

基于成长算子的改进遗传算法及仿真

阎镜予, 孙德敏, 凌 青

(中国科学技术大学 自动化系, 安徽 合肥 230027)

摘要: 模拟生物界成长发育过程, 加入成长算子对遗传算法框架进行改进, 形成新的算法框架—成长遗传算法(growth GA). 该算法能够克服简单遗传算法寻优速度较慢、局部搜索能力较弱的缺点. 利用爬山法局部搜索能力强的特点, 给出成长算子的一种具体实现, 并证明加入成长算子不改变算法收敛性. 与简单遗传算法和确定性拥挤遗传算法的对比函数优化实验证明: 成长遗传算法有利于兼顾寻优速度和收敛精度.

关键词: 成长遗传算法; 成长算子; 收敛性; 函数优化

中图分类号: TP273 **文献标识码:** A

Improved genetic algorithm based on growth operator and simulation

YAN Jing-yu, SUN De-min, LING Qing

(Automation Department, University of Science and Technology of China, Hefei Anhui 230027, China)

Abstract: By emulating the process of growth in nature and using growth operator, a growth genetic algorithm (GGA) is proposed to overcome the drawbacks of simple GA (SGA) such as slow optimization speed and weak local search ability. A practical realization of growth operator is proposed by making use of the strong local search ability of the hill climbing method. It has been demonstrated that adding the growth operator doesn't change the convergence property of SGA. The simulation result compared with SGA and deterministic crowding GA (DCGA) for function optimization verifies that the growth genetic algorithm facilitates the balance between optimization speed and convergence precision.

Key words: growth genetic algorithm; growth operator; convergence property; function optimization

1 引言(Introduction)

遗传算法(genetic algorithm, GA)是一类模拟生物进化机制与过程来解决问题的人工智能与优化技术^[1], 近年来在众多领域中得到了广泛的应用^[2,3]. 与其他进化计算相比, 遗传算法更强调对染色体的操作^[4,5]. 遗传算法的创始人J.H.Holland认为^[6]:进化是一个运行在染色体上的过程, 而不是运行在它们编码的生命体上的过程, 繁殖过程是进化发生的地点. 正是基于这个出发点, 遗传算法与强调个体级行为的演化策略不同, 少有考虑个体在进化过程中的主动性.

在标准遗传算法流程中, 个体一直处于被动地位, 交叉算子与变异算子均随机地对个体进行操作. 个体的演变过程在基因空间中完成, 仅仅在问题空间中确定个体所包含染色体的扩散或消亡, 没有充分利用问题空间信息.

正是由于基因的演变处于随机状态, 且个体不能

利用问题空间中的信息引导自身发展, 导致了标准遗传算法寻优较慢、局部搜索能力较弱的缺陷.

针对上述缺陷, 本文在简单遗传算法(SGA)基础上加入一个新的算子—成长算子(growth operator), 从而提出成长遗传算法(growth GA, GGA). 进一步给出一种具体实现方法及其参数选取规则, 并证明了GGA的收敛性与SGA等同. 该算法在函数优化实验中表现出良好的性能.

2 成长算子模拟背景与结构(Background and structure of growth operator)

成长算子所模拟的生物界背景为: 当个体诞生以后, 并不是马上进行交配繁殖, 存在着一个发育学习的过程. 在这个过程中, 个体学习前人的知识, 并通过自然界对这些知识的反馈来更新和创造新的知识, 从而使自身的适应度提高, 待自身发育成熟以后再再进行新一轮交配繁殖.

成长算子模拟上述生物背景, 使遗传算法不再

局限于对染色体的操作中,在交叉之前让个体存在“发育”的机会,能够主动地确定性地改善自身的适应度.

个体保留其父代染色体信息作为对前人知识的积累,然后在问题空间中将自己的适应度与父代适应度进行比较来获取问题空间的反馈信息,利用该信息采取某种确定性的算法来修改自身的染色体从而得以发育和成长,表现出一代更比一代强的趋势.

种群初始化时,第一代个体的父代可设定为其自身,故成长时因父子染色体完全相同不会改变基因.交叉操作时,新生成的个体采取某种准则与生成它的两个父代个体之一进行配对,个体除保留自身染色体之外,同时保留其父代的染色体.经过交叉、变异算子作用以后,子代染色体和父代染色体已经不同,当再次被成长算子作用时,基于子代与父代的差异则可对子代基因进行某种确定性的调整.

