

# 基于自适应 $\varepsilon$ 占优的多目标差分演化算法

许 金<sup>1,2</sup>, 谷 琼<sup>1,3</sup>, 蔡之华<sup>2</sup>, 龚文引<sup>2</sup>

(1.湖北文理学院数学与计算机科学学院,湖北 襄阳 441053)

(2.中国地质大学计算机学院,湖北 武汉 430074)

(3.西南大学逻辑与智能研究中心,重庆 400715)

**[摘要]** 求解多目标优化问题最重要的目的就是获得尽可能逼近真实最优解和分布性良好的非支配解集. 为此,本文提出了一种基于自适应  $\varepsilon$  占优的正交多目标差分演化算法,该算法具有如下特征:1.利用正交设计和连续空间的量化来产生具有良好分布性的初始演化种群,不仅能降低算法的时间复杂度,也能使演化充分利用种群中的个体;2.采用在线 Archive 种群来保存算法求得的非支配解,并用自适应的  $\varepsilon$  占优更新 Archive 种群,以自适应的方式维持种群的多样性、分布性. 最后通过 5 个标准测试函数对算法的有效性进行了测试,并与其他的一些多目标优化算法进行了对比,实验结果显示,算法能够很好地逼近 Pareto 前沿,并具有良好的分布性.

**[关键词]** 多目标优化, Pareto 最优解, 差分演化, 正交设计, 自适应  $\varepsilon$  占优

**[中图分类号]** TP301.6 **[文献标志码]** A **[文章编号]** 1001-4616(2015)01-0119-09

## Differential Evolution Algorithm for Multi-Objective Optimization Based on Adaptive $\varepsilon$ -Dominance

Xu Jin<sup>1,2</sup>, Gu Qiong<sup>1,3</sup>, Cai Zhihua<sup>2</sup>, Gong Wenying<sup>2</sup>

(1.School of Mathematics and Computer Science, Hubei University of Arts and Science, Xiangyang 441053, China)

(2.School of Computer Science, China University of Geosciences, Wuhan 430074, China)

(3.Institute of Logic and Intelligence, Southwest University, Chongqing 400715, China)

**Abstract:** The purpose to solve multi-objective optimization is to get solutions closing to the true Pareto front as much as possible and having good diversity. To meet these two demands, an algorithm is proposed in this paper, which has these characteristics: firstly, it adopts the orthogonal design method with quantization technology to generate initial population whose individuals are scattered uniformly over the target search space. So the algorithm can use them sufficiently in the subsequent iterations. What's more, it is based on an adaptive  $\varepsilon$  concept to obtain a good distribution along the true Pareto-optimal solutions. Finally, experiments on five benchmark problems with different features have shown that this algorithm does well not only in distribution, but also in convergence when compared to other evolution algorithms.

**Key words:** multi-objective optimization, Pareto optimal solution, differential evolution, orthogonal design, adaptive  $\varepsilon$ -dominance

无论在自然科学还是社会科学领域,我们都希望能够最大化利益、最小化成本. 最大化利益、最小化成本实质上是一个多目标优化问题<sup>[1]</sup>,在此问题中,各子目标之间一般是相互冲突的,这就导致了不可能获得唯一的全局最优解,注重于某一个目标性能上的优化可能会造成其他目标性能的损失,因此简单地评价所求解的优劣是不科学的,对于一个解优劣性的评价应该考虑到所有子目标. 在多目标优化问题中,算法所求得的是些最优解的集合,一般称为 Pareto 最优解集,这和单目标优化问题中存在全局的唯一最优解是有很大的差别的. 解集中的每个解之间是互不支配的,它们之所以被称为最优解是因为在目标搜索空间中,不存在一个解能支配解集中任意一个个体. 多目标算法的目的就是根据特定的问题找到具有良好

收稿日期:2014-08-20.

基金项目:国家自然科学基金(61203307)、湖北省科技支撑计划公益性科技研究类项目(2012BKB068)、中国博士后科学基金面上项目(2014M560700)、重庆博士后特别资助项目(XM2014057).

