

# 基于 $\epsilon$ -支配的多目标进化算法及 自适应 $\epsilon$ 调整策略

刘 鎏<sup>1)</sup> 李敏强<sup>1)</sup> 林 丹<sup>2)</sup>

<sup>1)</sup>(天津大学系统工程研究所 天津 300072)

<sup>2)</sup>(天津大学理学院应用数学系 天津 300072)

**摘 要** 提出了一类新的基于  $\epsilon$ -支配关系的多目标进化算法. 该算法采用配对比较选择和稳态替换策略, 提高了算法的收敛速度, 降低了计算时间. 首先, 在保持种群分布性上, 采用了一种新的基于  $\epsilon$ -支配关系的精英保留策略, 避免了传统修剪策略所引起的 Pareto 前沿面的退化. 其次, 根据不同  $\epsilon$  取值分析了算法收敛性, 提出了一种自适应  $\epsilon$  调整策略. 最后, 通过 5 个常用的双目标测试函数的计算, 验证了包括该自适应调整策略的多目标进化算法在求解质量上显著强于 NSGAII, SPEA2 和  $\epsilon$ -MOEA 等主流多目标进化算法.

**关键词** 多目标优化;  $\epsilon$ -支配; 进化算法;  $\epsilon$  自适应调整; 精英保留策略; 稳态策略

**中图法分类号** TP18

## The $\epsilon$ -Dominance Based Multi-Objective Evolutionary Algorithm and an Adaptive $\epsilon$ Strategy

LIU Liu<sup>1)</sup> LI Min-Qiang<sup>1)</sup> LIN Dan<sup>2)</sup>

<sup>1)</sup>(Institute of Systems Engineering, Tianjin University, Tianjin 300072)

<sup>2)</sup>(Department of Applied Mathematics, School of Science, Tianjin University, Tianjin 300072)

**Abstract** A novel multi-objective evolutionary algorithm, called  $\epsilon$ -dominance multi-objective evolutionary algorithm(EDMOEA), is proposed in this paper. In the EDMOEA, pair-comparison selective and steady-state replacement are used to replace the conventional Pareto-ranking strategy, which could effectively improve the convergence rate of the algorithm and reduce the computation time. The main component of the new algorithm is the truncating method in archive population. Based on  $\epsilon$ -dominance relationship, it maintains the diversity of the population and prevents the degradation of the Pareto front which often occurs in the conventional truncating strategies. Future more, a new adaptive  $\epsilon$  setting method is incorporated into EDMOEA. Finally, five binary-objective functions are used to test the performance of the EDMOEA, the Adaptive-EDMOEA(AEDMOEA), and conventional algorithms such as NSGAII, SPEA2, and  $\epsilon$ -MOEA. Experimental results demonstrate that the AEDMOEA and EDMOEA outperform other algorithms on these test functions.

**Keywords** multi-objective optimization;  $\epsilon$ -dominance; evolutionary algorithm;  $\epsilon$ -adaptive; elitism strategy; steady-state strategy

# 1 引言

求解最优化问题(也称数学规划问题)是指从所有可能的方案中选择最合理的一种以达到目标优化的过程. 当优化问题的目标个数多于一个时,称之为多目标优化. 在通常情况下,同一问题中的多个目标函数是彼此矛盾的,因此最终结果是获得一系列折衷解.

多目标进化算法是指利用进化搜索的技术去解决多目标优化问题. Schaffer<sup>[1]</sup>提出了第一个多目标进化算法即向量进化遗传算法,而后该领域专家又提出了多种多目标进化算法并应用于求解实际问题. Coello<sup>[2]</sup>总结了目前的多目标进化算法,并将它们分为两代:第一代强调简洁,第二代强调效率,它们之间的主要区别在于精英个体是否被引入种群的进化过程之中. Laumanns<sup>[3]</sup>归纳了采用精英策略的多目标进化统一模型(Unified model for MOEAs, UMMEA),通过将存储当前所有非被支配个体的种群同一般的进化种群相结合,实现精英参与种群的进化. 大部分第二代的多目标进化算法,如强度 Pareto 进化算法(SPEA)<sup>[4]</sup>、强度 Pareto 进化算法 2(SPEA2)<sup>[5]</sup>、Pareto 包络选择算法(PAES)<sup>[6]</sup>都符合这样的模型结构. 另一个常用的非劣排序遗传算法 2(NSGAI)没有直接利用外部种群. 它将子代种群和父代种群相结合,优先选择其中的精英个体去构成下一代的进化种群. 这种策略也实现了精英个体加入种群进化,并取得了很好的计算结果.

另一个分类标准即是否采用了 Pareto 支配排序. Goldberg<sup>[7]</sup>率先将 Pareto 优化的概念引入多目标进化算法. 当前的多目标进化算法大多通过 Pareto 支配关系的排序来计算种群中个体的适应值,从而引导种群朝向 Pareto 前沿面进化. 虽然这种方法可以较好地改善算法的收敛性,但是排序过程要耗费大量的计算.

为了提高进化算法效率,一些研究者采用了稳态的进化策略. 所谓稳态是当新个体产生后立即加入到种群的下一代进化过程之中,如简单进化算法(SEAMO)<sup>[8]</sup>、Pareto 收敛遗传算法(PCGA)<sup>[9]</sup>、 $\epsilon$ -多目标进化算法( $\epsilon$ -MOEA)<sup>[10]</sup>. 在选择个体进入交配池的操作中,它们均采用配对比较的方法,而没有进行个体适应值的计算. 实验结果表明这些基于稳态的进化算法在处理某些问题上要优于基于 Pareto 排序的算法<sup>[8,11]</sup>.

另外,由于非支配解的数量巨大,而外部种群存储容量有限,很多修剪策略,如 PAES 中的自适应网格、NSGAI 中的 Crowding-Distance、SPEA 中的聚类 and SPEA2 中的最近距离方法等都在各自算法中发挥了很好的作用. 然而,正如文献[12]所述,这些修剪策略很可能造成 Pareto 前沿的退化,进而影响到最终种群的收敛. Laumanns<sup>[13]</sup>根据  $\epsilon$ -支配关系,提出一种基于网格向量的种群修剪策略,可以很好地防止进化过程中种群的退化. 这种策略在计算网格向量时,除了  $\epsilon$  参数,还需要获得各个目标上的最小值,并且相同网格中的个体比较需要计算各自的欧式距离,相对来说计算量比较大.

本文提出了一种新的基于  $\epsilon$ -支配关系的多目标进化算法,即  $\epsilon$ -支配多目标进化算法( $\epsilon$ -Dominance MOEA, EDMOEA). 该算法采用一种新的基于  $\epsilon$ -支配关系的修剪策略,不仅可以防止退化现象,还可以有效地保存极端值个体以保证 Pareto 前沿面的广度. 该算法基于稳态替换策略,利用  $\mu+2$  的选择方法,可以更加快速有效地到达 Pareto 前沿面. 同时,在新算法中采用了新的自适应  $\epsilon$  调整策略,实验结果验证了这种新策略的优越性.

