

# 遗传算法运行机理的研究

## The Research On Running Mechanism Of Genetic Algorithm

作者：郭东伟

指导教师：刘大有 教授  
周春光 教授  
单 位：吉林大学计算机科学与技术学院  
专业名称：计算机软件  
论文答辩日期：2002 年 6 月 13 日  
授予学位日期：

答辩委员会主席：孙吉贵 教授  
论文评阅人：史忠植 梁艳春

二零零二年四月

---

论文提交日期：2002 年 5 月 8 日

论文答辩日期：2002 年 6 月 13 日

关键词：遗传算法、收敛性、动力学形态

Key Words: Genetic Algorithm, Convergence, Dynamical  
Shape

总页数：130 页

开本：16 开

有图表

---

未经本论文作者书面授权，收存和保管本论文书面版本、电子版本的任何单位和个人，均不得对论文的全部或部分内容进行任何形式的复制、修改、发行、出租、改编等有碍作者著作权的商业性使用（但纯学术性使用不在此限）。否则，应承担侵权的法律责任。

---

## 提 要

相对于进化算法的广泛应用，它的理论基础仍有待进一步研究。本文对进化算法，特别是遗传算法的运行机理进行一定的研究。使用动力学方法对包含选择算子、交叉算子的标准遗传算法进行建模，得到了标准遗传算法的在平凡不动点处的局部收敛性条件，明确了局部极值点的定义。以上结论不仅对于基于适应度比例选择的选择算子和两点交叉算子成立，对于其他形式的选择算子和交叉算子也有类似结论。同时，本文中针对一个典型问题分析了遗传算法的全局动力学形态，计算了不同收敛区域之间的分割和比例。本文明确定义了适应度比例选择算子的取代时间，得出了取代时间的阶与适应度函数的选取及初始群体的分布无关的结论。同时，提出了一种基于聚类分析和资源竞争模型的生境遗传算法，可以在一定程度上解决多模函数的优化问题。

---

# 目 录

<b>第一章 绪论 .....</b>	<b>1</b>
1.1 进化计算 .....	1
1.2 进化算法简史 .....	2
1.3 进化算法的研究内容与前景 .....	3
1.4 本文的研究内容 .....	4
<b>第二章 进化算法基础 .....</b>	<b>7</b>
2.1 进化算法 ( Evolutionary Algorithms, EAs ) .....	7
2.1.1 进化规划 ( Evolutionary Programming , EP ) .....	8
2.1.2 进化策略 ( Evolution Strategies , ES ) .....	8
2.1.3 遗传算法 ( Genetic Algorithms, GA ) .....	9
2.2 进化算法的基本结构 .....	9
2.2.1 编码 .....	10
2.2.2 适应度度量 .....	12
2.2.3 进化算子 .....	13
2.3 遗传算法的基本结构 .....	14
2.3.1 GA 中的选择算子 .....	15
2.3.2 GA 中的变异算子 .....	17
2.3.3 GA 中的交叉算子 .....	18
2.4 小结 .....	18
<b>第三章 进化算法的理论分析方法 .....</b>	<b>19</b>
3.1 收敛性的含义 .....	20
3.2 GA 建模使用的数学方法 .....	21
3.2.1 随机变量 .....	21
3.2.2 矩阵初步 .....	23
3.2.3 随机过程 .....	24
3.2.4 动力系统 .....	25
3.3 GA 建模的非 Markov 方法 .....	28

---

3.3.1 模式进化 .....	28
3.3.2 群体适应度的累积度 .....	29
3.3.3 等位基因边际分布 .....	30
3.4 使用 Markov 链的模型 .....	31
3.4.1 有限搜索空间上的齐次模型 .....	32
3.4.2 有限搜索空间上的非齐次 GA .....	33
3.5 独立算子的收敛性结果 .....	35
3.5.1 选择压力 .....	35
3.5.2 基因漂移 .....	36
3.5.3 随机漫游 .....	37
3.6 其他 GA 建模方法 .....	38
3.7 小结 .....	40
<b>第四章 GA 的动力学模型及收敛性分析 .....</b>	<b>41</b>
4.1 齐次 GA 的动力系统模型 .....	41
4.2 SGA 的一般动力学模型 .....	42
4.3 单点交叉算子的数学特性 .....	44
4.4 SGA 的局部收敛性 .....	46
4.5 一般形式的遗传算法 .....	50
4.6 不同选择算子的模型 .....	52
4.6.1 基于排名的选择策略 .....	52
4.6.2 基于锦标赛方式的选择策略 .....	55
4.6.3 不同选择算子的局部收敛性的比较 .....	56
4.7 GA 的收敛条件对算法改进的指导作用 .....	59
4.7.1 交叉概率的影响 .....	60
4.7.2 适应度函数的变化 .....	60
4.7.3 编码方式的变化 .....	61
4.7.4 交叉算子的变化 .....	61
4.7.5 基因连锁 .....	62
4.8 小结 .....	62
<b>第五章 遗传算法的全局动力学形态 .....</b>	<b>64</b>

---

5.1 全局动力学形态 .....	64
5.2 2bit 问题的动力学模型 .....	64
5.3 SGA 的模型及分析 .....	67
5.3.1 . 全部重组 , 没有变异的模型 .....	67
5.3.2 . 世代重叠模型 .....	71
5.3.3 . 部分交叉模型 .....	74
5.3.4 . 带有变异算子的模型 .....	76
5.4 小结 .....	78
<b>第六章 遗传算法取代时间的分析 .....</b>	<b>79</b>
6.1 选择压力 .....	79
6.2 选择算子与取代时间 .....	79
6.3 适应度比例选择算子的取代时间 .....	81
6.4 不同选择算子的取代时间的比较 .....	84
6.5 小结 .....	85
<b>第七章 生境遗传算法 .....</b>	<b>86</b>
7.1 背景 .....	86
7.2 生境机制 .....	86
7.2.1 拥挤 .....	87
7.2.2 共享 .....	88
7.2.3 聚类 .....	89
7.3 基于聚类分析的资源竞争(BCRC)算法 .....	90
7.3.1 算法结构 .....	90
7.3.2 改进的 K-means 算法 .....	90
7.3.3 算法中其他事项 .....	91
7.4 数学分析 .....	92
7.4.1 稳态时各生境所占的比例 .....	92
7.4.2 生境稳定存在的条件 .....	93
7.4.3 BCRC 算法与生境规则 .....	94
7.4.4 BCRC 算法的稳定性 .....	95
7.5 实验测试 .....	95

---

7.6 小结 .....	98
第八章 结论和进一步工作 .....	99
致 谢 .....	102
在读期间完成的主要研究工作 .....	103
参考文献 .....	104



# 第一章 绪 论

## 1.1 进化计算

进化论的出现作为 19 世纪三大重要自然发现之一,对人类文明的发展产生重要影响。在随后的一百多年来,人们逐渐意识到,进化论作为一种方法论,其影响远远超出了生物学,扩展到很多其他学科。近 30 年来,在计算机科学和数学的交叉领域,人们对借助进化规律解决问题的方法,产生了越来越深厚的兴趣。

进化算法(Evolutionary Algorithm, EA)是其中一种具有代表性的方法。这种方法借鉴生物进化过程,通过维持潜在解的群体,模拟自然界生物的繁殖、遗传和变异等机制,采用“优胜劣汰”的进化原则,形成了一系列用来进行问题求解的算法。更确切的说,进化算法是在使用计算机的问题求解系统中,基于进化概念的随机优化算法。这种研究和使用进化算法的新兴领域被称作进化计算(Evolutionary Computing, EC)<sup>[42]</sup>。

和其他优化算法相比,进化计算的主要特点有<sup>[98]</sup>:

### 智能性

进化计算的智能性包括自组织、自适应和自学习性等。自然选择消除了算法设计过程中的一个最大障碍,即需要实现描述问题的全部特性,并说明针对问题的不同特点算法应采取的措施。因此,使用进化计算的方法可以解决那些不了解具体结构的问题。从这个角度来看,进化算法属于人工智能中所说的弱方法,也就是不对问题领域作更多的假设,算法的结构和基础不依赖于领域相关知识,因此具有广泛的适应性。另一方面,经过针对实际问题仔细设计的编码方案、进化算子和适应度度量后,进化算法完全可以克服传统弱方法的缺点,以较快的时间得到优化解或较优解,避免了“组合爆炸”问题。

### 本质并行性

进化计算的本质并行性表现在两个方面:一是内在并行性(inherent

parallelism)。即算法本身非常适合于大规模并行处理。二是隐含并行性 (implicit parallelism)。由于进化计算采用种群的方式组织搜索,从而可以搜索解空间内的多个区域,并相互交流信息。例如,由遗传算法的隐含并行性定理<sup>[54]</sup>可以得知,虽然算法每代的计算量只与种群规模  $N$  成正比,而实质上已进行了大约  $O(N^3)$  次有效搜索。

另外,进化算法还具有全局优化性、通用性和可操作性强等特点。

## 1.2 进化算法简史

进化算法的初步尝试可以追溯到 20 世纪 50 年代。在 50 到 60 年代间的早期的尝试,没有吸引更多的追随者;在 70 到 80 年代间,独立发展了三个研究方向,即进化规划、进化策略和遗传算法。在 1990 年,出现了进化算法这个名词。作为一个超类,它包含了上述所有的分支和所有其他在问题求解领域基于进化感知的技术。从 90 年代早期,又提出了几种新的进化算法,其中,遗传规划 (Genetic Programming, GP)<sup>[77]</sup> 是其中具有代表性的一种算法。虽然每一类算法都有自己的特点,但在过去几年间,不同算法之间的界限变得越来越模糊。

进化计算在其早期没有得到应有的重视,其主要原因在于,在当时的人工智能领域,以自动推理、专家系统为代表的基于符号处理(知识或规则)的方法处于统治地位。到了 80 年代,人们逐渐意识到传统人工智能方法的局限性,因而提出了很多其他方法。以计算为基础的智能算法,如神经网络、模糊系统等,走上了舞台。这些导致了人们对进化计算进行了广泛和深入的研究。

目前,进化计算在诸如问题优化、自动控制、机器学习、模型预测等各个领域都有广泛应用。计算机科学、数学、物理学、化学、生物学和、经济学、社会科学及工程应用等各方面科学家都开展了进化计算的应用。在计算机科学领域,它涉及到人工智能、可计算性与计算复杂性、计算方法、并行处理和程序设计等方向的诸多课题。

从 80 年代中期开始,世界上许多国家都掀起了进化计算的研究热潮。目前,有数种以进化计算为主题的国际会议定期召开

[3,11,12,14,15,27,33,43,45,47,50,64,65,67,78,90,92,107,117,118,132,133]；互联网上有诸多关于进化计算的网站、邮件列表和新闻组。现已出版了两种关于进化计算的专门性国际期刊：“Evolutionary Computation”，由 MIT Press 出版，1993 年创刊；“IEEE Transaction on Evolutionary Computation”，由 IEEE 发布，1997 年创刊。此外，在其他刊物和学术会议上也能找到进化计算方面的文章。

从下面的数字中可以看出进化计算，特别是遗传算法的发展速度。在 1986 年 5 月由 Goldberg D.E. 等人汇编的遗传算法的参考文献汇总中<sup>[62]</sup>，只有 180 篇文章，而截止到 2000 年 12 月，仍由 Goldberg 等人重新修订的汇总中<sup>[63]</sup>，包括了 9000 个文章引用。这两个数字的比较说明，遗传算法在十几年间得到了蓬勃发展。

### 1.3 进化算法的研究内容与前景

进化计算的研究内容相当广泛，反映了多学科相互交叉的特点。目前，进化计算的研究主要涉及到理论和应用两个方面：

在实际应用方面，需要大量的实践，以积累各种有价值的经验和改进方法。在应用方式上，需要设计新的计算模型，如新近提出的免疫系统模型、协同演化模型和蚂蚁模型等。在实际计算中，并行和分布式计算方法对算法的影响日益得到重视。在应用领域中，除问题优化外，还涉及到工程应用的各个方面。在计算机科学领域，机器学习、模式识别和自动程序设计等子域也有广阔应用前景。进化计算中的各个分支之间的相互融合，进化计算与其他计算智能方法的结合成为目前研究的热点之一。为使进化计算真正进入实际应用，需要开发商品级的软件系统。在计算机体系结构领域，如果设计出适合进化计算的硬件，那么不仅会推动进化计算的应用，而且也将会对传统的冯·诺依曼体系结构产生巨大的挑战。

与进化计算方法广泛应用相比，在理论研究方面，尚存在很多不足。由于进化计算缺乏统一、完整的理论体系，目前一些理论结果主要集中在收敛性分析上，而且很难给出收敛速度的估计。如编码方案、控制参数和遗传算子等因素对算法性能的影响等问题，只能具体问题具体分析，并且在多数情况下还只能依赖于计算机模拟的实验数据。当前，迫切需要严密、

科学的一般规律和方法，对进化计算的研究进行指导。

从 90 年代初开始，进化计算、神经计算和模糊系统等逐步形成了一个新的研究方向：计算智能（Computational Intelligence, CI）。与传统人工智能（Artificial Intelligence, AI）方法相比，CI 得到了更加广泛的注意，也取得了很大进展。这主要由于 CI 有以下特征<sup>[57]</sup>：

**可缩放的结果：**传统的 AI 方法在实验室阶段能够工作，而对真实世界无能为力。而从最开始，CI 就是为了处理不同规模的问题，并随着 CI 的发展，能够处理的领域越来越大。

**实际性：**传统的 AI 方法只能处理理想环境，而 CI 从一开始就针对实际的计算问题：如复杂仿真、优化、自动控制等。这种特性现在仍在继续得到发扬。

**小而经济的模型：**当早期的 AI 似乎满足于严格算法理论与实际问题之间的屏障时，CI 找出了一套足够复杂以满足其设计的模型。虽然 CI 丢失了一些严格性，但这也正是使 CI 能够得到广泛应用的优点之一。

目前的人工智能领域已经包括传统的基于符号处理的符号主义学派、以神经网络为代表的连接主义学派、以人工生命为代表的现场学派和以进化计算为代表的进化主义学派等，呈现出百花齐放的态势。

## 1.4 本文的研究内容

本文针对进化算法，特别是遗传算法进行了一些理论上的探讨。这些探讨主要涉及遗传算法的收敛性、收敛速度和全局动力形态等方面，同时对目前得到广泛重视的生境遗传算法进行了初步研究。

对于标准遗传算法，建立数学模型，分析模型的收敛性是遗传算法理论研究中的重要一环。前人对此已做了许多研究工作，其中使用 Markov 模型的方法，虽然讨论了交叉算子的作用，只是将其作为一个模型中状态转换的算子，最终结论的取得需要依赖于变异算子的不可约性。在这种方法中，遗传算法中起关键作用的交叉算子，成为算法收敛性证明中的一个障碍，从而未体现出其在引导收敛目标中的作用。在动力系统的方法中，由于交叉算子的非线性，建模和分析的过程变得更加复杂。本文使用动力

系统的方法建立了包含选择算子和交叉算子的标准遗传算法的数学模型。首次阐明了局部极值点的含义，证明了算法在局部极值点的局部收敛性。在分析过程中，使用了按照海明距离重新排列基因型的方法，明确指出了交叉算子在算法收敛性中的作用。并将这个结论扩展到其他的选择算子和交叉算子上。

为深入了解遗传算法的运行机制，不仅要知道算法的收敛性，还需要了解算法在没有达到收敛时的特性。本文针对一个特定问题，深入到算法的运行过程，描述了其在各种算法参数时的全局状态，并说明了变异算子对算法全局动力形态的影响。

在讨论单一算子的作用时，选择算子收敛速度的衡量指标是取代时间。在以前定义取代时间时，混淆了群体规模和基因型种类这两个变量，因此本文重新定义了这个概念。并证明了取代时间的阶与适应度函数的选取和初始群体分布无关。进而提出了能更清晰反映选择算子效果的取代时间系数这个概念。

目前，采用生境遗传算法解决多模函数优化问题，得到了密切重视。本文将聚类分析和生境算法相结合，提出了一种新的生境遗传算法。该算法能够自适应地排除适应度较小的局部极值点。本文同时对这种算法作了初步的数学分析和实验验证。

本文的结构如下：

在第二章中，回顾了进化算法的历史发展，介绍了进化算法的基本结构，对算法中的重要因素，如编码、适应度度量等问题作了说明。对遗传算法的各个组成部分进行了描述。为后面为算法的建模做出了准备。

在第三章中，针对遗传算法，介绍了目前常见的各种理论分析工具和方法，并简要介绍了各种方法所得到的成果。对于我们后面使用的动力系统等方法，介绍了一些数学准备知识。

对于标准遗传算法，在第四章中证明了在无限群体和忽略变异算子时，其在局部极值点的收敛性，以及局部极值点的出现条件。证明首先是对适应度比例选择算子和单点交叉算子进行的。然后扩展到基于排名和基于锦标赛的选择算子，以及其他类型的交叉算子。然后根据这些结论，对改进遗传算法给出了一些理论上的建议和指导。

为进一步说明遗传算法的运行特性，在第五章中，我们讨论了遗传算法的全局动力形态。我们对于一个特定问题，建立了其在各种算法参数时的数学模型，找到了模型的全局吸引点和排斥不动点，并对全局吸引点计算了吸引区间。在这一章中，我们还对变异算子对算法全局动力形态的影响作了说明。

在第六章中，为了说明遗传算法，尤其是选择算子的收敛速度，我们重新明确定义了取代时间的概念，计算了适应度比例选择算子的取代时间，并证明了这个时间的阶不依赖于适应度函数的具体形式和初始群体的分布状态。

在第七章中，提出了一种新的基于聚类分析和资源竞争模型的生境遗传算法。这种算法将聚类分析、共享技术和拥挤技术有机地结合起来，可以有效地对多模函数进行优化，而无需事先确定生境的具体数目和生境半径的大小。通过数学分析，证明了这种算法可以控制收敛到的生境的数目，避免找到无效的极值点。

第八章是本文的结论和进一步的研究工作。

## 第二章 进化算法基础

### 2.1 进化算法 (Evolutionary Algorithms, EAs)

进化算法的基本思想是模拟自然界的进化方式。Darwin 的进化论的基本思想是“物竞天择”。在 EA 中，获得一个给定问题的解可以看作是一个生存任务：可行解之间为生存（或繁殖）的权利进行竞争，根据生物进化的经验，这种竞争可以产生出一个最具有适应性的解，也就是问题的优化解。

更详细地说，EA 维护一个群体，通过一定的进化算子来进行进化，这些算子包括选择、重组和变异。群体中的每一个个体都在进化，并赋予一个环境定义的适应度。选择操作着重于那些高适应度的个体，起到挖掘（exploiting）可用的适应度信息的作用；重组和变异算子改变这些个体，起到启发式扩展（exploration）的作用。虽然从生物进化的角度来看，这些算子比较简单，但已经可以提供足够健壮（Robust）和自适应(Adaptive)的搜索机制。

目前，越来越多的领域涉及到优化问题。这类问题目前尚没有快速合理的算法，其中很多问题属于 NP 问题，虽然 NP = P 问题尚未得到最后证明，但多数专家倾向于成立。因此这类问题在多项式时间内难以得到解决。

对于这类优化问题，可以被看作是对潜在解空间的一种搜索过程。如果规模较小，尚可以使用穷举搜索方法。当空间较大时，这种方法就不能使用了。当问题具有明确的数学特征时，可以使用基于数学的各种搜索算法，如线性规划、梯度下降及其他基于梯度的搜索算法等。对于没有显著的数学特征，经常是由离散空间组成的问题，如 TSP 问题（traveling salesman problem）、布线问题（wire routing）等，需要使用一些其他的人工智能技术。

进化算法作为一种概率算法，可以有效的对解空间进行搜索。同时，进化算法具有较强的通用性和适用性，它不依赖于问题的具体形式和领域知识（当然，引入领域知识可能有利于算法的运行），同时又具有内在的并

行性，因此能够在较短时间内找到最优解或准最优解。

下面我们介绍进化算法的发展历史和主要分支。

### 2.1.1 进化规划 ( Evolutionary Programming , EP )

EP 由 Fogel 在十九世纪六十年代年提出<sup>[46,48,49]</sup>。EP 根据问题使用特定的表现型，例如，在浮点优化问题中，个体的表示是实数值的向量；在 TSP 问题中使用有向序列表；在有限状态机问题中使用图。通常 EP 作为优化器使用，使用实数向量作为基因编码。

在初始化之后，所有的个体被选择为父本，然后变异，产生  $P$  个后代。这些个体被评价，然后使用概率的锦标赛选择算法从  $2P$  个个体中选择出  $P$  个个体组成下一代。通常保留最好的精英个体 ( Elitism )，以保证一旦得出优化解，就不会丢失。变异算子的形式基于表现型，通常是自适应的。例如，对于实数向量，个体中的每一个分量都有一个自适应的变异比例，变异通常是期望为零的分布（如正态分布）。在 EP 中不使用重组算子，由于变异的形式具有很强的机动性 ( flexible )，可以产生和重组类似的效果。

对于 EP，收敛定理保证基本算法以概率 1 全局收敛。这个结论是使用离散空间的 Markov 链得到的，也可以在计算机上由数值表现出来。在离散空间的格点上，对于包含距离最优点最近的个体组成的所有可能的群体，由于精英个体的存在，存在一个吸收态。

### 2.1.2 进化策略 ( Evolution Strategies , ES )

ES 由 Rechenberg 在 1973 年提出<sup>[108,109]</sup>，使用单个体的群体，选择和变异算子。后来，Schwefel 引入了重组算子，扩展到一个以上的个体，并与其他典型的优化技术进行了比较。由于最初是解决水力学问题，典型的 ES 使用实数向量表示个体。

在初始化和评价个体之后，随机均匀地选择个体作为父代。在典型的重组 ES 中，父代通过重组操作产生子代，然后子代通过变异算子进行变化。子代的数量大于父代的数量  $P$ 。生存选择是决定性的，通过下面两种方法之一来实现。第一种方法允许  $P$  个最好的子代个体存活，然后使用这些个体替换父代。这称之为 (  $\mu$ ,  $\lambda$  ) 方式， $\mu$  指群体规模， $\lambda$  指产生的子



代数目。第二种方法称为  $(\mu + )$  方式，允许在父代和子代中选择最好的  $\mu$  个存活。 $(\mu + )$  方法是一种精英算法，而  $(\mu, )$  算法不是。和 EP 一样，也是用自适应的变异算子。但和 EP 不一样在于，在 ES 中变异算子不扮演重要的角色。

ES 研究的很多理论涉及到收敛速率，试图加快这个速率。目前已有对于  $(\mu + )$  ES 算法的全局概率收敛证明。但对  $(\mu, )$  算法还没有得到类似的结论。

### 2.1.3 遗传算法 (Genetic Algorithms, GA)

GA 由 Holland 在 1975 年提出<sup>[71]</sup>，更早的思想可以追溯到六十年代<sup>[70]</sup>。典型的 GA 使用独立于问题的表示形式，也就是二进制位串，这种编码既适合于变异又适合于交叉，并且 Holland 强调交叉算子的搜索能力。随后，在其著作《Adaptation in Natural and Artificial System》中，他将其引入到自适应系统中，后来又推广到其他领域。

GA 的进化对象是由多个个体 (individuals) 组成的群体 (population)，在初始化之后，通过基于适应度的概率选择算法选择父代，并通过交叉 (crossover) 和变异 (mutation) 来维持群体的多样性。如此演化下去，直到期待的终止条件。

作为进化算法中最具有代表性的算法，遗传算法以其简单通用的编码技术和明白有效的进化操作得到了广泛的应用。目前的遗传算法已经不再局限于二进制编码。最近很多的应用尝试使用其他的形式，如图、Lisp 表达式、有序列表和实数向量等。这些改进说明，遗传算法在很大程度上与其他的进化算法已经融合在一起。

## 2.2 进化算法的基本结构

假设我们欲解决的优化问题如下：

$$\max\{f(x)|x??\}$$

这里  $?$  是问题空间，可以是离散空间，也可以是连续空间，即实空间  $R^n$  的一个子集。

使用进化计算方法进行优化通常是将原始问题进行编码 (encoding), 然后再进行优化的。这种编码就是将问题空间  $P$  中的点映射到基因空间  $O$  的过程。记作  $F: P \rightarrow O$ 。基因空间  $O$  既可以是二进制位串表示, 也可以是实空间  $R^n$  的一个子集。编码的逆过程称之为解码 (decoding)。对于我们重点研究的二进制位串空间,  $P \subseteq [0,1]^l$ ,  $l$  称为染色体长度。称问题空间中的点为表现型 (phenotype), 基因空间中的点为基因型 (genotype)。对于一个特定的基因型, 其对应的表现型的优化函数值称为适应度 (fitness)。当基因空间的编码方式为位串时, 也称之为染色体 (chromosome), 其中的每一位称之为基因 (gene)。基因的取值范围叫做等位基因 (allele)。

进化算法通常是针对多个潜在解同时进行的。这些潜在解构成一个数量为  $N$  的群体 (population)。群体中的每一个潜在解称之为个体 (individual)。进化是按代施加到群体上的。

广义的进化算法由以下几个步骤组成：

1. 初始化群体
2. Do
3. 计算适应度
4. 根据适应度选择下一代中存活的个体
5. 产生新的个体
6. Until 满足终止条件。

对于一个典型的 EA, 通常是从随机的初始化群体开始, 在某些情况下, 也可以使用相关的领域知识创建初始群体。适应度量度取决于个体对环境的价值。适应度的计算可以是一个简单的数学公式, 也可能复杂到运行一个特定的过程。选择通常分为两种: 繁殖选择和生存选择。繁殖选择决定谁将成为父代和可以产生多少个后代, 后代通过重组及变异来产生。生存选择决定群体中谁将存活。

