

改进的快速遗传算法及其性能研究

高 玮^{1,2}

(1. 武汉工业学院土木工程系, 湖北 武汉 430023; 2. 后勤工程学院土木工程系, 重庆 400041)

摘 要: 遗传算法作为一种有效的全局随机优化方法, 在工程界已得到了广泛应用。为了克服工程应用中遗传算法所表现出的计算费时、稳定性差等缺点, 提出了一种快速收敛的遗传算法, 从初始群体生成及遗传算子操作等几个方面作了改进。通过几个复杂函数的优化仿真实验, 证明该算法能大大提高遗传算法的收敛速度, 明显改善算法的计算稳定性。

关键词: 遗传算法; 稳定性; 优化

中图分类号: TP18 **文献标识码:** A

An Improved Fast-Convergent Genetic Algorithm and Its Performance Study

GAO Wei^{1,2}

(1. Department of Civil Engineering, Wuhan Polytechnic University, Wuhan 430023, China;

2. Department of Civil Engineering, Logistical Engineering Institute, Chongqing 400041, China)

Abstract: As an effective global search method, genetic algorithm has been used in many engineering problems. But this method has the disadvantages of slow convergence and poor stability in practical engineering. In order to overcome these problems, an improved fast-convergent genetic algorithm is proposed in terms of creation of the initial population, genetic operators, etc. Through the simulation experiments of some hard-optimizing functions, the proposed algorithm shows its faster convergence and better stability than other existing algorithms.

Keywords: Genetic algorithm; Stability; Optimization

1 引 言

遗传算法是近年来计算机科学、信息科学及人工智能领域研究的一个“热点”^[1~3]。它是由美国学者 Holland 提出的一种基于达尔文生物进化论及门德基因遗传理论的仿生学概率性迭代搜索算法。

遗传算法借鉴了物种进化的思想, 将欲求解问题编码, 每个可能解均表示成字符串的形式。初始化产生一个群体, 用合理的适值函数对种群进行评价, 在此基础上进行选择、交叉及变异等遗传操作。适值函数类似于自然选择的力量。3 个遗传算子则分别对应于自然界广泛存在的生物繁衍、交配及基因突变。选择算子根据父代中个体的适值大小进行选择或淘汰, 此算子保证了算法的最优搜索方向。交叉算子模拟基因重组及随机信息交换, 它保证了算法的搜索范围。变异算子以概率进行基因突变, 这保证了算法的全局搜索能力。

据此原理, Goldberg 总结出一种统一的算法形式——基本遗传算法(simple genetic algorithm, SGA), 它是研究遗传算法的基础。其主要组成部分包括二进制串群体、控制参数、

适值函数、遗传算子(交叉及变异)、选择操作及终止条件。

基本遗传算法的实现步骤如下。

- (1) 随机生成由定长二进制串组成的初始群体;
- (2) 循环下列各步, 直到新群体中出现满足终止条件的解为止;
 - ① 计算每个个体的适应值;
 - ② 基于适应值, 从群体中选出若干个体;
 - ③ 对选出的个体进行交叉及变异操作, 将此后的个体放入新群体;
 - (3) 选择最佳个体作为最终结果。

遗传算法与其它优化搜索方法的不同在于: (1) 它是一种并行算法, 可同时对一个群体进行评估和操作, 种群进化时搜索了若干点, 从而减小了算法陷入局部极值的可能性。同时, 变异保证了算法理论上能搜索到空间每一点; (2) 采用编码方式, 可使个体以位为单位进行信息重组; (3) 只采用问题本身所有的目标函数, 无需其它先决条件或辅助信息; (4) 遗传算法是一种概率性搜索而非确定性转移, 这更接近自然现象的本质非确定性。

遗传算法虽在复杂的工程优化问题中得到了广泛应

用^[4,5],但由于算法随机搜索而带来的算法收敛速度慢及计算稳定性差已成为其应用的最大阻碍。为了改善算法的性能,已对其进行了大量研究,提出了不少改进方法^[6-19]。为了弥补遗传算法的不足,本文提出了一种改进的快速遗传算法。仿真实验证明,此算法大大提高了计算速度,明显改善了计算稳定性,其性能好于很多现有算法。

