改进遗传算法搜索性能的大变异操作*

马钧水 刘贵忠 贾玉兰
(西安交通大学电信学院信息与通信工程研究所·西安,710049)

摘要:遗传算法是一种模仿自然界生物进化过程中选择和遗传的机理而构造出的一种优化搜索算法.但是,简单遗传算法的收敛速度较慢,稳定性较差.针对这些问题,本文提出了一种被称为"大变异"的改善遗传算法性能的操作,在文中分别讨论了该操作的思路,实现的方法,并给出了它的有效性的数值实验证明.

关键词:遗传算法;函数优化;大变异操作

1 引 言

遗传算法 GAs(Genetic Algorithms)产生于一些生物学家用计算机模拟生物进化过程的仿真实验.1960 年,美国的 J. H. Holland 把它创造性地应用于人工系统,并成功地利用它解决了一些实际问题,例如"旅行商问题".从此,它的实用价值逐渐被人们所认识.特别是近十年,由于计算机性能的提高,以及并行分布式计算的推广,GAs 由于自身独特的优势而越来越受到人们的重视[1,2].

但是,简单遗传算法 SGAs(Simple Genetic Algorithms)由于自身固有的缺陷,通常优化过程的收敛速度较慢,而且算法稳定性较差.本文针对这些问题提出了一种改进现有 SGAs 性能的操作——大变异操作 BMO(Big Mutation Operation).为了便于随后的讨论,我们将在第二节简单介绍 SGAs 的实现方法,在第三节将就 BMO 的基本思想和实现方法展开讨论,最后将给出该操作有效性的数值证明.

本文的讨论主要在遗传算法函数优化问题这一应用领域展开,同样的思路也适用于其它应用领域.另外,为了简化问题,我们对变量空间点统一采用二进制编码方式,而对于其它的编码方式,文章提出的基本思想也同样是有效的.

2 简单遗传算法

设待优化函数为 $g(X)$,X 是一个向量,它所有的可能的取值构成了该优化问题的变量空间.首先我们必须由 $g(X)$ 构造出一个适应度函数 $f(X)=h(g(X))$,该适应度函数的值域 $V\subset\mathbb{R}^+$ ($\mathbb{R}^+$ 表示全体非负实数的集合),并且要求它在 $g(X)$ 取得最优值的点取得最大值.然后,我们就可以利用 GAs 在待优化问题的变量空间中搜索到一个使 $f(X^*)$ 取得最大值的点 X^*,而该点也就是 $g(X)$ 的最优点.

算法具体的实现步骤如下[1~3]:

1) 确定遗传算法的参数:总个体数 N,交叉概率 P_c,变异概率 P_m,代沟 G.其中代沟 $G\in[0,1]$,它表示父代与子代有 $(N^*(1-G))$ 个个体重叠.

2) 初始化 N 个个体,并计算每个个体的适应度 F_i,并根据参数 G 确定两代之间重叠的个体数 $N_{overlap}$.

3) 利用遗传算法的三个基本操作,以及参数 P_c 和 P_m,对当前代的个体进行繁殖,产生(N

* 国家教委博士点基金项目(教技管中心类 1993 年第 11 号)和陕西省自然科学基金项目(96809)资助课题.
本文于 1996 年 5 月 6 日收到.1997 年 5 月 27 日收到修改稿.

$-N_{overlap}$) 个新一代的个体. 具体而言，首先利用选择复制(Reproduction) 操作从当前代选出两个个体，第 j 个个体被选中的概率是：

$$Pr(j)=F_j/S,\quad \text{其中}\quad S=\sum_{i=1}^{N}F_i,\quad j=1,2,3,\cdots,N. \tag{1}$$

然后，对选出的个体进行交叉互换(Crossover)操作，最后，再对交叉互换后的个体进行基因变异(Mutation)操作，便得到了新一代的两个个体.

4) 从当代选出 $N_{overlap}$ 个最高适应度的个体和 3) 中产生的($N-N_{overlap}$) 个新个体组合成下一代(子代)，以保持总个体数 N 不变.

5) 如果达到设定的繁衍代数，则返回当前代适应度最高的个体所对应的变量空间的点 X，作为优化结果；否则，回到 3) 继续繁衍下去.

3　大变异操作

3.1　大变异操作的思想

众所周知，简单遗传算法存在一个“早熟”的问题，也就是算法过早收敛于一个非全局最优点. 出现这种问题的主要原因是由于一种被称为“顶端优势”的现象存在，即当算法进行到某一代时，在种群中某个个体 m 的适应度远远大于其它任何个体的适应度. 因为个体被选中进行复制的概率由式(1) 确定，这样就会造成子代中许多个体来自同一祖先 m，从而彼此近似. 这种现象的极限情况就是所有个体来自同一祖先，这就是我们常说的“早熟”. 一旦出现“早熟”，SGAs 中的选择和交叉操作就会失效.

