

文章编号: 1001-0920(2009)02-0249-05

# 一种基于佳点集原理的约束优化进化算法

肖赤心<sup>1,2</sup>, 蔡自兴<sup>1</sup>, 王 勇<sup>1</sup>, 周经野<sup>2</sup>

(1. 中南大学 信息科学与工程学院, 长沙 410083; 2. 湘潭大学 信息工程学院, 湖南 湘潭 411105)

**摘 要:** 提出一种基于佳点集理论解决约束优化问题的进化算法. 它将实分圆域中均匀分布的佳点映射到求解问题的搜索空间, 使得所构造的个体能在搜索空间内分布比采用随机方式更加均匀, 并引进预交叉机制来平衡佳点取点个数与算法搜索能力之间的矛盾. 新算法的遗传算子基于佳点技术构造, 精度不受空间维数的限制, 有利于高维优化问题. 对 6 个标准测试函数的数值实验结果验证了新算法的通用性、有效性和稳健性.

**关键词:** 进化算法; 佳点集; 约束处理技术; 约束优化

**中图分类号:** TP18 **文献标识码:** A

## Constrained optimization evolutionary algorithm based on good lattice points principle

XIAO Chi-xin<sup>1,2</sup>, CAI Zi-xing<sup>1</sup>, WANG Yong<sup>1</sup>, ZHOU Jing-ye<sup>2</sup>

(1. School of Information Science and Engineering, Central South University, Changsha 410083, China;  
2. Information Engineering College, Xiangtan University, Xiangtan 411105, China. Correspondent: XIAO Chi-xin,  
E-mail: chixinxiao@hotmail.com)

**Abstract:** An approach is proposed to handle constrained optimization problems by using evolutionary algorithm. It maps the good lattice points (GLP) from  $S$ -dimensional unit hypercube to the  $S$ -dimensional search space, which makes the individuals in search space distribute more evenly than the random methods do. In order to balance the quantity of GLP and the exploitive capability of proposed algorithm, a new pre-crossover operator is proposed. The gene operators of the new method are constructed according to GLP principle whose precision is not be confined to the dimension of the search space, which is benefit for some high dimensional optimization problems. The new approach is compared with other evolutionary optimization techniques by six benchmark functions. The results obtained show that the new approach is general, effective and robust.

**Key words:** Evolutionary algorithm; Good points set; Constraint-handling techniques; Constrained optimization

### 1 引 言

约束优化问题<sup>[1]</sup> (COPs) 是科学和工程应用领域经常遇到的一类数学规划问题. 进化算法 (EAs) 是一种模拟自然进化过程的全局优化方法, 同传统优化方法相比, 进化算法更适合求解约束优化问题<sup>[1]</sup>. 近年来, 作为进化算法的一个重要部分, 进化策略在处理约束优化问题时受到了极大的重视. 对此有以下两个原因: (1) 具有理论背景支持进化策略收敛; (2) 进化策略的自适应机制对其处理约束搜索空间有一定的帮助<sup>[2,3]</sup>. 在相同的约束处理技术下, 采用进化策略时算法的整体性能明显优于遗传算

法<sup>[4]</sup>. 通过融合 3 个简单的个体比较准则, Mezura 和 Coello<sup>[2]</sup> 对几种不同的进化策略进行了实验比较. 结果表明对于具有高维搜索空间、非线性等式约束条件的测试函数, 利用进化策略不能得到很好的结果. 在进化策略中交叉操作往往被忽略. 然而, 就约束优化问题而言, 可行解与不可行解在一些重要的区域进行交叉, 对于找到全局最优解很有帮助, 特别是当全局最优解位于可行域边界上时. 尽管在进化策略中也可加入传统的交叉操作<sup>[5]</sup>, 但其搜索能力十分有限. 运用正交表技术来指导交叉操作取得了许多成果<sup>[6-8]</sup>. 但是, 为了达到精度, 当正交表因素

收稿日期: 2007-12-05; 修回日期: 2008-01-26.

基金项目: 国家自然科学基金项目 (60234030, 60404021); 国家基础研究项目 (A1420060159); 湖南省院士基金项目 (06JJ Y3035).