下面以 GA 为主要代表, 详细介绍进化算法的一些概念。

### 2.2.1 编码

在使用进化算法计算之前, 需要对所求解问题进行编码。编码方法决定了基因型和表现型之间的转换方法, 某些特定类型的编码方法还决定了

遗传算子的选择。编码的好坏在很大程度上决定了算法的优劣。

由于编码方案取决于具体情况，因此上并没有一定的直到理论和评价原则。作为参考，De Jong 提出了两条操作性比较强的编码原则<sup>[38]</sup>：

一、（有意义积木块原则）：应使用能易于产生与所求问题相关的具有低阶、短定义长度的编码方案。

二、（最小字符集编码原则）：应使用能使问题得到自然表示或描述的具有最小编码字符集的编码方案。

这两条原则具有一般的指导性。随着时间的推移，使用的编码方式也越来越丰富。在实际使用时，我们还需要考虑其他的原则。尤其是对于约束优化问题，可能存在大量致死基因（lethal）。一种方法是在编码时避免不合法个体的存在，这样有可能造成编码和算子的复杂化与不一致性。另一种方法是对于不合法的个体，对其适应度加以惩罚。

### ✍ 位串编码

对于标准的遗传算法，最常用的编码方式为位串编码。也就是说，在进行进化操作时，将基因看作一个有序的位串序列，而不考虑每一位的含义。这种编码类似于生物染色体的组成，交叉和变异等遗传操作很容易进行。

使用位串编码时，二进制编码同时表示的模式数最多。并且实现简单，对于很多离散优化问题（如背包问题等），基因型与表现型的对应关系非常明确。因此得到广泛应用。

对于连续优化问题，二进制编码的主要问题是，相邻整数之间的 Hamming 距离可能很大，带来很多不必要的局部极值点（详见第四章），影响算法的搜索性能。这一缺点问题也称为 Hamming 悬崖（Hamming Cliff）。

克服这一缺点的一种方法是使用 Gray 编码<sup>[87]</sup>。Gray 编码与二进制编码的对应关系如下：

设二进制串  $(?_1, ?_2, \dots, ?_{n-1}, ?_n)$ ，对应的 Gray 编码串为  $(?_0, ?_1, \dots, ?_{n-1}, ?_n)$ ，则

$$?_k = \begin{cases} ?_1, & \text{if } k = 1 \\ ?_{k-1} \oplus ?_k, & \text{else} \end{cases} \quad (2.1), \text{ 同样 } ?_k = \begin{cases} ?_1, & \text{if } k = 1 \\ ?_{k-1} \oplus ?_k, & \text{else} \end{cases} \quad (2.2)$$

这里 $\oplus$ 表示模 2 加法，也就是异或操作。

Gray 编码的显著特点是对于距离为 1 的二进制表示，其 Gray 编码之间的 Hamming 距离为 1。因此可以在一定程度上克服上面提到的缺点。

### 实数编码

当问题空间是实数连续空间时，可以直接采用实数进行编码。对于实数编码，从理论上讲，二进制编码的各种遗传操作都可以使用，但实际应用时通常都使用专门针对实数编码设计的算子。从进化计算的历史上来看，进化策略和遗传规划都采用实数编码。近年来，遗传算法在求解复杂连续优化问题时也经常使用实数编码。实际上，使用实数编码的遗传算法和进化策略的区别已经越来越小。

### 结构化编码

对于很多具有明确的数据结构的问题，更加自然地表示是直接对这种数据结构进行操作。称之为结构化编码。常见的编码方式有树和图。这种编码方式一般是针对具体问题设计具体的编码和遗传算子。很难具有通用性。对于由 Koza 提出的遗传规划 (Genetic Programming)<sup>[77]</sup>，可以看作是使用逆波兰表达式的二叉树作为结构化编码的进化算法的例子。

## 2.2.2 适应度度量

在自然界中，适应度反映个体对于环境的适应程度，是自然选择的唯一参考因素。同样，在进化算法中，适应度用来反映个体的优化程度。

在选择算子使用适应度时，一般要求适应度是一个正实数；在某些策略下，也可以使用非数值化的适应度度量。从代数结构的角度来看，个体的适应度集合一般需要构成一个有序集，至少要构成一个格 (Lattice)。也就是说，至少要在两个适应度之间明确定义“小于”关系。

当欲求解的原始问题是数值优化问题时，可以直接将求解函数作为适应度。个体的适应度取值通常为正的实数值。一般情况下，要求当个体的

性能越好时，其适应度值越大，而且要求非负（如 GA 中的比例选择策略）。因此，有时需要对原始的适应度函数进行变换。

当原始问题是非数值优化问题时，一种方案是选择恰当的度量函数充当适应度函数，将某个可行解的适应度变换到正实数空间；另外一种方案是使用不基于适应度函数具体数值的选择策略。如下文提到的排名选择和锦标赛选择。

在很多情况下，原始的适应度函数（及其简单变换）存在一些不适合选择使用的特点。有时，适应度值之间的差别较小，导致选择效果不明显；有时，优势个体的适应度值过大，可能产生早熟收敛。在这些情况下，需要对原始的适应度函数进行某种变换，以得到更好的性能。这种变换的具体形式通常是由经验和试验的方式获得的。

### 2.2.3 进化算子

一般来说,各种进化算法的不同点在于产生新的个体与选择的方式不同。这种方式也称之为算子（operator），也有的文献称之为策略（strategy）。进化算子可以分为两类：选择算子和演化算子。在有些算法的具体实现中，这两种算子是混合在一起的。

选择算子充当自然进化中自然选择的角色，起到指引搜索方向的作用。其目的是提高具有较高适应度的个体或其后代存活概率。通过选择算子，可以使群体向更高适应度的方向前进。不同的选择算子导致不同的选择压力（selection intensity）。选择压力较大，算法的收敛速度较快，但也容易导致早熟收敛。

选择算子按照选择阶段可分为繁殖选择和生存选择；按照比较范围可分为种群选择和生境选择；按照计算方式可分为确定性选择和概率性选择。繁殖选择指通过选择确定那些个体可以用来产生下一代；生存选择指通过选择确定那些个体可以存活。种群选择指选择是在整个种群的范围内进行的，个体的适应度要和整个种群的适应度分布进行比较；生境选择指选择是在两个或几个个体（通常具有血缘关系）之间进行的。确定性选择算子使用确定性的算法进行选择；而概率性选择算子在选择过程中引入了随机性的因素。

演化算子充当自然进化中繁殖过程中遗传和变异的角色，起到维护种群个体构成多样性（diversity）的作用。它包括交叉（crossover）和变异（mutation）两种。交叉算子又称重组（recombination）算子，用于从两个父本产生一个新的个体。演化算子在进化算法中起到构造的作用，可以从一个、两个或多个个体出发，构造出新的个体。在产生和维护群体多样性的同时，也起到局部搜索的作用。

## 2.3 遗传算法的基本结构

遗传算法和其他进化算法相比的显著特征是：使用交叉算子产生下一代个体；使用繁殖选择式的选择算子、根据适应度随机选取父本；一般使用二进制位串式编码及相应的交叉算子。遗传算法的基本结构如下：

算法：

随机化初始群体  $P(0), t=0$ 。

While(不满足终止准则) do

{

  计算所有个体的适应度

  计算每个个体的选择概率

  均匀随机选择  $p_g * N$  个个体，直接插入到下一代群体  $P(t+1)$  中。

  for( $i=0$ ;  $i < (1-p_g) * N/2$ ;  $i++$ )

  {

    根据选择概率在  $P(t)$  中选择两个父本

$r = \text{random}(0,1)$ 。

    If  $r < p_c$ ，将两个父本不加改变地插入到下一代群体  $P(t+1)$  中

    Else

    {

      执行重组操作，产生两个子代。

      按照变异概率  $p_m$  对两个子代执行变异操作。

      将其插入到  $P(t+1)$  中

    }

}  
}

上面的算法中包含四个基本参数，它们分别是代间隔  $p_g$ ，交叉概率  $p_c$ ，变异概率  $p_m$ ，以及群体规模  $N$ 。

虽然存在很多对遗传算法的改进，但以上算法作为 GA 的基本算法，在 GA 的理论分析中得到了深入研究。一般称这个算法为 SGA (Standard GA) 或者叫 CGA (Canonical GA)。在本文中的理论分析也是以此算法为基础的。

### 2.3.1 GA 中的选择算子

上面已经提到，在 GA 中使用繁殖选择。常用的选择策略有如下几种：

#### (1) 基于适应度比例的选择

##### 转盘式选择 (roulette wheel selection)

这是 GA 中最常用的选择策略。在本算法中，要先计算整个群体的平均适应度，以及每个个体的选择概率  $p_k = f_k / \sum_i f_i$ 。计算过程中，产生一个  $[0,1)$  区间的均匀随机数  $r$ ，若  $k$  满足  $\sum_{i=0}^{k-1} p_i < r \leq \sum_{i=0}^k p_i$ ，则选择个体  $k$  (这里，个体的编号是从 0 开始的)。在这种方式下，个体被选择的概率直接和其适应度成正比。

从概率论的角度来说，选择过程相当于“有放回抽样”过程，其单一选中概率为  $p_k = f_k / \sum_i f_i$ 。若抽样过程执行  $N$  次，则此个体被选中次数构成一个二项分布，分布函数为：

$$P(x) = C_x^N p_k^x (1 - p_k)^{N-x} \quad (2.3)$$

$x = 0, 1, \dots, n$

这个概率分布的期望值是：

$$E(x) = N p_k = N \frac{f_k}{\sum_i f_i} = N \frac{f_k}{\bar{f}} \quad (2.4)$$

$$\text{方差为: } D(x) = N p_k (1 - p_k) = N \frac{f_k}{\sum_i f_i} (1 - \frac{f_k}{\sum_i f_i}) \quad (2.5)$$

##### 繁殖池选择 (breeding pool selection)

这种算法计算个体的选择概率的方式与转盘式选择相同，然后根据

$N_i \approx \text{round}(p_i * N)$  计算出每个个体的繁殖量的期望，将此个体复制  $N_i$  份放入一个繁殖池中。在繁殖池中通过均匀随机抽取个体进行重组，产生下一代群体。在上面的公式中，使用取整函数计算繁殖量，也可以使用四舍五入函数。

这种方式剥夺了那些期望繁殖量小于 1 的基因型的繁殖权。当群体规模不大时，产生的抽样误差较大。当群体规模足够大时，这种方式与转盘式选择是等价的。另外，由于在此算法中，选择算子与重组算子明确分开，因此算法结构更加清晰。

### 其他改进

在基本方式中，个体后代的期望值都不一定是整数，因此当群体规模不大时，可能产生较大的抽样误差。为此，提出了其他的一些改进方式，如剩余随机抽样、确实性抽样、随机整群抽样<sup>[9]</sup>等。

### (2) 基于排名的选择 (Ranking Selection)

在使用基于适应度比例的选择策略时，会出现由于某些个体的适应度过大而导致的早熟收敛。同时这种方式依赖于适应度函数的具体形式，对于某些应用来说，影响计算的收敛性和收敛速度。

基于排名的选择策略根据个体在整个群体中的适应度的好坏排名来分配选择概率，可以避免上述问题。

这种算法的基本形式是先根据所有个体的适应度进行排序，设排序后的次序为 1..N。然后对于每一个个体指定一个选择概率函数  $p_i$ 。 $p_i$  只和  $i$  有关，与  $f_i$  无关；且满足  $\sum_{i=1}^N p_i \approx 1$ ，以及对于  $i$  是单调递减的。

常见的选择概率函数的形式有线性排名<sup>[8]</sup>和指数排名<sup>[89]</sup>两种。

对于线性排名， $p_i \approx \frac{1}{N} (a + \frac{2 * (a - 1) * i}{N - 1})$  (2.6)。通常取  $a=1.1$ 。

对于指数排名， $p_i \approx \frac{q(1 - q)^{i-1}}{(1 - q)^{N-1}}$ ,  $i \approx N$  (2.7)。

基于排名的选择策略可以看作是对适应度函数的一种自适应变换，只不过这种变换是动态的，随着群体进化而变化，同时只和适应度的相对值有关，而和其绝对值无关。在本文的第四章将会详细分析这种算子。



### (3) 基于局部竞争的选择

以上两种方案都是根据个体在整个群体中的相对地位来决定其选择概率，因此需要整个群体的信息。当群体规模很大时，需要一定的额外计算量。同时也不利于并行计算。基于局部竞争的选择策略可以在一定程度上解决这个问题。

锦标赛选择 (Tournament Selection) 通过在随机选出的若干个个体之间进行竞争，适应度最大的个体获得优胜，从而被选出产生其后代。锦标赛选择的详细特性也将在第四章讨论。

生境 (niche) 竞争是最近研究得很深入的一个方向<sup>[100]</sup>。通过在群体中寻找相似的个体，或者在子代和父代之间进行锦标赛选择，适应度高的个体取得胜利 (确定性的或者是概率的)。这种选择方法有利于同时寻找多个局部极值点，在多峰函数 (Multimodal Function) 优化中得到广泛应用。对于这种算法我们将在第七章详细介绍。

## 2.3.2 GA 中的变异算子

在 SGA 中，变异算子是一个按照一定概率  $p_m$  实施的操作，目的是保持群体中基因的多样性。在二进制编码中，基本的变异方式就是将某一位或几位进行翻转。

形式化的描述如下：

设父代的二进制编码为  $j$ ，存在一个变异模板  $m$ ，变异后的结果为

$$U(j) \oplus j \oplus m \quad (2.8)$$

具体操作上，存在两种典型的变异方式：1-比特变异和  $c/l$  变异。

1-比特变异指按照变异概率确定待变异的个体后，均匀的从中选取一位进行翻转。也就是模板中只有一位是 1。

$c/l$  变异指对于某个个体，每一位都按照概率  $p_m/l$  确定是否翻转。一次变异有可能改变多个位。

虽然这两种方式实现细节不同，但这两种方式都具有遍历性，也就是从某个特定的基因型出发，经过有限次操作可以达到任意的另一个基因型。这种特性对于后面介绍的 Markov 模型收敛性证明是必不可少的。

### 2.3.3 GA 中的交叉算子

交叉算子的具体形式和基因的编码方式密切相关。对于位串编码，最简单的交叉算子是单点交叉，也有复杂一些的多点交叉和均匀交叉算子。

单点交叉的实现如下：随机地在两个父本上选择一个交叉点，然后交换这两个串对应的子串，得到的子代分别由父本的连续的一部分构成。多点交叉则是随机生成多个交叉点，然后间断交换父本中对应的子串。均匀交叉则是依概率交换串中的每一位。

对于这三种算子，可以使用统一的形式化描述为：

设两个父本为  $j_1, j_2$ 。存在一个交叉模板  $m$ ，交叉后的子代分别为：

$$s_1 ? (j_1 ? m) ? (j_2 ? \bar{m}), s_2 ? (j_1 ? \bar{m}) ? (j_2 ? m) \quad (2.9)$$

上式中， $?, ?, \bar{m}$  分别表示二进制上的与、异或和补运算。

如果  $m$  的结构为在某位之前都为 1，其后都为 0（或相反），则上式表示单点交叉；若  $m$  由连续的 0 串和 1 串混合而成，则表示多点交叉；而  $m$  的一般形式代表均匀交叉。

从模式的角度来看，多点交叉和均匀交叉能够搜索到的模式更多，具有更强的搜索能力。但实践证明，多点交叉和均匀交叉的作用并不显著。

对于其他类型的编码方式，可以设计出不同的交叉算子。在实际使用时，为避免致死基因，也需要对交叉算子进行改进。实际上，很多特定领域中使用的 GA 都要为之设计包含领域知识的交叉算子。

## 2.4 小结

本章中介绍了进化算法的演化历史和基本结构，然后详细介绍了遗传算法的结构和诸算子的构造方式。在应用遗传算法时，需要仔细考虑这些问题，包括确定编码方案、适应度函数、进化算子和收敛条件等等，然后确定算法的控制参数。这些过程往往是经验性的和试验性的，需要在实际运行中不断手工调整。对算法进行理论分析的目的之一就是从理论上分析出各种方案的好坏，为实际使用提供指导。

## 第三章 进化算法的理论分析方法

进化算法和其他问题求解领域一样，最主要的理论问题就是：“算法是否能够得到优化解”。这个问题难以得到无条件的肯定的回答。于是，问题转化为“在何种条件下，可以保证算法得到优化解”，以及“这种保证的类型”。实际上，由于进化算法的本质随机性，算法只能保证以概率 1 得到优化解。或者说，算法的几乎收敛(almost sure convergence)，概率收敛(convergence in probability)，或者平均收敛(convergence in mean)都是一个含义。

从纯粹理论的观点来看，以概率 1 收敛也是可以满意的。从实际来说，收敛速度更加重要。算法有效性的一个重要度量就是获得优化解的期望步数。在 EA 中，适应度评估函数的计算次数就是评价有效性最通用的量度。

EA 的基本功能是执行优化过程。很多其他任务，如机器学习，约束满足问题等都可以转化为优化问题。然而，将 EA 只理解为一个优化器是一种狭隘的观点<sup>[39]</sup>。从更宽的范围来看，EA 是一个具有增加群体的平均和最佳适应度的这种本能(basic instinct)的自适应系统。严格说来，EA 中最重要的是优化过程，而不是优化器。从这种观点出发，群体动力学是理论探索的典型问题，例如，群体基因分布随时间的变化就是一个重要问题。又如在 GA 中，经常出现群体收敛到一个相对的好解，而不是最优解，这种早熟收敛的分析就是 GA 领域中的一个本质问题。

EA 的理论研究有两层含义。一方面，它包括一般性的、支持所有 EA 家族的进化机制的理论。对候选解群体进行一般性的搜索过程，然后经历了繁殖和选择的算法，使用 Markov 模型，已经得到了几个收敛结果，它们属于这个范畴（虽然有些在出版时是针对 GA 的，但可以很容易地扩展到 EA 的一般范畴）。在另一方面，它指对于 EA 中特定类别的理论研究。一个广为人知的例子就是 GA 中的模式理论。

当然，随着研究的深入，还会产生其他的一些理论问题。但在当前，可以确定地说，进化算法的极限行为、运行速度、动力学行为是进化计算

理论的关键主题<sup>[42]</sup>。

遗传算法模型的建立<sup>[35,66,94,97,127,131]</sup>

遗传算法的收敛性分析<sup>[5,36,41,79,80,81,99,113,114]</sup>

遗传算法的复杂性分析<sup>[5,29,96]</sup>

算法的动力学行为分析<sup>[23,24,25,59,94]</sup>

适合遗传算法的问题的特征的研究<sup>[51,58,68,82]</sup>

对于 GA 的理论分析来说，算法的建模是基础，收敛性分析是一个基本问题，复杂性分析是算法得到应用的保障。在本章中对于一些在文献中使用的理论分析方法和理论结果作一个初步的介绍。对于本文中主要使用的动力系统模型，只给出一些数学上的准备知识，其建模方法在以后的章节中说明。

### 3.1 收敛性的含义

算法的极限行为，也就是通常所说的算法的收敛性和收敛条件，是 EA 的理论分析中最根本的问题。在 EA，尤其是 GA 领域，收敛性的含义与通常在数学中的含义并不完全一致。在分析收敛性之前，首先要明确收敛性（Convergence）的概念。

一般来说，一个系统的收敛指系统运行进入不随时间变化的状态。对于随时间变化的系统状态函数  $F(t)$  来说，即存在正数  $T$ ，对于所有  $t > T$  时，有  $F(t+1) \approx F(t)$ 。或者指系统以极限方式趋于一个固定的状态，即

$$\lim_{t \rightarrow \infty} F(t) = F_s。$$

在 GA 的运行中，上面所说的收敛性，即是算法运行到一定次数后，群体中所有个体均维持不变的状态。这种收敛并不意味着群体中所有的个体均相同。但在 GA 的使用中，我们更关心的是这样一种状态，群体中所有的个体具有一致的基因型，并维持不变。我们这种状态为强收敛；相对的，上面所说的状态称之为弱收敛。也就是说，“收敛到不变状态”与“收敛到一致状态”是两个不同的概念。在关于 GA 的收敛性研究中，一般都指强收敛。在本文中，在不引起歧义的情况下，强收敛也简称为收敛。

对于实际问题，我们希望 GA 能够寻找到目标函数的全局极值点。当 GA 进入强收敛状态，并且所有个体均为全局极值点时，称为全局收敛。如果收敛到的目标函数的非全局极值点，称为局部收敛。从实际中可以发现，GA 常常不能全局收敛。因此研究算法全局收敛的条件具有重要的应用价值。

综上所述，为得到妥善的收敛理论，需要逐次证明以下几点。

1. 算法的弱收敛性和收敛条件
2. 算法的强收敛性和收敛条件
3. 算法全局收敛的条件。

## 3.2 GA 建模使用的数学方法

在进行理论分析之前，需要建立适当的数学模型。这个过程叫做建模。由于进化算法是一个新兴的交叉学科，在成长过程中从生物学、应用数学、物理学等学科吸取了很多营养。它的理论分析也涉及到了数学中的很多分支。在介绍这些建模和分析方法之前，需要了解一下必需的数学知识。一般说来，对于有限群体模型，由于 GA 的随机性，会涉及许多概率论与随机过程的知识；对于无限群体，还要了解动力系统的知识和方法。

在这里简单介绍一下随机过程和动力系统的一些准备知识。这里列出的定义和定理可以在介绍概率论、线性代数、随机过程和动力系统的书中找到<sup>[31,137]</sup>。

### 3.2.1 随机变量

以下关于概率论和随机过程的定义和定理可以在

**定义 2.1** 如果某个变量  $\xi$  的取值随偶然因素而变化，但又服从一定的概率分布规律，称之为随机变量。

这里给出的定义是一种直观的说明，严格的定义可以由概率公理给出，或者使用测度论的方式描述<sup>[31]</sup>。

给定随机变量  $\xi$ ，可以用  $F(x) = P(\xi \leq x)$  定义的分布函数来刻画它的概率规律。分布函数需要满足单调性、右连续性和边界性。

**定义 2.2** 如果随机变量  $\xi$  只取有限个值或在可数集合上取值，称之为离散型随机变量。如果其分布函数能够表示为  $F(x) = \int_{-\infty}^x p(t)dt$ ，则称  $\xi$  为连续型随机变量。

**定义 2.3** 若  $(\xi_1, \xi_2, \dots, \xi_n)$  是联系于同一组条件下的  $n$  个随机变量，则称之为  $n$  维随机变量。若  $(x_1, x_2, \dots, x_n)$  是实数空间  $R^n$  中的一点，则事件  $(\xi_1 \in x_1, \xi_2 \in x_2, \dots, \xi_n \in x_n)$  的概率称为这个随机变量的联合分布。如果从上面  $n$  个变量中取出  $m$  个分量构成的  $m$  维随机变量，则称这  $m$  维随机变量的联合分布函数为  $m$  维边缘分布函数。

**定义 2.4** 设  $\xi$  为一随机变量，事件  $B$  的概率  $P(B) > 0$ ，则称  $F(x|B) = P(\xi \leq x|B)$  为  $\xi$  在事件  $B$  已经发生的条件下的条件分布函数。

随机变量  $\xi$  的数学期望（或均值）记为  $E\xi$ ，它描述了随机变量的取值中心。随机变量  $(\xi - E\xi)^2$  的数学期望称为  $\xi$  的方差，记为  $D\xi$ ， $D\xi$  的平方根称为  $\xi$  的均方差（或标准差），它们描述了随机变量的可能取值与均值的偏差的疏密程度。

对于连续型随机变量：

$$\begin{aligned} E\xi &= \int_{-\infty}^{\infty} x dF(x) = \int_{-\infty}^{\infty} x p(x) dx \\ D\xi &= \int_{-\infty}^{\infty} (x - E\xi)^2 dF(x) = \int_{-\infty}^{\infty} (x - E\xi)^2 p(x) dx \end{aligned} \quad (2.1)$$

对于离散型随机变量：

$$\begin{aligned} E\xi &= \sum_{k=1}^{\infty} x_k p_k \\ D\xi &= \sum_{k=1}^{\infty} (x_k - E\xi)^2 p_k \end{aligned} \quad (2.2)$$

$$期望与方差之间满足 D\xi = E\xi^2 - (E\xi)^2 \quad (2.3)$$

当  $r \geq 0$  时，随机变量  $\xi^r$  和  $(\xi - E\xi)^r$  的数学期望（假设存在）分别称为随机变量  $\xi$  的  $r$  阶原点矩和  $r$  阶中心矩。特别的，均值是一阶原点矩，方差是二阶中心矩。

### 3.2.2 矩阵初步

在关于 Markov 模型和动力系统研究中，都要使用很多关于矩阵的概念和定理，下面作一初步介绍。

**定义 2.5** 设  $A$  是一个  $n$  阶方阵，若对于矩阵中任何元素都有  $a_{ij} \geq 0$ ，则称  $A$  是非负的，记作  $A \geq 0$ ；若任何元素都有  $a_{ij} > 0$ ，则称  $A$  是正的，记作  $A > 0$ 。

**定义 2.6** 设  $A$  是一个  $n$  阶非负矩阵，

- (1) 如果存在正整数  $k$ ，使得  $A^k$  是正的，则称  $A$  是本原的(primitive)。
- (2) 如果存在方阵  $C, T$ ，通过相同的行列置换可以将矩阵变换为如下形式：

$$\begin{pmatrix} C & O \\ R & T \end{pmatrix}$$
，则称  $A$  是可约的(reducible)，否则称  $A$  是不可约的。

- (3) 如果对于任意一行，都有  $\sum_{j=1}^n a_{ij} = 1$ ，称矩阵  $A$  是随机的(stochastic)。

容易验证，随机矩阵的乘积还是随机矩阵。

**定义 2.7** 设  $A$  是一个  $n$  阶方阵，如果存在一个  $n$  维非零向量  $a$  和一个数  $\lambda$ ，使得  $Aa = \lambda a$ ，则称  $\lambda$  为  $A$  的特征值(eigenvalue)， $a$  为此特征值对应的特征向量(eigenvector)。

特征值和特征向量满足以下性质：

- (1) 设  $A$  的  $n$  个特征值分别为  $\lambda_1, \lambda_2, \dots, \lambda_n$ ，则它们的和等于  $A$  的迹，它们的积等于  $A$  的行列式，即

$$\sum_{i=1}^n \lambda_i = \text{tr} A = \sum_{i=1}^n a_{ii}, \quad \prod_{i=1}^n \lambda_i = |A|$$

