统计遗传算法*

张 铃　　张 钹

（安徽大学人工智能所　合肥　230039）　　（清华大学计算机系　北京　100084）

（智能技术与系统国家重点实验室　北京　100084）

摘要　本文讨论了遗传算法中框架定理的不足之处，并对之进行了改进，然后分析了遗传算法与 A 算法的相似性，以及遗传算法的概率性质. 由此联想到它与 SA 算法的相似性，在此基础上，作者将原先发展的一套 SA 算法的理论移植到遗传算法中来，建立一个新的算法，称之为统计遗传算法（简记为 SGA 算法）. 为适合于优化计算，作者引入最大值统计量及其对应的 SA 算法（简称为 SMA 算法），并将 SMA 算法与 GA 算法相结合（记为 SGA(MAX)算法）. 新的算法不仅提高了算法的精度和降低了计算的复杂性，而且能克服 GA 算法中出现“早熟”的现象以及提供进行并行计算的可能性. 更主要的是新的方法为 GA 算法的精度、可信度和计算复杂性的定量分析提供了理论和方法上的有力工具.

关 键 词　遗传算法，统计推断，计算复杂性.

中图分类号　TP18

1　遗传算法

1.1　遗传算法

遗传算法（下面简记为 GA 算法）的提出可以追溯到 1967 年，2 位先驱者 Bagley 和 Rosenberg 在他们的博士论文[1,2]中就提出遗传算法的概念. 近 20 年来陆续有人在这方面进行研究，特别是近七、八年来遗传算法的研究和应用引起人们极大的兴趣. 80 年代初，Bethke[3]利用 WALSH 函数和框架变换方法，设计了一个确定框架的均值的有效方法，大大推进了对遗传算法的理论研究工作. 1987 年 Holland[4]推广了 Bethke 的方法. 如今遗传算法不但给出了清晰的算法的描述，而且也建立了一些定量分析的结果，并在各方面得到应用. 为叙述方便，下面简单介绍遗传算法中的一些术语结果.[5]

1.1.1　简单的遗传算法

设有一染色体集合 A，再设对每个染色体对应有一个对环境适应程度的度量，如设有一适应度函数 $f(\cdot)$，定义于染色体集合 A，其值表示对应的染色体的适应程度. 再给出一组遗传操作（复制、交换和突变）.

---

* 本文研究得到国家自然科学基金和国家 863 高科技项目基金资助. 作者张铃，1937 年生，教授，主要研究领域为人工智能理论，应用教学. 张钹，1935 年生，教授，博士导师，中国科学院院士，主要研究领域为人工智能及计算机应用.

本文通讯联系人：张铃，合肥 230039，安徽大学人工智能所

本文 1996-05-08 收到修改稿

遗传算法的操作可形式化为：

染色体用一个 L 维的向量(其每个分量只取有限值. 文中为讨论方便，设其分量仅取0、1两个值)，则所有长 L 的染色体集合就是一个 L 维的向量空间，记为 B_L，而染色体池 A 是一个染色体子集合(或说是 B_L 中的一个子集)，其中的元素可表示为

$$A=\{A_1,A_2,\ldots,A_n\}.$$

每个 A_i，则表示成

$$A_i=(a_1^i,a_2^i,\ldots,a_L^i).$$

适应度函数 $f(\cdot):B_L\to R^+$ 为实值函数. 其中 R^+ 表示非负实数集合，记为 $f(A_i)=f_i$.

遗传操作

复制：取 A_1 进行复制，即在池中增加一个 A_1.

杂交：取 A_1 和 A_2 进行杂交，令 $A_1=(a_1^1,a_2^1,\ldots,a_L^1)$，$A_2=(a_1^2,a_2^2,\ldots,a_L^2)$

随机取 $t(0<t<L)$，作新的染色体如下：

$$A_1'=(a_1^1,\ldots,a_t^1,a_{t+1}^2,\ldots,a_L^2)$$

$$A_2'=(a_1^2,\ldots,a_t^2,a_{t+1}^1,\ldots,a_L^1)$$

其中 A_1'，A_2' 就是由 A_1，A_2 杂交得到的新的染色体.

