

# 一种快速的基于占优树的多目标进化算法\*

石川<sup>1,2+</sup>, 李清勇<sup>1,2,3</sup>, 史忠植<sup>1</sup>

<sup>1</sup>(中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100080)

<sup>2</sup>(中国科学院 研究生院,北京 100049)

<sup>3</sup>(北京交通大学 计算机与信息技术学院,北京 100044)

## A Quick Multi-Objective Evolutionary Algorithm Based on Dominating Tree

SHI Chuan<sup>1,2+</sup>, LI Qing-Yong<sup>1,2,3</sup>, SHI Zhong-Zhi<sup>1</sup>

<sup>1</sup>(Key Laboratory of Intelligent Information Process, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)

<sup>2</sup>(Graduate School, The Chinese Academy of Sciences, Beijing 100049, China)

<sup>3</sup>(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

+ Corresponding author: Phn: +86-10-62565533 ext 5689, E-mail: shic@ics.ict.ac.cn

**Shi C, Li QY, Shi ZZ. A quick multi-objective evolutionary algorithm based on dominating tree. *Journal of Software*, 2007,18(3):505–516. <http://www.jos.org.cn/1000-9825/18/505.htm>**

**Abstract:** To solve the time-consuming problem of the fitness assignment in the multi-objective evolutionary algorithm, this paper proposes a novel fitness assignment—dominating tree. The dominating tree preserves the necessary relationships among individuals, contains the density information implicitly, and reduces the comparisons among individuals distinctly. In addition, a smart eliminating strategy based on the dominating tree maintains the diversity of the population without extra expenses. A new multi-objective evolutionary algorithm based on dominating tree is proposed on these innovations. By examining three performance metrics on six test problems, the new algorithm is found to be competitive with SPEA2 and NSGA-II in terms of converging to the true Pareto front and maintaining the diversity of the population, moreover, it is much faster than other two algorithms.

**Key words:** multi-objective evolutionary algorithm; evolutionary algorithm; dominating tree; eliminating strategy

**摘要:** 为了解决多目标进化算法中适应值指派(fitness assignment)的耗时问题,提出了一种新颖的适应值指派方法——占优树。占优树保存了个体之间的必要信息,暗含了个体的密度信息,而且显著减少了个体之间的比较。此外,基于占优树的淘汰策略没有花费额外的代价就保存了种群多样性。在此基础上,提出了一种新的基于占优树的多目标进化算法。通过6个测试问题和3个方面的测试标准,新算法在接近真实的最优前沿和保持种群的多样性方面,与SPEA2和NSGA-II性能相当,但速度要比它们快得多。

**关键词:** 多目标进化算法;进化算法;占优树;淘汰策略

\* Supported by the National Natural Science Foundation of China under Grant No.60435010 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2003CB317004 (国家重点基础研究发展规划(973)); the Natural Science Foundation of Beijing of China under Grant No.4052025 (北京市自然科学基金)

Received 2006-03-10; Accepted 2006-05-11

中图法分类号: TP18 文献标识码: A

多目标优化问题(multi-objective optimal problem,简称 MOP)是指那些同时优化多个目标的问题.一般来说,这些单个目标是相互冲突的,而且总体目标没有单个最优解.求解多目标优化问题常常是困难的,它不像单目标优化问题(single objective optimal problem,简称 SOP)那样只有一个最优解,这使得能够一次得到一组有希望的解的算法比那些一次只得到一个解的算法要好.正是由于这个原因,在过去的十几年里,越来越多的研究者将进化算法应用到 MOP 问题中.

研究者对多目标进化算法进行了深入而广泛的研究,提出了很多种多目标进化算法(multi-objective evolutionary algorithm,简称 MOEA)<sup>[1-6]</sup>.这些算法使用 Pareto 占优指导搜索,并且返回一组非占优解作为结果.与单目标优化问题不同,多目标优化的解要达到两个目标:(1) 靠近 Pareto 最优前沿;(2) 保持种群多样性<sup>[1]</sup>.为了达到这两个目标,使用了很多策略和方法<sup>[7]</sup>.这些算法对一些测试问题能够取得很好的结果,但是,现在的 MOEA 中也存在一些不足,很多 MOEA 比较复杂.为了取得更好的解,算法运用了很多启发式方法和策略,并且很多参数需要根据经验和给定问题的先验知识进行调整.另一方面,很多 MOEA 有较高的计算复杂度.因此,设计一种简单高效的 MOEA 是必要的.

在本文中,我们首先分析了 Pareto 占优的性质和适应值指派过程耗时的原因,然后提出了一种新颖的适应值指派方法——占优树.占优树是保存了个体占优信息的二叉树,它充分利用了 Pareto 占优的性质,显著减少个体之间的比较,而且隐含了个体之间的密度信息.此外,基于占优树的淘汰策略不需要额外的开销就能很自然地保持种群多样性.由此,我们提出了基于占优树的多目标进化算法(multi-objective evolutionary algorithm based on dominating tree,简称 DTEA).DTEA 将趋近前沿的策略和保持种群多样性策略整合到占优树中,这样使得算法更加简单.最后,模拟实验表明:DTEA 能够得到与 SPEA2 和 NSGA-II 性能相当的解,但是,DTEA 比其他两个算法要快得多.

## 1 多目标进化算法的回顾

我们将简短地回顾目前 MOEA 的研究.首先介绍文中使用的一些概念;然后分析目前 MOEA 中的技术和方法.另外也指出了目前 MOEA 研究中存在的一些问题.

### 1.1 基本定义

Veldhuizen 和 Lamont 已经严格地定义了多目标优化问题和相关的概念<sup>[7]</sup>.我们介绍本文中使用的两个主要概念.不失一般性,本文只讨论最小优化问题,因为最大优化问题很容易转化成最小优化问题.

