骤 8;

步骤 8  $COUNT = 3 < N$  进步骤 9;

步骤 9 对  $E_{E,F}$  的个体  $F$ , 有  $G = F_{\min}$ , 由于  $|E_{E,F}| > |E_{F,\min}| = |E_{F,G}|$  并且  $F$  未进,  $G$  已进下代种群, 因此转步骤 7; 同理边  $E_{C,D}$  中个体  $C$  进入下代种群(如图 1(d)),  $COUNT = 4 = N$ , 算法结束.

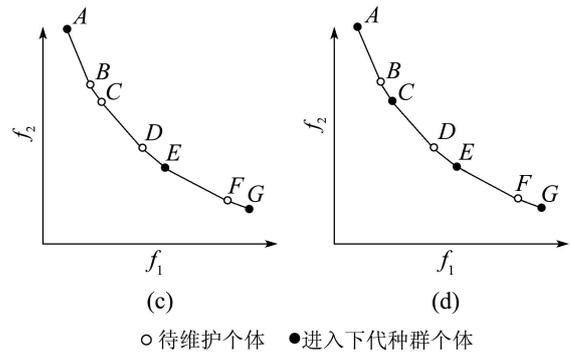
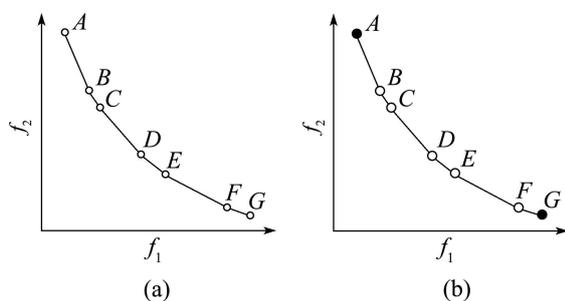


图 1 二目标下种群维护实例

Fig. 1 Illustration of population maintenance for 2-objective problem

图 2(a) (e) (f) 为三目标下种群维护实例, 图中非支配集  $|Q| = 7$ , 种群规模  $N = 5$ ,  $A, B, C, D, E, F, G$  为待维护的非支配个体.

步骤 1 生成一棵最小生成树(如图 2(a));

步骤 2 统计出度数为 1 个体数目;

步骤 3 比较  $COUNT$  与  $N$ , 由于  $COUNT = 4 < N = 5$ , 转步骤 6;

步骤 6 首先个体  $A, E, F, G$  进入下代种群(如图 2(e)), 然后对最小生成树中边进行排序得  $E_{C,F}, E_{D,G}, E_{A,B}, E_{B,D}, E_{D,E}, E_{B,C}$ ;

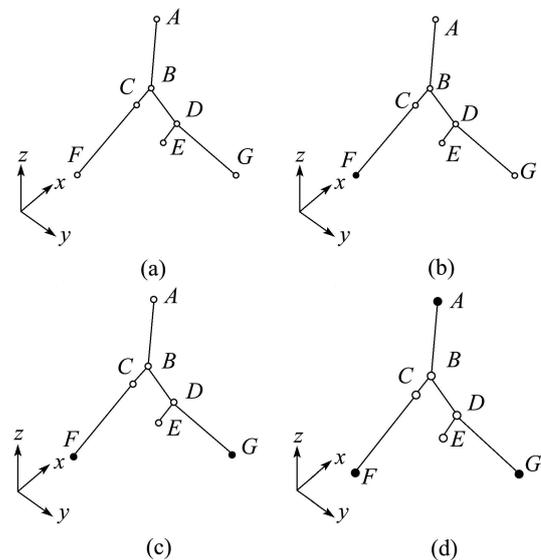
步骤 7 对边  $E_{C,F}$  的个体  $C$ , 有  $B = C_{\min}$ , 由于  $|E_{C,F}| > |E_{C,\min}| = |E_{B,C}|$  并且  $C$  和  $B$  并没有加入到下代种群, 因此  $C$  进入下代种群(如图 2(f)) 进步骤 8;

步骤 8  $COUNT = 5 = N$ , 算法结束.

