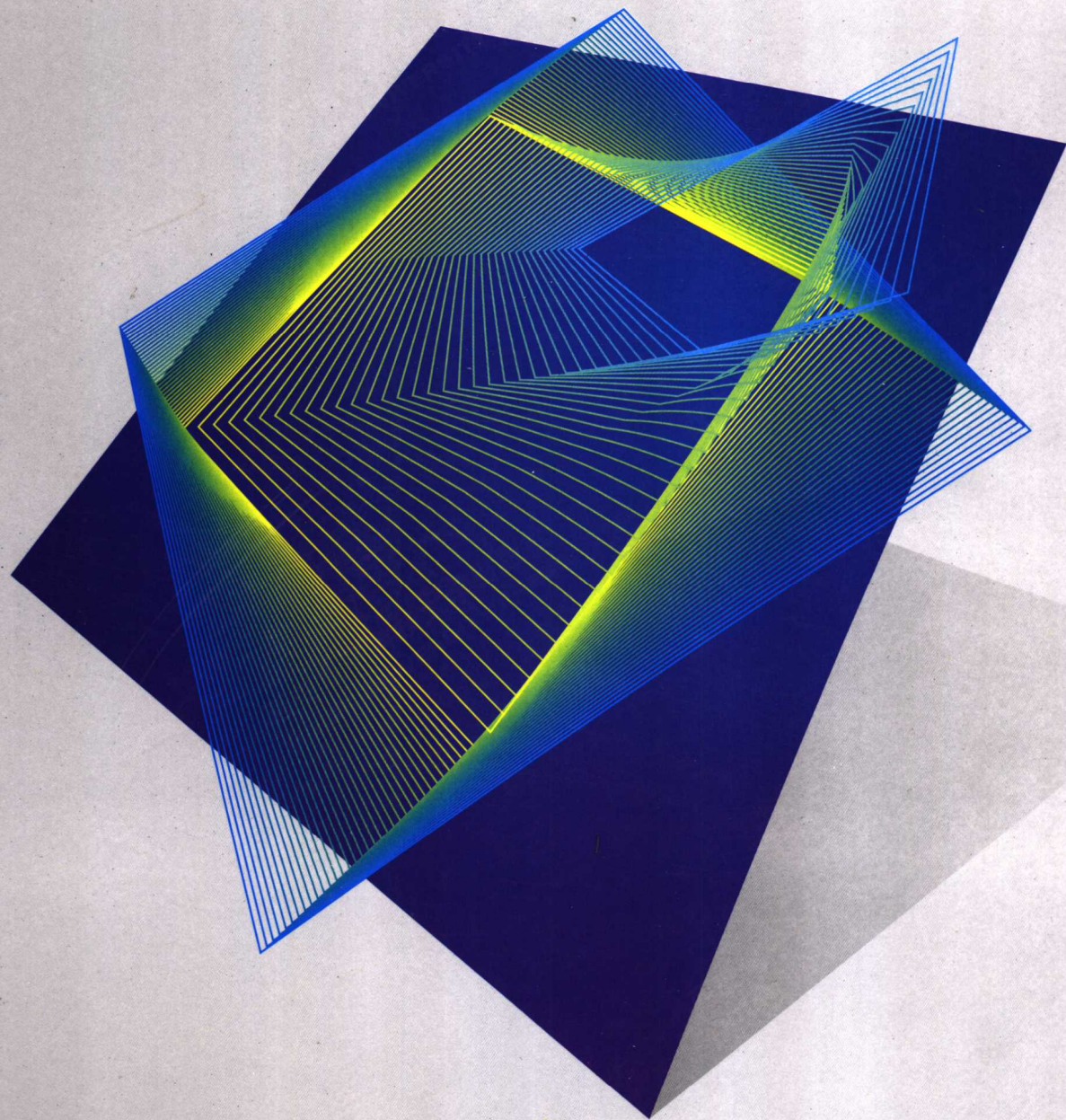


MATLAB

遗传算法工具箱及应用

雷英杰 张善文 李续武 周创明 编著



西安电子科技大学出版社
[http:// www.xduph.com](http://www.xduph.com)

MATLAB

遗传算法工具箱及应用

雷英杰 张善文
李续武 周创明 编著

西安电子科技大学出版社

2005

内 容 简 介

本书系统介绍 MATLAB 遗传算法和直接搜索工具箱的功能特点、编程原理及使用方法。全书共分为 9 章。第一章至第四章介绍遗传算法的基础知识,包括遗传算法的基本原理,编码、选择、交叉、变异,适应度函数,控制参数选择,约束条件处理,模式定理,改进的遗传算法,早熟收敛问题及其防止等。第五章至第七章介绍英国设菲尔德(Sheffield)大学的 MATLAB 遗传算法工具箱及其使用方法,举例说明如何利用遗传算法工具箱函数编写求解实际优化问题的 MATLAB 程序。第八章和第九章介绍 MathWorks 公司最新发布的 MATLAB 遗传算法与直接搜索工具箱及其使用方法。

本书取材新颖,内容丰富,逻辑严谨,语言通俗,理例结合,图文并茂,注重基础,面向应用。书中包含大量的实例,便于自学和应用。

本书可作为高等院校计算机、自动化、信息、管理、控制与系统工程等专业本科生或研究生的教材或参考书,也可供其他相关专业的师生及科研和工程技术人员自学或参考。

图书在版编目(CIP)数据

MATLAB 遗传算法工具箱及应用/雷英杰等编著.