2 改进遗传算法的基本原理

2.1 初始群体生成

研究表明^[5,7],初始群体的分布性质严重影响整个算法的收敛性能。初始群体性质差会使算法收敛速度慢,甚至不收敛。初始群体是在搜索空间中随机产生的,在大部分遗传算法设计中均采用这种随机产生初始群体的方法。这种方法没有考虑初始群体中个体在解空间的分布情况,因而可能导致很多个体都集中在某一局部区域内,不利于扩大搜索空间和搜索到全局最优点。为了改善初始群体的性质,有必要对此进行研究。文献[5]介绍了几种初始群体的生成方法:(1)小群体竞争法。首先随机生成一定数量的个体,从中找出适值最大者,并以之作为一个初始个体,重复此过程,直到生成初始群体为止。该方法尽管可提高初始群体的质量,但并没有改善初始群体的分布性质,而且会使计算量显著增加,不利于提高计算速度;(2)人工在全局最优点附近布点,以之作为初始群体。这种方法尽管能产生性能良好的初始群体,但其致命的缺点是:事先必须对求解问题有相当好的了解,而这种要求一般很难达到;(3)以广义海明距离为判据的均匀初始群体产生方法^[7]。该法对每个初始个体都应计算其与其它个体间的广义海明距离,并应保证它们的距离大于一个给定常数。这种方法的确能使初始群体分布均匀,但会增加计算量,且具体操作不便(距离的阈值不易确定)。本文提出一种既快速又不影响算法其它性能的初始群体产生方法,称为小区间生成法。即先把各待优化参数的取值范围分成群体总数个小区间,再在各小区间中分别随机生成一个初始个体。这样生成的初始个体将会均匀地分布在整个解空间上,并能保证随机产生的各个体间有明显的差距,保证了初始群体含有较丰富的模式,增强了搜索收敛于全局最优点的可能。仿真实验证明,这种方法生成的初始群体性质优良,可大大加快算法的收敛速度,很好地改善算法的收敛性质。

2.2 编码方式

遗传算法的收敛性能同其编码方式有密切关系^[11],因此确定编码方式是算法设计的首要任务。Holland的模式定理建议采用二进制编码方式,并由此产生了最小字符集编码原则。尽管二进制编码简单易行,且类似于生物染色体的组成,但在计算时要进行编码及解码操作,使计算效率下降;而且当进行连续参数计算时,二进制编码会产生编码及解码误差,从而使计算精度降低,此时为了提高计算精度应采用较长的码串,但码串太长又会使计算效率下降。而实数编码不

需要对待优化参数进行编码及译码操作,它采用直接把待优化参数连成一个实数向量的方式。这种编码方式既可以克服二进制编码引起的Hamming悬崖又具有微调功能,同时可在解的表现型上直接进行遗传操作,从而便于引入与问题相关的启发式信息,增强了算法的导向性,提高了算法的收敛性及收敛速度。实数编码还便于进行大空间搜索。Michalewica等^[11]通过实验比较了二进制编码同实数编码的性能,结果证明,实数编码无论在收敛速度还是收敛精度上均好于二进制编码。Antonise^[12]则从理论上证明了Holland在推导最小字符集编码规则时存在的错误,指出了大字符集编码设计反而可提供更多的模式,有利于算法的收敛。基于上述分析,本算法采用实数编码方式。

2.3 复制操作

为了保证搜索达到的最佳个体不会被各种遗传操作破坏,并保留上代群体的优良特性,本算法把上代群体全部复制保留到匹配池中。这是一种允许父代参加竞争的复制方式,此种方式对算法的影响已有人专门作过研究,并把此种改进遗传算法称为两代竞争遗传算法^[13]。他们的研究表明,此种操作有利于保持群体的多样性,可避免“早熟”现象,并能大大提高算法的效率及鲁棒性。文献[14]也从理论上研究了允许父代参加竞争对算法收敛性的影响,发现父代竞争对算法收敛性有决定性影响。

