

## 用于曲线拟合的种群再分布遗传算法

张遵麟, 杨光

(华东师范大学物理系, 上海 200062)

(larryzzl@sohu.com)

**摘要:**针对简单遗传算法在曲线拟合应用中局部搜索能力差、收敛精度低的特点,提出了一种新的基于种群再分布的改进遗传算法。该算法在遗传算法进行的过程中,根据最优解的优劣,调整种群在最优解附近的分布,从而增强了算法的局部搜索能力。实验证明,该方法对于曲线拟合问题能取得优于简单遗传算法和传统数值迭代方法的结果。

**关键词:**遗传算法;曲线拟合;种群分布;种群再分布遗传算法

**中图分类号:** TP301.6 **文献标识码:** A

## Population-redistribution GA for curve fitting

ZHANG Zun-lin, YANG Guang

(Department of Physics, East China Normal University, Shanghai 200062, China)

**Abstract:** In order to improve the poor local search capability and low convergence precision of GA when applied in curvefitting, a new improved GA, named Population Redistributing Genetic Algorithm (PRGA), was proposed. With the progress of GA, this new algorithm adjusted the distribution of the population according to the quality of the best solution, thus effectively improved GA's local search capability. According to the results of the experiments on simulated data, PRGA gives better results in curve fitting compared with simple GA and traditional numerical iterative method.

**Key words:** genetic algorithm; curve fit; population distribution; PRGA(Population Redistributing GA)

### 0 引言

曲线拟合一般属于非线性最优化问题,传统上可以使用数值迭代算法来求解。这些算法在解空间中从用户指定的起点出发,根据一定的规则计算出下一个点,然后反复迭代,直到达到最终的终点。这类算法有:多元函数的下山单纯行法、多元函数的变尺度法、多元函数的共轭梯度法、多元函数的POWELL法、Quasi-Newton法等<sup>[1-3]</sup>。有些算法要求用户提供拟合函数的导数形式。这类算法的主要特点是收敛速度快,但容易陷入局部小值,拟合的结果较大程度的依赖于初始值。特别对于解空间形态比较复杂的多参数曲线的拟合,效果往往比较差。

遗传算法<sup>[4]</sup>是一种新型的全局优化搜索算法,其主要特点是从多个点出发进行群体搜索,同时充分利用群体中个体之间的信息交换。算法计算过程简单,对搜索空间有广泛的适应性,其对函数本身没有任何依赖性,尤其适用于处理传统搜索方法难以解决的复杂和非线性问题。遗传算法自提出以来,得到了广泛的应用。但是,未加改进的简单遗传算法(以下简称SGA)应用于曲线拟合等需要高精度结果的场合,存在着局部搜索能力差,收敛精度较低的问题。

针对SGA局部搜索能力差、收敛精度较低的问题,人们提出了不少改进方法,比较常见的方法包括:使用组合式遗传算法来加强其局部搜索能力、动态调整遗传算法编码来提高遗传算法的编码精度、自适应加速遗传算法以及一些其他的综合方法等<sup>[5-8]</sup>,这些算法各有不同的侧重点,都可以在某些方面改善遗传算法的效果或效率。

本文利用曲线拟合问题的特点,提出了一种新的SGA的改进方法——种群再分布遗传算法(Population Redistributing GA, PRGA)。该方法在遗传算法进行的过程中,根据最优个体的好坏,人为地改变种群的分布,从而达到改善遗传算法的搜索精度的目的。

### 1 理论

给定 $n$ 个实验数据点 $(X_i, Y_i), i = 1, \dots, n$ ,以及一个带有 $k$ 个待定参数 $p_1, p_2, \dots, p_k$ 的曲线函数 $y = f_{p_1, p_2, \dots, p_k}(x)$ ,简单的曲线拟合的过程就是寻求如下的误差函数最小化的过程:

$$E(p_1, p_2, \dots, p_k) = \text{Sqrt} \left( \sum_{i=1}^n (f_{p_1, p_2, \dots, p_k}(X_i) - Y_i)^2 \right) \quad (1)$$