—西安:西安电子科技大学出版社,2005.4

ISBN 7-5606-1484-1

I. M... II. 雷... III. 计算机辅助计算—软件包, MATLAB IV. TP391.75

中国版本图书馆 CIP 数据核字(2004)第 142719 号

策 划 戚文艳

责任编辑 龙 晖 戚文艳

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 16.875

字 数 397 千字

印 数 1~4000 册

定 价 26.00 元

ISBN 7-5606-1484-1/TP·0788(课)

XDUP 1755001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

MATLAB 是 MathWorks 公司推出的一套高性能的数值计算和可视化软件。它集数值分析、矩阵运算、信号处理和图形显示于一体，构成一个方便的、界面友好的用户环境。MATLAB 强大的扩展功能和影响力吸引各个领域的专家相继推出了许多基于 MATLAB 的专用工具箱。MATLAB 强大的科学运算、灵活的程序设计流程、高质量的图形可视化与界面设计、便捷的与其他程序和语言的接口等功能，使之成为当今世界最有活力和最具影响力的可视化软件。

遗传算法(Genetic Algorithm, 简称 GA)是以自然选择和遗传理论为基础，将生物进化过程中适者生存规则与群体内部染色体的随机信息交换机制相结合的高效全局寻优搜索算法。GA 摒弃了传统的搜索方式，模拟自然界生物进化过程，采用人工进化的方式对目标空间进行随机优化搜索。它将问题域中的可能解看做是群体的一个个体或染色体，并将每一个个体编码成符号串形式，模拟达尔文的遗传选择和自然淘汰的生物进化过程，对群体反复进行基于遗传学的操作(遗传、交叉和变异)。根据预定的目标适应度函数对每个个体进行评价，依据适者生存、优胜劣汰的进化规则，不断得到更优的群体，同时以全局并行搜索方式来搜索优化群体中的最优个体，以求得满足要求的最优解。

自从 1975 年 John H. Holland 教授出版关于 GA 的经典之作《Adaptation in Natural and Artificial Systems》以来，GA 已获得广泛应用。从遗传算法的整个发展来看，20 世纪 70 年代是兴起阶段，20 世纪 80 年代是发展阶段，20 世纪 90 年代是高潮阶段。遗传算法作为一种实用、高效、鲁棒性强的优化技术，发展极为迅速，已引起国内外学者的高度重视。

遗传算法提供了一种求解非线性、多模型、多目标等复杂系统优化问题的通用框架，它不依赖于问题具体的领域，已经广泛应用于函数优化、组合优化、自动控制、机器人学、图像处理、人工生命、遗传编码、机器学习等科技领域，并且已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面的应用取得了成功。

由于 GA 在大量问题求解过程中独特的优点和广泛的应用,许多基于 MATLAB 的遗传算法工具箱相继出现,其中出现较早、影响较大、较为完备者当属英国设菲尔德(Sheffield)大学推出的基于 MATLAB 的遗传算法工具箱。另外,还有美国北卡罗莱纳那州立大学推出的可与 MATLAB 一起使用的遗传算法优化工具箱 GAOT(Genetic Algorithm Optimization Toolbox)。考虑到前者在内容上已经覆盖到后者,因此本书将着重介绍英国设菲尔德大学的基于 MATLAB 的遗传算法工具箱。值得注意的是,MathWorks 公司最新发布了一个专门设计的 MATLAB 遗传算法与直接搜索工具箱(Genetic Algorithm and Direct Search Toolbox),本书同时也详细介绍了这个遗传算法与直接搜索工具箱及其使用方法。

书中通过大量实例,介绍了如何利用提供的遗传算法工具箱函数来编写 MATLAB 程序,解决实际问题。

在此,作者非常感谢西安电子科技大学王宝树教授、周利华教授、李荣才教授等的指导和鼓励,以及空军工程大学计算机系吕辉教授等的支持和帮助,真诚感谢西安电子科技大学出版社的大力支持。

需要特别指出,虽然作者竭尽所能,精心策划章节结构和内容编排,详细测试书中的每一个实例,尽可能简明而准确地表述其意,但限于水平和资料,书中的错误和不足之处在所难免,恳请读者不吝指正。

