基于遗传算法的 Rosenbrock 函数优化问题的研究*

*梁艳春 #周春光 ※李寿范

*(吉林大学数学系 长春 130023)

*(中国科学院沈阳计算所现代制造 CAD/CAM 开放实验室 沈阳 110003)

#(吉林大学计算机科学系 长春 130023)

※(86003 部队 长春 130022)

摘要 本文利用遗传算法研究了极小化 Rosenbrock 函数的问题. 较多的计算机模拟实验结果表明,遗传算法可以有效地解决这一问题. 文中还分析了一些改进的遗传算法对于该问题搜索速度的影响,得到了适于解决此问题的合理的遗传操作,从而为有效地解决最速下降法所不能实现的某一类函数优化问题提供了一种新的途径.

关键词 遗传算法,函数优化,最速下降法,Rosenbrock 函数,搜索速度.

中图法分类号 TP18

无约束的函数优化是应用范围很广的一类优化问题. 随着实际需要的日益增加和电子计算机的广泛普及,对这类问题的研究日渐深入. 迄今为止,人们已经提出许多无约束最优化方法,如使用导数的梯度法(最速下降法)、牛顿法、共轭梯度法和变尺度法等以及不使用导数的各种搜索法.

在函数极小化问题中,目标函数等高面内经常出现(至少在局部范围内)"超山谷". 对于具有两个独立变量的目标函数,在由目标函数等值线绘出的曲线图中则表现为"山谷". 对任何一个成功的最优化方法的主要要求之一是具有能够在局部区域中沿着窄的山谷迅速移向目标函数极小值的能力,即一个好的算法应给出一个沿山谷有较大分量的搜索方向. 由 Rosenbrock[1]设计的 Rosenbrock 函数是考察优化方法是否具有这一能力的一个典型例子. [2,3]理论与数值研究结果表明,人们广泛采用的、尤其是在远离极小点时颇具优势的最速下降法对于极小化 Rosenbrock 函数是不成功的. [2,3]鉴于此,我们做了利用遗传算法解决这一问题的尝试,并考察了一些改进的遗传算法对于该问题搜索速度的影响,得到了适于解决此问题的合理的遗传操作,从而为有效地解决最速下降法所不能实现的某一类函数优化问

* 本文研究得到国家自然科学基金和国家教委符号计算与知识工程开放研究实验室资助. 作者梁艳春,1953 年生,副教授,主要研究领域为人工神经网络,遗传算法及其应用. 周春光,1947 年生,教授,主要研究领域为遗传算法、分布式神经计算. 李寿范,1954 年生,副教授,主要研究领域为参数识别,优化设计.

本文通讯联系人:梁艳春,长春 130023,吉林大学数学系

本文 1996-11-05 收到修改稿

题提供了一种新的途径.

1 最速下降法用于极小化 Rosenbrock 函数的讨论

首先概述利用最速下降法极小化目标函数 $f(x)$ 的问题，其中 $x=(x_1,...,x_n)^T$ 为 n 维矢量. 在任一点 x 处，目标函数 $f(x)$ 的梯度是 $f(x)$ 最大局部增长方向上的一个矢量. 于是 $f(x)$ 的梯度的反方向即为最速下降方向，因为在 x 点处，$f(x)$ 的负梯度指向 $f(x)$ 关于 x 的每个分量减小最快的方向，并与 $f(x)$ 在 x 处的等值线正交.

在最速下降法的第 k 步，由一点 $x^{(k)}$ 到另一点 $x^{(k+1)}$ 的移动可由公式

$$x^{(k+1)}=x^{(k)}+\triangle x^{(k)}=x^{(k)}+\lambda^{(k)}S^{(k)} \tag{1}$$

给出，其中

$$S^{(k)}=-\frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|} \tag{2}$$

为 $f(x)$ 在 $x^{(k)}$ 处的单位负梯度，即最速下降方向，$\lambda^{(k)}$ 为待定因子，由公式

$$\frac{df(x^{(k)}+\lambda^{(k)}S^{(k)})}{d\lambda^{(k)}}=0 \tag{3}$$

决定.