本文第 2 节介绍 Pareto 优化和  $\epsilon$ -Pareto 优化的概念;第 3 节简要引入协同 UMMEA 模型,并对其修正;第 4 节详细讨论新的 EDMOEA 算法和自适应  $\epsilon$  调整策略;第 5 节针对在一系列测试函数上的计算实验,将包含自适应调整策略的自适应多目标进化算法(AEDMOEA)同固定  $\epsilon$  策略的 EDMOEA 及 NSGAI, SPEA2,  $\epsilon$ -MOEA 等其它算法进行对比,分析新算法的性能优势;第 6 节总结本文研究结论.

## 2 基本概念

### 2.1 Pareto 支配关系

在多目标优化问题中, Pareto 优化解是最常用的优化概念. 它最早由 Francis Ysidro Edgeworth 在 1881 年提出而后经 Vilfredo Pareto 推广<sup>[14]</sup>(为方便讨论起见,本文的优化问题皆为最小化问题),其定义如下.

**定义 1**(Pareto 支配). 设  $f: R^m \rightarrow R^k$ ,  $x_1, x_2 \in \Omega \subseteq R^m$ , 称解  $x_1$  支配解  $x_2$  当且仅当  $f(x_1)$  部分地优于  $f(x_2)$ , 即  $\forall i \in \{1, 2, \dots, k\}, f_i(x_1) \leq f_i(x_2) \wedge \exists j \in \{1, 2, \dots, k\}, f_j(x_1) < f_j(x_2)$ , 记作  $x_1 \leq x_2$ .

**定义 2**(Pareto 最优解). 解  $x^* \in \Omega$  称之为解

集合  $\Omega$  的 Pareto 优化解当且仅当集合  $\{x | x \leq x^*, x \in \Omega\} = \emptyset$ .

**定义 3**(Pareto 最优解集). 对于给定的多目标优化问题, 设其定义域为  $\Omega$ , 则其 Pareto 最优集  $X^*$ , 定义为:  $X^* = \{x \in \Omega | \neg \exists x' \in \Omega, f(x') \leq f(x)\}$ .

将 Pareto 最优解集在函数空间上对应的集合称之为 Pareto 前沿, 记为  $PF^*$ .

## 2.2 $\epsilon$ -支配关系

$\epsilon$ -支配关系是传统的 Pareto 支配关系的弱化<sup>[15]</sup>, 其具有多种概念形式. 这里我们采用加  $\epsilon$  的形式, 且对于给定的  $\epsilon \in R^k, \epsilon_i > 0, \forall i \in k$ . 其定义如下:

**定义 4**( $\epsilon$ -支配). 设  $f: R^m \rightarrow R^k, x_1, x_2 \in \Omega \subseteq R^m$ , 对  $\epsilon \in R^k, \epsilon_i > 0$ , 称  $x_1 \epsilon$ -支配  $x_2$  当且仅当  $\forall i \in \{1, 2, \dots, k\}, f_i(x_1) - \epsilon_i \leq f_i(x_2)$ , 且  $\exists i, f_i(x_1) - \epsilon_i < f_i(x_2)$ . 记为  $x_1 \leq_\epsilon x_2$ .

**定义 5**( $\epsilon$ -Pareto 最优解集近似)<sup>[13]</sup>. 集合  $X^a$  称为  $X$  的  $\epsilon$ -Pareto 最优解集近似, 当且仅当对任意  $x \in X$  都存在  $\exists x' \in X^a$ , 使得  $x' \leq_\epsilon x$ .

**定义 6**( $\epsilon$ -Pareto 解集)<sup>[13]</sup>. 集合  $X^\epsilon$  称为集合  $X$  的  $\epsilon$ -Pareto 解集, 当且仅当  $X^\epsilon$  为集合  $X$  的  $\epsilon$ -Pareto 最优解集近似, 且  $X^\epsilon \subseteq X^*$ .

根据以上定义, 可以得到如下两个定理.

**定理 1.**  $x_1, x_2 \in X, x_1 \leq x_2 \Rightarrow x_1 <_\epsilon x_2$ .

证明. 因为  $x_1 \leq x_2$ , 故对  $\forall i$  有  $f_i(x_1) \leq f_i(x_2)$ , 且  $\exists i$  使得  $f_i(x_1) < f_i(x_2)$ . 又由  $\epsilon_i > 0$ , 则对  $\forall i, f_i(x_1) - \epsilon_i < f_i(x_2)$ . 因此可知  $x_1 <_\epsilon x_2$ . 定理得证.

证毕.

**定理 2.** 设  $x_1, x_2, x_3 \in X$ , 则有  $x_1 \leq x_2, x_2 <_\epsilon x_3 \Rightarrow x_1 <_\epsilon x_3$ .

证明. 由题设可知  $x_1 \leq x_2$ , 则有  $\forall i, f_i(x_1) \leq f_i(x_2)$ , 且  $\exists i$  使得  $f_i(x_1) < f_i(x_2)$ . 又由  $\epsilon_i > 0$ , 故可知  $\forall i, f_i(x_1) - \epsilon_i < f_i(x_2) - \epsilon_i$ . 另一方面, 由  $x_2 <_\epsilon x_3$ , 据定义可知  $\forall i, f_i(x_2) - \epsilon_i \leq f_i(x_3)$ ,  $\exists i, f_i(x_2) - \epsilon_i < f_i(x_3)$ . 显然可知, 对  $\forall i$ , 有  $f_i(x_1) - \epsilon_i < f_i(x_3)$ , 即  $x_1 <_\epsilon x_3$ . 定理得证.

证毕.

值得注意的是,  $X^\epsilon$  和  $X^a$  分别作为集合  $X$  的  $\epsilon$ -Pareto 解集和  $\epsilon$ -Pareto 最优解集近似, 都不是唯一的. 一些个体靠近 Pareto 前沿, 但不是 Pareto 最优解, 若满足  $\epsilon$ -支配关系, 都有可能包含在  $X^a$  中. 定义 6 可以保证  $X^\epsilon$  中的个体皆为 Pareto 最优解.

## 3 多目标进化算法的进化模型

Laumanns<sup>[3]</sup> 给出了采用精英策略的多目标进

化算法的一般模型结构 UMMEA, 协同进化个体种群和精英种群. 精英策略是指优良的个体不会因为劣个体的影响而被排除出种群的基因池. 其算法流程如下: 首先, 采用初始化算子生成进化种群和精英种群, 并选择参数精英强度  $p_e$ . 然后, 由当前的进化种群和精英种群生成子代种群. 其中, 精英强度  $p_e$  控制精英个体的参与程度, 即从精英种群中选择个体作为交配个体的概率. 取样过程由取样算子 (sample) 控制, 变化算子 (vary) 则通过交叉和变异操作而产生子代个体. 虽然 PAES, SPEA, SPEA2 皆可抽象为这样的构架, 然而对于 NSGAII 和  $\epsilon$ -MOEA 来说, 却需要做一些必要的修正, 如算法 1 所示.

