

自交叉算子*

任庆生

曾 进

(上海交通大学计算机科学与工程系·上海, 200030) (上海交通大学应用数学系·上海, 200030)

戚飞虎

(上海交通大学计算机科学与工程系·上海, 200030)

摘要: 研究表明交叉算子对染色体具有破坏和重构的双重作用, 从而在进化代数增加的时候能够使模式内部各基因趋于独立, 并且只要组成模式的各个基因都存在, 则该模式一定能够被搜索到, 此时模式的极限概率就等于组成该模式各基因的初始概率(也就是基因的极限概率)的乘积, 与模式的定义长度无关. 在对交叉算子性能深入认识的基础上, 我们提出了一种新的交叉算子——自交叉算子, 实际的算例显示算法的性能得到明显改善.

关键词: 遗传算法; 模式; 自交叉算子

文献标识码: A

Self-crossover Operator

REN Qingsheng

(Department of Computer Science and Engineering, Shanghai Jiaotong University · Shanghai, 200030, P. R. China)

ZENG Jin

(Department of Applied Mathematics, Shanghai Jiaotong University · Shanghai, 200030, P. R. China)

QI Feihu

(Department of Computer Science and Engineering, Shanghai Jiaotong University · Shanghai, 200030, P. R. China)

Abstract: By studying the dual function of the crossover operator, we proved that the genes would be independent. Any schema which consists of genes would be searched if the genes existed. The limit probability of the schema equals the product of the initial probability of the genes (i. e. the limit probability of genes) and had nothing to do with the definition length of the schema. By this study, the self-crossover operator was proposed. The numerical results showed the performance of the algorithm was improved.

Key words: genetic algorithms; schema; self-crossover operator

1 引言(Introduction)

遗传算法是一种模拟自然界遗传和进化机制的高度并行、随机、自适应搜索算法, 其两大特点是隐含并行性和对全局信息的有效利用能力, 前者使算法只需检测少量的结构就能反映搜索空间的大量区域, 而后者使之具有鲁棒性. 目前, 遗传算法已经在金融、图象处理、通信等很多领域得到了成功应用^[1~4]. 与此同时, 如何提高遗传算法的性能也引起了人们的注意. 以交叉算子为例, 文献[5]和文献[6]都针对具体问题对交叉算子进行了成功的改进, 但为何改善算法性能却不是很清楚. 在一些讨论模式定理等理论的文献[7, 8]中, 我们也可以看到对交叉算子分析的重点还是在于其对染色体的破坏作用, 忽视了染色体被重新生成的可能, 从而无法准确地

掌握交叉算子的作用.

针对这些问题, 本文首先综合考虑交叉算子对染色体破坏和重构的双重作用, 给出了在交叉算子作用下染色体数量的变化规律. 在此基础上我们对交叉算子的极限性能进行了分析, 证明了交叉算子在进化代数增加的时候能够使模式内部各基因趋于独立, 并且只要组成模式的各个基因都存在, 则该模式一定能够被搜索到, 此时模式的极限概率就等于组成该模式各基因的初始概率(也就是基因的极限概率)的乘积, 同时与模式的定义长度无关. 通过这些理论分析, 有助于我们理解遗传算法的搜索能力, 并进而提出了自交叉算子这一新的算子, 实验显示算法的性能能够得到明显改善.

* 基金项目: 国家自然科学基金(60072029)和上海交通大学博士启动基金(A987011)资助项目.

收稿日期: 1999-11-02; 收修改稿日期: 2000-09-12

2 交叉算子的双重作用(Dual function of the crossover operator)

我们考虑最常用的一点交叉算子,其作用过程如下:挑选出两个染色体,并随机产生一个交叉位置 i , i 为 $[1, n-1]$ 上的一个整数,然后被选出的两个染色体相互对应地交换从 $i+1$ 到 n 的位段.设群体规模为 N ,染色体 $(x_1 x_2 \cdots x_n)$ 在时刻 k 时出现的概率为 $P_k(x_1 x_2 \cdots x_n)$.

