

# 遗传算法优化速度的改进\*

杨启文, 蒋静坪, 张国宏

(浙江大学 电气工程学院, 浙江 杭州 310027)

E-mail: qwyang2k@263.net; eejiang@ dial.zju.edu.cn

http://www.zju.edu.cn

**摘要:** 分析了传统变异算子的不足, 提出用二元变异算子代替传统的变异算子, 并讨论了它在克服早熟收敛方面的作用. 同时, 针对二进制编码的遗传算法的特点, 提出了解码算法的隐式实现方案, 使得遗传算法的寻优时间缩短 6~50 倍. 实验从多方面对二元变异算子的遗传算法进行性能测试, 结果表明, 改进型算法收敛快, 参数鲁棒性好, 能有效地克服“早熟”收敛. 通过改进变异算子和解码算法, 遗传算法的优化速度得到了很大的提高.

**关键词:** 遗传算法; 优化速度; 二元变异算子; 早熟收敛

**中图分类号:** TP18 **文献标识码:** A

遗传算法 (genetic algorithms, 简称 GAs) 是近年来迅速发展的一种全新的优化算法<sup>[1]</sup>. 该算法由密执安 (Michigan) 大学教授 Holland 及其学生创建<sup>[2]</sup>, 其基本思想是基于 Darwin 的进化论和 Mendel 的遗传学说. 由于该算法具有很好的普适性, 不苛求问题的动力学信息 (如连续、可微等), 并且对非线性复杂问题显示出很强的求解能力, 因而被成功地应用于许多不同的领域<sup>[3]</sup>.

但是, 作为一种随机优化算法, 也存在着同其他优化算法相类似的不足之处. 例如, 为了寻求一个问题的最优解, 寻优程序有时不得不花费相当长的时间进行搜索. 造成这种现象主要有两方面的原因: (1) GAs 算法自身的优化能力有限, 不得不需要更多次的迭代运算才能找到最优解; (2) GAs 算法在实现中产生了大量的计算开支. 因此, 要提高 GAs 的优化速度, 就必须对上述两个方面进行改进.

## 1 遗传算法

根据编码方式的不同, GAs 大致可分为两类: 二进制编码和浮点编码. 基于二进制编码的 GAs 搜索能力强, 而且适合于模式识别、生产调度等这类非数值问题的优化, 因此应用范围广, 更具有普适性.

用 GAs 求解一个优化问题可以描述为

$$J = \max_{X \in \Omega} F(X), \quad X = (x_1, x_2, \dots, x_n)^T.$$

这里,  $F(X)$  为评价函数,  $X$  是问题所涉及的解向量,  $\Omega$  为解向量的定义域.

### 1.1 染色体的编码 (encode) 和解码 (decode)

GAs 对于不同类型的优化问题, 其求解过程基本相同. 但是, 由于变量的性质不同, 会导致染

\* 收稿日期: 1999-09-14; 修改日期: 1999-11-24

基金项目: 国家教育部博士点基金资助项目 (97033526); 浙江省自然科学基金资助项目 (598019)

作者简介: 杨启文 (1969-), 男, 湖北黄冈人, 博士生, 讲师, 主要研究领域为进化计算及其应用, 模糊控制, 神经网络技术; 蒋静坪 (1935-), 男, 浙江宁波人, 教授, 博士生导师, 主要研究领域为智能控制, 计算机控制; 张国宏 (1966-), 男, 江西南昌人, 工程师, 主要研究领域为计算机控制, 电机控制.

染色体在编码和解码上存在差异.