**算法 1.** 修正的具有精英策略的多目标进化算法.  $A'$  表示精英种群,  $B'$  表示进化种群,  $p'_e$  表示精英强度,  $S'$  表示子代种群.

1.  $t := 0$
2.  $(A^0, B^0, S^0, p_e^0) := \text{initialize}()$
3. while  $\text{terminate}(A', B', t) = \text{false}$  do
4.  $t := t + 1$
5.  $S' := \text{vary}(\text{sample}(\text{evaluate}(A'^{-1}, B'^{-1}, p_e'^{-1})))$
6.  $A' := \text{truncate}(\text{update}_1(A'^{-1}, S'))$
7.  $p'_e := \text{adapt}(A'^{-1}, B'^{-1}, p_e'^{-1})$
8.  $B' := \text{update}_2(B'^{-1}, S')$
9. end while

算法 1 在进化过程中强调了子代种群同父代种群之间的竞争作用. 这种竞争不仅在评估过程中而且也分别在两个种群的更新算子 ( $\text{update}_1$  和  $\text{update}_2$ ) 中. 显然,  $\epsilon$ -MOEA 可以看作算法 1 的一个实例, NSGAII 即是精英种群设置为零的特殊情形. 当直接把子代种群看作下一代进化种群时, 算法 1 同 Laumanns 所提的 UMMEA 是等价的<sup>[3]</sup>. 由于进化种群的更新过程同精英种群是相对独立的, 在进化过程中或许有部分的精英个体进入进化种群. 因此, 精英强度重新定义为从两个协同种群中选择的交配个体都为精英的概率.

## 4 EDMOEA 算法

在本节中, 我们提出了一类新的基于  $\epsilon$ -支配关系的多目标进化算法. 同  $\epsilon$ -MOEA 一样, 它也可以分解为算法 1 的实例. 这两个算法的主要区别即在于精英种群的更新策略上. 新算法的更新策略结构简单, 且具有较少的参数设置.

4.1 精英种群的更新策略

首先, 我们给出对于给定的集合  $X$  逼近其 Pareto 最优解集的迭代搜索算法的基本框架<sup>[13]</sup>, 如算法 2 所示.

**算法 2.** 迭代的搜索算法.

```
1.  $t := 0$ 
2.  $X^{(0)} := \varnothing$ 
3. while terminate ( $X^{(t)}, t$ ) = false do
4.    $t := t + 1$ 
5.    $x^{(t)} := generate()$ 
6.    $X^{(t)} := update(X^{(t-1)}, x^{(t)})$ 
7. end while
8. Output:  $X^{(t)}$ 
```

其中,  $t$  表示迭代次数,  $x^{(t)}$  表示在第  $t$  次循环时所产生的新的个体, 而集合  $X^{(t)}$  表示  $t$  时的精英种群. 算子 *generate*() 是指在第  $t$  次迭代时产生新的子代个体; *update*() 表示结合新的子代个体对原有的精英种群  $X^{(t-1)}$  做更新操作, 其流程如算法 3 所示.

**算法 3.** 生成 Pareto 最优解集的更新算法.

```
1. Input:  $X, x$ 
2. if  $\exists x' \in X$  such that  $x' \leq x$  then
3.    $X' := X$ 
4. else
5.    $D := \{x' \in X | x \leq x'\}$ 
6.    $X' = X \cup \{x\} \setminus D$ 
7. end if
8. Output:  $X'$ 
```

显然, 如果我们采用算法 3 的更新策略, 通过算法 2 所得到的最终种群将是 Pareto 最优解集的子集合. 然而若仅仅将算法 3 第 2 行的 Pareto 支配关系改为  $\epsilon$ -Pareto 支配关系, 最后所输出的解集仅仅是  $\epsilon$ -Pareto 最优解集近似<sup>[13]</sup>. 故一类新的用以生成  $\epsilon$ -Pareto 解集的算法构成如下.

**算法 4.**  $\epsilon$ -Pareto 解集的更新算法.

```
1. Input:  $X, x$ 
2. if  $\exists x' \in X$  such that  $x' \leq x$  then
3.    $X' := X$ 
4.   Output:  $X'$ ; //算法终止
5. else
6.    $D := \{x' \in X | x \leq x'\}$ 
7.    $X' = X \setminus D$ 
8. end if
9. if  $\exists x'' \in X'$  such that  $x'' \prec_{\epsilon} x$  then
10.   $X'' := X'$ 
11. else
12.   $X'' := X' \cup \{x\}$ 
```

```
13. end if
14. Output:  $X''$  //算法终止
```

**定理 3.** 设  $x^{(j)}$  是第  $j$  代生成的个体,  $X(t) = \bigcup_{j=1}^t x^{(j)}$  为到第  $t$  代为止所生成的个体的集合. 而  $X^{(t)}$  指根据算法 4 的更新策略在第  $t$  代时所输出的最终种群, 则  $X^{(t)}$  是  $X(t)$  的一个  $\epsilon$ -Pareto 解集.

证明. 首先, 可证  $X^{(t)} \subseteq X(t)^*$ . 令任意  $x \in X^{(t)}$ ,  $x = x^{(\tau)}$ ,  $\tau \leq t$ . 假设存在  $x' = x^{(\tau')}$ ,  $x' \prec x$ ,  $\tau' \neq \tau$ . 如果  $\tau' < \tau$ , 则  $x$  在  $t = \tau$  时不会被接收(算法 4, 第 3 行); 如果存在  $x'' = x^{(\tau')}$ ,  $\tau' < \tau'' < \tau$ , 使得  $x'' \prec x'$ , 则根据 Pareto 支配关系的传递性可知  $x'' \prec x$ , 故  $x$  依然被排除在最终种群之外. 如果  $\tau' > \tau$ , 则  $x$  将会从  $X(\tau')$  中被移除(算法 4, 第 7 行). 这皆与  $x \in X^{(t)}$  矛盾. 故假设不成立, 即  $X^{(t)}$  中的任一个体均为非被支配解.

其次, 可证  $X^{(t)}$  是集合  $X(t)$  的  $\epsilon$ -Pareto 最优解集近似. 假设其不正确, 则当且仅当存在  $x = x^{(\tau)}$ ,  $\tau \leq t$ , 其不被  $X^{(t)}$  的任何成员  $\epsilon$ -Pareto 支配, 但却不属于集合  $X^{(t)}$ . 倘若  $x$  不属于集合  $X^{(t)}$ , 则其或者在  $t = \tau$  时不被集合  $X^{(\tau-1)}$  接收, 或者在  $t = \tau$  时被接收但在以后的更新过程中被移除. 对于后者, 只有当可以支配  $x$  的个体进入  $X^{(t)}$  中, 移除才能进行. 根据定理 1, 若两个个体满足支配关系则意味着它们也满足  $\epsilon$ -支配关系. 而又因为支配关系具有传递性, 故在输出种群  $X^{(t)}$  中必有一个体可以  $\epsilon$ -支配  $x$ , 这与假设矛盾. 另一方面, 拒绝操作发生在算法 4 的第 2 或第 9 行. 同理可知, 如果个体  $x$  被拒绝, 则表明种群中存在某一个体对它具有  $\epsilon$ -支配关系. 而根据定理 2, 即使这样的个体在以后的更新过程中被移出, 其替代个体仍然可以  $\epsilon$ -支配  $x$ , 亦与假设矛盾.

故综合可知,  $X^{(t)}$  是  $X(t)$  的一个  $\epsilon$ -Pareto 解集. 定理得证. 证毕.