在进行曲线拟合时,通常用户会指定一套参数的初始值。它是代表用户对于函数中待定参数的一个合理估计,在一个具备图形界面的交互式曲线拟合环境中,用户常常能够实时地观察函数曲线和数据曲线的吻合程度来估计参数和实际参数的吻合程度。遗传算法的初始种群的产生,传统上是在解空间中进行随机分布,然而,这种产生初始种群的方法,完全丢弃了用户对拟合参数的合理估计,所以不利于遗传算法的有效进行。更合理的做法应该是使得初始种群围绕用户估计的拟合参数的初始值进行分布。同时,如果用户设定的初值越准确,种群的分布就应该越窄,这样更有利于算法在最优解附近发掘;反之如果用户设定的初值误差越大,则种群的分布就应该越宽,这样有利于算法在更大的解空间中探索。

考虑到高斯分布的普遍性,我们使得每个参数的初始种群都是以该参数初始值为中心的高斯分布。公式(1)中的误

差函数给出了评价用户给定初始值优劣的直接判据。从理论上说,可以寻找适当的函数,将每个参数的分布宽度和误差函数关联起来,然而考虑到参数误差对拟合误差函数影响的复杂性,很难找到普遍使用的公式。因此,在实践中采用了统计的方法。具体的做法是,对于给定的初始值  $p_{10}, p_{20}, \dots, p_{k0}$ , 利用式(2) 计算出对应于每个实验点  $X_i$  的函数值  $Y_i'$ , 然后在一定的范围内,随机地产生每个参数的偏差  $\Delta p_i$ , 然后根据下式计算在这组参数偏差下的误差函数:

$$E(p_1, p_2, \dots, p_k) = \text{Sqrt} \left( \sum_{i=1}^n f_{p_{10}+\Delta p_1, p_{20}+\Delta p_2, \dots, p_{k0}+\Delta p_k}(X_i) - Y_i' \right)^2 \quad (2)$$

重复上面这个过程多次(如 > 1000 次)后进行统计,对每个参数都可以得到一张拟合误差函数随参数误差变化的对照表。根据公式(1) 计算出当前初始值对应的误差函数值  $E_0$ 。后,对于每个参数,都可以在上述对照表中找出该参数的偏差的一个界限,只要样本中该参数的偏差小于这个界限,样本所对应的拟合误差函数都小于  $E_0$  (置信度 95%)。得到该参数界限后,就可以计算出在该参数界限内积分面积达到总积分的 95% 的高斯分布的宽度。这个宽度,就是我们产生种群所需要的分布宽度。由于参数的实际分布并不完全符合高斯分布,为了确保最优解对应的参数值处于产生的种群的分布范围内,通常将得到的高斯分布宽度再乘以一个大于 1 的安全系数。

在以上述方法产生初始种群之后,就可以按一般 GA 的步骤进行了。不过,在 GA 的收敛速度过低时,PRGA 就以当前的最优值为中心,重新根据上述方法确定的分布重新产生种群,因此我们将这种算法命名为种群再分布遗传算法。通过种群的再分布,随着最优解在优化过程中不断接近目标,种群也更加密集地围绕着最优解分布,这样可以使得算法可以更好地在最优解附近发掘。同时,重新分布也避免了种群中出现过多相同或者过于相似的个体,可以避免算法的过早收敛。

综上所述,整个算法的流程如下:

- 1) 设置遗传算法参数;
- 2) 通过统计计算获得各参数的分布宽度;
- 3) 根据正态分布产生初始种群;
- 4) 开始简单遗传算法;
- 5) 如果符合收敛条件,则转向 7);
- 6) 如果简单遗传算法不再继续收敛,则转向 2);
- 7) 算法结束。

这里需要指出:1) 在一轮遗传算法进行后,如果发现的最优解距离上一次进行统计的参数值的距离不大,则没有必要重新进行统计;2) 进行重新分布的判据,一般选择在遗传算法连续几代中的最佳样本的适应值得变化程度小于某个预设值;3) 在再分布时,一般保留上一代中的最优解。实验证明,在 SGA 不再继续收敛时,种群的再分布可以有效地促进算法的进一步收敛。