2.4 优选父代自适应交叉操作

借助优生学原理,该操作在进行交叉操作时,提高被交叉父代个体的质量,以期产生高质量的子代。具体操作为:从上代群体中随机选择两个个体,保留适应值大的个体(如二者适应值相同,则随机保留一个),再进行一次上述两两优选操作。对保留下来的两个个体进行交叉。交叉采用实数编码遗传算法中的整体算术交叉算子^[10]。具体操作如下。

设两交叉个体分别为

$$A = (a_1, a_2, \dots, a_i, \dots, a_l)$$

$$B = (b_1, b_2, \dots, b_i, \dots, b_l)$$

交叉后的个体为

$A' = (a'_1, a'_2, \dots, a'_i, \dots, a'_l)$ 和 $B' = (b'_1, b'_2, \dots, b'_i, \dots, b'_l)$
式中 $a'_i = \alpha_i a_i + (1 - \alpha_i) b_i$; $b'_i = \alpha_i b_i + (1 - \alpha_i) a_i$, α_i 是在 $[0, 1]$ 上取值的均匀随机数。

为保证交叉的质量,交叉概率采用自适应方式产生^[9],即交叉概率由进化程度控制。自适应交叉概率由下式确定

$$p_c = \begin{cases} (f_{\max} - f_c) / (f_{\max} - f_{\text{avr}}), & f_c \geq f_{\text{avr}} \\ 1.0, & f_c < f_{\text{avr}} \end{cases}$$

式中 f_c ——要交叉的两个串中较大的适应值; f_{\max} 、 f_{avr} ——上代群体中个体的最大适应值及群体的平均适应值。

2.5 非一致自适应变异操作

变异算子保证算法能搜索到解空间中的每一点,使算法具有全局收敛性。使用实数编码时,变异算子不再像二进制编码时仅是简单的恢复群体的多样性,它已成了一个重要的

搜索算子。变异对经过优选交叉的个体进行,具体采用实数编码的非一致变异方式^[11]。该算子的结果同进化代数密切相关,使得在进化初期,变异范围相对较大,而随着进化代数的推进,变异范围越来越小,起一种对进化系统微调的作用。

非一致变异操作如下:设待变异的个体向量为 $A = (a_1, a_2, \dots, a_i, \dots, a_l)$, 其中分量 a_i 被选出作变异操作,其取值范围为 $(a_{i\max}, a_{i\min})$, 则变异后它成为

$$a_i' = \begin{cases} a_i + f(t, a_{i\max} - a_i) & \text{rad} = 0 \\ a_i - f(t, a_i - a_{i\min}) & \text{rad} = 1 \end{cases}$$

式中 ran ——均匀随机数; $f(t, y)$ ——一个在 $[0, y]$ 范围取值的函数,其表达式为

$$f(t, y) = y(1 - r^{(1 - \frac{t}{T})^b})$$

式中 t ——群体进化的代数; r —— $[0, 1]$ 间取值的随机数; T ——最大进化代数; b ——系统参数,一般取值为 2。为保证变异的质量,采用自适应变异概率^[9],使变异大小由进化进度控制。其具体表达式为

$$P_m = \begin{cases} 0.5(f_{\max} - f_m)/(f_{\max} - f_{\text{avr}}), & f_c \geq f_{\text{avr}} \\ 0.5 & f_c < f_{\text{avr}} \end{cases}$$

式中 f_{\max}, f_{avr} ——上代群体中的最大适应值及其平均适应值; f_m ——需要变异个体的适应值。

2.6 选择操作

对于匹配池中的个体,采用如下两种方式进行选择以产生子代群体。

(1) 实施最优保留策略 将匹配池中适应值最大的个体无遗传变化的放入子代群体,以保留上代群体的最优解,使算法可以以概率 1 收敛于全局最优解,保证了算法的收敛性。