当交叉率 $p_c = 0$ 时,显然有 $P_k(x_1 x_2 \cdots x_n) = P_{k-1}(x_1 x_2 \cdots x_n)$,而当交叉率 $p_c = 1$ 时,我们可以证明:

$$P_k(x_1 x_2 \cdots x_n) = \frac{1}{(n-1)(N-1)} \sum_{i=1}^{n-1} (NP_{k-1}(x_1 \cdots x_i)P_{k-1}(x_{i+1} \cdots x_n) - P_{k-1}(x_1 x_2 \cdots x_n)).$$

证 $P_k(x_1 x_2 \cdots x_n) = \sum_{i=1}^{n-1} P(\text{交叉点为第 } i \text{ 位})P(x_1 x_2 \cdots x_n \text{ 由 } x_1 \cdots x_i \text{ 和 } x_{i+1} \cdots x_n \text{ 合成}).$

由一点交叉算子的定义, $P(\text{交叉点为第 } i \text{ 位}) = \frac{1}{n-1}$.

记 $H_1 = x_1 \cdots x_i$, $H_2 = x_{i+1} \cdots x_n$, $\bar{H}_1 = x_1 \cdots x_i$ 表示前 i 位基因为除 $x_1 \cdots x_i$ 外所有的可能, $\bar{H}_2 = x_{i+1} \cdots x_n$ 表示后 $n-i$ 位基因为除 $x_{i+1} \cdots x_n$ 外所有的可能.

则 $P(x_1 x_2 \cdots x_n \text{ 由 } x_1 \cdots x_i \text{ 和 } x_{i+1} \cdots x_n \text{ 合成}) = P(\text{I}) + P(\text{II}) + P(\text{III}) + P(\text{IV})$, 其中

$P(\text{I}) = P(x_1 x_2 \cdots x_n \text{ 由 } H_1 \bar{H}_2 \text{ 和 } \bar{H}_1 H_2 \text{ 交叉生成}) =$

$$\frac{P(H_1 \bar{H}_2)N \cdot P(\bar{H}_1 H_2)N}{N \cdot (N-1)},$$

$P(\text{II}) = P(x_1 x_2 \cdots x_n \text{ 由 } H_1 \bar{H}_2 \text{ 和 } H_1 H_2 \text{ 交叉生成}) =$

$$\frac{P(H_1 \bar{H}_2)N \cdot P(H_1 H_2)N}{N \cdot (N-1)},$$

$P(\text{III}) = P(x_1 x_2 \cdots x_n \text{ 由 } H_1 H_2 \text{ 和 } \bar{H}_1 H_2 \text{ 交叉生成}) =$

$$\frac{P(H_1 H_2)N \cdot P(\bar{H}_1 H_2)N}{N \cdot (N-1)},$$

$P(\text{IV}) = P(x_1 x_2 \cdots x_n \text{ 由 } H_1 H_2 \text{ 和 } H_1 \bar{H}_2 \text{ 交叉生成}) =$

$$\frac{P(H_1 H_2)N \cdot (P(H_1 \bar{H}_2)N - 1)}{N \cdot (N-1)},$$

因此

$$P_k(x_1 x_2 \cdots x_n) = \frac{1}{(n-1)(N-1)} \sum_{i=1}^{n-1} (NP_{k-1}(x_1 \cdots x_i)P_{k-1}(x_{i+1} \cdots x_n) -$$

$$P_{k-1}(x_1 x_2 \cdots x_n)).$$

证毕.

这样,我们可以知道染色体 $(x_1 x_2 \cdots x_n)$ 在 k 时刻时的概率为:

$$P_k(x_1 x_2 \cdots x_n) = (1 - p_c)P_{k-1}(x_1 x_2 \cdots x_n) + \frac{p_c}{(n-1)(N-1)} \sum_{i=1}^{n-1} (NP_{k-1}(x_1 \cdots x_i)P_{k-1}(x_{i+1} \cdots x_n) - P_{k-1}(x_1 x_2 \cdots x_n)). \quad (1)$$

实际上,(1)式的第一部分表示了上一代模式未参加交叉而幸存下来的概率,而后一部分则表示了通过交叉生成该模式的概率,以前模式定理的推导一般都忽略了这后一部分.

3 交叉算子的极限性能(Limit property of crossover operator)

我们首先考虑阶数为 1 的模式,即单个基因的情况.由(1)式, k 时刻第 j 位基因的概率为:

$$P_k(x_j) = \sum_{x_1 \cdots x_{j-1} x_{j+1} \cdots x_n} P_k(x_1 \cdots x_n) =$$

$$P_{k-1}(x_j) = \cdots = P_0(x_j).$$

即对阶数为 1 的模式,在只有交叉算子的情况下,其概率恒定不变,一直等于其初始概率.