通讯联系人:谷琼,副教授,研究方向:智能计算,机器学习,网络舆情. E-mail: gujone@163.com

多样性与分布性的非支配解。

在过去的几十年里,学者们提出了多种算法来求解多目标优化问题,这类算法主要有:聚合方法、VEGA<sup>[2]</sup>算法、字典序方法、 $\varepsilon$ -约束方法、目标向量、NSGA<sup>[3]</sup>、MOGA<sup>[4]</sup>、NPGA<sup>[5]</sup>等;90 年代后期,又提出了一些改进的算法包括:PAES<sup>[6]</sup>、SPEA<sup>[7]</sup>、NSGAII<sup>[8]</sup>、SPEA2<sup>[9]</sup>等算法。国内对于多目标进化算法的设计与理论研究也吸引了很多的学者,并取得了不错的成绩,比如,王宇平<sup>[10]</sup>、曾三友等人<sup>[11]</sup>分别将正交设计和演化算法相结合应用到多目标优化中,利用正交设计产生种群,加快了算法的收敛速度。关世华<sup>[12]</sup>等提出了一种基于  $\varepsilon$ -约束方法的增广 Lagrangian 多目标协同化算法。石川等人<sup>[13]</sup>提出了一种基于占优树的多目标进化算法,该算法利用占优树进行适应值分配,减少了个体间的比较。Gong 等人<sup>[14]</sup>把二水平正交杂交算子应用到 DE 中,并结合统计优化来选择最优个体,增加算法的鲁棒性。公茂果<sup>[15]</sup>等对进化多目标的最新进展进行了详细讨论,归纳出多目标优化的研究趋势,并对多目标优化的进一步发展进行展望。

尽管目前已经提出了很多演化算法来求解多目标优化问题,但是设计出一种高效、鲁棒性好的算法依然是一个具有挑战性的难题。为此,笔者提出了一种基于自适应  $\varepsilon$  的差分演化算法,称为 Pa $\varepsilon$ -ODEMO,它具有如下特性:

(1) 正交设计产生初始化种群,不仅降低了算法的时间消耗,而且使得初始种群能够均匀地分布于目标搜索空间;

(2) 采用在线归档集 Archive 来存储所求得的非支配解,并用自适应的  $\varepsilon$  值来动态地维持种群的多样性与分布性。

最后通过实验证明,所提出的算法能够很好地逼近真实 Pareto 前沿,以实验的方式证明了算法的有效性。

## 1 多目标差分演化算法

### 1.1 多目标优化问题及其相关定义

**定义 1**(多目标优化问题) 一个具有  $n$  个决策变量,  $m$  个约束限制函数,  $k$  个目标函数的多目标优化问题可以描述为:

$$\begin{aligned} \text{minimize: } & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x})), \\ \text{subject to: } & \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), \dots, e_m(\mathbf{x})) \geq 0, \\ \text{where: } & \mathbf{x} = (x_1, x_2, \dots, x_n) \in X, \\ & \mathbf{y} = (y_1, y_2, \dots, y_k) \in Y. \end{aligned} \quad (1)$$

其中  $\mathbf{x}$  是决策向量,  $\mathbf{y}$  是目标向量,  $X$  是决策向量空间,  $Y$  是目标向量空间, 它们满足约束限制函数  $\mathbf{e}(\mathbf{x})$ 。

**定义 2**(Pareto 占优) 也称 Pareto 支配, 设  $\mathbf{x} = (x_1, \dots, x_k)$  和  $\mathbf{y} = (y_1, \dots, y_k)$  是 2 个可行解, 当且仅当:

$$\forall i \in 1, \dots, k \quad x_i \leq y_i \quad \text{且} \quad \exists i \in 1, \dots, k \quad x_i < y_i. \quad (2)$$

就说  $\mathbf{x} = (x_1, \dots, x_k)$  Pareto 占优  $\mathbf{y} = (y_1, \dots, y_k)$ . 记作  $\mathbf{x} < \mathbf{y}$ .

**定义 3**( $\varepsilon$  占优) 设  $\mathbf{x} = (x_1, \dots, x_k)$  和  $\mathbf{y} = (y_1, \dots, y_k)$  是 2 个可行解, 当且仅当:

$$\forall i \in 1, \dots, k \quad x_i - \varepsilon \leq y_i. \quad (3)$$

就说  $\mathbf{x} = (x_1, \dots, x_k)$   $\varepsilon$  占优  $\mathbf{y} = (y_1, \dots, y_k)$ . 记作  $\mathbf{x} <_{\varepsilon} \mathbf{y}$ .

**定义 4**(Pareto 最优解) 给定一个解  $\mathbf{x}^*$ , 它是 Pareto 最优解, 当且仅当:

$$\exists \mathbf{x} \in X \quad \text{使得} \quad \mathbf{x} < \mathbf{x}^*. \quad (4)$$

**定义 5**(Pareto Front) 它是将所有的 Pareto 最优解映射到目标向量空间坐标中所形成的几何图形, 其数学描述如下:

$$POF = \{f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) \mid \mathbf{x} \in X\}. \quad (5)$$

### 1.2 差分演化算法

差分演化算法(DE)是一种随机的基于启发式的搜索算法,该算法采用实数编码的方式,是一种解决复杂优化问题的有效技术,目前已经是一种比较成熟的演化算法,其具体执行框架如下:

(1) 参数赋值,并产生初始的演化种群  $P_{G=0}$ , 令演化代数  $t=0$ ;

- (2) 计算初始种群  $P_{G=0}$  的适应值;
- (3) 如果满足停止条件,则停止算法的运行,并输出所得解,否则转步骤(4);
- (4) 对种群  $P_{G=t}$  进行变异、杂交及淘汰操作,并产生下一代演化种群  $P_{G=t+1}$ ;
- (5) 计算种群  $P_{G=t+1}$  的适应值;
- (6) 演化代数  $t=t+1$ ,转步骤(3).

