

文章编号: 1671-6833(2015)06-0038-09

## 基于分解的多目标进化算法在工程优化中的应用

张春江<sup>1,2</sup>, TAN Kay Chen<sup>2</sup>, 高亮<sup>1</sup>, 吴擎<sup>3</sup>

(1. 华中科技大学机械科学与工程学院, 湖北 武汉 430074; 2. Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583; 3. 华中农业大学工学院, 湖北 武汉 430070)

**摘要:** 将基于分解的多目标进化算法 (Multi-objective Evolutionary Algorithm Based on Decomposition, MOEA/D) 应用于工程优化问题时, 由于各目标函数在数量级及量纲上的不同, 需要对目标函数进行归一化处理。首先, 采用一种自适应  $\varepsilon$  约束差分进化算法 ( $\varepsilon$  Constrained Differential Evolution,  $\varepsilon$ DE) 寻找各个目标在 Pareto 前沿上的最大值和最小值, 利用这些值对各目标进行归一化处理; 然后, 用 MOEA/D 进行求解, 并在算法中加入了自适应  $\varepsilon$  约束处理技术; 最后, 采用一个标准测试问题和一个焊接梁设计优化问题对该算法进行测试, 并与其他两种归一化方法进行了比较。根据提出的方法, MOEA/D 能对 Pareto 前沿的一端进行集中优化, 因而能处理一些 Pareto 前沿两端难以优化的问题。

**关键词:** 多目标进化算法; MOEA/D; 归一化; 工程优化; 差分进化;  $\varepsilon$  约束处理

中图分类号: TP183

文献标志码: A

doi: 10.3969/j.issn.1671-6833.2015.06.008

### 0 引言

多目标优化问题 (Multi-Objective Optimization Problems, MOPs) 普遍存在于工程应用与科学研究中, 这些问题的各个目标往往相互冲突, 在对其优化时, 需要对各个目标进行折衷处理。MOPs 往往没有唯一的全局最优解, 而有一个 Pareto 最优解集 (Pareto Optimal Set, POS)。采用传统的数学规划方法求解 MOPs 时, 如权重法、目标规划法等, 一次运行只能求得一个解, 而多目标进化算法 (Multi-Objective Evolutionary Algorithm, MOEA) 因其内在的并行性, 能一次性求得一组 Pareto 最优解。自从 Schatter 和 David<sup>[1]</sup> 在 1985 年首次提出用于多目标优化的向量评估遗传算法 (Vector Evaluated Genetic Algorithm) 以来, MOEA 得到了学者的广泛关注。同时, 许多学者提出了优秀的 MOEA 算法, 如 Zitzler 等<sup>[2]</sup> 提出了 SAPE-II; Deb<sup>[3]</sup> 提出了 NSGA-II; Bader 和 Zitzler<sup>[4]</sup> 提出了基于评估指标的 HypE; 2007 年 Zhang 等<sup>[5]</sup> 提出

了一种全新的基于分解的 MOEA 算法 (Multi-Objective Evolutionary Algorithm based on Decomposition, MOEA/D)。该算法结合传统的数学规划法, 首先将多目标优化问题转发为众多单目标优化问题, 然后采用进化算子同时优化这些单目标优化问题, 最终获得一组 Pareto 最优解。与大多数 MOEA 算法相比, MOEA/D 被证明有更快的求解速度, 能获得分布性及收敛性更好的解集。在 2009 年的 IEEE 进化计算大会中的多目标优化竞赛中, MOEA/D 荣获了第一名<sup>[6]</sup>。近年来, 许多学者对 MOEA/D 做出了很多改进, 例如, Palmers 等<sup>[7]</sup> 让 MOEA/D 分解的每个子问题记录多个解; Qi 等<sup>[8]</sup> 为 MOEA/D 提出了一种自适应权重调整方法; Li 等<sup>[9]</sup> 为 MOEA/D 提出了一种自适应算子选择方法; 周爱民等<sup>[10]</sup> 提出了一种基于混合高斯模型的 MOEA/D。目前, 该算法也被成功应用于天线设计<sup>[11]</sup>、电力系统经济调配<sup>[12]</sup>、复杂网络聚类<sup>[13]</sup>、无线传感网络优化<sup>[14]</sup> 等应用领域。

多目标优化标准测试问题的 Pareto 最优前沿

收稿日期: 2015-07-28; 修订日期: 2015-08-19

基金项目: 国家自然科学基金资助项目 (51421062, 51435009); 湖北省自然科学基金资助项目 (2015CFB528); 国家留学基金管理委员会资助项目 (201406160084)。

作者简介: 张春江 (1989—), 男, 湖南醴陵人, 华中科技大学博士研究生, 主要研究方向为进化计算及其在约束优化和多目标优化方面的应用。

通信作者: TAN Kay Chen (1971—), 男, 马来西亚人, 新加坡国立大学副教授, IEEE 会士, 主要研究智能计算及其在多目标优化、车间调度及数据挖掘等方面的应用, E-mail: eletankc@nus.edu.sg。

(Pareto Optimal Front, POF) 的各目标值往往在  $[0, 1]$  区间, MOEA/D 在这些问题上表现良好. 但现实中的工程优化问题各目标函数往往有不同的单位量度, 并且其函数值常在不同的数量级上. 当直接应用 MOEA/D 求解时, 会造成极差的解集分布性<sup>[12]</sup>, 因此, 需对各目标函数进行归一化处理. Zhang 等<sup>[5]</sup> 利用当前种群的信息来归一化, 但该方法使得原目标函数动态变化, 并在某些目标难以优化的情况下, 该方法也会失效. Zhu 等<sup>[12]</sup> 采用目标函数的最大和最小值来归一化, 在某些情况下, 该方法不适合 MOEA/D.

笔者旨在提供一种新的归一化方法, 以便将 MOEA/D 应用于工程优化问题. 另外, 笔者采用一种自适应  $\varepsilon$  约束处理方法来处理工程优化问题中的约束.