作者简介: 肖赤心 (1973—), 男, 湖南湘潭人, 讲师, 博士生, 从事进化计算、神经网络的研究; 蔡自兴 (1938—), 男, 福建莆田人, 教授, 博士生导师, 从事人工智能、计算智能等研究.

个数和等级增多时,不但实验的规模迅速增加,而且构造对应的正交表也很麻烦,这些给正交交叉法的运用带来了一定的困难.

本文引进佳点集技术<sup>[9-11]</sup>重新设计交叉算子和变异算子<sup>[12-14]</sup>,在保持收敛速度的同时利用佳点集技术使个体在搜索空间中分布比采用随机方法更均匀.数值实验验证了新算法的有效性和可行性.

## 2 基于佳点集原理的约束优化进化算法 (GLPES)

一个非线性规划问题可描述为

$$\min f(x) = (x_1, x_2, \dots, x_n) \quad \mathbf{R}^l.$$

$$\text{s. t. } g_j(x) = 0, \quad j = 1, 2, \dots, l;$$

$$h_j(x) = 0, \quad j = l + 1, l + 2, \dots, m.$$

这里  $x$  为决策向量,  $D$  为可行域,  $S$  为决策空间  $D \subseteq S$ . 一般地,  $S$  为  $\mathbf{R}^n$  中的  $n$  维长方体:  $l(i) \leq x_i \leq u(i)$ ,  $l(i), u(i)$  为常数,  $i = 1, 2, \dots, n$ .  $f(x), g_j(x), h_j(x)$  均为  $\mathbf{R}^n$  上的  $n$  元函数,  $f(x)$  为目标函数,  $g_j(x) = 0$  为第  $j$  个不等式约束条件,  $h_j(x) = 0$  为第  $j$  个等式约束条件.  $l$  表示不等式约束条件的个数,  $m - l$  表示等式约束条件的个数.

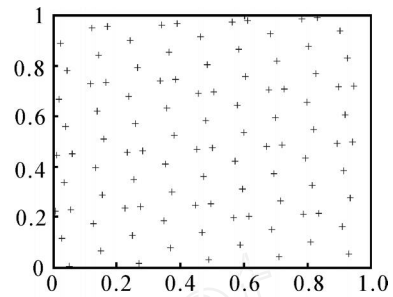
### 2.1 佳点集交叉算子

用任意给定  $n$  个点的函数值构成的任何加权和来近似计算函数在  $S$  维欧式空间单位立方体  $G_S$  上的积分时,取  $n$  个佳点<sup>[9-11]</sup> 所得的误差最小,即  $E_n = \mu - Q_n$ , 这里  $u = \int_x G_S f(x) dx, Q_n = \frac{f(p_n(i))}{n}$ ,  $P_n(k) (1 \leq k \leq n)$  是一个含  $n$  个点的点集. 也就是说,在相同取点个数的条件下,佳点序列要比其他方式选取的点序列更均匀,更逼近被积函数曲线. 如果将  $G_S$  上的佳点映射到求解空间,便得到一种很好的均匀取点方法. 另外,佳点集的构造与空间维数无关,克服了正交表设计法的不足,也为高维近似计算提供了一个非常好的算法. 下面构造一种适合进化策略的实数编码交叉算子.

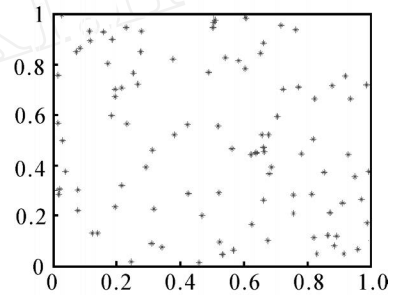
佳点集的构造: 如果  $r \in G_S$ , 形为  $P_n(k) = \{ \{ r_1^{(n)} * k \}, \dots, \{ r_s^{(n)} * k \}, \dots, \{ r_s^{(n)} * k \} \}$ ,  $k = 1, 2, \dots, n$ , 其偏差  $(n)$  满足  $(n) = C(r, ) n^{-(1+\epsilon)}$ , 其中  $C(r, )$  是只与  $r, (\epsilon > 0)$  有关的常数, 则称  $P_n(k)$  为佳点集,  $r$  称为佳点. 取  $r_i = \{ 2 \cos \frac{2-j}{p} \}, 1 \leq i \leq s$ ,  $p$  是满足  $\frac{p-3}{2} \leq s$  的最小素数; 也可取  $r_i = \{ e^i \}, 1 \leq i \leq s$ .  $\{ r_i * k \}$  表示  $r_i * k$  取 1 的模.