需要指出的是,此处算法 Pa $\varepsilon$ -ODEMO 使用了一种混合选择的机制<sup>[16]</sup>,具体来说就是在演化前期,由于个体离真正的最优解差距较大,演化指导性不强,算法直接从演化种群中选择个体进行演化杂交操作,但是随着演化的迭代,后续的个体将越来越逼近真实的 Pareto 最优解,此时演化将直接从 Archive 中选择个体.

$$\text{individual} = \begin{cases} \text{random}, & \text{eval} < (\lambda \times \text{Max\_eval}), \\ \text{elitist}, & \text{otherwise.} \end{cases} \quad (6)$$

## 2 改进的算法——Pa $\varepsilon$ -ODEMO

### 2.1 正交初始化种群

传统的差分演化算法一般都是随机产生初始种群,这种方法产生的种群一般分布不均,降低了算法对初始种群的利用能力. 采用正交设计<sup>[16]</sup>的方法产生初始种群能够使初始种群更好地分布在决策向量空间中,从而具有更好的分布性和多样性. 正交初始化种群不仅能提高初始种群的利用能力,同时可以加快算法的收敛速度.

正交初始化种群的产生主要有两个步骤:

- (1) 产生正交矩阵<sup>[17]</sup>,它是一种拉丁方矩阵,用  $L_R(Q^c)$  来表示,具有  $Q$  个不同水平的正交矩阵;
- (2) 假设某一优化<sup>[18]</sup>,假设问题的某一决策变量的取值范围为  $[l, u]$ ,对应地将其量化为  $Q$  个水平  $a_1, a_2, \dots, a_Q$ ,其计算方法为:

$$a_i = l + (i-1)(u-l)/(Q-1) \quad i=1, \dots, Q. \quad (7)$$

### 2.2 自适应 $\varepsilon$ 值维持种群多样性

算法在种群更新操作中,每产生一个子代个体,不仅要新个体与父代个体进行比较,同时还要与归档集 Archive 中的每个个体进行比较. 在演化的前期,如果子代个体优于归档集中的个体,那么直接删除归档集中被支配的个体;如果子代个体与归档集中的所有个体都是互不支配的,那么将新个体加入到归档集中;如果新个体被归档集中的某个个体支配,则抛弃新个体. 在这种情况下,随着演化的进行,Archive 种群的规模可能越来越大,同时分布性越来越差(某个区域内的个体密度很大,而其他区域的密度很小). 为了获得较好的分布性,同时也为减小存储空间,算法采用了自适应的  $\varepsilon$  网格来保持群体的多样性. 当 Archive 集的规模达到预定值时,将 Archive 种群中的个体映射到网格,使得每个网格中只有一个个体. 然后用网格的方法来更新. 为了存储映射后的个体,为 Archive 中的每个个体分配了一个标识数组  $B = (B_1, \dots, B_r)$ ,  $r$  为目标函数的个数. 标识数组的计算公式如下:

$$B_i = \lfloor (f_i - f_i^{\min}) / \varepsilon_i \rfloor \quad i=1, \dots, r. \quad (8)$$

其中,  $f_i^{\min}$  表示第  $i$  个目标函数的最小值,这个最小值是当前 Archive 群体中个体的最小值,它是一直变化着的.  $\varepsilon_i$  是第  $i$  个目标函数的容错差,利用  $\varepsilon_i$  我们就可以将每个个体都映射到网格中.

为了产生网格需要计算相应的  $\varepsilon_i$  值. 不失一般性,假设某个目标函数的取值范围为  $1 \leq f_i \leq K$  (取值范围的获得来源于 Archive 群体),那么对于有  $r$  个目标的优化问题,它将产生  $\lfloor (K-1)/\varepsilon \rfloor^r$  个网格,在这个网格中最多能够容纳  $\lfloor (K-1)/\varepsilon \rfloor^{r-1}$  个非支配解(对于双目标就是对角线网格),对于具有 NP 个个体的演化种群,我们希望能最多产生出 NP 个非支配解,那么它将满足:

$$NP = \left( \frac{K-1}{\varepsilon} \right)^{r-1}. \quad (9)$$

这样我们就能很容易地计算每一个对应的  $\varepsilon_i$  值. 之所以会有自适应的  $\varepsilon_i$  值是因为在多目标优化问题中,对于某个目标而言,刚开始并不知道它的取值范围,但是在计算  $\varepsilon_i$  的时候又必须知道其取值范围,此处对于某个目标函数的最大值和最小值是从 Archive 种群中的个体得到,算法认为 Archive 中的个体是目前最优的,虽

然它们只是暂时的最优. 随着演化的进行,对于某个目标函数而言,它的最大值和最小值可能会发生变化,这样我们就能通过这个取值范围的变化随时调整  $\varepsilon_i$  值,从而调整个体映射到网格中的位置.