(2) 两两竞争选择策略 对匹配池中除最优解以外的其它个体进行两两竞争选择,从匹配池中随机选取两个个体,比较它们的适应值,把适应值较大的个体保留到子代,直到产生完整子代群体。

2.7 终止条件

本算法的终止条件为群体进化稳定,即进化群体中最大个体适应值和群体平均适应值的差小于给定常数 $1e-5$ 时算法终止。为了防止迭代永不停止,规定最大进化代数为 50。

3 改进遗传算法的实现

为了实现上述算法,以 Fortran 90 语言编写了计算机程序,其程序流程图如图 1 所示。

4 改进遗传算法仿真实验

4.1 实验 1

为研究改进遗传算法解决多模态优化问题的能力,采用 Precy 函数 F_1 及 Schafer 函数 F_2 进行了仿真实验。两个函数的性质如表 1 所示。

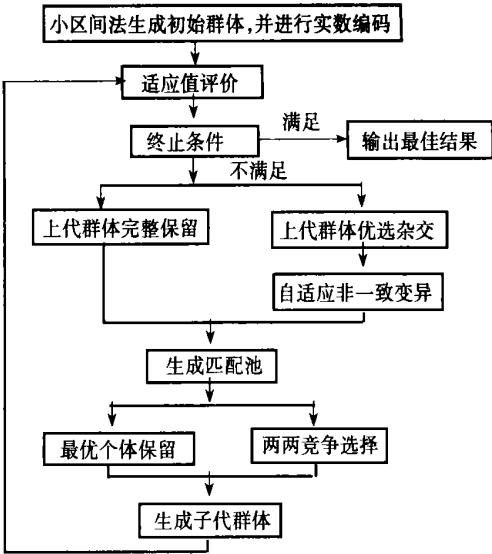


图 1 改进遗传算法流程图

表 1 仿真实验函数性质

函数表达式	参数取值范围	优化阈值
$F_1: f(x) = \frac{x_1^2 + x_2^2}{2} + \cos(20\pi x_1)\cos(20\pi x_2) + 2$	$[-10, 10]$	3 (最小值)
$F_2: f(x) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2 - 0.5}}{[1 - 0.001(x_1^2 + x_2^2)]^2}$	$[-100, 100]$	1 (最大值)

本算法同简单遗传算法及文献[9]中算法的仿真实验结果比较如表 2 所示。

表 2 仿真实验结果比较

函数	简单遗传算法		文献[9]算法		本文算法	
	平均进化代数	未收敛次数	平均进化代数	未收敛次数	平均进化代数	未收敛次数
F_1	81.9	15	51.7	6	11.2	2
F_2	70.6	4	30.8	0	6.4	0

仿真实验中,简单遗传算法的参数为: $N = 100, P_c = 0.65, P_m = 0.01$; 文献[9]算法的参数为: $N = 50, P_c = 0.65, P_m = 0.15$; 本文算法取 $N = 50$ 。实验中以独立优化 50 次作为一次实验,以达到函数优化阈值所需要的平均进化代数作为衡量算法速度的标准。如算法进化 200 代仍未收敛到优化阈值,则认为本次计算不收敛。将 3 种算法在 50 次实验中未收敛的次数作为衡量算法稳定性的标准。由表 2 可见,对函数 F_1 ,简单遗传算法的平均进化代数是本文算法的 7.3 倍,文献[9]算法是本文算法的 4.6 倍。对 200 代进化中的未收敛次数,简单遗传算法是本文算法的 7.5 倍,文

献 9 算法是本文算法的 3 倍。而对函数 F_2 ，简单遗传算法的平均进化代数是本文算法的 11 倍，文献[9] 算法是本文算法的 4 8 倍。可见，本文算法无论在收敛性还是稳定性上均极大地改善了遗传算法的性能，其计算效果理想。

