

## 遗传算法计算效率的改进

周克民, 胡云昌

(天津大学 建筑工程学院, 天津 300072)

**摘要:** 根据适应值的分布, 采用缩小、移动搜索区间的方法, 将整体和局部寻优能力有机地结合起来, 明显地提高了遗传算法的收敛速度和解的精度. 本文提出的方法对大范围、高精度寻优尤其适合. 最后以连续函数为例, 说明了算法的有效性.

**关键词:** 遗传算法; 计算效率; 变搜索区间; 连续函数优化

**中图分类号:** TP301.6      **文献标识码:** A

## Improvement of computational efficiency for genetic algorithms

ZHOU Ke-min, HU Yun-chang

(Civil and Building Engineering School, Tianjin University, Tianjin 300072, China)

**Abstract:** According to the distribution of fitness, the region of the search space is decreased and moved. By combining the ability of global optimization and local search, the convergence speed and the precision of solution are improved. It is rather better when optimizing solution with high precision is searched in large region. The effectiveness of the proposed method is demonstrated by optimizing several continuous functions.

**Key words:** genetic algorithm; computational efficiency; variable search space; continuous function optimization

### 1 引言(Introduction)

遗传算法作为一种非线性全局优化搜索算法, 得到广泛应用<sup>[1]</sup>. 但其计算效率过低是存在的一个重大问题. 为此, 人们从不同方面进行了许多工作<sup>[2,3]</sup>. 遗传算法是以 0,1 组成的一个字符串来表示一个连续或离散变量. 当它表示连续变量时, 实际上只能表示连续区间上的一些离散点, 这直接影响了解的表达精度. 在一些实际问题中, 有时很难事先确定最优解的可能范围, 只好把定义域选得很大, 如仍要保持表达式精度就需要很长的字符串, 这势必降低收敛速度, 计算效率很低. 如果表达式精度过低, 则还可能找不到最优解. 另外, 在优化后期明显表现出局部寻优能力的不足也是一重大缺点.

为解决这一问题, 本文根据群体的进化过程, 反复缩小寻优区间, 可以用较短的串长, 给出较高的精度. 为减少最优点落到寻优区间以外的机会, 还采取了移动寻优区间的方法.

### 2 遗传算法的改进(The improvement of genetic algorithms)

#### 2.1 简单遗传算法(SGA)的表述(The description of simple genetic algorithms(SGA))

以优化问题:

$$\begin{cases} \text{Find } x = (x_1, x_2, \dots, x_m), x_i \in [x_i^L, x_i^U], \\ i = 1, 2, \dots, m, \\ \text{maximize } f(x) \end{cases} \quad (1)$$

为例, 式中  $x_i^L, x_i^U$  为变量  $x_i$  的上下限, 每个变量  $x_i$  可以用  $n_i$  个由 0,1 组成的子字符串  $\hat{S}_i$  组成, 也可理解为二进制数.

$$\begin{cases} \hat{S}_i = B_i^{n_i} B_i^{n_i-1} \dots B_i^1, \\ B_i^j = 0 \text{ 或 } 1, j = 1, 2, \dots, n_i, i = 1, 2, \dots, m. \end{cases} \quad (2)$$

其中  $B_i^j$  称为基因, 与变量  $x_i$  的关系为:

$$x_i = x_i^L + \frac{x_i^U - x_i^L}{2^{n_i} - 1} \sum_{j=1}^{n_i} B_i^j 2^{j-1} (i = 1, 2, \dots, m). \quad (3)$$

将子串  $\hat{S}_i (i = 1, 2, \dots, m)$  排列构成  $S = \hat{S}_1 \hat{S}_2 \dots \hat{S}_m$ , 称为字符串或染色体, 也称个体. 式(3)这一关系式可记作  $x_i = \varphi_i(S)$ , 写成矢量形式  $x = \varphi(S)$ . 字符串所有可能的取值构成字符串空间. 对应变量  $x_i$  的空间称为设计空间. 由若干个字符串构

成群体  $S_i (i = 1, 2, \dots, np)$ ,  $np$  为群体规模. 用  $S_i(t)$  表示第  $t$  代的第  $i$  个个体. 称  $f(\varphi(S_i(t)))$  为第  $t$  代第  $i$  个个体的适应值.

在 SGA 中, 群体经过复制, 交叉和变异等基本遗传操作进行一代代演化, 使适应值不断升高, 从而找到最优解.

### 2.2 遗传算法的改进(The improvement of genetic algorithms)

**划分寻优空间** 字符串中表示各个变量  $x_i$  的子字符串  $\hat{S}_i$  的最高位(最左边)  $B_i^n$  可以是 0 或 1 (用  $b$  表示, 下同). 据此存在一种划分, 可以把字符串划分成对等的两个子空间. 由于有  $m$  个变量, 存在  $m$  个这种划分方式, 可以形成  $m$  对子空间, 用集合表示:

$$A_i^b = \{S \mid B_i^n = b\} \quad (i = 1, 2, \dots, m, b = 0, 1). \quad (4)$$

为划分区间, 在此将个体依适应值降序排列为  $S'_1, S'_2, \dots, S'_{np}$ , 即:  $f(\varphi(S'_i)) \geq f(\varphi(S'_{i-1})) (i = 2, 3, \dots, np)$ .