定义 1<sup>[7]</sup>. 一般的 MOP: 一个 MOP 是指最小化目标函数  $F(x)=(f_1(x), \dots, f_m(x))$  并满足约束条件  $g_i(x) \leq 0$ ,  $i=1, \dots, k, x \in \Omega$  ( $\Omega$  是决策变量空间)的问题. 一个 MOP 的解是最小化  $m$  维的目标向量  $F(x)$  的各个分量,这里,  $x=(x_1, \dots, x_n)$  是一个  $n$  维决策向量.

定义 2<sup>[7]</sup>. Pareto 占优: 如果向量  $U=(u_1, \dots, u_m)$  Pareto 占优  $V=(v_1, \dots, v_m)$ , 则记为  $U \leq V$ . 即  $U \leq V$ , 当且仅当

$$\forall i \in \{1, \dots, m\} \quad u_i \leq v_i \wedge \exists i \in \{1, \dots, m\} \quad u_i < v_i \quad (1)$$

当前, MOEA 的研究主要是基于 Pareto 占优的. 一个决策向量  $x_u \in \Omega$  是 Pareto 最优的, 当且仅当不存在  $x_v \in \Omega$  满足  $F(x_v) \leq F(x_u)$ . Pareto 最优决策向量的集合被称为问题的 Pareto 最优集. 对应的目标向量集合被称为非占优集或者 Pareto 前沿. 为了简化, 在下面的叙述中, 我们也使用占优代替 Pareto 占优. 很明显, Pareto 占优有如下特性:

性质 1. Pareto 占优有反自反和传递性.

限于篇幅原因以及证明比较简单, 证明略.

### 1.2 MOEA 中使用的技术

一个好的 MOEA 必须满足两个方面: (1) 非占优解集接近 Pareto 最优前沿; (2) 解集有良好的分散性<sup>[1,7]</sup>. 这两个目标也是大部分 MOEA 的性能评价标准. 为了满足这两个标准也设计了很多种方法.

为了满足第一方面,基于 Pareto 占优的适应值指派方法用来指导种群向真实的最优前沿搜索.基本思想是根据 Pareto 占优对解进行排序.很多适应值指派方法已经被提出来了.Goldberg 首先提出一种常用的方法并应用到 NSGA-II 等算法中<sup>[1]</sup>.该方法将解集划分成不同的层并给予不同的排名值(rank).Fonseca 和 Fleming 提出一种方法,即一个解的排名由种群中比该解占优的解的数目决定<sup>[8]</sup>.Zitzler 和 Thiele 在 SPEA2 提出了另一种方法:在进化种群和精英种群中的每一个体都被指派一个强度值,该值包含了占优和密度信息<sup>[2]</sup>.

对于第二方面,一些成功的 MOEA 利用密度估计方法保存种群的多样性.小生镜(niching)和适应值共享在很多 MOEA 中被广泛地使用,例如 MOGA<sup>[8]</sup>.这种方法设置一个共享参数  $\delta_{share}$ .计算目标空间中个体的 Niching 距离并且与  $\delta_{share}$  比较.一般而言,对  $\delta_{share}$  指定合适的值是很困难的,因为这常常需要给定问题的先验知识.在 SPEA2 中,与第  $k$  个最近个体之间的距离作为个体的密度信息<sup>[2]</sup>.NSGA-II 使用了一种新颖的密度估计方法.该方法不需要设置参数,并且与 NSGA 相比有更低的时间复杂度<sup>[1]</sup>.另一种常用的方法是利用超网格的方法将目标空间划分成格子.一个格子中的个体比较少就说明空间比较稀疏,可以和其他个体共享一个格子.PAES<sup>[3]</sup>和 DMOEA<sup>[5]</sup>使用了该方法. Atashkari 还提出了  $\epsilon$ -淘汰的多样性方法<sup>[4,9]</sup>.

此外,其他一些技术也应用到 MOEA 中来.研究表明:精英方法(elite)可以显著提高 MOEA 的性能,并且这有助于防止丢失已经找到的优秀解<sup>[10]</sup>.根据种群的在线特性和密度分布信息动态调整种群大小的方法与固定种群规模大小的方法相比,在避免局部最优和减少不必要的复杂性方面更加有效<sup>[5]</sup>,而且,使用不受限的精英种群大小可以避免 Pareto 前沿的衰退和震荡<sup>[11]</sup>.

国内学者也在这方面进行了大量研究.针对很多算法无法有效处理目标函数很多的优化问题,崔逊学提出利用不同准则之间引入偏好来解决该问题,并设计了多目标调和遗传算法<sup>[12]</sup>.雷德明提出了基于个体密度距离的外部种群维护方法,并在将所有个体根据 Pareto 支配关系分成 4 个层次的基础上,给出了一种由个体密度距离定义的适应值函数<sup>[13]</sup>.曾三友将正交设计的方法应用到了多目标优化中<sup>[14]</sup>.

一些成功的 MOEA 已经应用到一些测试问题并取得了好的结果.但是,这些算法还存在一些不足:一方面,MOEA 算法很复杂.为了得到好的解,很多 MOEA 方法为了趋近前沿和保持多样性使用了不同的技术,虽然这些算法常常根据这两方面评价个体的适应值<sup>[1,2,5]</sup>.在这两方面中使用的技术常常是比较复杂的,并且有很多参数需要根据给定的问题和经验进行调整.例如,DMOEA 中有 6 个策略和 4 个参数需要调整<sup>[5]</sup>.另一方面,MOEA 算法很耗时.现在,大部分 MOEA 有较高的计算复杂度,其中部分原因在于多目标的适应值指派比单目标要复杂;另一个重要原因在于算法研究者很少考虑计算复杂度<sup>[15]</sup>.本文的研究目标就是设计一个简单但是高效的 MOEA 算法.

## 2 占优树及其性质

适应值指派是 MOEA 的主要耗时部分<sup>[15]</sup>.很多有效的适应值指派已经被提出来了.但是,在适应值指派中有很多不必要的比较.减少不必要的比较可能是降低适应值指派的计算复杂度的一个捷径.