构造一个包含 100 个二维佳点的佳点集, 同时用随机方法在二维单位空间内取 100 个点, 图 1 分别给出了它们的分布效果. 显然佳点集的分布均匀,

且只要取点个数一定, 每次所得分布效果是一致的, 由此可知佳点集方法稳定性很好.



(a) 二维佳点集



(b) 二维随机点集

图 1 100 个佳点与 100 个随机点在二维空间分布的对比

在交叉操作中, 每随机选取两个个体 (不妨设为  $X_1 = (x_1^1, x_2^1, \dots, x_n^1), X_2 = (x_1^2, x_2^2, \dots, x_n^2)$ ), 便确定了一个局部搜索空间  $[x_{X_1, X_2}, \bar{x}_{X_1, X_2}]$ , 其中

$$\begin{cases} x_{X_1, X_2} = (\min(x_1^1, x_1^2), \min(x_2^1, x_2^2), \dots, \min(x_n^1, x_n^2)), \\ \bar{x}_{X_1, X_2} = (\max(x_1^1, x_1^2), \max(x_2^1, x_2^2), \dots, \max(x_n^1, x_n^2)). \end{cases} \quad (1)$$

令

$$J = \{ i / x_i^1 \text{ not } \epsilon\text{-similar with } x_i^2, 1 \leq i \leq n \}.$$

当  $d(x_i^1, x_i^2) = |x_i^1 - x_i^2| \leq \epsilon$ , 称两个个体第  $i$  维分量  $\epsilon$ -similar, 是一个相对于变量取值范围非常小的常数, 实验中  $\epsilon$  初值为  $0.0001 * x_i$ ,  $x_i$  为决策变量  $x_i$  的上下界之差. 另外对  $J$  还引进了动态机制, 它将随着进化代数  $t$  的增加不断缩小.  $\epsilon_{t+1} = \epsilon / C$ ,  $C$  为收敛系数. 不妨设  $X_1$  和  $X_2$  前  $t$  个分量  $\epsilon$ -similar 不成立, 后  $n - t$  个分量  $\epsilon$ -similar 成立.

在  $t$  维空间中取佳点的方法如下:

$t$  维空间  $H$  中作含  $m$  个点的佳点集  $P_n(k) = \{ \{ r_1 * k \}, \dots, \{ r_t * k \}, \dots, \{ r_t * k \} \}$ ,  $k = 1, 2, \dots, m$ , 其中  $r_i = \{ 2 \cos \frac{2-j}{p} \}, 1 \leq i \leq t$ .  $p$  是满足  $p \geq 2t + 3$  的最小素数, 或者取  $r_i = e^i, 1 \leq i \leq t, \{ r_i * k \}$  表示  $r_i * k$  的小数部分.

佳点集原理是基于实分圆域提出的, 佳点的每个分量均在  $[0, 1]$  内. 为了将单位立方体中的点映

射到局部搜索空间,  $[x_{L,M}, \bar{x}_{L,M}]$  不妨用  $T$  表示. 定义如下映射  $f: H \rightarrow T$ , 有

$$f(r_i * k) = \min(x_i^1, x_i^2) + \{r_i * k\} * (\max(x_i^1, x_i^2) - \min(x_i^1, x_i^2)). \quad (2)$$

其中:  $1 \leq i \leq t, 1 \leq k \leq m, t$  是空间的维数,  $m$  是取点的个数,  $\{r_i * k\}$  意义同上. 运用佳点集原理, 构造进化算法的佳点集交叉算子 (GLP crossover), 每运行一次该算子将产生  $\mu$  个新个体, 并加入子代中,  $\mu$  为父代个体数, 为子代个体数. 算子描述如下:

```
function GLP-crossover()
    设  $X_1 = \{x_1^1, x_2^1, \dots, x_n^1\}, X_2 = \{x_1^2, x_2^2, \dots, x_n^2\}$ 
```