## 1 问题定义

不失一般性, 工程优化中的多目标优化问题通常可以采用以下数学模型加以定义:

$$\begin{cases} \min f(X) = (f_1(X), f_2(X), \dots, f_m(X)). \\ \text{s. t. } \begin{cases} g_i(X) \leq 0 \quad (i=1, 2, \dots, k); \\ h_l(X) = 0 \quad (l=1, 2, \dots, l). \end{cases} \end{cases} \quad (1)$$

该问题有  $m$  个优化目标, 决策向量  $X = (x_1, x_2, \dots, x_n)$ , 有  $k$  个不等式约束及  $l$  个等式约束, 最大化问题可采用下式转化为最小化问题.

$$\max f_i(X) = -\min(-f_i(X)). \quad (2)$$

### 1.1 Pareto 最优解

MOPs 中的最优解通常被称为 Pareto 最优解 (Pareto Optimal Solution). 对于式 (1) 中的 MOP, 其 Pareto 最优解  $X^*$  定义如下: 对于  $X^* \in \Omega$  (表示满足 (1) 中所有约束的可行集), 如果不存在其他任何  $X' \in \Omega$  使得  $f_j(X') \geq f_j(X^*)$  ( $j=1, 2, \dots, m$ ) 同时成立, 且至少有一个严格不等式成立, 则称  $X^*$  是  $\min f(X)$  的 Pareto 最优解.

### 1.2 Pareto 最优解集

MOPs 的 Pareto 最优解往往不止一个, 而存在一组相互制约的最优解集, 被称为 Pareto 最优解集, 其定义如下:

$$\begin{aligned} P^* &= \{X^*\} = \{X \in \Omega \mid \neg \exists X' \in \Omega, \\ &f_j(X') \leq f_j(X), X' \neq X, j=1, \dots, m\}. \end{aligned} \quad (3)$$

Pareto 最优解集是所有 Pareto 最优解的集合.

### 1.3 Pareto 最优前沿

Pareto 最优前沿是 Pareto 最优解集在目标函数空间的映射, 定义如式 (4) 所示:

$$\begin{aligned} PF^* &= \{f(X)\} = \{f(X) = \\ &(f_1(X), f_2(X), \dots, f_m(X)) \mid X \in P^*\}. \end{aligned} \quad (4)$$

## 2 基于分解的多目标进化算法 (MOEA/D)

MOEA/D 是一种全新的多目标优化框架, 在该框架中, 多目标优化问题被转化为一系列单目标优化子问题, 然后利用一定数量相邻问题的信息, 采用进化算法对这些子问题同时进行优化, 因为每一个单目标优化问题的最优解对应于 Pareto 最优前沿上的一个解, 最终能求得一组 Pareto 最优解. 由于分解操作的存在, 该方法在保持解的分散性方面有着很大优势, 而通过分析相邻问题的信息来优化, 能避免陷入局部最优. 与 NSGA-II 相比, MOEA/D 能在更快时间内获得分散性更好并更精确的 Pareto 近优解集<sup>[5]</sup>.

较为常用的分解策略是切比雪夫 (Tchebycheff) 分解法. 该方法对 Pareto 最优前沿的形状不敏感. 笔者也采用该策略将 MOP 问题  $\min f(X)$  转化为  $\min g_j^t(X \mid \lambda^j, z^*)$ ,  $j=1, 2, \dots, N$ , 如下式所示:

$$g_j^t(X \mid \lambda^j, z^*) = \max_{1 \leq i \leq m} \{\lambda_i^j |f_i(X) - z_i^*|\}. \quad (5)$$

式中:  $\lambda^j = (\lambda_1^j, \lambda_2^j, \dots, \lambda_m^j)^T$  为相应的权重向量, 并满足条件  $0 \leq \lambda_i^j \leq 1$ ,  $\sum \lambda_i^j = 1$ ,  $\lambda^j$  的设定算法见文献 [5];  $z^* = (z_1^*, z_2^*, \dots, z_m^*)^T$  为参考点,  $z_i^* = \min\{f_i(X)\}$ . 可以证明, 每一个子问题的最优解都是 MOP 的 Pareto 最优解, 且 MOP 的每一个 Pareto 最优解都对应一个切比雪夫子问题.

MOEA/D 中, 与子问题  $\min g_j^t(X \mid \lambda^j, z^*)$  相似度最大的  $k$  个问题被定义为邻域子问题, 采用权重向量之间的欧几里德距离来定义问题间的相似度, 邻域子问题的个体被用来充当构建子代个体的父本. 个体进化时, 常采用 SBX 算子或差分进化 (DE) 算子<sup>[15]</sup>, 笔者采用 DE 算子.

## 3 归一化方法

MOEA/D 适用于求解各目标函数的范围和量纲一致的问题. 将 MOEA/D 直接应用于各目标值量纲和数量级不同的工程优化问题时, 目标函数会向着数值较大的目标进化, 往往只能获得分散性较差的 Pareto 近优前沿, 因而需要归一化处理. Zhu 等<sup>[12]</sup> 在采用 MOEA/D 优化电力环境经济调度问题时, 采用以下方法进行归一化.

$$f_i^{\text{trans}} = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (6)$$

式中:  $f_i^{\text{trans}}$  表示归一化后的第  $i$  个目标函数;  $f_i$  表

示原始的第  $i$  个目标函数值;  $f_i^{\min}$  和  $f_i^{\max}$  分别表示第  $i$  个目标函数的最小值和最大值. 有时某个  $f_i^{\max}$  会非常大, 甚至存在  $f_i^{\max}$  并不存在的情况, 这时候该归一化方法就会失效.

对于 MOEA/D, 最理想的归一化方法如下所示:

$$f_i^{\text{trans}} = \frac{f_i - f_i^{\min}}{f_i^{\text{nad}} - f_i^{\min}} \quad (7)$$

式中:  $f_i^{\text{nad}}$  表示 Pareto 最优前沿上第  $i$  个目标函数的最大值. 这种方法可使每个目标函数在 Pareto 最优前沿上的值都在  $[0, 1]$  区间. 采用式 (6) 进行归一化时, 转化后的第  $i$  目标函数在 Pareto 最优前沿上的最大值  $f_i^{\text{Tnad}}$  如下所示:

$$f_i^{\text{Tnad}} = \frac{f_i^{\text{nad}} - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (8)$$

当  $f_i^{\max}$  与  $f_i^{\text{nad}}$  相差很大时  $f_i^{\text{Tnad}}$  会远小于 1; 并且由于每个目标函数的  $f_i^{\text{Tnad}}$  很可能各不相同, 不能保证对各目标的均衡优化而得到分布性较差的 Pareto 近优前沿.