作 者

2004 年 10 月

目 录

第一章 遗传算法概述	1
1.1 遗传算法的概念	1
1.2 遗传算法的特点	3
1.2.1 遗传算法的优点	3
1.2.2 遗传算法的不足之处	4
1.3 遗传算法与传统方法的比较	4
1.4 遗传算法的基本用语	6
1.5 遗传算法的研究方向	7
1.6 基于遗传算法的应用	8
第二章 基本遗传算法及改进	11
2.1 遗传算法的运行过程	11
2.1.1 完整的遗传算法运算流程	11
2.1.2 遗传算法的基本操作	13
2.2 基本遗传算法	14
2.2.1 基本遗传算法的数学模型	14
2.2.2 基本遗传算法的步骤	14
2.2.3 遗传算法的具体例证	16
2.3 改进的遗传算法	22
2.3.1 改进的遗传算法一	23
2.3.2 改进的遗传算法二	24
2.3.3 改进的遗传算法三	25
2.3.4 改进的遗传算法四	28
2.4 多目标优化中的遗传算法	30
2.4.1 多目标优化的概念	30
2.4.2 多目标优化问题的遗传算法	31
第三章 遗传算法的理论基础	34
3.1 模式定理	34
3.2 积木块假设	36
3.3 欺骗问题	37
3.4 遗传算法的未成熟收敛问题及其防止	39
3.4.1 遗传算法的未成熟收敛问题	39
3.4.2 未成熟收敛的防止	40
3.5 性能评估	41
3.6 小生境技术和共享函数	43

第四章 遗传算法的基本原理与方法	45
4.1 编码	45
4.1.1 编码方法	46
4.1.2 编码评估策略	48
4.2 选择	48
4.3 交叉	52
4.4 变异	55
4.5 适应度函数	57
4.5.1 适应度函数的作用	57
4.5.2 适应度函数的设计主要满足的条件	58
4.5.3 适应度函数的种类	58
4.5.4 适应度尺度的变换	59
4.6 控制参数选择	60
4.7 约束条件的处理	61
第五章 遗传算法工具箱函数	62
5.1 工具箱结构	62
5.1.1 种群表示和初始化	63
5.1.2 适应度计算	63
5.1.3 选择函数	63
5.1.4 交叉算子	64
5.1.5 变异算子	64
5.1.6 多子群支持	64
5.2 遗传算法中的通用函数	64
5.2.1 函数 bs2rv	64
5.2.2 函数 crtbase	66
5.2.3 函数 crtbp	66
5.2.4 函数 crtrp	67
5.2.5 函数 migrate	68
5.2.6 函数 mut	69
5.2.7 函数 mutate	71
5.2.8 函数 mutbga	72
5.2.9 函数 ranking	74
5.2.10 函数 recdis	76
5.2.11 函数 recint	77
5.2.12 函数 reclin	78
5.2.13 函数 recmut	79
5.2.14 函数 recombin	81
5.2.15 函数 reins	81
5.2.16 函数 rep	84
5.2.17 函数 rws	84
5.2.18 函数 scaling	85
5.2.19 函数 select	86

5.2.20	函数 sus	88
5.2.21	函数 xovdp	88
5.2.22	函数 xovdprs	89
5.2.23	函数 xovmp	90
5.2.24	函数 xovsh	91
5.2.25	函数 xovshrs	92
5.2.26	函数 xovsp	93
5.2.27	函数 xovsprs	94
第六章 遗传算法工具箱的应用		95
6.1	安装	95
6.2	种群的表示和初始化	95
6.3	目标函数和适应度函数	96
6.4	选择	97
6.5	交叉	99
6.6	变异	101
6.7	重插入	101
6.8	遗传算法的终止	102
6.9	数据结构	102
6.10	多种群支持	104
6.11	示范脚本	105
第七章 遗传算法应用举例		107
7.1	简单一元函数优化实例	107
7.2	多元单峰函数的优化实例	111
7.3	多元多峰函数的优化实例	115
7.4	收获系统最优控制	118
7.5	装载系统的最优问题	122
7.6	离散二次线性系统最优控制问题	125
7.7	目标分配问题	128
7.8	双积分的优化问题	130
7.9	雷达目标识别问题	131
7.10	图像分割问题	134
7.11	一些测试函数对应的优化问题	136
7.11.1	轴并行超球体的最小值问题	136
7.11.2	旋转超球体的最小值问题	137
7.11.3	Rosenbrock's Valley 最小值问题	138
7.11.4	Rastrigin 函数的最小值问题	139
7.11.5	Schwefel 函数的最小值问题	140
7.11.6	Griewangk 函数的最小值问题	141
7.11.7	不同权的总和最小值问题	142
7.12	多目标优化问题	142

