

协同进化算法 及其应用

李 碧 郝志峰 著



科学出版社

协同进化算法及其应用

李 碧 郝志峰 著

广州市科学技术协会
广州市南山自然科学学术交流基金会
广州市合力科普基金会
广东外语外贸大学出版基金

资助出版

科学出版社
北 京

内 容 简 介

协同进化算法是通过模拟大自然生态进化中的协同进化现象而提出的仿生优化算法,是进化计算的一个新的重要分支。本书分为6章,分别简要介绍进化计算的发展和主要分支,阐述进化计算理论分析的含义和主要分析工具,综述协同进化算法的产生和发展,分析协同进化算法的实质,提出一种新的协同进化算法框架,并在此框架下实现两种算法,重点探讨协同进化算法在函数优化与图像矢量量化中的应用。

本书可作为计算机科学与技术、控制科学与工程及其相关专业的高年级本科生、研究生的参考书,也可供计算智能领域的相关教学科研人员 and 工程技术人员参考。

图书在版编目(CIP)数据

协同进化算法及其应用/李碧,郝志峰著. —北京:科学出版社,2013

ISBN 978-7-03-037213-0

I. 协… II. ①李… ②郝… III. 协同进化—算法—研究 IV. TP18

中国版本图书馆 CIP 数据核字(2013)第 056459 号

责任编辑:高 嵘 袁池华/责任校对:王望容

责任印制:彭 超/封面设计:苏 波

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>

武汉市首壹印务有限公司印刷
科学出版社发行 各地新华书店经销

*

开本:A5(890×1240)

2013年3月第 一 版 印张:5 1/4

2013年3月第一次印刷 字数:159 000

定价:35.00 元

(如有印装质量问题,我社负责调换)

前 言

自 20 世纪 60 年代以来,进化计算(evolutionary computation)得到了快速全面的发展。20 世纪 90 年代中后期,进化计算通过吸收新的理论,产生了一些新型的进化计算分支。协同进化算法(coevolutionary algorithm)就是通过模拟生态进化中普遍存在的协同进化现象而提出的一种新的进化算法。协同进化算法相比于遗传算法等传统进化算法的本质区别,是采用了不同的个体适应度的评价方式:传统进化算法根据个体的染色体来计算个体的适应度,是一种绝对适应度;而协同进化算法根据个体在与其他个体发生关系时的表现来确定适应度,是一种相对适应度。协同进化算法是对生态进化更高层次的模拟,具有更强的自适应性和更广阔的应用前景。

本书针对协同进化算法的算法模型、理论基础和优化应用进行探讨,本书共 6 章内容:第 1 章介绍进化计算的基本原理,把进化计算分为传统进化算法和新兴进化算法两大类,传统进化算法包括遗传算法、进化规划、进化策略;新兴进化算法包括分布估计算法、差分进化算法、协同进化算法。第 2 章阐述进化计算理论分析的含义,对主要的理论分析工具的优缺点进行述评。第 3 章首先分析协同进化算法的生态学基础和特征,并把协同进化算法分为竞争协同进化算法和合作协同进化算法。竞争协同进化算法中的个体适应度取决于个体在竞争中能够打败对手的数量和优秀程度,合作协同进化算法中的个体适应度决定于个体在合作中的表现。然后从算法流程、适应度计算、算法特点、算法应用四个方面对竞争协同进化算法和合作协同进化算法进行综述。第 4 章提出一种基于竞争协同进化算法的函数优化方法。该算法把竞争协同进化算法和遗传算法结合在一起,由个体在竞争中的表现决定个体的生存能力。第 5 章介绍一种统一的协同进化算法(unified coevolutionary algorithm, UCEA),该算法能够体现生态系统中个体之间的多种关系,把个体所受的影响归纳为收

益、受害、中性三种。根据影响方式的不同,提出两种具体的实现方式,即UCEA-I和UCEA-II,通过函数优化实验对UCEA-I和UCEA-II进行测试和分析。第6章在对图像矢量量化的经典算法LBG算法进行较为系统的分析之后,提出一种基于协同进化算法的图像矢量量化码书设计方法,提高码书的性能,为全局最优码书的设计提供新的思路。

