进化计算的理论和方法

王宇平 著

科学出版社

进化计算的理论和方法

王宇平 著

本书获得国家自然科学基金(60873099)资助

科学出版社
北京

内 容 简 介

进化算法是一类直接的、随机搜索的优化算法，它是基于进化论的思想而产生的一类新型优化方法。本书在介绍进化算法基本原理、方法和理论的基础上，也介绍了一些新的进化算法。本书共分 9 章。第 1 章介绍了进化算法的产生背景、主要特点、发展趋势及其 4 个主要分支；第 2 章介绍了进化算法的基本原理、模式定理、积木块假设和编码规则等；第 3 章介绍了经典遗传算法的收敛性分析；第 4 章介绍了求解无约束全局优化问题的传统遗传算法及三种新的进化算法：基于平滑技术的进化算法，正交遗传算法，以及基于水平集进化和拉丁方的进化算法；第 5 章介绍了求解约束全局优化问题的传统遗传算法、一个基于新的罚函数模型的进化算法，以及解无约束和约束全局优化问题的进化策略；第 6 章介绍了求解组合优化问题的进化算法；第 7 章介绍了多目标优化问题的基本概念、求解多目标优化问题的传统进化算法，以及新的进化算法、算法性能的度量，并且介绍了一个求解动态多目标优化问题新的进化算法；第 8 章介绍了一种求解非线性双层规划问题新的进化算法；第 9 章介绍了进化算法的收敛性理论，对文献中出现的一些不同形式的收敛性结果进行了归纳和总结。

本书可作为工程类各专业、运筹学专业和管理学科各专业研究生教材，也可供相关科研人员和工程技术人员参考。

图书在版编目(CIP)数据

进化计算的理论和方法/王宇平著. —北京：科学出版社，2011. 3
ISBN 978-7-03-030483-4

Ⅰ. ①进… Ⅱ. ①王… Ⅲ. ①数值计算-算法分析 Ⅳ. ①TP301. 6

中国版本图书馆 CIP 数据核字(2011)第 038234 号

责任编辑：杨向萍 孙 芳 / 责任校对：包志虹
责任印制：赵 博 / 封面设计：耕者设计工作室

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

丽源印刷厂 印刷
科学出版社发行 各地新华书店经销
*
2011 年 3 月第 一 版 开本：B5(720×1000)
2011 年 3 月第一次印刷 印张：15 1/4
印数：1—2 500 字数：294 000

定价：50. 00 元

(如有印装质量问题，我社负责调换)

前　言

随着计算机科学的迅猛发展，利用计算机通过新的计算模型和方法解决过去一些无法解决的难题已经成为可能，而且新的计算模型和方法还在不断涌现。进化计算就是在这种背景下产生的一种模拟生物进化过程的新型计算模型和方法。

早在20世纪40年代，就有学者开始研究如何利用计算机模拟生物进化过程，到60年代，美国密执安大学的Holland教授及其学生们发展了一种模拟生物遗传和进化机制的随机优化技术——遗传算法(genetic algorithms)。1967年，Holland的学生Bagley在其博士论文中首次提出"遗传算法"的概念。此后，Holland指导学生完成了多篇有关遗传算法研究的论文。1975年，Holland出版了学术著作*Adaptation in Natural and Artificial Systems*，这是第一本系统论述遗传算法的著作，标志着遗传算法的诞生。同年，De Jong完成了他的博士论文《一类遗传自适应系统的行为分析》，该论文将选择、交叉和变异操作进一步完善和系统化，为遗传算法及其应用打下了坚实的基础。与此同时，20世纪60～70年代，Rechenberg和Schwefel教授独立发展了进化计算的另一个分支——进化策略(evolution strategies，ES)；Fogel教授等提出进化计算的第三个分支——进化规划(evolutionary programming，EP)；进入20世纪90年代，Koza教授等提出了进化计算的第四个分支——遗传程序设计(genetic programming，GP)。此后，进化算法的研究引起了国内外学者的广泛关注和重视。