第八章 使用 MATLAB 遗传算法工具	146
8.1 遗传算法与直接搜索工具箱概述	146
8.1.1 工具箱的特点	146
8.1.2 编写待优化函数的 M 文件	148
8.2 使用遗传算法工具初步	149
8.2.1 遗传算法使用规则	149
8.2.2 遗传算法使用方式	150
8.2.3 举例:Rastrigin 函数	151
8.2.4 遗传算法的一些术语	156
8.2.5 遗传算法如何工作	157
8.3 使用遗传算法工具求解问题	160
8.3.1 使用遗传算法工具 GUI	160
8.3.2 从命令行使用遗传算法	172
8.3.3 遗传算法举例	177
8.4 遗传算法参数和函数	192
8.4.1 遗传算法参数	192
8.4.2 遗传算法函数	203
8.4.3 标准算法选项	207
第九章 使用 MATLAB 直接搜索工具	208
9.1 直接搜索工具概述	208
9.2 直接搜索算法	209
9.2.1 何谓直接搜索	209
9.2.2 执行模式搜索	209
9.2.3 寻找函数最小值	210
9.2.4 模式搜索术语	212
9.2.5 模式搜索如何工作	214
9.3 使用直接搜索工具	218
9.3.1 浏览模式搜索工具	218
9.3.2 从命令行运行模式搜索	226
9.3.3 模式搜索举例	229
9.3.4 参数化函数	243
9.4 模式搜索参数和函数	245
9.4.1 模式搜索参数	245
9.4.2 模式搜索函数	253
参考文献	259

第一章 遗传算法概述

遗传算法(Genetic Algorithm, 简称 GA)起源于对生物系统所进行的计算机模拟研究。美国 Michigan 大学的 Holland 教授及其学生受到生物模拟技术的启发,创造出了一种基于生物遗传和进化机制的适合于复杂系统优化的自适应概率优化技术——遗传算法。1967 年, Holland 的学生 Bagley 在其博士论文中首次提出了“遗传算法”一词,他发展了复制、交叉、变异、显性、倒位等遗传算子,在个体编码上使用双倍体的编码方法。Holland 教授用遗传算法的思想对自然和人工自适应系统进行了研究,提出了遗传算法的基本定理——模式定理(Schema Theorem),并于 1975 年出版了第一本系统论述遗传算法和人工自适应系统的专著《Adaptation in Natural and Artificial Systems》。20 世纪 80 年代, Holland 教授实现了第一个基于遗传算法的机器学习系统,开创了遗传算法的机器学习的新概念。1975 年, De Jong 基于遗传算法的思想在计算机上进行了大量的纯数值函数优化计算实验,建立了遗传算法的工作框架,得到了一些重要且具有指导意义的结论。1989 年, Goldberg 出版了《Genetic Algorithm in Search, Optimization and Machine Learning》一书,系统地总结了遗传算法的主要研究成果,全面完整地论述了遗传算法的基本原理及其应用。1991 年, Davis 出版了《Handbook of Genetic Algorithms》一书,介绍了遗传算法在科学计算、工程技术和社会经济中的大量实例。1992 年, Koza 将遗传算法应用于计算机程序的优化设计及自动生成,提出了遗传编程(Genetic Programming, 简称 GP)的概念。在控制系统的离线设计方面遗传算法被众多的使用者证明是有效的策略。例如, Krishnakumar 和 Goldberg 以及 Bramlette 和 Cusin 已证明使用遗传优化方法在太空应用中导出优异的控制器结构比使用传统方法如 LQR 和 Powell(鲍威尔)的增音机设计所用的时间要少(功能评估)。Porter 和 Mohamed 展示了使用本质结构分派任务的多变量飞行控制系统的遗传设计方案。与此同时,另一些人证明了遗传算法如何在控制器结构的选择中使用。

从遗传算法的整个发展过程来看,20 世纪 70 年代是兴起阶段,20 世纪 80 年代是发展阶段,20 世纪 90 年代是高潮阶段。遗传算法作为一种实用、高效、鲁棒性强的优化技术,发展极为迅速,已引起国内外学者的高度重视。

1.1 遗传算法的概念

生物的进化(Evolution)过程主要是通过染色体之间的交叉和变异来完成的。基于对自然界中生物遗传与进化机理的模仿,针对不同的问题,很多学者设计了许多不同的编码方

法来表示问题的可行解,开发出了许多种不同的编码方式来模仿不同环境下的生物遗传特性。这样,由不同的编码(Coding)方法和不同的遗传算子就构成了各种不同的遗传算法。

遗传算法是模仿自然界生物进化机制发展起来的随机全局搜索和优化方法,它借鉴了达尔文的进化论和孟德尔的遗传学说。其本质是一种高效、并行、全局搜索的方法,它能在搜索过程中自动获取和积累有关搜索空间的知识,并自适应地控制搜索过程以求得最优解。遗传算法操作使用适者生存的原则,在潜在的解决方案种群中逐次产生一个近似最优的方案。在遗传算法的每一代中,根据个体在问题域中的适应度值和从自然遗传学中借鉴来的再造方法进行个体选择,产生一个新的近似解。这个过程导致种群中个体的进化,得到的新个体比原个体更能适应环境,就像自然界中的改造一样。