感谢国家自然科学基金(61070033)、教育部人文社会科学研究青年基金(12YJC190031)广州市科技计划项目(2009J1-C401)、广州市科学技术协会、广州市南山自然科学学术交流基金会、广州市合力科普基金会和广东外语外贸大学出版基金的资助。还要感谢林土胜教授的精心修改,蒋盛益教授的细心指点,以及林用满、罗蓉芳、廖亮、刘清、罗忠亮、杨军、赵晓芳、金小莉、吴艳等博士的建议。

协同进化算法是进化计算领域中一个正在快速发展的新兴分支,其理论与应用方面均存在大量亟待进一步深入研究的问题。由于著者学识水平有限,本书难免疏漏之处,敬请专家、学者与诸位读者不吝指正。

著者

2012年9月于广州

目 录

前言

第 1 章 绪论	1
1.1 进化计算的发展	2
1.2 进化计算的主要分支	5
1.3 进化计算的应用.....	20
第 2 章 进化计算的理论分析	22
2.1 进化计算理论分析的含义.....	22
2.2 进化计算的理论分析方法.....	23
2.3 本章小结.....	33
第 3 章 协同进化算法的研究现状	34
3.1 协同进化算法的生态学基础.....	35
3.2 协同进化算法的内涵.....	41
3.3 竞争协同进化算法.....	46
3.4 合作协同进化算法.....	58
3.5 本章小结.....	67
第 4 章 基于竞争协同进化算法的函数优化	68
4.1 基准测试函数.....	69
4.2 多种群遗传算法.....	76
4.3 基于 GAMLC 算法的函数优化	82
4.4 基于 MGACC 算法的函数优化	87
4.5 本章小结.....	96

第 5 章 一种统一的协同进化算法	98
5.1 统一协同进化算法模型	98
5.2 简单统一协同进化算法	100
5.3 改进的统一协同进化算法	104
5.4 本章小结	111
第 6 章 基于协同进化算法的图像矢量量化码书设计方法	112
6.1 矢量量化的基本原理	113
6.2 初始码字间距最大化的初始化码书产生方法	120
6.3 基于边缘训练矢量的空胞腔消去方法	125
6.4 基于遗传算法的聚类分析方法	129
6.5 基于协同进化算法的码书设计算法	136
6.6 本章小结	140
参考文献	142

第 1 章

绪 论

64 301 8808 of Doc 9

在实际应用中,经常遇到这样的最优化问题:在一定的条件下,如何获取最佳的收益或效果。一般最优化问题的数学描述如式(1-1)。

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{s. t. } \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X} = \{\mathbf{x} \mid x \in \mathbf{D} \text{ and } g_i(\mathbf{x}) \leq c_i, i = 1, 2, \dots, m\} \end{aligned} \quad (1-1)$$

式中, $f(\mathbf{x})$ 为目标函数(objective function); \mathbf{x} 为决策向量(decision vector); \mathbf{X} 为参数空间(parameter space); $g_i(\mathbf{x}) \leq c_i$ 为约束函数(constraints); \mathbf{D} 为决策向量的定义域。

根据决策向量 \mathbf{x} 的取值的特点,可以把优化问题分为函数优化和组合优化两种。函数优化问题中的决策向量 \mathbf{x} 的取值是连续的,组合优化问题中的决策向量 \mathbf{x} 的取值是离散的^[1]。

如果式(1-1)中的目标函数是由多个函数组成,那么该优化问题就可以看成是一个多目标优化(multiobjective optimisation)问题^[2],其数学描述如式(1-2)。如果式(1-2)的目标函数中包含的函数个数 k 超过 5,那么该多目标优化问题一般被看成一个高维多目标优化问题。

$$\begin{aligned} & \min \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & \text{s. t. } \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X} = \{\mathbf{x} \mid x \in \mathbf{D} \text{ and } g_i(\mathbf{x}) \leq c_i, i = 1, 2, \dots, m\} \end{aligned} \quad (1-2)$$