自1985年以来，已有多个国际组织和学术团体组织了许多高水平的进化计算国际学术会议，如IEEE Congress on Evolutionary Computation、International Conference on Genetic Algorithms、Parallel Problem Solving from Nature等，同时，很多国际杂志也刊登了进化计算方面的学术论文，如*IEEE Transactions on Evolutionary Computation*、*Evolutionary Computation*、*Artificial Intelligence*、*Machine Learning*、*Information Science*、*Parallel Computing*、*Genetic Programming and Evolvable Machine*、*IEEE Transactions on SMC*、*IEEE Transactions on Neural Networks*等，这些会议和杂志为研究和应用遗传算法的学者提供了国际交流的机会和平台，同时也标志着进化计算进入了一个快速发展的时期。

近年来，进化计算已被成功应用于工业、经济管理、交通运输、工业设计等不同领域，解决了许多有价值的实际问题，如可靠性优化、流水车间调度、作业车间调度、机器调度、设备布局设计、图像处理及数据挖掘等，但新的应用问题不断出现，已有的算法和理论还不能完全满足解决这些应用问题的需要，有必要对进化计算

进行深入的研究。

本书试图从进化计算的方法和理论两个方面系统介绍进化算法的主要特点、基本原理、已有的典型算法和理论，同时重点介绍一些新的进化算法及其相关理论，希望对相关领域的研究生和科技工作者有所帮助。

由于作者水平有限，不妥之处在所难免，敬请专家和读者批评指正，作者将不胜感激。

王宇平

2011 年 1 月

目　录

第1章　绪　论

1.1　进化算法产生的背景

进化算法是求解全局最优化问题的一种新型算法。考虑如下最优化问题：

$$\min_{x\in D} f(\boldsymbol{x})$$

式中，$D\subseteq \mathbf{R}^n$。传统求解方法的迭代步骤一般可以写为如下迭代格式：

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k,\ k=1,2,\cdots$$

式中，$\boldsymbol{d}_k$ 为 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}_k$ 处的一个下降方向；α_k 为搜索步长。一般地，求 $\boldsymbol{d}_k$ 通常要用 $f(x)$ 的一、二阶导数信息，α_k 通常由精确或不精确线性搜索确定，如最速下降法、拟牛顿法、共轭梯度法、牛顿法等[1,2]。由于传统优化方法通常要用相关函数的导数信息，而这些导数信息是由极限确定的，只能反映相关函数的局部特性，不可能反映距离当前解较远处函数的特性，因此，利用导数构造出的上述格式的算法通常难以求出函数的全局最优解，往往只能求出局部最小值点（局部最优点）。而在大量的实际问题中，通常需要求出全局最优点，而且函数的一阶导数、二阶导数等信息不容易求得或无法求出，这时，传统优化方法要么难以求出全局最优解，要么无法使用。如何设计一种有效的算法来解决这些实际问题是摆在我们面前迫切需要解决的问题，也是对传统优化方法的一个很好的补充。

进化算法正是在这种背景下诞生的一种新型优化算法，它是模仿生物进化与遗传原理而设计的一类随机搜索的优化算法。根据达尔文的进化论，自然界的进化（演化）是一个优胜劣汰、适者生存的过程[3]。这个过程对一种生物或者物种来说，如对整个人类来说，都是在不断进步，聪明程度和健康程度都在不断提高，即都是在不断改变自身的特点以适应在自然界中生存。因此，整个群体[或称为种群（population），在后续叙述中，群体和种群表示同一个概念]的进化过程可以看成是一个不断优化的过程。如何将这个过程抽象出来，建立起一个优化模型和设计一种优化方法，并利用计算机来模拟这个过程，就是进化算法需要解决的问题。

整个群体的进化过程可以看成是一个优化过程，但单个个体（individual）的进化轨迹未必是一个优化过程。举一个直观的例子，如对整个人类而言，随着时间的推移，会越来越聪明和健康，但对单个个人来讲，其后代（offspring）未必比其祖先更聪明和健康。因此，模拟生物进化的进化算法的迭代（进化）不应该是从一个点移动到另一个点，而应是从一群点通过进化算子[杂交或交叉（crossover）、变异

(mutation)和选择(selection)]等的作用移动到另一群点,形象地讲,就是群体通过相互的作用(交叉、变异和选择等进化算子作用)进行不断的改进和提高。用数学模型的观点看,它就是并行地搜索多个山谷,因而往往可求出全局最优点,它不需目标函数的连续性、可微性等条件,只利用适应度(fitness)函数的信息(它可由目标函数确定),因而可用于不可微、不连续的复杂优化问题(包含许多局部最优解的优化问题)。

