

遗传算法 (综述)^{***}

任 平

(暨南大学数学系, 广州 510632)

摘 要

遗传算法提供了在很宽的领域中实用的优化技术, 本文综述其基本概念与近期的发展

关键词 遗传算法, 人工智能, 运筹学

分类号 AMS(1991) 68T, 92B; CCL TP18

近年来, 一种被称为遗传算法 (Genetic Algorithms, GA) 的优化技术有了很大发展, 由于在许多重要领域获得成功应用, 它受到普遍关注而成为当前十分热门的研究领域

1 GA 的基本概念

遗传算法, 有的作者译为基因算法^[1], 是一种模拟自然选择和遗传机制的寻优程序, 它是 60 年代中期美国密执安大学 Holland J H 教授首先提出^[2]并随后主要由他和他的一批学生发展起来的。

把计算机科学与进化论撮合到一块的最初尝试是在 50 年代末 60 年代初。但由于过分依赖突变 (mutation) 而不是配对 (mating) 来产生新的基因, 所以收效甚微。Holland 的功绩在于开发一种既可描述交换也可描述突变的编码技术, 这是最早的遗传算法, 文献中现在把它称为简单遗传算法 (Simple GA, SGA)

一般的遗传算法由四个部分组成: 编码机制、控制参数、适应度函数、遗传算子。

编码机制 (encoding mechanism) 这是 GA 的基础。GA 不是对研究对象直接进行讨论, 而是通过某种编码机制把对象统一赋予由特定符号 (字母) 按一定顺序排成的串 (string)。正如研究生物遗传, 是从染色体着手, 染色体则是由基因排成的串。在 SGA, 字符集由 0 与 1 组成, 码为二元串。对一般的 GA, 自然可不受此限制。串的集合构成总体, 个体就是串。对 GA 的码可以有十分广泛的理解。在优化问题, 一个串对应于一个可能解; 在分类问题, 串可解释为一个规则, 即串的前半部为输入或前件, 后半部为输出或后件、结论, 等等。这也正是 GA 有广泛应用的重要原因。

适应度函数 (fitness function) 优胜劣败是自然进化的原则。优、劣要有标准。在 GA,

* 本文 1996 年 1 月 16 日收到

** 国务院侨办重点学科科研基金资助项目

用适应度函数描述每一个体的适宜程度。对优化问题,适应度函数就是目标函数。引进适应度函数的目的在于可根据其适应度对个体进行评估比较,定出优劣程度。为方便起见,在SGA,适应度函数的值域常取为 $[0, 1]$ 。

遗传算子 (genetic operator) 最重要的算子有三种: 选择 (selection) 交换 (crossover), 突变 (mutation)。

选择算子也称复制 (reproduction) 算子。它的作用在于根据个体的优劣程度决定它在下一代是被淘汰还是被复制。一般地说,通过选择,将使适应度即优良的个体有较大的存在机会,而适应度小即低劣的个体继续存在的机会也较小。有很多方式可以实现有效的选择。例如,两两对比的方式,即随机从父代抽取一对个体进行比较,较好的个体在下一代将被复制继续存在。SGA采取的则是按比例选择的模式,即适应度为 f_i 的个体以 $f_i / \sum f_k$ 的概率继续存在,其中分母为父代中所有个体适应度之和。

如果只有选择算子,GA不会有什么新意。因为后代的群体不会超出初始群体即第一代的范围。因此,还需要一些更合理的算子。常用的有交换和突变。

交换算子有多种形式,最简单的是所谓的单点交换 (single point crossover),这也是SGA使用的交换算子,即从群体中随机取出两个字符串,设串长为 L ,随机确定交叉点,它在1到 $L - 1$ 间的正整数取值。于是,将两个串的右半段互换再重新连接得到两个新串。当然,得到的新串不一定都能保留在下一代,需和原来的串 (亲本) 进行比较,保留适应度大的两个,如图1所示。