Zhang 等<sup>[5]</sup> 认为很难求解  $f^{\min}$  和  $f^{\text{nad}}$ , 并提出了一种自适应归一化方法, 如式 (9) 所示:

$$f_i^{\text{trans}} = \frac{f_i - z_i^*}{\bar{f}_i^{\text{nad}} - z_i^*} \quad (9)$$

式中:  $\bar{f}_i^{\text{nad}}$  表示当前种群中第  $i$  个目标函数的最大值;  $z_i^*$  的定义如上文所示. 一般情况下  $\bar{f}_i^{\text{nad}}$  会逐渐逼近  $f_i^{\text{nad}}$ ,  $z_i^*$  也会逼近  $f_i^{\min}$ , 因而这种方法有效. 但这种方法有两个缺陷: 其一, 对于非常难的问题  $\bar{f}_i^{\text{nad}}$  很难逼近  $f_i^{\text{nad}}$ , 甚至存在  $\bar{f}_i^{\text{nad}}$  不能得到改善的情况,  $z_i^*$  也可能不会逼近  $f_i^{\min}$ . 此时, 会造成  $f_i^{\text{Tnad}}$  各不相同, 从而对各目标的优化不均衡而得到分布性较差的解. 其二, 当采用这种方法时, 相当于子问题一直在变化, 很显然, 这种变化会对优化过程造成一定的困难, 如收敛速度与解的精度都会受到影响.

笔者将采用式 (7) 的方法来进行归一化. 现在的问题便转化为了怎样求  $f^{\min}$  和  $f^{\text{nad}}$ . 对于 MOP 存在以下定理: 在一般情况下, 对于有  $m$  个目标的 MOP, 它的 Pareto 最优解无论在解空间还是目标空间都是一个分段连续的  $m-1$  维流形. 这说明, 当  $m=2$  (双目标优化问题) 时, Pareto 最优前沿是一个分段连续的一维流形, 一般情形下是一条一维曲线, 显然,  $(f_1^{\min}, f_2^{\text{nad}})$  和  $(f_1^{\text{nad}}, f_2^{\min})$  恰好是曲线的两个端点.

对于双目标优化问题, 采用以下方法来求解  $f^{\min}$  和  $f^{\text{nad}}$ . 首先, 采用一种高效的约束优化算法, 即自适应  $\varepsilon$ DE<sup>[16]</sup> 分别优化  $f_1$  和  $f_2$  而求得  $f_1^{\min}$  和  $f_2^{\min}$ . 要求解  $f_1^{\text{nad}}$  和  $f_2^{\text{nad}}$ , 只需采用自适应  $\varepsilon$ DE 求解式 (10) 和式 (11), 可证明这两式的最优值为  $f_1^{\text{nad}}$  和  $f_2^{\text{nad}}$ . 式 (11) 最优解的示意图如图 1 所示. 约束条件  $f_1(X) \leq f_1^{\min}$  实际上只能取到  $f_1(X) = f_1^{\min}$ , 如图中的虚线所示. 在这虚线上  $f_2$  能取到的最小值只能是  $f_2^{\text{nad}}$ . 当虚线与 Pareto 前沿相交时取到该最小值, 此交点也是 Pareto 前沿的一个端点.

$$\begin{cases} \min f_1(X), \\ \text{s. t. } \begin{cases} f_2(X) \leq f_2^{\min}; \\ g_i(X) \leq 0 \quad (i=1, 2, \dots, k); \\ h_i(X) = 0 \quad (i=1, 2, \dots, l). \end{cases} \end{cases} \quad (10)$$

$$\begin{cases} \min f_2(X), \\ \text{s. t. } \begin{cases} f_1(X) \leq f_1^{\min}; \\ g_i(X) \leq 0 \quad (i=1, 2, \dots, k); \\ h_i(X) = 0 \quad (i=1, 2, \dots, l). \end{cases} \end{cases} \quad (11)$$

与公式 (9) 相比, 公式 (6) 和公式 (7) 的方法要先求解归一化所需的  $f^{\min}$  和  $f^{\max}$  或  $f^{\text{nad}}$ . 但因为单目标优化的计算复杂度要低于多目标优化, 并且单目标优化所需的种群大小也一般小于多目标优化. 另外, 当采用公式 (6) 和公式 (9) 进行归一化时, 不需要更新  $z^*$ , 可节约计算时间. 因而采用公式 (6) 和公式 (9) 的方法并不会增加额外的计算时间, 甚至比公式 (7) 的方法所需时间更少.

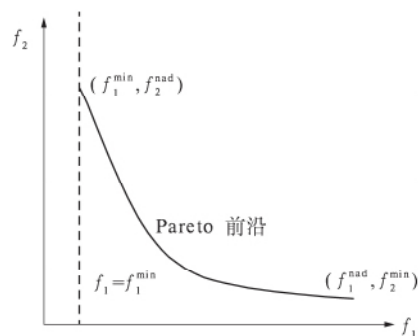


图 1 式 (11) 的最优解示意图

Fig. 1 Illustration of optimal solution for formula (11)

#### 4 自适应 $\varepsilon$ 约束处理及自适应 $\varepsilon$ 差分进化算法

$\varepsilon$  约束处理方法最先由 Takahama 等<sup>[17]</sup> 提出. 该方法用来比较两个候选解的优劣. 在该方法中, 存在一个  $\varepsilon$  值, 当两个解的约束违反量都小于  $\varepsilon$  时, 则根据目标函数值的大小来比较, 否则采用

违反约束的程度来比较. 数学表达式如下:

$$\begin{aligned} & (f(X_1) \quad \varphi(X_1)) \leq_{\varepsilon} (f(X_2) \quad \varphi(X_2)); \\ \Leftrightarrow & \begin{cases} f(X_1) \leq f(X_2) & \text{if } \varphi(X_1) \leq \varphi(X_2) \leq \varepsilon; \\ f(X_1) \leq f(X_2) & \text{if } \varphi(X_1) = \varphi(X_2); \\ \varphi(X_1) \leq \varphi(X_2) & \text{otherwise.} \end{cases} \quad (12) \end{aligned}$$

其中,  $\varphi$  表示约束违反量, 计算方式如下:

$$\varphi(X) = \sum_{i=1}^k \max(0, g_i(X)) + \sum_{j=1}^l \max(|h_j(X)| - \delta, 0). \quad (13)$$

式中:  $\delta$  为等式约束的容许误差值.