进化算法的发展非常迅速,其传统的4个分支为:①遗传算法,1975年由Holland提出[4];②进化策略,分别由Rechenberg和Schwefel于1971年和1975年独立提出[5,6];③进化规划,1966年由Fogel等提出[7];④遗传程序设计,1990年由Koza等提出[8,9]。这4个分支有很多相似之处,它们的求解步骤大致可用下面的算法框架来统一描述:

(1) 产生初始终群体 $p(0)$,令 $t=0$,$p(0)=\{x_1^0,\cdots,x_N^0\}$。

(2) 计算 $p(t)$中每个个体的适应度来评价当前群体的优劣,用进化算子产生新一代群体 $p(t+1)$,令 $t=t+1$。

(3) 若终止条件成立,停止;否则,转(2)。

尽管这4个分支有很多相似之处,但它们是相互独立地发展起来的,它们之间也有本质的区别,下面分别简要介绍这些方法。

1.2 进化算法传统的4个分支

由于各分支内容各有侧重,下面介绍其各自特点及相互区别。

1.2.1 遗传算法

1. 算法步骤

(1) 按某种方式产生初始群体 $p(0)=\{x_1^0,\cdots,x_N^0\}$,令 $t=0$,对 $p(t)$ 中每一个个体 x_i^0 进行编码(encoding),得到的相应的点叫染色体(chromosome)或字符串(string)b_i^0,$x_i^0=01\cdots11\triangleq b_i^0$(一般采用二进制编码(binary encoding),即 x_i^0 将表示成二进制数),计算 $p(t)$ 中每个个体 x_i^0 的适应度。

(2) 按某种规则从 $p(t)$中选择一个子群体 $p'(t)\subset p(t)$ 作为产生后代的父母,用交叉算子作用于 $p(t)$产生一些后代,再用变异算子作用于每个后代产生新的后代集合 O,计算 O 中每个后代的适应度。