个体或当前近似解被编码为由字母组成的串,即染色体(Chromosome),使基因(Gene,染色体值)能在(表现)域决策变量上被惟一地描述。尽管可以使用二进制、整数、实值等,但是在遗传算法表现型上最常用的仍是二进制字符串。例如,一个问题具有两个变量 X_1 和 X_2 ,它们的染色体结构能用图 1.1 所示的方法描述。

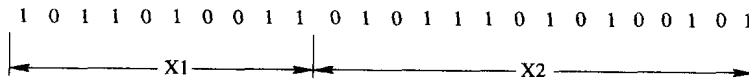


图 1.1 个体的染色体结构

X_1 被编码为 10 位, X_2 被编码为 15 位(位数的多少能够反映精确度水平或个体决策变量的范围),是一个不含人们试图解决问题的信息的染色体串。这只是表现值的染色体编码,任何意义均可应用于表现型。无论如何,就像下面的描述,搜索过程将在决策变量的编码中而不是它们自身中操作,当然除了在实值基因被使用的情况。

在决策变量域中的染色体表现型已被编码,可以估计种群的个体成员的特性或适应度。通过特征目标函数来估计个体在问题域中的特性。在自然世界中,这就是个体在现行环境中的生存能力。因此,目标函数建立的基础是在整个繁殖过程中选择成对的个体进行交配。

在再生(复制)期间,每个个体均被计算适应度值,它来自没有加工的原始特性度量,由目标函数给出。这个值用来在选择中偏向更加适合的个体。相对整个种群,适应度高的个体具有高的选中参加交配的概率,而适应度低的个体具有相对低的选中概率。

一旦个体计算了适应度值,个体能根据它们的相对适应度从种群中被选中并重组,产生下一代。遗传算子直接操作染色体的特征(基因),使用一般情况下个体的基因代码,产生更适合的个体。重组算子用在一对个体或一大组个体中交换基因信息。最简单的重组算子是单点交叉。

考虑两个二进制父代串:

$$A=10010110 \text{ 和 } B=10111000$$

如果一个整 I 是随机地在 1 到串长 L 减 1 之间(即 $[1, L-1]$)选择的,在这点后,两个个体间的基因进行交换,随后两个子代串产生。例如,当交叉点 $I=5$ 时,两个子代串产生如下:

$$A'=10010000 \text{ 和 } B'=10111110$$

交叉算子并不是必须在种群的所有串中执行的。当一对个体被选中培育下一代时,代

替的是应用一个概率 P_x 。进一步的遗传算法称为变异，再次使用一个概率 P_m 应用到新染色体上。变异能根据一些概率准则引起个体基因表现型发生变化，在二进制表现型中，变异引起单个位的状态变化，即 0 变 1，或者 1 变 0。

例如，在 A' 中变异第四位，1 变 0，产生新串为 10000000。

变异通常被认为是一后台算子，以确保研究问题空间的特殊子空间的概率永不 0。变异具有阻止局部最优收敛的作用。

在重组和变异后，如果需要，这些个体串随后被解码，进行目标函数评估，计算每个个体的适应度值，个体根据适应度被选择参加交配，并且这个过程继续直到产生子代。在这种方法中，种群中个体的平均性能希望得到提高，好的个体被保存并且相互产生下一代，而低适应度的个体则消失。当一些判定条件满足后，遗传算法则终止，例如，一定的遗传代数、种群的均差或遇到搜索空间的特殊点。

1.2 遗传算法的特点

遗传算法是一种借鉴生物界自然选择(Natural Selection)和自然遗传机制的随机搜索算法(Random Searching Algorithms)。它与传统的算法不同，大多数古典的优化算法是基于一个单一的度量函数(评估函数)的梯度或较高次统计，以产生一个确定性的试验解序列；遗传算法不依赖于梯度信息，而是通过模拟自然进化过程来搜索最优解(Optimal Solution)，它利用某种编码技术，作用于称为染色体的数字串，模拟由这些串组成的群体的进化过程。遗传算法通过有组织的、随机的信息交换来重新组合那些适应性好的串，生成新的串的群体。

1.2.1 遗传算法的优点

遗传算法具有如下优点：

(1) 对可行解表示的广泛性。遗传算法的处理对象不是参数本身，而是针对那些通过参数集进行编码得到的基因个体。此编码操作使得遗传算法可以直接对结构对象进行操作。所谓结构对象，泛指集合、序列、矩阵、树、图、链和表等各种一维或二维甚至多维结构形式的对象。这一特点使得遗传算法具有广泛的应用领域。比如：