Takahama 和 Sakai 将  $\varepsilon$  约束法与差分进化算法相结合, 赢得了 2006 年与 2010 年 IEEE 计算智能大会上约束优化竞赛上的第一名<sup>[18-19]</sup>. 笔者采用 Zhang 等改进的自适应  $\varepsilon$  约束法. Zhang 等<sup>[16]</sup>利用当前种群中的信息而采用一种自适应方式来控制  $\varepsilon$  值. 具体控制方法如式(14)所示:

$$\varepsilon(t) = \begin{cases} 0 & \text{if } \varphi_{\max} > Th \text{ or } \varphi_0 > ap_1 \cdot N \text{ or } t > Tc; \\ ap_2 \cdot \varphi_{\max}, & \text{otherwise;} \end{cases} \quad (14)$$

式中:  $t$  表示迭代次数;  $\varphi_{\max}$  表示种群中约束违反量的最大值;  $\varphi_0$  表示种群中约束违反量为 0 的个体数;  $N$  表示种群大小;  $Tc$  表示迭代次数的阈值;  $Th$  也是阈值;  $ap_1$  和  $ap_2$  是介于 0 和 1 之间的系数. 上式表示, 当种群中  $\varphi_{\max}$  很大或者可行解很多时, 直接让  $\varepsilon$  取 0; 否则, 让  $\varepsilon$  取值为  $\varphi_{\max}$  的  $ap_2$  倍.

DE 算法由 Storn 和 Price<sup>[20]</sup>提出. Zhang 等人将自适应  $\varepsilon$  约束方法与最简单的一种 DE 算法 (DE/rand/1/exp) 结合. 数值试验证明, 该方法具有求解速度快、精度高、鲁棒性强等优点. 用来求解  $f^{\text{nad}}$  的  $\varepsilon$ DE 算法的伪代码如图 2 所示.

## 5 算法步骤

将 MOEA/D 应用于工程优化问题的具体步骤如下:

(1) 参数设置. 设置自适应  $\varepsilon$ DE 算法的参数以及 MOEA/D 的参数.

(2) 采用自适应  $\varepsilon$ DE 算法求解  $f^{\min}$  和  $f^{\text{nad}}$ .

(3) 初始化 MOEA/D. ① 初始化权重向量  $\lambda^1, \dots, \lambda^N$ , 具体方法见参考文献[5], 计算这些权重向量之间的欧氏距离, 为每个权重向量找出  $T$  个与其距离最近的向量, 从而构成  $B(i) = \{i_1, \dots, i_T\}$ ; ② 采用均匀分布在定义域中生成种群大小为  $N$  的种群  $X$ , 并计算初始种群的目标函数与约束违反量; ③ 迭代次数初始化:  $Gen = 0$ .

```

ε DE/rand/1/exp
{
    for (i=1; i≤N; i=i+1) { //生成初始种群
        for (j=1; j≤n; j=j+1) {
            Xji = Lj + u(0,1)(Uj - Lj) //均匀分布初始化
        }
        计算 f(X) 和 φ(X)
    }
    for (t=1; t≤Tmax; t=t+1) { //终止条件
        for (i=1; i≤N; i=i+1) {
            ε(t) = { 0, if φmax > Th or φ0 > ap1 * N or t > Tc
                    ap2 * φmax, otherwise
            随机从 [1, N] 中选择 p1, p2, p3
            s.t. p1 ≠ p2 ≠ p3 ≠ i;
            Xnew = Xi;
            从 [1, n] 中随机选择 j;
            k=1;
            do { //交叉:
                Xjnew = Xjp1 + F(Xjp2 - Xjp3) //突变
                if (Xjnew < Lk || Xjnew > Uk) //边界处理
                    Xjnew = Xji;
                j = (j+1) % n;
                k++;
            } while (k ≤ n && u(0,1) < CR);
            计算 f(Xnew) 和 φ(Xnew)
            if ((f(Xnew), φ(Xnew)) < ε (f(Xi), φ(Xi))) //ε 比较
                {Z = Xnew; f(Z) = f(Xnew); φ(Z) = φ(Xnew); }
            else
                {Z = Xi; f(Z) = f(Xi); φ(Z) = φ(Xi); }
        }
        for (i=1; i≤N; i=i+1) { //更新种群
            Xi = Zi; f(Xi) = f(Zi); φ(Xi) = φ(Zi);
        }
    }
    输出最优解
}

```

图2 自适应  $\varepsilon$ DE 的伪代码

Fig. 2 Pseudo-code of adaptive  $\varepsilon$ DE

(4) 算法迭代更新.

a) 根据式(14)定义  $\varepsilon$  值. 对每一个个体  $i$  进行如下操作.

b) 根据下式选择重组父本的范围:

$$P = \begin{cases} B(i), & \text{if rand} < \delta; \\ \{1, 2, \dots, N\}, & \text{otherwise.} \end{cases} \quad (15)$$

其中,  $\text{rand}$  为服从均匀  $[0, 1]$  分布的随机数.

c) 重组. 令  $r_1 = i$ , 从  $P$  中随机选择  $r_2$  和  $r_3$ . 根据 DE 算子重组出新解  $\bar{Y} = (\bar{Y}_1, \dots, \bar{Y}_n)$ . 每一维  $\bar{Y}_k$  如下式所示:

$$\bar{Y}_k = \begin{cases} X_k^{r_1} + F(X_k^{r_2} - X_k^{r_3}), & \text{if rand} < CR; \\ X_k^{r_1}, & \text{otherwise.} \end{cases} \quad (16)$$