- (2) 实对称矩阵的特征值都是实数，并且有  $n$  个线性无关，正交的特征向量。

- (3) 矩阵的特征值在相似变换下不变。

**定义 2.8** 若实对称矩阵  $A$  的所有特征值都大于零，则称之为正定(positive definite)矩阵，若所有特征值都不小于零，称之为半正定的(positive semidefinite)。

**定义 2.9** 若矩阵  $A$  的对角线以下的元素都为 0，则称  $A$  为上三角矩阵。

上三角矩阵的和、差、积以及数乘的结果仍然是上三角矩阵。同时，上三角矩阵的特征值就是其对角线元素。

### 3.2.3 随机过程

若对于每个  $t \in T$  ( $T$  是某个固定的指标集合),  $X(t)$  是个随机变量, 则把这样的随机变量族称为随机过程。

如果  $T$  是整数集或其子集, 则称之为离散时间的随机过程, 也称为随机序列。如果  $T$  是实数集或其子集, 则称之为连续时间的随机过程。

**定义 2.10** 对于  $T$  是非负整数集合的随机序列  $\{X_n\}_{n \in T}$ , 如果满足,

$$P(X_n = j | X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) = P(X_n = j | X_{n-1} = i_{n-1})$$

则称此随机序列为马尔可夫链 (Markov Chain)。

Markov 链描述的随机过程, 可以直观地理解为, “将来” 的状态只取决于 “当前” 的状态, 而与 “过去” 的状态无关。

用  $p_{ij}(t_1, t_2)$  表示已知在时刻  $t_1$ , 系统处于  $E_i$  的状态下, 在时刻  $t_2$  系统处于状态  $E_j$  的概率。也叫做转移概率。

若转移概率只与两个时刻的差有关, 而与具体时刻无关, 即  $p_{ij}(t, t + \tau) = p_{ij}(\tau)$ , 称此 Markov 链为齐次的 (homogeneous)。否则称为非齐次的。

对于齐时 Markov 链, 一次转移概率可以排列为一个概率转移矩阵

$$P = (p_{ij}) = \begin{pmatrix} p_{00} & p_{01} & \dots \\ p_{10} & p_{11} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$n$  步的转移概率矩阵是一步转移概率矩阵的  $n$  次幂。

#### 定义 2.11

称状态空间  $E$  的一个子集  $C$  为闭集, 若  $C$  以外的任意状态都不能从  $C$  中任意状态到达。若  $E$  中不存在真子集为闭集, 则称此链为不可约的。

#### 定义 2.12 (状态的分类)

对于状态  $E_j$ ,

- (1) 若单点集  $\{E_j\}$  为闭集, 就称之为吸引状态,
- (2) 若  $\sum_{n=1}^{\infty} p_{jj}^{(n)} = \infty$ , 则称此状态为常返的; 若小于  $\infty$ , 则为非常返的。



(3) 若此状态为常返的, 则当  $\lim_{n \rightarrow \infty} n p_{ij}^{(n)} > 0$  时, 称之为积极常返的, 否则称之为消极常返的。

(4) 若正整数集  $\{n | p_{ij}^{(n)} > 0\}$  的最大公约数  $t$  大于 1, 则称此状态是周期的, 周期为  $t$ 。否则称之为非周期的。

(5) 若此状态为常返的、非周期的, 则称之为遍历的。

**定理 2.1 (遍历性定理)** 对于不可约的, 非周期的常返的 Markov 链, 它是积极链的充分必要条件是存在唯一的平稳分布, 这个平稳分布也是链的极限分布, 并且各个分量严格为正。

### 3.2.4 动力系统

在十九世纪末到二十世纪初, Poincaré 等人从经典力学和微分方程的定性理论的研究中, 提出了动力系统的概念。从二十世纪七十年代开始, 这一学科在各个领域中得到广泛应用。如气象预报、统计力学、化学反应、生态问题、数量遗传学和经济数学等方面。下面简要介绍动力系统的一些概念<sup>[137]</sup>。

动力系统研究的目的是研究系统状态变量随时间变化的特性, 它主要考察变量的变化方式和最终特性, 如是否存在不动点、周期点或极限环, 它们是吸引的还是排斥的, 它们的吸引空间等。

简单地讲, 对于一个变换  $H: X \rightarrow X$ , 及其迭代构成了差分动力系统。更严格的定义如下:

**定义 2.13:** 设  $X$  是拓扑空间,  $f: X \rightarrow X$  是连续映射, 如果  $f$  满足:

$$f(0, x) = x, \forall x \in X$$

$$f(s + t, x) = f(s, f(t, x)), \forall s, t \in \mathbb{R}, x \in X$$

则称  $f$  为  $X$  上的连续动力系统。如果对此动力系统进行离散采样, 得到的序列称之为离散动力系统。

本文只涉及实数空间的离散动力系统, 即状态空间  $X \subset \mathbb{R}^n$ 。根据  $O$  的维数可分为一维或高维动力系统。

**定义 2.14** 如果  $H(x) = x$ , 则  $x$  是  $H$  的不动点。如果  $H^n(x) = x$ , 则  $x$  是周期为  $n$  的周期点。满足上式的最小正  $n$  称为真周期。

对于一维动力系统的不动点，如果其邻域的点经过变换收敛到此点，称之为吸引点。如果邻域的点都逃离此点，称之为排斥点。

**定理 2.2** 设  $p$  是  $f(x)$  的不动点，且导数  $|f'(p)| < 1$ ，则存在包含  $p$  的开区间  $V$ ，若  $x \in V$ ，则  $\lim_{n \rightarrow \infty} f^n(x) = p$ 。点  $p$  称为吸引不动点（又称吸引子）。

**定理 2.3** 设  $p$  是  $f(x)$  的不动点，且导数  $|f'(p)| > 1$ ，则存在包含  $p$  的开区间  $V$ ，使若  $x \in V, x \neq p$ ，则存在  $n > 0, f^n(x) \notin V$ 。点  $p$  称为排斥不动点（又称排斥子）。

对于高维动力系统中，存在类似的概念。

**定义 2.15（Jacobian 矩阵）**

设有定义在某一  $n$  维区域  $D$  上的关于  $n$  个自变量的  $n$  个函数：

$$\begin{cases} y_1 = f_1(x_1, x_2, \dots, x_n) \\ y_2 = f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ y_n = f_n(x_1, x_2, \dots, x_n) \end{cases}, \text{ 或简记为向量形式: } \mathbf{y} = \mathbf{F}(\mathbf{x})$$

并且关于自变量有连续偏导数。则由这些偏数组成的矩阵

$$D\mathbf{F}(\mathbf{x}) = \begin{vmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \end{vmatrix} \text{ 称为函数组的 Jacobin 矩阵。}$$

在高维动力系统中，这个矩阵的作用与一维系统中的导数的作用相类似。

**定理 2.4（多元函数微分的连锁法则）**

设有两个函数组  $\mathbf{y} = \mathbf{F}(\mathbf{x})$ ， $\mathbf{x} = \mathbf{G}(t)$ 。它们的连锁函数  $\mathbf{y} = \mathbf{F}(\mathbf{G}(t))$  的 Jacobin 矩阵等于两个函数组的 Jacobin 矩阵的乘积。

**定义 2.16** 若  $n$  维动力系统  $F: R^n \rightarrow R^n$  的不动点为  $p$ ，如果  $F$  在  $p$  点的 Jacobin 矩阵  $DF$  的所有特征值的绝对值都不为 1，称  $p$  为双曲不动点。定义  $p$  的类型如下：

若所有特征值的绝对值都小于 1，则  $p$  是吸引不动点。

若所有特征值的绝对值都大于 1，则  $p$  是排斥不动点。

若有些特征值的绝对值大于 1，有些小于 1，则  $p$  是鞍点。

**定理 2.5** 设映射  $F$  在  $p$  点是吸引不动点，那么在  $p$  处有一开集，其中一切点在  $F$  的迭代下趋于  $p$ 。

**定理 2.6** 设映射  $F$  在  $p$  点是排斥不动点，那么存在一个包含  $p$  的开集，其中一切点在  $F$  的逆向迭代下趋于  $p$ 。

**定义 2.17** 如果映射  $F: X \rightarrow Y$  是一对一的，满的，连续的，则称之为同胚(homeomorphism)。

**定义 2.18** 称两个同胚  $f: X \rightarrow X, g: Y \rightarrow Y$  是拓扑共轭的，如果存在同胚  $h: X \rightarrow Y$ ，使得  $h(f(x)) = g(h(x))$ 。

拓扑共轭的两个系统，具有相同的不动点和轨道结构，因此可以通过研究某系统的拓扑共轭系统来研究原来的系统。

**定理 2.7 (Hartman 线性化定理)** 设动力系统  $F$  以 0 为双曲不动点，由  $F$  在 0 点的 Jacobin 矩阵定义的线性映射为  $DF$ ，则存在 0 的开邻域  $V$ ，使得  $F$  和  $DF$  在  $V$  内拓扑共轭。

**定理 2.8 (不稳定流形定理)**

若  $n$  维动力系统  $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$  存在一个鞍点  $p$ ，在鞍点处其 Jacobin 矩阵  $DF$  的特征值中有  $k$  个小于 1。必存在  $\epsilon > 0$  和一定义在鞍点附近的  $n-1$  维光滑流形  $\Sigma: \mathbb{R}^k \rightarrow \mathbb{R}^n$ ，流形可以由含参数  $t$  的参数方程来表示。使得：

1.  $\Sigma(0) = p, r'(t) \neq 0$
2.  $r'(0)$  是  $DF(p)$  的一个不稳定的特征向量
3. 在  $F^{-1}$  作用下， $\Sigma$  不变
4. 当  $m \rightarrow \infty$  时， $F^{2m}(r(t)) \rightarrow p$

这个定理的含义指在鞍点附近存在一个不稳定流形，流形上的点经过变换后仍然处于此流形，上面的一切点在  $F^{-1}$  迭代下都趋向此不动点。此定理对于稳定和不稳定流形具有全局性的对等概念。在鞍点处同样也存在一个稳定流形，其维数等于 Jacobin 矩阵的特征值中大于 1 的个数，流形上的点经过正变换后趋向于鞍点。

在高维动力系统中，稳定流形和不稳定流形将状态空间划分开来，确定各自的收敛目标。

高维动力系统的全局形态是相当复杂的，除了不动点的三种分类外，还可能存在周期点（甚至是无限多个），极限环，以及混沌等。

### 3.3 GA 建模的非 Markov 方法

在这里简要地列出 GA 建模的一些非 Markov 方法，同时也列出了一些根据这些建模方法而改进的方法。这些方法中的大多数考虑二进制串的搜索空间，每一个串作为等位基因（alleles）的长度为  $l$  的向量。

在第一节中回顾了和说明了模式理论；第二节说明了从物理学中借来的一种方法，估计群体适应度分布的进化；第三节研究等位基因的边际分布的进化。这种方法在最近 5 年中得到更多的注意，产生了很多改进的算法。这些算法不能保证找到全局极值点，但可以比 SGA 收敛得更快。

#### 3.3.1 模式进化

关于 GA 的最开始的探索是基于模式的进化，集中在算法的宏观视角。模式定理最初在<sup>[71]</sup>中证明，然后在<sup>[54]</sup>中得到扩展。

仍考虑以下基本 GA：二进制空间  $E=\{0,1\}^l$ ，使用比例选择，单点交叉和  $1:l$  变异。

模式（Scheme） $H$  被定义为  $E$  中的一个子集，其中某些位的值可以不确定。模式中取确定值位置的数目称为模式的阶，记作  $o(H)$ 。 $?(H)$  为模式中第一个取确定值的位置和最后一个取确定值的位置之间的距离，称为模式的定义长度。

**定理 3.1（模式定理）**<sup>[54,71]</sup>：

设标准 GA 的变异概率为  $p_c$ ，交叉概率为  $p_m$ ， $N(H,n)$  为群体在第  $n$  代时模式  $H$  所代表的元素的数目，则

$$E(N(H,n+1)) \leq N(H,n) \frac{f(H,n)}{f(t)} [1 - p_c \frac{?(H)}{l+1} - p_m o(H)] \quad (3.1)$$

模式定理的含义是：在群体中，定义长度较短的，低阶的，且适应值超过平均适应度的模式  $H$ ，在群体中的数目的期望值在下一代中增长。

在近似无限群体时，模式所代表的数目的期望可以用它的实际值所替换。一个明显的结论是大于平均适应度的模式试图扩展到整个群体。

模式理论的有效性已经遭到严重的质疑<sup>[6,26,34,66,93,105,126,134]</sup>。首先，由于群体的平均适应度每代都在变化，模式定理并没有表达出这种变化的影响。因此它只在一代中有效，不能多次迭代，不能用来研究长期特性。模式理论不能解决 GA 的动力学行为和极限行为。其次，它隐含的假设问题可以被分解为几个片断，对于一般性的适应度函数，这种假设不能成立。目前，为克服这种限制，一些学者提出了“构造块假设”（Building Block Hypothesis）<sup>[4,120,121,128]</sup>，将模式通过交叉和变异而重建的影响考虑在内，提供了模式和进化之间新的联系，试图解释 GA 的运行机理。对于一些简单的适应度函数，如 BinInt 问题等，这一方面的研究取得了一定的成果<sup>[83,124]</sup>，但仍难以得到一般性的结论。

目前来看，使用模式理论难以导出任何强有力的收敛性理论结果，也无法得出算法的计算复杂性。

### 3.3.2 群体适应度的累积度

在统计力学的公式中，群体是用几个宏观的度量来描述的：群体磁化 (magnetization) 分布的累积度 (cumulant)，又称为矩<sup>[102,106]</sup>。矩包括平均值  $K_1$ ，方差  $K_2$ ，以及其他更高阶的矩。一个串的磁化定义为：

$$M(x) = \sum_{i=1}^l x_i \quad (3.2)$$

对于大部分问题来说，磁化值分布与适应度分布没有关系。但对于一些特殊的例子（如 One-Max 问题），磁化值等于适应度值。这样很多人研究群体矩的期望的进化，发现了在使用变异、选择或交叉算子时，这些矩的期望的一些迭代关系。文献<sup>[102]</sup>详细回顾了这些应用和简化假设。

在通常的假设下，只有 2 阶、3 阶或  $n$  阶矩是非空的。当使用强度为  $\beta$ （较小）的 Boltzman 选择作用于  $N$  个个体时，矩的变化公式（ $\beta$  更高阶的作用被忽略）为：<sup>[102,104]</sup>

[illegible]

当对所有的群体成员，以概率  $c/l$  使用变异算子时，公式为：

$$\begin{aligned} & \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} (1 - 2c) \frac{1}{2} \frac{1}{2} \frac{1}{2} \\ & \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} (1 - 2c)^2 \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} (1 - (1 - 2c)^2) \end{aligned} \quad (3.4)$$

这里， $\langle \rangle$  符号表示在无限群体上所有个体的矩的期望。

组合这些公式，对于 One-Max 问题，找出一个在选择、变异算子（甚至包括交叉算子）作用下的矩变化公式就是随理成章的了。但对于更复杂的问题，这种公式是不适用的，可能需要使用数值仿真的方法来获取经验公式。这种近似的模型已经扩展到其他的繁殖和选择方法，在一些情况下与实际 GA 多次运行的平均特性相当接近<sup>[102,111]</sup>。

这种方法也用来研究其他不同于 One-Max 的函数，但在很多情况下，适应度是磁化值的函数(或者是和唯一的极值点之间的 Hamming 距离)<sup>[119]</sup>

以一些近似为代价，统计力学模型使我们在一些适应度函数上获得了有意义的观点。在当前阶段，使用精确模型（如 Markov 链）很难获得这些结果。但这种方法难以扩展到一般的适应度函数上。

### 3.3.3 等位基因边际分布

等位基因边际分布 (Allelic Marginal Distributions) 的方法通过构造一个辅助的随机算法, 具有比 GA 更小的收敛时间来研究 GA 的理论特性。

下面要讨论的模型最先在实际数值计算中被提出<sup>[10]</sup>。选择不再变异的群体，进行配对，然后将等位基因边际分布更新如下：

$$p_i(x_i, t \neq 1) \neq p_i(x_i, t) \neq (p_i^s(x_i, t) \neq p_i(x_i, t)) \quad (3.5)$$

这里,  $p_i^s(x_i, t)$  是被选择的群体成员的等位基因边际频率。然后按照如下的概率描述, 通过拖动  $N$  个串构造新的群体。

$$p(x, t \geq 1) \leq p_i(x_i, t \geq 1) \quad (3.6)$$

这种算法的收敛速度严格依赖于  $\alpha$ 。对于  $\alpha=0$ ，是随机搜索；对于  $\alpha=1$ ，

称之为 UMDA<sup>[95]</sup>。Mühlenbein 在一个无限群体的框架内，使用了比例选择和交叉算子，但没有使用变异算子，从而形式化地描述了 UMDA。Mühlenbein 说明了这种算法构成了在势能函数  $W(t) = \sum_x p(x,t)f(x)$ （也就是平均适应度）上的梯度搜索。

$$p(x_i, t+1) = p(x_i, t) \frac{p(x_i, t)(1 - p(x_i, t)) \frac{\partial W / \partial p(x_i)}{W(t)}}{W(t)} \quad (3.7)$$

这里  $p(x_i, t)$  表示在  $t$  时刻  $x_i = 1$  的概率。但是，这个梯度是不容易直接实现的，因为需要在  $2^l$  个参数上的知识。

为了直接搜索分布空间，提出了另外一种减少分布空间复杂性的方法<sup>[16]</sup>。这可以通过约束搜索到一个参数分布族上，允许在参数空间进行简单搜索。这里，空间  $E$  上的向量  $x$  被由  $l$  个随机变量组成的向量替代。这些随机变量带有参数化的密度分布。

$$p(x, \theta) = (p_1(x_1, \theta_1), \dots, p_l(x_l, \theta_l)) \quad (3.8)$$

这里  $p_i(\theta_i)$  定义了等位基因  $x_i$  在  $\{0,1\}$  上的分布，由参数  $\theta$  决定。例如，文献<sup>[16]</sup>中的数值仿真假定  $p_i(\theta_i)$  服从独立的均值为  $\theta$  的 Bernoulli 分布，或者是均值和方差由  $\theta$  给出的 Gaussian 分布。

这样，空间  $E$  的离散优化问题变成了 Kullback-Leibler 散度场（divergence）上的在  $p$  和吉布斯（Gibbs）分布  $p_T^*$  之间的连续最小化问题，这里，将空间  $E$  上的高适应度状态对应于低的温度  $T$ 。可以证明，系统  $F(\theta)$  在自由能量下可以通过梯度达到最小。但对于上面的  $W(t)$ ，计算最小能量仍然需要  $2^l$  个参数上的知识。

为了使这种梯度最小化计算有意义，文献<sup>[16]</sup>中提出了  $\theta$  的一种估计的，随机的梯度下降更新原则，允许算法在多种适应度函数和不同温度下成功地使自由能量最小化。但是，由于这种估计性，在给定适应度上，全局极值点的渐近收敛特性难以达到。

### 3.4 使用 Markov 链的模型

不带经验估计的收敛性和极限行为的主要理论结果是使用 Markov 模型得到的。下面分别介绍齐次和非齐次的模型。这种模型一般是建立在有

限群体的基础上的，与实际计算情况比较接近。

### 3.4.1 有限搜索空间上的齐次模型

很容易想到，在每一步，由当前群体作为 GA 的状态来建造 Markov 链。

若状态空间是  $E^N$ ，则状态空间为  $S \subset B^{Nl}$ ，此处  $l$  是基因的长度， $N$  是群体的规模。状态空间中的每一个元素可以看作是一个长度为  $N \times l$  的二进制串表示的整数。由于 GA 的其次性，转移矩阵是一个  $N \times l$  的方阵。

在每一次迭代，选择  $N$  个个体，构成一个群体（选择后群体）。选择的每一个个体进行变异操作。对于任意  $n > 0$ ，可以计算转移概率：

$$P(x_{n+1} | x_n = y) = \prod_{p=1}^N \prod_{i=1}^l \frac{f(y^i)}{\sum_{j=1}^l f(y^j)} K(y^i, x^p) \quad (4.1)$$

$K(y, x)$  给出了从  $y$  变异到  $x$  的概率。

一般的，可以将 SGA 的转移矩阵分解为三个随机矩阵：交叉矩阵  $C$ ，变异矩阵  $M$  和选择矩阵  $S$  的乘积。

很多论文，对于二进制空间和广泛使用的 positive  $1/l$  变异，应用 Markov 链模型，证明了一些收敛性定理。

**定理 4.1** <sup>[98]</sup>：SGA 是一个遍历的 Markov 链，不能以概率收敛。也就是说，存在与初始分布无关的极限分布，使得在任何时刻从一个状态转移到另一个状态的概率不为 0。

另外，我们有以下关于 Markov 链著名的定理，证明了改进的 SGA 以概率 1 收敛到优化状态。

**定理 4.2** <sup>[69]</sup>：如果群体最优适应度序列对于时间是单调的，并且  $E$  中的任意点在有限步内通过变异和重组操作可以到达，则：

$$P(\lim_{n \rightarrow \infty} x_n^* = x^*) = 1$$

注意到，这种单调性可以很容易通过使用精英(Elite)选择策略或其他能够保证每一代中最优个体存活下去的策略来实现。例如典型的精英策略是保留群体中当前的最优解；还可以采取的一种方法是：在每一代中，通过任意选择策略选择一个个体，变异与交叉，然后替换群体中最坏的个体。



上述两个定理的满足条件相类似，其证明也经历了一个发展的过程。Rudolph 的收敛性结果<sup>[113,114,115]</sup>依赖于变异算子矩阵的正定性（空间中任意两点之间的变异概率是非零的）；由于上述假设很强，Agapie 扩展了上述结果<sup>[2]</sup>，要求变异矩阵是可约的，对角化正定的（diagonal positive transition）。所有这些作者都强调单调性条件，一般是通过在进化过程中维护和更新群体中最好的个体来实现。换句话说，当使用遍历性的 GA 时，迭代过程中得到的最优适应度值几乎逼近于适应度函数的优化点。另一方面，Agapie 证明了对于使用比例选择，可约的变异算子的齐次 GA，以概率 1 收敛的必要条件是满足单调性。

在上面的证明中，变异算子是所有收敛性结果的核心。不可约的变异算子通常意味着在有限时间内所有状态的可到达性。可以直观地认为，正是由于所有状态之间都可以通过变异算子来相互转移，因此整个 SGA 群体的状态不能够固定下来。如果忽略变异算子，SGA 的极限行为就无法使用上述定理来描述了。

对于很多在无限的实值搜索空间中运行的进化算法（如进化规划和进化策略），也存在相似的理论。实际上，这些算法可以看作是全局随机搜索算法的一些具体实现，而全局随机搜索算法是以概率 1 收敛到优化状态的<sup>[138]</sup>。文献<sup>[26]</sup>中讨论了连续空间算法和一致全局随机搜索算法之间的联系。

最近，几乎所有的使用齐次模型的收敛性结果都证明了，群体以概率 1 进入一个优化集合。另一方面，非齐次参数的 GA 可以首先证明，几乎所有群体成员都趋向于优化解，并得到了对于最优集合的有限访问时间。

### 3.4.2 有限搜索空间上的非齐次 GA

同样是在 Markov 的框架中，使用类似模拟退火<sup>[125]</sup>的收敛性类似的方法也得到了其他的一些结果。GA 被看作是由选择操作确定的动力系统，而由变异和交叉操作对此带来随机混淆(perturbation)。当时间趋于无限时，这些混乱的比例趋于 0。

Davis 提出了初步的结果<sup>[36,37]</sup>。将群体看作一个描述每一个个体的出现数目的向量，证明了非齐次 GA 能够收敛到一个确定的个体。最近，在 Davis 的模型上，证明了优化状态的渐进（asymptotical）收敛性<sup>[122]</sup>。

Cerf 使用 Markov 模型，将群体看作  $N$  个个体的向量，大大扩展了上述结果<sup>[28]</sup>。

Cerf 使用动力系统中随机混乱的 Friedlin-Wentzell 理论，获得了群体尺寸的下界，从而证明了非齐次 GA 的有限时间收敛结果。他的工作主要基于变异算子。不过，他说明了交叉算子的引入将加速收敛性，同时群体尺寸的下界也要减少。

Cerf 的收敛结果成立的条件包括下面两个前提：

1. 变异为  $y$  的概率由一个的不可约的核所控制，并按指数衰减。即：

$$p_k(x, y) = \begin{cases} k^{-a} (x, y) k^{-a}, & \text{if } x \neq y \\ \sum_{z \in E, z \neq x} a_k(x, z), & \text{if } x = y \end{cases} \quad (4.2)$$

2. 选择概率由下式所控制：

$$p_k(y_i, ?) = \frac{\exp(cf(y_i) \ln(k))}{\sum_{j=1}^m \exp(cf(x_j) \ln(k))} \quad (4.3)$$

$p(x, y)$  是  $E$  上的不可约的内核。 $k$  是参数，趋于无穷，所以级数  $\sum_{t=0}^{\infty} k(t)^{-a}$  的收敛半径满足  $\frac{1}{2} < \frac{1}{2}$ ，上下界由适应度函数决定。第一个条件保证了变异概率趋于零，第二个条件说明了选择压力逐渐变大。

最后，Cerf 给出了一个函数  $N^*$ ，当  $N \geq N^*(f, a, c)$  时保证下面结果成立。

$$P(x \in E^N, \lim_{n \rightarrow \infty} P([?_n] \mid f^* | ?_0 \ni x) = 1) = 1 \quad (4.4)$$

$$P(x \in E^N, P(?N, ?n \in N, [?_{T_n}] \mid f^* | ?_0 \ni x) = 1) = 1 \quad (4.5)$$

这里  $T_n$  是第  $n$  次访问具有相等的适应度集合的时间。上面的一个公式保证优化状态集合的概率 1 吸引力。第二个公式保证对于优化状态群体的循环访问时间是有限的。

文献<sup>[52]</sup>中提出了另一个非齐次的算法。将群体向量中的前  $K(n)$  个点进行随机变异，其他的使用每一代中的优化点来代替。收敛性的证明依赖于 Perron-Frobenius 定理，证明了如果  $K$  服从参数以  $p \sim n^{1/D}$  的速率（ $D$  表示变异图的直径）趋近于 0 的二项分布，以及群体的尺寸足够大（一个充分的条件是  $N > D$ ），则

$$P(x \in E^N, \lim_{n \rightarrow \infty} P([?_n] \mid f^* | ?_0 \ni x) = 1) = 1 \quad (4.6)$$