可见利用算法 4 的更新策略, 可以保证最终的输出种群为 Pareto 最优解集的子集, 且其中的个体成员的  $\epsilon$ -邻域内都无其它个体, 保证了均匀的分布性. 值得注意的是, 在输出种群中可能存在一些个体, 这里我们称之为“临近点”. 它们位于 Pareto 前沿的边缘, 可以  $\epsilon$ -支配一些虽然处在前沿面上但不具有 Pareto 支配关系的个体. 而当进化结束时, 如果能够支配这些“临近点”的个体没有生成, 则它们将会保留到最终的输出种群中. 不过根据定理 2 可知, 这些点的存在不会影响算法 4 的收敛性. 实际上, 它们仍可以看作当前种群中的 Pareto 最优解, 因为一旦能够支配它们的个体生成, 它们将会从最

终种群中移除。

#### 4.2 EDMOE 算法流程

结合算法 4 中的更新策略, EDMOE 的算法流程如下:

1. 随机生成初始种群  $P(t)$ , 将其中的不可支配个体复制到第二种群  $A(t)$ , 并求出其在目标  $i$  上的极值点  $p_i, i=1, 2, \dots, m$ , 并设置进化代数  $t=0$ .
2. 从  $A(t)$  中随机选择一个个体, 设其为  $a$ .
3. 以  $p_i$  和  $a$  作为父代, 如果  $p_i=a$  则更换其它的目标的极值个体  $p_j, j \neq i$ . 对所选父代遗传操作(交叉和变异), 生成的子代个体设为  $q_1, q_2$ .
4. 比较  $q_1, q_2$  的支配关系, 选择非被支配的个体; 如果两者互为不可支配, 则比较  $\epsilon$ -支配关系, 如果均不可行, 则随机选择一个作为优胜者, 然后利用算法 4 的更新策略, 将其加入种群  $A(t)$  之中.
5. 如果不满足终止条件, 令  $t=t+1$ , 从步 2 开始重新进化; 否则,  $A(t)$  即为最终的输出种群.

显然, EDMOE 算法可以分解为统一模型的特例. 因为其选择进行交配的个体都是当前种群中的非被支配个体即精英个体, 故 EDMOE 在进化过程中的精英强度将一直为 1.0, 为高精英强度进化算法. 如文献[16]所述, 在多目标进化算法中高精英参与比例有利于种群的收敛. 由于父代个体之一  $p_i (i=1, 2, \dots, m)$  为所对应目标的当前极值, 故若  $p_i=a$ , 则  $p_j$  必不等于  $a, j \neq i$ , 除非精英种群中只有唯一的个体. 所以这种策略可以避免对相同的个体进行遗传操作.

EDMOE 与  $\epsilon$ -MOEA 在算法结构尤其是精英种群的更新策略上有显著的区别:

首先,  $\epsilon$ -MOEA 根据  $\epsilon$  的支配关系将目标函数空间中的点转化为网格向量, 这样通常会导致一些在某一目标上的函数值是极端值的个体被支配, 从而缺失于最终的输出种群. 如文献[17]所述, 如果这些代表 Pareto 前沿边界的个体被排除, 精英种群所逼近的前沿面将会收缩, 从而算法将收敛到部分的 Pareto 前沿面. 但是采用算法 4 的更新策略, 当前种群的极端个体将会保留到下一代, 除非有可以支配它们的个体产生, 而若其能够支配具有极端值的个体, 则必在对应目标上具有更好的边界取值.

其次, 在网格向量的计算过程中, 尤其在对实际问题的处理时, 需要各个目标函数上可能取得的最小值的信息. 虽然它们可以在算法进行过程中获得, 但当新的最小值产生后, 需要对当前的精英种群中个体的网格向量值进行更新. 这些都将增加计算量. 采用 EDMOE 的更新策略, 只需要各个目标上的  $\epsilon$

参数值, 而且更新操作中仅需比较新个体与精英种群的支配关系, 计算复杂度同种群大小呈线性关系.

最后, 如图 1 所示, 在  $\epsilon$ -MOEA 中, 对于同一网格中个体的处理, 无法保证相邻网格的个体保持一致的分布. 而根据算法 4 的更新策略, 一旦个体在精英种群中确定, 其的  $\epsilon$ -邻域内将不再有个体被包括在最终的种群中(如图 2 所示). 因此相比  $\epsilon$ -MOEA, 其将更有效地保持种群的多样性.

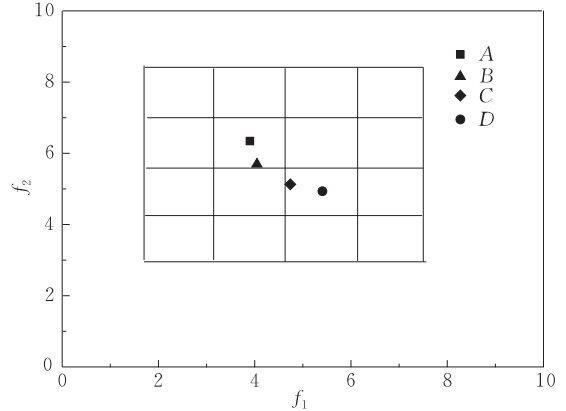


图 1  $\epsilon$ -MOEA 的网格操作策略(图中所有个体都属于 Pareto 前沿面; 点 A 和 B, 点 C 和 D 分别位于同一个网格内. 根据  $\epsilon$ -MOEA 算法网格内的操作策略, 点 A 和 D 将分别被点 B 和 C 取代, 种群分布的均匀性将降低)

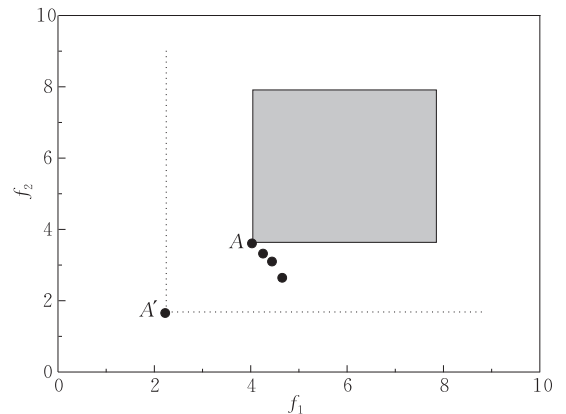


图 2 EDMOE 的  $\epsilon$ -支配策略(点 A 所支配的区域如图阴影所示. 根据  $\epsilon$ -支配关系, 其支配域扩展到点 A' 所支配的区域即虚线所包括的范围, 从而点 A 临近的个体将从精英种群中排除)

#### 4.3 自适应 $\epsilon$ 调整策略

根据上述讨论可见,  $\epsilon$  取值在基于  $\epsilon$ -支配的进化算法中起关键作用. Laumanns<sup>[13]</sup> 曾给出  $\epsilon$  的估值公式, 并提出一种动态调整  $\epsilon$  值以获取给定规模的解集的方法. 首先从一个最小的  $\epsilon$  值开始, 当精英种群中的个体数量超过事先给定的最大规模时, 则按照某种规则增大  $\epsilon$  取值. 然而当新的  $\epsilon$  值设定后, 精

英种群中个体的网格向量值需要更新,这需要额外的计算量.另一方面,当新的网格向量值取定后,精英种群内的个体需按照其新的网格向量坐标重新确定网格的支配关系,同样将耗费额外的计算量.