突变：取染色体 A，随机取整数 t，设 $A=(a_1,a_2,\ldots,a_L)$，作新的染色体 B，$B=(a_1,a_2,\ldots,a_{t-1},b_t,a_{t+1},\ldots,a_L)$，其中 $b_t=\bar{a}_t$(即 a_t 的非).

再给定概率 p_c，p_m(分别是杂交概率和突变概率).

遗传(GA)算法

(1) 每次进行遗传操作时，以概率 $p_i=f_i/\Sigma f_i$ 取到 A_i.

(2) 对已取出的染色体(进行杂交时，需同时取出2个染色体)，分别以概率 $(1-p_c-p_m)$，p_c，p_m，进行复制、杂交和突变3种遗传操作.

(3) 把经过遗传操作后得到的染色体都放入染色体池中. 对新得到的染色体，计算其适应度值.

若假定染色体池的容量一定，当染色体的个数超过容量时，就将适应度最小的染色体从池中删去.

(4) 进行上述的遗传算法至第 T 代后(T 是预先给定的常数)，则在第 T 代的染色体池中取适应度最大的染色体，即为所求的染色体.

注：这样求到的染色体，当然一般不一定是最优的染色体(这里最优是指对应的适应度值最大)，只是较优的染色体.

1.1.2 染色体框架

设染色体为长 L 的向量，若此种向量中若干个分量的值固定不变，其余的分量可任取0或1，这样，所有可能的不同向量的集合称为染色体的一个框架(Schema). 如 $L=6$，设其中第1,2,5分量的值分别为1,0,1(不变)，其余的分量可任取0或1，则其对应的框架就表示为 $H=(1,0,*,*,1,*)$，还不确定的分量以 $*$ 表示.

例如：设框架 $H=(1,*,0,*,1)$，则它共包含4个染色体：(1,0,0,0,1)，(1,1,0,0,1)，(1,0,0,1,1)，(1,1,0,1,1).

再对染色体框架引入一些术语.

定义. 设框架 H，称其中固定分量的个数为 H 的阶，记为 $O(H)$. 其第1个固定分量位

置与最后1个固定分量位置的差称为 H 的确定长,记为 $\delta(H)$.

例如:令 $H=(0,*,1,*,1,*,*,)$,则有 $O(H)=3,\delta(H)=5-1=4$.

到目前为止,遗传算法中最主要的定量结果是所谓的"框架定理",其内容如下:

设以 $m(H,t)$ 表示 H 在第 t 代时在染色体池中的元素个数,$f(H)$ 表示 H 在池中的平均适应度,$\bar{f}$ 表示染色体池中元素的平均适应度,p_c,p_m 分别表示杂交与突变的概率,$O(H),\delta(H)$ 分别表示 H 的阶与确定长,H 在第 $t+1$ 代时在染色体池中的元素个数记为 $m(H,t+1)$,则有下面定理.

框架定理. 框架 H 在第 $t+1$ 代时含在染色体池 A 中的元素个数 $m(H,t+1)$ 满足

$$m(H,t+1)\geqslant m(H,t)\left[\frac{f^{-}(H)}{f}(1-p_c\frac{\delta(H)}{L-1}-O(H)p_m)\right] \qquad (*)$$

其中 L 是池中染色体的长度.

若 $\delta(H),O(H)$ 比较小,可将最后括号中后2项略去不计,再设存在常数 $c>0$,使得 $f(H)\geqslant(1+c)\bar{f}$,则有:

推论. 若上述条件满足,则有 $m(H,t+1)\geqslant m(H,0)(1+c)^t$

这个推论说明具有高适应度(即满足 $f(H)\geqslant(1+c)\bar{f}$)、低阶(即 $O(H)$ 比较小)和短确定长(即 $\delta(H)$ 比较小)的染色体,在遗传过程中其个数将随代数按指数律增加. 这个结论也许可以说明,为什么在自然界按"优胜劣汰"的原则进行进化能使生物很好地适应不断变化的环境而代代相传.