其次考虑阶数为 2 的模式的情况.记 $I = \{1, 2, \cdots, n\}$, $I_c = \{s, t\}$, 则 k 时刻时第 $s, t (s < t)$ 位基因的概率为

$$P_k(x_s x_t) = \sum_{x_1 \cdots x_n} P_k(x_1 \cdots x_n) = (1 - p_c \frac{N(t-s)}{(n-1)(N-1)})P_{k-1}(x_s x_t) + p_c \frac{N(t-s)}{(n-1)(N-1)} P_{k-1}(x_s)P_{k-1}(x_t) = (1 - p_c \frac{N(t-s)}{(n-1)(N-1)})^k P_0(x_s x_t) + [1 - (1 - p_c \frac{N(t-s)}{(n-1)(N-1)})^k] P_0(x_s)P_0(x_t).$$

由于 $p_c \frac{N}{N-1} \cdot \frac{t-s}{n-1} < 1$ 在实践应用中一般总能满足(只要 $p_c < \frac{N-1}{N}$ 即可),因此我们有

$$\lim_{k \rightarrow \infty} P_k(x_s x_t) = P_0(x_s)P_0(x_t).$$

在进一步的讨论之前,我们首先给出一个引理:

引理 设 $a_k = (1 - \alpha)b_{k-1} + \alpha a_{k-1}$, $0 < \alpha < 1$, 并且 $\lim_{k \rightarrow \infty} b_k = b$, 则有 $\lim_{k \rightarrow \infty} a_k = b$.

证 因为 $\lim_{k \rightarrow \infty} b_k = b$, 所以 $\forall \epsilon > 0, \exists N$, 当 $k > N$ 时, 有 $|b_k - b| < \epsilon$. 此时如果 $|a_k - b| < \epsilon$,

则 $|a_{k+1} - b| \leq (1 - \alpha) |b_k - b| + \alpha |a_k - b| < \epsilon$, 从而 $\lim_{k \rightarrow \infty} a_k = b$. 否则必然有 $|a_k - b| > \epsilon$. 不妨设 $a_k - b > \epsilon$, 则 $a_k - b > b_k - b$, 并且 $b_k < a_{k+1} < a_k$, 即 a_k 在 $k > N$ 后是单调下降的, 并且有下界, 所以 a_k 存在极限. 设 $\lim_{k \rightarrow \infty} a_k = a$, 则有 $a = (1 - \alpha)b + \alpha a$, 即 $a = b$, 从而 $\lim_{k \rightarrow \infty} a_k = b$. 证毕.

在此基础上, 我们可以证明

$$\lim_{k \rightarrow \infty} P_k(x_1 \cdots x_n) = P_0(x_1) \cdots P_0(x_n). \quad (2)$$

证 利用数学归纳法.

1) 当 $n = 1, 2$ 时, 已证明等式是成立的;

2) 设当小于 n 时等式也都成立, 则由

$$\begin{aligned} P_k(x_1 x_2 \cdots x_n) = & (1 - p_c) P_{k-1}(x_1 x_2 \cdots x_n) + \\ & \frac{p_c}{(n-1)(N-1)} \sum_{i=1}^{n-1} (N P_{k-1}(x_1 \cdots x_i) P_{k-1}(x_{i+1} \cdots x_n) - \\ & P_{k-1}(x_1 x_2 \cdots x_n)). \end{aligned}$$

我们有:

$$\begin{aligned} P_k(x_1 \cdots x_n) = & (1 - p_c) P_{k-1}(x_1 \cdots x_n) + \\ & \frac{p_c}{(n-1)(N-1)} [N(i_1 - 1) P_{k-1}(x_1 \cdots x_{i_1}) + \\ & N(i_2 - i_1) P_{k-1}(x_{i_1}) P_{k-1}(x_{i_2} \cdots x_{i_2}) + \cdots + \\ & N(i_n - i_{n-1}) P_{k-1}(x_{i_1} \cdots x_{i_{n-1}}) P_{k-1}(x_{i_n}) + \\ & N(n - i_n) P_{k-1}(x_1 \cdots x_{i_n}) - (n-1) P_{k-1}(x_1 \cdots x_n)] = \\ & (1 - \frac{N(i_n - i_1)p_c}{(N-1)(n-1)}) P_{k-1}(x_1 \cdots x_n) + \\ & \frac{N(i_n - i_1)p_c}{(n-1)(N-1)} \left\{ \frac{i_2 - i_1}{i_n - i_1} P_{k-1}(x_{i_1}) P_{k-1}(x_{i_2} \cdots x_{i_2}) + \cdots + \right. \\ & \left. \frac{i_n - i_{n-1}}{i_n - i_1} P_{k-1}(x_{i_1} \cdots x_{i_{n-1}}) P_{k-1}(x_{i_n}) \right\}. \end{aligned}$$