其中  $F$  和  $CR$  是两个参数. 再通过一个多项式突变算子得到解  $Y$ . 具体见文献[15].

d) 修复. 如果新解  $Y$  的某个元素值超出边界, 则赋予该元素一个边界内随机值. 计算新解  $Y$  的函数值  $f(Y)$  及约束违反量  $\varphi(Y)$ .

e) 更新解. 随机从  $P$  中选择一个值  $j$ . 根据式(5)计算  $g_j^{\text{fe}}(X^j | \lambda^j, z^*)$  和  $g_j^{\text{fe}}(Y | \lambda^j, z^*)$ . 注意, 此处需采用归一化方法, 故式(5)需更改如下:

$g_j^{te}(X|\lambda^j) = \max_{1 \leq i \leq m} \{ \lambda_i^j | f_i^{trans}(X) | \} = \max_{1 \leq i \leq m} \left\{ \lambda_i^j \cdot \left| \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \right| \right\}$  然后根据  $\varepsilon$  约束法比较  $(g_j^{te}(X|\lambda^j), \varphi(X^j))$  和  $(g_i^{te}(Y|\lambda^i), \varphi(Y))$ . 如果后者优于前者, 则用  $Y$  取代  $X^j$ .

(5) 终止准则. 如果迭代次数  $gen = Genmax$ , 则停止迭代, 输出结果; 否则  $gen = gen + 1$ , 并返回步骤(4).

## 6 数值实验

为了验证笔者提出归一化方法的有效性, 选取了另两种方法来进行对比, 这两种方法分别采取公式(6)和公式(9)来归一化. 首先选用 CEC2009 中多目标优化竞赛<sup>[21]</sup>中的一个问题 CF6 来比较 3 种归一化方法.

### 6.1 CF6 测试问题

对于 CF6, CEC2009 的标准测试问题的最大函数评估次数为 300 000, 种群大小为 600, 最大迭代次数为 500, 其他参数的设置请参考文献[15]. 在求解  $f^{\min}$ ,  $f^{\max}$  和  $f^{\text{nad}}$  时, 种群大小设置为 40, 最大迭代次数设置为 2 000, 其他参数设置请参考文献[16].

CF6 的两个目标函数范围都在  $[0, 1]$ , 本不需要归一化, 因此, 将原目标函数  $f_1$  乘以 10, 经改造的 CF6 如下所示:

$$\min \begin{cases} f_1 = 10(x_1 + \sum_{j \in J_1} y_j^2); \\ f_2 = (1 - x_1)^2 + \sum_{j \in J_2} y_j^2. \end{cases} \quad (17)$$

其中  $J_1 = \{j | j \text{ 是奇数且 } 2 \leq j \leq n\}$ ,  $J_2 = \{j | j \text{ 是偶数且 } 2 \leq j \leq n\}$ , 且

$$y_j = \begin{cases} x_j - 0.8x_1 \cos(6\pi x_1 + \frac{j\pi}{n}) & \text{if } j \in J_1; \\ x_j - 0.8x_1 \cos(6\pi x_1 + \frac{j\pi}{n}) & \text{if } j \in J_2. \end{cases} \quad (18)$$

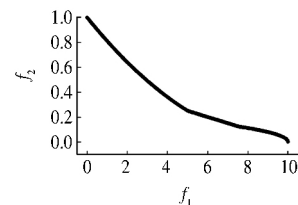
约束条件为

$$\begin{aligned} x_2 - 0.8x_1 \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - \text{sign}(0.5(1 - x_1) - (1 - x_1)2) \sqrt{|0.5(1 - x_1) - (1 - x_1)2|} \geq 0. \end{aligned} \quad (19)$$

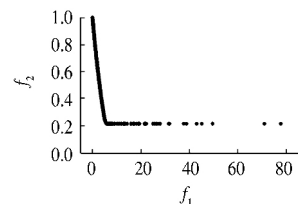
$$\begin{aligned} x_4 - 0.8x_1 \sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - \text{sign}(0.25\sqrt{1 - x_1} - 0.5(1 - x_1)) \sqrt{|0.25\sqrt{1 - x_1} - 0.5(1 - x_1)|} \geq 0. \end{aligned} \quad (20)$$

决策空间为  $[0, 1] \times [-2, 2]^{n-1}$ .

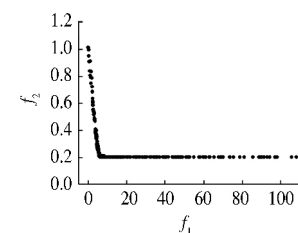
采用公式(6)的归一化方法, 需要先求  $f^{\min}$  和  $f^{\max}$ . 这里采用自适应  $\varepsilon$ DE 算法来求解, 求得  $f^{\min} = [0, 0]$ ,  $f^{\max} = [270.9, 30.9]$ . 采用本文方法, 也就是公式(7)的方法, 需要求  $f^{\text{nad}}$ , 求得  $f^{\text{nad}} = [10, 1]$ . 对于公式(6)的方法, 可以先根据式(8)计算  $f^{\text{nad}} = \left[ \frac{10}{270.9}, \frac{1}{30.9} \right]$ . 因为  $f_1^{\text{nad}}$  与  $f_2^{\text{nad}}$  相差不大, 因此, 可以推断公式(6)的方法与本文方法效果相当. 3 种方法均采用相同的算法参数, 最终求得的解如图 3 所示.