另一方面,为使(*)式更准确,应将 $f(H),\bar{f}$ 换成 $f(H,t),\bar{f}(t)$ 才行,因为不同代池中的适应度的均值是不一样的. 这么一来,这个推论的假设条件太强,一般情况下是无法满足的,因为当遗传不断进行下去时,池中的染色体的均值将不断提高,从而使池中框架最大的 $f(H,t)$ 值与 $\bar{f}(t)$ 值之差越来越小,而不是总保持 $f(H,t)\geqslant(1+c)\bar{f}(t)$ 的关系. 这个关系若不能保持,则推论的结果也就不能成立.

下面举一个简单的例子来说明. 设开始时池中有10个染色体,5个 f 值为2,另5个 f 值为1. 下面列出经复制后各代的情况.

t	n	$m(H_1,t)$	$m(H_2,t)$	$f(H_1)$	$f(H_2)$	$\bar{f}(t)$
1	10	5	5	2	1	3/2
2	10	20/3	10/3	2	1	5/3
3	10	8	2	2	1	9/5
4	10	80/9	10/9	2	1	17/9
5	10	160/17	10/17	2	1	33/17
...						
i	10					c_{i+1}/c_i
...						

其中 $t,n,m(H_1)(m(H_2)),\bar{f}(t)$ 分别代表代数、每代的染色体个数、每代 $H_1(H_2)$ 的个数及每代适应度的均值.

由 $c_{i+1}=c_i-1,c_1=2$.

显然,当 $i\to\infty$,有 $c_i\to\infty,\bar{f}(t)\to f(H)=2$.

不存在常数 c,使 $f(H,t)\geqslant(1+c)\bar{f}(t)$ 对任意的 t 均满足.

1.1.3 框架定理的改进

本节我们将对框架定理进行必要的改进，克服上面所述的缺陷.

设染色体池中只有2个适应度不同的染色体，设框架 H 的适应度为 f_1，其余染色体的适应度为 f_2，且 $f_1>f_2$.

设第 t 代框架 H 的染色体个数占全体个数的比率为 $d(t)$，其染色体适应度的均值为 $\bar{f}(t)$.

显然，有 $$\bar{f}(t)=f_1-(1-d(t))(f_1-f_2) \tag{1}$$

另一方面，由复制定义容易得 $d(t+1)=d(t)f_1/\bar{f}(t)$ (2)

将(1)式代入(2)式得

$$d(t+1)=d(t)f_1/[f_1-(1-d(t))(f_1-f_2)]=d(t)/[1-(1-d(t))b] \tag{3}$$

其中 $0<b=(f_1-f_2)/f_1<1$

(3)式是关于 $d(t)$ 的非线性差分方程，要解此方程不是易事. 为了得到改进的框架定理，我们作如下处理.

因为 $b>0$，所以由(3)式得 $d(t)<d(t+1)\leqslant 1$

所以 $d(t)$ 是 t 的单调上升有上界的数列，故有有限极限. 令 $d(t)$ 的极限为 d. 对(3)式取极限得 $$d=d/[1-(1-d)b] \tag{4}$$

(4)式的解为 $d=0$ 或 $d=1$，但由 $d(t)>0$，得 $d=1$.

现对(1)取极限，当 $t\to\infty$ 时，$\bar{f}(t)\to f_1$.

也就是说，当复制的代数增加时，池中染色体的适应度的均值将趋向 f_1（最高的适应度值）.

上面是就池中只有2个不同的适应度值的情况进行推导的. 对于一般情况，设最高适应度的框架 H 在第 t 代的适应度为 $f_1(t)$，其余染色体的适应度均值为 $f_2(t)$. 由上面推导知，$f_2(t)$ 随 t 单调增加，而 $f_1(t)$ 将趋于框架 H 适应度的均值. 故对(3)式取极限时仍可得到(4)，于是上述的结论对一般情况也成立.

对同时进行3种遗传操作的情况，我们将得到类似(*)式的公式

$$m(H,t+1)\geqslant m(H,t)[\frac{f_1(H,t)}{f(t)}(1-p_c\frac{\delta(H)}{L-1}-O(H)p_m)] \tag{**}$$

框架定理（改进的）. 若染色体中存在有结构块(Building Block，即高适应度、低阶和短确定长的框架)，则经过若干代遗传操作后，池中染色体的平均适应度至少趋近池中结构块的最高适应度值.