因为混乱的降低速率很低，需求的群体数量太大，这些非齐次的 GA 的收敛性结果只有理论上的意义。同时由于这种方法与一般的 GA 在算法实现上有不少差距，因此并不能代表传统意义的 GA。如果在适应度函数方面的作一些假设，可以继续发展这些结果。

### 3.5 独立算子的收敛性结果

在任何进化算法中，由于选择和进化算子（交叉和变异）之间强烈的相互作用，尤其是交叉算子的二次性，导致一般建模的复杂性。为避免这种影响，简单的方法是从独立考虑每个算子开始。这些方法包括：没有进化算子时选择算子的取代时间（Takeover Time）的研究<sup>[59,30]</sup>；没有选择算子时交叉或变异算子的创新时间（Innovation Time）的讨论<sup>[56,123]</sup>；以及由于变异算子或交叉算子的随机漫游（Random Walk）导致的进化速率<sup>[40,53]</sup>。

#### 3.5.1 选择压力

考虑由  $N$  个成员构成的有限群体  $E^N$ ，假设群体中的一些成员  $\phi$  可能被选中多次来构造新的群体。这定义了一个初步的动力学过程，我们通常对达到群体中充满了  $N$  个同样个体的时间感兴趣。这个问题虽然在群体遗传学中得到广泛研究，但在遗传算法中，有关的方法和选择的过程与生物学有着显著的不同。在遗传算法中，我们对问题求解感兴趣，选择操作直接依赖于群体成员的适应度：高适应度的成员比低适应度的更能被选中。

早期的研究使用 Markov 链来计算期望吸收时间。使用这种方法，首先在空间  $E=[0,1]$  上，对比例选择算子计算出了期望吸收时间。但是，这里研究的空間是只有一位长度的二进制空間。

在同样的空間上，使用同样的方法，也已经得到了带有共享的选择算子的结果<sup>[72]</sup>。计算表明，共享选择算子的期望吸收时间远远大于标准选择算子。这也和使用不同的小群体和不同的  $f(1)/f(0)$  比例的仿真计算结果相一致。在这篇文章中，也对此提出了一些解释。由于吸收时间很长，作者研究了吸收态之前的准平衡(quasi-equilibrium)状态以及达到这种状态的期望时间。

同样，在 Markov 模型的框架下，在一些简化条件下，考虑在通常空间  $E=[0,1]^l$  中，Boltzmann 锦标赛选择也有很大的期望吸收时间<sup>[85]</sup>。由于这种选择方式在维护群体多样性方面具有与共享选择同样的效果，因此这个结果是情理之中的。

在 GA 中，另一种评价选择算子的方法是建立每一代中最优值所占比例  $p_n$ （对于无限群体）的递推公式。

在空间  $[0,1]$  中，文献<sup>[5]</sup>解出了这个递推公式，表明从从一个随机一致的群体开始，到达最优解比例为  $1-1/N$  的期望时间为

$$\ln(N-1)/\ln(f(1)-f(0)) \quad (5.1)$$

在无限群体的近似下，文献<sup>[59]</sup>比较了在连续空间  $E=[0,1]$  上不同的选择算子。结果如下，表示了从包含一个最优解的群体出发，最优解的比例达到  $1-1/N$  的期望时间。复杂性的范围从  $O(\ln N)$  到  $O(N \ln N)$ 。

选择算子	取代时间	备注
比例选择	$\frac{1}{c}(N \ln N - 1)$	$f(x) \propto x^c$
	$\frac{1}{c}(N \ln N)$	$f(x) \propto \exp(cx)$
线性排名 $r(f(x)) \propto c + 2(c-1)x$	$\frac{1}{\ln 2}(\ln N - \ln \ln N)$	For $c=2$
	$\frac{2}{c-1} \log(N-1)$	For $1 < c < 2$
p-锦标赛	$\frac{1}{\ln p}(\ln N - \ln \ln N)$	

表 3.1 选择算子的取代时间

上表中关于比例选择算子的取代时间计算存在一定问题，在本文的第六章将对此进行详细分析。

### 3.5.2 基因漂移

在群体遗传学的范畴，选择通常是相同作用在群体上的。群体中累进

的多样性丧失称为基因漂移 (genetic drift)。基因漂移的根本性原因是由于群体数目的有限性, 因此对于无限群体, 不存在这个问题。基因漂移的分析通常利用计算 Markov 转移矩阵链的吸收时间来获得。

计算基因漂移的另外一个方法是讨论在一致选择下的群体适应度方差的减少<sup>[102,103,106]</sup>, 在很多情况下已经得到了一些精确的分析结果<sup>[110]</sup>。每一代之间群体适应度方差的变化情况如下表。(第一个是精确的, 其他的是估计的, 准确到  $1/N$  阶)。

	变化率	
世代	$1-1/N$	独立一致选取
代间隔 $G$	$1-(2-G)/N$	父代子代重叠
CHC	$1-1/(2N)$	父代复制一份, 然后随机删除 $N$ 个

表 3.2 基因漂移的速率

上面的结果显示 CHC 迭代的基因漂移比典型的世代 (无重叠) 方法要慢, 这种方法又比世代重叠的慢。方差的变化率越小, 也就是基因漂移的速率越大, 群体由于规模有限导致的多样性丧失就越快。

### 3.5.3 随机漫游

如果不考虑选择算子, 群体状态的变化与适应度无关。从 Markov 模型的观点来看, 群体的每一代变化, 从当前状态以一定概率转变到算子空间中相邻的另一个状态, 构成了一个在算子邻域上的多维衍支过程 (branching process)。

对于变异算子, 在<sup>[53]</sup>中指出, 对于大的基因长度  $l$  值, 到达时间 (hitting time) 服从独立的指数分布。在二进制空间中, 存在两种典型的变异方式, 两种方式的到达时间分别为:

$$\frac{2^l}{N(1 - e^{-c})} \text{ 对于 } cN \text{ 变异 (每一位的变异概率是 } cN \text{)}.$$

$$2^l / N \text{ 对于 1-比特变异 (随机翻转一位)}.$$

这个结果表示, 广泛使用的  $cN$  变异要比 1-比特变异做的更好。

对于大规模的群体和  $cl$  变异, 整个算法的动力学可以描述为群体磁力值的分布。前面的方程(3.4)给出了每代平均值和方差之间的估计迭代公式。

下面考虑在交叉算子下的进化。在上一代中一致、独立的选出  $N$  个随机对, 然后交叉, 产生  $N$  个子女, 构成下一代群体。

无限群体的一般结果可以见文献<sup>[21]</sup>。证明了所有完全的(指交叉操作可以产生父代等位基因的任意组合)交叉模式导致同样的极限分布, 这个分布由有限群体的 Robbins 均衡分布给出。

$$p(x) = \prod_i p(x_i) \quad (5.2)$$

$p(x_i)$  是位置  $i$  处等位基因的常数边际分布。这个分布在交叉操作时不变。

最近, 文献<sup>[40]</sup>证明了有限群体中, 一个位置上可以取多个值的 GA 串在完全的交叉下, 也趋向于 Robbins 均衡分布。

在二进制模型下, 使用统计力学讨论有限群体的情况。对所有基于位的交叉取平均, 群体磁力的平均值和方差的迭代公式如下<sup>[102,103]</sup>:

$$\begin{aligned} \bar{x}_c &= \bar{x} \\ \sigma_c^2 &= (a^2 + (1-a)^2) \sigma^2 + 2a(1-a) \frac{l(1-q)}{4} \end{aligned} \quad (5.3)$$

$$q = \frac{1}{N(N-1)} \sum_{x \neq y} \left( \frac{1}{l} \sum_{i=1}^l (2x_i - 1)(2y_i - 1) \right) \quad (5.4)$$

$a$  是从父本  $x$  中选择子代的等位基因的概率,  $1-a$  是从另一个父本  $y$  中选择的概率。 $q$  是反映群体成员之间的相关性的参数。注意到交叉算子不改变群体磁力的平均值; 方差以一定速率 ( $a=0.5$  时最大) 向  $l(1-q)/4$  接近 (令  $\sigma_c^2 = 0$  可以解出)。

作为结论, 上面的这些初步的结果使我们了解到不同算子收敛性的边界, 但是这些结果既不能扩展到一般的同时使用变异、交叉和选择的 GA, 也不能对于一般的函数证明 GA 作为函数优化器的收敛性。

### 3.6 其他 GA 建模方法

进行理论分析使用的和可以使用的其他方法还有

### 1. 维量分析

虽然精确的 Markov 模型准确地描述了 EA，但只能得出有限的结论。因此人们发展了维量分析理论<sup>[60]</sup>。这种方法是从工程科学中借来的。维量分析试图确定复杂系统中的重要维量或者是关键特征，然后建立它们之间的函数关系。当应用到 EA 时，不同进化算子（选择，交叉，变异）的功能被放到一些由仿真试验估计出的函数中。也就是说，这些函数是一种“好的猜测”，但这些描述模型为更细致的研究提供了一些有益的线索。

### 2. 顺序统计量(Order Statistics)

对于凸(convex)的适应度函数，顺序统计量的理论提供了一种有力的工具来确定 ES 的收敛速率<sup>[18,116]</sup>。此外，如果群体规模足够大，这种方法与数量遗传学具有理论上的联系。

### 3. 数量遗传学(Quantitative Genetics)

初看起来，由生物学引发的 EA 可以使用理论生物学的结果。但问题是，在进化计算中的理论问题与理论生物学中的问题并不一样，因此这种应用需要一定的改变。一个例子是在文献<sup>[124]</sup>中使用了和数量遗传学中类似的方法来讨论基因漂移的影响。在文献<sup>[96]</sup>中，提出了一种进化算法。这种算法可以使用数量遗传学中的结果来进行分析。虽然这种应用被限制在可分离的的适应度函数和无限群体的条件上，它仍然是进化计算理论中的一个组成部分，将会得到不断发展

### 4. 正交函数分析(Orthogonal Functions Analysis)

像 Fourier，Walsh 和 Harr 这些正交函数变换，都可以用来分析进化算法的适应度函数，特别是揭示待优化问题的难度<sup>[17]</sup>。在这些函数中，Walsh 函数扮演了特别的角色，尤其是在下面提到的二次动力学系统中。

### 5. 二次动力学系统(Quadratic Dynamical, QDS)

二次动力学系统的典型应用是在物理学和生物学中处理不同的自然现象。在文献<sup>[129]</sup>和随后的文章中表明，当群体规模无限时，简单遗传算法可以转化为 QDS 问题。由于 QDS 的仿真是一个 PSPACE 完全问题<sup>[7]</sup>（PSPACE 完全问题是一个在确定型图灵机上，在多项式空间内可判定的语言类。各类复杂度问题的关系是， $P ? NP ? PSPACE ? NPSpace ? EXPTIME$ 。上式中至少有一个是真包含，虽然还没有证明，但大部分研究人员相信上面

的包含都是真的)，因此这种变换并没有提供一种有效的分析方法。因此，这一方面近来的工作都集中在确定系统的特征值和稳定性上。此外，QDS方法的结论只对充分大的群体有效。

#### 6. 统计物理学(Statistical Physical)

上面已经提到了一些从物理学中借鉴的方法。在统计物理学中，物理学家发明了很多种方法来处理随机系统，特别是大规模的随机系统（如热力学问题）。这样，计算机科学家完全可以从吸取更多的经验和方法，应用到EA的建模和分析中。

### 3.7 小结

前面介绍GA建模和分析的多种方法，并给出了使用这些方法得到的主要的收敛性结论。在分析有限群体算法的极限性能时，Markov方法得到广泛应用。无限群体的模型可以使我们了解算法运行中的各种特性，这种特性使用Markov方法是难以计算的。

目前，主要得到的结论是，对于SGA，算法不能收敛；对于保证最佳适应度单调性的改进算法，可以收敛。这两个结论的基础是变异算子的不可约性。若变异算子的作用逐渐衰减，同时选择压力逐渐增大，则算法也可以收敛。

这些方法对于GA中起重要作用的交叉算子的特性考虑不够，因此很多结论与广义的全局搜索算法相同。只有深入了解交叉算子的特性以及其他进化算子的相互作用，才能真正了解GA的运行特性。



## 第四章 GA 的动力学模型及收敛性分析

### 4.1 齐次 GA 的动力系统模型

当搜索空间是有限的，适应度函数和进化算子是不随时间变化的（对于时间是齐次的），可以使用动力系统（dynamical system）的模型来描述 SGA。这种研究方法也是本文中主要使用的方法。

这种理论主要由 Michael Vose 发展<sup>[76,128]</sup>。这种方法将群体看作基因型比例的向量： $p^n = (p_0^n, \dots, p_{d-1}^n)^T$ ，其中  $p_k^n$  表示搜索空间中的第  $k$  个基因型在第  $n$  代时在整个群体中的比例。由于向量的所有分量的归一性，因此，向量在  $d$  维单纯形  $\Delta^d = \{x \in R^d, x_i \geq 0, \sum_{i=0}^{d-1} x_i = 1\}$  的边界上取值，其中  $d$  是群体基因型的所有可能种类： $d = 2^l$ 。这样群体向量在单纯形中的轨迹（trajectory）确定了一个 Markov 链。向量  $p$  有两种解释，一种是确定性的，它表示了一个实际群体，其中每个基因型的数量等于  $N \cdot p_i$ ；另一方面，从概率的角度来看，也可以将此向量看作是群体的一个样本分布，每个分量代表对应的基因型出现概率的期望值。在无限群体模型中，这两种解释经常不加区别地使用。

可以定义算子： $G: \Delta^d \rightarrow \Delta^d$ ，将第  $n$  步的群体变换到第  $n+1$  步。对于无限群体，有  $p_{n+1} = G(p_n)$ 。对于有限群体，这个算子也就是 Markov 链的状态转移算子，并有期望值  $E[p_{n+1}] = G(p_n)$ 。这个算子可以分解为三个部分，即选择（S），变异（M）和交叉（C），即  $G = C \circ M \circ S$ 。也可以将交叉和变异算子组合在一起，构成混和（mixing）算子，记为  $X$ 。

$G(p)$  是下一代群体样本概率的期望。当群体规模趋于无穷时，样本的差异将趋于 0。这样，Markov 链的轨迹变成了一个确定性的序列  $p_0, G(p_0), G^2(p_0), \dots$ ，也就从一个 Markov 模型转变为动力系统模型。因此，这个模型也可以称作无限群体动力系统模型。对于适应度函数随时间变化的情况，也有一些分析<sup>[32,74,75]</sup>。

值得注意的是，这个无限群体的序列也提供了有限群体模型的信息。实际上，在  $\|G(p) - p\|$  足够小的区域上，有限群体将产生一些周期点。这些

区域 (例如  $G$  的不动点附近) 称为亚稳态 (metastable)<sup>[44,75]</sup>。最近, Douglas 等人研究了一种群体的抽象属性  $p$ , 并发现了有限群体和无限群体的这种属性的联系<sup>[40]</sup>。他证明了满足  $E[p_{n+2}] \leq G^2(p_n)$  的充分条件是  $p$  是有限群体向量的一个线性算子, 并且对于所有的有限群体, 满足  $G(p_n) \leq p_{n+1}$ 。

比例选择算子是一个比例线性算子:  $F(p) = Sp / \|Sp\|_1$  ( $\|\cdot\|_1$  代表向量的一阶距离范式)。这里  $S$  是一个对角矩阵, 有  $S_{k,k} = f(k)$ 。

变异也是一个线性算子, 由变异矩阵  $M$  给出,  $M_{i,j}$  是由基因型  $i$  变异为基因型  $j$  的概率。变异矩阵一般是对称的, 不可约的。

假设我们给出一个双线性 (bilinear) 算子  $B: R^d \times R^d \rightarrow R^d$ , 对于所有两个参数是线性的, 这样定义为  $Q(x) = B(x, x)$  的映射  $Q: R^d \rightarrow R^d$  称为二次的 (quadratic)。交叉算子就是这样一个算子。它可以表示为  $d$  个矩阵, 其中  $C_{i,j}^k$  是基因型  $i$  和  $j$  的后代是基因型  $k$  的概率。在本文的后面, 将给出这个矩阵特性更详细的分析。

当 GA 没有交叉时, 由于选择算子和变异算子都是线性算子, 因此整个模型也是一个线性系统。这样, 存在不动点满足  $M \cdot S \cdot p = p$ , 这些不动点是矩阵  $M \cdot S$  的特征向量, 而  $\lambda$  是其特征值,  $p$  的平均适应度。Perron-Frobenius 定理保证了, 对于最大的特征值, 在单纯形  $\Delta^d$  中有一个确定的特征向量。其他的特征向量也可能是有意义的, 例如, 如果他们接近单纯形的边界, 可以产生一个前面提到的亚稳态区域。

当存在交叉算子时, 系统就不再是线性的, 因而难以使用直接的方式进行分析。本文中就试图对此进行一定的探讨。

## 4.2 SGA 的一般动力学模型

下面对于 SGA, 在群体规模足够大和不考虑变异算子的前提下, 使用动力系统方法进行建模。设基因编码的长度为  $l$ , 每一位的取值为 0 或 1, 所以整个染色体空间为  $R = 2^l$ , 同时记  $m = R - 1$ 。每一个基因型可以看作是一个二进制整数, 在以后经常使用这个数来代替对应的基因型。要注意, 与使用 Markov 链的有限群体模型不同, 这里不关心个体的次序, 因此所有

的下标均不表示个体，而表示基因型。

对于一个特定的基因型  $i(0 \leq i < R)$ ，它的适应度为  $f(i)$ ，简记为  $f_i$ ，令  $x_i$  表示其在整个群体中所占的比例。所有基因型比例构成的向量  $x^n = (x_0^n, \dots, x_m^n)$  表示第  $n$  代群体的分布情况。在不引起误会的情况下，可以忽略上标。

群体的平均适应度为： $f_{avg} = \sum_{i=0}^m x_i f_i$ 。

使用选择算子后，频率的变化为：

$$sel_i(\dot{x}) = x_i f_i / f_{avg} \quad (2.1)$$

在这里， $\dot{x}$  意味着函数的参数是群体空间中的所有个体。函数取值的下标表示得出某一个个体的频率。在下面的公式中也将使用这种表示方法。

在交叉时，子代一部分按照交叉概率由交叉产生，其他的由父代直接继承。经过此算子作用后，频率的变化为：

$$mcrs_i(\dot{x}) = crs_i(\dot{x})c + x_i(1 - c) \quad (2.2)$$

上式中的  $crs$  函数为一般的单点交叉函数，即选择两个父本之后，等概率的从 1 到  $l$  选择一个交叉点进行交叉得到两个子代，从中随机选择一个作为交叉结果。在某些实现中，允许交叉点在染色体序列的头部或尾部，这相当于没有交叉，直接由父代继承，这种概率已经包含在上式的第二项中，只要对交叉概率作一简单变换即可。 $crs$  函数的一般形式为：

$$crs_k(\dot{x}) = \sum_i \sum_j c_{i,j}^k x_i x_j \quad (2.3)$$

其中系数  $c_{i,j}^k$  表示先后选取编号为  $i$  和  $j$  的染色体交叉后子代为  $k$  的概率， $c$  为直接从父代继承的比例，即交叉概率。

综合上面两个算子，总的变换公式为：

$$F_i(\dot{x}) = mcrcs_i(sel(\dot{x})) \quad (2.4)$$

这样，SGA 的基本模型为：

$$\begin{aligned}
& \bar{f}_{avg} = \frac{1}{m} \sum_{i=1}^m f_i \\
& \bar{sel}_i(\bar{x}) = f_i / \bar{f}_{avg} \\
& \bar{crs}_k(\bar{x}) = \sum_{i=1}^m \sum_{j=1}^m \bar{c}_{i,j}^k \bar{x}_i \bar{x}_j \\
& \bar{mcrs}_i(\bar{x}) = \bar{crs}_i(\bar{x}) \cdot c \cdot (1 - c) \\
& \bar{F}_i(\bar{x}) = \bar{mcrs}_i(\bar{sel}(\bar{x})) \quad i = 1..m
\end{aligned} \tag{2.5}$$

### 4.3 单点交叉算子的数学特性

前面已经提到，只有深入了解交叉算子的特性以及与其他进化算子的相互作用，才能真正了解 GA 的运行特性。

一般的单点交叉算子的执行过程如下：选择两个父本之后，等概率的从 1 到  $l$  选择一个交叉点进行交叉得到两个子代，从中随机选择一个作为交叉结果。每一个子代包含了两个父本的一部分信息。

单点交叉算子的一般形式为： $\bar{crs}_k(\bar{x}) = \sum_i \sum_j \bar{c}_{i,j}^k \bar{x}_i \bar{x}_j$  (3.1)。

上式左端的下标表示对整个向量进行交叉后，结果是基因型  $k$  的概率。右端的系数  $\bar{c}_{i,j}^k$  表示基因型  $i$  和基因型  $j$  相互交叉后能够生成  $k$  的概率。这个系数有如下特征。

1.  $\bar{c}_{i,j}^k = \bar{c}_{i',j',s}^{k'}$ 。这里  $\bar{c}_{i',j',s}^{k'}$  表示异或运算。这一点通过交叉算子的对称性可以很容易证明。

因此，在整个群体中， $\bar{c}_{i,j}^k$  的三个变量中只有两个变量是独立的。下面我们只讨论  $\bar{c}_{i,j}^0$ ，简记为  $c_{i,j}$ 。

2.  $c_{i,j} = c_{j,i}$ ，特别的， $c_{0,0} = 1, c_{i,i} = 0 (i \neq 0)$ 。因此， $c_{i,j}$  构成一个  $R$  阶对称方阵。其对角线元素除第一个为 1 外，其余为 0。这样(2.1)式可以使用矩阵的形式重写为：

$$\bar{crs}_k(\bar{x}) = \bar{x} \cdot \bar{C}_k \cdot \bar{x}^T \tag{3.2}$$

如果引入一个置换矩阵  $\bar{e}_k$ ，满足  $\bar{e}_k \bar{e}_i = \bar{e}_{i'}$ ，其中  $\bar{e}_i$  是第  $i$  个单位向

量，则交叉算子的作用可以使用矩阵表示为一个统一的形式：

$$\overline{crs}(x) = (\sum_k x_k)^T C \sum_k x_k \quad (3.3)$$

同时可以看出，矩阵  $C$  是一个正定矩阵，可以分解为  $C = Q^T Q$ 。

3. 若 Hamming 距离  $Ham(i,0) \leq 1$ ，则  $c_{i,0} \leq 1$ ；若  $Ham(i,0) \leq 1$ ，则  $c_{i,0} \leq 1/2$ ；若  $Ham(i,0) \leq l$ ，则  $c_{i,0} \leq 0$ 。

这可以通过算子的定义直接得出。

$$4. \sum_{i \neq 0}^{R-1} \sum_{j \neq 0}^{R-1} c_{i,j} \leq R \quad (3.4)$$

这一点仍然可以从交叉算子的对称性可以证明。当群体中所有基因型都存在，并且比例都相等时（为  $1/R$ ），所有基因型处于对称的地位。因此，经过交叉操作后，所有的比例不发生变化。即：

$$\sum_{i \neq 0}^{R-1} \sum_{j \neq 0}^{R-1} c_{i,j} \frac{1}{R} \frac{1}{R} \leq \frac{1}{R}, \text{ 因此命题得证。}$$

5. 若令  $sc(l) = 2 \sum_{i \neq 0}^{R-1} c_{i,0}$  (3.5)，将它看作是  $l$  的函数，则

$$sc(l+1) \leq 2 \cdot sc(l) \leq 4, \text{ 且 } sc(2) \leq 4 \quad (3.6)$$

证明：

首先定义一个基因型  $k$  在每个交叉点与 0 相交叉，结果为 0 的个数之和（如果这个基因型不是 0 本身，那么每一个交叉点的计数为 0 或 1，反之，每一个交叉点的计数为 2）为此基因型的交叉计数，简记为函数  $cc(k)$ 。

这个函数实际上就是  $cc(k) = 2 \sum_{i \neq 0}^{R-1} c_{i,0}$ ，所以  $sc(l) = \sum_{k \neq 0}^{R-1} cc(k)$

从前面数，模式为 01\*\*\* 的有  $2^{l-2}$  个，其交叉计数为 1；模式为 001\*\*\* 的有  $2^{l-3}$  个，其交叉计数为 2；如此等等，最后，模式为 00...01 的有 1 个，交叉计数为  $l-1$ 。总的计数为  $\sum_{i=2}^l 2^{l-i} \cdot i$ 。从后面数的计数和此数相同。然后再加上 0 自身的交叉计数  $2 \cdot (l-1)$ ，则总的计数为  $sc(l) = 2 \sum_{i=2}^l 2^{l-i} \cdot i + 2(l-1)$ 。

这样有：

$$\frac{1}{2}sc(l+1) \leq sc(l) + [\sum_{i=0}^{l-1} 2^{l-i} \cdot 2^{l-i} \cdot (i+1) \cdot 2] \cdot \sum_{i=0}^{l-1} 2^{l-i} \cdot 2^{l-i} \cdot (i+1) \leq l \cdot 2(l+1) \leq 2 \cdot 2^{l+1} \cdot l \leq l \cdot 2(l+1) \leq 2$$

即  $sc(l+1) \leq 2 \cdot sc(l) \leq 4$

命题得证

6. 求解上面的递推公式，有  $sc(l) \leq 2^{l+1} \leq 4$  (3.7)

推导过程如下：

若令  $b(l) = cc(l) \leq k$ ，其中  $k$  为待定参数。代入(3.6)式，有

$b(l+1) \leq k + 2 \cdot b(l) \leq 2k + 4$ 。若想消去参数  $k$ ，则解出  $k=4$ 。因此，

$b(l+1) \leq 2 \cdot b(l)$ ，即  $b(l) \leq b(2) \cdot 2^{l-2}$ ，又因  $b(2) \leq 8$ ，代回即可得出上面公式。

因此，若令  $c_0(l) = \sum_{i=0}^{R-1} c_{i,0} \leq sc(l)/(2l+2)$ ，则  $c_0(l) \leq \frac{2^{l+1} \cdot 4}{2l+2} \leq \frac{2^l \cdot 2}{l+1}$ 。