### 2.1 现有适应值指派的不足之处

MOEA 一个主要的耗时部分在于适应值指派<sup>[15]</sup>.适应值指派有很高的计算复杂性,其中部分原因在于多目标的适应值指派是一个很难的问题;另一个更重要的原因在于,研究者在适应值指派过程中很少利用 Pareto 占优的性质和 MOP 的特性来减少个体之间不必要的比较.不必要的比较主要由两方面组成:一些关系可由现存的关系推导出来;另外,一些关系对最终决策是不必要的.在本文中,个体之间的占优关系是通过个体比较得到的.这些关系能够可视化,因此,解之间的比较可以用图表示,图中的一个结点表示一个个体.例如,图 1 利用图完全显示了 5 个结点之间的关系.我们直观地发现图中有很多冗余关系.那么,如何消除这些关系呢?

因为 Pareto 占优有反自反和传递性,很多关系可以从现有的关系中推导出来.以图 1 为例, $N_4$  占优  $N_3$ , $N_3$  占优  $N_2$ ,因此  $N_4$  占优  $N_2$ ,所以, $N_4$  和  $N_2$  之间的比较是没有必要的.我们也注意到另一个事实:只需要得到每一代种群中的 Pareto 最优集,因为决策者常常根据最优解作出决策,他们并不关注其他解.因此,解集中的很多关系是不需要知道的,很多比较是可以避免的.很明显, $N_1$  和  $N_4$  构成了图 1 中 5 个结点的 Pareto 最优集.因为  $N_4$  占优

$N3$ ,并且  $N1$  占优  $N2$ ,因此, $N2$  和  $N3$  的比较是不必要的.

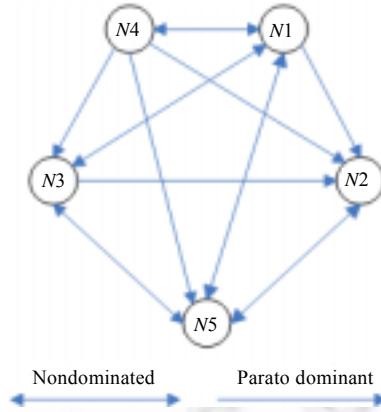


Fig.1 Illustration of the dominating relationships using graph

图 1 利用图表示的结点占优关系的示意图

2.2 占优树的结构

通过上面的分析我们发现,减少不必要的比较需要满足两个条件:(1) 必须消除冗余关系;(2) 为了容易找到最优解,必须保存必要的关系.

对第 1 个条件,我们需要进一步地分析 Pareto 占优的性质.SOP 中解的关系是二值的:大于或者小于关系(这里,我们不考虑等于的情况),MOP 中解的关系是三值的,可以定义如下:

定义 3. Better 函数.

$$Better(x_1, x_2) = \begin{cases} 1, & F(x_1) \preceq F(x_2) \\ 2, & F(x_2) \preceq F(x_1) \\ 0, & \text{Nondominated} \end{cases} \quad (2)$$

Better 函数定义了 MOP 中解的关系.图 1 中结点的 Better 关系见表 1.

Table 1 The Better relationships among nodes in Fig.1

表 1 图 1 中结点的 Better 关系

|    | N1 | N2 | N3 | N4 | N5 |
|----|----|----|----|----|----|
| N1 | 0  | 1  | 0  | 0  | 0  |
| N2 | 2  | 0  | 2  | 2  | 0  |
| N3 | 0  | 1  | 0  | 2  | 0  |
| N4 | 0  | 1  | 1  | 0  | 1  |
| N5 | 0  | 0  | 0  | 2  | 0  |

经过分析发现,Better 函数有如下性质:

性质 2. If  $Better(x_1, x_2)=1$ ,then  $Better(x_2, x_1)=2$ .If  $Better(x_1, x_2)=2$ ,then  $Better(x_2, x_1)=1$ .If  $Better(x_1, x_2)=0$ ,then  $Better(x_2, x_1)=0$ .

性质 3. If  $Better(x_1, x_2)=1, Better(x_2, x_3)=1$ ,then  $Better(x_1, x_3)=1$ .

性质 4. If  $Better(x_1, x_2)=1, Better(x_2, x_3)=0$ ,then  $Better(x_1, x_3) \neq 2$ .

限于篇幅原因以及证明比较简单,这部分的证明略.这些性质可以减少解之间的比较.性质 2 反映了 Pareto 占优的相反反性. $x_1$  和  $x_2$  的关系决定了  $x_2$  和  $x_1$  的关系,因此, $x_2$  和  $x_1$  的比较可以省略.性质 3 反映了 Pareto 占优的传递性,因此, $x_1$  和  $x_3$  的比较可以省略.性质 4 在比较过程中也是很有用的.

对于第 2 个条件,必须要设计一个好的方法,不仅容易找到 Pareto 最优集,而且要使用更少的比较.这种方法应该要满足下面的条件:(1) Pareto 最优集中结点的关系必须保存以便容易找到最优结点;(2) 每个结点必须至

少与其他结点有一个关系,否则它是一个孤立结点;(3) 一些中间结点至少与其他结点有两种关系,否则这些结点不是连通的.经过分析,二叉树满足这些条件.图 2 使用二叉树表示图 1 中结点的关系.如图 2 所示,Pareto 最优解集为 {N1,N4};主要关系被保存并且很多冗余关系被省略.因此,树可以有效地保存解的关系,并且使用更少的比较.这棵树不是传统的二叉树,因此,我们称它为占优树(dominating tree,简称 DT),一棵占优树可以如下定义:

定义 4. 占优树:一棵占优树是定义如下的二叉树:

(1) 一棵占优树要么是一个外部结点或者是一个连接到两棵二叉树的内部结点,这两棵二叉树分别叫作左子树和右子树;

(2) 占优树中的每一个结点有 4 个域:id,count,left-link,right-link,这里:id 表示结点所代表的个体;count 表示左子树的大小(包括自身);left-link 指向左子树,该子树的根结点被该结点占优; right-link 指向右子树,该子树的根结点和该结点无法比较.