这个定理没有对结构块的适应度作任何的限制的假定，这个定理同时证明了原定理的推论中假设：存在常数 c 对每一代都有 $f(H,t)\geqslant(1+c)\bar{f}(t)$ 的要求是不可能成立的.

现在的问题在于是否一定存在有结构块呢？若有，其最高的适应度值与池中的最高适应度值的关系如何？此值与染色体的最优值的关系又如何？

从框架定理的推导可以看出其结论性质主要是由复制操作效应（因证明中将杂交和突变的效应略去），而单单复制是不能发现新的东西的，特别是从目前的遗传算法对求全局的最优值的效果到底有多大尚不清楚.

从表面上看，简单的遗传算法与随机算法无多大区别. 为了便于对GA算法进行理论上

的分析,人们引入染色体框架的概念.下面我们将全部染色体框架表示成某种网络形式,这样也许更便于使用也更直观些.

易见,长为 L 的染色体集合中,所有可能的框架共有 3^L 种.

若将所有长 L 的染色体全体的集合看成是 L 维逻辑向量空间 B^L 中的点的全体,则所有染色体框架的集合就是这个空间中的全体超平面的集合.

在所有框架集合(将框架理解为超平面)中按集合的包含关系引入如下的半序关系:

$$H_1 < H_2 \leftrightarrow H_1 \subset H_2,$$

即 H_1 是 H_2 的某个子超平面.

显然,所有的框架在上述半序下构成一个半序格.对这样半序格用有向网络表示,则其对应的网络是一个无环有向图.我们要证明,(框架)遗传算法就相当于在上述有向图中进行随机的 BF(最好优先)搜索.

下面看一个简单的例子.设 $L=3$,则其对应的框架格网络如图1所示.此网络共4层.第0层只有1个节点,第1层有6个节点,第2层有12个节点,第3层有8个节点,共 $3^3=27$ 个元素,此网络不是树.显然,第 i 层的节点是所有 i 阶框架;第 i 层的每个节点有 $2(L-i)$ 个子节点,这些子节点均在第 $i+1$ 层上.

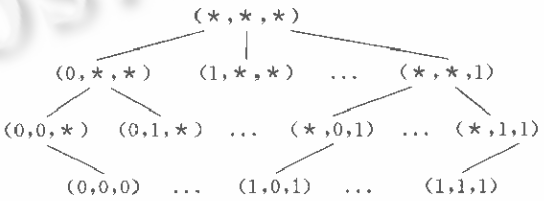

图1

按上述方法构造得到的网络称为对应的框架半序格网络,记为 P_L.

2　SA 算法与 GA 算法的关系

上面我们分析过 A 算法与 GA 算法的异同,简言之,可以将 GA 算法看成是随机型的 A 算法(指通常的启发式搜索算法.[6])我们在文献[6]中曾建立过 SA 算法(即将统计推断方法与 A 算法相结合),那么,将统计推断方法与 GA 算法相结合将是更加自然的了.

统计启发式搜索技术(SA 算法)实质上就是将 A-算法与给定的某个统计推断方法 S 相结合的一种搜索技术,它在展开节点时用 A-算法的"最好优先"原则,每展开一次时就对某些子集进行统计,求出其统计量,并用此统计量进行指定的统计推断法 S,利用统计推断的结果,对搜索图进行删除与剪枝,从而加快搜索的过程,提高搜索的质量.

SA 算法是将 A-算法与统计推断法相结合,而遗传算法,每次按适应度大小(适应度大,选取的概率也大)来选取染色体,并进行遗传操作,这一点与 A-算法的"最好优先"原则相似(可称之为随机的"最好优先"原则);另一方面,同一框架中染色体对应的适应值正好相当于一个随机变量的样本集合,遗传算法的每一步操作又是随机进行的,故经过遗传操作得到新染色体运算就相当于统计推断中的随机取样的运算.这些相似性提示我们:完全可以将统计推断方法与 GA 算法相结合.下一节将进行这方面的讨论.