如果式(1-1)中约束条件中还包含最优化问题,那么该最优化问题就是一个双层(bilevel)优化问题,其数学描述如式(1-3)。如果双层优化问题中的目标函数由多个函数组成,那么该双层优化问题是一个多目标

双层优化问题^[3]。

$$\begin{aligned}
 & \min_{x \in X} f(\mathbf{x}, \mathbf{y}) \\
 & \text{s. t. } g(\mathbf{x}, \mathbf{y}) \leq 0 \\
 & \text{where } \mathbf{y} = \arg \min_{y \in Y} \phi(\mathbf{x}, \mathbf{y}) \\
 & \text{s. t. } \varphi(\mathbf{x}, \mathbf{y}) \leq 0
 \end{aligned} \tag{1-3}$$

没有特别说明的最优化问题一般指的是单层单目标优化问题。对于多目标优化问题和双层优化问题,常用的策略是转为单层单目标优化问题。所以,单层单目标优化问题受到了较多研究人员的关注。

许多科学与工程问题都可以转化为优化问题,优化问题得到了广泛和深入的研究^[4]。特别是计算机硬件、计算机软件技术的快速发展,以及多学科的交叉渗透,极大地促进了优化算法的发展。解决最优化问题的优化算法可以分为经典优化算法和启发式优化算法。经典优化算法常指运筹学中的线性规划、动态规划、整数规划等算法。经典优化算法一般无法求解大规模的优化问题。

启发式优化算法一般是受大自然中的优化过程的启发而提出来的,因此而得名。启发式优化算法又叫智能优化算法、现代优化算法、黑盒优化算法(black-box optimization algorithm)。由于启发式优化算法只需要考虑输入与输出之间的关系,而不需要深究造成这种关系的原因,所以启发式优化算法便于处理因果关系不是很明确的优化问题。典型的启发式算法有模拟大自然中进化现象的进化算法(evolutionary algorithm, EA)^[5]、模拟金属物体退火现象的模拟退火(simulated annealing, SA)^[6]、模拟人类大脑的组织结构和运行机制的人工神经网络(artificial neural network, ANN)^[7]、模拟蚂蚁觅食原理的蚁群算法(ant colony algorithm, ACA)^[8]、模拟鸟群飞行觅食行为的粒子群算法(particle swarm optimization, PSO)^[9]。

1.1 进化计算的发展

在自然界,生物通过优胜劣汰和遗传变异来达到进化的目的。生态系统经历了从简单到复杂、从低级到高级的生物进化过程,进化过程本身是

一个自然的、自适应的、并行的、鲁棒的优化过程。生态学上的这些认识在计算方法中的渗透导致进化计算(evolutionary computation, EC)的产生和发展。

对于进化计算的研究报道最早见于20世纪50年代,Box等计算机领域的科学家独立地开始研究进化系统^[10],其思想是将自然界中的进化思想引入工程研究领域以解决工程中的优化问题^[4]。应用进化算法来求解优化问题的含义是:使用进化过程中的遗传、变异、选择等概念,并把它们作为算子参与优化。进化计算在60年代得到了快速的发展。Holland等提出了遗传算法和模板理论^[11]。由于遗传算法简单、易用、有效,所以得到了广泛的研究和应用。Rechenberg提出了进化策略(evolution strategies)^[12]。Fogel等提出了进化规划(evolution programming)^[13],进化规划把给定的问题描述成为有限自动机,通过施加进化算子,从而达到优化的目的。进化计算成熟于70年代。Holland于1975年出版著作“*Adaptation in Natural and Artificial System*”^[5]是标志性的事件。

遗传算法、进化规划和进化策略三者一起构成了进化计算的主要框架,它们从不同层次、不同角度模拟生态进化的原理,从而达到求解问题的目的。进化计算的求解过程示意图如图1-1,求解算法伪代码如下^[14]:

```
begin
  t:=0;
  initialize P(t);
  evaluate P(t);
  while not terminate do
    {
      P'(t):=variation[P(t)];
      evaluate [P'(t)];
      P(t+1):=select[P'(t) ∪ Q(t)];
      t:=t+1;
    }
  End
```