**算法 1** 自适应  $\varepsilon$  占优更新 Archive 种群

```

adaptive  $\varepsilon$ -domination to update the archive population
assign identify array  $B_a$  to the offspring using formula(8)
for all  $i=1$ :objective_size do
    if  $B_a[i] < -3$  then
        add the offspring to Archive;
        find maximum and minimum of each objective
        function in Archive;
        regenerate grid;return;
    end if
end for
for all  $i=1$ :ar_size then
    if  $B_a$  dominate  $B[i]$ 
        delete individual  $i$  from Archive;
    else if  $B[i]$  dominate  $B_a$ 
        discard the offspring;return;
    else
         $B_a$  and  $B[i]$  share a same grid
        if the offspring Pareto dominates individual  $i$ 
            using the offspring replace individual  $i$ ;
        end if
    else
        add the offspring to Archive;
    end if
end if
end for.
```

每当一个新个体产生时,我们就用式(8)来计算此个体的标识数组  $B_a$ . 显然,如果  $B_a[i] < 0$  说明  $f_i < f_i^{\min}$ ,此时我们需要更新第  $i$  个目标函数的最小值,然后重新产生新的网格并将所有个体重新映射到网格中. 但是为了避免频繁产生新网格而造成的时间耗费,在本算法中(如算法 1 所示)只有当  $B_a[i] < -3$  时,我们才更新目标函数最小值,产生新的网格,然后再将所有个体映射到网格中.

**算法 2** Pa $\varepsilon$ -ODEMO 总体算法流程

```

The main procedure of Pa $\varepsilon$ -ODEMO
Generate orthogonal initial evolution population;
flag=true,neval=0;
while neval<Max_eval
    for all  $i=1$ :NP
        if neval< $\lambda$  * Max_eval then
            Random selection;
            Apply DE/rand/1 to the population
        else
            Elitist selection;
            Apply DE/rand/1 to the population
        end if
        Evaluate the offspring;
        if offspring dominate pop[i] then
            Using the offspring replace pop[i];
        else if offspring non-dominated pop[i]
```

```

        Add offspring to the Child Population;
    end if
    if flag == true then
        Using Pareto dominance to update the Archive;
    else
        Using adaptive( to update the Archive;
        Combine the parent population and Child population and generate the next evolution population;
    if ar_size > N_F && flag == true then
        Generate the initial grid, map all individual to the grid and apply adaptive  $\varepsilon$ -domination to update the Archive;
        flag = false;
    end if
end for
Add extreme point to Archive;
end while.
    
```

### 3 算法性能测试与结果分析

#### 3.1 测试函数

为了测试具有各种特征(凸的、凹的、离散的、均匀分布等)的 Pareto 前沿的多目标问题,我们采用了下面 5 个标准测试函数:

测试用例 1:

$$\text{ZDT1} \quad \begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x) [1 - (1 - (x_1)/g(x))^{0.5}], \\ g(x) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n - 1). \end{cases}$$

ZDT1 中,决策变量个数  $n=30$ ,对每一个变量有  $x_i \in [0, 1]$ ,它有凸的 Pareto 最优阵面.

测试用例 2:

$$\text{ZDT2} \quad \begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x) [1 - (1 - (x_1)/g(x))^2], \\ g(x) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n - 1). \end{cases}$$

ZDT2 中,决策变量个数  $n=30$ ,对每一个变量有  $x_i \in [0, 1]$ ,它有凹的 Pareto 最优阵面.

测试用例 3:

$$\text{ZDT3} \quad \begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x) [1 - (1 - \sqrt{x_1/g(x)} - x_1 \sin(10\pi x_1)/g(x))], \\ g(x) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n - 1). \end{cases}$$

ZDT3 中,决策变量个数  $n=30$ ,对每一个变量有  $x_i \in [0, 1]$ ,它有离散的凹的 Pareto 最优阵面.

测试用例 4:

$$\text{ZDT6} \quad \begin{cases} f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1), \\ f_2(x) = g(x) [1 - (f_1(x)/g(x))^2], \\ g(x) = 1 + 9 \left[ \left( \sum_{i=2}^n x_i \right) / (n - 1) \right]^{0.25}. \end{cases}$$

ZDT6 中,决策变量个数  $n=10$ ,对每一个变量有  $x_i \in [0, 1]$ ,它有凹的非一致性分布的 Pareto 最优阵面.

测试用例 5:

$$\text{DTLZ1} \begin{cases} f_1(x) = 0.5x_1x_2(1 + g(x_M)), \\ f_2(x) = 0.5x_1(1 - x_2)(1 + g(x_M)), \\ f_3(x) = 0.5x_1(1 - x_1)(1 + g(x_M)), \\ g(x_M) = 100[|x_M| + \sum_{x_i \in X_M} (x_i - 0.5)^{0.5} - \cos(20\pi(x_i - 0.5))]. \end{cases}$$

DTLZ1 中,决策变量个数  $n=12$ ,对每一个变量有  $x_i \in [0,1]$ ,它是一个三目标的优化问题.