3　SGA 算法(统计遗传算法)

3.1　SGA 算法

遗传算法可以看成是在一个定义域为 L 维的 B_L(为方便计,下面只限于讨论各向量的分量仅取0,1的情况)上有一适应度函数 $f(\cdot)$,现对此适应度函数依其适应度大小,随机进行遗传3个操作来求此函数的最大值.即将遗传算法看成是在某个空间中进行求最大值的搜索技术.

要将SA算法用于遗传算法,首先要将遗传算法的定义域化成有向图形式.

这一点可按下面方法来完成,利用"框架"概念,按上节的方法,将 B_L 中所有框架全体用其对应的半序格有向图来表示.于是各个框架对应的适应度函数值就是这个半序格图中各节点对应的统计量.这样一来,遗传算法就是在这个半序格有向图上的一个搜索技术.在这种形式下,我们就很容易将 SA 算法移植到遗传算法上来.

在实际进行时,不必取整个框架半序格 P_L,只要取其子网络 E_L,满足覆盖条件:对 E_L 中每个 $i(0\leqslant i\leqslant L-1)$ 阶节点 H,在 E_L 中 H 的子节点集合为 $\{H_1,H_2,\ldots,H_k\}$,有 $H\subset\bigcup H_i$,即 $\{H_1,H_2,\ldots,H_k\}$ 之并覆盖 H(将各节点理解为 B_L 中的超平面),就可以.如0阶框架 H 有 $2L$ 个子节点,只要取 $H_1=(0,*,\ldots,*)$ 和 $H_2=(1,*,\ldots,*)$,则 H_1 和 H_2 之并就可覆盖 H.

遗传算法中每次进行遗传操作得到新的染色体,相当于进行一次随机取样得到一个新的样本,得到新的染色体后进行各框架的适应度函数值的计算,相当于SA算法中对各子图计算对应的统计量.故将SA算法应用于遗传算法,比单纯用遗传算法多一个对各子图进行统计推断S,并根据统计推断的结果,对搜索图进行相应的删除与剪枝.

称将统计推断方法与遗传算法相结合得到的算法,统称为SGA算法.

SGA(ASM)算法(以ASM法为本算法的统计推断法S的SGA算法,称为SGA(ASM)算法).

关于ASM(Asymptotically Efficient Sequential Fixed-width Confidence Estimation of the Mean)法的详细内容请见文献[6,7].

在 P_L 中取一个满足覆盖条件的子网络 E_L,在 B_L 上定义一个适应度函数 $f(\cdot)$,给定概率 p_c,p_m,并以ASM法为本算法的统计推断法 S.

(1) 建立染色体框架对应的半序格网络的子网络 E_L.

(2) 令 $(*,*,\ldots,*)$ 为节点 s_0,求其对应的 $f(s_0)$.设定框架阶指标 i,开始时 $i=0$.

(3) 若 $i\geqslant L$,算法停止,当前池中适应度最大的染色体为所求;不然,展开节点 s_i,得其子节点,求各子节点 H_j 的 $f(\cdot)$ 值.

(4) 按概率 $p_j(p_j=f_j/\Sigma f_j)$,随机选取框架 H_j 在染色体池中的染色体(进行杂交操作时一次需选取2个染色体).

(5) 对被选出的染色体分别按概率 $(1-p_c-p_m)$,p_c,p_m 进行遗传的复制、杂交和突变操作,再将得到的新染色体放入池中.

(6) 对各子节点进行统计推断 S,并作出相应的判断,

(6.1) 若有某个子节点被接受,则将以其余的子节点为根的子图删去;

(6.2) 若有某子节点被拒绝,则将以该节点为根的子图删去;

(6.3) 若对某一层节点进行统计量时,均满足停止变量条件,但仍未能作出接受或拒绝的判断,则取 $f(\cdot)$ 值最大的2个染色体框架 H_1,H_2(或设 H_1 对应的区间 $I(H_1,\delta)$ 在数轴

的最右边,而取 H_2 是其对应的区间 $I(H_2,\delta)$ 与 $I(H_1,\delta)$ 相交,且 $H_1\cap H_2\neq\varnothing$ 的任一染色体). 若 $H_3=H_1\cap H_2\neq\varnothing$,令 $H_3=s_{i+1}$,若 $H_3=\varnothing$,则取 $H_1=s_i$,$i\leftarrow i+1$,回第(3)步.