## 2 实验

为了更好地评价我们的算法,利用模拟的数据对 SGA、PRGA 和 Quasi-Newton 数值迭代方法<sup>[9]</sup> 的曲线拟合的结果进行了比较。Matlab 中提供了多种数值迭代优化方法,我们发现,在这些方法中,Quasi-Newton 对于我们的拟合问题能够给出最佳的结果。通过模拟实验比较了拟合参数的个数、初始参数设置的偏差程度、实验数据中的噪声等各种因素对三种算法的影响。

实验所用的函数形式是常见的多个 Lorentzian 函数叠加的形式:

$$f(x) = \sum_{i=1}^n \frac{A_i}{1 + (x - x_i)^2 / T_i^2} \quad (3)$$

其中,每个组分有三个参数,  $A_i$  决定最高点的强度,  $x_i$  决定曲线的中心位置,  $T_i$  决定了曲线的宽度。用于拟合运算的实验数据,即利用公式(3) 产生,其中  $x_i$  的取值在 0 ~ 100 之间,  $T_i$  的取值在 1 ~ 10 之间,  $A_i$  的取值在 50 ~ 150 之间,模拟数据点数为 100。

计算所用的 SGA,使用了 GALIB 软件包<sup>[10]</sup>,遗传算法采用了实数编码,控制参数为:种群规模 80,交叉率 0.8,变异率 0.01,PRGA 中遗传算法的控制参数与 SGA 相同;Quasi-Newton 算法的实验,使用 Matlab 进行。对于每种算法,都重复进行 100 次,然后对结果进行统计。

### 2.1 拟合参数个数的影响

待拟合的曲线的参数个数越多,解空间就越复杂,这时就可以考验不同算法的搜索能力。我们使用三种算法分别进行了 1 ~ 8 个组分的 Lorentzian 函数叠加的曲线的拟合。优化前设定的参数初始值对于  $A_i$  和  $T_i$  而言,在真值的  $\pm 50\%$  之间随机产生,对于  $x_i$  而言,在真值  $\pm 20$  范围内随机产生。图 1 中列出了每种算法的对于不同组分数的曲线拟合时的平均误差函数值。从图 1 中可以看出,PRGA 的结果明显优于另两种算法。需要指出的是,由于随着组分数的增加,实验数据的平均值也上升,不同组分之间的数据难以直接比较。

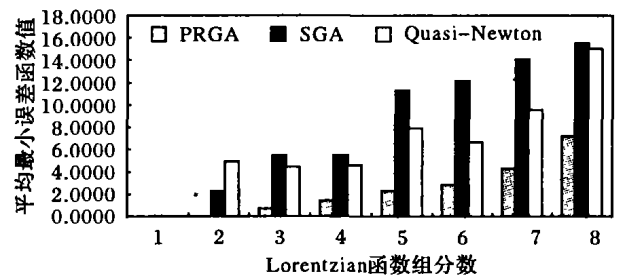


图 1 三种算法的结果比较

表 1 显示了三种算法比较的具体实验数据,包括平均误差函数值和误差函数值的均方差。误差函数的均方差可以反映算法的稳定性,从表 1 中可以看出,PRGA 和 SGA 的均方差大致相仿,都远远优于 Quasi-Newton 方法,这是因为遗传算法不容易陷入局部极小点,所以从不同出发点开始算法的解,相对比较稳定。

表 1 三种算法的误差函数的均值和均方差

	PRGA		SGA		Quasi-Newton	
	AveE	StdE	AveE	StdE	AveE	StdE
一组分	0.0000	0.0000	0.1169	0.0449	0.0000	0.0000
二组分	0.0007	0.0000	2.2072	2.4068	4.9438	7.7748
三组分	0.7353	0.8970	5.4708	2.1450	4.4194	5.5443
四组分	1.4508	1.3910	5.4720	2.4775	4.5726	5.9059
五组分	2.2928	1.8224	11.3742	1.3734	7.9400	6.0501
六组分	2.8166	1.5454	12.2186	1.6186	6.6817	6.2178
七组分	4.2640	2.5124	14.1347	0.8124	9.5942	6.2545
八组分	7.2155	2.9519	15.5628	1.4760	15.0608	6.0899