### 3.2 评价标准

(1)收敛性(Convergence  $\gamma$ ),用来衡量所求得的最优解集  $Q$  与真实 Pareto Front 之间的逼近程度,其计算公式如下:

$$\gamma = \sum_{i=1}^{|Q|} d_i / (|Q|). \quad (10)$$

其中  $d_i$  表示  $Q$  中第  $i$  个个体与 PF 中个体的最小欧氏距离.  $\gamma$  的值越小,表明算法所得到的解集越逼近真正的 Pareto 前沿,算法的收敛性也越好,这是首要的衡量指标.

(2) $\Delta$  指标由 Deb 等人于 2002 年提出,它主要用来评价算法所求解沿 Pareto 前沿分布的均匀程度,其计算公式如下:

$$\Delta = (d_f + d_l + \sum_{i=1}^{|Q|-1} |d_i - \bar{d}|) / (d_f + d_l + (|Q|-1)\bar{d}). \quad (11)$$

式中  $d_i$  是所求的解集  $Q$  中连续 2 个解(已排序)之间的欧式距离,  $\bar{d}$  是所有  $d_i$  的平均值,  $d_f$  和  $d_l$  分别表示真正 Pareto 前沿与所求得的解集中的边界点的欧式距离,它主要应用于双目标优化的解集的评估.  $\Delta$  值越小,表明算法的多样性越好,特别当  $\bar{d} = d_i$  时,  $\Delta = 0$ ,此时算法所求得的解沿着 Pareto 前沿等间距均匀分布.但是经过分析可知,这个性能评价指标也有它的缺陷性,比如:它主要用来评价双目标,有一定的局限性;其次若算法所求得的解中只有 2 个边界点,也就是说只有 2 个解,此时  $\Delta = 0$ ,但是这里并不能说明算法是一个好的算法,而且它更多的是评价算法的分布均匀程度.但是用它来评价正常算法与测试用例所求的解,也是一个有效的评价指标.

### 3.3 实验结果与分析

本次实验是在 VC++6.0 下进行的,实验中每个算法独立运行 30 次,然后取其平均值.在算法中,演化参数设置为:种群规模  $NP=100$ ,最大适应值评估次数  $\text{Max\_eval}=25000$ ,比例因子  $F=0.5$ ,选择控制参数  $\lambda=0.1$ ,正交设计指数  $J=2$ ,对于 ZDT1、ZDT2、ZDT3,  $Q=29$ ;产生网格时归档种群的规模  $N_F=100$ ,交叉概率  $CR=0.9$ .文中把 Paε-ODEMO 算法的运行结果和 NSGA-II, SPEA2, INSGA-II<sup>[19]</sup>, AEPSo<sup>[20]</sup>, ε-DEMO, NPCA<sup>[21]</sup>, PBFO<sup>[22]</sup>, DEMO<sup>[23]</sup>, ε-DEMO<sup>[24]</sup> 进行了比较,用以验证 Paε-ODEMO 算法的性能.表中每个算法的上行表示算法的均值,下行表示算法运行所获得的方差.

从表 1 可以看出, Paε-ODEMO 具有很好的收敛性,其收敛性在所有的测试用例上都是最优的.在 ZDT6 和 DTLZ1 上, NSGA-II 和 SPEA2 这两个算法容易陷入到局部 Pareto 前沿. NSGA-II 算法采用拥挤排序来排序两个种群, INSGA-II 通过使用循环拥挤排序的策略后也改善了其收敛性, DEMO 收敛性较差,但是通过 ε 占优方法后其收敛性明显有所改善.但是与 Paε-ODEMO 相比,在所有的测试用例上其收敛性都相差很多,这表明 Paε-ODEMO 能够很好地收敛到真正的 Pareto 前沿.通过算法独立运行 30 次的方差可以看出,算法同时也具有较好的稳定性.

从表 2 可以看出,本算法在多样性评价方面也达到了不错的效果,在 ZDT1 测试用例上,本算法仅次于 INSGA-II,在 DTLZ1 上本算法多样性处于中间水平.除此之外,在其他测试用例上本算法都是最优的.在多样性的维持方面,算法采用了等间隔的 ε,这就确保了任意两个解直接的距离固定在一个较小的范围内,这样就可以使得所有的非支配解集能够较好地分布在解空间中.但是对于不连续的问题,比如 ZDT3,它将解空间分成了几个间断的部分,虽然 ε 网格依然是等距的,但是由于它的不连续性使得所求得的解在各自间断的连续范围内有较好的分布性,但是断与断之间的解由于距离较远,从而影响了整体的多样性值.此时如果将 ε 值扩大一倍,这样可以使得断内解直接的距离,与断间边界点之间的距离差值缩小.通过实验发现,对于 ZDT3 用例,当 ε 扩大一倍后,其多样性值可以达到 0.4.通过测试用例 ZDT3,也可以发

现此多样性评价指标也存在着一一定的缺陷. 同时通过方差分析可以看出,算法具有良好的稳定性.