下面我们来分析利用最速下降法极小化 Rosenbrock 函数的过程中所遇到的问题.

Rosenbrock 函数定义为

$$f(x)=100(x_2-x_1^2)^2+(1-x_1)^2, \tag{4}$$

它的两条等值线（$f(x)=4$ 和 $f(x)=8$）示于图1中. 从几何上看，$f(x)$ 可以解释为一个缓慢地向下弯曲的“山谷”，以 $x^*=(1,1)^T$ 为其最低点，在该点有 $f(x^*)=0$.

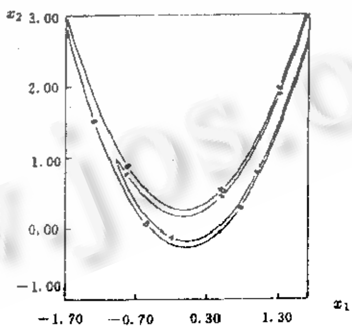


图1 Rosenbrock 函数的等值线图

由于 Rosenbrock 函数的响应曲面上有一条弯曲而又狭长的山谷，而在山谷中的点的最速下降方向几乎与通向 $f(x)$ 极小值的最好方向垂直，所以在山谷中的曲线周围，最速下降法收敛得极慢，甚至在合理的时间内完全不收敛.[2]

最速下降法是无约束优化问题中经常采用的方法之一，尤其在远离极小点时该方法颇具优势. 但在处理目标函数响应曲面上具有狭长山谷的函数优化问题中，最速下降法是不适用的. 为此，下面我们采用遗传算法来解决最速下降法所不能实现的这类函数优化问题.

2　遗传算法简介

遗传算法 GA(genetic algorithm)是由美国学者 Holland 于 1975 年首次提出的.[4]它基于达尔文适者生存、优胜劣汰的进化原则,对包含可能解的群体反复使用遗传学的基本操作,不断生成新的群体,使种群不断进化,同时以全局并行搜索技术来搜索优化群体中的最优个体,以求得满足要求的最优解. 与其它搜索算法,如随机查找、梯度下降、模拟退火等方法比较,GA 的主要优点是简单、鲁棒性强. 由于 GA 实现全局并行搜索,搜索空间大,并且在搜索过程中不断向可能包含最优解的方向调整搜索空间,因此,易于寻找到最优解或准最优解. 和其它方法相比,需要解决的问题越复杂,目标越不明确,GA 的优越性越大. 近年来,GA 在组合优化求解、机器学习、人工生命等领域已显示了它的应用前景和潜力,因而已成为国内外十分热门的研究课题.

2.1　基本概念

GA 操作的基本对象是染色体或个体(Chromosome),每个染色体是一个知识结构,代表求解问题的一个可能解,染色体通常用字符串或位串来表示,若干长度的串称之为构成染色体的基因(Gene).

群体或种群(Population)是由一组染色体所构成,它描述 GA 搜索的遗传空间. 在搜索过程中,用适应度函数(Fitness Function)来评价每个染色体的优劣,其值越大(适应度大),相应染色体所代表的解越优. 适应度函数的选择是能有效地指导搜索空间沿着面向优化参数组合方向逐渐逼近最佳参数组合,而不会导致不收敛或陷入局部最优.

2.2　基本操作

编码(Encoding)操作是将问题解的表示变换成遗传空间解的表示,即用字符串或位串构造染色体,其相反操作为解码(Decoding).

选择(Selection)操作是根据各染色体的适应度,在群体中按一定概率选取可作为父本的染色体,选择的依据是适应度大的染色体被选中的概率大.

交换(Crossover)操作是按照一定概率随机地交换一对父染色体的基因以形成新的子染色体.

变异(Mutation)操作是按照一定概率随机地改变染色体的基因值.