理论上，SGAs 中的变异操作可以使算法跳出“早熟”. 但是，为了保证算法的稳定性，变异操作的变异概率通常取值很小(例如 0.01)，所以算法一旦出现“早熟”，单靠传统的变异操作需要很多代才能变异出一个不同于其它个体的新个体 m. 而且，如果新个体 m 的适应度 F_m 满足以下关系

$$F_m \ll F_s,\quad \text{其中}\quad F_s=\sum_{i=1}^{N}F_i.$$

由式(1)可知，新个体 m 在进行下一次选择操作时被选中的概率会远远小于 1. 目前，对于这个问题已有的解决方法是采用对个体的适应度进行一个称为尺度化(Scaling)[2]的操作. 但是，这种方法改变了原问题的适应度函数，会在很大程度上影响遗传算法的收敛速度.

大变异操作就是在这样的背景下提出的. 它的思路是：当某代中所有个体集中在一起时，我们以一个远大于通常的变异概率的概率执行一次变异操作，具有大变异概率的变异操作(即“大变异操作”)能够随机、独立地产生许多新个体，从而使整个种群脱离“早熟”.

这个思路在实现时有两个要解决的问题：1) 如何判断某代中所有个体集中的程度？我们采用比较每代中所有个体的最大适应度 F_{max} 和平均适应度 F_{avr} 的接近程度的方法来判断，因为通常某代的最大适应度 F_{max} 和平均适应度 F_{avr} 越接近，该代中个体就越集中. 2) 如何保证前代具有最大适应度的个体不被大变异操作破坏掉？我们采用的策略是：在当代所有适应度等于最大适应度的个体中，选择一个个体，对它不进行大变异操作.

此外，在进行大变异操作前，我们还将所有个体都设为具有最大适应度的个体的形式. 进行这个操作的原因是对于一个未知的待优化问题，在一个较优的点的附近常常会以较大的概率出现一个更优的点. 逆向思考就是，最优点的附近的点也较优. 我们可以用以下的变异概率转移矩阵 M^*[4] 来描述变异操作：

$$M=\{m_{ij}\}_{m\times n},\quad \text{其中}\quad m_{ij}=P_m{}^{\rho(ij)}(1-P_m)^{L-\rho(i,j)}. \tag{2}$$

其中 L 是个体长度，P_m 是变异概率，$\rho(i,j)$ 是 i 号个体** 与 j 号个体间的 Haming 距离. 我们在图 1 中绘出了在 $L=6, P_m=0.12$ 时，个体 010101 和个体 111100 进行变异操作时的概率转移曲线. 由曲线和式(2) 可以得出：当变异概率小于 0.5 时每个个体以较大的概率变异为与它“临近”的个体. 我们把所有个体设为当代最大适应度的个体，再对设定后的个体进行大变异操作的结果是：新个体会以很大的概率出现在当代最大适应度的个体的“临近”，这正是我们进行优化时所期望的.

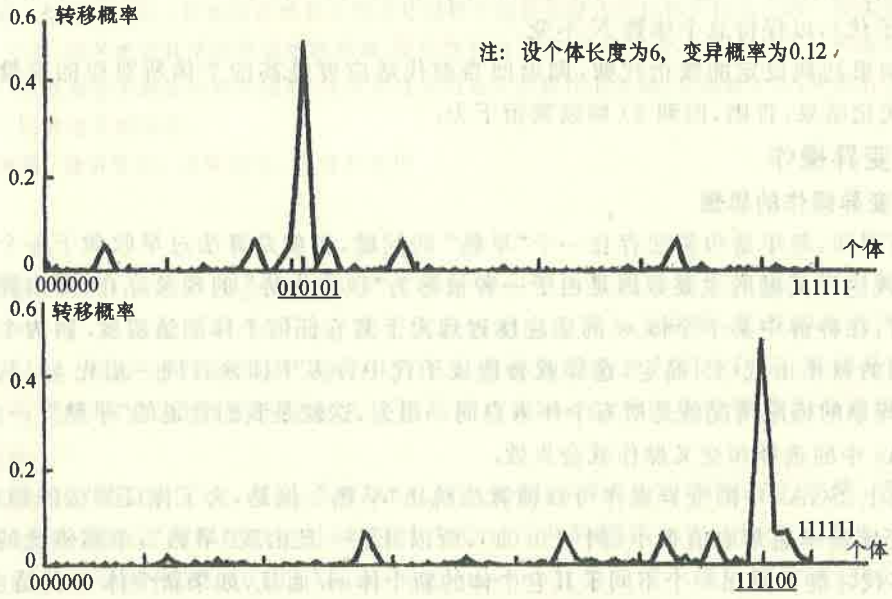

图 1 个体 010101 和个体 111100 的概率转移曲线

3.2 大变异操作的实现方法

基于以上的讨论，大变异操作由两个子操作组成——集中和打散.

当某一代的最大适应度 F_{max} 与平均适应度 F_{avr} 满足：

$$a * F_{max} < F_{avr},$$

其中 $0.5 < a < 1.0$，被称为密集因子，表征个体集中的程度. 我们就将该代中所有个体设为具有最高适应度个体的形式，这就是“集中”. 随后，我们以一个比通常变异概率大 4 倍以上的概率 P_{big} 对集中了的参数进行一次变异操作. 这就是“打散”.