是两个个体;

```
For  $k = 1$  to 每对个体交叉将产生新个体的个数  $\mu$  Do
```

```
    If ( $i \in J$ )
```

```
        If (flip(50%))
```

```
             $x_{ki}^{new} = x_i^1;$ 
```

```
        Else
```

```
             $x_{ki}^{new} = x_i^2;$ 
```

```
        End If
```

```
    Else
```

```
         $x_{ki}^{new} = f(r_i * k);$ 
```

```
    End If
```

```
End For
```

```
End For
```

```
End
```

其中:  $J$  是上面的定义的集合; flip( $P$ ) 函数的功能是, 如果随机获得的概率大于或等于  $P$ , 则返回 true.

### 2.2 预交叉策略

佳点集交叉算子能使产生的新个体在局部搜索空间内均匀分布. 但如果单位空间内的个体太稀疏, 搜索效果便不会明显. 实验证明, 单纯采用佳点集交叉算子在搜索空间较大, 而个体较稀疏的情况下效果并不理想. 为了克服这个问题可以适当增大  $\mu$ , 而  $\mu$  的变化又会引起进化代数的变化和早熟收敛等问题. 为了让佳点集交叉算子产生最好的效果, 同时均衡算法的整体稳定, 需在佳点集交叉算子之前加入一个预交叉算子, 目的在于迅速收敛搜索空间. 因为每次产生个体的数目一定, 要快速增加个体分布密度的最好办法便是采用一种有效机制使搜索空间迅速缩小. 本文希望用这样的机制来达到局部空间个体密度和整个算法搜索能力的平衡 (本文中  $\mu = 4, \mu = 100, \mu = 400$ ). 预交叉算子 (pre-crossover operator) 对从父代种群中随机抽取的个体  $P_1, P_2$  进行预交叉, 并产生两个中间个体  $X_1$

$= (x_1^1, x_2^1, \dots, x_n^1), X_2 = (x_1^2, x_2^2, \dots, x_n^2)$  作为佳点集交叉算子的输入. 此时两个中间个体  $X_1$  和  $X_2$  之间的相同维决策变量所确定的长度空间是原个体  $P_1$  和  $P_2$  所确定空间的  $1/2$ . 该算子描述如下:

```
function Pre-crpspver()
```

设  $P_1, P_2$  是  $\mu$  个父代中随机选取的两个个体, 相应的  $X$  分量为

$$(x_1^{p1}, x_2^{p1}, \dots, x_n^{p1}), (x_1^{p2}, x_2^{p2}, \dots, x_n^{p2});$$

```
For  $i = 1$  to 个体的维数  $n$  Do
```

```
     $x_{1i}^{new} = r * x_i^{p1} + (1 - r) * x_i^{p2};$ 
```

```
    If (flip(50%))
```

```
         $x_{2i}^{new} = x_i^{p1};$ 
```

```
    Else
```

```
         $x_{2i}^{new} = x_i^{p2};$ 
```

```
    End If
```

```
End For
```

```
End
```

其中 flip( $P$ ) 定义同前所述,  $r = 0.5$ . 也就是说在局部空间加入预交叉算子后, 个体的密度较单纯采用佳点集交叉增加一倍. 因为在后续迭代中, 被抛弃的空间是可能重现的, 所以不必担心抛弃空间带来的弊端. 这样既保证了取点的均匀, 又使算法的搜索能力大大提高.

### 2.3 变异

在  $\mu$  个个体组成的初始种群中, 每个个体代表一个实值向量集合  $(x_i, i), i = 1, 2, \dots, \mu, x_i$  和  $i$  有  $n$  个独立分量.  $x_i = \{x_i(1), x_i(2), \dots, x_i(n)\}, i = 1, 2, \dots, \mu, i = \{i(1), i(2), \dots, i(n)\}, i = 1, 2, \dots, \mu$ . 新算法采用高斯变异算子<sup>[10]</sup>. 实验验证, 在产生  $\mu$  个子种群时用上述佳点集交叉算子来产生子代个体的策略参数, 会使  $x_i$  的值分布过于均匀, 达不到变异效果. 此外, 作者认为  $x_i$  的变化还应具有一定的继承性. 设每对父代个体  $P_1$  和  $P_2$  会产生  $\mu$  个后代, 对于每个后代, 新算法分别以 50% 的概率选取  $P_1$  或  $P_2$  的分量作为交叉操作的个体之一<sup>[1]</sup>, 另一个交叉参与个体<sup>[2]</sup> 在每一维交叉时从父代群体中随机选取. 交叉方式如下:

```
function Sigma-crossover()
```