3 成长算子的一种实现方法(Realization of growth operator)

交叉算子形成新的个体时,一般由两个父代个体产生两个子代个体,此时如何确定谁是谁的父代是成长算子必须解决的问题.

具体实现时有两种选择方法:

1° 从模式出发.若子代与某一父代个体的海明距离小于另一个体,则子代的模式主要由该父代确定,令其为父.如图1所示, a 与 A 的海明距离为2,与 B 的海明距离为6,则 a 的父代为 A ,同理 b 的父代为 B .

2° 从数值出发.若子代继承了某一父代的高位基因,则子代在数值上与该父代接近,令其为父.如图1所示, A, B, a, b 的十进制值分别为1238, 633, 726, 1145,故 a 的父代为 B , b 的父代为 A .

父代A	1	0	0	1	1	0	1	0	1	1	0
父代B	0	1	0	0	1	1	1	1	0	0	1
子代a	0	1	0	1	1	0	1	0	1	1	0
子代b	1	0	0	0	1	1	1	1	0	0	1

图1 交叉实例

Fig. 1 Cross instance

成长算子是在问题空间中以适应度的方式对个体修正,故以第2种方式为宜.

需要注意的是,采用成长算子后的遗传算法由于加快了收敛速度故容易出现早熟,需要配合一些防止早熟的算法或自适应变异概率.

本文采用基于模拟退火步长的爬山法实现成长算子.令群体大小为 N ,当前遗传代数为 i ,当前个

体为 Gen ,相应的父代为 $Father$,爬山步长为 $Step$,问题空间取值上下界 MIN, MAX ,成长概率 $P_g \in [0, 1]$,变异概率为 $P_m \in [0, 1]$,则成长算子伪代码如下:

1) 模拟退火步长

$$Step(i) = \frac{Step(0) - Step(\infty)}{\log(10 + i\alpha)} + Step(\infty).$$

其中: $Step(0), Step(\infty)$ 为初始步长和结束步长, α 为降温速度,值越大降温速度越快,等于0时相当于固定步长.

2) 自适应成长概率与变异概率,成长概率越大,变异概率也应当越大.两者也可采用模拟退火得以实现:

$$P_g(i) = \frac{P_g(0) - P_g(\infty)}{\log(10 + i\beta)} + P_g(\infty),$$

$$P_m(i) = \frac{P_m(0) - P_m(\infty)}{\log(10 + i\gamma)} + P_m(\infty).$$

其中: $P_g(0), P_g(\infty)$ 分别为成长概率初始值和最终值; $P_m(0), P_m(\infty)$ 分别为变异概率初始值和最终值; β, γ 分别为两者的降温速度,等于0时相当于固定成长概率和变异概率.

3) 随机选择 $N \times P_g(i)$ 个个体操作如下:

i) 从基因型到表达型映射:

$Gen = \text{decoding}(Gen)$,

$Father = \text{decoding}(Father)$.

ii) 基于爬山法对子代进行修改:

If $Fitness(Gen) < Fitness(Father)$

$Gen = Gen - Step(i)[Gen - Father]$;

Else $Gen = Gen + Step(i)[Gen - Father]$;

iii) 越界处理:

If $Gen > MAX$ Then $Gen = MAX$,

If $Gen < MIN$ Then $Gen = MIN$.

iv) 从表达型到基因型映射:

$Gen = \text{coding}(Gen)$.

采用基于爬山法的成长算子意义如下:

1° 成长算子实际上是在遗传算法循环中加入一次局部搜索,收敛速度将加快.

2° 在算法初期采用较大的步长能够促使处于各个峰底的个体具有向峰顶聚集的趋势,加快收敛速度.

3° 在算法后期采用较小的步长可使爬山速度降低,范围变细,避免在峰的两侧来回震荡,重在提高解的精度.

4° 成长算法充分利用了问题空间的信息,使得遗传算法的不确定性与确定性兼顾,个体主动性得以表现.

4 成长遗传算法收敛性分析(Analyse of convergence property of GGA)

参照文献[7], 对加入成长算子后遗传算法的收敛性做如下分析:

标准遗传算法可被描述为马尔科夫链: 状态空间 $S = IB^n = nIB^l$, 其中 n 表示群体大小, l 表示串个体长度. 映射 $\pi_k(i)$ 提取状态 i 的二进制表示中的第 k 个长度为 l 的段, 用于区分个体和群体, $f(x)$ 为适应度计算函数, $F(x)$ 为定义在 $\{0, 1\}^l$ 上的实值函数.