2.3　基本算法

设 $s_i(i=1,2,\ldots,m)$ 是字母表 0,1 上的定长串,其长度为 n,每位称为一个基因. 集合 $P(t)=s_1,s_2,\ldots,s_m$ 称为群体. 根据适应度函数可求得串 s_i 的适应度,记为 $F_i=F(s_i,t)$. 时刻 t 作用在 $P(t)$ 上的遗传算法[5]如下

(1) 计算 $P(t)$ 中的平均适应度 $F(t)$,将串 s_i 的适应度归一化为 $F_i/F(t)$;

(2) 依据归一化后的适应度值,赋给每个串(染色体)一个生存概率,按这一概率选择 k $(k\ll m)$ 对串,并将其复制;

(3) 对每一对被选出的串实行交换和变异操作,形成 $2k$ 个新串(子染色体). 交换操作为随机选择一位置 $r(1\leqslant r<n)$,交换两个串位置 r 右边的部分;变异操作为随机选择一位置 $r(1\leqslant r\leqslant n)$,对第 r 位取反;

(4) 在 $P(t)$ 和 $2k$ 个新串中淘汰掉适应度低的 $2k$ 个串，形成新的群体 $P(t)$；

(5) 若满足中止条件(某一染色体的适应度 F_i 达到预先给定的适应度估计值，或运算达到容许迭代次数)，则解码 $P(t)$ 中适应度最大的染色体得到问题的解. 否则 $t \leftarrow t+1$，返回(1).

2.4 若干改进的基本操作

由于GA在运行过程中需要较长的搜索时间，妨碍了其实际应用，因此，改进GA的基本操作一直是人们关注的课题，这方面的研究已经取得了相当多的成果，并且针对各种不同类型的问题，达到了提高GA运行效率的目的. 本节简要介绍在下节极小化Rosenbrock函数时采用的某些改进的基本操作.

2.4.1 线性选择[6]

线性选择是根据群体 P 中各染色体的适应度，按照一定的概率选出一定数量的染色体，并将它们拷贝到基因池 T 中. 设 P 和 T 中染色体数目各为 m，则基因池将由 P 中染色体 s_i 的 n_i 个拷贝构成，其中

$$n_i=\frac{F_i}{\sum_{j=1}^{m}F_j}m. \tag{5}$$

若 n_i 不是整数，则进行舍入计算使

$$\sum_{i=1}^{m}n_i=m. \tag{6}$$

2.4.2 非线性选择[6]

在产生基因池时引入某些非线性函数，如采用 S-型函数

$$S(F_i)=\begin{cases}0, & \text{当 } F\leqslant F_{\min}\text{ 时}\\ \dfrac{2(F_i-F_{\min})^2}{(F_{\max}-F_{\min})^2}, & \text{当 } F_{\min}\leqslant F_i\leqslant(F_{\min}+F_{\max})/2\text{ 时}\\ 1-\dfrac{2(F_{\max}-F_i)^2}{(F_{\max}-F_{\min})^2}, & \text{当 }(F_{\min}+F_{\max})/2\leqslant F_i\leqslant F_{\max}\text{ 时}\\ 1, & \text{当 } F_i\geqslant F_{\max}\text{ 时}\end{cases} \tag{7}$$

其中 $F_{\max}$ 和 $F_{\min}$ 分别为预先估计的染色体的最大和最小的适应度值. 取

$$n_i=\frac{S(F_i)F_i}{\sum_{j=1}^{m}S(F_j)F_j}m. \tag{8}$$

2.4.3 单点交换[4,7]和多点交换[6]

单点交换是随机选择一位置 $r(1\leqslant r<n)$，交换两个父本的位置 r 右边的部分，产生两个子代.

多点交换是把长度为 n 的染色体划分为 q 个长度为 L 的子串 $s_i(i=1,2,...,q)$. 产生 q 个在 $[1,L-1]$ 区间内的随机数 r_i，将 r_i 作为子串 s_i 的交换点，对每个子串实行单点交换.