表中 AveE 表示平均最小误差函数值, StdE 表示均方差。

### 2.2 噪声的影响

噪声的增加也增加了解空间的复杂程度。模拟实验所用的数据,通过在利用公式(3) 产生模拟的数据上添加最大幅值为 Lorentzian 函数的强度的  $\pm 10\%$  的 Gauss 噪声获得。同

样对1-8个组分的Lorentzian函数进行了拟合,结果如图2和表2所示。

从图2、表2中可以看出,对于存在噪声的情况,显然数值迭代算法受的影响比较大,两种遗传算法的结果都明显优于Quasi-Newton方法,而PRGA也明显优于SGA。同时,PRGA和SGA的稳定性也远远优于Quasi-Newton算法。

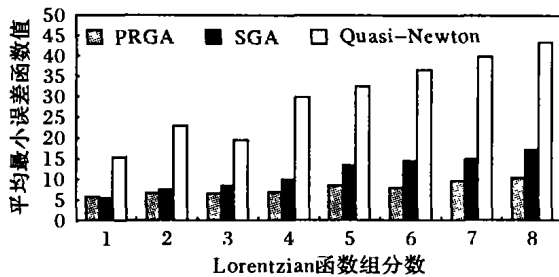


图2 有噪声情况下三种算法拟合结果的比较

表2 有噪声情况下三种算法的拟合结果的比较

	PRGA		SGA		Quasi-Newton	
	AveE	StdE	AveE	StdE	AveE	StdE
一组分	5.9444	0.0000	5.4459	0.0017	15.2985	12.3248
二组分	6.7243	0.0000	7.6021	1.0691	22.8723	13.1997
三组分	6.5988	0.4743	8.4957	0.8696	19.5413	7.7231
四组分	6.9395	0.3317	10.0032	1.1704	29.8964	10.4613
五组分	8.4665	0.5062	13.3979	1.0830	32.6587	11.2380
六组分	7.8972	0.7373	14.5499	1.0547	36.5337	10.8226
七组分	9.6803	0.9723	14.9620	0.7125	39.8899	9.8497
八组分	10.3430	1.4871	17.3138	0.7702	43.3682	8.6040

### 2.3 参数初始值设置的影响

拟合参数初始值的偏差大小会在一定程度上影响最终拟合的效果。PRGA算法人为地缩小了GA算法搜索的范围,那么是否会导致算法陷入局部极小呢?为了说明这个问题,使用了三个组分的Lorentzian函数的叠加函数作为拟合对象。同时,在随机产生参数初始值时,使得参数的最大允许误差由小变大(对于拟合函数公式(3)中参数 $A_i$ 和 $T_i$ 从 $\pm 10\%$ 变化到 $\pm 200\%$ , $x_i$ 从 $\pm 10$ 变动到 $\pm 50$ )。对PRGA、SGA以及Quasi-Newton方法在同样的初始条件下进行实验,结果如图3所示。从图3可以看出,PRGA在抗初值偏差上具有比其他两种算法更强的能力,算法的稳定性也较好。表3显示了实验的具体数据。

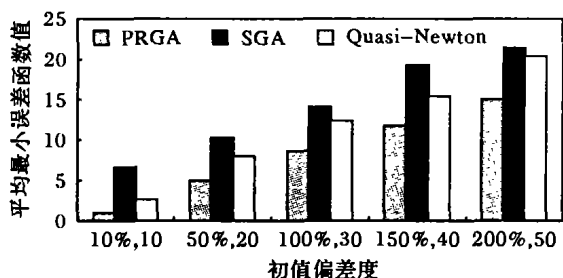


图3 不同初始值误差情况下三种算法的结果的比较

表3 不同初始值误差情况下三种算法的结果的比较

参数偏差	PRGA		SGA		Quasi-Newton	
	AveE	StdE	AveE	StdE	AveE	StdE
10%, 10	0.9130	0.8769	6.7198	2.6152	2.6281	4.5529
50%, 20	4.9939	2.7064	10.2533	3.2396	7.9713	5.7734
100%, 30	8.6790	3.4372	14.1425	3.8761	12.3987	7.5344
150%, 40	11.7224	3.1670	19.2451	5.7548	15.3946	8.2813
200%, 50	15.0005	4.1923	21.3842	7.9345	20.4266	12.9472