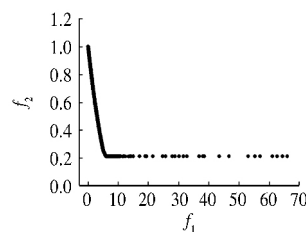
(a) Pareto Front



(b) 公式(6)



(c) 公式(9)



(d) 公式(7)

图 3 对 CF6 各方法 Pareto 前沿的比较

Fig. 3 Pareto fronts of different methods for CF6

图 3 中第一个子图为真实的 Pareto 前沿. 3 种方法都没有求得  $f_2$  的范围在  $[0, 0.2]$  的 Pareto 最优前沿. 公式(6)的方法与本文方法相当, 这是因为采用公式(6)时, 该问题的  $f_1^{\text{nad}}$  与  $f_2^{\text{nad}}$  相差不大; 公式(9)的方法结果较差, 其一为分散性较

差,  $f_2$  的范围在  $[0.4, 1]$  的 Pareto 前沿非常稀疏, 原因在于此问题比较难于优化. 在优化后期该问题的  $z^*$  约为  $[0, 0.2]$ ,  $f_1^{\text{nad}}$  (种群中的最大目标函数值  $f_1^{\text{max}}$ ) 约为  $[110, 1]$ . 此时, 计算  $f_1^{\text{Tnad}}$  为  $\left[ \frac{10-0}{110-0} \frac{1-0.2}{1-0.2} \right] = \left[ \frac{1}{11}, 1 \right]$ , 可见  $f_2^{\text{Tnad}}$  远大于  $f_1^{\text{Tnad}}$ , 因此导致了较差的解集分布性. 公式(9)的结果较差还表现在精度不高. 从图3(c)可见, 公式(9)方法所获的前沿不光滑, 有些解不是非支配解, 这是因为采用公式(9)的方法使子问题变成了动态优化问题, 收敛速度变慢了.

对于该问题, 采用自适应  $\varepsilon$ DE 求解  $f^{\text{min}}$  时, 可获得  $f^{\text{min}} = [0, 0]$ . 但采用 MOEA/D 进行求解时, 却无法求得  $f_2$  的范围在  $[0, 0.2]$  的 Pareto 最优前沿. 在已知  $f^{\text{min}} = [0, 0]$  和  $f^{\text{nad}} = [10, 1]$  的情况下, 可以结合归一化方法让更多的解集中在  $f_2$  的范围为  $[0, 0.2]$  的 Pareto 最优前沿处. 具体方法是在归一化时加大  $f_1^{\text{nad}}$ . 比如, 当分别采用  $f_1^{\text{nad}} = [20, 1]$  和  $[100, 1]$  进行归一化时, 可获得如图4所示的最优前沿. 很显然, 当  $f_1^{\text{nad}}$  增加的越大时, 对 Pareto 前沿的下端的优化就越集中, 因而图4中右图 Pareto 前沿的下端要优于左图, 而上端却较左图稀疏.

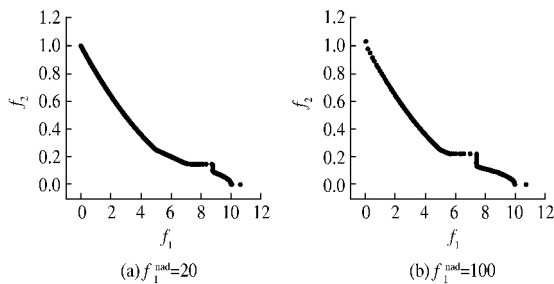


图4 增加  $f_1^{\text{nad}}$  时所获得 Pareto 前沿

Fig.4 Pareto fronts obtained by increasing  $f_1^{\text{nad}}$

下面选取逆向世代距离 (Inverted Generation Distance, IGD) 进一步比较这几种方法. IGD 的定义如下式所示:

$$\begin{cases} \text{IGD}(\text{PF}, \text{PF}^*) = \frac{1}{|\text{PF}|} \sum_{x \in \text{PF}} d(x, \text{PF}^*); \\ d(x, \text{PF}^*) = \min_{y \in \text{PF}^*} \|x - y\|. \end{cases} \quad (21)$$

式中:  $|\text{PF}|$  表示所获 Pareto 前沿的数目;  $\text{PF}^*$  表示理想的 Pareto 最优前沿;  $\|x - y\|$  表示欧几里德距离; IGD 可以同时度量  $\text{PF}$  的收敛性及分布性.

分别采用公式(6)、公式(9)、公式(7)以及

采用公式(7)时将  $f_1^{\text{nad}}$  增大为 100 来归一化处理, 将算法随机运行 30 遍. IGD 的平均值与标准差及每种方法所需的平均运行时间如表 1 所示. 每种方法获得 IGD 做出箱线图如图 5 所示.

由表 1 和图 5 可知, 采用公式(9)的归一化方法所获的 IGD 最大, 公式(6)和公式(7)的方法没有明显的差异, 这之前分析是一致的. 相对而言, 公式(7)方法所获的解 IGD 更集中, 而采用公式(7)方法时, 将  $f_1^{\text{nad}}$  增大为 100 能明显提高算法的性能, 不但能获得更小的 IGD 值, 并且比其他方法更加集中. 这就是因为当  $f_1^{\text{nad}} = 100$  时, 会有大量的解集中到难以优化的 Pareto 前沿附近, 因而能跳出该处的局部最优, 获得更为完整 Pareto 最优前沿, 结果如图 4 所示.

表 1 4 种方法对 CF6 运行 30 次求得的 IGD 平均值、标准差及所需平均时间

Tab.1 IGD's average value, standard error and average time by the four methods running 30 times for CF6

方法	平均值	标准差	运行时间/s
公式(6)	4.22E-02	1.27E-02	154
公式(9)	6.68E-02	1.86E-02	167
公式(7)	4.31E-02	1.32E-02	155
$f_1^{\text{nad}} = 100$	2.81E-02	8.66E-03	153

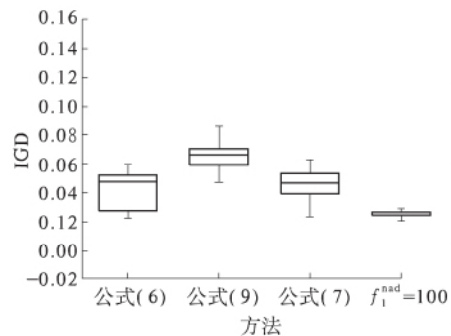


图5 4 种方法对 CF6 运行 30 次的 IGD 的箱线图

Fig.5 Boxplot of four methods for CF6

由表 1 可见, 虽然需要先计算归一化所需的  $f^{\text{min}}$  和  $f^{\text{max}}$  或  $f^{\text{nad}}$ , 但公式(6)和公式(7)不需要更新  $z^*$ , 所以这两种方法反而比公式(9)的方法的计算时间更少.