① 通过对连接矩阵的操作，遗传算法可用来对神经网络或自动机的结构或参数加以优化。

② 通过对集合的操作，遗传算法可实现对规则集合和知识库的提炼而达到高质量的机器学习目的。

③ 通过对树结构的操作，用遗传算法可得到用于分类的最佳决策树。

④ 通过对任务序列的操作，遗传算法可用于任务规划，而通过对操作序列的处理，可自动构造顺序控制系统。

(2) 群体搜索特性。许多传统的搜索方法都是单点搜索，这种点对点的搜索方法，对于多峰分布的搜索空间常常会陷于局部的某个单峰的极值点。相反，遗传算法采用的是同时处理群体中多个个体的方法，即同时对搜索空间中的多个解进行评估。这一特点使遗传算法具有较好的全局搜索性能，也使得遗传算法本身易于并行化。

(3) 不需要辅助信息。遗传算法仅用适应度函数的数值来评估基因个体，并在此基础上进行遗传操作。更重要的是，遗传算法的适应度函数不仅不受连续可微的约束，而且其定义域可以任意设定。对适应度函数的惟一要求是，编码必须与可行解空间对应，不能有死码。由于限制条件的缩小，使得遗传算法的应用范围大大扩展。

(4) 内在启发式随机搜索特性。遗传算法不是采用确定性规则，而是采用概率的变迁规则来指导它的搜索方向。概率仅仅是作为一种工具来引导其搜索过程朝着搜索空间的更优化的解区域移动的。虽然看起来它是一种盲目搜索方法，实际上它有明确的搜索方向，具有内在的并行搜索机制。

(5) 遗传算法在搜索过程中不容易陷入局部最优，即使在所定义的适应度函数是不连续的、非规则的或有噪声的情况下，也能以很大的概率找到全局最优解。

(6) 遗传算法采用自然进化机制来表现复杂的现象，能够快速可靠地解决求解非常困难的问题。

(7) 遗传算法具有固有的并行性和并行计算的能力。

(8) 遗传算法具有可扩展性，易于同别的技术混合。

应重点注意的是，遗传算法对给定问题给出了大量可能的解答，并挑选最终的解答给用户。要是是一个特定问题并没有单个的解，例如 Pareto 最优解系列中，就像多目标优化和日程安排的案例中，遗传算法将尽可能地用于识别可同时替换的解。

1.2.2 遗传算法的不足之处

遗传算法作为一种优化方法，它存在自身的局限性：

(1) 编码不规范及编码存在表示的不准确性。

(2) 单一的遗传算法编码不能全面地将优化问题的约束表示出来。考虑约束的一个方法就是对不可行解采用阈值，这样，计算的时间必然增加。

(3) 遗传算法通常的效率比其他传统的优化方法低。

(4) 遗传算法容易出现过早收敛。

(5) 遗传算法对算法的精度、可信度、计算复杂性等方面，还没有有效的定量分析方法。

1.3 遗传算法与传统方法的比较

对于一个求函数最大值的优化问题(求函数最小值也类似)，一般可描述为带约束条件的数学规划模型：

$$\begin{cases} \max & f(X) \\ \text{s. t.} & X \in R \\ & R \subseteq U \end{cases} \quad (1.1)$$

式中， $X = [x_1, x_2, \dots, x_n]^T$ 为决策变量， $f(X)$ 为目标函数， U 为基本空间， R 是 U 的一个子集。满足约束条件的解称为可行解(Feasible Solution)，集合 R 表示由所有满足约束条件的解所组成的一个集合，称为可行解集合。它们之间的关系如图 1.2 所示。

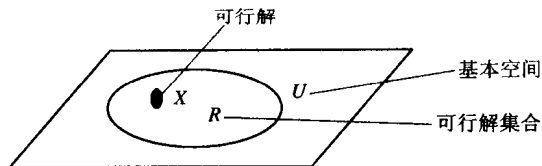


图 1.2 最优化问题的可行解及可行解集合

对于上述最优化问题，目标函数和约束条件种类繁多，有的是线性的，有的是非线性的；有的是连续的，有的是离散的；有的是单峰值的，有的是多峰值的。随着研究的深入，人们逐渐认识到在很多复杂情况下要想完全精确地求出其最优解是不可能的，也是不现实的，因而求出其近似最优解或满意解是人们主要研究的问题之一。

对于类似上述最优化问题，求最优解或近似最优解的传统方法主要有解析法、随机法和穷举法。解析法主要包括爬山法和间接法。随机法主要包括导向随机方法和盲目随机方法。而穷举法主要包括完全穷举法、回溯法、动态规划法和限界剪枝法。