图3中横坐标的两个值,例如100%和30表示 $A_i$ 和 $T_i$ 的最大允许偏差为 $\pm 100\%$ ,而 $x_i$ 的最大偏差为 $\pm 30$ 。

### 2.4 同简单遗传算法收敛情况的比较

为了评价PRGA和SGA的收敛情况,使用了3组分Lorentzian函数的叠加作为测试对象,对每种算法进行了100次实验,同时记录每代中的最优样本,总共追踪300代,将100次实验中每代的最优样本的误差函数求平均后对算法的代数作图,结果如图4所示。从图4中可以看出,SGA在进行到后期时,收敛速度明显下降,存在着发掘能力不足的情况。而种群再分布的引入,显然改善了这种情况,使得算法能在更长的时间里持续收敛。因此,PRGA确实改善了SGA局部搜索能力弱的问题。

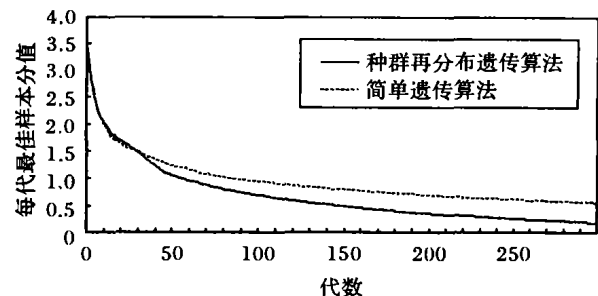


图4 PRGA和SGA的收敛情况的比较

## 3 结语

大量的运算实验表明,种群再分布遗传算法提高了简单遗传算法的局部搜索能力,应用于多参数曲线拟合的场合,能够取得优于简单遗传算法和传统数值迭代方法的结果。由于该算法需要进行额外的数据统计,所以会在速度上稍有影响。在我们采用的种群规模下,每次统计会带来10代左右演化的开销。提出一个更好的种群分布的判据,并将该算法扩展到更一般的应用场合,是进一步的工作方向。

### 参考文献:

- [1] 解可新. 最优化方法[M]. 天津: 天津教育出版社, 1997.
- [2] 李新社, 李忠科. 三次参数曲线拟合算法的优化研究[J]. 计算机工程与应用, 2001, 37(5): 73-77.
- [3] 黄海林. 单纯形算法对指数曲线拟合的应用[J]. Journal of Mathematical Medicine, 1997, 10(3): 206-207.
- [4] HOLLAND JH. Adaption in Natural and Artificial Systems [M]. The university of Michigan Press, Ann Arbor, 1975.
- [5] PARK BJ, CHOI HR, KIM HS. A hybrid genetic algorithm for the job shop scheduling problems[J]. Computers & Industrial Engineering, 2003, 45: 597-613.
- [6] XU ZB, LENUG KS, LIANG Y, et al. Efficiency speed-up strategies for evolutionary computation: fundamentals and fast-GAs[J]. Applied Mathematics and Computation, 2003, 142: 341-388.
- [7] AHUJA RK, ORLIN JB, TIWARI A. A greedy genetic algorithm for the quadratic assignment problem[J]. Computers & Operations Research, 2000, 27: 917-934.
- [8] MISEVICIUS A. An improved hybrid genetic algorithm: new results for the quadratic assignment problem[J]. Science Direct, 2004, 17: 65-73.
- [9] FORD JA. Implicit updates in multistep quasi-Newton methods[J]. Computers and Mathematics with Applications, 2001, 42: 1083-1091.
- [10] Matthew Wall. GALIB V2. 4[CP]. <http://lancet.mit.edu/ga/>, 2005-01.