作出 IGD 为中位数时各方法的 IGD 收敛曲线如图 6 所示. 可见, 公式(9)的方法最差, 波动也较大, 这正是  $f^{\text{nad}}$  和  $z^*$  不断变化的结果. 采用公式(6)和公式(7)求得的 IGD 非常接近. 而采用公式(7)并将  $f_1^{\text{nad}}$  增大为 100 时, 能收敛到更低的 IGD 值. 这是因为有更多的解集中到 Pareto 前沿的下端, 而能跳出局部最优.



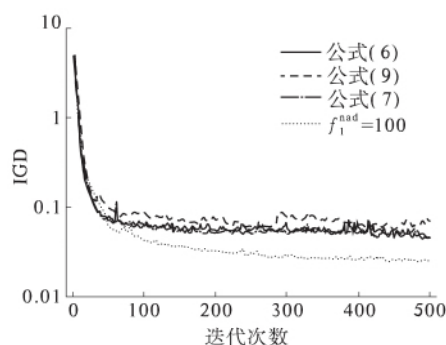


图6 4种方法求解CF6的IGD收敛图

Fig. 6 IGD's converge plots of four algorithms for CF6

## 6.2 焊接梁设计问题

笔者选用焊接梁设计问题<sup>[22]</sup>验证该算法的有效性,其目标是 minimized 制造成本和末端挠度,其约束为在一定的负载下剪切应力和弯曲应力均满足要求。

对于该问题,种群大小设置为100,最大迭代次数为100;自适应 $\varepsilon$ DE的种群大小为20,最大迭代次数为500。

用自适应 $\varepsilon$ DE算法求的 $f^{\min} = [1.8616, 0.00044]$ ,  $f^{\max} = [1.220, 0.07128]$ ,求得 $f^{\text{nad}} = [35.23, 0.0157]$ 。对于公式(6)的方法,计算其 $f^{\text{nad}} = [0.027, 0.215]$ ,可见两个值相差较大。采用3种方法所获得的 Pareto 前沿如图7所示。

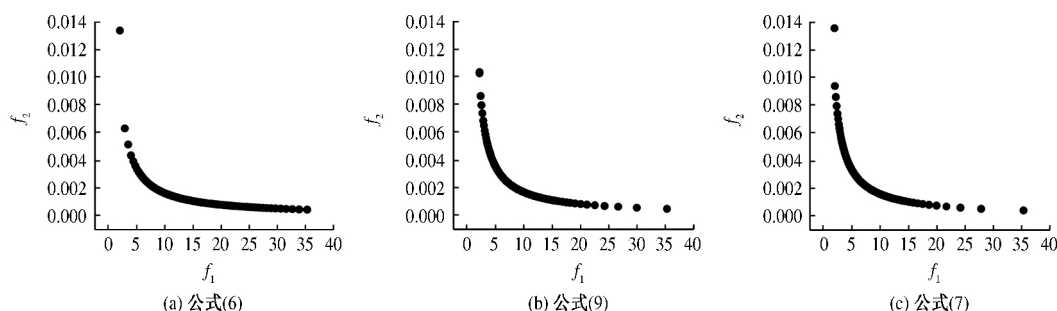
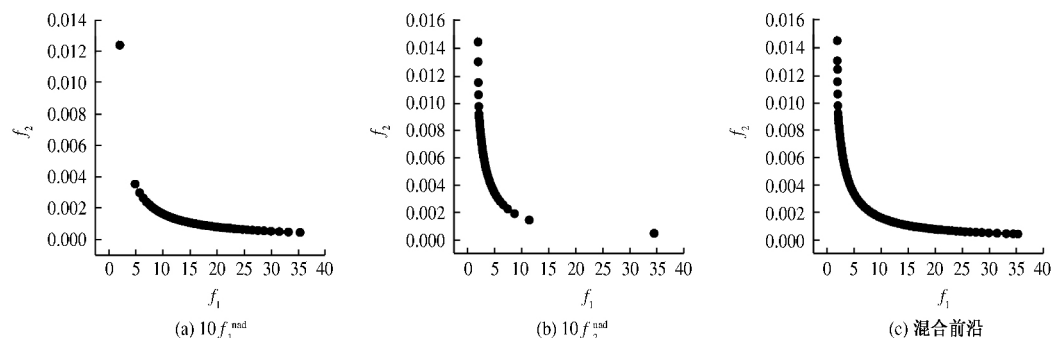


图7 对焊接梁设计问题各方法 Pareto 前沿的比较

Fig. 7 Pareto Fronts obtained by three methods for weld beam design

图8 将 $f_1^{\text{nad}}$ 和 $f_2^{\text{nad}}$ 扩大10倍所得的 Pareto 前沿及混合前沿Fig. 8 Pareto fronts obtained by enlarging  $f_1^{\text{nad}}$  and  $f_2^{\text{nad}}$  to 10 times and the mixed front

从图7可见,对于公式(6)的方法,由于 $f_2^{\text{nad}}$ 远大于 $f_1^{\text{nad}}$ ,所获得 Pareto 前沿在上部的所获得解非常稀疏。不过在 Pareto 前沿的下部所获得的解要优于公式(9)和公式(7)方法。因为该问题易于求解,采用公式(9)的方法与公式(7)的方法,所获得 Pareto 前沿非常接近。

从图7可见,该问题的 Pareto 最优前沿有比较陡峭的尖端和很平缓的尾端。当采用 MOEA/D 求解时,所获的解在尖端和尾端非常稀疏。对于此问题,结合本文的归一化方法提供一种解决思路。对于同一个问题采用两个种群分别求解,对一个种群归一化时增加 $f_1^{\text{nad}}$ ,对另一个种群归一化时增加 $f_2^{\text{nad}}$ ,最后所得解合二为一,就能得到在整个 Pareto 最优前沿上都分布较好的解。这里每个种群采用50个粒子,分别将 $f_1^{\text{nad}}$ 和 $f_2^{\text{nad}}$ 扩大10倍,最终分别获得的 Pareto 前沿以及混合后的 Pareto 前沿如图8所示。