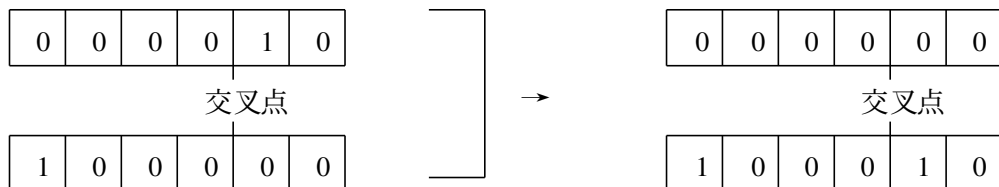


图 1 单点交换

进行交换后,可进行突变。突变算子是改变字符串的某个位置上的字符。在SGA,即为0与1互换: 0突变为1,1突变为0,如图2所示。一般认为,突变算子重要性次于交换算子,但其作用也不能忽视。例如,若在某个位置上,初始群体所有串都取0,但最优解在这个位置上却取1,于是只通过交换达不到1而突变则可做到。

控制参数 (control parameters) 在GA的实际操作时,需适当确定某些参数的值以提高选优的效果。这些参数是

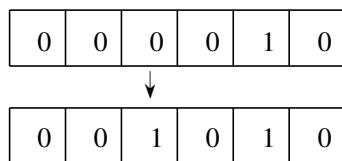


图 2 突变

字符串所含字符的个数,即串长。在SGA,这一长度为常数,即为定长,记为 L ;

每一代群体的大小,即所包含字符串的个数,也称群体的容量,记为 n ;

交换率 (crossover rate),即施行交换算子的概率,记为 P_c ;

突变率 (mutation rate),即施行突变算子的概率,记为 P_m 。在SGA,若群体容量较大,如 $n = 100$,通常取 $P_c = 0.6$, $P_m = 0.001$;若群体容量较小,如 $n = 30$,通过取 $P_c = 0.9$, $P_m = 0.01$ 。此外还有遗传的“代”数,或其他可供确定中止繁殖的指标,等等。

为具体理解GA的思想,现以一个可用手算求解的简例说明。此例取自GA一本著名的

教材^[3]。

例 求函数 $f(x) = x^2$ 的最大值, 变量 x 在 0 与 31 之间的整数取值。

解 为用 SGA 解此问题, 容易想到将决策变量 x 取的值以二进制数表示从而得到一种自然的编码; 每一个体均为长度是 5 的二进制位串

初始群体的容量取 4 于是, 从总体中随机抽取 4 个个体组成第一代群体, 即初始群体。具体操作可通过掷硬币确定。例如, 将一枚硬币连续掷 20 次, 或指定了顺序的 5 枚硬币各掷 4 次, 正面为 1, 反面为 0, 得 4 个 5 位二进制字符串, 不妨记为 (01101), (11000), (01000), (10011)。SGA 采取按适应度大小比例进行选择的机制, 则可用专门设计的简易轮盘来决定第一代群体中哪个个体能被保留。结果如表 1 所示

表 1 第一代群体的选择

串 No.	初始群体 (随机生成)	x	$f(x) = x^2$	$\sum f$	$\frac{nf}{\sum f}$	实际生存数 (由轮盘决定)
1	01 101	13	169	0. 144	0. 576	1
2	11 000	24	576	0. 492	1. 968	2
3	01 000	8	64	0. 055	0. 220	0
4	10 011	19	361	0. 309	1. 236	1
和			1170	1. 000	4. 000	4
平均			293	0. 250	1. 000	1
max			576	0. 492	1. 968	2

交换率取为 1, 即肯定施行交换算子。同样, 可通过掷硬币的方法将复制出的 4 个串配成两对并随机确定交叉点进行交换, 结果如表 2

表 2 交换的进行

	交换后的配对 (竖线为交换点的位置)	配对 (随机选择)	交叉点 (随机选择)	新种群	x	$f(x) = x^2$
	0110 1	2	4	01100	12	144
	1100 0	1	4	11001	25	625
	11 000	4	2	11011	27	729
	10 011	3	2	10000	16	256
和						1754
平均						439
max						729

突变率取为 0. 001, 这意味着每 1000 位平均有一位产生突变。本例的群体只包含 4 个 5 位的字符串共 20 位, 平均只有 0. 02 个位产生突变, 可认为实际上不产生突变。于是, 经过选择、交换完成了一代的遗传。事实证明, 第二代群体的质量有了明显的提高, 平均适应度由 293 增加为 439, 最大适应度由 576 增加到 729

2 应 用

GA 最直接的应用或许就是多元函数的优化问题。如果函数表达式是清楚的, 又具有良好的分析性质, 自然用不着 GA。但若所讨论的函数受到严重的噪声干扰而呈现出非常不规则的形状, 同时所求的也不一定非是精密的最大 (小) 值, 则 GA 就可以找到它的应用。