在本文中,占优树的兄弟链是指由树根和右链结点构成的链.以图 2 为例,N1 和 N4 是 DT 的兄弟链,根结点是 N4,而且 N3 和 N5 也构成了兄弟链.占优树的定义与二叉查找树(binary search tree,简称 BST)相似.如果我们考虑占优关系对应于 BST 中的小于关系,并且非占优关系对应于 BST 中的大于关系,DT 就成了 BST.但是,BST 中结点关系是二值的:小于和大于关系;但是,DT 中的关系是三值的:0,1,和 2.因此,占优树的构造过程与 BST 是相似的,但却也是不同的.

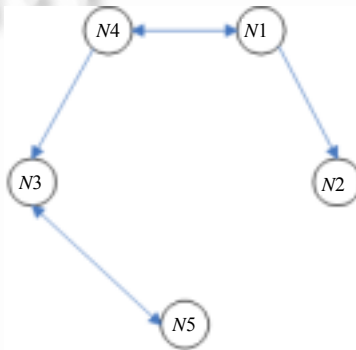


Fig.2 Illustration of the dominating relationships with tree  
图 2 用树表示图 1 中结点关系的示意图

### 2.3 占优树排序算法

因为占优树的创建过程与二叉搜索树的创建过程比较相似,占优树的构造算法也是一种递归算法.与 BST 不同,一个新结点插入到 DT 中有 3 种可能:当新结点被根结点占优时,它被插入到根结点的左子树,这与 BST 中的小于关系相似;当新结点和根结点无法比较时,它被插入到根结点的右子树中,这与 BST 中的大于关系相似;但是,当新结点占优根结点时,根据性质 4,根结点的兄弟链中的其他结点不可能占优新结点.那么,新结点不仅替代根结点的位置,让根结点成为它的左子树,而且应该继续和兄弟链中的其他结点比较.如果有结点被新结点占优,则这些结点应该从兄弟链中删除,然后插入到新结点的左子树中.

如图 3 所示,它显示了图 2 中占优树的创建过程(结点里面左边的数字是结点的 id,右边的数字是它的 count.图按字母排序).结点的输入过程是 N1,N2,N3,N4 和 N5;这些结点插入到树中的结果分别如图 3(a)~图 3(e)所示.5 个结点插入后,占优树如图 3(e)所示.很明显,占优树是不平衡的,因为 N4 的 count 值大于 N1 的 count 值.为了平衡二叉树,N4 和它的左子树沿着兄弟链向左移,如图 3(f)所示.这个移动没有改变结点间的关系,但可以使得占优树更加平衡.图 2 和图 3(f)的结果一样.平衡的占优树有更多优点,这使得增加、删除一个结点所花费的代价更少.平衡过程对淘汰策略也是关键的步骤(见第 3.1 节).占优树的创建过程实际上也是适应值的指派过程,结点间的关系保存在树中.因此,我们也把占优树的构造算法叫做占优树排序算法.

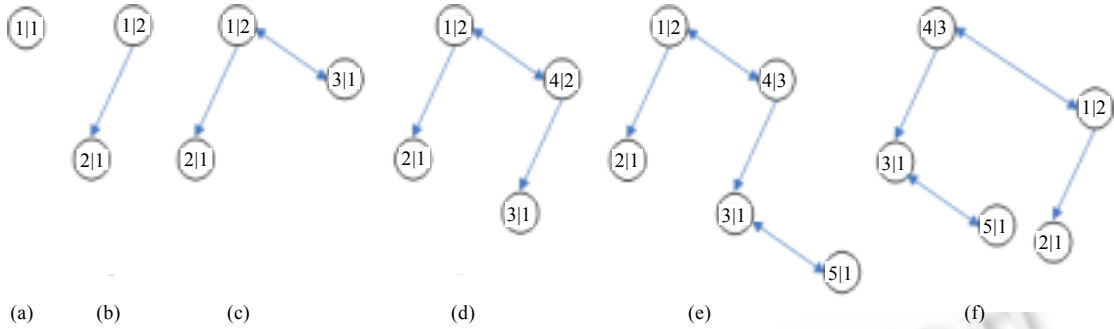


Fig.3 Illustration of the creating process of the tree in Fig.2

图 3 图 2 中树的创建过程

基于上面的分析,我们有占优树构造算法. *ConstructTree* 是创建占优树的主过程. 算法中有 3 个函数: *AddinTree* 和 *AddinSibling* 表示了一个新结点插入到占优树中的操作; 当结点的 count 值改变时, *BalanceTree* 使得该结点连同它的左孩子结点沿着兄弟链向左或者右移动, 这样使得占优树更加平衡. 这一算法充分利用了 Pareto 占优的性质, 减少了个体之间的比较.

算法 1. 占优树的排序算法.

/\* Creating a dominating tree. tree is the root of the whole tree. \*/

Input: all nodes in the population.

Output: the dominating tree.

Link *ConstructTree*() {

For each node (newnode) in the population

*AddinTree*(tree, newnode);

return tree;

}

/\* Add newnode into the left subtree of the tree whose root node is root when newnode is dominated by root. \*/

*AddinTree*(Link root, Link newnode) {

    root->count=root->count+newnode->count;

    if (root->left-link==null)

        then root->left-link=newnode;

    else *AddinSibling*(root, root->left-link, newnode);

}

/\* Newnode is compared with child whose parent node is parent when newnode is inserted into parent's left subtree. \*/

*AddinSibling*(Link parent, Link child, Link newnode) {

    switch (*Better*(newnode, child))

        case 0: //nondominated

            if (child->right-link==null)

                then child->right-link=newnode;

            else *AddinSibling*(parent, child->right-link, newnode);

        case 1: //dominating

            newnode takes the place of child;

*AddinTree*(newnode, child);

            while there exists a pnode (a node in the newnode's sibling chain) and *Better*(newnode, pnode)==1,

            do {

                delete pnode from the sibling chain;

*AddinTree*(newnode, pnode);

            }

}

```

    }
    BalanceTree(parent,newnode,L);
case 2: //dominated
    AddinTree(child,newnode);
    BalanceTree(parent,child,L);
}

/* Sort the sibling chain of the parent's left subtree in their count descending order. If the count of movenode (a
node in the sibling chain) increases, it moves along the sibling chain in the left direction, or else in the right
direction */
BalanceTree(Link parent,Link movenode,int direction){
    if (direction==L)
        while the count of movenode is larger than that of its left nodes which is also in the same sibling chain;
        do movenode with its left subtree moves along the sibling chain in the left direction.
    if (direction==R)
        while the count of movenode is smaller than that of its right nodes which is in the same sibling chain;
        do movenode with its left subtree moves along the sibling chain in the right direction.
}

```

## 2.4 占优树的性质

从占优树的创建过程可以得到占优树的一些重要性质.