其中, $P(t)$ 表示第 t 代的种群, $Q(t)$ 是 $P(t)$ 的一个子集,执行选择操作的依据是个体适应度的大小。进化计算的搜索空间是一个集合,根据进

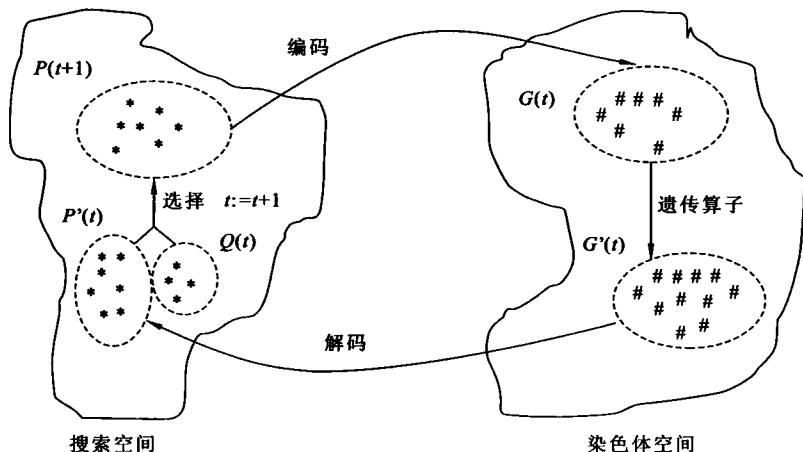


图 1-1 搜索空间和染色体空间的关系示意图

化计算具体的应用对象的不同,进化计算的搜索空间中的元素也不一样。例如,搜索空间中的元素可以是函数的参数、人脸检测系统中的参数、矢量量化中的码书、计算机程序、电路等。进化计算中的遗传算子(如交叉、变异)的作用对象是染色体,所以应用进化计算时,必须把搜索空间中的元素编码成遗传算法能够操作的染色体。在计算染色体的适应度时,又得把染色体解码成搜索空间中的元素。搜索空间和染色体空间的关系如图 1-1。由于进化算法的作用对象是染色体串,所以必须把解空间中的解数据转化为染色体串,这个转换过程叫做编码。常用的编码方案是二进制编码,也可以根据实际需要采用实数编码、字符编码、格雷(Grey)编码等其他的编码方案。

进化计算、模糊系统、人工神经网络是当前计算智能的三大研究领域,而其中的进化计算是计算智能的基础,受到了最多的关注^[15]。计算智能研究领域的一个标志性的事件是 1994 年 6 月在美国佛罗里达州的奥兰多(Orlando)召开的第一届世界计算智能大会(IEEE World Congress on Computational Intelligence, WCCI)。WCCI 包含三个学术会议,即国际神经网络会议(the International Conference on Neural Networks, ICNN)、模糊系统国际会议(IEEE International Conference on Fuzzy Systems, FUZZY-IEEE)、

进化计算大会(IEEE Congress on Evolutionary Computation, CEC)。前四届 WCCI 是每四年举行一次,第二届 WCCI 于 1998 年 5 月在美国阿拉斯加州的安科拉基(Anchorage)召开,第三届 WCCI 于 2002 年 5 月在美国夏威夷的檀香山(Honolulu)召开,第四届 WCCI 于 2006 年 7 月在加拿大大不列颠哥伦比亚省的温哥华(Vancouver)召开。从第五届开始是每两年举行一次,2008 年 6 月在中国香港召开了第五届 WCCI,2010 年 7 月在西班牙东北部港市巴塞罗那(Barcelona)召开了第六届 WCCI,2012 年 7 月在澳大利亚的布里斯本(Brisbane)召开了第七届 WCCI,第八届 WCCI 将于 2014 年在中国北京举行。WCCI 是目前计算智能领域规模最大的学术活动。