## 4.4 SGA 的局部收敛性

下面分析上面的动力系统模型的局部收敛性。我们的思路是求解系统在平凡不动点处的 Jacobin 矩阵的特征值，判断这些特征值是否小于 1。

**定义 4.1** 每一个形如  $ps = (0,0,\dots,x_k=1,\dots,0,0)$  的点，即只有一个变量为 1，其他变量为 0 的点都是动力系统(2.5)的不动点，称之为第  $k$  个平凡不动点。

简单的将数值代入计算即可证明上述定义。由于每个平凡不动点对应 GA 演变到整个群体都成为此基因型的状态，因此平凡不动点的吸引性和 GA 是否能够收敛到这种状态是等价的。下面就来分析平凡不动点的吸引性。

下面引入一种按照某个平凡不动点  $k$  对所有基因型的编号进行重新排序的方法 hsort。这种排序方式是基于 Hamming 距离的。

首先, 令平凡不动点  $k$  的新序号为 0, 然后按照任意次序将与基因型  $k$  的 Hamming 距离为 1 的所有基因型排列 0 之后, 接着排列与  $k$  距离为 2 的基因型, 如此一直排列到与  $k$  距离为  $l$  的那个基因型, 它的新次序为  $m=R-l$ 。在这种变换下, 各种基因型按照它们与  $k$  的 Hamming 距离构成了一个格 (lattice)。

按照这种排序方式, 将原来基因频率空间中的  $x_i$  分量重写为  $y_j(j=\text{hsort}(i))$ 。这种重写只是对变量排列次序的一种变化, 并不改变模型(2.5)的公式形式和系统的动力学形态。重写之后原来的平凡不动点  $k$  对应的新的不动点向量为  $ps=(1,0,\dots,0)$ 。下面就通过重写后的模型来讨论这个不动点的性质。

注意到映射(2.5)在空间  $[0,1]^R$  上由于满足  $\sum y_i = 1$ , 所以它不是满的, 也不存在逆。但如果将此变换中的第一个分量用其他分量的组合  $y_0 = 1 - \sum_{i=1}^m y_i$  (4.1) 替换, 在空间  $[0,1]^{R-1}$  上变换是满的, 原来的不动点  $ps$  映射为新空间中的原点 0。

在此变换下, 选择算子的形式不变, 交叉算子变为:

$$crs_k(\vec{y}) = \sum_{i=1}^m \sum_{j=1}^m c_{i,j}^k y_i y_j + 2 \sum_{i=1}^m [c_{i,j}^k y_i (1 - \sum_{j=1}^m y_j)] + c_{0,0}^k (1 - \sum_{j=1}^m y_j)^2 \quad (4.2)$$

由于染色体自身交叉的子代只能还是自己, 所以  $c_{0,0}^k = 0$ , 这样, 模型(2.5)转换为以下形式:

$$\begin{aligned} \vec{f}_{avg} &= \sum_{i=1}^m y_i \vec{f}_i + (1 - \sum_{i=1}^m y_i) \vec{f}_0 \\ \vec{sel}_i(\vec{y}) &= y_i \vec{f}_0 / f_{avg} \\ \vec{crs}_k(\vec{y}) &= \sum_{i=1}^m \sum_{j=1}^m c_{i,j}^k y_i y_j + 2 \sum_{i=1}^m [c_{i,j}^k y_i (1 - \sum_{j=1}^m y_j)] \quad (4.3) \\ \vec{mcrs}_i(\vec{y}) &= \vec{crs}_i(\vec{y}) \vec{c} + y_i (1 - \vec{c}) \\ \vec{F}(y_i) &= \vec{mcrs}(\vec{sel}(y_i)) \quad i = 1..m \end{aligned}$$

下面就在  $[0,1]^{R-1}$  空间上讨论原点的吸引力。

由于每个平凡不动点也是选择算子和交叉算子的不动点，因此  $F(0)$  在此点的 Jacobin 矩阵  $DF(0)$  可以分解为  $DF(0) = D_{mcrs}(0) D_{sel}(0)$ 。下面分别计算这两个矩阵的值。

为求选择算子的 Jacobin 矩阵，计算如下：

$$\frac{\partial Sel_k}{\partial y_k} = \frac{f_k}{f_{avg}} - \frac{y_k \cdot f_k \cdot (f_k - f_0)}{f_{avg}^2}, \frac{\partial Sel_k}{\partial y_{i,i \neq k}} = - \frac{y_k \cdot f_k \cdot (f_i - f_0)}{f_{avg}^2} \quad (4.4)$$

注意到在平凡不动点， $y_{i,i \neq k} = 0, f_{avg} = f_0$ ，因此上式化简为：

$$\frac{\partial Sel_k}{\partial y_k} = \frac{f_k}{f_0}, \frac{\partial Sel_k}{\partial y_{i,i \neq k}} = 0 \quad (4.5)$$

这样，矩阵的具体形式如下：

$$D_{sel}(0) = \frac{1}{f_0} \begin{pmatrix} f_1 & & & \\ & f_2 & & 0 \\ & & \dots & \\ & 0 & & \dots \\ & & & & f_m \end{pmatrix} \quad (4.6)$$

对单点交叉函数  $c_{rs}$  求  $y_i$  的偏微分，有：

$$\frac{\partial c_{rs_k}(y)}{\partial y_n} = 2 \cdot \left( \sum_{i=0, i \neq n}^m c_{i,0}^k \cdot y_i - c_{n,0}^k \cdot (y_0 - y_i) \right) = 2 \cdot c_{n,0}^k \quad (4.7)$$

为计算系数  $c_{n,0}^k$  的值，先叙述和证明如下引理。

**引理 4.1** 若染色体  $p1$  与  $p2$  作为父本进行交叉，它们的 Hamming 距离  $Ham(p1, p2) = d$ ，交叉产生的任意子代为  $s$ ，则  $Ham(s, p1) \leq d$ ， $Ham(s, p2) \leq d$ 。上式的等号成立当且仅当  $s$  等于  $p1$  或  $p2$ 。

此引理可通过交叉操作的定义简单得证。另外可以看出，此引理对于一般的交叉算子也是成立的。

**定理 4.2** 对于任意  $0 \leq i < k < m$ ，有  $c_{0,i}^k = c_{i,0}^k = 0$ 。

证明：设  $Ham(0, k) = d$ ，按照上面介绍的排序方法可知，对于任意  $i < k$ ，有  $Ham(0, i) \leq d$ ，由上面的引理，0 和  $i$  号染色体交叉后的子代与 0 号染色体的 Hamming 距离小于等于  $d$ ，若子代与 0 的 Hamming 距离小于  $d$ ，则子



代显然不能是  $k$ ；若等于  $d$ ，按引理 2.1 知，子代只能是 0 或  $i$ ，也不能是  $k$ 。所以  $k$  不可能由 0 和  $i$  交叉产生。因此定理得证。

**定义 4.2** 对于两个不同的染色体  $i, j$ ，定义函数  $inh(i, j)$  为这两个染色体相交叉后子代与  $i$  相同的概率，即  $inh(i, j) = 2^{-c_{i,j}^i}$ 。

可以通过计算两个染色体从头开始相同的基因长度和从末尾向前相同的基因个数之和，再除以整个染色体长度  $L-1$  得到  $inh$  的函数值，对于单点交叉，明显有  $0 \leq inh(i, j) \leq 1$ 。特别的，若  $Ham(i, j) = 1$ ，则  $inh(i, j) = 1$ ；若  $Ham(i, j) = L$ ，则  $inh(i, j) = 0$ 。

综合上面的定理，可知简单交叉算子的 Jacobin 矩阵有如下形式：

$$D_{crs}(0) = \begin{pmatrix} inh(0,1) & & & \\ & inh(0,2) & \dots & \dots \\ & & \dots & \\ & 0 & & \dots \\ & & & & inh(0,m) \end{pmatrix} \quad (4.8)$$

这是一个上三角矩阵，对角线之上的值不需详细计算，对角线之下值为 0。

**定理 4.3** 动力系统(4.3)在原点的 Jacobin 矩阵的所有特征值为

$$\lambda_i = (inh(0, i) \dots 1 \dots) f_i / f_0, i = 1..m \quad (4.9)$$

证明：因为  $DF(0) = D_{mcrs}(0) D_{sel}(0) = (D_{crs}(0) \oplus (1 - \dots) I) D_{sel}(0)$ 。由上可知参与运算的都是上三角矩阵，所以结果也是上三角矩阵，并且  $DF$  的对角线值是  $D_{mcrs}$  和  $D_{sel}$  的对角线值的对应乘积。又由于上三角矩阵的特征值就是它的对角线，所以定理得证。

**定理 4.4** 动力系统(2.5)在第  $k$  个平凡不动点处的 Jacobin 矩阵的所有特征值为

$$\lambda_i = (inh(k, i) \dots 1 \dots) f_i / f_k, i = 0..m, i \neq k \quad (4.10)$$

证明：由上面排序过程中选取平凡不动点的任意性，结合定理 4.3 知本定理成立。

**定义 4.3** GA 的所有基因型中，如果某个基因型  $k$  满足

$f_k = (inh(k, i) \dots 1 \dots) f_i$ ，对于任意  $i = 0..m, i \neq k$ 。则称在搜索空间中此基因型频率为 1 的点为局部极值点。

可以看出,动力系统(2.5)在局部极值点的 Jacobin 矩阵的所有特征值都小于 1, 所以此点为其邻域的拓扑共轭的吸引点。也就是说 GA 算法在此基因型频率接近于 1 的邻域内收敛到此基因型频率为 1 的状态。

由于交叉因子的引入, GA 算法的局部极值点与一般的连续空间的局部极值点的定义和要求是不一样的。另外, 由于当  $Ham(i,j)=1$  时  $inh(i,j)=1$ , 因此局部极值点的一个必要条件是此点的适应度值比所有与其 Hamming 距离为 1 的点的适应度大, 这个条件与  $\mu$  值的选取无关。

**定义 4.4** 如 GA 算法中某个基因型  $k$  的适应度值为全体适应度的唯一最大值, 则称此点为全局极值点。

由于  $inh(i,j) \leq 1$ ,  $\mu \leq 1$ , 所以当  $f_k$  为最大值时,  $f_k \geq (inh(k,i) \cdot \mu) \cdot f_i$  对于任意  $i \in \{0, \dots, m\}, i \neq k$  都成立。因此全局极值点也一定是局部极值点。

由于不使用变异算子的标准遗传算法的收敛性, 可以得出: 遗传算法能够收敛到某一全局一致状态的充要条件是此点为局部极值点。但算法能否收敛到某一个特定的局部极值点或全局极值点是依赖于算法的全局动力形态和初始条件的。

## 4.5 一般形式的遗传算法

上面对于 SGA 证明了局部极值点的存在条件。从中可以看出, 证明过程依赖于以下条件:

1. 选择算子和交叉算子的可分解性。也就是说, 整个系统的平凡不动点也分别是选择和交叉算子的不动点。因此, 此处的 Jacobin 矩阵可以分解为两个算子的 Jacobin 矩阵的乘积。

2. 两个算子的 Jacobin 矩阵都是上对角矩阵的形式。因此可以简单的得到乘积矩阵的形式和特征值。

3. 交叉算子的 Jacobin 矩阵是上对角矩阵的缘由在于引理 4.1, 有了这个引理, 在基于 Hamming 距离的排序方式下, 才有定理 4.2。

对于其它改进的遗传算法, 如果满足这些条件, 也可以使用这种方法进行类似的证明。

首先应该注意到，上一节的证明中并没有使用二进制空间的条件，因此上面的结论可以推广到基因的每一位可以取多个值的情况。

对遗传算法的一个基本改进是引入代间隔。下面的定理描述了具有代间隔和交叉概率的 GA 的情况。

**定理 5.1** 对于一个变异算子可忽略的 GA 模型，若其选择算子和交叉算子满足可分解性。并且在某一平凡不动点  $ps = (0, 0, \dots, x_p = 1, \dots, 0, 0)$  处，选择算子的 Jacobin 矩阵的形式为  $DS(ps) = \text{diag}(s_1, s_2, \dots, s_m)$ ，交叉算子的 Jacobin 矩阵为一上三角矩阵，具体形式为：

$$DC(ps) = \begin{pmatrix} ch(p,1) & & & \\ & ch(p,2) & \dots & \\ & & \dots & \\ & 0 & & \dots \\ & & & & ch(p,m) \end{pmatrix}$$

那么，当综合考虑代间隔  $g$  和交叉概率  $\mu$  时，算法模型在此平凡不动点处的所有特征值为： $\lambda_i = c_i \mu s_i g + 1, g, i = 1..m$  (5.1)，其中  $c_i = ch(p,i) \mu + 1$  (5.2)。

**证明：**

上述的 GA 可以建立如下数学模型：

$$F(x) = (1 - g)x + g(C \circ S(x) + (1 - \mu)S(x))$$

其中  $C$  和  $S$  分别表示交叉算子和选择算子。

由算子的可分解性，在平凡不动点处，模型的 Jacobin 矩阵为

$$DF(ps) = (1 - g)I + g((\mu DC(ps) + (1 - \mu)I) \circ DS(ps))$$

按照假设，参与运算的矩阵都是上三角矩阵，因此结果也是上三角矩阵，其对角元素分别为  $\lambda_i = (ch(p,i) \mu + 1) \mu s_i g + 1, g, i = 1..m$ 。这也就是矩阵的特征值。

证毕

**定理 5.2** 代间隔参数不影响 GA 模型在平凡不动点处的吸引性。

**证明：**由于  $\lambda_i = 1 - \mu s_i g$ ，反之亦然，可见定理成立。

因此，在以下的讨论中，不再涉及代间隔参数  $g_0$ 。

**定义 5.1** 在满足定理 3.1 的 GA 模型中，如果某个基因型  $k$  满足  $\sum_{i=0}^m c_i s_i = 1$ ，对于任意  $i = 0, \dots, m, i \neq k$ 。则称在搜索空间中此基因型频率为 1 的点为局部极值点。

## 4.6 不同选择算子的模型

不仅上面证明的基于适应度比例的选择算子，其他一些常用的选择算子也可以满足上述的三个条件，从而可以求出其 Jacobin 矩阵，然后引用定理 5.1 得到一般性的结论。

下面对于基于排名的两种选择算子，即线性排名和指数排名算子以及基于锦标赛的选择算子进行计算。这些算子显然都满足平凡不动点的可分解性，即一致群体经过选择算子作用后仍为一致群体。

在计算过程中，对于每种算子，首先建立有限群体下算子的作用公式  $s_N(k)$ ，然后将其推广到无限群体，得到这时的选择算子模型  $s_k(x)$ 。然后在平凡不动点处，求  $s_k(x)$  对于  $x$  的偏导数，从而得到选择算子的 Jacobin 矩阵。计算的结果表明这些矩阵都是对角矩阵，满足上面的条件 2。

### 4.6.1 基于排名的选择策略

基于排名的方法最早由 Baker 提出<sup>[8]</sup>，他使用了线性排名的方法，Michalewicz 又提出了指数排名的方法<sup>[89]</sup>。这种方法的基本做法是，计算个体的适应度在整个群体中的排名，然后根据这个排名进行某种变换得到个体的选择概率。

对于有限群体，设群体规模为  $N$ ，首先对所有的基因型按适应度排序，整个群体的排名从 1 到  $N$ 。不妨设基因型的适应度的排序就是  $f_0 \leq f_1 \leq \dots \leq f_m$ ，对于某一个特定的基因型  $k (0 \leq k < R)$ ，设其在群体中共有  $N x_k$  个，因此适应度比它小的个体的数目共有  $rl_N(k) = \sum_{i=0}^{k-1} N x_i$  (6.1) 个。它的排名就是  $rl_N(k)$ ，对于那些和它相同的

基因型，依次取这些基因型的排名从  $rl_N(k)$  到  $rl_N(k) + N - x_k - 1$ 。

然后，对排名进行某种变换  $rx(k) = rx(r(k))$ ，然后进行归一化处理，就得到个体的选择概率。通常所用的变换函数有两种：

$$\text{线性排名： } rx(x) = ax + b \quad (6.2)$$

$$\text{指数排名： } rx(x) = q^x (q + 1) \quad (6.3)$$

上面出现的  $a, b, q$  都是实际使用时可以调节的参数。由于使用从小到大的排名方式，因此  $a > 0, b > 0, q > 1$ 。

对于某个个体，选择概率是个体的排名变换除以群体中所有个体的变换函数之和。对于基因型  $k$ ，选择后的个体数目比例是所有相同基因型的个体总的选择概率之和。即：

$$s_N(k) = \frac{\sum_{i=0}^{N-x_k-1} rx(rl_N(k) + i)}{\sum_{i=1}^N rx(i)} \quad (6.4)$$

**线性排名**

这时有：

$$\begin{aligned} s_N(k) &= \frac{\sum_{i=0}^{N-x_k-1} a(rl_N(k) + i) + b}{\sum_{i=1}^N (ai + b)} \\ &= \frac{a \cdot rl_N(k) + N \cdot x_k + b \cdot N + a \cdot N x_k (N x_k + 1) / 2}{a \cdot N(N + 1) / 2 + b \cdot N} \\ &= \frac{a \cdot x_k (rl_N(k) + (N - x_k + 1) / 2) + b \cdot x_k}{a \cdot (N + 1) / 2 + b} \end{aligned} \quad (6.5)$$

当  $N$  趋于无穷大时，取极限，有

$$\begin{aligned} s_k(x) &= \lim_{N \rightarrow \infty} s_N(k) = \lim_{N \rightarrow \infty} \frac{a \cdot x_k (\sum_{i=0}^{k-1} N \cdot x_i + 1 + (N - x_k + 1) / 2) + b \cdot x_k}{a \cdot (N + 1) / 2 + b} \\ &= x_k (2 \sum_{i=0}^{k-1} x_i + x_k) \end{aligned} \quad (6.6)$$

因此，对于  $N$  趋于无穷的情况，可以令

$$r(k) = 2 \sum_{i=0}^{k-1} x_i + x_k \quad (6.7)$$

作为此基因型的选择概率。

从公式(6.6)中可以看出，对于无限群体，线性排名变换函数中的参数

已经失去作用，具体的取值对于算法的无限特性没有影响。

下面计算在平凡不动点处  $s(\mathbf{x})$  的 Jacobin 矩阵。在计算过程中，仍然使用按照适应度的相对大小对基因型的排列次序。对于某个平凡不动点  $\mathbf{p}_s$ ， $x_p \neq 1, x_i \neq 0 (i \neq p)$  (6.8)。计算偏导数：

$$\frac{\partial s_k(\mathbf{x})}{\partial x_{j,j \neq k}} \neq 2x_k \neq 0 \quad (6.9)$$

$$\frac{\partial s_k(\mathbf{x})}{\partial x_k} \neq 2(\sum_{i=0}^{k-1} x_i \neq x_k) \neq 2 \sum_{i=0}^k x_i \quad (6.10)$$

注意到由于条件(6.8)，当  $k > p$  时，上式值为 2，否则值为 0。

若令函数  $\delta(x) = \begin{cases} 0 & \text{when } x \neq 0 \\ 1 & \text{else} \end{cases}$  (6.11)，上式可以表示为

$$\frac{\partial s_k(\mathbf{x})}{\partial x_k} \neq 2\delta(k \neq p) \neq 2\delta(f_k \neq f_p) \quad (6.12)$$

虽然上面的计算中依照适应度相对值的排列次序，但得到的结果(6.9),(6.12)与这个次序无关，可以适用于其他的情况。

为了和交叉算子相结合，需要按照上面介绍的按照 Hamming 距离排序的方式重新整理次序。在新的次序下，使用线性排名的选择算子在平凡不动点处的 Jacobin 矩阵是一个对角矩阵，其形式为：

$$\mathbf{DS}(\mathbf{0}) = \text{diag}(2\delta(f_1 \neq f_p), 2\delta(f_2 \neq f_p), \dots, 2\delta(f_m \neq f_p)) \quad (6.13)$$

### 指数排名

这时，结合公式(6.3)，(6.4)，某个基因型选择后的个体数目比例为：

$$s_N(k) = \frac{\sum_{i=0}^{N-1} q^{rl(k) \cdot i} (q \neq 1)}{\sum_{i=1}^N q^i (q \neq 1)} \neq \frac{\exp(\ln q \cdot N \sum_{i=0}^{i-1} x_i) (q^{N-1} \neq 1)}{(q^N \neq 1)} \quad (6.14)$$

若使用固定参数的指数排名方法，则当群体数目趋于无穷时，其选择概率不存在极限。若令参数  $q$  随群体数目变化而变化，即取  $q \neq q_0^{1/N}$ ，则上式与  $N$  无关，也就是无限群体中使用指数排名的选择算子的作用结果：

$$s_k(\mathbf{x}) - s_N(k) = \frac{\exp(\ln q_0 - \sum_{i=0}^{i^k} x_i)(q_0^{x_k} - 1)}{q_0 - 1} \quad (6.15)$$

同样计算在平凡不动点处的偏导数，结合公式(6.8)，有：

$$\left. \frac{\partial s_k(\mathbf{x})}{\partial x_{j, j^k}} \right|_{\mathbf{x}^*} = \frac{\exp(\ln q_0 - \sum_{i=0}^{i^k} x_i)(q_0^{x_k} - 1) \ln q_0}{q_0 - 1} \Big|_{\mathbf{x}^*} = 0 \quad (6.16)$$

$$\frac{\partial s_k(\mathbf{x})}{\partial x_k} = \frac{\exp(\ln q_0 - \sum_{i=0}^{i^k} x_i) q_0^{x_k} \ln q_0}{q_0 - 1} \quad (6.17)$$

在上式中，若  $k > p$ ，也就是  $f_k > f_p$  时， $\sum_{i=0}^{i^k} x_i = 1$ ，否则  $\sum_{i=0}^{i^k} x_i = 0$ ，因此上式的值为：

$$sd(k) = \left. \frac{\partial s_k(\mathbf{x})}{\partial x_k} \right|_{\mathbf{x}^*} = \begin{cases} \frac{\ln q_0}{q_0 - 1} & \text{when } f_k > f_p \\ \frac{q_0 \ln q_0}{q_0 - 1} & \text{else} \end{cases} \quad (6.18)$$

这个结果也和基因型的排列方式无关。同样，将矩阵中元素的次序转换为按 Hamming 距离排列，使用指数排名的选择算子在平凡不动点处的 Jacobin 矩阵的形式为：

$$\mathbf{DS}(\mathbf{0}) = \text{diag}(sd(0), sd(1), \dots, sd(m)) \quad (6.19)$$

## 4.6.2 基于锦标赛方式的选择策略

基于锦标赛方式的选择策略是一种局部竞争的选择策略。使用这种选择方式，每次从群体中均匀随机选出  $t$  个个体，从中选出适应度最大的个体作为获胜者参与下一步进化过程。 $t$  为锦标赛规模(tournament size)，常用值为  $t=2$ 。

由于这种选择方式仍然依赖于个体间适应度的相对值，因此这里仍然使用上一节的假设和排序方式。

在文献<sup>[20]</sup>中已经证明，经过规模为  $t$  的锦标赛选择后，某个基因型  $k$  的比例的期望值变成：

$$s_N(k) = \sum_{i=0}^k x_i^t = \sum_{i=0}^{k-1} x_i^t \quad (6.20)$$

当群体规模趋于无穷时，上式的值不变，也就是无限群体时锦标赛选择的作用公式为： $s_k(\mathbf{x}) = \sum_{i=0}^k x_i^t = \sum_{i=0}^{k-1} x_i^t$ 。

可以注意到一点，当  $t=2$  时

$$s_k(\mathbf{x}) = \sum_{i=0}^k x_i^2 = \sum_{i=0}^{k-1} x_i^2 + x_k (2 \sum_{i=0}^{k-1} x_i + x_k) \quad (6.21)$$

与基于线性排名的作用公式(6.6)完全相同。这说明对于无限群体，规模为 2 的锦标赛选择算子与基于线性排名算子的作用完全相同。

下面求偏导数，有

$$\left. \frac{\partial s_k(\mathbf{x})}{\partial x_{j,j \neq k}} \right|_{x \neq ps} = \begin{cases} 0 & \text{when } j = k \\ 2 \sum_{i=0}^k x_i^{t-1} & \text{when } j \neq k \end{cases} \quad (6.22)$$

当  $j < p$  时， $\sum_{i=0}^k x_i = \sum_{i=0}^{k-1} x_i + 0$ ；当  $j > p$  时， $\sum_{i=0}^k x_i = \sum_{i=0}^{k-1} x_i + 1$

$$\text{综合上面各种情况，均有 } \left. \frac{\partial s_k(\mathbf{x})}{\partial x_{j,j \neq k}} \right|_{x \neq ps} \geq 0 \quad (6.23)。$$

$$\left. \frac{\partial s_k(\mathbf{x})}{\partial x_k} \right|_{x \neq ps} = \begin{cases} 2 \sum_{i=0}^k x_i^{t-1} & \text{when } k \neq p \\ 0 & \text{when } k = p \end{cases} \quad (6.24)$$

上式同样与基因型的排列次序无关，锦标赛选择算子的 Jacobin 矩阵仍然是一个对角矩阵：

$$DS(0) = \text{diag}(t(f_1 - f_p), t(f_2 - f_p), \dots, t(f_m - f_p)) \quad (6.25)。$$

### 4.6.3 不同选择算子的局部收敛性的比较

在上面得到了无限群体模型中，各种选择策略在平凡不动点处的 Jacobin 矩阵。它们都是一个对角矩阵，分别由公式(6.13)，(6.19)，(6.25)表述。这些策略和基于适应度比例的选择策略一样，都存在局部极值点。在这些局部极值点附近，遗传算法可以收敛到这个全局一致状态。

**定理 6.1** 对于使用基于适应度比例的选择，指数排名选择和锦标赛选择这三种选择算子的 GA，若某个基因型  $p$  的适应度满足

$$f_i > c_i \sum_{i=1}^m s_i, i = 1..m, i \neq p \quad (6.26)$$



对于基于适应度比例的选择,  $s_i = f_i / f_p$

对于指数排名选择,  $s_i = \begin{cases} \frac{\ln q_0}{q_0 - 1} & \text{when } f_i = f_p \\ \frac{q_0 \ln q_0}{q_0 - 1} & \text{else} \end{cases}$