事实上,我们发现在 EDMOEA 算法中大的  $\epsilon$  取值反而更有利于算法的收敛(见 5.1 节),故据此,我们提出一种自适应的  $\epsilon$  调整方法如下.

1. 开始于一个最大的  $\epsilon$  取值;
2. 如果满足终止条件则终止调整策略,否则按照预先设定的下降幅度  $\Delta_i, i=1,2,\cdots,m$ ,减少当前的  $\epsilon$  取值,即如下式所示:

$$\epsilon_i \leftarrow \epsilon_i - \Delta_i.$$

显然当从一个大的  $\epsilon$  取值开始时,精英种群中现存个体的  $\epsilon$ -支配关系不会因为  $\epsilon$  值的降低而发生变化.因此没有必要再对它们进行重新的比较.结合如文献[17]的终止法则,我们给出了如下的终止条件:

- (1) 当适应值函数计算次数超过给定的最大上限(这里设为 25000);
- (2) 当在给定的一段连续的进化代数后,精英种群没有新的个体出现.因为精英种群中新产生的个体将丰富当前 Pareto 前沿的多样性或者将其朝向真实的 Pareto 前沿逼近.
- (3) 当在给定的一段连续的进化代数后,精英种群中的极端个体的目标函数值没有变化.精英种群中的极端值的变化在一定程度上也反应了种群的收敛性.于是同上一个终止法则相结合,如果它们都在给定的计算次数后保持不变,则表明深度搜索已经充分进行,需要降低  $\epsilon$  的取值开始进行当前 Pareto 前沿的广度搜索.这样可以增加依靠当前 Pareto 前沿的多样性,引导搜索朝向真正的 Pareto 前沿面推进.
- (4) 当  $\epsilon$  的值达到其的底线.

可见这种自适应的调整策略类似于模拟退火的思路,当按照既定的判断标准,种群进化没有进展时,则按照如上的调整公式,降低  $\epsilon$  在各个目标函数上的限制范围,从而促使种群的进化.注意到第 2 和

第 3 个终止法则中有一个重要的参数即给定的一段连续进化代数,本文中我们将其设置为 200.同时设定最大的  $\epsilon$  取值为 0.06,最小的为 0.0006.

### 5 实验结果和讨论

在本节中,计算实验采用了多目标进化算法测试中被广泛应用的 5 个多目标函数<sup>[10]</sup>,这些函数包括双目标函数,具有凸性、多模态等不同特性.首先讨论了在不同  $\epsilon$  取值下的 EDMOEA 的性能,然后通过其它通用多目标进化算法,如 NSGAI<sub>II</sub>、SPEA2、 $\epsilon$ -MOEA 等,进行比较分析,表明了 EDMOEA 和 AEDMOEA 的优越性.

#### 5.1 对于不同 $\epsilon$ 取值的 EDMOEA 算法的收敛性

首先,分别采用 5 个不同的  $\epsilon$  值 0.0006,0.006,0.06,0.6,0.9(这里我们认为在各个目标上的  $\epsilon$  取值一致且算法进化种群大小为 100,进行 25000 次适应值评价),结果表明当  $\epsilon$  为较大的取值,EDMOEA 可以更好地收敛到 Pareto 前沿曲面.

采用当代距离指标(General Distance, GD)<sup>[18]</sup>包括的 GD-Max 和 GD-Min 两种性能指标,评测最终种群同 Pareto 前沿面的接近程度.它们各自表示最终近似种群中的个体到 Pareto 前沿面的距离的最大值和最小值.实验结果如表 1 和表 2 所示.

如表 1 和表 2 所示,随着参数  $\epsilon$  值的增大,EDMOEA 在各个问题上的 GD-Max 和 GD-Min 的取值呈先减小后增大趋势,表中加粗部分即是各自变化的转折点.对于 ZDT1,ZDT2,ZDT3 问题,当  $\epsilon$  增至 0.06 时 GD-Min 显著减小,表明种群中已有个体充分接近真实 Pareto 前沿.对于 ZDT4 问题,尽管当  $\epsilon$  取得 0.0006 时,GD-Min 最小,但当  $\epsilon$  增至 0.06 时,GD-Max 值最小,即整体收敛性在  $\epsilon=0.06$  时最好.当  $\epsilon=0.06$  时,ZDT6 问题的 GD-Max 显著减小,在  $\epsilon=0.6$  时达到最小,而后其 GD-Min 随  $\epsilon$  增大呈小幅下降,当  $\epsilon=0.6$  时最小.需要注意的是,

表 1 GD-Max 值的平均值(Mean)和标准差(Stdev)

| $\epsilon$ |       | ZDT1              | ZDT2              | ZDT3              | ZDT4              | ZDT6              |
|------------|-------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 0.0006     | Mean  | 1.2119E-02        | 1.0863E-03        | 1.1407E-02        | 1.1129E-02        | 1.3745E-02        |
|            | Stdev | 2.7043E-03        | 1.4911E-05        | 1.6469E-03        | 3.4330E-03        | 9.8181E-05        |
| 0.006      | Mean  | 2.6564E-03        | 1.0806E-03        | 6.9647E-03        | 3.3678E-03        | 1.3027E-02        |
|            | Stdev | 1.3827E-03        | 8.6884E-05        | 1.4223E-03        | 1.2395E-03        | 3.3852E-04        |
| 0.06       | Mean  | 6.2141E-04        | 7.8769E-04        | <b>1.4519E-03</b> | <b>1.4695E-03</b> | 7.3574E-03        |
|            | Stdev | 6.8781E-05        | 1.8808E-04        | 3.3364E-04        | 1.3116E-03        | 2.2523E-03        |
| 0.6        | Mean  | 2.9802E-09        | <b>2.4039E-15</b> | 1.8433E-03        | 1.9133E-03        | <b>6.0361E-04</b> |
|            | Stdev | 4.0809E-09        | 1.9155E-16        | 1.0071E-03        | 1.0973E-03        | 6.0102E-05        |
| 0.9        | Mean  | <b>1.4901E-09</b> | 2.8339E-15        | 1.6932E-02        | 5.4584E-03        | 6.9877E-04        |
|            | Stdev | 3.3320E-09        | 3.6043E-16        | 3.6553E-02        | 3.6419E-03        | 4.2815E-05        |