1.2 进化计算的主要分支

Foster 根据进化计算实现形式的不同,把进化计算分为两大类^[16]。第一类进化计算是在进化的过程中对某一问题的参数进行优化,进化的结果是某一问题的解,这一类进化计算叫做进化算法。进化算法可以进一步分为三类:遗传算法(genetic algorithm, GA)、进化规划(evolutionary programming, EP)、进化策略(evolution strategies, ES)。另一类进化计算是在进化的过程中对能够执行计算的对象进行描述,进化的结果是有关问题的解决过程,这一类进化计算叫做遗传规划(genetic programming, GP)。在遗传规划中,计算机自己写出程序,或者设计出其他的计算机。遗传规划又可以分为进化程序(evolving programs, EP)与进化硬件(evolvable hardware, EH)两类。在进化程序中,进化的结果是软件;在进化硬件中,进化结果是构成计算机的硬件。Foster 对进化计算的分类如图 1-2^[16]。

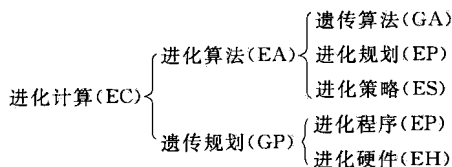


图 1-2 Foster 的进化计算分类图

进化计算的不同分支从不同的层次和不同的角度出发来模拟生物进化过程,但是进化计算的各分支之间的区别不是绝对的。进化计算应该理解为一个通用的解决问题的思想,而不是现成算法的集合。进化计算的发展趋势是各种不同算法的混合,有时将某一种算法叫做遗传算法、进化策略或者进化规划甚至会产生误导。

不同的学者对于进化计算的分类也有不同的看法。Eberhart 等认为进化计算不仅包括遗传算法、进化规划、进化策略、遗传规划,还包括粒子群算法^[15]。Bäck 把遗传规划归入遗传算法领域,把进化计算分为遗传算法、进化规划、进化策略三个分支^[14]。K. A. De Jong 在 2006 年出版的专著“*Evolutionary Computation: A Unified Approach*”中,对进化计算中的概念、思想、方法、流程图等关键问题进行了统一,并且把遗传算法、进化规划、进化策略这三个分支合并为进化算法^[17]。

本书把进化过程中使用了选择、遗传、变异等概念的优化算法归纳为进化计算。把进化计算分为传统进化计算和新兴进化计算两大类,如图 1-3。传统进化计算包括遗传算法、进化策略和进化规划。进化程序和进化硬件被看成是遗传算法的两种典型应用方式,属于遗传算法。新兴进化计算包括分布估计算法(estimation of distribution algorithm,EDA)、差分进化算法(differential evolution algorithm,DEA)、协同进化算法(coevolutionary algorithm,CEA)。传统遗传算法优化的对象一般是参数,而进化程序和进化硬件优化的对象是计算机程序和计算机硬件,但是进化程序和进化硬件采用遗传算法的思想,所以本书把进化程序和进化硬件归入遗传算法,把进化程序和进化硬件视为遗传算法的两个特殊的应用领域。分布估计算法、差分进化算法、协同进化算法使用了选择、遗传、变异等概念,但是采用了很多有别于传统进化计算的新思想和新理论。

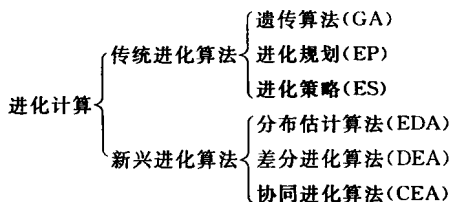


图 1-3 进化计算的分类图

1.2.1 遗传算法

遗传算法是整个进化计算领域发表研究成果最多的分支,它是进化计算体系的中心^[18]。遗传算法一般被认为是密歇根大学的 Holland 等借鉴生物进化中的生存竞争和优胜劣汰现象而提出的全局优化算法^[5],Holland 还提出了模板理论,为遗传算法奠定了一定的理论基础。Holland 的同事和学生进一步发展了遗传算法,K. A. De Jong 是 Holland 学生中的杰出代表,在进化计算领域作出了突出的贡献^[19]。Vose^[20]、Goldberg^[21,22]、Schaffer^[23]、Rowe^[24]、Mininno^[25]、Figueroa^[26]、陈国良^[18] 等很多学者进一步做了很多有关遗传算法的研究工作。