设  $P_1, P_2$  是输出 Pre-crossover 算子的两个个体, 对应的  $X$  分量为

$$P^1 = (p_1^1, \dots, p_n^1), P^2 = (p_1^2, \dots, p_n^2)$$

```
For  $j = 1$  to 每对个体交叉将产生新个体的个数  $\mu$  Do
```

```
    If (flip(50%))  $x_j^1 = P^1;$ 
```

```
    Else  $x_j^1 = P^2;$ 
```

```
End If
```

```

For  $i = 1$  to 个体的维数  $n$  Do
  随机地  $\mu$  个父代个体中选择一个个体设为  $T$ ,
 $^2 = ^r$ ;
  If (flip (50 %))
    If (flip (50 %))  $_{ji}^{new} = ^1_i$ ;
    Else  $_{ji}^{new} = ^2_i$ ;
  End If
Else
 $_{ji}^{new} = (^1_i + ^2_i)/2$ ;
End If
End For
End

```

其中 flip( $P$ ) 定义同前所述,  $r = 0.5$ . 这样做保证了值的多样性和一定的代间继承性, 实验证明了它的有效性. 变异算子的参数和定义如下 ( $n$  为个体的维数):  $^1_i = 1/\sqrt{2}\sqrt{n}$ ,  $^2_i = 1/\sqrt{2n}$ . 初始化  $^0_i(0) = 0.4 * (\frac{x_i}{\sqrt{n}})$ , 这里  $x_i$  为决策变量  $x_i$  的上下界之差.

#### 2.4 初始化种群的佳点集方法

由图 1 不难理解, 要使固定数目的个体在空间中可靠地均匀分布, 佳点集方法是一个很好的选择. 所以对于某些初始种群数目一定而搜索空间的范围较大, 即个体浓度相对稀疏的问题, 可用佳点集方法代替随机选种, 使个体在初始种群中尽量均匀分布, 从而提高算法的稳定性. 初始化的具体做法类似于在  $i$  维空间中取佳点的方法, 如 2.1 节式(2). 实验分析中, 在初始种群数为  $\mu = 100$  的前提下, 对 6 个测试问题<sup>[2, 15, 16]</sup> 采用了这种方法, 对问题 g10 效果尤为明显.

#### 2.5 选择下一代种群

在基于佳点集技术的约束优化进化算法 (GLPES) 中, 本文采用 Deb 的比较机制<sup>[17]</sup> 以及 Mezura 和 Coello<sup>[2]</sup> 的多样性机制, 并采用  $(\mu + \lambda)$ -ES 策略. 如果选择概率为 0.97, 则在  $\mu$  个父代个体与  $\lambda$  个子代群体中, 按照 Deb 的个体比较机制选择最好的个体进入下一代种群; 如果选择概率为 0.03, 则分别以二分之一的概率在  $\mu$  个父代种群或者  $\lambda$  个子代种群中选择最好的不可行解进入下一代种群. 直到选满  $\mu$  个新个体为止.

#### 2.6 佳点集进化算法流程

- 1) 用佳点集方法构造初始种群  $p(0)$ , 种群数  $\mu = 100$ ;
- 2) 是否达到最大进化代数  $T_{max}$  (设为 600), 是则结束算法; 否则转下一步;
- 3) 从父代中随机选择两个个体  $P_1, P_2$  进行预

交叉得两个中间个体  $X_1, X_2$ ;

- 4) 由佳点集交叉算子对  $X_1$  和  $X_2$  进行交叉, 产生  $\lambda/\mu$  个新的后代个体;
- 5) 用 的交叉算子为各个  $\lambda/\mu$  后代产生变异参数的值;
- 6) 对新产生的  $\lambda/\mu$  个体进行高斯变异, 并加入子种群(子种群数 = 400);
- 7) 循环次数是否达到  $\mu = 100$  次, 是转下一步; 否则转 3);
- 8) 从父代和子代群体共  $\mu + \lambda$  个体中选出新一代  $\mu = 100$  个新个体, 转 2).