表 2   GD-Min 值的平均值和标准差

| $\epsilon$ |       | ZDT1              | ZDT2              | ZDT3              | ZDT4              | ZDT6              |
|------------|-------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 0.0006     | Mean  | 1.1677E-06        | 8.1417E-07        | 4.1711E-07        | <b>4.4446E-05</b> | 4.5246E-04        |
|            | Stdev | 3.2006E-07        | 4.2412E-07        | 7.0470E-08        | 3.4413E-05        | 7.1094E-06        |
| 0.006      | Mean  | 1.0209E-07        | 4.2881E-08        | 4.4036E-09        | 2.1387E-04        | 4.4955E-04        |
|            | Stdev | 2.8099E-08        | 2.5259E-08        | 2.3637E-09        | 2.1250E-04        | 3.7961E-05        |
| 0.06       | Mean  | 8.1530E-16        | 6.6754E-16        | <b>2.3022E-09</b> | 4.6551E-04        | 4.3190E-04        |
|            | Stdev | 1.0178E-16        | 1.5594E-16        | 3.1532E-09        | 6.6613E-04        | 8.7040E-05        |
| 0.6        | Mean  | <b>5.9999E-16</b> | <b>3.9996E-16</b> | 4.4703E-09        | 9.1825E-04        | <b>4.0733E-04</b> |
|            | Stdev | 2.1477E-16        | 9.9460E-17        | 4.0808E-09        | 4.9524E-04        | 5.1721E-05        |
| 0.9        | Mean  | 8.9474E-16        | 8.4396E-16        | 1.5822E-04        | 2.4948E-03        | 4.7580E-04        |
|            | Stdev | 2.7979E-17        | 9.9292E-17        | 5.4430E-05        | 1.6005E-03        | 4.7043E-05        |

当  $\epsilon$  取值过大时,最终收敛种群的规模有限(表 3 可见),例如当  $\epsilon=0.6,0.9$  时,最终种群中只有两个个体在 Pareto 前沿两端,即各个目标函数的极值点.所以尽管这样得到的 GD-Max 和 GD-Min 值很小,但这十分不利于决策者做出最好的选择.事实上,在进化算法的搜索过程中,深度搜索和广度搜索是相互交叉进行的.当选择大的  $\epsilon$  取值的时候,EDMOEA 偏重于深度搜索即引导搜索朝向真实的 Pareto 前沿面,而当采用小的  $\epsilon$  取值时,EDMOEA 则在当前的 Pareto 前沿面进行搜索而取得更好的分布性.故我们推荐从较大的  $\epsilon$  取值开始,如  $\epsilon=0.06$ ,直至减少到最小的  $\epsilon$  值.

表 3   最终种群大小的平均值

| $\epsilon$ | 平均值   |       |       |       |       |
|------------|-------|-------|-------|-------|-------|
|            | ZDT1  | ZDT2  | ZDT3  | ZDT4  | ZDT6  |
| 0.0006     | 681.7 | 769.8 | 302.6 | 782.0 | 528.6 |
| 0.006      | 98.9  | 90.7  | 41.5  | 99.9  | 87.6  |
| 0.06       | 10.4  | 8.5   | 5.9   | 10.0  | 9.4   |
| 0.6        | 2.0   | 2.0   | 3.0   | 2.0   | 2.0   |
| 0.9        | 2.0   | 2.0   | 2.0   | 2.0   | 2.0   |

5.2   EDMOEA 和 AEDMOEA 同其它多目标进化算法的比较

在本节中,EDMOEA、采用自适应  $\epsilon$  调整策略的 AEDMOEA、NSGAI、 $\epsilon$ -MOEA、SPEA2 等,均用于求解 5 个测试函数,并针对收敛性、分布性及计算效率对各个算法进行比较分析.

具体的设置如下:所有初始种群中个体的基因都是随机产生且采用实数编码.交叉操作采用仿二进制交叉算子(Simulated Binary Crossover)<sup>[19]</sup>,变异操作采用多项式变异算子(Polynomial Mutation)<sup>[20]</sup>,其中的交叉算子参数  $\eta_c=15$ ,变异算子参数  $\eta_m=20$ .对于 EDMOEA 算法,在 5 个测试函数上都设置相同的  $\epsilon$  参数, $\epsilon_i=0.006(i=1,2)$ .SPEA2 算法的  $k$ -近邻参数  $k$  计算方法如下: $k=\sqrt{N+\overline{N}}$ ( $N$  表示种群, $\overline{N}$  表示精英种群大小).AEDMOEA,

EDMOEA 和  $\epsilon$ -MOEA 都采用稳态的进化策略,且适应值评价的最大次数为 25000.所有算法的其它共有参数设置如表 4 所示.

表 4   参数设置( $nreal$  表示自变量个数)

| 算法               | 交叉概率 | 变异概率      | 种群大小 | 精英种群大小 |
|------------------|------|-----------|------|--------|
| AEDMOEA          | 1.0  | $1/nreal$ | 100  | —      |
| EDMOEA           | 1.0  | $1/nreal$ | 100  | —      |
| $\epsilon$ -MOEA | 1.0  | $1/nreal$ | 100  | —      |
| NSGAI            | 1.0  | $1/nreal$ | 100  | 100    |
| SPEA2            | 0.5  | $1/nreal$ | 100  | 100    |

为了比较 5 个算法的性能,采用一元评价指标函数.所谓一元评价指标函数是从向量集到实数集的映射<sup>[21]</sup>: $I:\Omega\rightarrow R$ .利用实数集合上的全序关系,可以比较在  $\Omega$  空间中的不同向量集合之间的质量.不同的一元指标函数采用了不同的偏好信息.对于给定的集合  $A$ ,定义  $I(A)=I(A,R)$ ,计算  $A$  与参考集合  $R$  的指标函数值,其值越小,表明所用集合的质量越高.我们采用比较常用的两个指标函数:Hypervolume 指标函数(记为  $I_H$ )<sup>[4]</sup>和 Epsilon 指标函数(采用加的形式)(记作  $I_\epsilon$ )<sup>[22]</sup>.因为两个指标函数采用了不同的偏好信息,若对于相同的比较集合  $A$  和  $B$ ,它们指标值的序关系分别为  $I_H(A)<I_H(B)$  和  $I_\epsilon(A)>I_\epsilon(B)$ ,则可以推断这两个集合是等价的.其中,参照集合  $R$  构造如下<sup>[21]</sup>:联合 5 个算法在运行中所产生的最终近似种群,然后去除其中被支配的和重复的个体.这样,参照集合  $R$  中的个体即不被任何联合种群中个体所支配.实验结果如表 5~表 9 所示.

表 5   ZDT1 问题的 Hypervolume 和 Epsilon 指标的平均值(Avg)和标准差(Stdev)

| 算法               | Hypervolume       |            | Epsilon           |            |
|------------------|-------------------|------------|-------------------|------------|
|                  | Avg               | Stdev      | Avg               | Stdev      |
| AEDMOEA          | <b>2.3221E-03</b> | 4.5232E-04 | <b>1.0038E+00</b> | 6.1064E-04 |
| EDMOEA           | <b>4.1366E-03</b> | 7.3935E-05 | <b>1.0052E+00</b> | 1.5816E-04 |
| $\epsilon$ -MOEA | 5.5432E-03        | 1.1594E-04 | 1.0054E+00        | 4.0785E-04 |
| NSGAI            | 6.6350E-03        | 1.7487E-04 | 1.0097E+00        | 1.4182E-03 |
| SPEA2            | 5.1888E-03        | 1.7263E-04 | 1.0057E+00        | 1.9069E-04 |