20 世纪 80 年代,工程领域不断涌现出超大规模的非线性系统,激发了遗传算法的发展高潮,并延续至今。从 1985 年在美国卡耐基·梅隆大学召开的第一届国际遗传算法会议(International Conference on Genetic Algorithms; ICGA'85),1993 年麻省理工大学主办的“*Evolutionary Computation*”创刊,到 1997 年 IEEE 的“*IEEE Transactions on Evolutionary Computation*”创刊,遗传算法作为具有系统优化、适应和学习的高性能计算和建模方法的研究日渐成熟。

由于遗传算法不苛求问题的动力学信息(如连续、可微等),所以它在解决非线性等复杂问题上具有传统优化算法所不具备的优越性,已被成功地应用于各种优化问题,已经在函数优化^[23]、多目标函数优化^[2]、神经网络的参数优化^[27]、特征提取^[28]、边缘检测^[29]、图像压缩^[30]、图像重构^[31]、矢量量化码书设计^[32,33]、多媒体广播路由^[34]、盲信号分离^[35]、滤波器设计^[36]、社区发现^[37] 等很多领域中取得了成功的应用。

1. 遗传算法的结构

遗传算法将选择、交叉、变异等遗传算子应用于一群对搜索空间编码的染色体中,在每一代,遗传算法同时作用于整个搜索空间的不同区域,

通过“优胜劣汰”，去掉解空间中期望值较低的部分，保留高期望值部分，从而能以较大的概率找到最优解。遗传算法的流程示意图如图 1-4。

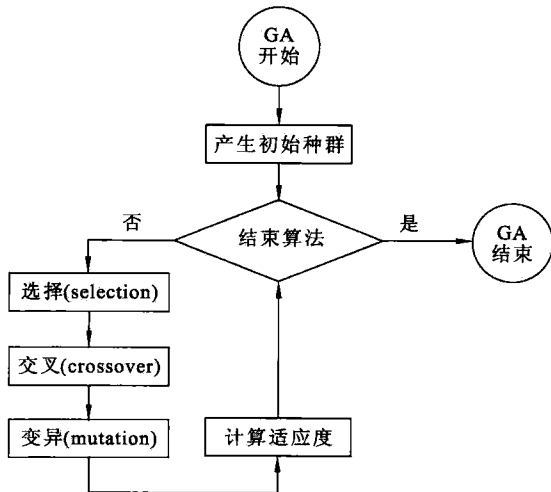


图 1-4 遗传算法流程示意图

(1) 产生初始种群。由于遗传算法的群体型操作的需要，首先必须为遗传算法提供由若干初始解组成的初始群体，初始群体可以只包含一个群体，也可以包含多个子群体。群体的规模非常关键，群体的规模指的是群体中个体的个数，一般情况下，群体的规模越大越好，因为大规模的群体有利于保持种群的多样性。但是，群体的规模太大会导致时间消耗很多。同时，Chen 等证明了进化算法中的大规模的群体可能没有帮助^[38]。在遗传算法中，初始群体一般是随机产生的，采用不同的随机产生器来得到初始群体，对遗传算法的性能有一定的影响。

(2) 选择。选择算子就是从当前的群体中选出优良的个体，淘汰劣等的个体，从而实现生物进化中的“适者生存”原则，选择算子有时又称为再生算子或复制算子。选择算子是影响遗传算法性能的最主要的因素，选择压力(selective pressure)描述了选择机制挑选种群中不同个体作母体的概率大小的差异，选择压力的合适程度在很大程度上决定了选择算子

的性能。选择压力过大,会造成几个较好可行解(不一定是近似全局最优解)迅速占据了整个种群,迅速出现局部最优;选择压力过小,会使算法呈现纯粹的随机徘徊行为。理想的选择算子能够有效避免有效基因的缺失,提高全局收敛性和效率,在收敛速度和保持种群的多样性之间取得一个好的平衡。目前常用的选择算子有以下几种方法:适应度比例方法(fitness proportional model)、最佳个体保存方法(elitist model)、排序选择方法(rank-based model)^[39]。