**定理 1.** 占优树的根结点的兄弟链只保存该树的所有的 Pareto 最优结点.

**推导:**如果只有一个结点,定理很显然正确.如果对  $n$  个结点定理都成立,根据占优树排序算法,对下一个结点  $pnode$  有 3 种可能:如果  $pnode$  被根结点点占优,则它被插入到根结点的左子树;如果  $pnode$  与根结点无法比较,则它应该和占优树的兄弟链中的其他结点进行比较;如果它被其中一个结点占优,则它应该被插入到第 1 个占优结点的左子树;否则,它成为兄弟链中的一个结点.如果  $pnode$  占优根结点,则根结点应该从兄弟链中删除;对于兄弟链中的其他结点,如果它们被  $pnode$  占优,则它们也应该从兄弟链中删除.因此,构造过程表明,兄弟链只保存所有的最优结点.

**定理 2.** 占优树的根结点占优它左子树的所有结点.

**定理 3.** 占优树根结点的 count 值比其兄弟链中其他结点的 count 值要大.

定理 2 的证明和定理 1 类似;定理 3 由构造算法也很容易得到,由于篇幅原因,证明省略.定理 1 和定理 2 是占优树保存的主要关系;定理 3 保证了占优树是规则的.当结点插入和删除时,规则的占优树有更好的平均性能.

## 3 基于占优树的多目标进化算法

根据占优树的性质,我们可以设计一个简单、高效的多目标优化进化算法.

### 3.1 淘汰策略

DTEA 的淘汰策略是根据占优树设计的,它淘汰最差结点.最差结点定义为有最大 count 值的占优树的左叶结点.根据定理 3,最差结点也就是占优树的最左叶结点.例如,图 3(f)中的最差结点是  $N3$ .这样的淘汰策略是合理的,因为最左叶结点总是被占优树中的一些结点占优.下面给出淘汰策略的算法.我们一次只淘汰一个结点,这样使得算法更加简单、稳定.另一方面,淘汰策略保证了被淘汰的结点总是树中被很多结点占优的结点.

**算法 2.** 淘汰策略的算法.

```

/* Delete the worst node of the dominating tree */
Input: the dominating tree
Output: the worst node
Link DeleteWorstNode(Link root){

```

```

Link p=root->left-link;
root->count--;
if (p->left-link == null)
    root->left-link = p->right-link;
    return p;
else
    return DeleteWorst(root);
}
Link DeleteWorst(Link root){
    Link p=root->left-link;
    Link q=p->left-link;
    p->count--;
    if (q->left-link==null)
        p->left-link=q->right-link;
    else
        q=DeleteWorst(p);
    BalanceTree(root,p,R);
    return q;
}

```

淘汰策略也自然地保存了种群的多样性.一个有较大 count 值的结点意味着目标空间中较多结点被该结点占优,即被它占优的空间里面的结点更拥挤,因此,这些结点应该有更大的概率被淘汰.淘汰有最大 count 值的占优树的最左叶结点,迫使它的占优空间中的结点去其他地方,这使得结点更加均匀地分布在目标空间.这种淘汰策略不仅迫使结点更加均匀地分布,而且指导种群朝最优前沿进化.事实上,占优树在 count 域中隐含了密度信息.结点的 count 值越大,被它占优的空间里面的点越拥挤.我们注意到,当所有个体都无法比较时,被淘汰的结点是兄弟链中的第 1 个结点,这样,淘汰策略不再起到保持种群多样性的作用了.但是事实上,种群的多样性并不差,实验也证实了这一点.

### 3.2 主程序

根据占优树和淘汰策略,我们可以得到 DTEA 算法.

DTEA 算法的主过程(其中: $P$  表示种群; $t$  表示运行代数):

Step 1. Randomly create  $P(0)$  with population size  $N$ . Set counter  $t=0$ .

Step 2. Creating a *dominating tree* using the construction algorithm.

Step 3. **If** stopping criterion is satisfied, **then** stop.

Step 4. Generate an offspring and insert it into the *dominating tree*.

Step 5. Eliminate a worst individual from the *dominating tree*.

Step 6.  $t:=t+1$ , then turn to Step 3.

算法首先随机产生一个初始化大小为  $N$  的种群  $P(0)$ . $t$  是运行代数,首先被设置为 0.这些个体根据占优树排序算法产生一个占优树.在进化阶段,新产生的后代被插入到占优树中,被淘汰的个体从占优树中删除.这个过程一直重复,直到停机条件被满足为止.在该算法中,可以利用各种合适的算子产生新的后代,例如后面实验中所用的 SBX 算子.该算法在实际应用过程中,每一代运行在父代中选择若干个个体(依据算子要求),按照算子操作一次产生若干后代,将它们插入到占优树中,并同时删除相同个数的最差个体.

DTEA 是一个新颖的多目标进化算法,它很少使用其他多目标进化算法中所使用的技术.以往,很多 MOEA 在靠近前沿和保持多样性方面使用了不同的方法,这使得算法很复杂.但是,DTEA 把收敛性策略和多样性策略整合到占优树中,这样不仅保存了解的占优关系,而且自然地保持了种群的多样性.因此,DTEA 比很多 MOEA 都简单.DTEA 是一种稳态的算法.因为一代只产生或淘汰一个解,优秀个体被淘汰的可能性是很小的.一旦发现一



一个好的解,直到它成为种群中最差的解才会被淘汰.进化算法是一种随机算法,一些 MOEA 在进化过程中淘汰好的解<sup>[7]</sup>,这使得算法比较低效.精英策略就是用来解决这个问题<sup>[6]</sup>,但是,这种方法增加了时间和空间的复杂性.DTEA 没有使用额外的代价就实现了精英策略.