此类问题可以利用遗传算法求解。而对于求解此类问题，遗传算法与一般传统方法有着本质的区别。图 1.3 所示为传统算法和遗传算法对比示意图。

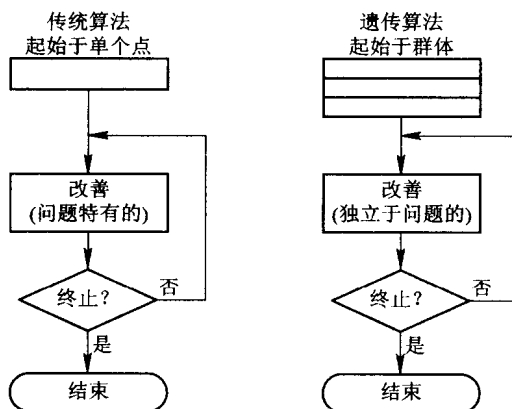


图 1.3 传统算法和遗传算法对比

1. 遗传算法与启发式算法的比较

启发式算法是指通过寻求一种能产生可行解的启发式规则，找到问题的一个最优解或近似最优解。该方法求解问题的效率较高，但是它对每一个所求的问题必须找出其特有的启发式规则，这个启发式规则一般无通用性，不适用于其他问题。但遗传算法采用的不是确定性规则，而是强调利用概率转换规则来引导搜索过程。

2. 遗传算法与爬山法的比较

爬山法是直接法、梯度法和 Hessian 法的通称。爬山法首先在最优解可能存在的地方选择一个初始点，然后通过分析目标函数的特性，由初始点移到一个新的点，然后再继续这个过程。爬山法的搜索过程是确定的，它通过产生一系列的点收敛到最优解（有时是局部最优解），而遗传算法的搜索过程是随机的，它产生一系列随机种群序列。二者的主要差异可以归纳为如下两点：

(1) 爬山法的初始点仅有一个，由决策者给出；遗传算法的初始点有多个，是随机产

生的。

(2) 通过分析目标函数的特性可知,爬山法由上一点产生一个新的点;遗传算法通过遗传操作,在当前的种群中经过交叉、变异和选择产生下一代种群。对同一优化问题,遗传算法所使用的机时比爬山法所花费的机时要多。但遗传算法可以处理一些爬山法所不能解决的复杂的优化问题。

3. 遗传算法与穷举法的比较

穷举法就是对解空间内的所有可行解进行搜索,但是通常的穷举法并不是完全穷举法,即不是对所有解进行尝试,而是有选择地尝试,如动态规划法、限界剪枝法。对于特定的问题,穷举法有时也表现出很好的性能。但一般情况下,对于完全穷举法,方法简单易行,但求解效率太低;对于动态规划法、限界剪枝法,则鲁棒性不强。相比较而言,遗传算法具有较高的搜索能力和极强的鲁棒性。

4. 遗传算法与盲目随机法的比较

与上述的搜索方法相比,盲目随机搜索法有所改进,但它的搜索效率仍然不高。一般而言,只有解在搜索空间中形成紧致分布时,它的搜索才有效。而遗传算法作为导向随机搜索方法,是对一个被编码的参数空间进行高效搜索。

经过上面的探讨,能看到遗传算法与更多的传统优化方法在本质上有着不同之处,主要有四点不同:

(1) 遗传算法搜索种群中的点是并行的,而不是单点。

(2) 遗传算法并不需要辅助信息或辅助知识,只需要影响搜索方向的目标函数和相对应的适应度。

(3) 遗传算法使用概率变换规则,而不是确定的变换规则。

(4) 遗传算法工作使用编码参数集,而不是自身的参数集(除了在实值个体中使用)。

1.4 遗传算法的基本用语

由于遗传算法是自然遗传学和计算机科学相互结合渗透而成的新的计算方法,因此遗传算法中经常使用自然进化中有关的一些基本用语。了解这些用语对于讨论和应用遗传算法是十分必要的。

生物的遗传物质的主要载体是染色体,DNA 是最主要的遗传物质,而基因(Gene)又是控制生物性状的遗传物质的功能单位和结合单位。复数个基因组成染色体,染色体中基因的位置称为基因座(Locus),而基因所取的值叫做等位基因(Allele)。基因和基因座决定了染色体的特征,也就决定了生物个体的性质状态。染色体有两种相应的表示模式,即基因型和表现型。所谓表现型,是指生物个体所表现出来的性质状态,而基因型是指与表现型密切相关的基因组成。同一种基因型的生物个体在不同的环境条件下可以有不同的表现型,因此表现型是基因型与环境条件相互作用的结果。在遗传算法中,染色体对应的是数据或数组,在标准的遗传算法中,通常是由一维的串结构数据表现的。串上各个位置对应上述的基因座,而各位置上所取的值对应上述的等位基因。遗传算法处理的是染色体,或者叫基因型个体。一定数量的个体组成了群体,也叫集团。群体中个体的数目称为群体的大小,也叫群体规模。而各个体对环境的适应程度叫适应度。执行遗传算法时包