对上例若种群规模  $N = 3$ , 由于  $COUNT = 4 > N = 3$ , 进步骤 4;

步骤 4 对边  $E_{A,\min}, E_{E,\min}, E_{F,\min}, E_{G,\min}$  进行排序, 得个体顺序为  $F, G, A, E$ ;

步骤 5 选前  $N$  个个体进入下代种群, 即  $F, G, A$  依次进入下代种群, 如图 2(b) (c) (d) 所示, 算法结束.



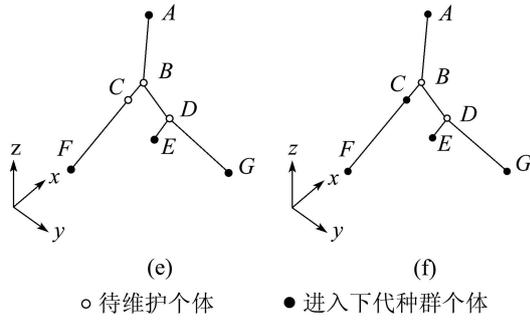


图2 三目标下种群维护实例  
Fig. 2 Illustration of population maintenance for 3-objective problem

本文的算法首先选择度数为1的个体进入下代种群,这是因为最小生成树中的度数在一定程度上反映了个体的疏密度,度数低表明个体与其它个体联系松散,个体密度相对较低,个体较具有代表性.另外,度数还能反映个体的位置关系,通常边界个体的具有较低的度数.特别是对二目标问题,根据定义1可知非支配种群中两个边界个体即为最小生成树中仅有的两个度数为1的个体.这样本文的算法将使边界更早的进入下代种群,很好的维护了种群的分布广度.在度数为1的个体进入下代种群后,本文对最小生成树中的边进行排序,选择较长边中的个体进入下代种群.需要指出的是,并不是所有较长边中的个体都具有较低的密度,图1中的 $F, G$ 个体就非常接近,因而标记出距离每个个体最近的个体,避免了彼此很近的相邻个体同时进入下代种群,这样较好的维护了种群的均匀性.下面对算法1的时间复杂度进行分析.

算法1中步骤1的时间耗费包括两个方面,一为计算种群中任意两个个体之间的欧氏距离,二为对种群生成最小生成树.计算任意两个个体之间欧氏距离的时间复杂度为 $O(MN^2)$ ,由Prim算法对种群生成最小生成树的时间复杂度为 $O(N^2)$ ,注意通常 $2N \geq |Q| \geq N$ ,本文用种群规模 $N$ 表示时间复杂度.步骤2为统计树中度数为1个体数目,其时间复杂度为 $O(N)$ .步骤3根据COUNT与 $N$ 的关系把算法流程分为两个分支,若COUNT大于 $N$ 执行步骤4,5;若COUNT小于 $N$ 执行步骤6到10.下面分别考虑其时间复杂度.步骤4中由于连接度数为1的个体边长数小于 $N$ ,因此对其快速排序的时间复杂度为 $O(N \log N)$ .步骤5选择前 $N$ 个度数为1的个体进入下代种群的时间复杂度为 $O(N)$ ,因此COUNT大于 $N$ 情况的时间复杂度为

$$O(N \log N) + O(N) = O(N \log N).$$

步骤6对 $|Q| - 1$ 条边进行排序,其时间复杂度为 $O(N \log N)$ ,步骤7到10为选择 $(N - \text{COUNT}_{d=1})$

个个体进入下代种群,其中 $\text{COUNT}_{d=1}$ 为树中度数为1个体数目,其时间复杂度为 $O(N)$ .因此COUNT小于 $N$ 情况的时间复杂度同样为

$$O(N \log N) + O(N) = O(N \log N).$$

综上所述算法1的总时间复杂度为

$$O(MN^2) + O(N^2) + O(N \log N) = O(MN^2).$$

## 4 实验设计与结果(Experiment designs and results)

### 4.1 实验参数设置(Experiment parameter settings)

为检验所提分布性维护方法的有效性,将其与拥有较快速度的3种多目标进化算法NSGA-II<sup>[3]</sup>, C-NSGA-II<sup>[6]</sup>, PESA-II<sup>[5]</sup>的分布性进行实验比较. NSGA-II拥有很快的速度; C-NSGA-II为对NSGA-II的改进,其分布性要好于NSGA-II; PESA-II采用网格技术维持种群分布性.本文的方法在其他方面的设置(如适应度赋值等)与NSGA-II相同.这里选择了5个测试函数来比较测试4种方法的性能,描述如表1所示,这些多目标测试函数有连续的,非连续的,凸的,凹的等.