### 3 实验结果与分析

在通用的 24 个标准测试函数 (benchmark)<sup>[16]</sup> 中选取 6 个高维优化问题测试新算法的性能, 并与其他几种已有算法的试验结果进行比较. 实验中, 对于每个问题, 取父代种群数  $\mu = 100$ , 子代种群数  $\lambda = 400$ , 运行代数  $t = 600$ , 共产生 240000 个新个体.

对于等式约束优化问题, 采用 Hamida<sup>[18]</sup> 在 ASCHEA 算法中提出的方法. 偏差值随着种群的代数的递增逐渐下降, 其表达式如下:

$$j(t+1) = j(t)/C,$$

其中  $C$  是常数, 问题 g03 中的值设为 1.00195, 问题 g13 中的值设为 1.0145.

对于每一个问题在相同的条件下独立运行了 30 次, 记录其最好结果、平均结果和最差结果. 为了对比实验结果, 将新算法 (GLPES) 与其他 4 种有代表性的算法——SMES<sup>[5]</sup>, ASCHEA<sup>[18]</sup>, SR<sup>[15]</sup> 和 CW<sup>[19]</sup> 进行比较. 表 1 分别列出了实验中 5 种算法在 6 个测试问题上的最好值、最坏值、平均值以及方差结果.

从表 1 可以看出, 新算法在 6 个测试问题中全部达到了全局最优解. 其中问题 g13 中, 找到一个解  $x = (-1.689588, 1.564713, 1.877753, 0.758274, 0.776972)$ ,  $f(x) = 0.0536792$ , 优于目前已知的最优解. 对于问题 g02, 新算法独立运行 30 次, 10 次接近最优解, 且所找到的最优解优于文献[5, 15, 18]; g10 的搜索空间较大, 按 2.4 节方法对种群进行初始化, 并对  $\lambda$ -similarity 动态机制中的收缩力度相对加强  $i_{t+1} = i_t/C$ , 收敛系数  $C$  设为 1.125; 如果强度不够, 佳点集交叉算子将在后期退化为直接交叉, 整个交叉过程退化成为只有预交叉在起作用, 大大降低了搜索效率. 对于其他问题, 可让  $\lambda$  的收缩强度相对弱一些, 实验中设为 1.0165, 这样更容易继承父代的优秀基因. 从方差分析来看, 新算法优于 SMES, SR 和 ASCHEA, 说明该算法稳定. 新算法的结果稍

表 1 5 个算法 SMES<sup>[5]</sup>, ASCHEA<sup>[18]</sup>, SR<sup>[15]</sup>, CW<sup>[19]</sup>, GLPES 在 6 个标准测试函数上的实验结果比较

Fcn/optimal	status	methods				
		SMES	ASCHEA	SR	CW	GLPES
g02/ - 0.803619	best	- <b>0.803601</b>	- 0.785	- 0.803515	- <b>0.803619</b>	- <b>0.803619</b>
	mean	- 0.785238	- 0.788950	- 0.781975	- 0.803220	- 0.79984
	worst	- 0.751322	NA	- 0.726288	- 0.792608	- 0.784106
	st. dev	1.7E-02		7.0E-02	2.0E-03	0.010254
g03/ - 1.000	best	- <b>1.000</b>	- <b>1.000</b>	- <b>1.000</b>	- <b>1.000</b>	- <b>1.000023</b>
	mean	- 1.000	- 0.99989	- 1.000	- 1.000	- 1.000031
	worst	- 1.000	NA	- 1.000	- 1.000	- 1.000051
	st. dev	2.1E-04		1.9E-04	2.8E-16	3.29E-08
g07/ 24.306	best	24.327	24.33230	<b>24.307</b>	<b>24.306</b>	<b>24.30646</b>
	mean	24.475	24.66360	24.374	24.306	24.39919
	worst	24.843		24.642	24.306	24.45859
	st. dev	1.3E-01		6.6E-02	5.7E-12	0.047817
g10/ 7049.248	best	<b>7051.903</b>	7061.13	7054.316	<b>7049.248</b>	<b>7049.652632</b>
	mean	7253.047	7193.11	7559.192	7049.248	7108.625
	worst	7638.366	NA	8835.655	7049.248	7208.321
	st. dev	1.4E+02		5.3E+02	4.0E-09	0.5090E+02
g12/ - 1.000	best	- <b>1.000</b>	NA	- <b>1.000</b>	- <b>1.000</b>	- <b>1.000</b>
	mean	- 1.000	NA	- 1.000	- 1.000	- 1.000
	worst	- 1.000	NA	- 1.000	- 1.000	- 1.000
	st. dev	0		0	0.0E+00	0
g13/ 0.0539498	best	0.053986	NA	<b>0.053957</b>	<b>0.0539498</b>	<b>0.053914718</b>
	mean	0.166385	NA	0.067543	0.0539498	0.108915872
	worst	0.468294	NA	0.216915	0.0539498	0.467101246
	st. dev	1.8E-01		3.1E-02	6.5E-17	0.1101