在其他常用 MOEA(如 SPEA2 和 NSGA-II)的适应值指派方法中,要评价一个个体的适应值,需要与种群中的其他所有个体进行比较,因此,评价一个个体的比较次数为  $N-1$ ( $N$  为种群规模).在 DTEA 中,评价一个个体只需要将其插入到树结构中,在一般情况下,比较次数是小于  $N$  的.而且在多样性策略(对 DTEA 而言就是淘汰策略)中,淘汰一个最差个体的时间复杂度也是小于  $N$  的,而很多 MOEA 中的多样性策略都是  $O(N)$ (对一个个体而言)<sup>[8]</sup>.因此,DTEA 要比很多 MOEA 都要快,实验也验证了这一点.DTEA 中使用了占优树结构,它的空间复杂度为  $O(N)$ .很多 MOEA 都使用了精英存档器(elite archive),其空间复杂度也是  $O(N)$ .因此,DTEA 并没有使用更多的存储空间.

与 DTEA 相似,PAES 也是每代产生和删除一个结点.PAES 是(1+1)的进化策略,它使用了局部搜索的算子以及精英种群,保存已经发现的比较好的解<sup>[3]</sup>.PAES 只使用了变异算子,但是,包括杂交变异算子在内的很多算子都可以在 DTEA 中使用.为了保持种群多样性,PAES 中使用了超网格的方法,但是,DTEA 中没有专门的保持多样性的策略.MOEA 中也使用了其他一些很有趣的树结构.Mostaghim 利用了 Quad-tree 这种有效的数据结构保存最优个体<sup>[16]</sup>.当精英种群大小不受限制时,Fieldsend 设计了 dominated/nondominated tree 用于加速个体的查找和排序<sup>[11]</sup>.占优树与它们相比是完全不同的:一方面,占优树与其他方法有完全不同的数据结构;另一方面,它们解决的问题也不同.占优树是一种新的适应值指派方法,它存储所有解的占优信息.但是,dominated/nondominated tree 和 Quad-tree 只用于精英种群中的个体的存储和排序.

#### 4 实验与讨论

为了检验所提出的 DTEA,我们比较 DTEA 与其他两个著名的 MOEAs-SPEA2 和 NSGA-II 的性能.其他两种算法根据文献[1,2]按照 C 语言来实现.实验运行在 3GHz 512M 内存运行 Windows 2000 的 Pentium IV 机器上.

最初的 DTEA 已经利用一些常用的测试函数在靠近最优前沿方面和一些 MOEA 作比较<sup>[17]</sup>.实验表明:该算法能够接近最优前沿,并且对大部分问题都保持了种群的多样性.在本文中,我们增加 3 个目标的测试函数,并且 3 种算法将从 3 个方面进行比较:靠近最优前沿、保持种群多样性和运行时间.这些函数选自本领域的一些重要研究,这些函数的信息见表 2.在实验中,每个函数计算 30 次,每次计算结束后,保存当前种群中的最优解,合并 30 次计算的最优解,再求出合并后的解集中的最优解,将其作为最终结果.这样做的目的在于减少算法的随机性的影响,使其结果比较更加客观.不同算法的最后结果按照标准进行比较.运行时间是平均运行时间.

对 SPEA2 和 NSGA-II,我们按照文献中建议的参数设置;对 DTEA,没有特别的参数需要设置.为了公平比较这 3 个算法,它们都使用了模拟二进制杂交(simulated binary crossover,简称 SBX)和多项式变异算子(polynomial mutation)<sup>[18]</sup>.种群大小是 100;精英种群大小是 100.在 QV 和 KUR 中,算子的参数设置为  $\eta_c=20$  和  $\eta_m=20$ ;对其他函数设置为  $\eta_c=15$  和  $\eta_m=20$ .在 QV 和 KUR 中,杂交概率为 0.9,在 DTLZ1~DTLZ4,该值为 1.对所有的测试函数,变异概率为  $1/n$ , $n$  为决策变量的个数.被评价个体的数目设置如下:QV 和 KUR 为 15 000;DTLZ1 和 DTLZ2 为 30 000;DTLZ3 为 50 000;DTLZ4 为 20 000.SPEA2 和 NSGA-II 的运行代数数为评价个体的数目除以种群规模.因为 SBX 算子一次产生和删除两个个体,所以,DTEA 的运行代数数为评价个体的数目除以 2.

本文使用了两个常用的评价标准.我们利用  $C(X,Y)$  来比较两个最优解集靠近前沿的程度<sup>[10]</sup>. $C(X,Y)=1$  表示  $Y$  中所有解都被  $X$  中的解占优.相反地, $C(X,Y)=0$  表示  $Y$  中没有解被  $X$  中的解占优.注意到, $C(X,Y)$  和  $C(Y,X)$  都需要被考虑,因为  $C(Y,X)$  并不一定等于  $1-C(X,Y)$ . $\Delta$  表示了解集的分散程度<sup>[20]</sup>,一个好的分散性将会使得  $\Delta$  接近 0.这两个评价标准是独立于最优前沿的,它们在一定程度上反映了解的质量,并且也可以应用在高维目标空间中.