前面已经提到多目标进化算法中的分布性是指解集分布的均匀性和广泛性.在此本文用SP<sup>[1]</sup>, UA<sup>[8]</sup>对不同算法得到的解集进行均匀性评价,用D<sup>[9]</sup>对其进行广泛性评价.下面简要地介绍3种评价方法. SP用来评价解集的均匀性,其评价函数定义如下:

$$\text{SP} = \sqrt{\frac{1}{|Q| - 1} \sum_{i=1}^{|Q|} (\bar{d} - d_i)^2}, \quad (4)$$

$$d_i = \min_{q_j \in Q \wedge q_j \neq q_i} \sum_{m=1}^M |f_m(q_i) - f_m(q_j)|. \quad (5)$$

$d_i$ 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.解集的SP值越小表明分布的越均匀,理想的情况是 $\text{SP} = 0$ ,但SP受一些其他因素影响(如:种群规模,收敛性,连续性等)并不完全准确<sup>[10]</sup>. UA是一种鲁棒性很强的解集均匀性的评价方法,基本不受其他因素的影响,其评价结果在0到1之间,越接近1表明分布的越均匀. D用来评价解集的分布广度,值越大表明解集分布越广泛.

在实验中,4种算法都采用实数编码,交叉概率0.9,变异概率 $1/\text{nreal}$ , nreal为决策变量维数,2维测试函数的规模为100,评价次数为20000;3维测试函数的规模为200,评价次数为100000; PESA-II 2维函数下网格数为 $2^5 \times 2^5$ ,在3维函数下网格数为 $2^3 \times 2^3 \times 2^3$ .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. 实验运行在1.7 GHz CPU, 256 M内存, Windows XP环境下.

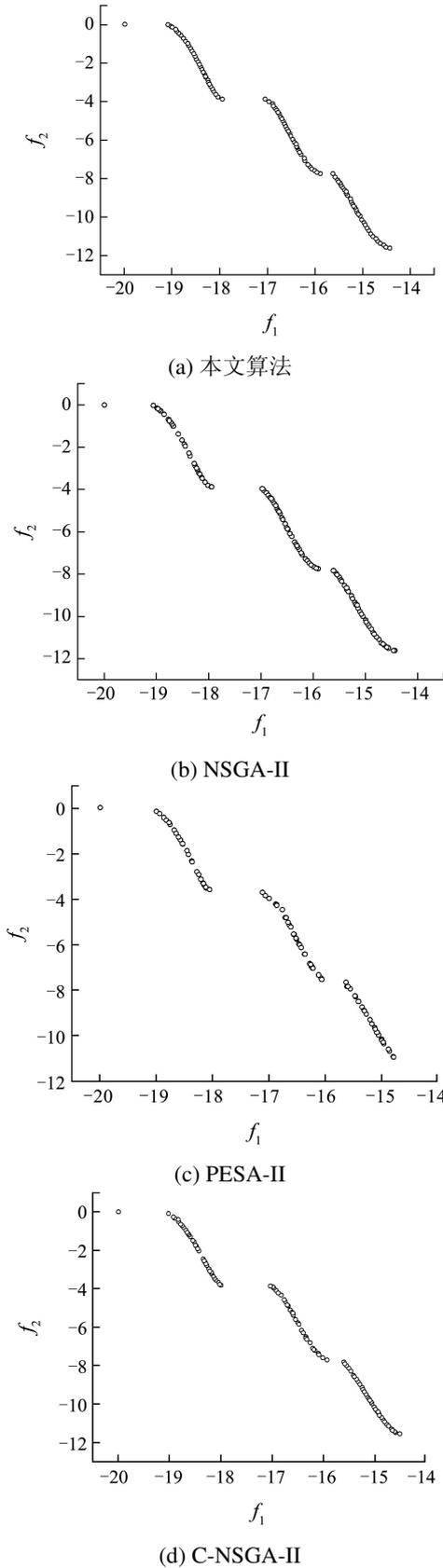


图 3 4种维护方法在KUR上的最终解集分布  
Fig. 3 The final solutions obtained by four methods on KUR