适应度比例方法是目前遗传算法中最基本也是最常用的选择方法,它也叫作赌轮选择(roulette wheel selection),在该方法中,个体通过选择算子的概率和其适应度值成正比。其优点是实现简单,收敛速度快;缺点是选择的压力过大,容易出现超级个体,导致早熟现象。适应度定标(fitness scaling)是为了缓和选择压力过大的毛病而提出来的,适应度定标的方法有线性定标、指数定标等方法。

最佳个体保存方法的思想是把群体中适应度最高的个体不进行配对交叉而直接复制到下一代中。此种选择算子又称复制(copy)。采用此选择方法的优点是,进化过程中某一代的最优解可不被交叉和变异算子所破坏。此方法一般都与其他选择方法结合使用。

排序选择方法是指在计算每个个体的适应度后,根据适应度大小顺序对群体中个体排序,然后把事先设计好的概率表按序分配给个体,作为各自的选择概率。其优点是克服了赌轮选择的缺点,但是它的收敛速度比较慢。

(3) 交叉。交叉算子,也叫复制算子(reproduction),类似于生物进化中的交配现象,遗传算法中的交叉算子的含义是把父体和母体中的染色体按一定的概率进行组合,得到下一代,从而实现生物进化中的遗传现象。交叉算子是一种全局算子,它是遗传算法中最主要的遗传算子。

在自然界生物进化过程中起核心作用的是生物遗传基因的重组(加上变异)。同样,遗传算法中起核心作用的是遗传算子中的交叉算子。通过交叉,遗传算法的搜索能力得以飞跃提高。Holland认为性能好的遗传算法要能识别好的积木块,并能把这些好的积木块组合成更大的积木块。也就是说一个理想情况下的交叉算子既要破坏母体和父体中的积木块,

又要能够把那些积木块组合起来,组成更大的积木块。但是,参与交叉的母体和父体的除非是相同的,否则积木块是很有可能被破坏的。交叉算子的性能对遗传算法性能的高低有着极大的影响。基本的交叉算子有三个:单点交叉(one-point crossover,简称1P)、两点交叉(two-point crossover,简称2P)和均匀交叉(uniform crossover,简称UC)。交叉算子的性能与交叉概率的大小有着紧密的联系,交叉概率是指配对的两个个体发生交叉操作的概率。由于交叉算子是全局搜索,所以交叉概率不能太小,可取90%~100%,而交叉概率可取0.75~0.95。

单点交叉又叫简单交叉,具体操作是:在个体串中随机设定一个交叉点,实行交叉时,该点前或后的两个个体的部分结构进行互换,并生成两个新个体。两点交叉的操作与单点交叉类似,只是设置两个交叉点(依然是随机设定)。所谓均匀交叉是指能通过设定屏蔽字(mask)来决定新个体的基因继承两个旧个体中哪个个体的对应基因。个体A与个体B均匀交叉操作产生个体A'和个体B',过程为:当屏蔽字为0时,新个体A'继承旧个体A中对应的基因;当屏蔽字为1时,新个体A'继承旧个体B中对应的基因,由此生成一个完整的新个体A'。反之,可生成新个体B'。显然,均匀交叉包括在多点交叉范围内。

单点交叉与两点交叉能较好地保存母体和父体中的积木块,但是它们的重组能力比较小,即它们的探测能力(exploration)比较小。均匀交叉的性能则和前两者相反,它们的重组能力比较强,即它们很有可能发现新的积木块;但是它们保存以前搜索的成果的能力比较差,即它们很有可能破坏现存的积木块。那么总的来说,两点交叉和均匀交叉这两个交叉算子中哪一个是较好的呢?学术界还没有统一,Holland建议采用两点交叉,有的学者认为均匀交叉的性能较好^[5]。所以,可以这么认为,当群体很小时,应采用均匀交叉来提高算法的搜索性,当群体很大时,则可以采用两点交叉来保留母体和父体中的积木块。由于交叉算子的“保存能力”和“重组能力”无法兼得,所以不少学者根据需要设计出一些改进的算子,在上述的两个能力之间取得一个折中(trade-off),比较典型的是多点交叉。

(4) 变异。变异算子的含义是群体中的个体的染色体的每一个基因