2.4.4 均匀交换[8]

对于被选作父本的染色体随机产生一个同样长度的二进制串，对应于该串中出现"1"的位置，两个父本交换相应的基因.

2.4.5 有指导的变异[9]

设 s_t^* 为第 t 代诸染色体中最适合者，F_t^* 为相应的适应度值，设 $x_t^*=(x_{t_1},x_{t_2},...,x_{t_n})^T$ 为 s_t^* 的解码，若在第 $\tau(\tau>t)$ 代最适合的染色体为 s_τ^*，相应的适应度值为 $F_\tau^*(F_\tau^*>F_t^*)$，则变异操作产生的一个新的点由下式决定

$$x^* = x_\tau^* + \alpha(1 - t/t_{\max}) g(x_t^*, x_\tau^*) \tag{9}$$

其中 $\alpha(0<\alpha<1)$ 为常数因子，τ 代表在第 t 代后最适应的染色体达到 $F_\tau^* > F_t^*$ 的第 1 代，$t_{\max}$ 为迭代的最大次数. 若 $F(x)$ 处处可微，则 $g(x_t^*, x_\tau^*)$ 可取为

$$g(x_t^*, x_\tau^*) = F'(x_\tau^*). \tag{10}$$

3　模拟实验结果与分析

针对常规遗传操作和某些改进的遗传操作，我们对于极小化 Rosenbrock 函数问题进行了较多的模拟实验和分析对比. 除了成功地进行了利用遗传算法解决这一问题的尝试以外，我们还考察了一些改进的遗传算法对于该问题搜索速度的影响，得到了适于此问题求解的遗传操作方式，从而为解决最速下降法所不能实现的某一类函数优化问题提供了一种新的、有效的途径.

首先将 x_1, x_2 编码为 $\{0,1\}$ 上的位串，它由 $q=2$ 个子串构成，每个子串长度为 $L=16$，子串中有 1 个符号位，4 个整数位，其余位表示小数部分. 群体内染色体数目取为 $m=24$. 每次选取 2 个染色体作为父本，变异率取为 0.1，适应度函数取为

$$F(x) = \frac{1}{1+f(x)}, \tag{11}$$

其中 $f(x)$ 由(4)式定义.

(1) 选择操作

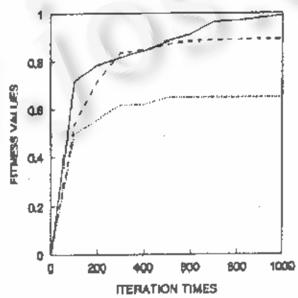
在父本选择中，我们比较了随机选择、线性选择与非线性选择三种选择操作. 实验结果表明，对于 Rosenbrock 函数极小化问题，随机选择是最理想的. 尽管线性选择与非线性选择的出发点是好的，它们都可以使得那些适应度值较高的染色体在群体中拥有较多的子代，并且它们已经被证明在处理某些优化问题时是卓有成效的[6]，但在极小化 Rosenbrock 函数的过程中，这两种方法并不能有效地加快搜索速度，反而会导致在有效时间内达不到理想的结果. 在模拟实验中我们观察到，执行几百代运算后，种群中的染色体都相差无几，适应度值也非常接近. 然而整个种群却距离最优点还相差甚远，此后在较长时间内整个种群基本上不再发生变化.

通过分析 Rosenbrock 函数的特殊性，我们可以得到线性选择与非线性选择失效的原因. 由于 Rosenbrock 函数在山谷内的一个很小范围内(与谷底轴向相垂直的方向上)函数值变化很大，因此初始种群中适应度值最大的染色体很可能与其它的染色体的适应度值相差若干个数量级. 这样，经过线性或非线性选择，该染色体被选中的概率特别大，甚至在较长时间内几乎都是它自身的相互交换. 这样，就使得相差无几的染色体迅速扩张，充满整个种群. 它所导致的后果是：① 整个种群的适应度值较为接近，使得线性与非线性选择失去作用；② 在以后的交换过程中，相同基因交换的概率增大；③ 交换后所产生的子代与种群中已有的染色体相同的概率增大. 这些结果将导致搜索陷入局部极小或过早收敛.