表 1 各算法收敛性分析比较(加粗表示获得的最好解)

Table 1 Comparison on convergence of each algorithm(best if bolded)

测试函数	ZDT1	ZDT2	ZDT3	ZDT6	DTLZ1
NSGA-II	0.000 894 0	0.000 824 0	0.043 411 0.000 042	7.806 80 0.001 667	3.759 97 1.892 55
INSGA-II	0.000 57 0	0.000 27 0	0.003 30 0.000 28		
SPEA2	0.023 285 0	0.167 62 0.000 815	0.184 09 0	0.232 55 0.004 945	3.235 731 1.382 36
AEPSO	0.001 00 0	0.000 78 0.000 032	0.004 62 0.000 26		
DEMO	0.005 54 0.000 692	0.143 45 0.012 37	0.079 51 0.009 22	0.574 03 0.029 34	0.063 61 0.112 29
NPCA *	0.000 236 4.88e-10	0.000 442 8 4.09e-8	0.000 231 3.08e-10	0.001 752 1.29e-7	0.004 036 1.8e-7
PBFO	0.017 0 0.001 8	0.005 8 0.000 9	0.005 8 0.000 9	0.001 4 0.001 2	
$\varepsilon$ -DEMO	0.004 76 0.005 18	0.016 82 0.009 11	0.008 6 0.000 807	0.543 72 0.614 89	0.040 365 0.023 64
Pa $\varepsilon$ -ODEMO	<b>0.000 187</b> 0.000 009	<b>0.000 195</b> 0.000 013	<b>0.000 226</b> 0.000 021	<b>0.001 237</b> 0.000 451	<b>0.001 911</b> 0.000 641

表 2 各算法多样性分析比较(加粗表示获得的最好解)

Table 2 Comparison on diversity of each algorithm(best if bolded)

测试函数	ZDT1	ZDT2	ZDT3	ZDT6	DTLZ1
NSGA-II	0.463 293 0.041 462	0.435 112 0.024 607	0.575 606 0.005 078	0.644 477 0.035 042	0.950 23 0.265 4
INSGA-II	0.240 73 0.000 17	0.400 57 0.159 38	0.569 63 0.001 36		
SPEA2	0.472 254 0.097 072	0.473 808 0.093 900	0.606 826 0.191 406	0.670 549 0.077 009	0.850 766 5 0.236 54
AEPSO	0.557 43 0.001 42	0.519 01 0.000 95	0.551 82 0.000 87		
DEMO	0.336 48 0.037 44	0.682 99 0.040 95	0.706 69 0.053 65	0.862 5 0.044 53	0.463 83 0.074 82
NPCA *	0.402 266 0.000 552	0.511 157 0.002 101	0.580 019 0.000 422	0.463 253 0.002 104	
$\varepsilon$ -DEMO	0.321 87 0.022 62	0.365 93 0.034 7	0.485 75 0.027 48	0.518 63 0.277 61	<b>0.434 65</b> 0.083 16
Pa $\varepsilon$ -ODEMO	<b>0.288 694</b> 0.003 50	<b>0.304 138</b> 0.028 41	<b>0.484 418</b> 0.027 53	<b>0.248 236</b> 0.001 54	0.520 197 0.061 2

综上所述,从收敛性和多样性两个角度考虑,显然算法都取得了不错的效果. 但是 Zitzler 等人<sup>[25]</sup>指出仅采用数值评价标准并不能完全反应算法性能的优劣. 为此,文中将程序所得的解在 Matlab 上进行了图形化的显示,以更直观的方式来了解算法的性能. 图 1~图 5 为 Pa $\varepsilon$ -ODEMO 一次运行所绘制的 Pareto 前沿,并与真正的 Pareto 前沿进行了比较.

通过 ZDT1、ZDT2、ZDT3、ZDT6、DTLZ1 的运行图像(图 1~图 5)可以看出,文中提出的 Pa $\varepsilon$ -ODEMO 能够很好地逼近真实的 Pareto 前沿,并且能够完全且均匀地覆盖整个 Pareto 前沿. 也就是说,本文提出的算法在收敛性和多样性上都有较好的效果,它是有效的. 值得注意的是,在 DTLZ1 中,设定的种群规模只有 100 个,对于一个三目标优化问题,它的种群规模是比较小的,在我们建立网格的时候可能会出现那种与大众个体有很大差异的个体. 所以在第一次建立网格的时候有一个剔除差异性解的过程. 剔除差异性个体的过程是通过计算种群中每个个体到其他所有个体的欧式距离的平均值. 对于那些平均值较大众个体差别很大的个体,算法将其剔除.