一般说来,在一些实际问题中应用 GA 时,首先要解决两个问题: (1) 编码的设计; (2) 适应度函数的确定。对函数优化问题,这两个问题乍一看来都比较容易解决。特别是,有现成的二进位制可将变量取的值编为二元的位串,很是方便。实际不然。因为对一般的二进位制,有时增加 1,需要改变所有的位元,如 011 为十进制的 3,而 4 为 100,这就造成困难。为此,设计了一种所谓的“灰码”(gray codes)。这种码的特点是增加或减少 1 都只需要改动一个位元。当然,根据不同需要还可设计各种不同的码。这在包括飞机设计、信号检测以至非线性差分方程组稳定域的确定等近百种问题中有广泛应用。关于这方面的情况可参阅文 [3] 表 4.4 所列举的项目。

GA 另一个十分引入注目的应用领域是机器学习。这是因为,与传统的优化技术比较,GA 的特点在于

- 1) GA 的工作对象不是决策变量本身,而是将有关变量进行编码所得的码,即位串;
- 2) 传统的寻优技术都是从一个初始点出发,再逐步迭代以求最优解。GA 则不然,它是从点的一个群体出发经过代代相传求得满意解;
- 3) GA 只充分利用适应度函数(目标函数)的信息而完全不依靠其它补充知识;
- 4) GA 的操作规则是概率性的而非确定性的。

这就表明,GA 较为适合于维数很高、总体很大、环境复杂、问题结构不十分清楚的场合,机器学习就属这类情况。一般的学习系统要求具有随时间推移逐步调整有关参数或者改变自身结构以更加适应其环境,更好完成目标的能力。由于其多样性与复杂性,通常难以建立完善的理论以指导整个学习过程,从而使传统寻优技术的应用受到限制,而这恰好能使 GA 发挥其长处。关于 GA 在机器学习中的应用,Jonng K De 在文 [4] 中有很好的综述。

有时,问题的数学结构十分清楚,但由于某些原因,如维数太高或计算量太大,依靠传统方法实际上难以求解。运筹学中许多排序问题,如旅行商问题、工序安排、设备布置等都属此类。GA 在这方面也有大量工作。以旅行商问题为例:设有几个城市,则很自然会把旅行方式编为 n 个正整数 $\{1, 2, \dots, n\}$ 的一个排列;但是,这种编码不适用于 GA,因为交换算子作用于两个排列所得的结果有时是没有意义的。为克服这个困难,或是要设计新的编码规则;或是要适当修改原有的交换算子。对这方面工作的评述可参阅文 [5]。GA 在这一领域中的应用,引起运筹学界的重视:1995 年,国际运筹学专业期刊《Computers & Operation Research》以 GA 为主题编发了一期特刊^[6],组织了包括四个方面内容的 12 篇论文。期刊主编 Raff S J 在特刊前言中把 GA 称为“对当前运筹学有巨大兴趣的课题。”这一点决非偶然,进入 70 年代后期,运筹学面临新的挑战^[7],由于研究工作过分“学院化”,使理论发展在一定程度上与实际脱节。原来生气勃勃的运筹学出现了沉闷的局面;另一方面,计算技术、控制技术的智能化趋势已不能满足于传统优化方法的框架;特别是,传统运筹学理论缺乏对变动环境的适应性,“造成‘最优’解维持最优的寿命在很多情况下甚至比求解所耗费的时间还要来得短”。因此,GA 的崛起,对运筹学无疑是一个有力的推动。

在复杂性问题的研究中,GA 也崭露头角,倍受青睐。什么叫复杂性问题,各家看法不一。共同认识还是有的,即复杂性问题应是多层次、多因素,其相互作用是非线性、不确定和不稳定的,这样的学习问题自然属复杂性研究的范畴。事实上,在复杂系统例如适应性系统学习策略的研究中,GA 占重要地位^[8]。另一个典型例子是地震数据反演问题^[9]。由于介质参数的模型非常大,同时观测数据不完备,噪音的存在,源的情况复杂且未知,很难用传统的方