对于锦标赛选择,  $s_i = t \cdot (f_i - f_p)$

则对于状态空间中这个基因型的比例为 1 的状态, GA 在这个状态点的邻域内收敛到这种状态。这个基因型称为 GA 的局部极值点。

证明:

基于适应度比例的选择策略已经在前面中证明。

在基因型 p 的比例为 1 的状态, 也就是 GA 的第 p 个平凡不动点, 如果 GA 在这个不动点处的 Jacobin 矩阵的所有特征值都小于 1, 则这个不动点是吸引点。

对于指数排名选择, 选择算子的 Jacobin 矩阵如公式(6.19); 对于锦标赛选择, 选择算子的 Jacobin 矩阵如公式(6.25)。结合定理 3.1, 知 GA 在此处的特征值为  $\lambda_i = c_i \cdot s_i$ 。因此定理成立。

证毕

下面探讨构造局部极值点的条件。首先有如下定理成立:

**定理 6.2** 对于使用基于适应度比例的选择, 指数排名选择和锦标赛选择这三种选择算子的 GA, 局部极值点 p 存在的一个必要条件是, 若  $Ham(p, j) = 1$ , 则  $f_p = f_j$ 。

证明: 由于  $Ham(p, j) = 1$ , 因此  $inh(p, j) = 1$ , 即  $s_j = s_p$ 。当局部极值点 k 存在时, 有  $s_j = 1$ 。

对于基于适应度比例的选择策略, 有  $s_j = f_j / f_p = 1$ , 因此,  $f_p = f_j$ 。

对于指数排名策略, 注意到  $\ln q_0 < q_0 - 1$ ,  $q_0 \ln q_0 < q_0 - 1$ , 所以

$s_d(j) = 1$  when  $f_j = f_p$ ,  $s_d(j) < 1$  else

因此定理同样成立。

对于锦标赛选择策略, 有  $s_j \geq t \Rightarrow (f_j \geq f_p) \Rightarrow 1$ , 因此,  $f_p \geq f_j$ 。

证毕

因此, 一个基因型  $p$  若能够收敛到的必要条件是, 它的适应度需大于所有与其 Hamming 距离为 1 的基因型。若另一个基因型  $j$  与它的 Hamming 距离大于 1, 则对于不同的选择算子, 情况并不一样。

这里讨论局部极值点的存在条件与不同选择策略之间的关系。从定理 6.1 可以看出, 局部极值点必须满足公式(6.26), 但其中公式的具体形式是不一样的, 因此完全可以构造出一种适应度分布, 使得在不同的选择策略下, 某一个点成为局部极值点的条件不同。

下面举一个例子来说明这一点。如下一个问题, 基因型长度为 4, 各基因型的适应度与它们之间的关系如下图所示, 其中箭头表示 Hamming 距离为 1。所有没有列出的基因型的适应度都是 0, 不参与进化操作。(0000)点的适应度暂设为, 下面讨论在不同的选择策略下,  $x$  取何值可以使(0000)点成为局部极值点。

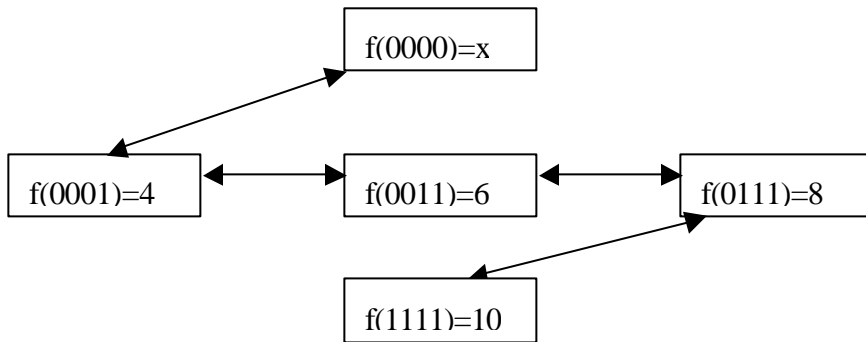


图 4.1 各基因型的相互关系

首先, 必须有  $x < 10$ , 否则的话, 它将成为全局极值点。其次, 要求  $x > 4$ , 否则它小于(0001)点的适应度, 按定理 5.2, 不可能成为局部极值点。在这些条件下, 从图 4.1 中可以看出, (1111)点是一个局部极值点, 并且因为所有其他点的适应度都小于它, 这个点也是整个函数空间的全局极值点。除 (0000)和(1111)以外其他的点都至少有一个比自身适应度大的相邻点, 因此

都不是局部极值点。

参数选取为：交叉概率  $\mu=0.8$ ，指数排名中的  $q_0=1.5$ ，锦标赛规模为  $t=2$ 。

计算对于(0000)这个点，其他的基因型的  $?_j$  值，列表如下：

基因型		0001	0011	0111	1111
适应度 $f$		4	6	8	10
$inh()$		1	2/3	1/3	0
$?c_j$		1	0.733	0.467	0.2
$?c_j ??s_j$	比例选择	$4/f_0$	$4.4/f_0$	$3.73/f_0$	$2/f_0$
	锦标赛选择	2	1.467	0.933	0.4
	指数排名	1.217	0.892	0.568	0.243

表 4.1 各种选择算子的数据

对于锦标赛选择和指数排名选择策略，由于当  $f_j ? f_0$  时， $?s_j ? 1$ ，因此上表中对应位置列出的是当  $f_0 ? f_j$  时， $?c_j ??s_j$  的值。

从上表可以看出，若要保证  $?_j, ?c_j ??s_j ? 1$ ，则对于比例选择，要求  $x > 4.4$ ；对于锦标赛选择，要求  $x > f(0011)=6$ ；对于指数排名，要求  $x > f(0001)=4$ 。

从上面的一个例子中可以看出，对于不同的选择策略，局部极值点的存在条件是不一致的。对于更加复杂的问题，对于  $x$  的同一个值，它是否是局部极值点还依赖于不同的选择策略。

## 4.7 GA 的收敛条件对算法改进的指导作用

从上面的证明中可以看出，GA 算法能够收敛到局部极值点，由于初始条件的影响，算法不一定收敛到期望的全局极值点。从理论上讲，有两种解决方法，一是设法减少非全局极值点的数量；二是研究初始条件对算法收敛目标的影响。下面就从第一个方面讨论已有的一些改进方案对算法

的影响，并提出一些新的改进方法。

遗传算法及各种改进的目的在于算法的探索性 (Exploration) 和开发性 (Exploitation) 之间的一种均衡<sup>[130]</sup>。当搜索空间中的局部极值点的数目多时，算法的探索性增强，很容易从初始条件出发找到一个极值点；反之，算法的开发性增强，算法中更多的工作是从极值点的附近过渡到极值点。通过上面的证明过程可以看出，影响局部极值点的参数为  $\text{inf}$  函数和  $\mu$  参数，因此在下面的讨论中，讨论算法改进对于这两个参数的影响直接决定算法的性能和效率。

### 4.7.1 交叉概率的影响

在 SGA 中，如果交叉概率  $c=1$ ，则相当于子代全部都是交叉的结果；在  $c$  为 0 时相当于取消交叉过程，全部由选择算子进化。可以容易证明，在其他条件不变时，随着  $c$  的减少，非全局极值点逐渐减少，当  $c$  等于 0 时，除了全局极值点外不存在其他的局部极值点。但在实际实现时，由于群体数目有限，不可能包含所有的基因型，因此必须靠交叉算子来实现探索性。不过了解上面的原则对于实现也具有指导意义：当发现群体的进化有早熟的趋势时，我们可以通过调整  $c$  参数来脱离局部极值点。

### 4.7.2 适应度函数的变化

在使用遗传算法进行优化问题求解时，问题的适应度函数的一般形式  $f(x)$  是给定的，为了提高遗传算法的性能，往往需要对它进行某些形式的变换。这种变换应该将适应度函数变换到非负值，并且是保序的。即对一个变换  $h(x)$ ，若  $f(a) > f(b)$ ，则  $h(f(a)) > h(f(b))$ ，也就是说变换  $h$  本身是单调不减函数。在使用选择算子的过程中，可以看出，真正起作用的是适应度函数的相对比值。在这个基础上，可以把变换  $h$  分为扩张的和压缩的，如果  $f(a) > f(b)$ ，且  $h(f(a))/h(f(b)) > f(a)/f(b)$ ，称之为扩张的，反之称为压缩的。可以看出，扩张的变换对于局部极值点的影响和上面讨论的  $\mu$  参数减少是类似的，压缩的变换对应于  $c$  参数的增加。使用扩张的或是压缩的变换与算法的探索性和开发性之间的抉择是一致的。

### 4.7.3 编码方式的变化

使用遗传算法进行优化计算时的一个特点就是要对问题空间进行编码。由于 GA 是依赖于位串的, 因此不同的编码方式对计算结果和性能有很大的影响。特别是对于连续变量优化问题, 使用二进制编码还是实数编码仍在讨论中<sup>[55]</sup>。

一般的二进制编码具有任一码位信息最大的优点, 但相近的实数变量的二进制表达的 Hamming 距离并不一定是接近的(如 01111 和 10000 等)。使用 Gray 编码可以解决这个问题, 由 Gray 码的构造方式可以保证相邻的数的 Hamming 距离为 1。

在使用 Gray 码表示的前提下, 可以得出一元函数的极值点与 GA 算法局部极值点的对应关系, 这就是:

**定理 7.1** 所有 GA 算法的局部极值点都是函数的一元极值点。使用 GA 算法进行优化时一定能够达到局部极值点。

因为局部极值点的适应度大于与其 Hamming 距离为 1 的所有点的适应度, 也就是大于这一点的实数变量值的邻近点, 也就是一般意义上的函数极值点。由上面的分析过程可知此定理成立。

使用 Gray 编码具有上述的特点, 因此以后有必要研究使用 Gray 编码与自然编码的区别和特点。

### 4.7.4 交叉算子的变化

在交叉算子的改进中, 一些研究者提出了两点交叉, 多点交叉等方法。在上面的证明过程中可以看出, 是交叉算子的引入增加了极值点条件中的  $inh$  项。交叉算子的不同并不影响上面的证明, 而只是改变了  $inh$  函数的数值。 $inh$  值的增加意味着局部极值点数目的减少。下面用下标标记不同的  $inh$  函数, 一般来讲, 两点以上的交叉算子对于  $inh$  函数起了平滑的作用, 即若  $inh_0$  函数值大的时候,  $inh_n$  的函数值减少; 而当  $inh_0$  函数值小的时候,  $inh_n$  的函数值增加。这样看来, 多点交叉提高了算法的探索性而牺牲了算法的开发性。

### 4.7.5 基因连锁

仿照生物基因的交叉方式，也可以采用基因连锁的方式对交叉算子进行改进，也就是限定交叉点的位置，某些点不允许交叉，这是生物遗传中的一个重要规律。使用这种方式可以有效地缩小局部极值点的数目，从而达到全局优化的目的。在连续两个允许交叉的位置之间的编码在遗传进化过程中不被破坏，这也是被模式定理所支持的。

通过本文的证明可知，由于连锁的片断是不可分割的，这将缩短基因型之间的有效 Hamming 距离，因此增加了 *inh* 函数的函数值。使原来的一些非全局极值点消失。

对于基因连锁的作用，已经有一些研究<sup>[91]</sup>。使用这种方法，只要连锁片断不是很长，就不会减少群体的多样性，通过实验证明基因连锁的方法是有效的。这种方法的缺点是要事先确定连锁的位置，以及带来附加的计算量。

## 4.8 小结

上面首先针对使用适应度比例选择算子和单点交叉算子的 SGA，在种群规模足够大和没有变异算子的情况下，讨论了遗传算法的收敛性和收敛目标问题。证明遗传算法能够收敛到某一点的充要条件是这一点为符合定义(4.3)与(5.1)的局部极值点。这个结论是在完全包含交叉算子的情况下得到的，与实际使用的 GA 更加接近。

上述结论的证明的核心在 4.5 节的开始列出。然后分析了其他各种变形的 GA。对于存在代间隔的情况，本文证明了代间隔不影响算法的收敛目标。对于其他的选择算子，同样证明了类似的结论。并通过实例指出，在不同的选择策略下，局部极值点的出现条件的不同。同时又证明了在无限群体情况下，线性排名选择策略与其中的参数无关，并且与规模为 2 的锦标赛选择策略相等价。最后，通过理论的分析，说明了一些改进方法的理论依据和相互之间的关系。

本文使用动力系统的方法来研究遗传算法的运行机理，讨论了算法在

吸引点局域的动力学形态，由于没有涉及到算法的大范围运行轨道，尚未得到算法收敛到全局极值点的充要条件；同时，在实现计算过程中，由于群体规模的限制和变异算子的引入，对算法的运行机理产生了随机的影响，这些都需要作进一步的研究。

## 第五章 遗传算法的全局动力学形态

### 5.1 全局动力学形态

要了解遗传算法的运行机理，不仅需要知道它的极限行为，而且需要了解算法在未收敛时的运行形态。简单地说，就是非平凡不动点的存在性和收敛性。

由于 GA 中算子的非线性作用，使得其全局动力形态极为复杂。其中非全局吸引子、极限环和混沌等行为的存在性尚未得到确认。文献<sup>[135]</sup>中举出一个例子，说明如使用一种不自然的，在实际中从未应用过的变异和交叉算子，GA 会产生稳态循环行为。对于 1-bit 问题，在密度变异（Density-Dependent Mutation）算子下，存在混沌现象<sup>[136]</sup>。

如果考虑 GA 特有的算子，建立全局动力描述，将是一个非常复杂的工作。因此很多分析工作建立在某一个特定的简化问题之上。如 One-Max 问题，以及 BigInt 问题<sup>[112]</sup>等。本章中针对一个简化的 2-bit 问题来进行分析。这个问题可以看作是 Two-Max 问题的一个特例<sup>[1]</sup>。它可以反映两个局部极值点竞争的情况，具有很强的代表性。

本文针对 2-bit 问题，全面分析了 GA 的全局动力学形态。将遗传算法中常用的各种进化算子都进行了形式化描述，并按照实际使用的方式进行结合。通过对标准遗传算法的各种参数的选取，分别建立了数学模型。分析了这些模型的吸引子，揭示了不同进化算子对动力学形态的影响。对吸引子的收敛区间的计算，讨论了不同局部极值点之间的竞争关系。从全局说明了算法的收敛性。

### 5.2 2bit 问题的动力学模型

由于一般实际问题的解空间过大，为分析 SGA 的动力学特性，我们构造了一个 2-bit 问题，其基因型空间  $O=[0,1]^2$ ，其中点 01 和 10 的适应度值相等，设为 1， $f(00)=s_1$ ， $f(11)=s_2$ 。如果  $s_1 < 1$ ， $s_2 > 1$ ，此问题具有唯一极值



点。若  $s_1 > 1, s_2 > 1$ ，则存在一个全局极值点和一个局部极值点。

这个问题具有相当的普遍性。很多实际问题在极值点（全局和局部）附近都可以约化为此问题。尤其是当全局极值点和局部极值点都存在时，讨论此模型可以了解极值点之间的竞争方式与收敛区间，所得到的结果对于一般问题有很强的参考意义。

同样，我们在无限群体上讨论此问题，针对标准遗传算法（SGA）建立动力学模型。记  $p_i(t), 0 \leq t \leq 3$  为每种基因型出现的比例，当使用分量表示时，另记为  $x \leq p_0, y \leq p_3, r_1 \leq p_1, r_2 \leq p_2$ 。

SGA 的迭代过程包括以下几步：

1. 从当前群体中选出一定比例  $g$  的群体进行进化操作。称  $g$  为代间隔。
2. 按照基于适应度比例的原则（如轮盘赌选择）选出父代个体。
3. 按照交叉概率  $c$  对父代个体进行交叉操作，其他不加变更，得到子代。
4. 按照变异概率  $\mu$  对子代个体进行变异操作。

上面涉及到的三个参数  $g, c, \mu$  就是 SGA 的调节参数。

下面据此对每一步建立数学模型。

群体的平均适应度为  $\bar{f} = \sum_{i=0}^3 f_i p_i = p_0 s_1 + p_1 + p_2 + p_3 s_2$  (2.1)

经过选择算子作用后，各种基因型的比例变为：

$$S(p)_i = p_i \cdot f_i / \bar{f} \quad (2.2)$$

对于 2-bit 问题，最常用的单点交叉算子的交叉矩阵  $C$  为：

$$C = \frac{1}{4} \begin{vmatrix} 4 & 2 & 2 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad (2.3)$$

使用分量表示为：

$$p_i^{\alpha} = p_i + p_i \cdot p_{i3}^{\epsilon} - p_{i1}^{\epsilon} \cdot p_{i2}^{\epsilon} \quad \text{for } 0 \leq i \leq 3 \quad (2.4)$$

$$\begin{aligned}
 c(x) &= x^f - x^f y^f + 1^f y^f 2^f \\
 \text{即 } c(y) &= y^f - x^f y^f + 1^f y^f 2^f \\
 c(1) &= 1^f y^f x^f y^f + 1^f y^f 2^f \\
 c(2) &= 2^f y^f x^f y^f + 1^f y^f 2^f
 \end{aligned} \tag{2.5}$$

常用的变异算子有两种，一种 1-bit 变异，另一种是  $c/l$  变异。下面讨论 1-bit 变异，也就是按照变异概率确定待变异的个体后，均匀的从中选取一位进行翻转。设每一位的变异概率为  $\mu$ ，则变异矩阵为：

$$M_m = \begin{pmatrix} 1 & 2 & ? & ? & 0 \\ ? & 1 & 2 & 0 & ? \\ ? & 0 & 1 & 2 & ? \\ 0 & ? & ? & 1 & 2 \end{pmatrix} \tag{2.6}$$

变异算子的作用使用矩阵表示为： $M(p) = M_m p$  (2.7)

例如：对于 2-bit 问题， $p_0$  经变异后的比例为：

$$M(x) = (1 \ 2) x + r_1 + r_2 \tag{2.8}$$

综合上面三个算子的作用，即可得到 SGA 下 2-bit 问题的动力学模型：

$$SGA(p) = C \circ M \circ S(p) \tag{2.9}$$

由于我们主要观察最大值和局部极大值的动力学特性，注意到对于这三个算子，若在某一代，01 点和 10 点的比例相同，则经过一次迭代后的比例仍然相同。即若  $p_1 = p_2$ ，则  $C \circ M \circ S(p_1) = C \circ M \circ S(p_2)$  (2.10)。因此如果 01 点和 10 点在初始群体中的频率相同，则在之后的任一代中都将保持相同。下面不妨设  $r_1 = r_2 = r/2$ 。这样，交叉算子(2.5)的形式变为：

$$\begin{aligned}
 c(x) &= x + x y + r^2/4 + \frac{1}{4}(1 - x - y)^2 \\
 c(y) &= y + x y + r^2/4 + \frac{1}{4}(1 - y - x)^2 \\
 r &= 1 - x - y
 \end{aligned} \tag{2.11}$$

变异算子(2.7)变为

$$\begin{aligned} \dot{m}(x) &= (1 - 3x)xy \\ \dot{m}(y) &= (1 - 3y)xy \end{aligned} \quad (2.12)$$

上面介绍了在所有三个遗传算子作用下群体比例向量的变化，接下来使用这些公式对 SGA 进行建模和分析。

## 5.3 SGA 的模型及分析

下面，分别对于不同情况，建立使用 SGA 解决 2-bit 问题的动力学模型，并分析不同模型下的全局动力学形态。

### 5.3.1 . 全部重组，没有变异的模型

下面首先分析一个最简化的 GA 模型：没有变异算子，代间隔为 1，交叉概率为 1 的情况。即  $g = 1, c = 1, \mu = 0$ 。综合(2.2)、(2.11)，总的动力学模型如下：

$$\begin{aligned} \dot{f} &= x(n)s_1 + y(n)s_2 - (1 - x(n) - y(n)) \\ \dot{x}(n+1) &= \frac{1}{4} \left( 1 + \frac{x(n)s_1}{f} - \frac{y(n)s_2}{f} \right)^2 \\ \dot{y}(n+1) &= \frac{1}{4} \left( 1 + \frac{x(n)s_1}{f} - \frac{y(n)s_2}{f} \right)^2 \end{aligned} \quad (3.1.1)$$

这是一个 2 维动力学模型。但此模型存在一个特点，可以将其转化为一维问题。

#### 引理 3.1

模型(3.1.1)满足  $(x(n) - y(n))^2 = 2(x(n) - y(n)) - 1 \leq 0$  for  $n \geq 1$  (3.1.2)

证明：

计算可知：

$$x(n+1) - y(n+1) = x'(n) - y'(n), x(n+1) - y(n+1) = \frac{1}{2}(x'(n) - y'(n))^2 \leq \frac{1}{2}。其中 x', y'$$

表示经过选择算子后的比例值。因此(3.1.2)成立。

从中可以解出  $y = x - 1 \pm \sqrt{x}$  (3.1.3)，代回(3.1.1)式得

$$f(x(n)) = \frac{1}{4} \left( 1 + \frac{x(n)s_1}{f} + \frac{(x(n)+1+\sqrt{x(n)})s_2}{f} \right)^2 \quad (3.1.4)$$

为求这个动力系统的不动点，解方程  $x(n+1) = x(n)$  得

$$x = \begin{cases} 0 \\ 1 \\ x_s = \frac{(s_2+1)^2}{(s_1+s_2+2)^2} \end{cases} \quad (3.1.5)$$

上式说明此动力系统存在三个不动点，0 和 1 点的存在与我们的直观相符。第三个不动点一般在仿真模拟运算中不能发现。对于模型(3.1.1)的全局动力学特性，由如下定理阐述：

### 定理 3.2：

动力系统(3.1.1)在  $s_1 > 1, s_2 > 1$  的情况下只存在 3 个不动点，如式(3.1.5)所示。其中 0 和 1 点是吸引不动点， $x_s$  点是排斥不动点。若  $s_1 < 1$  或  $s_2 < 1$ ，只存在 2 个不动点 0 和 1；当  $s_2 < 1$  时，1 点是吸引点，0 点是排斥点；当  $s_1 < 1$  时，0 点是吸引点，1 点是排斥点。每种情况下，所有的点都趋向于吸引点，不存在其他的周期点。

证明：

上面已经计算出三个不动点。首先确定第三个不动点的存在条件，这要求  $0 < x_s < 1$  且  $0 < y(x_s) < 1$ ，计算可知，这相当于要求  $s_1 > 1, s_2 > 1$  (3.1.6)。

计算在这三个不动点处的导数：

$$\lim_{x \rightarrow 0} \frac{dg(x)}{dx} = \frac{1}{s_2}, \lim_{x \rightarrow 1} \frac{dg(x)}{dx} = \frac{1}{s_1}, \left. \frac{dg(x)}{dx} \right|_{x=x_s} = \frac{2s_1s_2 + s_2 + s_1}{s_1s_2 + 1} \quad (3.1.7)$$

注：在计算 0 点处的导数时，由于直接的导函数在此处是奇点，因此需要求导数的极限。

前两个不动点的导数很容易分析，为分析第三个点，注意到

$$\frac{2s_1s_2 + s_2 + s_1}{s_1s_2 + 1} > 1 \Leftrightarrow (s_1 + 1)(s_2 + 1) > 0$$

因此当满足(3.1.6)，第三个不动点存在时，其导数一定大于 1，因此此

点一定是排斥性的。

由于当  $x < x_s$  时,  $g(x) < x$ ; 当  $x > p$  时,  $g(x) > x$ 。即当  $x \neq p$  时  $|g(x) - 0| < |x - 0|$ ; 当  $x \neq p$  时  $|g(x) - 1| < |x - 1|$ , 也就是说, 所有的点都向吸引点移动。

证毕

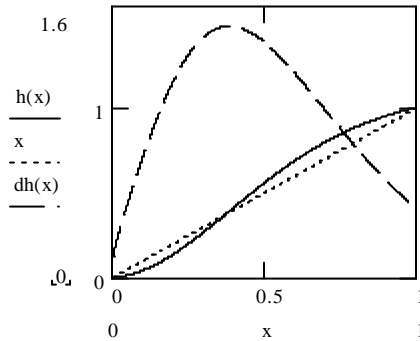


图 5.1 模型(3.1.1)的不动点及其导数

如果只有一个极值点, 那末此模型直接收敛到这个点。这是一个比较平凡和显而易见的结论。在本节以下和以后的模型中, 重点研究存在两个局部极值点的情况, 即  $s_1 \neq 1, s_2 \neq 1$ 。

存在两个局部极值点时, 两个吸引点的收敛范围是不一样的。收敛到那个极值点由初始条件决定, 下面讨论两个吸引点在整个空间上的收敛域。

### 定理 3.3 :

当存在第三个不动点时, 如果随机确定初始化群体, 算法收敛到  $y=1$  那个极值点的概率由下式决定:

$$ar(s_1, s_2) = \begin{cases} \frac{2 \cdot s_2^2 (s_2 - 1)^2}{(s_1^2 - s_2^2 - s_1 - s_2)(2s_2^2 - 3s_2 - s_1)}, & \text{when } s_1 \neq s_2 \\ 1 - \frac{2 \cdot s_1^2 (s_1 - 1)^2}{(s_1^2 - s_2^2 - s_1 - s_2)(2s_1^2 - 3s_1 - s_2)}, & \text{else} \end{cases} \quad (3.1.8)$$

证明:

$$\text{在第三个不动点处, } y_s = \frac{(s_1 - 1)^2}{(s_1 - s_2 - 2)^2}, x_s = y_s = \frac{s_2 - s_1}{s_1 - s_2 - 2} \quad (3.1.9)$$

所有满足  $x \leq y \leq x_s \leq y_s \leq \frac{s_2 \wedge s_1}{s_1 \wedge s_2 \wedge 2}$  的点经过一次迭代后转移到排斥

点。计算此方程，得  $\frac{x' s_1}{f} ? \frac{y' s_2}{f} ? \frac{s_2 ? s_1}{s_1 ? s_2 ? 2}$

即：

$$ls(x, y) \leq x(2 \leq s_1 \leq 3 \leq s_1 \leq s_2) \leq y(2 \leq s_2 \leq 3 \leq s_2 \leq s_1) \leq s_1 \leq s_2 \leq 0 \quad (3.1.10)$$