群体中各基因(个体中的一位)由于遗传算子引起的概率变化由变换矩阵 P 描述, 且 $P = CMS$, 其中 C 描述交叉算子引起的变化, M 描述变异算子的作用, S 描述选择算子引起的变化. 加入成长算子后, 令 $P = GCMS$, G 描述成长算子引起的变化.

定理 1 如果变异概率为 $P_m \in (0, 1)$, 交叉概率为 $P_c \in [0, 1]$, 成长概率为 $P_g \in [0, 1]$, 同时采用比例选择法, 则加入成长算子后的遗传算法的变换矩阵 P 仍然是基本的.

证 交叉算子可以被认为是状态空间 S 中的一个随机满射函数, 即 S 的每个状态都随机地映射到另一状态, 因此 C 是随机矩阵. 同理 M, S 也是随机矩阵.

成长算子中, 子代与其父代的差异由随机的交叉和变异算子造成, 故差异也是随机的, 且 S 中的每个状态都可能被该随机差异映射到另一状态, 因此 G 也是随机矩阵, 故 GC 的乘积是随机矩阵.

因为变异算子独立地施加到群体中的每个基因位, 因此在变异算子作用下, 从状态 i 变成状态 j 的概率总计为

$$M_{ij} = P_m^H (1 - P_m)^{n-H} > 0.$$

其中 H 表示状态 i 和 j 的海明距离, 因此 M 是定正的.

选择算子不改变变异算子产生的状态的概率下界为

$$S_{ii} \geq \frac{\prod_{k=1}^n f[\pi_k(i)]}{(\sum_{k=1}^n F[\pi_k(i)])^n} > 0,$$

因此 S 是列可容的.

根据文献[7]引理2.1可知, P 是基本的.

成长算子的加入没有改变变换矩阵 P 的性质, 故关于标准遗传算法的收敛性结论均成立.

5 仿真试验(Simulation experiment)

为验证GGA的性能, 与SGA和具有较快寻优速度的DCGA^[8]进行了函数优化对比实验, 主要从寻优速度和寻优精度两个方面进行讨论.

对求最大值的Schaffer F7函数^[9]仿真:

$$\max F(x) =$$

$$25 - (\sum x_i^2)^{0.25} \times (\sin^2(50 \times (\sum x_i^2)^{0.1}) + 1),$$

其中 $-100 \leq x_i \leq 100$.

两变量时, 函数值的空间分布如图2, 该函数峰值众多, 具有较高的复杂度.

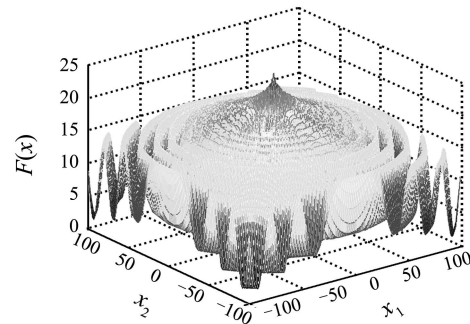


图2 仿真函数值的空间分布

Fig. 2 Landscape of the test function

为突出GGA的局部搜索性, 取较长的编码长度, 每个变量16位编码, 两变量时编码长度 $L = 32$, 群体大小 $N \approx 3L$ 取100, 强制结束代数60, 均采用精英保存策略, $\text{Step}(0)=0.1$, $\text{Step}(\infty)=0$, 降温速度 $\alpha = 20$, 固定成长概率和变异概率为 $P_g = 0.8$, $P_m = 0.15$.

分别采用上述3种方法对函数在同一初始状态下进行20次仿真, 取各自最优的一次, 曲线如图3所示, 结果如表1.

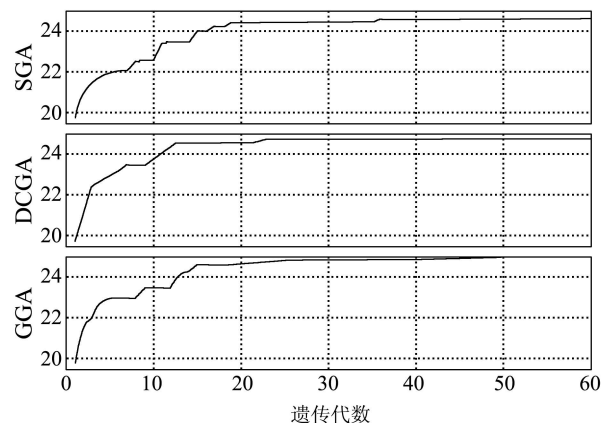


图3 最佳个体曲线对比图

Fig. 3 Curves of the elitists of each method

表1 优化结果对比表
Table 1 Optimization results

算法名称	收敛代数	36代时最佳值	39代时最佳值	46代时最佳值
SGA	36	24.543	24.602	24.602
DCGA	39	24.765	24.765	24.765
GGA	46	24.759	24.767	24.898

优化初期, GGA成长步长较大, 最佳个体值上升速度明显高于SGA, 与DCGA相差无几.