(3) 用选择算子在 $p(t)\cup O$中选出下一代(next generation)的群体 $p(t+1)$,令 $t=t+1$。

(4) 若终止条件成立,则停止;否则,转(2)。

2. 遗传算法的一些常用术语

(1) 染色体＝字符串＝编码前的个体。

(2) 基因(gene)＝位(bit)＝染色体中每一位(例如,X＝101001 中每一位的 0 或 1)。

(3) 交叉＝重组(recombination)＝繁殖(reproduction)。

(4) 变异。

(5) 群体。

(6) 群体规模(population-size 或 pop-size)。

(7) 适应度。

(8) 编码。

(9) 译码(decoding)。

(10) 交叉概率(p_c)。

(11) 变异概率(p_m)。

3. 遗传算法的五要素

(1) 对解进行编码表示(通常用二进制编码)。

(2) 确定一种方式产生初始群体(initial population)。

(3) 设计适应度函数。

(4) 设计遗传算子(genetic operators)(交叉、变异、选择算子)。

(5) 参数设定(群体规模、p_c、p_m 等)。

1.2.2 进化策略

1. (μ+1)进化策略

(1) 产生 μ 个个体组成初始群体。

(2) 随机地选取一个个体用正态分布进行变异得一个后代。

(3) 用后代取代群体中最差个体。

2. (μ+λ)进化策略

(1) 产生 μ 个个体组成初始群体。

(2) 用交叉方法使两个个体产生一个后代,用变异由一个个体产生一个后代,共产生 λ 个后代。

(3) 从(μ+λ)个个体中选出 μ 个最优者作为下一代群体。

3. (μ,λ)进化策略

(1) 产生 μ 个个体组成初始群体。

(2) 用交叉方法使两个个体产生一个后代,用变异由一个个体产生一个后代,共产生 λ 个后代。

(3) 在新产生的 λ 个后代中($\lambda>\mu$)选取 μ 个最优者作为下一代群体。

4. 进化策略与遗传算法的区别

(1) 遗传算法要对解进行编码,交叉、变异等遗传算子要在编码后的染色体上进行;而进化策略直接在问题的解空间上进行,不需编码。

(2) 遗传算法以交叉为主要进化算子,而进化策略以变异算子作为主要进化算子。

(3) 进化策略主要用于求解数值优化问题。近来,遗传算法也用十进制编码(浮点数编码)技术来求数值优化问题,两者互相渗透已使得它们没有明显的界线。

1.2.3 进化规划

1. 算法步骤

(1) 随机选取初始群体(群体规模依赖于问题的难度)。

(2) 对每个个体按在其上定义的一个函数的值来确定一个分布,再按此分布对该个体进行变异得新群体。

(3) 计算新群体中个体的适应度,用随机竞赛选择方法选出下一代群体,具体做法为:在新群体中,每次随机选 k 个个体($k<$群体规模),将其中适应度最大(最好)的个体放入下一代群体中,重复这一过程直到新群体中包含预定数目的个体为止。

2. 进化规划与遗传算法的区别

(1) 进化规划对解的表示没有任何限制;遗传算法需对解编码。

(2) 进化规划不用交叉,只用变异,且当解越来越逼近最优解时,变异程度越小;而遗传算法以交叉作为主要算子。

3. 进化规划与进化策略的区别

(1) 进化规划通过竞赛使用随机选择(stochastic selection via tournament),而进化策略用确定性选择来抛弃最差的解。

(2) 进化规划不用交叉算子产生后代,而进化策略通常使用。

1.2.4 遗传程序设计

对于许多问题来讲，作为问题解的最自然的表示形式是计算机程序，也就是说，这些问题可看成是要求找到对特定输入产生特定输出的计算机程序，从而求解这些问题的过程可以重新表示为在可能的计算机程序空间中寻找适应性最好的计算机程序，其中，适应性是度量解决一个特定问题的计算机程序的好坏程度的。

从这个角度看，求解问题的过程等价于在可能的计算机程序空间中进行搜索以找到适应性最好的计算机程序。此时，搜索空间就是所有可能的计算机程序。

遗传程序设计就是利用遗传算法的思想，通过对计算机程序进行进化(交叉、变异和选择等)而找出能解决给定问题的最好的计算机程序。

原则上讲，作为遗传程序设计作用的对象——计算机程序，可以是任意的计算机语言，然而，大多数计算机语言的语法经过遗传程序设计算子的作用后将产生大量语法错误的程序。为了克服这种困难，Koza 提出用 LISP 语言。当然，需要注意的是，遗传程序设计并非必须用 LISP 语言来实现，任何一种能以解剖树形式表示的计算机程序均可。目前，大部分的遗传程序设计软件包均是用 C、C ++或 Java 语言，而不是 LISP 语言。

1.3 进化算法的主要特点及发展趋势

1. 进化算法的主要特点

(1) 智能性。智能性包括自组织、自适应和自学习性等。算法利用进化过程中获得的信息自行组织搜索，又基于自然选择策略，即适者生存，不适者淘汰，使适应值大的个体具有较多的生存机会。

(2) 本质并性行。① 内在并行性(inherent parallelism)。可在多台计算机上并行运行进化算法，运算结束时各机器交流信息，进行选择，再进行并行计算。② 隐并行(implicit parallelism)。由于是群体方式搜索，它可在搜索空间内多个区域同时搜索，并交流信息，虽然每次只执行与群体规模 N 成比例的计算，而实质上已进行了比 $O(N)$ 高阶数的有效搜索。

2. 进化算法的发展趋势

(1) 理论方面。缺乏统一、完整的理论体系，主要在收敛性和收敛速度分析方面的成果还不完善。

(2) 计算方面。如何设计有效的进化算法、如何评价算法的性能等还有待于进一步研究。

(3) 新的计算模型。目前实现的只是生物进化的一小部分模型，近来有两个新的动向：① 模拟免疫系统的模型；② DNA 模型。

(4) 进化优化。尽管已有很多方法，但实际问题复杂多变，要想用一种方法解决所有的问题是不现实的，如何根据具体问题的特点来设计有效的进化方法，以提高计算速度和有效处理约束是十分重要的；同时，对已有的各有效算法，寻找其适用的应用问题类型也是一个重要的研究课题。