法求得目标函数的全局最优值,而只能求一定意义下的“满意解”。这时,可供选择的方法之一自然是 GA

值得注意的是,近年来,GA在商业应用方面取得一系列重要成果 或许这也是它受到学术界之外的企业界 政府部门以及更广泛的社会阶层普遍重视的原因。GA的商业应用五花八门,覆盖面甚广,Coldborg 在 Comm. ACM 上的一篇专论^[10]较为详细地介绍了美国近年来的一些成果 例如,通用电器的计算机辅助设计系统 Engeneous, 这是一个混合系统 (hybrid system),采用了 GA以及其他传统的优化技术做为寻优手段 Engeneous 已成功地应用于汽轮机设计,并改善了新的波音 777发动机的性能 美国新墨西哥州州立大学心理学系开发了一个所谓的 Faceprint 系统,可根据目击者的印象通过计算机生成嫌疑犯的面貌 计算机在屏幕上显示出 20种面孔,目击者按十分制给这些面孔评分。在这基础上,GA按通常的选择、交换和突变算子生成新的面孔 Faceprint 的效果很好,已申报专利 同一个州的一家企业——预测公司 (Prediction Company)则首先开发了一组用于金融交易的时间序列预测和交易工具,其中 GA起了重要作用: 据说,这一系统实际运行效果很好,可以达到最好的交易员的水平,引起银行界的关注 GA在军事上的应用也有报道: 如用于红外线图象目标判别的休斯遗传程序系统 (Hughes genetic programming system),效果很好,以至准备把它固化成硬件。

在 GA的应用研究和推广普及中,相应的程序环境的开发占有重要地位 文 [11]将 GA程序环境 (GA Programming Environments)区分为三类: 面向应用系统,面向算法系统以及工具 表 3列出了文 [11]所评价的一些系统,文中并附有系统开发者的联系地址

表 3 文 [11]所列 GA程序的分类

面向应用系统	面向算法系统		工 具	
	特定算法系统	算法库	教育系统	通用系统
Evolver	Escapade	EM	GA Workbench	Engeneer
Omega	GAGA			GAME
Pc/Beagel	GAUCSD			Micro GA
Xpert Rule	Genesis	OOGA		Pegasus
Gen Asys	Genitor			Splicer

例如,Splicer为 NASA 约翰逊航天中心信息系统理事会软件技术部研制的,是当时最完备的一种通用系统 它由三个部分构成: GA核、可交换表现库和适应度模块;有两个用户界面库,一对 Macintoshes,一对 X windows 当用户使用 Splicer解某个具体问题时,只要用 C构造一个适应度模块而无需其他编程工作 为大力推广 GA,NASA以及政府项目承担者可免费使用 Speicer 1. 0版本 这些系统的研制开发和推广使用,对 GA在美国的发展无疑起着积极推动作用

3 GA的理论基础

GA在应用上确已取得显著成功,但其基础研究相对来说却比较薄弱 就目前情况看,能称得上为理论成果的只有所谓的型式定理^[12] (Schema Theorem)。

型式 (schema) 是在某些确定位置上取相同字符的字符串集合。例如,在 SGA形式

* * 000 表示后三个位置都取 0 的字符串的集合, 即 {00000, 01000, 10000, 11000}; 类似地, $10^* 0^*$ 表示 {10000, 10001, 10100, 10101}。因此, 可以直观地把型式看成是一个相似性模板 (similarity template), 在对应的确定位置, 0 和 0 匹配, 1 和 1 匹配, * 为通配符, 可和 0 1 两者匹配。型式表示的就是和它匹配的字符串。这些字符串称为型式的实现 (instance) 或解 (solution)。在某个型式中, 确定字符的个数称为型式的阶 (order), 其最左端确定字符到最右端确定字符间的距离称为型式的定义长度 (defining length), 例如, 型式 * * 000 的阶是 3, 定义长度是 $5 - 3 = 2$, 型式 $10^* 0^*$ 的阶是 3, 定义长度是 $4 - 1 = 3$ 。

为简便起见, 只就 SGA 进行讨论。引入下列符号

H : 某个型式;

f_i : 第 i 个字符串 (解、实现) 的适应度;

$\bar{f}(t)$: 第 t 代群体的平均适应度;

$\bar{f}(H, t)$: 型式 H 在第 t 代群体的平均适应度;

$n(H, t+1)$: 由 (第 t 代) 型式 H 的某个解 i 在第 $t+1$ 代生成的后代数的期望值;

$n(H, t)$: 第 t 代属于型式 H 的解的个数;

$W(H)$: H 的定义长度;