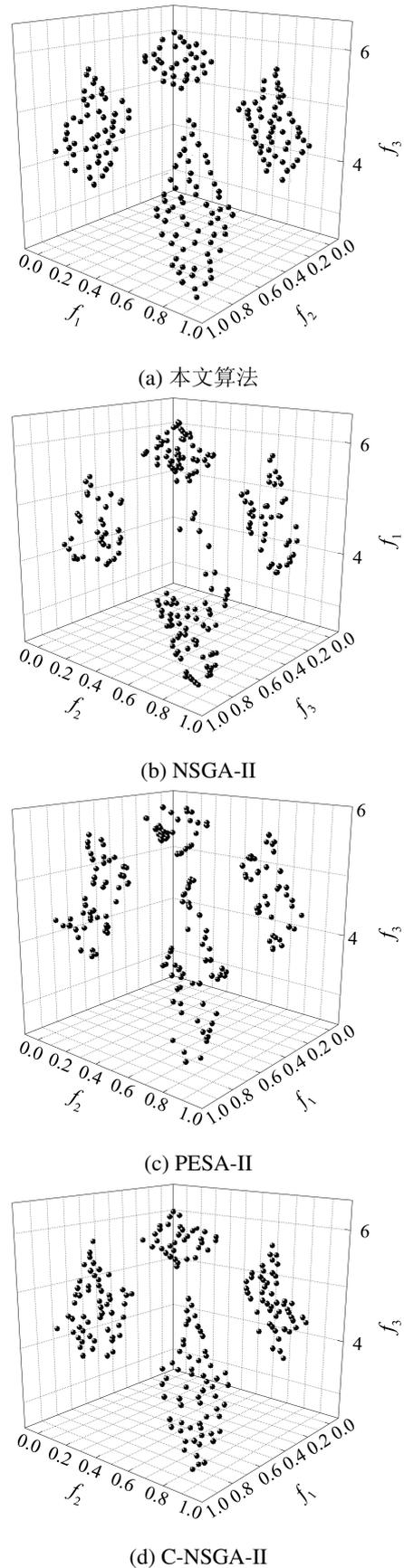


图 4 4种维护方法在DTLZ7上的最终解集分布  
Fig. 4 The final solutions obtained by four methods on DTLZ7

## 4.2 分布结果比较(Distribution results comparison)

选取了5个测试函数, KUR<sup>[1]</sup>, ZDT1, ZDT6<sup>[9]</sup>为2维测试函数, DTLZ2, DTLZ7<sup>[11]</sup>为3维测试函数. KUR的最优边界是4段非连续的曲线, 其中最上面一段仅为一个点, ZDT1, ZDT6的最优边界分别为凸形, 凹形的曲线, DTLZ2的最优边界为球形, 而DTLZ7具有一组不连续的最优边界.

表1为测试函数分布性评价结果, 在每个单元格中, 第1行为均值, 第2行为标准差. 在UA上本文的

结果要明显优于其他3种算法的结果. 在SP上, 除了在KUR上略逊于PESA-II, 在DTLZ7上略逊于C-NSGA-II外, 在其他函数上均拥有最优值. 事实上, 由于SP受一些其他因素影响造成的, 其均匀性评价结果并不准确<sup>[10]</sup>, 从图3中可以看出对函数KUR本文的方法得到的解集均匀性要好于PESA-II, 但由于对函数KUR, PESA-II得到的解集最上面一段通常为两个个体, 因而对SP值产生干扰. 在分布广度评价指标D上, 本文的算法除了在DTLZ7上要略逊于C-NSGA-II外, 在其他函数上都拥有最优值.