例如,  $F(X)$  有两个变量. 其中  $x_1$  需要回答“是”还是“不是”的问题;  $x_2$  是一个数值变量, 其定义为  $[a, b]$ , 精度要求  $\leq \epsilon$  ( $\epsilon$  为一给定的正数). 这里, 我们只需用一位基因“0”或“1”来完成对  $x_1$  染色体  $x'_1$  的编码.  $x_2$  由于存在精度要求, 因此需要确定其染色体  $x'_2$  编码长度  $l$ ,  $l$  通常可由下式给出:

$$l \geq \ln \left\lceil \frac{b-a}{\epsilon} + 1 \right\rceil.$$

这样, 解向量  $X$  染色体就可以由长度为  $l+1$  的二进制字符串  $x'_1, x'_2$  组成.

$x_1$  的解码十分简单. 实数  $x_2$  的解码一般可由下面的公式确定:

$$x_2 = a + (b-a) \frac{X_{10}}{2^l - 1}. \quad (1)$$

其中  $X_{10}$  表示二进制染色体  $x'_2$  所对应的十进制数.

## 1.2 杂交(crossover)

按一定的杂交概率  $p_c$  随机地从种群中选择两个个体(染色体), 部分地交换其中的某些基因位. 杂交方式通常有一点杂交、两点杂交、多点杂交和一致杂交. 杂交的作用在于产生新的基因组合, 即生成新染色体.

## 1.3 变异(mutation)

按一定的变异概率  $p_m$  对染色体中的基因进行变异操作. 目前变异的主要方式是取反操作, 其目的在于发现新基因, 克服早熟收敛.

## 1.4 选择策略(selection strategy)

按照某种方式从生成的子代中选择一定数量的个体作为下一代进化的初始种群. 选择策略通常包括: 轮盘赌、比例选择、锦标竞争和最优个体保留.

# 2 变异算子的改进

早熟收敛一直是 GAs 中存在的主要问题. 算法一旦出现早熟收敛, 将无法得到问题的最优解. 变异算子作为克服算法早熟的最有效手段, 极大地影响着 GAs 的优化速度.

## 2.1 传统变异算子的缺陷

**定义 1.** 对待优化问题的最优解而言, 其对应染色体上的每个基因称为该基因位上的有效基因.

**定义 2.** 同一基因位上等位基因的多样性是指在种群中, 染色体相同基因位的基因既有“0”又有“1”, 即在该基因位上, 基因全为“0”(或“1”)的概率  $P(“0”)$ (或  $P(“1”)$ )为

$$P(“0”) = 0 \quad \text{或} \quad P(“1”) = 0. \quad (2)$$

J. Craig Potts 等人认为, 产生早熟收敛主要是由有效等位基因的缺失<sup>[4]</sup>造成的. 由于杂交算子不会产生新的基因, 而选择策略目的在于加快基因的收敛过程, 这不可避免地会造成某一类基因在特定基因位上的比例下降, 导致该基因位上的基因缺损. 因此, 为了预防早熟收敛, 在有效基因未知的情况下, 变异算子必须有能力保持同一基因位上等位基因的多样性, 这样才有助于防止有效基因的缺失, 从而能够最大限度地避免早熟收敛.

**定理.** 传统的变异算子无法有效地保持同一基因位上等位基因的多样性.

证明:在一个规模为  $N$  的种群中,假设在染色体第  $j$  个基因位上有  $n_1$  个“0”和  $n_2$  ( $n_2 = N - n_1$ ) 个“1”,则该基因位上的所有基因经取反变异后变为同一个基因的概率为

$$\begin{cases} P(\text{"0"}) = (1 - p_m)^{n_1} p_m^{n_2} \neq 0 \\ P(\text{"1"}) = p_m^{n_1} (1 - p_m)^{n_2} \neq 0 \end{cases} \quad (3)$$

与式(2)矛盾。□

## 2.2 二元变异算子

传统的变异算子执行的是一元操作(取反),即操作数只需一个基因.对于由二进制字符串组成的染色体,我们将数字技术中的二元逻辑算子引入到 GAs 中,对传统的变异方式进行彻底“变异”,提出了一种全新的变异算子——二元变异算子:同或/异或.