从图8可见,将 $f_1^{\text{nad}}$ 和 $f_2^{\text{nad}}$ 扩大10倍所求的解分别在 Pareto 前沿的一半分布性较好,两者混合后能获得整体上分布性较好的前沿。与图7中公式(7)获得 Pareto 前沿相比,两者有着相同数量的解,但混合前沿的在尖端和尾端的分布性远远好于前者,可见该方法是有效的。

## 7 结束语

为了将 MOEA/D 算法更好地应用于各目标数量及量纲不同的多目标优化问题,笔者提出了一种新的归一化方法,并采用了一种自适应  $\varepsilon$  约束处理方法来处理约束。归一化方法采用一种自适应  $\varepsilon$ DE 算法来求解归一化所需的  $f^{\min}$  和  $f^{\max}$ 。通过数值实例,证明了该方法能获得分布性较好的 Pareto 前沿,并且该方法能克服其他两种归一化方法的缺点。另外,通过灵活运用该归一化方法,MOEA/D 能对 Pareto 前沿的一端集中优化,因而能处理一些 Pareto 前沿的两端比较难以优化的多目标优化问题。

## 参考文献:

- [1] SCHAFFER, DAVID J. Some experiments in machine learning using vector evaluated genetic algorithms [D]. Nashville, TN (USA): Vanderbilt Univ, 1985.
- [2] ZITZLER E, LAUMANN S M, THIELE L. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization [R]. Zurich: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 2001: 19–21.
- [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] BADER J, ZITZLER E. HypE: An algorithm for fast hypervolume-based many-objective optimization [J]. Evolutionary Computation, 2011, 19(1): 45–76.
- [5] ZHANG Qingfu, LI Hui. MOEA/D: A multiobjective evolutionary algorithm based on decomposition [J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712–731.
- [6] ZHANG Qingfu, LIU Wudong, LI Hui. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances [C]//In IEEE Congress on Evolutionary Computation. Trondheim: IEEE Press 2009: 203–208.
- [7] PALMERS P, MCCONAGHY T, STEYAERT M, et al. Massively multitopology sizing of analog integrated circuits [C]//Conference on Design, Automation and Test in Europe, DATE 2009. Nice: IEEE Press 2009: 706–711.
- [8] QI Yutao, MA Xiaoliang, LIU Fang, et al. MOEA/D with adaptive weight adjustment [J]. Evolutionary computation, 2014, 22(2), 231–264.
- [9] LI Ke, CARVALHO A, KWONG S, et al. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(1): 114–130.
- [10] 周爱民, 张青富, 张桂成. 一种基于混合高斯模型的多目标进化算法 [J]. 软件学报, 2014, 25(5): 913–928.
- [11] CARVALHO R D. A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas [J]. IEEE Transactions on Magnetics 2012, 48(2): 803–806.
- [12] ZHU Yongsheng, WANG Jie, QU Boyang. Multi-objective economic emission dispatch considering wind power using evolutionary algorithm based on decomposition [J]. International Journal of Electrical Power & Energy Systems, 2014, 63: 434–445.
- [13] GONG Maoguo, CAI Qing, CHEN Xiaowei, et al. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(1): 82–97.
- [14] KONSTANTINIDIS A, YANG K. Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D [J]. Applied Soft Computing, 2011, 11(6): 4117–4134.
- [15] LI Hui, ZHANG Qingfu. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284–302.
- [16] ZHANG Chunjiang, LIN Qun, GAO Liang. A novel adaptive  $\varepsilon$ -constrained method for constrained problem [C]//In Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems. Singapore: Springer 2015: 573–586.
- [17] TAKAHAMA T, SAKAI S, IWANE N. Constrained optimization by the epsilon constrained hybrid algorithm of particle swarm optimization and genetic algorithm [C]//AI 2005: Advances in Artificial Intelligence. Sydney, Australia: Springer 2005: 389–400.
- [18] TAKAHAMA T, SAKAI S. Constrained optimization by the epsilon constrained differential evolution with gradient-based mutation and feasible elites [C]//IEEE Congress on Evolutionary Computation. Anchorage, BC: IEEE Press 2006: 1–8.
- [19] TAKAHAMA T, SAKAI S. Constrained optimization by the  $\varepsilon$  constrained differential evolution with an Archive and gradient-based mutation [C]//2010 IEEE Congress on Evolutionary Computation (CEC). Barce: IEEE Press 2010: 25–32.



- [20] STORN R, PRICE K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces [J]. *Journal of Global Optimization*, 1997, 11(4): 341–359.
- [21] ZHANG Qingfu, ZHOU Aimin, ZHAO Shizheng, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition [R]. University of Essex, Colchester, UK and Nanyang Technological University, Singapore, 2008: 1–30.
- [22] RAY T, TAI K. An evolutionary algorithm with a multilevel pairing strategy for single and multiobjective optimization [J]. *Foundations of Computing and Decision Sciences*, 2001, 26(1): 75–98.

## Multi-Objective Evolutionary Algorithm Based on Decomposition for Engineering Optimization

ZHANG Chunjiang<sup>1,2</sup>, TAN Kay Chen<sup>2</sup>, GAO Liang<sup>1</sup>, WU qing<sup>3</sup>

(1. School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; 2. Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583; 3. School of Engineering, Huazhong Agricultural University, Wuhan 430070, China)

**Abstract:** In order for effective application of Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) in engineering optimization, normalization of the range of objective values is needed. A self-adaptive  $\varepsilon$  constrained Differential Evolution ( $\varepsilon$ DE) algorithm is proposed to obtain the minimum and maximum values of each objective on the Pareto Front (PF). After normalization, MOEA/D can then be effectively applied. In addition, the self-adaptive  $\varepsilon$  constraint method is combined with MOEA/D for constraint handling. A benchmark problem and a weld bead design problem are used to evaluate the performance of the algorithm against two other normalization methods. One main advantage of the proposed method is the selective concentrated optimization on some regions on the Pareto front which allows handling of problems where regions of Pareto front are difficult to be optimized.

**Key words:** multi-objective evolutionary algorithm; MOEA/D; normalization; engineering optimization; differential evolution;  $\varepsilon$ -constraint handling