**Table 2** Test problems used in the study  
**表 2** 实验测试函数

| Problem               | $n$ | Domain     | Objective functions  | Comments                                  |
|-----------------------|-----|------------|--|---|
| QV <sup>[2]</sup>     | 100 | $[-5,5]^n$ | $f_1(\mathbf{x}) = \left( \frac{1}{n} \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \right)^{0.25}$ $f_2(\mathbf{x}) = \left( \frac{1}{n} \sum_{i=1}^n ((x_i - 1.5)^2 - 10 \cos(2\pi(x_i - 1.5)) + 10) \right)^{0.25}$   | 2 objectives<br>nonconvex                 |
| KUR <sup>[1,2]</sup>  | 3   | $[-5,5]^n$ | $f_1(\mathbf{x}) = \sum_{i=1}^{n-1} (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(\mathbf{x}) = \sum_{i=1}^n ( x_i ^{0.8} + 5 \sin x_i^3)$   | 2 objectives<br>nonconvex<br>disconnected |
| DTLZ1 <sup>[19]</sup> | 7   | $[0,1]^n$  | $f_1(\mathbf{x}) = \frac{1}{2} x_1 x_2 (1 + g(\mathbf{x}_M))$ $f_2(\mathbf{x}) = \frac{1}{2} x_1 (1 - x_2) (1 + g(\mathbf{x}_M))$ $f_3(\mathbf{x}) = \frac{1}{2} (1 - x_1) (1 + g(\mathbf{x}_M))$ $g(\mathbf{x}_M) = 100( \mathbf{x}_M  + \sum_{x_j \in \mathbf{x}_M} ((x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))))$                       | 3 objectives<br>hyper-plane               |
| DTLZ2 <sup>[19]</sup> | 12  | $[0,1]^n$  | $f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1 \pi / 2) \cos(x_2 \pi / 2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1 \pi / 2) \sin(x_2 \pi / 2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1 \pi / 2)$ $g(\mathbf{x}_M) = \sum_{x_j \in \mathbf{x}_M} (x_j - 0.5)^2$  | 3 objectives<br>spherical                 |
| DTLZ3 <sup>[19]</sup> | 7   | $[0,1]^n$  | $f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1 \pi / 2) \cos(x_2 \pi / 2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1 \pi / 2) \sin(x_2 \pi / 2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1 \pi / 2)$ $g(\mathbf{x}_M) = 100( \mathbf{x}_M  + \sum_{x_j \in \mathbf{x}_M} ((x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))))$ | 3 objectives<br>spherical                 |
| DTLZ4 <sup>[19]</sup> | 12  | $[0,1]^n$  | $f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^{100} \pi / 2) \cos(x_2^{100} \pi / 2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^{100} \pi / 2) \sin(x_2^{100} \pi / 2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1^{100} \pi / 2)$ $g(\mathbf{x}_M) = \sum_{x_j \in \mathbf{x}_M} (x_j - 0.5)^2$                    | 3 objectives<br>spherical                 |

表 3 显示了 3 种算法利用趋近前沿的评价标准 C 得到的比较结果(D 表示 DTEA;S 表示 SPEA2;N 表示 NSGA-II).对测试函数 QV 和 KUR,DTEA 很明显地更靠近前沿.但是,对 DTLZ1 函数,DTEA 的结果较差;对其他函数,它们的性能相当.很明显:对这些测试函数而言,在接近最优前沿的性能上,DTEA 相比其他算法是很有竞争力的.

**Table 3** Convergence comparison of different algorithms using C  
**表 3** 不同算法利用 C 比较趋近前沿的结果

|        | QV | KUR   | DTLZ1 | DTLZ2 | DTLZ3 | DTLZ4 |
|--------|----|-------|-------|-------|-------|-------|
| C(D,S) | 1  | 0.592 | 0.016 | 0.089 | 0.019 | 0.158 |
| C(D,N) | 1  | 0.553 | 0.006 | 0.118 | 0.019 | 0.078 |
| C(S,D) | 0  | 0.004 | 0.708 | 0.101 | 0.060 | 0.181 |
| C(N,D) | 0  | 0.004 | 0.572 | 0.081 | 0.056 | 0.192 |

表 4 显示的是 3 种算法利用分散性评价标准 A 得到的比较结果.在保持多样性方面,除了 KUR,DTEA 并不比其他两种算法差.实验结果显示:DTEA 中的淘汰策略是有效的,它保持了种群的多样性.第 3.1 节中已经提到:当所有的结点无法比较时(此时,占优树演变成一个只有右链的队列),DTEA 中的淘汰策略不再起到保持种群多样性的作用.但是,DTEA 算法与其他算法一样较好地保持了种群多样性.总的来说,一方面,新产生的结点比原先的结点要好,而新产生的非占优结点一般排序到占优树的兄弟链的末端,因此,被删除的结点总是比较差的结点;另一方面,当新产生的结点占优于兄弟链中的一个结点时,这个队列又变成了一棵占优树.因此,淘汰策略在大部分时间都是有效的.

**Table 4** Distribution comparison of different algorithms using  $\Delta$ **表 4** 不同算法利用 $\Delta$ 得到的多样性比较结果

|         | QV    | KUR   | DTLZ1 | DTLZ2 | DTLZ3 | DTLZ4 |
|---------|-------|-------|-------|-------|-------|-------|
| DTEA    | 0.003 | 0.041 | 0.137 | 0.235 | 0.249 | 0.287 |
| SPEA2   | 0.004 | 0.014 | 0.116 | 0.246 | 0.268 | 0.253 |
| NSGA-II | 0.004 | 0.015 | 0.130 | 0.233 | 0.253 | 0.284 |

表 5 显示了这 3 种算法的运行时间。DTEA 的运行时间很明显地小于其他两种算法,因为 NSGA-II 中使用了有效的密度估计算法和快速非占优排序算法<sup>[1]</sup>,NSGA-II 很明显快于 SPEA2。实验结果显示,占优树排序确实加快了适应值指派过程。

**Table 5** Comparison of different algorithms on running time (ms)**表 5** 不同算法的运行时间 (毫秒)

|         | QV    | KUR   | DTLZ1  | DTLZ2  | DTLZ3  | DTLZ4 |
|---------|-------|-------|--------|--------|--------|-------|
| DTEA    | 955   | 236   | 619    | 1083   | 1105   | 720   |
| SPEA2   | 4 237 | 3 095 | 10 333 | 10 630 | 17 611 | 8 339 |
| NSGA-II | 1 642 | 772   | 2 039  | 2 553  | 3 502  | 1 753 |