与传统的取反变异不同,二元变异操作需要两条“染色体”参与,例如:

$$\begin{array}{r} 01101011 \quad 01000101 \quad \text{"同或"运算} \\ \quad \quad \quad \downarrow \\ 11010001 \quad 10111010 \quad \text{"异或"运算} \end{array}$$

由“同或/异或”的逻辑运算规律可知,运算后得到的两种逻辑状态互补,即一个为“0”,另一个必为“1”.也就是说,同一基因位上的基因经过变异之后,该基因位至少有一个“0”和一个“1”同时存在,而不会出现全“0”或全“1”的情况.因此,二元变异算子能够满足式(2)的条件.

基于二元变异算子的 GA 算法(genetic algorithm based on dyadic mutation operator,简称 GADMO)流程如下:

- (1) 种群初始化;
- (2) 遗传操作:从种群中选择一对“父母”,执行杂交和二元变异操作;
- (3) 计算适合值,若满足程序终止条件,则程序结束;
- (4) 转向第(2)步,完成本代整个种群的进化;
- (5) 转向第(2)步,进行种群下一代的进化.

## 2.3 实验测试

遗传算法作为一种随机优化算法,影响其性能的参数有很多,例如,初始化种群、种群规模、编码长度、遗传概率等.不同的参数设置会对问题的求解带来不同的影响,合适的参数有助于算法的快速收敛,而有些参数可能使 GAs 收敛显著变慢,有的参数则会导致算法“早熟”,因而得不到问题的最优解.但如何为 GAs 选择合理参数,至今还没有一个很好的解决办法.因此,参数对算法性能的影响程度不但反映了算法的好坏,而且也影响着算法的应用.本节从 3 个不同的侧面对基于二元变异算子的遗传算法进行参数敏感度的测试.

实验所用函数为

$$F_1 = 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6},$$

$$F_2 = (x_1^2 + x_2^2)^{0.25} [\sin^2(50(x_1^2 - x_2^2)^{0.1}) + 1.0].$$

实验 1. 种群规模和编码长度对算法性能的影响

遗传算法采用一点杂交方式,变异算子分别为传统的取反变异和改进型变异算子.测试函数为  $F_1$ .如果算法在 100 代内未能优化到阈值 1.0,则认为搜索失败.

实验结果见表 1.在 20 次不同的种群规模和编码长度的测试中,改进型遗传算法只有 1 次未

能在 100 代内找到阈值,而传统遗传算法有 7 次.传统的变异算子对种群规模比较敏感,种群越小,算法的成功率越低,而改进型变异算子能很好地维持算法在小种群时的收敛性.编码长度对算法的影响不大.

**Table 1** The influence of mutation operators on the convergence of GAs

**表 1** 变异算子对算法收敛性能的影响

Convergence generations (unimproved/improved) <sup>①</sup>	Encoding length of the independent variables <sup>②</sup>				
	12	14	16	18	
Population size <sup>③</sup>	20	F/77	F/96	F/F	F/72
	50	F/85	31/8	F/42	16/25
	80	F/14	31/13	79/37	29/39
	100	41/43	28/21	90/27	14/20
	120	13/22	17/25	21/28	38/12

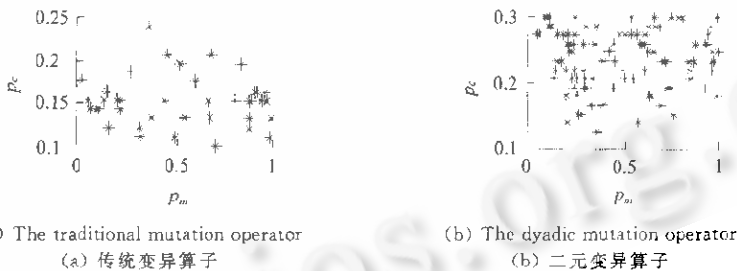
“F” denotes that the maximum of the function can not be found within 100 generations<sup>④</sup>.