鉴于线性与非线性选择不适于极小化 Rosenbrock 函数问题，我们采用随机选择操作. 这种选择操作虽然不能使得整个种群的平均适应度迅速提高，但它可以保证整个基因池中有较丰富的基因供交换使用，从而避免陷入局部极小.

(2) 交换操作

在交换操作中,我们比较了单点交换、多点交换与均匀交换三种模式.实验结果表明,对于 Rosenbrock 函数极小化问题,单点交换并不能在短时间内达到良好的效果,而且有可能在适应度值很低的情况下过早收敛而达到局部极小.这是由于在山谷中欲达到极小点只能沿斜线运动,而单点交换每次只能改变 x_1 值或 x_2 值的缘故.多点交换与均匀交换则完全避免了这一限制,它们都是同时改变 x_1 值和 x_2 值.因此这两种交换操作可以明显地提高搜索速度并避免陷入局部极小.较多的模拟结果表明,对于 Rosenbrock 函数极小化问题,均匀交换是最理想的.图 2 给出了对于 10 个不同的初始种群,根据分别采用单点交换、多点交换与均匀交换得到的适应度的平均值绘出的适应度随迭代次数变化的比较曲线.表 1 给出了对于 20 个不同的初始种群,采用多点交换与均匀交换操作使适应度值分别达到 0.9 和0.9999 时所需的迭代次数.



……:单点交换 — — — —:多点交换 ———:均匀交换

图 2 单点、多点与均匀交换下适应度随迭代次数变化的比较曲线

表 1 多点与均匀交换迭代次数比较

	适应度达到 0.9		适应度达到 0.9999	
序号	多点交换	均匀交换	多点交换	均匀交换
1	345	823	743	1 979
2	187	180	2 314	1 459
3	3 139	261	6 394	1 409
4	258	98	703	527
5	109	409	2 563	4 436
6	1 529	81	1 991	2 629
7	9 776	690	11 895	2 501
8	150	368	1 981	1 349
9	152	97	7 636	2 033
10	106	282	3 966	1 345
11	65	136	531	1 423
12	238	684	1 165	4 986
13	110	159	1 013	719
14	93	393	2 392	2 811
15	810	520	13 901	2 760
16	3 752	395	5 762	5 608
17	184	74	3 014	1 066
18	923	369	2 815	926
19	810	173	15 345	727
20	314	147	1 181	1 482

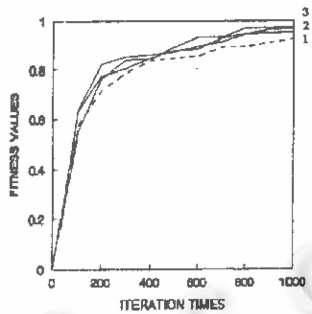
(3) 变异操作

变异是遗传算法的重要操作之一,它既可以产生种群中原本没有的较优基因,也可以恢复在先前的迭代中遗失的较优基因.所以它对于加快搜索速度、确保种群不至于陷入局部极小方面具有重要的意义.人们已经提出许多改进技术,如有指导的变异[9]和各种自适应变异[10~13]等,它们都针对不同的问题较常规的变异操作具有明显的优越性.

本文分别将常规的变异操作和有指导的变异操作两种方式用于 Rosenbrock 函数极小化问题,并比较和分析了优化结果.表 2 给出了对于 10 个不同的初始种群、α 取不同值并分别迭代到 950 次时所达到的适应度平均值,图 3 给出了 $\alpha=0, 0.001, 0.002, 0.009$ 时适应度随迭代次数变化的比较曲线,其中 $\alpha=0$ 代表常规的变异操作.由这些结果可以看出,对于 Rosenbrock 函数极小化问题,有指导的变异操作的引入,可以稍许改进优化结果,但并不能如文献[9]那样明显地提高搜索速度.分析(9)和(10)式可知,有指导的变异相当于在变异操作中加入了一些最速下降技术,而如第 1 节所述,最速下降法本身是不宜于解决 Rosenbrock 函数极小化问题的.