**设计空间退化** 在演化到某一代时, 如果适应值最高的前  $np_0$  ( $np_0$  取群体规模的一事先确定的比例, 本文取  $0.3np$ ) 个个体都位于同一字符串子空间(如  $A_k^b$ ) 内:

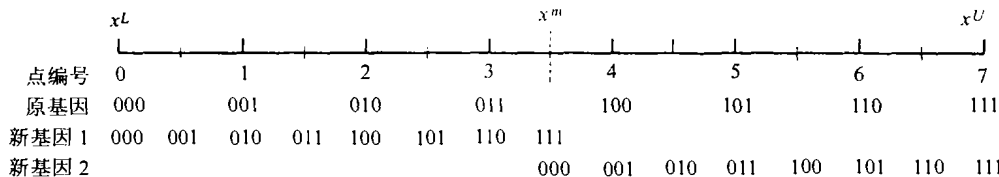
$$S'_j \in A_k^b (j = 1, 2, \dots, np_0; b = 0 \text{ 或 } 1). \quad (5)$$

可以认为最优点以很大概率落入  $A_k^b$  中, 以此作为下一代的寻优空间. 对应变量为:

$$x_i \in \begin{cases} [x_i^L, x_i^U], & i \neq k, \\ [x_k^L, x_k^U], & i = k, \end{cases} \quad [x_k^L, x_k^U] = \begin{cases} [x_k^L, x_k^m], & b = 0, \\ [x_k^m, x_k^U], & b = 1, \end{cases} \quad x_k^m = \frac{1}{2}(x_k^L + x_k^U). \quad (6)$$

由于在该空间内表示第  $i$  个变量的子字符串, 式(2)中的  $\hat{S}_i$  最高位  $B_i^n$  都一样, 为提高编码效率, 提高变量表达精度, 同时保证各基因位按模式定理解释时含义不变, 将  $\hat{S}_i$  中的各位基因位从左边第二位  $B_i^{n-1}$  开始, 依次左移一位:

$$B_i^{j+1} \Leftarrow B_i^j, \quad j = n_i - 1, n_i - 2, \dots, 1. \quad (7)$$



式中符号“ $\Leftarrow$ ”为算法语言中的赋值操作, 表示把符号右侧的值复制到左侧. 而最后一位再由随机数填充. 为保护最优个体, 使其在区间退化时对应变量不变, 最优个体的最后一位与移动前的首位一致. 由于设计空间的不断退化, 每个变量的串长  $n_i$  无需太长, 取 4 至 6 位, 不影响精度. 本文取 4 位.

**寻优空间的移动** 如果当前最优解的某个分量  $x_k$  处在当前设计空间的边界, 该变量对应的子串(见式(2)), 的各位相同, 均为 0 或 1 (用  $b$  表示), 则认为最优解有可能在当前寻优区间以外. 此时, 在该分量方向移动寻优空间, 以减少寻优空间缩减而导致的失去最优解的机会. 本文取移动距离为  $2d_k$ .  $d_k$  为沿  $x_k$  方向相邻两个离散点间的距离:

$$d_k = (x_k^U - x_k^L) / (2^{n_k} - 1). \quad (8)$$

移动方法是调整边界如下:

$$[x_k^L, x_k^U] = \begin{cases} [x_k^L - 2d_k, x_k^U - 2d_k], & b = 0, \\ [x_k^L + 2d_k, x_k^U + 2d_k], & b = 1. \end{cases} \quad (9)$$

然后改变对应子串, 改变方法是把该子串作为二进制数, 当  $b = 1$  时减 2; 反之加 2. 这样操作保证了处在移动前后两个空间的重叠部分的个体处在设计空间的同一位置上. 当有进位或借位发生时, 说明该点将被移出当前寻优空间, 略去进位或借位, 就会落入新移入的那部分寻优空间内. 可以理解为随机产生的新个体.

图 1 用于说明上述过程. 为简单计, 这里假设一个变量的串长为 3. 如果当前几个最优解的首位为 0, 说明位于点 0 至 3 之间, 此时去掉右半区间, 形成图 1 中的新基因 1. 形成新的基因的方法是: 各位左移, 末位由随机数产生. 为保留最优个体, 此时最优个体的末位填 0, 以保证对应设计空间的位置不变. 如点 2 的 010 变成了 100. 新产生的位置是否有个体由随机数决定. 当最优点位于点 0 时, 基因各位均为 0, 寻优区间左移两格, 点 0 在移动后的区间内的编码变为 010, 相当于加 2 操作. 如果当前几个最优解的首位为 1, 说明位于点 4 至 7 之间, 此时去掉左半区间, 形成图 1 中的新基因 2. 最优个体的末位填 1.