注: NA 表示原文中没有提供实验结果.

逊于 CW 算法,但它的进化机制要简单得多,这使得算法复杂度相对较小. 本文使用 C++ 编写程序代码,VC++ 6.0 环境编译,Intel Celeron2.67,256M 内存的机器以及 WinXP 平台独立运行 30 次的速度最长不超过 6 min(g12 除外).

#### 4 结 论

新算法引进佳点集技术使空间采样点可靠地均匀分布,使采样点更具代表性,从而克服传统交叉算子搜索能力不足的缺点. 同时由于佳点集方法可以不受搜索空间维数的限制,从而给出了一种替代正交方法解决高维优化问题的思想. 实验结果表明了本文结论与设计方案的有效性. 在今后的工作中,将用大量的实际问题进一步检验新方法,更深入地研究佳点集技术,继续对算法加以改进,并将其应用于神经网络结构和权值的优化.

#### 参考文献(References)

[1] Michalewicz Z, Schoenauer M. Evolutionary algorithm for constrained parameter optimization problems [J]. Evolutionary Computation, 1996, 4(1): 1-32.  
 [2] Mezura-Montes E, Coello Coello CA. On the usefulness of the evolution strategies self-adaptation mechanism to handle constraints in global optimization [R]. Mexico

City: CINVESTAV-IPN, 2003.

[3] Asselmeyer T, Ebeling W, Ros 'e H. Evolutionary strategies of optimization[J]. Physical Review E, 1997, 59(1): 1171-1180.  
 [4] Mezura-Monte 's E. Alternative techniques to handle constraints in evolutionary optimization[D]. Computer Science Section, Electrical Engineer Department, Mexico City: CINVESTAV-IPN, 2004.  
 [5] Mezura-Montes E, Coello Coello CA. A simple multimembered evolution strategy to solve constrained optimization problems[J]. IEEE Trans on Evolutionary Computation, 2005, 9(1): 1-17.  
 [6] Qingfu Zhang, Yiu-Wing Leung. An orthogonal genetic algorithm for multimedia multicas e routing [J]. IEEE Trans on Evolutionary Computation, 1999, 3(1): 53-62.  
 [7] Yiu-Wing Leung, Yuping Wang. An orthogonal genetic algorithm with quantization for global numerical optimization [J]. IEEE Trans on Evolutionary Computation, 2001, 5(1): 41-43.  
 [8] Jinr-Tsong Tsai, Tung-Kuan Liu, Jyh-Horng Chou. Hybrid taguchi-genetic algorithm for global numerical optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8(4): 365-377.