①收敛代数(改进前/后),②自变量编码长度,③种群规模,④表中“F”表示在 100 代内算法能找到函数的最大值.

**实验 II. 变异算子对遗传概率选择范围的影响**

遗传算法参数:种群规模  $N=20$ ,自变量编码长度  $l=16$ ,杂交概率在  $0\sim 1.0$  之间变化,变异概率在  $0\sim 0.3$  之间变化;函数测试重复 5 次,每次初始种群均随机产生.若算法在 100 代内均能优化到阈值 1.0,则记录所用的参数  $(p_c, p_m)$ .

实验结果如下:改进前  $(p_c, p_m)$  为 39 组,改进后范围扩大了 2.5 倍,为 97 组(如图 1 中阴影部分所示);平均收敛代数也由改进前的 53.72 代降至 48.27 代.由此可见,二元变异算子不但增大了对遗传概率选择的自由度,而且也提高了算法的收敛速度.



**Fig. 1** The influence of mutation operator on the selection range of the genetic probabilities  
图 1 变异算子对遗传概率选择范围的影响

**实验 III. 3 种遗传算法的对比实验**

M. Srinvas 等人为解决 GAs 的早熟问题曾提出一种自适应遗传算法(AGA)<sup>[5]</sup>.我们将改进型遗传算法和基于标准遗传算子的固定遗传概率(SGA)、自适应遗传概率(AGA)的遗传算法进行实验比较.改进型遗传算法取  $(p_c, p_m) = (0.9, 0.17)$ ,种群规模均为 100. SGA 和 AGA 方案详见文献[5].

实验结果(见表 2)表明,改进型遗传算法收敛速度快,克服早熟的能力强.

**Table 2** The experimental results of three kinds of GAs

**表 2** 3 种遗传算法的实验结果

Number of functions <sup>①</sup>	Encoding length <sup>②</sup>	Average generations of convergence (sum of prematurity) <sup>③</sup>			Threshold <sup>④</sup>	Max. generation <sup>⑤</sup>
		SGA	AGA	GADMO		
F1	17	64.06(7)	36.63(0)	23.23(0)	1.0	100
F2	22	419.9(21)	220.51(5)	24.27(0)	0.995	500

①函数编号,②编码长度,③平均收敛代数(早熟次数),④阈值,⑤最大进化代数.

通过上述3组实验,我们不难发现:在实验范围内,二元变异算子大大增强了算法的寻优能力,在不同的参数设置下,一般只需较少代的进化就能优化到指定的阈值,早熟情况得到明显的改善;同时,改进型遗传算法的参数选择范围大,参数对算法性能的影响小,这为算法的实际应用提供了极大的方便。

### 3 解码算法的隐式实现

当需要对浮点变量进行求解时,GAs 往往存在如式(1)所示的解码算法.为了得到染色体  $x'_2$  的十进制数  $X_{10}$ ,不得不为其转换编制相应的算法,每一个浮点变量的解码过程至少需增加  $2l$  次耗时的乘法运算,从而延长了 GAs 寻找到最优解的时间.因此,用求解时间作为衡量指标,GAs 的实现算法也需要进一步优化。

基于二进制 GAs 的传统实现方式是将染色体用字符串来表示,其相应的遗传操作也都是基于字符串间的相关运算.在数字计算机上,操作系统中的任何整数在其物理内存中都是以二进制数的形式存储的,其数据格式的转换由计算机硬件和操作系统协作完成,无须编程实现.利用计算机的这一特点,将整数  $X_{10}$  在物理内存中的数据格式作为染色体  $x'_2$  的编码信息,这样,其物理内存中的二进制数经过遗传操作后,其解码后的结果就直接反映到变量  $X_{10}$  中,从而实现了解码算法由传统的显式转换变成隐式转换.同时,一个基因也由原来的 8 bits 字符表示降至 1 bit 位表示,节省了不内存空间。