3. 常用期刊介绍

IEEE Transaction on Evolutionary Computation、*Evolutionary Computation*、*IEEE Transactions on Systems*, *Man*, *and Cybernetics*，以及有关计算机学科、人工智能方面的中文期刊。

第 2 章　进化计算的基本原理

2.1　经典遗传算法简介

经典遗传算法(canonical genetic algorithms,CGA)是遗传算法的一个典型代表,它是由 Holland 于 20 世纪 70 年代提出的一个简单易行的遗传算法框架[4],其具体步骤如下:

(1) 产生初始群体,个体编码后的群体为 $P(0)=\{x^1,x^2,\cdots,x^n\}$, 令 $t=0$。

(2) 计算 $P(0)$ 中每个个体的适应度。

(3) 进行选择(选择进化的父母)。

(4) 进行交叉、变异来产生后代,计算各后代的适应度。

(5) 选择下一代群体 $P(t+1)$,令 $t=t+1$,转(3)。

1. 二进制编码

首先注意,一个串长为 m 的二进制数$\overbrace{b_1b_2\cdots b_m}^{m}$, 其中, $b_i=0$ 或 1, $i=1,\cdots,m$, 其表示的最小整数为

$$\overbrace{00\cdots0}^{m}=0$$

最大整数为

$$\overbrace{11\cdots1}^{m}=2^0+2^1+\cdots+2^{m-1}=2^m-1$$

共可表示 2^m 个整数,分别为 $0,1,\cdots,2^m-1$。

下面以变量 x 为一维变量为例说明编码过程。假设变量 x 变化范围为 $x\in[a,b]$,解的精度为 $10^{-\alpha}$。如何用串长为 m 的二进制数表示$[a,b]$ 中的解呢?方法的思想如下:将$[a,b]$ 等分为若干小区间,只用二进制数表示等分点和区间的端点。为达到精度要求,只要将$[a,b]$ 等分至$(b-a)\times10^{\alpha}$ 个小区间。若只考虑$[a,b]$ 的两端点和等分点,则任两相邻等分点间距离都小于或等于 $10^{-\alpha}$。若所考虑的函数连续,小区间的长又充分小(其长度小于或等于精度,所以,可以认为其长充分小),则与最优解最近的等分点将会比其他等分点好,将等分点(包括端点)的最好点作为解的近似,则最优解和最好点的距离不超过精度,于是此最好点满足精度要求。

这种思想可描述为:要用二进制数表示 $[a,b]$ 中所有的等分点及两端点,则至

少需用多少位二进制数才能满足精度要求呢?设至少需用 m 位二进制数。因为 m 位二进制数可表示 2^m 个整数,所以,它不应小于$[a,b]$ 等分点的个数(包括两端点)$(b-a)\times 10^{\alpha}+1$,即 m 应满足:

$$(b-a)\times 10^{\alpha}+1\leqslant 2^m$$

同时,$m-1$ 位二进制数又不能满足精度要求,即

$$2^{m-1}<(b-a)\times 10^{\alpha}+1$$

因此,m 应满足:

$$2^{m-1}-1<(b-a)\times 10^{\alpha}\leqslant 2^m-1$$

现将$[a,b]$等分为2^m-1等分,则有2^m个等分点,用线性变换一对一地将m位二进制数所表示的 2^m 个整数点 $0,1,\cdots,2^m-1$ 映射到$[a,b]$ 中的 2^m 个等分点,于是,$[a,b]$ 中的等分点便与 m 位二进制数所表示的 2^m 个整数点一一对应。这个过程具体实现如下。

编码方法:

(1) 先由不等式 $2^{m-1}-1<(b-a)\times 10^{\alpha}\leqslant 2^m-1$ 确定自然数 m,再将$[a,b]$等分为 2^m-1 等分,等分点和两个端点共有 2^m 个。

(2) 用线性变换一对一地将 $[0,2^m-1]$ 中 2^m 个整数点 $0,1,\cdots,2^m-1$ 映射到$[a,b]$ 中的等分点。于是,$[a,b]$ 中的等分点便与$[0,2^m-1]$ 中整数点一一对应。因此,只要用二进制数表示出$[0,2^m-1]$ 中所有整数,便可对应出$[a,b]$ 中的所有等分点,产生一个 m 位二进制数 $B=b_{m-1}b_{m-2}\cdots b_0$ 作为编码形式。