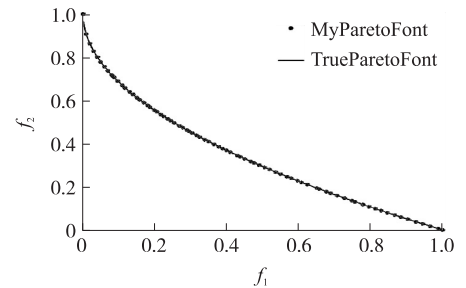


图 1 ZDT1 运行结果

Fig. 1 Pareto front of ZDT1 at  $D=30$



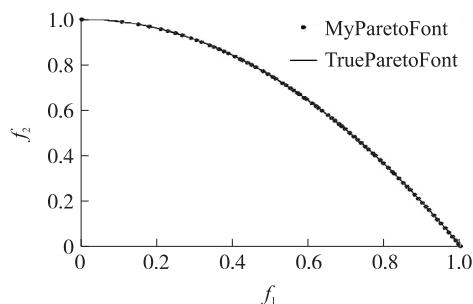


图2 ZDT2 运行结果

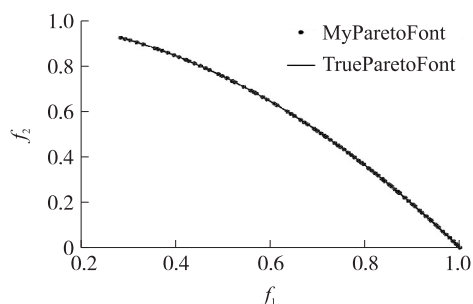
Fig. 2 Pareto front of ZDT2 at  $D=30$ 

图4 ZDT6 运行结果

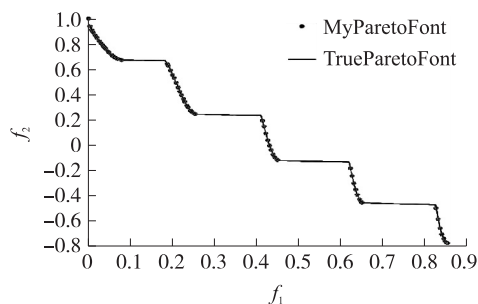
Fig. 4 Pareto front of ZDT6 at  $D=10$ 

图3 ZDT3 运行结果

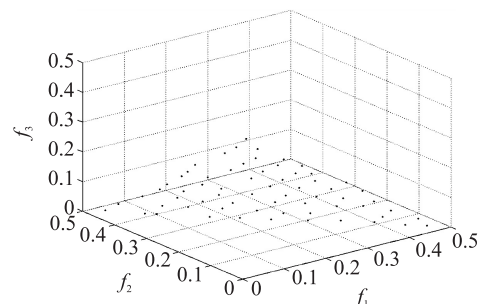
Fig. 3 Pareto front of ZDT3 at  $D=30$ 

图5 DTLZ1 运行结果

Fig. 5 Pareto front of ZTLZ1  $D=12$ 

## 4 结束语

本文提出了一种简单的基于自适应  $\varepsilon$  占优的正交多目标差分演化算法. 新算法的主要改进之处在于使用了自适应的  $\varepsilon$  占优来更新 Archive 种群, 以自适应的方式维持种群的多样性、分布性.

通过 5 个标准测试用例对 Pa $\varepsilon$ -ODEMO 进行测试, 并与 NSGA-II、SPEA2、INSGA-II、AEPSo、DEMO、NPCA\* 和  $\varepsilon$ -DEMO 算法进行了比较, 实验结果表明 Pa $\varepsilon$ -ODEMO 具有很好的收敛性和多样性, 且算法稳定性好.

通过算法运行图像, 可以看出, 在切线斜率趋于 0 或者  $\infty$  的时候, 算法取点是比较稀疏的. 作者今后的工作将致力于调整  $\varepsilon$  网格的密集度, 也就是说  $\varepsilon$  网格将不再是等间距的, 在切线斜率趋于 0 或者  $\infty$  时,  $\varepsilon$  网格将会很密集, 其他的地方  $\varepsilon$  网格将会相对稀疏.

## [参考文献]

- [1] 安伟刚. 多目标优化方法研究及其工程应用[D]. 西安: 西北工业大学航空学院, 2005.
- [2] Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms[C]//Proceedings of the 1st International Conference on Genetic Algorithms. USA: Pittsburgh, 1986(2): 93-100.
- [3] Srinivas N, Kalyanmoy D. Multi-objective optimization using the nondominated sorting in genetic algorithms[J]. Evolution Computation, 1994, 2(3): 221-248.
- [4] Tadahiko M, Hisao I. MOGA: multi-objective genetic algorithms[C]//Proceedings of the 2nd IEEE International Conference on Evolutionary Computing. Perth, Australia: IEEE Press, 1995(3): 289-294.
- [5] Jeffrey H, Nicholas N. Multi-objective optimization using the niched pareto genetic algorithm, technical report 93005[R]. Urbana, Illinois, USA: University of Illinois at Urbana Champaign, 1993.
- [6] Joshua D K, David W C. The pareto archive evolutionary strategy: a new baseline algorithm for multiobjective optimization[C]//Proceedings of the 1999 Congress on Evolutionary Computation. Washington D C, US: IEEE Press, 1999: 98-105.
- [7] Eckart Z, Lothar T. An evolutionary algorithm for multi-objective optimization; the strength pareto approach, technical report 43[R]. Zurich, Switzerland: Swiss Federal Institute of Echnology. 1998: 411-436.