表 6 ZDT2 问题的 Hypervolume 和 Epsilon 指标的平均值和标准差

| 算法               | Hypervolume        |             | Epsilon            |             |
|------------------|--------------------|-------------|--------------------|-------------|
|                  | Avg                | Stdev       | Avg                | Stdev       |
| AEDMOEA          | <b>1. 7763E-03</b> | 4. 7114E-04 | <b>1. 0027E+00</b> | 5. 9819E-04 |
| EDMOEA           | 4. 2077E-03        | 9. 6955E-05 | 1. 0043E+00        | 4. 7361E-04 |
| $\epsilon$ -MOEA | <b>3. 9445E-03</b> | 1. 5821E-04 | <b>1. 0035E+00</b> | 7. 7450E-05 |
| NSGAI            | 6. 5375E-03        | 4. 3199E-04 | 1. 0077E+00        | 1. 1035E-03 |
| SPEA2            | 5. 4256E-03        | 1. 4433E-04 | 1. 0050E+00        | 1. 8684E-04 |

表 8 ZDT4 问题的 Hypervolume 和 Epsilon 指标的平均值和标准差

| 算法               | Hypervolume        |             | Epsilon            |             |
|------------------|--------------------|-------------|--------------------|-------------|
|                  | Avg                | Stdev       | Avg                | Stdev       |
| AEDMOEA          | <b>5. 6634E-03</b> | 1. 8432E-03 | 1. 0090E+00        | 2. 1064E-03 |
| EDMOEA           | <b>6. 9780E-03</b> | 2. 9405E-03 | <b>1. 0079E+00</b> | 1. 6726E-03 |
| $\epsilon$ -MOEA | 3. 5970E-02        | 2. 2407E-02 | 1. 0968E+00        | 6. 9649E-02 |
| NSGAI            | 9. 2137E-03        | 3. 0440E-03 | 1. 0096E+00        | 1. 2534E-03 |
| SPEA2            | 7. 9398E-03        | 2. 7921E-03 | <b>1. 0071E+00</b> | 1. 4421E-03 |

ZDT1 问题是检测算法寻找具有凸性的 Pareto 最优解集的能力,如表 5 所示. AEDMOEA 在 Hypervolume 和 Epsilon 指标上各自取得最小值而 EDMOEA 取得次小值,NSGAI 所得到的近似集合质量最低. ZDT2 问题是检测算法对于非凸性 Pareto 最优解集的处理能力, AEDMOEA 最优,  $\epsilon$ -MOEA 次之,而 NSGAI 表现最差. ZDT3 问题的 Pareto 前沿面是非连续的, SPEA2 在 Hypervolume 指标上次优,而 EDMOEA 在 Epsilon 指标上要优于 SPEA2 劣于 AEDMOEA. 这表明 SPEA2 和 EDMOEA 在 ZDT3 问题上等价,且均劣于 AEDMOEA. 表 8 所示

表 10 ZDT4 和 ZDT6 问题的 Hypervolume 指标的非参数 kruskal-wallis 检验的  $p$  值<sup>[21]</sup>

|                  | ZDT4 问题 $p$ 值 |             |                  |             |             | ZDT6 问题的 $p$ 值 |             |                  |             |             |
|------------------|---------------|-------------|------------------|-------------|-------------|----------------|-------------|------------------|-------------|-------------|
|                  | AEDMOEA       | EDMOEA      | $\epsilon$ -MOEA | NSGAI       | SPEA2       | AEDMOEA        | EDMOEA      | $\epsilon$ -MOEA | NSGAI       | SPEA2       |
| AEDMOEA          | —             | 6. 2008E-02 | 4. 3926E-13      | 1. 6544E-05 | 1. 2902E-03 | —              | 9. 0874E-05 | 6. 7804E-09      | 3. 4672E-14 | 4. 7204E-12 |
| EDMOEA           | 9. 3799E-01   | —           | 3. 2003E-10      | 2. 6306E-03 | 6. 0566E-02 | 9. 9991E-01    | —           | 9. 0874E-05      | 7. 6634E-12 | 4. 9021E-09 |
| $\epsilon$ -MOEA | 1. 0000E+00   | 1. 0000E+00 | —                | 9. 9997E-01 | 1. 0000E+00 | 1. 0000E+00    | 9. 9991E-01 | —                | 9. 4174E-09 | 5. 9400E-05 |
| NSGAI            | 9. 9998E-01   | 9. 9737E-01 | 2. 9312E-05      | —           | 9. 0304E-01 | 1. 0000E+00    | 1. 0000E+00 | 1. 0000E+00      | —           | 9. 9979E-01 |
| SPEA2            | 9. 9871E-01   | 9. 3943E-01 | 2. 0391E-07      | 9. 6957E-02 | —           | 1. 0000E+00    | 1. 0000E+00 | 9. 9994E-01      | 2. 1363E-04 | —           |

注. 原假设:行样本同对应的列样本服从相同的分布;备择假设:行样本具有比列样本更好的分布. 其中  $p$  为单尾值,置信参数  $\alpha=0.01$ .

上述讨论表明,对于双目标优化问题,除含有大量局部 Pareto 前沿面的 ZDT4 问题外, AEDMOEA 算法性能均要优于其它算法. 即使在 ZDT4 问题上,其也是等价于 SPEA2 和 EDMOEA 算法. 另外,根据两个性能评价指标的标准方差值,可以发现上述算法的标准差显著小于其平均值,表明算法具有很好的鲁棒性,并不依赖于初始种群.

因此,与固定  $\epsilon$  取值 EDMOEA 和其它常用多目标进化算法相比,采用自适应  $\epsilon$  调整策略的 AEDMOEA 算法具有更好的性能. 事实上,动态  $\epsilon$

表 7 ZDT3 问题的 Hypervolume 和 Epsilon 指标的平均值和标准差

| 算法               | Hypervolume        |             | Epsilon            |             |
|------------------|--------------------|-------------|--------------------|-------------|
|                  | Avg                | Stdev       | Avg                | Stdev       |
| AEDMOEA          | <b>4. 1403E-04</b> | 4. 9025E-05 | <b>1. 0010E+00</b> | 1. 2560E-04 |
| EDMOEA           | 4. 1264E-03        | 2. 7001E-04 | <b>1. 0050E+00</b> | 3. 1227E-04 |
| $\epsilon$ -MOEA | 5. 6172E-03        | 2. 1776E-04 | 1. 0064E+00        | 5. 7231E-05 |
| NSGAI            | 2. 6110E-03        | 2. 4214E-04 | 1. 0052E+00        | 9. 0772E-04 |
| SPEA2            | <b>2. 5096E-03</b> | 1. 7234E-04 | 1. 0052E+00        | 5. 8042E-04 |

表 9 ZDT6 问题的 Hypervolume 和 Epsilon 指标的平均值和标准差

| 算法               | Hypervolume        |             | Epsilon            |             |
|------------------|--------------------|-------------|--------------------|-------------|
|                  | Avg                | Stdev       | Avg                | Stdev       |
| AEDMOEA          | <b>7. 7802E-04</b> | 1. 4189E-04 | <b>1. 0012E+00</b> | 1. 3933E-04 |
| EDMOEA           | <b>5. 0907E-03</b> | 9. 8829E-05 | <b>1. 0047E+00</b> | 1. 1774E-04 |
| $\epsilon$ -MOEA | 8. 2441E-03        | 5. 6592E-04 | 1. 0058E+00        | 3. 6547E-04 |
| NSGAI            | 1. 8575E-02        | 1. 3916E-03 | 1. 0129E+00        | 1. 7363E-03 |
| SPEA2            | 1. 5011E-02        | 1. 4401E-03 | 1. 0110E+00        | 1. 9763E-03 |

为 5 个算法在处理具有大量局部 Pareto 前沿点的 ZDT4 问题的结果. AEDMOEA 和 SPEA2 分别在 Hypervolume 和 Epsilon 指标上各自取得最小值, EDMOEA 在 Hypervolume 指标上次优,而其 Epsilon 指标值要略优于 AEDMOEA,故可知 3 个算法在 ZDT4 问题上等价. ZDT6 问题的 Pareto 前沿面的个体呈非均匀分布. AEDMOEA 在两个指标函数上各自取得最优值, EDMOEA 次之, NSGAI 最差. 由表 10 的关于 ZDT4 和 ZDT6 问题的 Hypervolume 指标的非参数 kruskal-wallis 检验,亦可知 AEDMOEA 和 EDMOEA 均要显著优于其它的比较算法.