(6.4) 不然,回到第(4)步.

(7) 若算法的计算量超过 Q 次(Q 是预先给定的正整数),则取在当前染色体池中适应度值最大的染色体为所求的解. 算法停止.

从算法本身看,SGA 算法与 GA 算法的不同主要表现在 SGA 算法多了一个对所得到的统计量进行统计判断,然后进行剪枝运算. 从其生物意义上看,SGA 算法与 GA 算法的差异就好比一个是人工育种,一个是自然繁殖,SGA 算法中的剪枝就好比是在人工育种中,有意识地将那些性能不好的品种删除,只保留性能好的品种继续进行培育,这样就大大加快了育种的速度. 人工育种几十年的结果比自然繁殖几千年的结果还要好. 这可看作是 SGA 算法比 GA 算法好的一个有力的佐证.

3.2 SGA(ASM)算法的分析

上一节我们引入了 SGA(ASM)算法,下面我们将 SA 算法的理论用于 SGA(ASM)算法,对 SGA(ASM)算法的计算复杂性进行定量分析.

显然,若不对适应度函数作一些限制,则 GA 算法是很难求到函数最优解的.

为了能用上 SA 算法的理论,对遗传算法中的适应度函数作一些必要的假设.

设 $f(\cdot):B_L\to R^+$ 的适应度函数. 固定一染色体框架,当我们对此框架中的染色体进行随机取样时,得到一组对应的适应度函数值$\{f(\cdot)\}$,将这些函数值看作是一组随机变量,对这些随机变量,做如下的假设.

假设 Ⅱ:设上述的$\{f(\cdot)\}$是一组互相独立的随机变量,且具有正方差及有限四阶矩. 设 $H_1,H_2,\ldots,H_k$ 染色体 B_L 中的一组同阶的框架集合,若适应度值最大的染色体 A 包含在 H_1 中,而不包含在其它的 H_i 中,则 $f(H_1)>f(H_i)$,$(i>1)$,其中 $f(H_i)$ 表示 H_i 在整个 B_L 中染色体的适应度函数值的均值.

若把适应度值最大的染色体的基因称为优化基因,则假设 Ⅱ 的生物意义是含有优化基因的染色体,从总体来说(统计意义)比不含优化基因的染色体对环境的适应能力更强.

从上面的生物意义看来,我们的假设还是比较合理的. 因为优化基因在长期进化的过程中之所以能保留下来,也许就是因为含有这些基因的染色体,它们对变化的环境的适应能力比其它的染色体更强的缘故. 当然,另一方面,我们也能举出不满足假设 Ⅱ 的函数 f 来. 这说明假设 Ⅱ 还存在着缺陷,下面我们将进一步研究.

有了这个假设,我们就可以对 SGA(ASM)算法中第(6.3)步的合理性进行说明.

(6.3)步合理性的说明,当 H_1,H_2 对应的统计量均满足停止变量条件时,这说明 H_1,H_2 对应的均值(以概率$>d$)是落在它们相应的区间内,而这些区间相交则说明它们的均值没有显著的差异,这可能是因为这 2 个染色体同时包含有最优染色体 A.

另一方面,令 $H_1=H_1^0\cup H_3$,$H_2=H_2^0\cup H_3$,H_1^0 和 H_2^0 不含有 A(最优染色体),$H_1\cap H_2=H_3$.

于是有 $f(H_1)=[d_1*f(H_1^0)+d_2*f(H_3)]$,$f(H_2)=[d_1*f(H_2^0)+d_2*f(H_3)]$,

因为 H_1^0 和 H_2^0 不含有 A(最优染色体),那么,若最优染色体 A 包含 $f(H_1)$ 和 $f(H_2)$ 中,则最优染色体必包含在 H_3 中,故由假设 Ⅱ 得 $f(H_1^0)$,$f(H_2^0)$ 均小于 $f(H_3)$. 故我们在

(6.3)步中取 $H_3=s_{i+1}$,然后算法继续进行;若 $H_3=\varnothing$,这可能是因为我们取的 δ 过大引起的,故我们取 $H_1=s_{i+1}$,然后继续算法进行,这样选择也是比较合理的.