大变异操作要求有两个参数是：密集因子 a，大变异概率 P_{big}. 密集因子 a，用来决定大变异操作在整个优化过程中所占的比重. a 越接近 0.5，大变异操作被调用得越频繁. 大变异概率 P_{big} 越大，含大变异操作的遗传算法(即“大变异遗传算法”) 的稳定性就越好，但是，这是以牺牲收敛速度为代价的. 当 $P_{big}=0.5$ 时，大变异操作就近似蜕化成为随机搜索. 这一点可以从式(2) 得到.

3.3 大变异操作效率实验证明

下面我们以数值实验的方法来验证大变异操作的有效性.

我们采用 Dejong 提出的三个用于检测遗传算法性能的目标函数[5,6]，它们如表 1 所示.

---

* 变异概率转移矩阵 M；$A = B * M$ B 是进行变异操作前的所有个体在变量空间的分布情况，A 是进行变异操作后的所有个体的分布情况. 分布情况是指某个个体在空间某个位置出现的概率值.

** 个体的编号是个体的二进制编码所对应的十进制数.

表 1　实验采用的目标函数

函数号	函数表达式	变量取值范围	个体编码长度	优化阀值*
F1	$f(X)=\sum_{i=1}^{3}X_i^2$	$[-5.12,5.12]$	30	75.0
F2	$f(X)=100(X_1^2-X_2)^2+(1-X_1)^2$	$[-1.048,2.048]$	30	3850
F3	$f(X)=\sum_{i=1}^{5}[X_i]$	$[-5.12,5.12]$	50	50.0

* 优化阀值是指当适应度函数取得该值后，我们就认为整个优化过程已经达到了优化目标.

简单遗传算法的四个参数设为：个体数 $N=50$，交叉概率 $P_c=0.95$，变异概率 $P_m=0.01$，代沟 $G=49/50$. 大变异操作的两个参数是：密集因子 $a=5/6$，大变异概率 $P_{big}=0.04$.

我们分别利用 SGAs 和大变异遗传算法对待优化的三个目标函数进行优化，以优化 50 次作为一次实验，将优化到目标函数的优化阀值所需要的平均数作为衡量算法收敛速度的标准. 如果算法繁殖 300 代还未优化到目标函数的阀值，则该次优化失败，将这两种算法在 50 次中失败的总数作为衡量算法稳定性的标准. 实验结果如表 2 所示. 从实验结果我们可以看到，有大变异操作的遗传算法较简单遗传算法不但在收敛速度上有 50%～110%的提高，而且，算法的稳定性也有显著的提高.

表 2　实验结果

函数号	简单遗传算法		大突变遗传算法		改进率
	平均进化代数	未收敛次数	平均进化代数	未收敛次数	收敛速度%
F1	30.83	4	18.48	1	69.45
F2	32.46	4	16.00	3	108.41
F3	19.75	5	12.13	1	53.73

4　结　论

实验表明，大变异遗传算法正如我们所分析的，对于传统的遗传算法在性能(收敛速度和稳定性)上有显著地改善. 但是，从理论上定量的分析改进的程度仍有相当大的困难. 而且，对于不同的优化问题它的改进程度也不同. 这些更具体的工作还有待进一步研究.

参 考 文 献

1　刘健庄，谢维信，黄建军，李文化. 聚类分析的遗传算法方法. 电子学报，1995，23(1)：81－83

2　Goldberg. D.. Genetic Algorithms in Search, Optimization and Machine Learning. USA: Addison-Wesley Publising Company, 1988, 7-10, 59－308

3　Holland, J.. Adaption in Natural and Artificial System. Ann Arbor: University of Michigan Press, 1975

4　Eiben, A., Aarts, E. and Van Hee, K.. Global Convergence of Genetic Algorithm: An Infinite Markov chain analysis. Parallel Problem Solving from Nature, H. P. Schwefel and R. Nanner, Eds, Heidelberg. Berlin: Springer-Verlag, 1991, 4－12

5　DeJong, K. A.. The Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Dissertation, University of Michigan, Michigan, 1975

6　章珂，刘贵忠. 交叉位置非等概率选取的遗传算法. 信息与控制. 1997，26(1)：53－60

The Big Mutation: An Operation to Improve the Performance of Genetic Algorithms

MA Junshui, LIU Guizhong and JIA Yulan

(Institute for Information & Communication Engineering, Xi'an Jiaotong University · Xi'an, 710049, PRC)

Abstract: With the increasingly wide applications of genetic algorithms (GAs), its instinct drawback, the extraordinarily slow convergence, becomes the biggest obstacle of its further acceptation in many application fields. To combat this problem, a kind of operation, named big mutation, is discussed in this paper. After giving some explanation of the background and implementation of this operation, this paper provides the results of a numerical experiment to prove its efficiency and stability.

Key words: genetic algorithms; function optimization; big mutation operation

本文作者简介

马钧水 1971年生. 西安交通大学电子与信息工程学院硕士研究生.

刘贵忠 1962年生. 获荷兰 Einhoven 大学博士, 现任西安交通大学电子与信息工程学院教授, 博士生导师, 信息工程研究所副所长.

贾玉兰 1950年生. 西安交通大学信息工程研究所高级工程师.