含两个必要的数据转换操作，一个是表现型到基因型的转换，它把搜索空间中的参数或解转换成遗传空间中的染色体或个体，此过程称为编码操作；另一个是基因型到表现型的转换，它是前者的一个相反操作，称为译码操作。表 1.1 为自然遗传学和人工遗传算法中所使用的基本用语的对应关系。

表 1.1 遗传学和遗传算法中基本用语对照表

自然遗传算法	人工遗传算法
染色体(Chromosome)	解的编码(数据、数组、位串)
基因(Gene)	解中每一分量的特征(特性、个性、探测器、位)
等位基因(Allele)	特性值
基因座(Locus)	串中位置
基因型(Genotype)	结构
表现型(Phenotype)	参数集、解码结构、候选解
遗传隐匿	非线性
个体(Individual)	解
适者生存	在算法停止时，最优目标值的解有最大的可能被留住
适应性(Fitness)	适应度函数值
群体(Population)	选定的一组解(其中解的个数为群体的规模)
复制(Reproduction)	根据适应函数值选取的一组解
交配(Crossover)	通过交配原则产生一组新解的过程
变异(Mutation)	编码的某一个分量发生变化的过程

1.5 遗传算法的研究方向

遗传算法是多学科结合与渗透的产物，它已经发展成一种自组织、自适应的综合技术，其研究方向主要有下述几个方面。

1. 基础理论

遗传算法的数学理论并不完善，张铃等对遗传算法的“模式定理”和“隐性并行性”进行了分析研究，指出其不足并指出遗传算法本质上是一个具有定向制导的随机搜索技术。在遗传算法中，群体规模和遗传算子的控制参数的选取非常困难，但它们又是必不可少的实验参数，在这方面，已有一些具有指导性的实验结果。遗传算法还有一个过早收敛的问题，如何阻止过早收敛也是人们正在研究的问题之一。

2. 分布并行遗传算法

遗传算法在操作上具有高度的并行性,许多研究人员都在探索在并行机和分布式系统上高效执行遗传算法的策略。对分布遗传算法的研究表明,只要通过保持多个群体和恰当控制群体间的相互作用来模拟并发执行过程,即使不使用并行计算机,也能提高算法的执行效率。遗传算法的并行性主要从三个方面考虑,即个体适应度评价的并行性、整个群体各个个体适应度评价的并行性及子代群体产生过程的并行性。

3. 分类系统

分类系统属于基于遗传算法的机器学习中的一类,包括一个简单的基于串规则的并行生成子系统、规则评价子系统和遗传算法子系统。分类系统被人们越来越多地应用在科学、工程和经济领域中,是目前遗传算法研究中一个十分活跃的方向。

4. 遗传神经网络

遗传神经网络包括连接级、网络结构和学习规则的进化。遗传算法与神经网络相结合,成功地用于从分析时间序列来进行财政预算。在这些系统中,训练信号是模糊的,数据是有噪声的,一般很难正确给出每个执行的定量评价,如果采用遗传算法学习,就能克服这些困难,显著提高系统性能。Muhlenbein 分析了多层感知网络的局限性,并猜想下一代神经网络将是遗传神经网络。

5. 进化算法

模拟自然进化过程可以产生鲁棒的计算机算法——进化算法。遗传算法是其三种典型的算法之一,其余两种算法是进化规划和进化策略,这三种算法是独立发展起来的。

6. 人工生命与遗传算法

近几年来,通过计算机模拟再现种种生命现象,以达到对生命更深刻理解的人工生命的研究正在兴起。已有不少学者对生态系统的演变、食物链的维持以及免疫系统的进化等用遗传算法做了生动的模拟。但是实现人工生命的手段很多,遗传算法在实现人工生命中的基本地位和能力究竟如何,这是值得研究的课题。

1.6 基于遗传算法的应用

遗传算法提供了一种求解复杂系统优化问题的通用框架,它不依赖于问题具体的领域,对问题的种类有很强的鲁棒性,所以广泛应用于许多学科。近十年来,遗传算法得到了迅速发展。目前,遗传算法在生物技术和生物学、化学和化学工程、计算机辅助设计、物理学和数据分析、动态处理、建模与模拟、医学与医学工程、微电子学、模式识别、人工智能、生产调度、机器人学、开矿工程、电信学、售货服务系统等领域都得到应用,成为求解全局优化问题的有力工具之一。下面列出遗传算法一些主要的应用领域。

1. 函数优化

函数优化(Function Optimization)是遗传算法的经典应用领域,也是对遗传算法进行性能评价的常用算例。可以用各种各样的函数来验证遗传算法的性能。对一些非线性、多模型、多目标的函数优化问题,使用遗传算法可得到较好的结果。

2. 组合优化

随着问题规模的增大,组合优化问题的搜索空间也急剧扩大,有时在目前的计算机上