优化中期, GGA成长步长开始变小, 上升速度放慢, 重在局部搜索, 但仍高于SGA; DCGA保持快速上升性, 并且收敛.

优化后期, 当DCGA和SGA均已收敛时, GGA在小步长成长算子的作用下, 对最优点附近进行更加细微的搜索, 最终达到较高的收敛精度.

从整个过程来看, GGA陷入局部平稳代数较少, 一直保持上升趋势, 且同代相比, GGA的寻优精度优于SGA, 中期较短时间内虽然略低于DCGA, 但后期更高.

需要指出的是: 尽管GGA收敛代数较SGA和DCGA长, 但从效果来看, 当SGA和DCGA收敛时, GGA的未收敛最优值并不比上述二者差, 即: GGA未收敛时就已经能够达到SGA和DCGA的收敛精度.

6 结论(Conclusion)

成长遗传算法通过模拟自然界成长发育过程, 加入成长算子获取问题空间中的有用信息, 指导个体进行确定性的改进, 且不改变标准遗传算法的收敛性. 变化的成长步长能够兼顾寻优速度和收敛精度. 基于爬山法的一种具体实现, 在对比仿真中表现出了良好性能, 证明成长遗传算法行之有效.

本文中只是给出了一种简单的成长算子实现方法, 不同的实现方法将给算法带来不同的寻优侧重性和优化效果. 继续对成长算子的实现方法进行研究将是一项有意义的工作.

参考文献(References):

- [1] HOLLAND H J. *Adaptation in Natural and Artificial Systems*[M]. Cambridge, MA: MIT Press, 1975.
- [2] 李敏远, 都延丽. 基于遗传算法学习的复合神经网络自适应温度控制系统[J]. 控制理论与应用, 2004, 21(2): 242 – 246.
(LI Mingyuan, DU Yanli. Composite neural networks adaptive control system of temperature based on GA learning[J]. *Control Theory & Applications*, 2004, 21(2): 242 – 246.)
- [3] 刘锋, 陈国良, 吴昊. 基于遗传算法的方程求根算法的设计和实现[J]. 控制理论与应用, 2004, 21(3): 467 – 469.
(LIU Feng, CHEN Guoliang, WU Hao. Design and implementation of genetic algorithm for finding roots of complex functional equation[J]. *Control Theory & Applications*, 2004, 21(3): 467 – 469)
- [4] DE J, WILLIAM S. On the state of evolutionary computation[C]//*The Proc of the Inte Conf on Genetic Algorithms*. San Maieo, CA, USA: Morgan Kuafmann, 1993: 618 – 623.
- [5] BACK T, HAMMEL U, SCHWEFEL H P. Evolutionary computation: comments on the history and current state[J]. *IEEE Trans on Evolutionary Computation*. 1997, 1(1): 7 – 13.
- [6] 王正志, 薄涛. 进化计算[M]. 长沙: 国防科技大学出版社, 2000.
(WANG Zhengzhi, BO Tao. *Evolutionary Computation*[M]. Changsha: National University of Defense Technology Press, 2000.)
- [7] 陈国良, 王煦法, 庄镇泉, 等. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 1996.
(CHEN Guoliang, WANG Xifa, ZHUANG Zhenquan, et al. *Genetic Algorithm and Application*[M]. Beijing: People's Telecom Press, 1996.)
- [8] MAHFOUD S.W. Crossover interactions among niches[J]. *Evolutionary Computation*, 1994, 1(6): 188 – 193.
- [9] MICHALEWICZ Z. *Genetic Algorithms + Data Structures = Evolution Programs*[M]. Heidelberg, Berlin: Springer Verlag, 1996.

作者简介:

阎镜予 (1981—), 男, 研究方向为演化计算、计算智能, E-mail: jyayan@mail.ustc.edu.cn;

孙德敏 (1938—), 男, 教授, 博士生导师, 研究方向为先进控制与优化;

凌青 (1981—), 男, 博士研究生, 研究方向为全息光栅设计、演化计算.