表 2　常规变异与有指导变异在相同迭代次数下达到的适应度值比较

α	0	0.001	0.002	0.006	0.009	0.012	0.04	0.08	0.1	0.3	0.6	0.9
适应度	0.9171	0.9486	0.9637	0.9629	0.9713	0.9599	0.9670	0.9649	0.9691	0.9655	0.9665	0.9665



------:常规变异　1:$\alpha=0.001$　2:$\alpha=0.002$　3:$\alpha=0.009$

图 3　常规变异与有指导变异下适应度随迭代次数变化的比较曲线

4　结　论

(1) 利用遗传算法可以有效地解决极小化 Rosenbrock 函数的问题.

(2) 得到了适于解决此问题的合理的遗传操作.线性与非线性选择、单点交换以及有指导的变异不宜用于此问题,随机选择、均匀交换与常规的变异是较为适宜的遗传操作组合.

(3) 为解决最速下降法所不能实现的某一类函数优化问题提供一种新的有效的途径.

参考文献

1　Rosenbrock H H. An automatic method for finding the greatest or least value of a function. Computer Journal, 1960, **3(3)**: 175~184.

2　Hilmmelblau D M. Applied nonlinear programming. McGraw-Hill Book Company, 1972.

3　Wilde D J, Beightler C S. Foundations of optimization. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

4 Holland J H. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press, 1975.

5 Booker L B, Goldberg D E, Holland J H. Classifier systems and genetic algorithms. Artificial Intelligence, 1989, **40**(1～3):235～282.

6 Pal Sankar K, Bhandari Dinabandhu. Selection of optimal set of weights in a layered network using genetic algorithms. Information Sciences, 1994, **80**(3～4):213～234.

7 Goldberg D E. Genetic algorithms in search, optimization, and machine learning, New York: Addison-Wesley, 1989.

8 Davies R, Clarke T. Parallel implementation of a genetic algorithm. Control Eng. Practice, 1995, **3**(1):11～19.

9 Bhandari Dinabandhu, Pal Nikhil R, Pal Sankar K. Directed mutation in genetic algorithms. Information Sciences, 1994, **79**(3):251～270.

10 Whitley D, Starkeather T, Bogart C. Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Computing, 1990, **14**:347～361.

11 Whitley D, Hanson T. Optimizing neural networks using faster, more accurate genetic search. Proceedings of the International Conference on Genetic Algorithms, 1989. 391～396.

12 Kitano H. Empirical studies on the speed of convergence of neural network training using genetic algorithms. Proceedings of AAAI-90, 1990. 789～795.

13 Yoon Byungjoo, Holmes Dawn J, Langholz Gideon *et al.* Efficient genetic algorithms for training layered feed forward neural networks. Information Sciences, 1994, **76**(1～2):67～85.

OPTIMIZATION OF ROSENBROCK'S FUNCTION BASED ON GENETIC ALGORITHMS

*LIANG Yanchun #ZHOU Chunguang ※LI Shoufan

*(*Department of Mathematics Jilin University Changchun* 130023)

*(*Open Laboratory of CAD/CAM Technology for Advanced Manufacturing The Chinese Academy of Sciences Shenyang* 110003)

#(*Department of Computer Science Jilin University Changchun* 130023)

※(*Unit* 86003 *Changchun* 130022)

Abstract This paper deals with the optimization of Rosenbrock's function based on genetic algorithms. The simulated results show that the problem can be solved effectively using genetic algorithms. The influence of some modified genetic algorithms on search speed is also examined. Some genetic operations suitable to the optimization technique are obtained, therefore, a novel way of solving a class of optimizations of functions that can not be realized using the method of steepest descent is proposed.

Key words Genetic algorithms, function optimization, steepest descent method, Rosenbrock's function, search speed.

Class number TP18