由引理及假设, 可知 $\lim_{k \rightarrow \infty} P_k(x_1 \cdots x_n) = P_0(x_1) \cdots P_0(x_n)$. 根据数学归纳法, 原命题成立.

从以上的讨论我们可以归纳得到如下定理:

定理 (交叉算子的极限性能) 在只有交叉算子的情形下: 1) 随着进化代数的增加, 模式内部各基因将趋于独立; 2) 随着进化代数的增加, 只要组成模式的各个基因都存在, 则该模式一定能够被搜索到; 3) 模式的极限概率等于组成该模式各基因的初始概率(也就是基因的极限概率)的乘积, 并且与模式的定义长度无关.

由此定理, 我们很容易得到如下推论:

推论 在只有交叉算子的情形下, 若群体中存

在两个染色体, 其每一位基因都不相同, 则随着进化代数的增加, 任意模式都能够被搜索到.

4 自交叉算子及算例(Self-crossover operator and numerical results)

在遗传算法进化了一段时间后, 群体往往会向搜索空间中包含较优解的区域收敛, 这时群体中会存在较多相同的染色体, 而传统的交叉算子在两个染色体相同时将失效, 从而降低了算法的搜索能力, 容易形成“早熟”. 为解决这一问题, 我们在前面讨论的基础上给出自交叉算子这一新算子, 其操作步骤如下:

1) 设染色体 X 被选中参与交叉, 则生成该染色体的伴随染色体 Y , 染色体 Y 的每一位都与 X 不同;

2) 染色体 X 和 Y 利用一般的一点交叉算子进行交叉运算, 生成新染色体 X' 和 Y' ;

3) 任取新染色体 X' 和 Y' 中的一个取代原染色体 X .

由自交叉算子的定义及前面的理论分析, 我们知道自交叉算子能够避免失效的可能性, 增强算法的搜索能力. 下面我们给出一些算例来检验新算子的有效性.

算例 1^[8] (背包问题) 给出 n 个尺寸为 $s_1, \dots, s_n$ 的物体和容量为 C 的背包, 要求找出 n 个物体的一个子集使其尽可能多地填满容量为 C 的背包. 该问题可表示为如下数学模型:

$$\begin{aligned} \max \quad & \sum_{i=1}^n s_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^n s_i x_i \leq C, \quad x_i \in \{0, 1\}. \end{aligned}$$

设 $s = \{253, 245, 243, 239, 239, 239, 238, 238, 237, 232, 231, 231, 230, 229, 228, 227, 224, 217, 213, 207, 203, 201, 195, 194, 191, 187, 187, 177, 175, 171, 169, 168, 166, 164, 161, 160, 158, 150, 149, 147, 141, 140, 139, 136, 135, 132, 128, 126, 122, 120, 119, 116, 116, 114, 111, 110, 105, 105, 104, 103, 93, 92, 90, 79, 78, 77, 76, 76, 75, 73, 62, 62, 61, 60, 60, 59, 57, 56, 53, 53, 51, 50, 44, 44, 42, 42, 38, 36, 34, 28, 27, 24, 22, 18, 12, 10, 7, 4, 4, 1\}$, $C = 6666$.

我们使用遗传算法来求解此问题, 所用参数同文献[8], 即: 群体规模 = 100, 染色体长度 = 100, 交叉率 = 0.88, 变异率 = 0.0088. 为得到合理的解, 我们使用如下的惩罚函数作为算法的适应值函数:

$$f(X) = \begin{cases} \sum_{i=1}^n s_i x_i, & \sum_{i=1}^n s_i x_i \leq C, \\ \sum_{i=1}^n s_i x_i - \alpha (\sum_{i=1}^n s_i x_i - C), & \sum_{i=1}^n s_i x_i > C, \end{cases}$$

其中惩罚因子 $\alpha = 1.1$.

算例 2 $\min f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$,
 $-2.048 \leq x_i \leq 2.048$.