表1 4种维护方法性能比较

Table 1 Performance comparison of the four methods

问题	算法	SP	UA	D	t/s
KUR	NSGA-II	1.0551E-01 ± 3.7602E-03	4.1422E-01 ± 4.6492E-02	1.2893E+01 ± 2.2128E-03	9.3428E-01 ± 3.0036E-02
	PESA-II	8.4301E-02 ± 5.4566E-03	2.7496E-01 ± 2.3771E-02	1.2464E+01 ± 1.6276E-01	3.9508E+00 ± 7.9086E-02
	C-NSGA-II	1.0226E-01 ± 6.2411E-03	5.3969E-01 ± 1.6539E-02	1.2851E+01 ± 3.9352E-02	1.9344E+00 ± 3.3380E-02
	本文算法	8.5711E-02 ± 3.6514E-03	7.0834E-01 ± 4.1462E-02	1.2894E+01 ± 9.8002E-03	1.6406E+00 ± 3.9088E-02
ZDT1	NSGA-II	7.5588E-03 ± 1.0122E-03	4.1389E-01 ± 4.7899E-02	1.4134E+00 ± 8.4719E-04	1.4017E+00 ± 3.1219E-02
	PESA-II	8.2116E-03 ± 1.1264E-03	3.8801E-01 ± 3.1464E-02	1.3842E+00 ± 2.1365E-02	3.8786E+00 ± 7.8469E-02
	C-NSGA-II	5.3985E-03 ± 4.9801E-04	5.7705E-01 ± 3.1117E-02	1.4041E+00 ± 4.3405E-03	2.4283E+00 ± 6.8198E-02
	本文算法	4.2296E-03 ± 2.9553E-04	7.2798E-01 ± 1.5245E-02	1.4139E+00 ± 5.0169E-04	2.1062E+00 ± 7.1795E-02
ZDT6	NSGA-II	5.1495E-03 ± 4.6986E-04	4.5949E-01 ± 3.7466E-02	1.0461E+00 ± 6.9547E-03	1.1469E+00 ± 9.0027E-02
	PESA-II	8.1591E-03 ± 1.0517E-03	3.5731E-01 ± 4.1607E-02	1.0456E+00 ± 3.9879E-03	2.8130E+00 ± 4.8158E-02
	C-NSGA-II	3.9843E-03 ± 4.1080E-04	5.6189E-01 ± 2.9189E-02	1.0415E+00 ± 1.9781E-03	1.7704E+00 ± 8.9926E-02
	本文算法	3.1148E-03 ± 2.0878E-04	7.0462E-01 ± 2.9053E-02	1.0468E+00 ± 2.5272E-03	1.2874E+00 ± 4.4884E-02
DTLZ2	NSGA-II	4.1831E-02 ± 2.1634E-03	3.6837E-01 ± 1.1992E-02	1.7455E+00 ± 6.7949E-03	1.0600E+01 ± 4.0608E-02
	PESA-II	4.1732E-02 ± 1.7522E-03	3.5879E-01 ± 2.8999E-02	1.7550E+00 ± 2.2388E-02	4.0177E+01 ± 8.6841E-02
	C-NSGA-II	2.7415E-02 ± 1.3051E-03	6.9146E-01 ± 1.3551E-02	1.7853E+00 ± 4.4669E-02	4.2678E+01 ± 4.1237E-01
	本文算法	2.5539E-02 ± 1.7344E-03	7.0906E-01 ± 2.2469E-02	1.7865E+00 ± 2.1568E-02	3.9059E+01 ± 1.0049E-01
DTLZ7	NSGA-II	3.2491E-02 ± 1.6232E-02	4.0483E-01 ± 1.3817E-02	3.5301E+00 ± 4.6207E-02	8.8854E+00 ± 3.0742E-02
	PESA-II	4.7411E-02 ± 4.5493E-03	3.1750E-01 ± 2.3706E-02	3.3987E+00 ± 1.0835E-01	3.0925E+01 ± 7.9898E-02
	C-NSGA-II	2.5098E-02 ± 2.0022E-03	6.6147E-01 ± 1.7277E-02	3.7529E+00 ± 1.7491E-01	4.8428E+01 ± 1.3242E-01
	本文算法	2.6771E-02 ± 2.1200E-03	6.9202E-01 ± 1.0267E-02	3.7032E+00 ± 6.6769E-02	3.5669E+01 ± 3.2826E-01