$o(H)$: H 的阶。

先考虑选择的效果。在 SGA, 选择标准是采用按适应度大小比例的原则, 因此, 第 i 个体经选择算子的作用在下一代继续存在的个数的期望值为 $n(f_i \sum f)$, 注意到

$$\bar{f}(H, t) = \frac{1}{n(H, t)} \sum f_i$$

则 $n(H, t+1) = n(H, t) \cdot \bar{f}(H, t) / \bar{f}(t)$

上述等式表明, 选择算子的作用将使适应度高于 (低于) 平均水平的型式在代代相传时增大 (减小) 其容量, 从而提高群体的素质。

现在分析交换算子的作用。若不进行交换或虽交换但交叉点落在型式最左、右两端确定字符所处位置之外, 该型式在下一代显然能被保留。于是, 型式 H 在下一代得以继续存在的概率 P_s 应满足

$$P_s \geq 1 - P_c \cdot W(H) / (L - 1)$$

综合考虑选择和交换的作用则有

$$n(H, t+1) \geq n(H, t) \bar{f}(H, t) \cdot [1 - P_c \cdot W(H) / (L - 1)] / \bar{f}(t)$$

最后, 由于 P_m 表示突变算子作用的概率, 不变的概率为 $1 - P_m$, 若所有确定字符都不变, 型式 H 在下一代自然能继续存在, 概率为 $(1 - P_m)^{o(H)}$, P_m 通常很小, $P_m \ll 1$, 则经过突变算子作用后, H 继续存在的概率为

$$(1 - P_m)^{o(H)} \approx 1 - P_m \cdot o(H)$$

被淘汰的概率近似地为 $o(H) \cdot P_m$ 。于是, 综合考虑选择、交换和突变算子的作用, 最后有

$$n(H, t+1) \geq n(H, t) \bar{f}(H, t) \left[1 - P_c \cdot \frac{W(H)}{L-1} - o(H) \cdot P_m \right] / \bar{f}(t)$$

这就是所谓的型式定理, 也称 GA 的基本定理。型式定理表明, 如果只考虑选择算子的影响, 某个型式所含解的个数随同群体代代相传而递增或减少与型式的平均适应度大小有关。特别地, 若有 $\bar{f}(H, t) = f(t)(1 + c)$, $c > 0$ 为常数, 则

$$n(H, t) = n(H, t-1)(1 + c) = n(H, 0)(1 + c)^t$$

即高平均适应度的型式在与其它型式的竞争中其容量呈指数增长。然而,只是高平均适应度还不足以保证高的成长率。当综合考虑其它算子影响时,要求优良的型式的定义长度应较短,阶数应较低。高平均适应度、短定义长度和低阶数的型式在群体代代相传中其所含解的个数是指数增长,正是型式定理的本质。

但是,低阶数意味着较大的搜索空间,这对寻优自然是不利的。GA的实践表明,GA的威力不仅仅在于开始时能将搜索方向引导到高适应度低阶数的型式,随后,还能利用交换算子将低阶型式的信息组合成高阶型式,从而最后收敛于高适应高阶数、即较小的搜索空间。这一事实并非普遍成立,但确实是非常重要的,被称为建筑块假设 (building blocks hypothesis)。有趣的是,建筑块假设揭露了 GA的工作方式和傅立叶分析之间的某些相似之处。特别是,Bethke提出了所谓的 Walsh-型式交换,并用它有效地计算型式的平均适应度^[12]。这方面的进一步工作可参看文 [5] 所介绍的文獻。

4 展 望

GA在应用方面取得的丰硕成果,使人们对它的发展前景充满信心。认识到这一点,GA的奠基人之一,Goldberg D戏言^[10]:“已不再需要水晶球”。

今后几年,可以预期,拓广更加多样的应用领域,其中包括各种 GA程序设计环境的开发,仍将是 GA发展的主流。事实上这也是本世纪高新技术迅速发展带有规律性的特点,即面向应用。与此同时,这并不意味着理论研究会被忽视,这方面同样有大量工作要做。例如:

控制参数的选择;

交换和突变这两类最重要的算子的确切作用;

并行 GA和分布式 GA的研究;