注意到,m 位二进制数中最小者为 0,最大者为 $2^0+2^1+\cdots+2^{m-1}=2^m-1$,共有 2^m 个整数,故当二进制的位数为 m 位时,便可以表示$[0,2^m-1]$ 中所有整数,则$[0,2^m-1]$ 中所有整数均可用 m 位二进制数唯一表示。将 m 位二进制数(对应于$[0,2^m-1]$ 中整数) 与$[a,b]$ 中等分点一一对应如下。

解码方法:

(1) 设 $B=b_{m-1}b_{m-2}\cdots b_0$ 为任一个 m 位二进制数,则 $B\sim\sum_{i=0}^{m-1}b_i2^i=x^1$,而 x^1 为$[0,2^m-1]$ 中的整数,其中,$B\sim\sum_{i=0}^{m-1}b_i2^i=x^1$ 表示二进制数 B 对应于十进制数 $x^1=\sum_{i=0}^{m-1}b_i2^i$。

(2) 用如下线性变换将 x^1 对应于$[a,b]$ 中等分点 x,$x=a+(b-a)\times x^1/(2^m-1)\left(\text{其中,}\frac{b-a}{2^m-1}\text{ 为每等分小区间的长度}\right)$。

例 2.1 设 $[a,b]=[-1,2]$,精度 $\varepsilon=10^{-6}$,对解进行二进制编码。

解 因为 $2^{21}-1=2097151<3\times 10^6\leqslant 2^{22}-1=4194303$,所以,$m=22$。于是,

每一个 22 位的 0,1 字符串便可表示一个解。如 $B=1000101110110101000111$ 可表示解 x 如下：

$$B\sim x^1=2^{21}+2^{17}+2^{15}+2^{14}+2^{13}+2^{11}+2^{10}+2^8+2^6+2^2+2^1+2^0=2288967$$

$$x=-1+3x^1/(2^{22}-1)\approx 0.637197$$

2. 初始群体

设群体中所含个体数为 N，N 称为群体规模。随机产生 N 个 m 位二进制数，每个 m 位二进制数按上述方法与 $[a,b]$ 中某个等分点一一对应，这 N 个 m 位二进制数构成了初始群体，例如，$B_1=1001010001001011010101$，…，$B_N=0100111111000001010101$。

3. 适应度函数

适应度函数是衡量个体优劣的一个数量指标，通常取为一个正函数，当然，其值未必一定要取正，只要可以度量个体的优劣，且可使后续计算能简单执行即可。比较常见的取法如下。

对问题 $\max f(x)$：可取适应度函数为 $\mathrm{Fit}(x)=f(x)-f_{\min}$，其中，$f_{\min}$ 为 $f(x)$ 的下界。

对问题 $\min f(x)$：可取适应度函数为 $\mathrm{Fit}(x)=f_{\max}-f(x)$，其中，$f_{\max}$ 为 $f(x)$ 的上界。

当然，对 $\min f(x)$，也可取 $\mathrm{Fit}(x)=\dfrac{1}{M+f(x)}$ 等，其中，M 为常数且使分母为 π。一般地，适应度函数具有如下性质：① 取正值；② x 越好，$\mathrm{Fit}(x)$ 越大。

例 2.2　对 $\max f(x)=x\sin(10\pi x)+1$，$x\in[-1,2]$，若取 $\mathrm{Fit}(x)=f(x)+1=x\sin(10\pi x)+2$，则对一个 m 位二进制数 B，假设 B 对应于 $[-1,2]$ 中的等分点为 $x=0.637197$，则其适应度值为

$$\mathrm{Fit}(x)=f(x)+1=f(0.637197)+1=1.586345+1$$

若取例 2.1 中的二进制数

$$\bar{B}=11100000001111111000101\sim\bar{x}=1.627888\in[-1,2]$$

则有

$$\mathrm{Fit}(\bar{x})=f(\bar{x})+1=2.250650+1>\mathrm{Fit}(x)$$

故 $\bar{B}$(或 $\bar{x}$) 优于 B (或 x)。

4. 选择进化的父母

下面介绍赌轮选择方法(roulette wheel selection)。