调整策略使得算法在搜索过程中更加侧重深度搜索. 当搜索停滞时,根据第二和第三终止法则,通过减少  $\epsilon$  取值提高了当前 Pareto 前沿的多样性,可以将搜索导向新的 Pareto 前沿.

实验表明,给定的连续进化代数 and  $\epsilon$  值下降幅度对于算法性能有重要影响,如何更加精确地设定这些参数需要更加细致的研究. 另外,一旦个体在精英种群中被确定,则在其  $\epsilon$ -邻域内不再有效局部搜索. 因此,若采用新的变异算子,并以较大概率将个体跳出其  $\epsilon$ -邻域,也可以有效地保持种群多样性.

## 6 结 论

参考修正的统一模型 UMMEA, 本文提出了一类新的多目标进化算法 EDMOEA, 该算法采用了  $\epsilon$ -支配关系和精英种群升级策略并详细分析了  $\epsilon$  取值对算法收敛性的影响, 提出了一种新的自适应  $\epsilon$  调整策略. 实验表明, 与 SPEA2、 $\epsilon$ -MOEA、NSGAII 等常用多目标算法相比, 这种包括自适应的  $\epsilon$  调整策略的 EDMOEA 算法在处理双目标问题上具有显著的优越性.

## 参 考 文 献

- [1] Schaffer D J. Multipleobjective optimization with vector evaluated genetic algorithms [Ph. D. dissertation]. Vanderbilt University, 1984
- [2] Coello C A C. Evolutionary multi-objective optimization: A historical view of the field. *Computational Intelligence Magazine*, 2006, 1(1): 28-36
- [3] Laumanns M et al. A unified model for multi-objective evolutionary algorithms with elitism//Proceedings of the 2000 Congress on Evolutionary Computation(CEC'2000). La Jolla, CA, USA, 2000: 46-53
- [4] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257-271
- [5] Zitzler E, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. Swiss Federal Institute of Technology (ETH); Computer Engineering and Networks Laboratory (TIK), Zurich; Technique Report: 103, 2001
- [6] Knowles J, Corne D. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization//Proceedings of the Congress on Evolutionary Computation (CEC'99). Washington, DC, USA, 1999: 98-105
- [7] Goldberg D E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989
- [8] Valenzuela C L. A simple evolutionary algorithm for multi-objective optimization (SEAMO)//Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02). Honolulu, HI, USA, 2002, 1: 717-722
- [9] Rajeev K, Peter R. Improved sampling of the Pareto-front in multiobjective genetic optimizations by steady-state evolution: A Pareto converging genetic algorithm. *Evolutionary Computation*, 2002, 10(3): 283-314
- [10] Deb K et al. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 2005, 13(4): 501-525
- [11] Berry A, Vamplew P. An efficient approach to unbounded bi-objective archives: Introducing the mak\_tree algorithm//Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. Seattle, Washington, USA, 2006: 619-626
- [12] Zitzler E et al. A tutorial on evolutionary multiobjective optimization. Workshop on Multiple Objective Metaheuristics (MOMH 2002). Berlin, German; Springer-Verlag, 2004
- [13] Laumanns M et al. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 2002, 10(3): 263-282
- [14] Coello C A C et al. *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic, 2002
- [15] Cui Xun-Xue, Li Miao, Fang Ting-Jian. Multiobjective cooperation evolutionary algorithm. *Chinese Journal of Computers*, 2001, 24(9): 979-984(in Chinese)  
(崔逊学, 李森, 方廷健. 多目标协调进化算法研究. *计算机学报*, 2001, 24(9): 979-984)
- [16] Parks G T, Miller I. Selective breeding in a multiobjective genetic algorithm//Proceedings of the 5th International Conference on Parallel Problem Solving from Nature. Amsterdam, The Netherlands, 1998: 250-259
- [17] Fieldsend J E et al. Using unconstrained elite archives for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2003, 7(3): 305-323
- [18] Veldhuizen V. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations [Ph. D. dissertation]. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999
- [19] Deb K, Agrawal R B. Simulated binary crossover for continuous search space. *Complex Systems*, 1995, 9(1): 115-148
- [20] Deb K, Goyal M. A combined genetic adaptive search Genetic Algorithm for engineering design. *Computer Science and Informatics*, 1996, 26(4): 30-45
- [21] Knowles J et al. A tutorial on the performance assessment of stochastic multiobjective optimizers. Swiss Federal Institute of Technology (ETH). Computer Engineering and Networks Laboratory (TIK), Zurich; Technique Report: 214, 2005
- [22] Zitzler E et al. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 117-132



**LIU Liu**, born in 1982, Ph.D. candidate. His research interests include the theory and applications of multi-objective evolutionary optimization.

**LI Min-Qiang**, born in 1965, professor, Ph.D. supervisor. His research interests include the theory of evolutionary algorithm, data mining and machine learning.

**LIN Dan**, born in 1968, associate professor. His research interests include the theory and applications of genetic algorithm.

Background

The research of this paper is supported by the National Science Foundation of China under grant Nos. 70171002, 70571057 and the Program for New Century Excellent Talents in University of China under grant No. NCET-05-0253. The project No. 70171002 made a detailed investigation on the theory of evolutionary computation; and the project No. 70571057 and the NCET-05-0253 focus on co-evolution computation and the applications in handling multi-objective optimization problems.

The evolutionary algorithms have been recognized as an efficient approach to solve the MOPs since 1985. The existing evolutionary algorithms for the MOPs employed mainly the Pareto domination relation to determine the fitness function so that the selection pressure was achieved to drive the population towards the true Pareto front. However, it is not

easy to manipulate and computationally expensive. Recently, the steady-state evolutionary algorithms that make use of the pair-wise comparison instead of the fitness functions to bias the selection of parents are proposed. Further more, a relaxed form of the Pareto dominance, denoted as the  $\epsilon$ -dominance, is becoming the popular mechanism to regulate convergence of multi-objective evolutionary algorithms.

This paper adopts a simple truncating method based on the  $\epsilon$ -dominance relationship in archive population, and presents the  $\epsilon$ -dominance multi-objective evolutionary algorithm (EDMOEA) and adaptive EDMOEA (AEDMOEA). Both EDMOEA and AEDMOEA could guarantee the convergence of the population to the subset of the true Pareto optima and the preservation of boundary solutions.