我们用  $F1$  作为测试函数,GAs 进化到 500 代终止.实验结果见表 3.经过对代码的优化,在相同的实验条件下,GAs 程序的运行时间缩短了 6~50 倍,而且这种优势随着种群规模和编码长度的增大更加明显。

Table 3 Comparison of run time with unimproved and improved program

表 3 优化前、后程序运行时间对照表

Run time (s) (unimproved/improved) <sup>①</sup>	Encoding length of the independent variables <sup>②</sup>				
	10	15	20	25	
30	7.79/2.64	24.61/2.69	29.39/2.86	36.61/3.02	
60	57.62/5.16	82.34/5.38	104.63/5.55	128.25/5.88	
Population size <sup>③</sup>	90	119.79/7.75	171.75/8.07	224.04/8.46	277.54/8.79
	120	205.23/10.33	297.37/10.76	386.46/11.26	481.48/11.7
	150	315.61/12.91	455.49/13.46	581.49/14.06	743.86/14.61

①运行时间(s)(程序优化前/后),②自变量编码长度,③种群规模。

### 4 结束语

本文利用二元变异算子改进传统的变异方式,有效地克服了早熟收敛,提高了 GAs 的优化速度,增强了 GAs 对复杂问题的求解能力,同时,针对 GAs 在求解数值解时存在的不足,提出了解码算法中的隐式实现方案,极大地缩短了程序的执行时间.通过上述两个方面的改进,GAs 作为一种优化工具,其优化能力更强,速度更快。

### References:

- [1] Zhang, Xiao-hui, Dai, Guan-zong, Xu, Nai-ping. Genetic algorithms—a new optimization and search algorithms. Control Theory and Application, 1995, 12(3): 265~273 (in Chinese).
- [2] Holland, J. H. Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University Michigan Press, 1975.
- [3] Xi, Yu-geng, Cai, Tian-you, Yun, Wei-min. Survey on genetic algorithm. Control Theory and Application, 1996, 13(6):

696~704 (in Chinese).

- [4] Potts, J. C., Terri, D. G., Surya, B. Y. The development and evolution of an improved genetic algorithm based on migration and artificial selection. *IEEE Transactions on SMC*, 1994, 24(1):73~86.
- [5] Srinivas, M., Patnaik, L. M. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on SMC*, 1994, 24(4):656~666.

附中文参考文献:

- [1] 张晓继,戴冠中,徐乃平. 一种新的优化搜索算法——遗传算法. *控制理论与应用*, 1995, 12(3):265~273.
- [3] 席裕庚,柴天佑,恽为民. 遗传算法综述. *控制理论与应用*, 1996, 13(6):697~704.

## Improving Optimization Speed for Genetic Algorithms\*

YANG Qi-wen, JIANG Jing-ping, ZHANG Guo-hong

(*Electrical Engineering School, Zhejiang University, Hangzhou 310027, China*)

E-mail: qwyang2k@263.net; eejiang@diaf.zju.edu.cn

<http://www.zju.edu.cn>

**Abstract:** The disadvantage of the traditional mutation operator of GAs was analyzed in this paper, and a DMO (dyadic mutation operator) was presented to take the place of the traditional one. The function of DMO to prevent premature convergence was also discussed. Meanwhile, according to the features of binary-based GAs, an implicit implementation for decoding the chromosomes for GAs was presented so that the run time of the improved program for GAs was shortened by 5~50 times compared with the original one. The performance of the genetic algorithm is tested based on the DMO (GADMO) in several aspects. The experimental results show that the GADMO can converge quickly and its robustness of parameters is strong. The GADMO can prevent the premature convergence effectively. By improving the mutation operator and the decoding algorithm, the optimization speed of GA is speeded up greatly.

**Key words:** genetic algorithm; optimization speed; dyadic mutation operator; premature convergence

\* Received September 14, 1999; accepted November 24, 1999

Supported by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 97033526; the Natural Science Foundation of Zhejiang Province of China under Grant No. 598019