(下转第 258 页)

## 参考文献(References)

- [1] Gao Y, Lygeros J, Quincampoix M. On the reachability problem of uncertain hybrid systems [J]. IEEE Trans on Automatic Control, 2007, 52(9): 1572-1586.
- [2] Gao Y, Lygeros J, Quincampoix M, et al. On the control of uncertain impulsive systems: Approximate stabilization and controlled invariance [J]. Int J of Control, 2004, 77(16): 1393-1407.
- [3] Quincampoix M, Seube N. Stabilization of uncertain control systems through piecewise constant feedback [J]. J of Mathematical Analysis and Applications, 1998, 218(1): 240-255.
- [4] Cardaliaguet P, Quincampoix M, Saint-Pierre P. Pursuit differential games with state constraints[J]. SIAM J of Control and Optimization, 2002, 39(5): 1615-1632.
- [5] 高岩. 线性控制系统的生存域[J]. 控制与决策, 2007, 22(7): 833-835.  
(Gao Y. On viable set for a linear control system[J]. Control and Decision, 2007, 22(7): 833-835.)
- [6] 高岩. 一类非线性控制系统关于非光滑区域生存性的判别[J]. 控制与决策, 2006, 21(8): 923-925.  
(Gao Y. Determining the viability for a class of nonlinear control system on a region with nonsmooth boundary [J]. Control and Decision, 2006, 21(8): 923-925.)
- [7] 高岩. 一类非线性控制系统可生存性的判别[J]. 信息与控制, 2005, 34(4): 510-512.  
(Gao Y. Determining the viability for a class of nonlinear control system [J]. Information and Control, 2005, 34(4): 510-512.)
- [8] Aubin J-P. Viability theory[M]. Boston: Birkhauser, 1991.
- [9] Clarke F H, Ledya Yu S, Stern R J, et al. Nonsmooth analysis and control theory[M]. New York: Springer-Verlag, 1998.
- [10] Hiriart-Urruty J B, Lemarechal C. Convex analysis and minimization[M]. Berlin: Springer-verlag, 1993.
- [11] Demyanov V F, Rubinov A M. Constructive nonsmooth analysis [M]. Frankfurt am Main: Peterlang, 1995.
- [12] Fisher M C. A special newton-type optimization method[J]. Optimization, 1992, 24(2): 269-284.
- [13] Garcia-Palomares U M. A superlinearly convergence projection algorithm for solving the convex inequality problem[J]. Operations Research Letter, 1998, 22(2): 97-103.
- [9] 华罗庚, 王元. 数论在近似分析中的应用[M]. 北京: 科学出版社, 1978.  
(Hua L G, Wang Y. The application of number theory in approximate analysis [M]. Beijing: Science Press, 1978.)
- [10] 张铃, 张钊. 佳点集遗传算法[J]. 计算机学报, 2001, 24(9): 917-922.  
(Zhang L, Zhang B. Good point set based genetic algorithm[J]. Chinese J of Computers, 2001, 24(9): 917-922.)
- [11] 程军盛, 张铃. 基于佳点集遗传算法求解 Job-shop 调度问题[J]. 计算机科学, 2002, 29(4): 67-68.  
(Cheng J S, Zhang L. Solving Job-shop scheduling problem using good point set based genetic algorithm [J]. Computer Science, 2002, 29(4): 67-68.)
- [12] Xiao C X, Cai Z X. Using good nodes set principle to evolution strategy for constrained optimization[C]. The 26th Chinese Control Conf. Zhangjiajie, 2007: 722-726.
- [13] Xiao C X, Cai Z X, Wang Y. Incorporating good nodes set principle into evolution strategy for constrained optimization[C]. ICNC2007. Haikou, 2007: 243-247.
- [14] Xiao C X, Cai Z X, Wang Y. A good nodes set evolution strategy for constrained optimization [C]. 2007 IEEE Congress of Evolutionary Computation. Singapore, 2007: 943-950.
- [15] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Trans on Evolutionary Computation, 2000, 4(3): 284-294.
- [16] Liang J J, Thomas Philip Runarsson, Efren Mezura-Montes, et al. Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization [EB/OL]. (2006-09-18) <http://www.ntu.edu.sg/home/EPNSugan/>.
- [17] Deb K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics Engineering, 2000, 86(2-4): 311-338.
- [18] Hamida S B, Schoenauer M. ASCHEA: New results using adaptive segregational constraint handling [C]. Proc of the Congress on Evolutionary Computation. Honolulu, 2002: 884-889.
- [19] Cai Z, Wang Y. A multiobjective optimization based evolutionary algorithm for constrained optimization[J]. IEEE Trans on Evolutionary Computation, 2006, 10(6): 658-675.

(上接第 253 页)