这是一个直线方程，如图 2 中的虚线所示，此直线将状态空间划分为两部分，在直线上方的点经过一次迭代后将转移到不动点 S 的上方，在以后的迭代中向此区域中的吸引点逼近。反之亦然。

因此当随机确定初始化群体时, 算法收敛到  $y=1$  那个极值点的概率就是直线  $l_s$  的上方在整个状态空间  $OXY$  中的比例

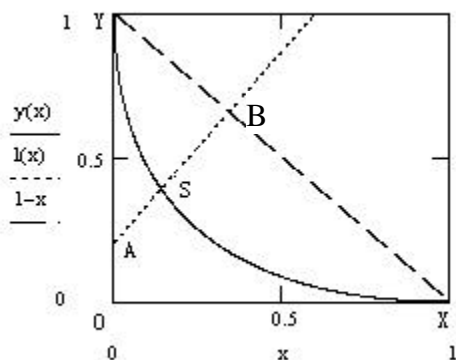


图 5.2 模型 1 的收敛区域与分界线( $s_1=1.5, s_2=1.3$ )

当  $s_1 > s_2$  时, 直线通过  $y$  轴的上方, 计算  $y$  轴截距得  $A_y = \frac{s_1 + s_2}{2s_1 + 3s_2 + s_1}$ ,

以及此直线与  $x+y=1$  的交点 B 的 x 坐标  $B_x = \frac{s_2 \cdot (s_2 + 1)}{s_1^2 + s_2^2 + s_1 + s_2}$ 。

上方三角形面积等于

$$S_{\gamma ABY} \leq \frac{1}{2}(1 \otimes A_y) \otimes B_x \leq \frac{s_2^2(s_2 \otimes 1)^2}{(s_1^2 \otimes s_2^2 \otimes s_1 \otimes s_2)(2s_2^2 \otimes 3s_2 \otimes s_1)} \quad (3.1.11)$$

因此，状态空间中，收敛于  $x=0, y=1$  点的部分占整个状态空间的比例，也就是直线(3.1.9)上方的面积占整个三角形 XOY 的比例，就是公式(3.1.8)的上半部分。

同样当  $s_1 \leq s_2$  时，直线  $l_s$  在 XOY 内与  $x$  轴相交，相似的计算可得：

$$S_{ABX} = \frac{s_1^2 (s_1 + 1)^2}{(s_1^2 + s_2^2 + s_1 + s_2)(2s_1^2 + 3s_1 + s_2)} \quad (3.1.12)$$

这样就可得出公式(3.1.8)的下半部分。证毕

### 5.3.2 . 世代重叠模型

下面讨论简单的世代重叠模型。这里  $g < 1$ ，其他两个参数仍然是  $c=1, \mu=0$ 。

此模型的数学表示为：

$$\begin{aligned} f &= x(n)s_1 + y(n)s_2 + (1 - x(n) - y(n)) \\ x(n+1) &= \frac{1}{4}g \left( 1 + \frac{x(n)s_1}{f} + \frac{y(n)s_2}{f} \right)^2 + (1-g)x(n) \\ y(n+1) &= \frac{1}{4}g \left( 1 + \frac{x(n)s_1}{f} + \frac{y(n)s_2}{f} \right)^2 + (1-g)y(n) \end{aligned} \quad (3.2.1)$$

首先求解此模型的不动点，令  $x(n+1) = x(n), y(n+1) = y(n)$ ，可以得出与求解模型(3.1.1)时相同的方程，而与参数代间隔  $g$  无关。因此此模型的不动点与模型(3.1.1)相同。即：

$$\begin{array}{l|l} x & 0 \\ & 1 \\ x_s & \frac{(s_2 + 1)^2}{(s_1 + s_2 + 2)^2} \end{array}$$

但是，此模型的动力学形态与(3.1.1)不同。由于它不能满足(3.1.2)，也不能通过其他方式转变为 1 维问题，因而这是一个 2 维动力学问题。当讨论不动点的吸引性时，需要求不动点处 Jacobin 矩阵和特征值。

**定理 3.4：**

模型(3.2.1)的不动点中，0 点和 1 点都是吸引点，而  $x_s$  点是鞍点。

证明：可以求出(3.2.1)的 Jacobin 矩阵。由于其过分复杂，具体形式这里就不再列出。

在(x=1,y=0)的矩阵为：

$$J_{10}(s_1, s_2, g) = \begin{pmatrix} 1-g & g/s_1 & g(1-2s_2)/s_1 \\ 0 & 1-g & 0 \end{pmatrix} \quad (3.2.2)$$

其特征值为  $\lambda_1 = 1-g$  和  $\lambda_2 = 1-g$ 。

在(x=0,y=1)的矩阵为：

$$J_{01}(s_1, s_2, g) = \begin{pmatrix} 1-g & 0 \\ g(1-2s_2)/s_1 & 1-g-g/s_2 \end{pmatrix} \quad (3.2.3)$$

其特征值为  $\lambda_1 = 1-g$  和  $\lambda_2 = 1-g-g/s_2$ 。

这两种平凡情况下，所有特征值都小于 1，因此不动点都是吸引的。

在第三个不动点处，矩阵的具体形式相当复杂，但可以求出矩阵的行

$$\text{列式的值为：} \|J_s\| = \frac{(1-g)(gs_1s_2 - g(s_1 - s_2) - s_1s_2 - g)}{s_1s_2 - 1} \quad (3.2.4)$$

$$\text{矩阵的迹为 } \text{tr} J_s = \frac{2s_1s_2 - gs_1 - gs_2 - 2g}{s_1s_2 - 1} \quad (3.2.5)$$

设矩阵的两个特征值为  $\lambda_1, \lambda_2$ ，则

$$(\lambda_1 - 1)(\lambda_2 - 1) = \lambda_1\lambda_2 - (\lambda_1 + \lambda_2) + 1 = \|J_s\| - \text{tr} J_s + 1$$

$$\text{计算得：} \|J_s\| - \text{tr} J_s + 1 = \frac{g^2(s_1 - 1)(s_2 - 1)}{s_1s_2 - 1} \quad (3.2.6)$$

显然上式小于零，因此两个特征值一个大于 1，一个小于 1。这样，这个不动点是一个鞍点。证毕

在模型(3.2.1)这个二维动力系统中，由于存在一个鞍点，按照稳定流形定理，在此点存在一条稳定流形曲线和一条不稳定流形曲线。不稳定流形的具体形式难以求得，稳定流形由如下定理所示。

### 定理 3.5

直线  $ls(x, y) = x(2s_1^2 - 3s_1 - s_2) - y(2s_2^2 - 3s_2 - s_1) - s_1 - s_2 = 0$  是动力学模型(3.2.1)通过鞍点的稳定流形。

证明：



这条直线就是上面提到的(3.1.10)，直线通过模型的鞍点。并且满足

$$\frac{x-s_1}{f} - \frac{y-s_2}{f} = x_s - y_s = \frac{s_2 - s_1}{s_1 - s_2 - 2}。与定理中的方程等价，可另写为：$$

$$(s_1^2 - s_2^2 - s_1 - s_2)(x - y) + (s_1 - s_2)(s_1 - s_2 - 2)(x - y) + s_1 - s_2 = 0 \quad (3.2.7)$$

在此模型中，有

$$\begin{aligned} u(n) &= \frac{x(n)-s_1}{f} - \frac{y(n)-s_2}{f} \\ x(n+1) - y(n+1) &= g[u(n) + (1-g)(x(n) - y(n))] \\ x(n+1) - y(n+1) &= g\left(\frac{1}{2} - \frac{1}{2}u(n)^2\right) + (1-g)(x(n) - y(n)) \end{aligned} \quad (3.2.8)$$

若  $x(n), y(n)$  满足定理中的直线方程，将(3.2.8)代入(3.2.7)，计算后仍然满足  $ls(x(n+1), y(n+1)) = 0$ 。因此此直线是模型的一条不变曲线。

$$\text{在鞍点处，令 } v_s = x_s - y_s = \frac{(s_1 - 1)^2 - (s_2 - 1)^2}{(s_1 - s_2 - 2)^2}, \quad v(n) = x(n) - y(n) - v_s$$

$$\text{计算得：} v(n+1) = g(1 - u(n)^2)/2 + (1-g)(v(n) - v_s) + (1-g)v(n)$$

因此当  $n \rightarrow \infty$  时， $v(n) \rightarrow 0$ ，所以在这条直线上的迭代将趋近于鞍点。

证毕

这样，世代重叠的模型的全局动力学形态如图 3 所示，包括两个吸引点，一个鞍点，和由稳定流形划分出的两个区间。在每个区间内，所有点都收敛于这个区间中的吸引点。由于稳定流形与模型(3.1.1)的分界线一致，因此，收敛到某一个吸引点的比例也与(3.1.8)相同。

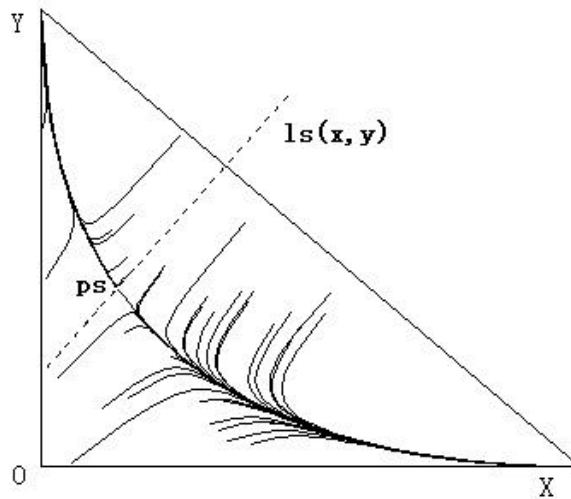


图 3.3 模型 2 的全局动力学形态

### 5.3.3 . 部分交叉模型

下面讨论部分交叉的模型，这里  $c < 1$ ，其他两个参数仍然是  $g=1, \mu=0$ 。此模型的数学表示为：

$$\begin{aligned} f &= x(n)s_1 + y(n)s_2 + (1 - x(n) - y(n)) \\ x(n+1) &= \frac{1}{4}c \left(1 + \frac{x(n)s_1}{f} + \frac{y(n)s_2}{f}\right)^2 + (1-c)\frac{x(n)s_1}{f} \quad (3.3.1) \\ y(n+1) &= \frac{1}{4}c \left(1 + \frac{x(n)s_1}{f} + \frac{y(n)s_2}{f}\right)^2 + (1-c)\frac{y(n)s_2}{f} \end{aligned}$$

为求此模型的不动点，变换坐标为：

$$\begin{aligned} u(n) &= x(n) + y(n) \\ v(n) &= x(n) - y(n) \end{aligned} \quad (3.3.2)$$

在新的坐标下，

$$\begin{aligned}
 & f(s_1, s_2) = (u(s_1, s_2) + v(s_1, s_2)) / 2 \\
 & u(n+1) = \frac{x(n)s_1}{f} + \frac{y(n)s_2}{f} + \frac{1}{2}[u(s_1, s_2) + v(s_1, s_2)] / f \quad (3.3.3) \\
 & v(n+1) = \frac{1}{2}(1-c)[u(s_1, s_2) + v(s_1, s_2)] + \frac{1}{2}c(u^2(n+1) + 1)
 \end{aligned}$$

求解此方程的不动点，得：

$$\begin{aligned}
 & d(s_1, s_2) = (s_1 s_2 + 1)^2 c^2 + 4(s_1 + 1)^2 (s_2 + 2)^2 (1-c) \\
 & u_s = \frac{s_2 + s_1}{c(2s_1 s_2 + s_1 + s_2)(s_1 + s_2 + 2)} [(2-c)(s_1 + 1)(s_2 + 1) + \sqrt{d(s_1, s_2)}] \quad (3.3.4) \\
 & v_s = u_s \frac{s_2 + s_1 + u_s s_1 + u_s s_2 + 2}{s_2 + s_1 + u_s s_1 + u_s s_2 + 2u_s}
 \end{aligned}$$

以及平凡不动点：

$$\begin{aligned}
 & u = 1, v = 1 \\
 & u = 1, v = 1
 \end{aligned}$$

由于满足  $x(n+1) = y(n+1) = (x(n)s_1 + y(n)s_2) / f$ ，因此经过鞍点的直

线  $ls_4(x, y) = (x s_1 + y s_2) / f = u_s$  (3.3.5) 也是它的不稳定流形。此模型和上

一节相比，除了分界线的具体方程不一样外，全局的动力学形态是相似的。

和完全交叉的模型(3.2.1)相比，两个模型的收敛区域是不同的，有如下定理：

### 定理 3.6

若参数  $s_1, s_2$  相同，当  $s_1 < s_2$  时，不完全交叉模型(3.3.1)收敛到  $x=1, y=0$  的区域比完全交叉的模型(3.2.1)小，反之亦然。

证明：

下面证明  $s_1 < s_2$  的情况。记  $u_0 = \frac{s_2 + s_1}{s_1 + s_2 + 2}$  为模型(3.2.1)不动点处  $x=y$

的值（参见公式 3.1.9），此时  $u_0 > 0, u_s > 0$ 。首先要证明  $u_s / u_0 > 1 > 0$ 。计算得：

$$u_s / u_0 > 1 - 4c(1-c)t_1 t_2 (t_1 + t_2 + 2t_1 t_2) > 0$$

上式中,  $t_1 \neq s_1 \neq 1, t_2 \neq s_2 \neq 1$

这样, 直线(3.3.5)处于直线(3.1.10)的下方(因为这两条直线的斜率相同)。因此收敛区域也小。

相反的情况也同样可以证明。证毕

### 5.3.4 . 带有变异算子的模型

下面讨论有变异算子的情况。这里仍然只考虑一个参数  $\mu > 0$ , 其他两个参数仍然是  $g=1, c=1$ 。

$$\begin{aligned} f &= x(n) \cdot s_1 + y(n) \cdot s_2 - 1 - x(n) - y(n) \\ x_c &= \frac{1}{4} \left( 1 - \frac{x(n) \cdot s_1}{f} - \frac{y(n) \cdot s_2}{f} \right)^2 \\ x(n+1) &= (1 - 3\mu) x_c + \mu y_c \\ y_c &= \frac{1}{4} \left( 1 - \frac{x(n) \cdot s_1}{f} - \frac{y(n) \cdot s_2}{f} \right)^2 \\ y(n+1) &= (1 - 3\mu) y_c + \mu x_c \end{aligned} \quad (3.4.1)$$

这个模型同样可以简化为一维模型, 上式满足:

$$(x - y)^2 / k + 2 * (x - y) - 1 \geq 0, \text{ 其中 } k = \frac{(1 - 2\mu)^2}{1 - 4\mu} \quad (3.4.2)$$

从中可以解出  $y = x + k \pm \sqrt{4kx + k^2 - k}$  (3.4.3), 代回即可得到一维模型如下:

$$\begin{aligned} y(n) &= x(n) + k \pm \sqrt{4kx + k^2 - k} \\ f &= x(n) \cdot s_1 + y(n) \cdot s_2 - 1 - x(n) - y(n) \\ z &= (x(n) \cdot s_1 + y(n) \cdot s_2) / f \\ x(n+1) &= \frac{1}{4} (1 - 3\mu) (1 - z)^2 + \frac{1}{4} (1 - z)^2 \end{aligned} \quad (3.4.4)$$

求解这个模型的不动点的解析表达式是相当复杂的, 下面使用一组数据讨论数值计算的结果。设  $s_1 = 1.5, s_2 = 1.2$ 。

当变异概率很小时，如  $\mu = 0.01$  时，模型的不动点为： $x = (0.0058, 0.053, 0.941)$ 。注意到其中两个是很接近于前面几个没有变异算子模型的平凡不动点(0 和 1)的。计算在不动点处的导数得  $dx = (0.915, 1.071, 0.7)$ 。中间一个仍然是排斥性的。

当变异概率逐渐增大时，系统的动力学特性发生了变化，由原来的三个不动点变为一个不动点。

当  $\mu = 0.2$  时，系统的不动点为  $x = 0.311, y = 0.193$ 。在不动点处的导数也小于 1。因此系统的动力学形态随着变异概率的增加发生了突变。这样，当变异概率足够大时，算法就不能收敛到通常的某个局部极值点附近，而收敛到一个所有基因型共存的状态。

这两种情况时，模型(3.4.4)的形状如图 3.4 所示。

当代间隔  $g < 1$ ，变异概率  $\mu > 0$  时，系统的模型是两者的结合。当变异概率很小时，模型与上面的 (3.2.1) 类似，其全局动力学状态与图 3.3 也类似。当变异概率足够大时，也只保留了一个吸引不动点，如图 3.5 所示。

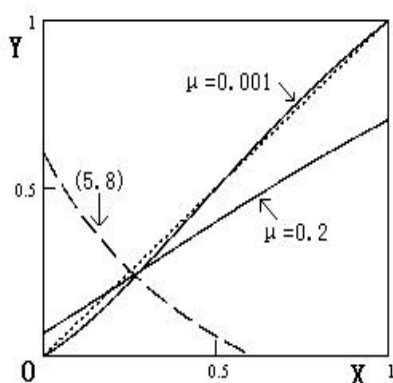


图 3.4 函数(3.4.4)的形状及不动点

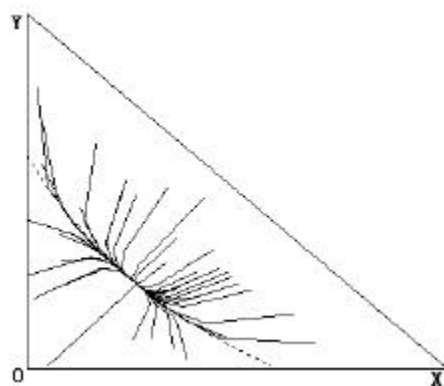


图 3.5 变异算子较大的世代重叠模型

## 5.4 小结

上面对于 2-bit 问题, 分析了使用 SGA 算法的 4 个不同的动力学模型, 对于每一个模型分析了全局动力学形态, 全面证明了这个问题的收敛性和收敛方式。总的来说, 2-bit 问题的全局形态比较简单, 当存在两个局部极值点时, 对应的数学模型有两个吸引点和一个鞍点 (对于模型 1 来说是排斥点), 除此之外不存在其他的不动点或周期点。算法能够收敛到哪一个极值点完全由初始条件决定, 并且对初值不敏感。不同的模型中相应的收敛区域的比例只取决于算法参数的选取。

从上面的分析可以看出, 遗传算法中的各个进化算子对于动力学形态的影响各不相同。适当选取代间隔不影响收敛区域的大小, 只是增加了模型的维数。部分交叉改变了收敛区域的大小, 使得适应度大的极值点更容易达到。当变异概率很小时, 模型与没有变异算子相类似。当变异算子足够大时, 模型的动力学形态随着变异概率的增加发生了突变。原有的吸引不动点消失, 原来的排斥不动点变成了吸引的。上述结论有助于加深对于 GA 的动力学形态的理解。对于参数的选取具有一定的参考作用。对于更一般性的问题, 其全局动力学形态会更加复杂, 可能出现其他的特性。在以后的研究中, 将试图将这种方法扩展到更具一般性的问题上, 以获得更加一般性的结论。

## 第六章 遗传算法取代时间的分析

### 6.1 选择压力

在传统遗传算法的执行过程中，变异算子给群体中带来新的基因已恢复由于选择算子的作用而失去的个体多样性；杂交算子对群体内现有的信息进行重组已发现与环境更为适应的个体，而选择则起着向导的作用以使搜索朝着搜索空间中的可能最优区域进行。综合起来看，杂交与变异的作用是勘探搜索空间以寻找那些可能最优区域，它以保持群体内的多样性为主要目的；而选择的作用是开采搜索空间以充分利用群体内当前所具有的有效信息，它使算法的侧重点放在那些具有较高适应度的个体上。在求解复杂的优化问题上，如何在勘探和开采之间进行有效的权衡是使遗传算法获得较高性能的关键性问题。这时，常用术语“选择压力(selection intensity)”来进行刻画。

选择策略对算法性能具有很大的影响。不同的选择策略的选择压力不同。较大的选择压力使最优个体具有较高的复制数目，从而使算法的收敛速度加快，但也较容易出现过早收敛的问题。相对而言，较小的选择压力一般能使群体保持足够的多样性，从而增大了算法收敛到全局最优的概率，但算法的收敛速度降低。

### 6.2 选择算子与取代时间

常用的选择策略有基于比例的选择策略、基于排名的策略和基于局部竞争的选择策略等。本文着重研究基于比例的选择。

最直接的基于比例选择策略是基于适应度比例的选择策略，它根据个体的适应度值和群体的平均适应度之比来进行选择。每个个体的被选择概率等于其适应度除以群体的平均适应度。即

$$f_{avg} = \frac{1}{N} \sum_{i=0}^{N-1} f_i, p_i = \frac{f_i}{f_{avg}} \quad (2.1)$$

基于排名的选择可以看作是适应度变换的比例选择。在这种方式下，首先根据个体的适应值在群体中的排名来分配其选择概率，然后使用这个概率进行转盘选择。这样个体的绝对适应值不影响后代的数量，不需要进行适应度函数的标准化和调节。

为比较选择算子的选择压力，一个重要方法是计算取代时间。

在文献<sup>[24]</sup>中，给出的取代时间的定义如下：

设初始群体为  $P(0)$ ，其规模为  $N$ ；其中包含唯一的最优个体为  $x^*$ ，选择算子  $s$  的取代时间为：

$$t^* = \min\{P(t) = s^t(P(0)) = \{x_i = x^*\}\} \quad (2.2)$$

即初始群体  $P(0)$  在经过选择算子作用  $t$  次后，得到的群体  $P(t)$  中所有个体都是最优个体的最短时间。

后来，文献<sup>[98]</sup>中对此定义进行了改进

设群体规模为  $N$ ，最优个体在  $t$  代后的比例为  $p^*(t)$ ，则选择算子的取代时间为使得  $p^*(t) = 1 - \frac{1}{N}$  成立的最小代数。

在上述文献中，分别使用这两种定义对特定适应度函数和不同选择算子的取代时间进行了计算。

由于有限群体的随机作用，在有限群体中为使最优个体在演化过程中不灭绝，需要使用择优策略，即在演化的过程中保存最优个体。同时，上述定义中的取代时间实际上是一个随机变量，而并不是一个确定的值。在理论分析中，可以求这个公式的期望值作为取代时间。这种计算是很复杂的。为此，应在无限群体中讨论。这样，不需要择优策略，并根据大数定理，可以直接计算这个值。

在考虑取代时间的计算时必须明确两个限制，即如何定义适应度足够大的群体和它们所占的比例。本文认为取代时间应如下定义：

**定义 2.1：** 设群体中适应度高于某个界限的个体集合  $o(x) = \{x > M, f(x) > f(M); M < 1\}$  在群体中所占的总比例为  $P(M)$ ，选择算子的取代时间  $t^*$  定义为经过若干次演化后，满足  $P(M) = 1 - \frac{1}{N}$  的最小代数。

**定义 2.2：** 适应度界限为  $f(M) = f(1 - 1/2^l)$ ，即  $o(x)$  为群体中适应度最大的



个体的取代时间称为标准取代时间  $ts^*$ 。

### 6.3 适应度比例选择算子的取代时间

为计算取代时间，首先对模型中的一些概念和符号进行明确说明。

设编码长度为  $l$ ，则共有  $N=2^l$  种个体，可以将这些个体看作是 0 到 1 之间的小数  $x$ （包括 1，不包括 0），它的适应度是  $Fit(x)$ 。在只考虑选择算子的情况下，不妨将这些个体按适应度的升序排列，然后重新编号，并将适应度相同的个体合并，另外将函数归一化（即除以  $Fit(x)$  的最大值）。这时， $Fit(x)$  在  $x$  的  $2^l$  个离散取值点上是单调递增的。在此基础上，将函数在区间  $[0,1]$  连续化，记变换后的函数为  $f(x)$ 。 $f(x)$  有以下性质： $x \in (0,1], f(x) \in (0,1], f'(x) \geq 0, f'(x)$  有界。在个体种类足够多的情况下，第  $n$  代中每个个体所占的比例可以构成一个概率分布，使用  $p_n(x)$  来表示这个分布的概率密度函数， $p_0(x)$  就是初始群体的构成。

首先有：

**引理 3.1：**

使用比例选择算子的情况下，第  $n$  代群体的分布概率密度函数为：

$$p_n(x) = \frac{p_0(x) * f^n(x)}{\int_0^1 p_0(x) * f^n(x) dx} \quad (3.1)$$

证明：

由比例选择算子的定义可知，每一个个体的选择概率等于此个体的适应度除以群体的平均适应度，即  $p_{t+1}(x) = p_t(x) * f(x) / f_{avg}$ ，而群体的平均适应度  $f_{avg}$  在群体足够大的情况下可以用积分  $\int_0^1 p_t(x) * f(x) dx$  表示，因此有：

$$p_{t+1}(x) = \frac{p_t(x) * f(x)}{\int_0^1 p_t(x) * f(x) dx}$$

递推此公式，可知引理成立。

由此引理和定义 3.1 相结合，可知，计算取代时间就是计算满足不等式

$$\frac{\int_0^1 p_0(x) * f^n(x) dx}{\int_0^1 p_0(x) * f^n(x) dx} \approx \frac{1}{N} \text{ 的 } n \text{ 值, 上式可化为, 计算}$$

$$G(n) \approx \frac{\int_0^M p_0(x) * f(x)^n dx}{\int_0^1 p_0(x) * f(x)^n dx} \approx \frac{1}{N} \quad (3.2)$$

例如当初始群体服从均匀分布  $p(x)=1$  适应度函数为  $f(x)=x^b$  ( $b$  为参数) 时可以计算得:

$$t^* \approx \frac{1}{b} * \frac{\ln N}{\ln M} \approx \frac{1}{b} * \frac{\ln N}{1 - M}$$

$$t_S^* \approx \frac{1}{b} * 2^l * \ln N$$

与文献<sup>[98]</sup>中的结果  $t_S^* \approx \frac{1}{b} * N * \ln N$  相比, 上述文献的定义中混淆了群体规模与个体种类两个概念。在那里, 认为群体规模等同于个体数目, 因此认为最优个体 (只有一个) 在整个个体总数中所占的比例为  $1/N$ , 实际上这个比例应该是  $1/2^l$ 。