另外有了假设Ⅱ,我们可以直接将SA算法有关计算复杂性的结论移植到SGA(ASM)算法上来.

定理1. 设定义在染色体空间 B_L 上的适应度函数 $f(\cdot)$ 满足假设Ⅱ,若对 B_L 运用强度为 α 的SGA(ASM)算法,当算法成功,则SGA(ASM)算法能以概率 $>(1-b)(b\leqslant\alpha)$ 找到最优解. 其计算复杂性的阶为 $O(L(\ln L))$.

这里"算法成功"是指算法进行到第 L 层后停止.

证明见文献[6].

4 作为优化计算的GA算法

我们研究GA算法的目的不在于从中研究出生物的进化过程,而是想从遗传过程中得出一个求最优化计算的普适算法.

若将上面的GA,SGA(ASM)算法应用于一般求最优化计算,发现有2个不足之处.

①框架的统计量的定义不当. 我们既然是求最优值,但每次却用均值作为统计量,而含"具有最大适应度染色体"的框架,其适应度的均值未必就一定最大,于是上节的假设Ⅱ就未必成立,假设Ⅱ不成立,那么定理1的结论也就有问题. 也就是说我们用框架的适应度的均值定义为框架的统计量,对求最优值来说不是最适当的. 从直观上看每次似乎应取框架当前适应度的最大值为统计量才较适宜.

②在SGA(ASM)算法中取样的随机性不足. 应用统计方法要求每次取样必须是随机的,但我们在SGA(ASM)算法中取样受到GA 3个操作的限制,其随机性就不能完全满足,而且复制的操作对求最大值没有直接的作用,因为复制对求最大值没有提供新的有用的信息,它只是间接地使选取到适应度大的染色体的概率增大,这个概率增大是否对求最大值有益呢?(若不对适应度的性状作一定的假设,就很难证明"复制"操作对求优有好处.)故由此可得GA操作对求最优化计算未必是一个好办法,这正如制造飞机未必一定要象飞鸟那样用"骨肉"构成. 所以我们可以从下面几个方面对GA算法进行改进.

(1) 引入最大值统计量及其对应的统计推断方法——MAX方法,以MAX方法为SGA算法中统计推断方法S,这样得到的算法记为SGA(MAX)算法.

(2) 进一步,除用MAX方法作为SGA算法中的统计推断法S外,对算法中的取样,不再受3种GA操作限制,而是完全按随机进行,这样得到的算法称之为SGA(MAX)算法.

关于SA(MAX)算法(包括算法的精、可信度、计算复杂性等)将另文讨论. 由于篇幅所限,这里直接利用MAX法的结论,得下面定理.

设函数为 $f:D\to R$ 上的有界可测函数,$\mu(D)=1$,$F(y)=P\{z\mid f(z)<y\}$ 连续,则有下面定理成立.

定理2. 给定 $0<\varepsilon<1$,则存在适当参数的MAX算法,用此MAX法对应的SGA(MAX)算法求解函数 $f(\cdot)$ 的最大值. 设算法结束时,求到的最大值为 $f(z^*)$,则:

$f(z^*)\in\{y\mid F(y)>1-e\}$ 的概率大于 $(1-\varepsilon)$,其中 $e=\dfrac{\varepsilon|\ln(\varepsilon/L)|}{L}$.

其计算复杂性为 $O(L^2/\varepsilon)$,其中 L 是染色体的码长.

定理说明:用 SGA(MAX)算法求解函数的最大值,其可信度的概率可以接近 $1(>1-\varepsilon)$;其精度也可以任意接近函数的真正的最大值(因为,当 $\varepsilon\to 0$ 时,$e\to 0$. 故$(1-e)\to 1$);其计算复杂性与染色体码长的平方和 ε 倒数乘积成正比. 总之,利用 *SGA* 算法,我们可以对 *GA* 算法的精度、可信度、计算复杂性进行定量的分析,这是其它方法所不及的.

5　几个模拟的结果

现举 3 个函数求最大值的例子,并与 GA 算法进行比较(例子是柳常青博士提供的).