- [8] Kalyanmoy D, Samir A, Amrit P, et al. A fast elitist non-dominate sorting genetic algorithm for multi-objective optimization: NSGA-II[J]. Transactions on Evolutionary Computation, 2002, 6(2): 182–197.
- [9] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm, technical report TIK-Report 103[R]. Swiss: Swiss Federal Institute of Technology Zurich(ETH), 2001.
- [10] 王宇平, 焦永昌, 张福顺. 解多目标优化的均匀正交遗传算法[J]. 系统工程学报, 2003, 13: 481–486.
- [11] 曾三友, 魏巍, 康立山. 基于正交设计的多目标演化算法[J]. 计算机学报, 2005, 28(7): 1 153–1 162.
- [12] 关世华, 寇纪淞, 李敏强. 基于  $\varepsilon$ -约束方法的 Lagrangian 多目标协同进化算法[J]. 系统工程与电子技术, 2002, 24(9): 33–37.
- [13] Shi C, Li Q Y, Shi Z Z. A quick multi-objective evolutionary algorithm based on domination tree[J]. Journal Software, 2007, 18(3): 505–516.
- [14] Gong W, Cai Z, Ling C. ODE: a fast and robust differential evolution based on orthogonal [C]//LNAI 4304: pROC of Advances in Artificial Intelligence. Berlin: Springer, 2006: 709–718.
- [15] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. 软件学报, 2009, 2(20): 271–289.
- [16] 龚文引, 蔡之华. 基于  $\varepsilon$  占优的正交多目标差分演化算法研究[J]. 计算机研究与发展, 2009(4): 655–666.
- [17] Leung Y, Wang Y. An orthogonal genetic algorithm with quantization for global numerical optimization[J]. IEEE Transaction on Evolutionary Computation, 2001, 5(1): 41–53.
- [18] Gong W Y, Cai Z H. An improved multi-objective differential evolution based on Pareto adaptive  $\varepsilon$ -dominance and orthogonal design[J]. European Journal of Operational Research, 2009, 198: 576–601.
- [19] 罗辞勇, 陈民铀, 张聪誉. 采用循环拥挤排序策略的改进 NSGA-II 算法[J]. 控制与决策, 2010, 25(2): 227–231.
- [20] 陈民铀, 张聪誉, 罗辞勇. 自适应进化多目标粒子群算法[J]. 控制与决策, 2009, 24(12): 1 851–1 855.
- [21] 贺群, 程格, 安军辉, 等. 基于 Pareto 的多目标克隆进化算法[J]. 计算机科学, 2012, 39(6A): 489–492.
- [22] 杨尚军, 王社伟, 陶军, 等. 基于混合细菌觅食算法的多目标优化方法[J]. 计算机仿真, 2012, 29(6): 218–222.
- [23] Robić T, Filipič B. DEMO: Differential evolution for multiobjective optimization [C]//LNCS 3410: Proc of EMO'05. Berlin: Springer, 2005: 520–533.
- [24] Deb K, Mohan M, Mishra S. Evaluating the  $\varepsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions[J]. Evolutionary Computation, 2005, 13(4): 501–525.
- [25] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multi-objective optimizer: an analysis and review[J]. IEEE Trans on Evolutionary Computation, 2003, 7(2): 117–132.

[责任编辑: 黄 敏]

(上接第 118 页)

- [9] Liu B, Hsu W, Ma Y. Integrating classification and association rule mining [C]//Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98). New York, USA: AAAI, 1998: 80–86.
- [10] Silva A, Meira Jr W, Zaki M J. Structural correlation pattern mining for large graphs [C]//Proceedings of the Eighth Workshop on Mining and Learning with Graphs. USA: ACM, 2010: 119–126.
- [11] 赵建邦, 董安国, 高琳. 一种用于生物网络数据的频繁模式挖掘算法[J]. 电子学报, 2010, 38(8): 1 803–1 807.
- [12] 丁悦, 张阳, 李战怀, 等. 图数据挖掘技术的研究与进展[J]. 计算机应用, 2012, 32(1): 182–190.
- [13] 薛冰, 张俊峰, 郑超, 等. 基于分割图集的频繁闭图挖掘算法[J]. 计算机应用研究, 2011, 28(1): 61–64, 68.

[责任编辑: 黄 敏]