该函数又称 Rosenbrock 函数,虽然它是单峰函数,但却是病态的且难以极小化.求解此题时所用参数为:群体规模 = 60,染色体长度 = 80,交叉率 = 0.95,变异率 = 0.01.

算例 3 $\min f(x) = x_1^2 + x_2^2 - 8\cos(2\pi x_1) -$

$8\cos(2\pi x_2)$.

这是一个多峰函数,局部极小点落在大小为 1 的矩形网格上,全局极小点为 (0,0),此时 $f = 0$;次最佳极小点为 $(\pm 1, 0)$ 和 $(0, \pm 1)$,此时 $f = 1$;随着到全局极小点距离的增加,局部极小点的目标函数值逐渐变大.求解此题时所用参数为:群体规模 = 30,染色体长度 = 40,交叉率 = 0.95,变异率 = 0.01.

为避免随机误差,我们从不同的初始群体开始求解这三个问题,共运行 20 次.为得到全局解,我们首先使用标准的遗传算法,然后在不改变其它任何参数及算子的情况下,仅仅用自交叉算子取代原来的一点交叉算子再次进行求解,统计结果如下:

表 1 20 次数值计算的统计结果

Table 1 The numerical results of the 20 runs over the problems

	算例 1		算例 2		算例 3	
	平均进化代数	平均计算时间	平均进化代数	平均计算时间	平均进化代数	平均计算时间
标准交叉算子	15.1	0.347	335.23	0.5535	7506.70	32.19
自交叉算子	14.2	0.314	120.15	0.2205	1968.21	8.46

从上面的统计结果我们可以看到,对所给的算例自交叉算子的使用对算法的性能有了明显的改善,改善幅度由 10% 至 75% 不等.

5 结论(Conclusion)

本文首先对交叉算子进行了详尽的分析,给出了染色体在交叉算子作用下的变化规律,并对交叉算子的极限性能进行了分析.通过这些分析提出了自交叉算子这一新的算子,实验显示算法的性能确实能够得到改善.

参考文献(References)

- [1] Kenchi F, Yoshikazu M, Koji T. Recursive ARMAX speech analysis based on a glottal source model with phase compensation [J]. Signal Processing, 1999, 74(3):279 - 295
- [2] Fumiaki T, Toshihiro N, Sigeru O. Banknote recognition by means of optimized masks, neural network, and genetic algorithms [J]. Engineering Applications of Artificial Intelligence, 1999, 12(2):175 - 184
- [3] Philippe S, Jean-Marc V, Thierry L, Reto M. Robust parameter estimation of intensity distributions for brain magnetic resonance images [J]. IEEE Trans. on Medical Imaging, 1998, 17(2):172 - 186
- [4] Ren Qingsheng, Ye Zhongxing. Genetic searching algorithm for opti-

mal runlength-limited codes with error control [J]. Journal of China Institute of Communications, 1997, 18(3):54 - 59

- [5] Ko Myung-Sook, Kang Tae-Won, Hwang Chong-Sun. Function optimization using an adaptive crossover operator based on locality [J]. Engineering Applications of Artificial Intelligence, 1997, 10(6):519 - 524
- [6] Correa R C, Ferreira A, Rebreyend P. Scheduling multiprocessor tasks with genetic algorithms [J]. IEEE Trans. on Parallel and Distributed Systems, 1999, 10(8):825 - 837
- [7] Kang Lishan, Chen Yuping. Non-Numerical Parallel Algorithm(II)—Genetic Algorithm [M]. Beijing: Science Press, 1995 (in Chinese)
- [8] Wang Xufa, Zhuang Zhenquan, Wang Dongsheng. Genetic Algorithm and Its Application [M]. Beijing: People's Post & Telecommunication Press, 1996 (in Chinese)

本文作者简介

任庆生 1972 年生,1998 年于上海交通大学获通信与电子系统专业博士学位,现为上海交通大学计算机系副教授.主要研究方向为计算智能和图像处理.

曾进 1970 年生,1998 年于上海交通大学获热力涡轮机械专业博士学位,现为上海交通大学应用数学系副教授.主要研究方向为数值计算和最优化控制.

戚飞虎 1938 年生,1961 年大学毕业后到上海交通大学任教,现为上海交通大学计算机系教授,博士生导师.主要研究方向包括图像处理与计算机视觉,人工神经网络与软计算,视频压缩与多媒体技术