(1)例 1:求函数 $f_1(x)=((1.02-(x-1/3)^2)^{10}$的最大值

SGA(MAX)计算次数分别为 $N=64,128,\ldots$,计算得到最优解为 $x=0.3333333(f_1(x)=1.2189944)$.

GA 计算次数为 $N=100,200,\ldots$,得到的最接近的值为 $x=0.3333216$.

(2)例 2:求函数 $f_2(x)=|(1-x)x^2*sin(200\pi x)|$的最大值

用 SGA(MAX)算法计算次数分别为 $N=64,128,\ldots$,求到 f_2 的最大值为 0.1481475 $(x=0.6675003)$,用 GA 算法算了 20 代,每代 100 个个体,求到的最大值为 0.1479531$(x=0.6624222)$.

注:$f_2(x)$在[0,1]区间内有 100 个局部极大值,故用传统的方法求优,相当困难.

(3)例 3:求函数 $f_3(x)=(1-2\sin^{20}(3\pi x)+\sin^{20}(20\pi x))20$ 的最大值

SGA(MAX)计算的总次数分别为 $N=64,128,\ldots$,得到的最优解 $x=0.1749125$, $x=0.8250875(f_3(x)=2231.01075)$.

GA 算法计算的次数为 $N=100,200,\ldots$,得到的最接近的值为:$x=0.8246953(f_3(x)=2052.376)$.

从上面初步模拟结果来看,SGA(MAX)算法比 GA 算法在精度和计算量方面都有改善,另外 SGA 算法提供了进行并行计算的可能性,同时 SGA(MAX)算法可以避免 GA 算法中的"早熟"现象的发生.

更主要的是 SGA(MAX)算法提供了对算法的精度、可信度和计算复杂性进行定量分析的有力的工具. 这为今后对 GA 算法进行深入地研究提供了方法和工具.

6　结　论

我们将 SA 算法引入 GA 算法之中,不但可提高算法的效率和精度,而且可以避免 GA 算法出现"早熟"的现象,更主要的是提供了对 GA 算法的精度、可信度和计算的复杂性进行定量分析的有力工具,为今后进一步的研究工作提供了新的途径.

参考文献

1　Bagley J D. The behavior of adaptive systems which employ geneticand correlation algorithms. Dissertation Abstracts International, 1968,**28(12)**.

2　Rosenberg R S. Simulation of genetic populations with biochemical properties. Dissertation Abstracts International, 1968,**28(7)**.

3 Bethke A D. Genetic algorithm as function oplimizers. Dissertation Abstracts International, 1981, **41(9)**.
4 Holland J H. Genetic algorithms and classifier systems: foundations and future directions, genetic algorithms and their applications. Proceedings of the Second International Conference on Genetic Algorithms, 1987. 82～89.
5 David E. Goldberg genetic algorithms. Addision-Wesley Publishing Company, Inc., 1989.
6 张钹,张铃. 问题求解理论及应用. 北京:清华大学出版社,1990.
7 Narayan C Giri. Introduction to probability and statistics. Marcel Dekker, INC. New York, 1975.

THE STATISTICAL GENETIC ALGORITHMS

*,**ZHANG Ling **,**ZHANG Bo

*(*Institute of Artificial Intelligence Anhui University Hefei* 230039)
**(*Department of Computer Science Tsinghua University Beijing* 100084)
**(*Laboratory of Intelligence Technology and Systems Beijing* 100084)

Abstract The deficiency of the schema theory in GA (genetic algorithms) and its improvement are discussed in this paper. The similarity between GA and heuristic search algorithm (a algorithm) and the probabilistic properties of GA are analyzed as well. Form the discussion, the similarity between GA and SA (statistical heuristic search) proposed by the authors is discovered. Therefore, when transferring the theory and results of SA to GA, a new statistical genetic algorithm can be established. In order to adapt to optimization computation, the maximal statistic and its corresponding SA called SMA are introduced. By combining the SMA and GA, a new algorithm SMA(MAX) is obtained. Using the new algorithm, the prematurity in general GAs can be overcome. The new algorithm also provides the possibility for parallel computing and a powerful tool for quantitative analysis of accuracy, confidence and computational complexity of GA.

Key words Genetic algorithm, statistical inference, computational complexity.
Class number TP18