其他类型生物机制的模仿,如免疫、病毒、寄生等,以丰富 GA的内容,等等。

自然,不论从理论还是应用的角度看,最紧迫的应是关于算法收敛性问题的研究,特别是过早收敛的防止。这对 GA的实际应用关系重大。

当前一个特别值得重视的趋势是一些面向对象的智能技术,其中主要是模糊逻辑 (Fuzzy Logic, FL),神经网络 (Neural Network, NN) 以及 GA等的综合应用。众所周知,FL有较强的知识表达能力,NN的长处在于自学习,它们与 GA相结合形成新的集成化技术,即所谓的混合智能系统 (Hybrid Intellectual System)。这一思想在 90年代初逐步形成,而由模糊集论的创始人,美国 Zadeh L A在 1993年于汉城召开的国际模糊系统协会 (IFSA) 第五届世界会议首先明确提出^[13],随后在许多有关的国际学术会议上得到充分体现。应该指出,我国学者对这一趋势的认识较早。例如,清华大学李衍达院士领导的研究集体在几乎同一时期开展了这一重要方向的研究^[14]。1995年,Zadeh在 IFSA的第六届世界会议上再次强调了这一方向的重要性,并且认为上述所谓的混合智能系统的应用将覆盖从消费品生产到核反应堆设计以至证券管理,而“在未来几年中可能无处不在”。^[15]

就 GA本身的研究而言,应该说,我国起步较晚,近几年才陆续看到一些介绍性的文章,不多于两三部的专著以及初步的研究报告。和国外工作比较,一个显著区别是,国内工作多只停留在论文这一层次,几乎没有看到具体实际应用,与研究成果商品化的差距就更远。理论研究与实际应用不够紧密,阻碍了我国高新技术的迅速发展,几乎已经成为顽症。因此,在我国发展 GA,当前应该特别重视它的应用和推广普及。学术界要主动和企业界联手开发

GA 的应用,要重视引进或自行研制类似于 Splicer 的程序设计环境,使 GA 的应用更加方便和快捷。国家组建的工程研究中心应该在这方面发挥更大的作用。工科数学教育也应有所调整,以适应高新技术发展的需要。例如,工科运筹学和最优化方法的课程应该适当加入有关 GA 等方面的内容,以开阔学生的视野,同时也有利于加快传统课程内容的更新。

参考文献

- 1 Holland J H. 基因算法. 科学, 1992; 11: 24~ 31
- 2 Holland J H. Adaptation in natural and artificial systems. Univ of Michigan Press, Ann Arbor Mich, 1975
- 3 Goldberg D E. Genetic algorithms in search. Optimization and Machine Learning, Addison-Wesley, Reading, Mass, 1989
- 4 De Jong K A. Learning with genetic algorithms an overview. Machine learning 3, Kluwer Academic, Hingham, Mass, 1988; 121~ 133
- 5 Forrest S. Genetic algorithms principles of natural selection applied to computation. Science, 1993; vol 261, 872~ 878
- 6 Computer & operation research, 1995; vol 22, no 1
- 7 桂湘云, 章祥荪. 关于 OR 发展前途的讨论及其对 OR 应用工作的启示. 运筹学杂志, 1987; 6(1): 73 ~ 80
- 8 赵凯华. 适应性系统与自涌行为. 复杂性研究, 中国科学院《复杂性研究》编委会, 北京: 科学出版社, 17~ 30
- 9 马在田, 王华忠. 地震数据反演中的非线性问题. 自然杂志, 1995; 17 78~ 88
- 10 Goldberg D E. Genetic and evolutionary algorithms come of age. Comm ACM 1994; 37(3): 113~ 119
- 11 Jose L, Ribeiro Filho etc. Genetic-algorithms programming environments. Computer, June, 1994; 28 ~ 43
- 12 Bathke A D. Thesis, Univ of Michigan, Ann Arbor, MI, 1980(转引自文 [5])
- 13 Zadeh L A. Fuzzy logic, Neural Networks and soft computing, 5th IFSA World Congress, Seoul, 1993
- 14 张良杰, 李衍达. 模糊神经网络技术的新近发展. 信息与控制, 1995; 24(1): 39~ 46
- 15 Zadeh L A. New frontiers in fuzzy logic. 6 IFSA World Congress, Sao Paulo, 1995

Genetic Algorithms (an Overview)

Ren Ping

(Jinan University, Guangzhou 510632)

Abstract

GA provide an alternative to traditional optimization techniques that mimic natural evolution to solve problems in a wide variety of domains. The basic idea and recent advances are reviewed in this paper.