通过上面的分析和比较,我们发现:在趋近最优前沿和保持解的分散性方面,DTEA 与 SPEA2 和 NSGA-II 相比是很有竞争力的;特别是在运行时间方面,它比另外两种算法要快得多。总之,占优树是一种高效的适应值指派方法,而且,基于占优树的淘汰策略也有效地保存了种群的多样性。

## 5 总 结

用进化计算解决多目标优化问题是被广泛研究的有效方法,但是,目前很多算法都比较复杂,时间复杂度较高。针对这种情况,本文提出一种新的多目标进化算法——DTEA。DTEA 可以被概括为:(1) 使用了 Better 函数(一个 3 值函数)比较解的好坏;(2) 提出了占优树这种数据结构,用于保存解之间的占优信息,并且暗含了密度信息。这种方法完全利用了 Pareto 占优的性质,并且显著减少了个体之间的比较;(3) 基于占优树的淘汰策略,可以不需要额外的代价就保持种群多样性。DTEA 将收敛性和多样性策略整合到占优树中,因此,DTEA 比很多算法都简单。因为算法的稳态特征,DTEA 不需要任何额外的代价就实现了精英策略。通过比较实验,我们发现:在 6 个测试函数上,DTEA 取得了与 NSGA-II 和 SPEA2 相比性能相当的解,但是,DTEA 比它们要快得多。

## References:

- [1] Deb K, Pratab A, Agarwal S, MeyArivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002,6(2):182-197.
- [2] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report 103*, 2001.
- [3] Knowles JD, Corne DW. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multi-objective optimization. In: *Proc. of the 2003 IEEE Conf. on Evolutionary Computation*. 1999. 98-105. <http://ieeexplore.ieee.org/iel5/6342/16952/00781913.pdf?isnumber=16952&arnumber=781913>
- [4] Atashkari K, Nariman-Zadeh N, Pilechi A, Pilechi A, Yao X. Thermodynamic Pareto optimization of turbojet engines using multi-objective genetic algorithm. *Int'l Journal of Thermal Sciences*, 2005,AID.2338:1-11.
- [5] Yen G, Lu H. Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation. *IEEE Trans. on Evolutionary Computation*, 2003,7(3):253-274.
- [6] Zitzler E, Thiele L. Multi-Objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. on Evolutionary Computation*, 1999,3(4):257-271.
- [7] Veldhuizen DAV, Lamont GB. Multi-Objective evolutionary algorithms: Analyzing the state-of-the-art. *IEEE Trans. on Evolutionary Computation*, 2000,18(2): 125-147.
- [8] Fonseca CM, Fleming PJ. Multi-Objective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation. *IEEE Trans. on Systems, Man and Cybernetics—Part A: Systems and Humans*, 1998,28(1):26-37.

- [9] Xin Y, Xu Y. Recent advances in evolutionary computation. *Journal of Computer Sciences and Technology*, 2006,21(1):1-18.
- [10] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results. *IEEE Trans. on Evolutionary Computation*, 2000,18(2):173-195.
- [11] Fieldsend JE, Everson RM, Singh S. Using unconstrained elite archives for multi-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2003,7(3):305-323.
- [12] Cui XX, Lin C. A preference-based multi-objective concordance genetic algorithm. *Journal of Software*, 2005,16(5):761-770 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/761.htm>
- [13] Lei DM, Wu ZM. Crowding-Measure based multi-objective evolutionary algorithm. *Chinese Journal of Computers*, 2005,28(8): 1320-1326 (in Chinese with English abstract).
- [14] Zeng SY, Wei W, Kang LS, Yao SZ. A multi-objective evolutionary algorithm based on orthogonal design. *Chinese Journal of Computers*, 2005,28(7):1153-1162 (in Chinese with English abstract).
- [15] Jensen MT. Reducing the run-time complexity of multi-objective EAs: The NSGA-II and other algorithms. *IEEE Trans. on Evolutionary Computation*, 2003,7(5):503-515.
- [16] Mostaghim S, Teich J, Tyagi A. Comparison of data structures for storing Pareto-sets in MOEAs. In: *Proc. of the World Congress on Computational Intelligence*. 2002. 843-849. <http://ieeexplore.ieee.org/iel5/7875/21693/01007035.pdf?isnumber=&number=1007035>
- [17] Shi C, Li Y, Kang LS. A new simple and highly efficient multi-objective optimal evolutionary algorithm. In: *Proc. of the 2003 IEEE Conf. on Evolutionary Computation*. 2003. 1536-1542. <http://ieeexplore.ieee.org/iel5/9096/28878/01299855.pdf?isnumber=&number=1299855>
- [18] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. *Complex Systems*, 1994,1(9):115-148.
- [19] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: *Proc. of the 2002 Congress on Evolutionary Computation*. 2002. 825-830. <http://ieeexplore.ieee.org/iel5/7875/21693/01007032.pdf?isnumber=&number=1007032>
- [20] Deb K, Samir A, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Technical Report.

#### 附中文参考文献:

- [12] 崔逊学,林闯.一种基于偏好的多目标调和遗传算法. *软件学报*,2005,16(5):761-770. <http://www.jos.org.cn/1000-9825/16/761.htm>
- [13] 雷德明,吴智铭.基于个体密度距离的多目标进化算法. *计算机学报*,2005,28(8):1320-1326.
- [14] 曾三友,魏巍,康立山,姚书振.基于正交设计的多目标演化算法. *计算机学报*,2005,28(7):1153-1162.



石川(1978 - ),男,湖北洪湖人,博士生,主要研究领域为遗传算法,机器学习.



史忠植(1941 - ),男,研究员,博士生导师,CCF高级会员,主要研究领域为人工智能,机器学习,神经计算,认知科学.



李清勇(1979 - ),男,博士生,主要研究领域为机器学习,图像理解,视觉信息挖掘.