4. 2 实验 2

为了比较算法的收敛速度，本次实验以一个简单的单峰函数(Dejong 提出的球函数)进行研究，并与其它一些算法进

行比较。待优化函数形式为

$$f(x_i) = 100 - \sum_{i=1}^3 x_i^2$$

式中 $x_i(i = 1 \sim 3)$ 的取值范围为 $x_i \in [-5. 12. 5. 12]$ 。问题为极大值问题，该函数的极大值在(0, 0, 0)处为 100。仿真实验结果比较如表 3 所示。

表 3 不同算法达到指定函数值的进化代数比较表

目标函数值	99	99. 5	99. 9	99. 95	99. 99	99. 995	99. 999	99. 999 5
简单遗传算法	3	8	30	45	185	—	—	—
均匀交换遗传算法	3	8	54	90	157	—	—	—
带移民的遗传算法	4	7	40	78	179	—	—	—
一种快速遗传算法	3	7	26	58	134	163	—	—
文献[7] 算法	2 63	4 47	8. 4	11. 63	38. 83	46. 1	51. 27	52. 87
本文算法	1	1	1	1	1. 73	1. 87	2. 4	3. 2

注：表中进化代数为独立实验 30 次的平均值，“—”表示实验不收敛，表中其它算法参见文献[7] 相应文献。

由表 3 可以发现，本文的改进遗传算法在收敛速度上有了明显提高，甚至在达到最优值 100 时，本文算法在 50 次实验中的平均进化代数也只有 6. 68，因此本文算法比很多现有遗传算法均有很大改善，具有很大的工程适用性。

5 结束语

由上述分析及仿真实验可以看出，由于本文的改进遗传算法从初始群体的均匀生成、实数编码方案的采用、遗传算子概率的自适应生成、最优个体的无变化保留等策略上改进了现有遗传算法，因此，该算法无论在收敛速度还是计算稳定性上都有了极大的改进，在算法性能上优于很多现有算法，工程实用性很强，值得在工程实践中推广应用。

参考文献：

[1] Srinivas M, Patnaik L M. Genetic Algorithm: A Survey[J] . IEEE Computer, 1994, 27(6): 17—26.
[2] Foge D B. An Introduction to Simulated Evolutionary Optimization[J] . IEEE Trans. on SMC, 1994, 24(1): 3—14.
[3] Atmar W. Notes on the Simulation of Evolution[J] . IEEE Trans. on SMC, 1994, 24(1): 130—147.
[4] 高玮, 郑颖人. 岩体参数的进化反演[J] . 水利学报, 2000(8): 1—5.
[5] Zalala A M, Fleming P J. Genetic Algorithms in Engineering Systems

[M] . London: The Institution of Electrical Engineers, 1997.
[6] Srinivas M, Patnaik L M. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms[J] . IEEE Trans. on SMC, 1994, 24(4): 656—667.
[7] 吴斌, 吴坚, 涂序彦. 快速遗传算法[J] . 电子科技大学学报, 1999, 28(1): 49—53.
[8] Potts J C, Giddens T D, Yadav S B. The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial Selection[J] . IEEE Trans. on SMC, 1994, 24(1): 73—86.
[9] 韩万林, 张幼蒂. 遗传算法的改进[J] . 中国矿业大学学报, 2000, 29(1): 102—105.
[10] Millar J A, Potter W D. An Evaluation of Local Improvements Operators for Genetic Algorithms[J] . IEEE Trans. on SMC, 1993, 23(5): 1340—1351.
[11] Michalewicz Z. Genetic Algorithms + Data Structure = Evolution Programs[M] . New York: Springer-Verlag, 1996.
[12] Antonise J. A New Interpretation of Schema Notation that Overturns the Binary Encoding Constraint[C] . In: Proc. of the 3rd Int. Conf. on GA. CA: Morgan Kaufmann, 1989, 86—91.
[13] 于海斌, 王浩波, 徐心和. 两代竞争遗传算法及其应用研究[J] . 信息与控制, 2000, 29(4): 309—314.
[14] 张讲社, 徐宗本, 梁怡. 整体退火遗传算法及其收敛充要条件[J] . 中国科学(E), 1997, 27(2): 154—164.