## 4.3 运行时间比较(Time comparison)

从表1中可以发现, NSGA-II拥有最快的速度, 本文的算法除了在DTLZ7上逊于PESA-II外, 在其他函数上都要好于PESA-II和C-NSGA-II. 需要指出的是, C-NSGA-II与本文的方法具有相同的时间复杂度, 但由于本文的方法是选择优秀个体而不是淘汰低劣个体, 不需要对种群密度信息进行调整, 节省了时间, 因此具有更快的维护速度.

## 5 结论(Conclusion)

种群维护是多目标进化算法的重要组成部分, 维护方法的好坏决定着最终解集的分布状况. 本文尝试这方面的工作, 提出一种基于最小生成树的种群维护方法, 利用最小生成树的度数和边长对种群进行维护. 由于采用一次性选择个体进入下代种群, 并不需要每移出一个个体对种群信息进行调整, 因而提高了时间效率. 另外, 度数的加

入使边界个体基本得到保留,有利于种群获得广泛的非支配个体;最后通过与3个著名算法NSGA-II, C-NSGA-II和 PESA-II的比较实验,表明本文的方法在以较快速度对种群进行维护的同时,拥有很好的均匀性和广泛性。

### 参考文献(References):

- [1] DEB K. *Multi-objective Optimization using Evolutionary Algorithms*[M]. UK: John Wiley and Sons, 2001.
- [2] KNOWLES J, CORNE D. Properties of an adaptive archiving algorithm for storing nondominated vectors[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 100 – 116.
- [3] 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.
- [4] ZITZLER E, LAUMANN S M, THIELE L. SPEA2: Improving the strength Pareto evolutionary algorithm[R] // *Swiss Federal Institute of Technology (ETH) Zurich, Technical Report: TIK – Report103*. [S.l.]: [s.n.], 2001.
- [5] CORNE D, JERRAM N, KNOWLES J, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization[C] // *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*. San Francisco, California: Morgan Kaufmann Publishers, 2001, 283 – 290.
- [6] DEB K, MOHAN M, MISHRA S. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions[J]. *Evolutionary Computation*, 2005, 13(4): 501 – 525.
- [7] LI M Q, ZHENG J H, XIAO G X. An efficient multi-objective evolutionary algorithm based on minimum spanning tree[C] // *Proceedings of IEEE Congress on Evolutionary Computation (CEC'2008)*. Hongkong: IEEE, 2008: 617 – 624.
- [8] LI M Q, ZHENG J H, XIAO G X. Uniformity assessment for evolutionary multi-objective optimization[C] // *Proceedings of IEEE Congress on Evolutionary Computation (CEC'2008)*. Hongkong: IEEE, 2008: 625 – 632.
- [9] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: empirical results[J]. *Evolutionary Computation*, 2000, 8(2): 173 – 195.
- [10] 李密青, 郑金华, 谢炯亮, 等. 一种MOEA分布度的逐步评价方法[J]. *电子学报*, 2008, 36(10): 1986 – 1991.  
(LI Miqing, ZHENG Jinhua, XIE Jiongliang, et al. A stepwise diversity metric for MOEA[J]. *Acta Electronica Sinica*, 2008, 36(10): 1986 – 1991.)
- [11] DEB K, THIELE L, LAUMANN S M, et al. Scalable test problems for evolutionary multiobjective optimization[C] // *Proceedings of Evolutionary Multiobjective Optimization, Theoretical Advances and Applications*. USA: Springer, 2005: 105 – 145.

### 作者简介:

李密青 (1981—), 男, 硕士, 助教, 主要研究方向为多目标进化算法, E-mail: limit1008@126.com;

郑金华 (1963—), 男, 教授, 博士生导师, 主要研究方向为进化计算及智能科学等, E-mail: jhzheng@xtu.edu.cn, 本文通讯作者;

伍军 (1982—), 男, 硕士研究生, 主要研究方向为进多目标进化算法.