图 1 寻优区间缩减和移动  
Fig. 1 Searching space degenerating and moving

### 3 算例(Examples)

本文使用文献[4]的  $f_1$  和文献[5]的  $f_2$  和  $f_3$  三个目标函数作为验证算法有效性的函数. 在所有算例中均采用了最优个体保留策略及适应值线性比例变换方法, 杂交率为 0.7, 平均收敛代数取 50 次计算的平均值, 每代的个体数为 40. 对于  $f_1$ , 收敛是指优化结果与理论值的误差小于 1%, 由于  $f_2, f_3$  较难收敛, 误差取作 5%. 在改进前的算法中, 对每个函数优化过程中使用的串长及变异率都进行了优化. 优化范围是串长在 [4, 64] 范围内, 间距为 4; 变异率在 [0.01, 0.12] 范围内, 间距为 0.01. 表 1 中所列结果为参数优化后的计算结果. 在改进后的算法中, 采用了统一参数.

表 1 计算结果比较

Table 1 The comparison of results

方法	改进前			改进后		
	$f_1$	$f_2$	$f_3$	$f_1$	$f_2$	$f_3$
平均收敛代数	56.24	228.2	183.1	32.04	69.46	26.64
单个变量串长	20	16	20	4	4	4
变异率	0.08	0.08	0.08	0.1	0.1	0.1

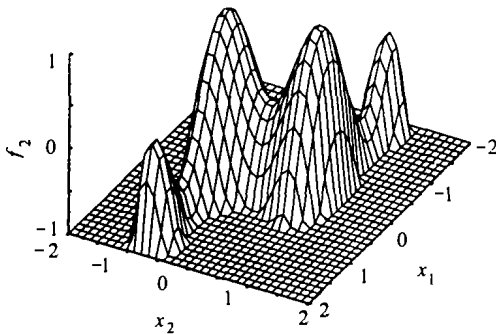


图 2 函数  $f_2$  在全局最优点附近的性态  
Fig. 2 The character near global optimum point of function  $f_2$

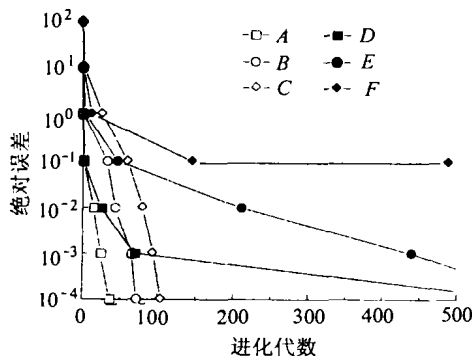


图 3 自变量的收敛情况  
Fig. 3 Independent variable convergence

为说明本文提出的方法适合于大范围、高精度寻优, 以较难优化的函数  $f_2$  为例进行了进一步的分析. 图 2 描述了  $f_2$  在全局最大点附近的性态. 图 3 和图 4 给出了在不同大小的寻优区间内, 自变量和目标函数值的收敛特性. 数据取三次平均值. 竖轴表示绝对误差, 横轴表示进化代数. 自变量的误差是指欧氏距离.  $A, B, C$  为变量范围分别取  $[-1, 1], [-10, 10]$  和  $[-100, 100]$  时改进后计算结果,  $D, E, F$  为改进前的对应结果.

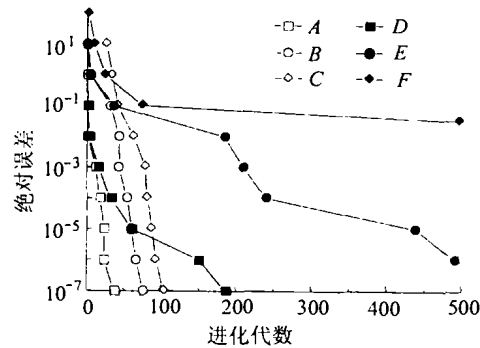


图 4 目标值的收敛情况  
Fig. 4 Objective value convergence

### 4 结论(Conclusions)

计算结果表明, 改进后的计算效率明显提高, 特别是在变量范围比较大时, 比改进前的算法更容易得到足够高精度的结果, 而改进前的算法则很难再进一步提高精度.

### 参考文献(References)

- [1] Xi Yugeng, Chai Tianyou, Yun Weimin. Survey on genetic algorithm [J]. Control Theory and Applications, 1996, 13(6): 697 - 708
- [2] Shi Kuifan, Chen Yuehui. A method of improving the convergence speed of genetic algorithm [J]. Information and Control, 1998, 27(4): 289 - 293
- [3] Rajeev S, Krishnamoorthy C S. Genetic algorithms-based methodologies for design optimization of trusses [J]. Journal of Structural Engineering, ASCE, 1997, 123(3): 350 - 358
- [4] Lu Yong, Kang Lishan, Chen Yuping. Non-Numerical Parallel Algorithms-Genetic Algorithm [M]. Beijing: Science Press, 1998, 90 - 91
- [5] Hou Gexian, Wu Chengke. Performance analysis for genetic algorithms [J]. Control and Decision, 1999, 14(3): 257 - 264

### 本文作者简介

周克民 1962年生. 1988年在天津大学获硕士学位. 现为天津大学建工学院博士研究生, 天津城市建设学院副教授. 研究方向为结构优化.

胡云昌 1936年生. 天津大学建工学院博士生导师. 研究方向为结构可靠性及优化.