下面首先讨论当群体的初始分布是均匀分布的情况, 即  $p_0(x)=1$ 。

**定理 3.2 :**

当群体的初始分布是均匀分布时, 比例选择算子的取代时间为  $O(\ln N)$ , 与适应度函数的具体形式无关。

证明:

由第一积分中值定理有:

$$\int_a^b f(x)^k dx \approx \int_a^b \frac{f(x)^k}{f'(x)} df(x) \approx \frac{1}{f'(?) } * \int_a^b f(x)^k df(x) \approx \frac{1}{f'(?) } * \left[ \frac{1}{k+1} * f(x)^{k+1} \right]_a^b$$

其中  $a < \xi < b$

这样, (3.2)式化为:

$$G(n) \approx \frac{\left[ \frac{1}{f'(\xi_1)} * \frac{1}{k+1} * f(x)^{n+1} \right]_0^M}{\left[ \frac{1}{f'(\xi_2)} * \frac{1}{k+1} * f(x)^{n+1} \right]_0^1} \approx \frac{f'(\xi_2) * \frac{f(M)^{n+1} - f(0)^{n+1}}{f(1)^{n+1} - f(0)^{n+1}}}{f'(\xi_1) * \frac{[f(M)]^{n+1} - [f(0)]^{n+1}}{[f(1)]^{n+1} - [f(0)]^{n+1}}}$$

因为  $f(1) > f(M) > f(0)$ ，当  $n$  足够大时， $[\frac{f(M)}{f(0)}]^{n+1} \gg 1, [\frac{f(1)}{f(0)}]^{n+1} \gg 1$ ，

上式继续化为：

$$G(n) \approx \frac{f'(\xi_2)}{f'(\xi_1)} * \frac{[\frac{f(M)}{f(0)}]^{n+1}}{[\frac{f(1)}{f(0)}]^{n+1}} \approx \frac{f'(\xi_2)}{f'(\xi_1)} * [\frac{f(M)}{f(1)}]^{n+1} \quad (3.3)$$

在上式中， $0 < \xi_1 < M$ ， $0 < \xi_2 < 1$ ，由于  $f'(x) \neq 0, f'(x)$  有界，所以

$\frac{f'(\xi_2)}{f'(\xi_1)}$  有界，令  $c = \frac{f'(\xi_2)}{f'(\xi_1)}$ ，当  $n$  足够大时，可以认为  $c$  是常数。

代回(3.2)式，可得取代时间为  $t^* \approx \frac{\ln N * \ln c}{\ln f(1) * \ln f(M)} \approx O(\ln N)$  (3.4)

证毕

下面我们讨论初始群体分布不是均匀分布的情况。

由上面的定理 3.2 可知，取代时间的阶与适应度函数的具体形式无关，在下面的讨论中，不妨设  $f(x) = x$ 。

**定理 3.3：**

比例选择算子的取代时间为  $O(\ln N)$ ，与适应度函数的具体形式无关，与群体初始分布也无关。

证明：

在一般意义上，可以把  $p_0(x)$  写成多项式  $p_0(x) = \sum_{i=0}^k a_i x^i, k \leq 2^l$  的形式

(由于它在  $2^l$  个点上取值，总可以通过多项式插值的方法扩展为次数不大于  $2^l$  的多项式)。因此计算

$$G(n) \approx \frac{\int_0^M p_0(x) * f(x)^n dx}{\int_0^1 p_0(x) * f(x)^n dx} \approx \frac{\int_0^M x^n * \sum_{i=0}^k a_i x^i dx}{\int_0^1 x^n * \sum_{i=0}^k a_i x^i dx}$$

由于  $\int_0^1 x^n * \sum_{i=0}^k a_i x^i dx = \sum_{i=0}^k a_i * \frac{1}{n+1} * x^{n+1} \approx C$

$$\text{所以, } G(n) \approx \frac{\sum_{i=0}^k a_i \cdot \frac{1}{n-i-1} \cdot M^{n-i-1}}{\sum_{i=0}^k a_i \cdot \frac{1}{n-i-1}} \approx M^{n-1} \cdot \frac{\sum_{i=0}^k a_i \cdot \frac{1}{n-i-1} \cdot M^i}{\sum_{i=0}^k a_i \cdot \frac{1}{n-i-1}}$$

由于  $M \geq 1$ , 所以有  $G(n) \approx M^{n-1}$

代回(3.2)式, 仍然有  $t^* \approx \frac{\ln N}{\ln M} \approx 1 \approx O(\ln N)$ 。

证毕

这里也可以看出, 初始群体的分布对取代时间只有附加常数项的影响, 甚至不影响  $\ln N$  的系数。

## 6.4 不同选择算子的取代时间的比较

从上面的证明中可以看出, 基于适应值比例的选择算子的取代时间的阶是一致的, 不因适应度函数的具体形式和初始群体的分布而变化, 只有系数的不同。称此系数为取代时间系数。

### 定义 4.1

对于一个特定的适应度函数, 其标准取代时间  $t_s(N)$  与  $\ln N$  的比值极限

$q \approx \lim_{N \rightarrow \infty} \frac{t(N)}{\ln N}$  称为这种方式下的取代时间系数。对于基于比例的选择策略,

$$q \approx \frac{1}{\ln f(1) \cdot \ln f(M)} \approx \frac{1}{\ln f(1) \cdot \ln f(1/2^l)}。$$

在遗传算法的实际使用时, 经常对适应度函数进行某种变换以获得更好的性能。我们可以从适应度变换对取代时间系数的影响来定量分析这种变换的作用。

由于取代时间系数取决于  $\ln f(1) \cdot \ln f(M)$ 。很显然, 对适应度函数作

乘法变换不影响取代时间系数。如果做加法变换, 由于  $\frac{f(1)+C}{f(M)+C} \approx \frac{f(1)}{f(M)}$ ,

所以将适应度都加上一个正数将缩小取代时间系数, 减慢收敛速度, 反之则加快收敛速度。在有限群体模型中, 使用基于排名的选择策略时, 在迭代一定时间后, 适应度比较小的群体几乎消失, 相当于适应度值减去一个

常数，因此这种策略相对更加有效。对适应度函数作指数变换，这个指数将以乘数的形式作用到取代时间系数上。大于 1 的指数将缩短取代时间。

## 6.5 小结

目前的研究中，大部分的定量分析结果都是建立在某种特定的适应度函数以及均匀的初始分布这种假设基础上，没有取得一般性的结论。对于使用不同的适应度函数对取代时间的影响也没有进一步的说明。

为此，本文给出了取代时间的严格定义。对于基于比例的选择算子，详细地分析了算子的取代时间，证明了取代时间的阶不依赖于适应度函数的选取和初始群体的分布，为  $O(\ln N)$  阶。据此，本文提出了取代时间系数的概念，各种不同类型的适应度函数及其变换将只影响取代时间系数。也就是说，基于比例的选择算子的收敛速度是同阶的。

在以后的研究中，将对于其他的选择策略（如锦标赛选择、小生境选择等）的取代时间的一般表达、有限规模群体中基因随机漂移的影响，以及交叉算子的作用等作更加深入的探讨。

## 第七章 生境遗传算法

### 7.1 背景

目前，遗传算法在各个领域得到广泛应用。随着研究的深入，遗传算法具有的大范围广域搜索能力得到确认。在各种单模函数（即有唯一极值点的函数）中，GA 已被证明是有效的。但是，一些领域需要对多模函数（Multimodal Function）进行优化，识别其中的多个极值点。典型的 GA 只能收敛到一个极值点。如果单纯重复算法的运行，并假定所有的优化解有相同的可能性被找到，在文献<sup>[13]</sup>中指出，独立运行的次数是  $p \sum_{i=1}^p 1/i \approx p (\ln p + \gamma)$ ，其中  $p$  是优化解的数目， $\gamma \approx 0.577$  为 Euler 常数。但是，在多数问题中，优化解多半不等，被发现的概率也不会相同，因此上式只是理想情况。

在很多情况下，并不一定要求找到最优解，而是希望能够了解最优解和准最优解的构造情况，从而根据需要加以取舍。这是因为在计算机上建立的模型都有其不精确性，不能把所有因素考虑在内；有时从成本代价等方面考虑，要求在合理的计算时间内求得与最优解相差无几的准最优解；在有些情况下，则要求得几个方案，然后使用人工或其他的方法进行评价。

典型的 GA 在选择压力和基因漂移的作用下，即使多个极值点具有相等的适应度，最后仍收敛到一个极值点。因此群体难以维持多个优化解，这是 GA 中的一个难题。目前已经提出了各种群体多样化机制来维护在搜索中的多样化群体，从而避免群体收敛到一个极值点。这种研究的兴趣点集中在 GA 得到的极值点的数量和质量上。大部分研究的基础来源于基于自然界中生境（niche）的概念。使用生境机制解决复杂问题的能力已经得到广泛研究<sup>[100]</sup>。

### 7.2 生境机制

在自然界中，整个环境中有多个子空间(niches)可以支持不同类型的物

种。生境中物种的数量取决于生境的负荷能力(carrying capacity)和每一个物种对生境能力的利用程度。如果在一个生境中有太多的物种,将会出现资源缺乏,最次的物种将会死亡。相反,如果在一个很肥沃的生境中只有很少的物种,这些物种将迅速生长来充分利用生境的能力。在资源有限的情况下,群体规模的增長可以使用 Logistic 方程  $N_t = m N_{t-1} (1 - N_{t-1} / K)$  (2.1) 来模拟,其中  $m$  是群体的名义增长率,  $K$  是环境的负荷量。

使用相似的机制, GA 可以在一个多模函数中维护群体的多样性。多模函数的每一个极值点可以看作是一个能够容纳一定数量个体的生境。文献<sup>[86]</sup>中给出了生境的一个数学上的定义:生境(niche)是一个适应度函数值的集合,它们在一些局部搜索算法下达到相同的局部极值点。生境能够负荷的最大个体数量直接和生境的负荷能力成正比,负荷能力由生境峰值点的适应度和域中其他极值点的适应度来决定。令  $q$  是生境的数目,生境规则(Niching rule)  $N_i = f_i / \sum_{j=1}^q f_j$  (2.2)给出了分配给第  $i$  个生境的个体的数量<sup>[88]</sup>,  $f_j$  是这个生境的一种适应度量度,例如最优适应度、平均适应度或适应度之和。这个公式也被称之为生境均衡分布(niche proportionate population)。

使用生境机制实现群体多样性机制的难点在于多模函数定义域上生境的位置、大小和形状是未知的。由于生境位置的未知,正确地按照它们的适应度比例来决定每个生境是一个难题。理想情况下, GA 应该能同时识别出每个生境的位置和相应的比例密度。

下面是一些由各个研究者提出的各种生境机制

### 7.2.1 拥挤

在 De Jong 1975 年最早提出的拥挤机制中,通过替换最接近的父代个体来减少早熟收敛。对于每一个子代,一个确定参数 CF (拥挤因子)的个体从群体中被随机地选中,通过相似性测量,最相似的个体被子代所替换。拥挤机制使一代与下一代之间的群体分布的变化最小。这减缓了 GA 收敛到一个极值点的趋势。缺点在于它阻碍了对于那些与初始分布不接近的区域的探索。后来的研究证明拥挤机制对于多极值点识别问题不是很有效的<sup>[84]</sup>。

### 确定性拥挤 (Deterministic Crowding, DC)

DC 算法是 De Jong 的基本拥挤机制的发展<sup>[84]</sup>。在替换时, DC 直接比较新的子代和它们的父代, 只有当子代有更高的适应度时才替换父代。这提供了 GA 的选择压力。在两个子代-父代对中, 高相似性的用来进行替换的比较。

### 概率拥挤 (Probabilistic Crowding Algorithm)

在 PCA 算法<sup>[88]</sup>中, 使用了概率的替换准则。一个子代  $x$  以概率  $ps = f(x)/(f(x) + f(y))$  替换最接近的父本  $y$ 。这种方法在简单问题中可以获得与适应度分布相同的稳态比例分布。这种方法会导致非极值点在群体中占有相当的比例, 算法难以达到普通意义上的收敛。

## 7.2.2 共享

另一种维护群体多样性的机制是共享, 最先由 Holland 提出<sup>[71]</sup>, 由 Goldberg 和 Richardson 发展<sup>[61]</sup>。共享机制的做法是减少那些在群体中有很多相似成员的个体的适应度, 对那些独立探索域中某些区域的个体进行奖励。这产生了群体多样性压力, 可以形成生境。

个体  $i$  的共享适应度由  $f_{sh,i} = f_i / m_i$  给出, 这里  $f_i$  是个体的原始适应度,  $m_i$  是生境计数, 给出了个体  $i$  与群体中其他个体的重叠程度。生境计数通过对所有个体的共享函数进行求和算出  $m_i = \sum_{j=1}^N sh(d_{i,j})$ 。距离函数  $d_{i,j}$  表示个体  $i$  和  $j$  之间的距离 (或相似性)。如果距离在一个给定的生境半径  $r_{sh}$  之内, 共享函数返回一个 0 和 1 之间的数, 并随相似性的增加而增长, 否则返回 0。实质上, 每一个个体都作为生境的中心来考虑, 它的共享适应度值受那些距离小于  $r_{sh}$  的个体的影响。经常使用如下共享函数:

$$sh(d_{i,j}) = \begin{cases} 1 - (\frac{d_{i,j}}{r_{sh}})^a, & \text{if } d_{i,j} \leq r_{sh} \\ 0, & \text{otherwise} \end{cases}$$

对于已知峰值点的个数  $q$ , 和整个域空间的超空间体积时, 存在确定  $r_{sh}$  的适当的经验公式<sup>[61]</sup>。



上面的共享方法有几个难点<sup>[22]</sup>。首先，难以精确估计峰值点的个数  $q$ 。其次，在高度、宽度和形状上各式各样的不规则的峰值点使  $\rho_{sh}$  的确定更加复杂。 $q$  和  $\rho_{sh}$  的值经常是保守估计的，分别是域中可能的最多的峰值点的数量和最小的生境半径。最后，计算共享适应度导致每代附加的  $O(n^2)$  阶的计算复杂性。对于很大的群体，或使用了复杂的域的相似性度量，附加的费用是难以接受的。

文献<sup>[22]</sup>中提出了一种动态共享 (Dynamical Sharing, DS) 方法来减少标准共享方法带来的附加计算量。这种方法使用一种“贪心法”来在每一代中直接找出  $q$  个生境中心。如果一个个体与某个生境中心的距离小于  $\rho_{sh}$ ，则使用这个生境中个体的数量来作为上述的生境计数  $m_i$ 。否则认为个体属于“非极值点”，仍然使用上面的公式来计算生境计数。这种方法可以减少计算共享适应度需要的计算量。

动态共享方法的假设与标准共享相同，因此仍然存在上述的问题。同时，使用贪心法找出生境中心与实际的生境中心可能相去甚远。

### 7.2.3 聚类

文献<sup>[101]</sup>将每一代的个体使用聚类分析方法划分为若干相互隔离的子空间。交叉操作只在同一个子空间内进行。它的目的是解决对称问题，同时也使用这种方法将生境引入到 PMBGA 算法中。在文献<sup>[22]</sup>中也指出，可以使用包括聚类分析在内的其他方法替代他所使用的贪心法来寻找生境中心，但没有实现。文献<sup>[101]</sup>使用的聚类分析方法是 K-means 方法，需要预先确定聚类的数目。另外，限制在同一聚类中进行交叉，将大大减少 GA 探索未知区域的能力，导致早熟收敛。这种方法可以维持群体的多样性，但难以产生新的群体多样性。

上面的各种方法都要使用一种相似性度量函数。它可以是基于基因型或表现型的。基因型的相似性度量经常是不依赖于问题知识的，如基因位串的 Hamming 距离等。表现型的相似性度量利用了问题知识来计算更有意义的距离量度。无论那种情况，相似性度量必须返回一个随相似性增加而减少的距离函数。如果有充足的领域知识，通常构造和使用基于表现型的相似性度量函数。

## 7.3 基于聚类分析的资源竞争(BCRC)算法

通过上面对生境遗传算法的分析，参照生物群体增长的模型，本文提出了一种新的基于聚类分析和资源竞争的遗传算法（GA Based on Clustering and Resource Competition, BCRC）。这种算法将聚类分析、共享技术和拥挤技术有机地结合起来，可以有效地搜索多模函数空间的多个极值点，同时可以通过调节拥挤因子控制收敛到的生境的数目，避免找到无效的极值点。这种算法无需事先确定生境的具体数目和生境半径的大小，带来的附加计算量比较小，能够适应各种问题的优化。

### 7.3.1 算法结构

算法的基本流程如下：

1. 随机产生初始群体。
2. 每隔若干代，使用改进的 K-means 算法对群体进行聚类分析，划分为  $q$  个聚类。
3. 计算每个聚类中个体的数目。计算个体的竞争适应度。
4. 按照个体的竞争适应度进行常规的选择、重组、变异操作。生成子代。
5. 确定子代属于哪个聚类。计算子代与子代与两父本之间的相似性度量，按照一定概率替换最接近的父代。
6. 循环执行操作 2-5，直到达到预定的结束条件。

### 7.3.2 改进的 K-means 算法

在聚类分析中，最简单的算法是 K-means 算法。尚有一些其他的基于统计的算法、使用神经网络的算法，也有使用进化算法进行聚类分析的研究<sup>[19,73]</sup>。

K-means 算法的基本流程如下：

随机产生  $q$  个中心。

将每一个点按照某种距离量度分配到最近的一个中心。

对于每一个中心，计算所有属于此中心的点的重心，作为新的中心坐标。

如果某个中心发生变化，转到第 2 步。

计算结束，返回  $q$  个中心位置。

K-means 算法的问题在于只能且必须产生预定的  $q$  个聚类中心，前面已经提到，在计算之前，难以确定生境（或聚类）的数目，往往只能取一个保守值。这样，如果第一步中中心的位置不理想，会导致最后计算出来的中心不能反映群体的生境组成。

为此修改上面的算法，在第 3 步之后增加一步：

3.a 如果有两个中心之间的距离小于最小聚类距离  $d$ ，则将这两个中心合并。

使用改进后的算法，产生出来的聚类数目  $r$  可能小于预定值  $q$ ，这反映了群体中实际的生境数目。

### 7.3.3 算法中其他事项

#### ✎ 竞争适应度

竞争适应度的含义与上面的共享适应度相类似，均反映了由于生境中个体的数目而导致的适应度的下降，公式如下：

$$f_{c_i} = (1 - c \cdot x_i) \cdot f_i$$

其中  $c$  为拥挤参数，( $0 < c < 1$ ，在实际使用时一般取接近于 1 的数，如 0.8)，反映了群体的拥挤程度，其数学意义见后面的分析。这个公式从形式上类似于自然界有限资源下种群增长的 Logistic 方程(2.1)。

#### ✎ 归类

由于计算聚类是一个需要计算量比较大的操作，因此不是在每一代中都进行。在本算法中允许不同类别的个体进行交叉，新生成的子代的类别如下决定：如果两个父代的类别相同，则子代的类别与父代相同；如果父代的类别不同，则子代的类别与最接近的父代的类别相同。

#### ✎ 计算复杂性

此算法的附加计算量包括以下几处：竞争适应度的计算、类别决定和聚类分析。前两项在每一代中都只附加一个常数项的计算量。K-means 算

法本身的复杂性为  $O(n^2)$  ( $n$  是群体数目), 但只在若干代才执行一次, 对整个计算量的增加并不显著。

## 7.4 数学分析

对于上述的 BCRC 算法, 可以生成和维护群体的多样性。下面讨论在稳态时模型的动力形态。

### 7.4.1 稳态时各生境所占的比例

在稳态时, 假设群体中形成了  $q$  个生境, 并且每个生境中的个体都收敛到其中的极值点。并且不同生境的个体交叉后生成的个体的适应度都远小于极值点的适应度, 因此交叉后产生的子代与父代竞争不能取胜。稳态时群体的分布完全由生境适应度值和选择算子决定。

下面, 假设在稳态时比例不为 0 的基因型共有  $q$  个。其中, 第  $i$  个基因型在群体中所占的比例为  $x_i$ , 当使用按适应度比例选择的选择算子时, 群体中下一代的的比例可以表示为方程组

$$F_k(\vec{x}) = \frac{(1 - c)x_k f_k}{\sum_{i=1}^q (1 - c)x_i f_i + (1 - c)x_q f_q}, k = 1..q-1 \quad (4.1)$$

由于  $q$  个变量之和为 1, 所以上面的方程组是  $q-1$  元的。

在稳态时,  $F_k(\vec{x}) = x_k^s$ 。(上标  $s$  表示稳态值) 可以推出

$$f_j / f_i (1 - c)x_j = 1 - c x_i. \quad (4.2)$$

将上式左右两端变量  $i$  从 1 到  $q$  求和得:

$$(1 - c)x_j \sum_{i=1}^q \frac{f_j}{f_i} = \sum_{i=1}^q (1 - c)x_i = q - c \quad (4.3)$$

令  $m = \sum_{i=1}^q \frac{1}{f_i}$  (4.4), 可以解出

$$x_j^s = \frac{m(f_j - c)q}{m(c - f_j)} + \frac{1}{c} \left(1 - \frac{q - c}{m - f_j}\right) \quad (4.5)$$

由  $j$  的一般性可知此式对所有  $j$  都成立。

这就是在稳态时不为 0 的基因型所占的比例，也就是各个有效的生境在整个群体中的比例。显而易见，这个比例随适应度值而单调递增。

### 7.4.2 生境稳定存在的条件

为使生境稳定存在，需满足  $0 \leq x_i^s \leq 1$ ，即要求  $m - f_i \leq (1 - c) \leq q \leq c \leq m - f_i$ 。

先指出这个不等式蕴含的结论，即由求解问题蕴含的（或我们先验估计出来的）所有生境中，并不是所有的生境均能在本算法的稳态结果中占有一席之地。只有那些适应度较高的生境才可以稳定地存在。这也是 BCRC 算法的重要特点之一。

下面说明生境能够稳定存在的标准。假设我们认为问题空间中存在  $r$  个生境。由于  $c$  接近于 1，不等式的左端一般都成立。不等式的右端可以变形为  $q \leq m - f_i \leq c \leq 1$  (4.6)。由于  $f_i$  取最小值时，上式的左端取最大值。

将适应度按降序排列，令  $M(n) = n \sum_{i=1}^n \frac{1}{f_i} \leq f_n$ 。  $M(n)$  表示  $r$  个生境中，假

设有  $n$  个生境能够稳定存在时，(4.6) 式左端可能的最大值。由于：

$$M(n) \leq M(n-1) \leq (f_{n-1} \leq f_n) \leq \sum_{i=1}^n \frac{1}{f_i} \leq 0, \quad M(1)=0.$$

也就是， $M(n)$  是单调递增的，因此完全可能存在某个  $q$ ，使得当  $n > q$  时， $M(n) > c$ 。这样对于较小的适应度，和整体适应度的调和平均值差距较大，不能满足(4.6)，因此不能在稳态时存在。对于很小的  $c$ ，这样的  $q$  是一定存在的。当然也有可能，对于所有生境，都满足上式，这样，这些生境都可以稳定存在。

下面举一个例子说明，假设最小的适应度为  $h$ ，其他的适应度按等差

数列递增，即  $f_i \geq h(1 + (i-1)d)$ ,  $d \geq 0$ ,  $d \neq 1$ ，则

$$m \geq \sum_{i=1}^r \frac{1}{f_i} \geq \sum_{i=1}^r \frac{1}{h(1 + (i-1)d)} \geq \sum_{i=1}^r \frac{1 + (i-1)d}{h} \geq \frac{1}{h}(r + d \frac{r^2}{2}) \quad (4.7)$$

代入(4.6)得  $q \geq m f_1 \geq r(r + d \frac{r^2}{2}) \frac{d}{2} \geq c$ ，即  $r \geq \sqrt{\frac{2c}{d}}$ 。也就是

说，在这种适应度分布下，只有最高的  $\sqrt{2c/d}$  个生境能够满足稳态条件，可以存在。

在上述例子中，代入(4.5)，求得各基因型的稳态比例为：

$$x_i^s \geq \frac{1}{c}(1 + \frac{q \geq c}{m f_i}) \geq \frac{1}{c}(1 + \frac{h}{h_i}) \geq \frac{1}{c}(1 + \frac{1}{1 + (i-1)d}) \geq \frac{1}{c}(i-1)d, i \geq 1.. \sqrt{2c/d}$$

$$\text{且满足：} x_{i+1}^s \geq x_i^s \geq \frac{d}{c} \geq \frac{1}{c}(\frac{f_{i+1}}{f_i})$$

上式说明，在这种假设条件下，稳定存在的各基因型在群体中所占的比例之差与它们的适应度之间的相对差距成正比。

一般来说， $x_{i+1}^s \geq x_i^s \geq \frac{q \geq c}{c \geq m}(\frac{1}{f_i} \geq \frac{1}{f_{i+1}})$ 。说明稳定存在的各基因型在群

体中所占的比例之差与它们的适应度的倒数的差距成正比。

### 7.4.3 BCRC 算法与生境规则

使用共享方式和概率拥挤方式的算法在稳态时可以获得如公式(2.2)确定的生境比例。BCRC 算法的稳态分布如公式(4.5)所示，与(2.2)并不一致。但公式(4.5)仍反映了一般的生境准则，即适应度越高的生境在整个群体中所占的比例越高。另外，(2.2)也只是对于广义的适应度而言。如果先对原始的适应度进行恰当的变换，再使用 BCRC 算法，就可以使最终的比例相对于原始的适应度符合此公式。下面说明此变换：

令原始的适应度为  $of$ ，其算术平均值为  $of_{avg}$  计算中使用的适应度变换为：

$f_i \neq 1/(b + of_i)$  , 其中  $b \neq q \cdot of_{avg} / c$  ,

则  $m \neq \sum_{i=1}^q \frac{1}{f_i} \neq \sum_{i=1}^q b + of_i \neq q(b + of_{avg})$  , 代入公式(4.5)有

$$x_i^s \neq \frac{1}{c} \cdot (1 + \frac{q \cdot c}{m \cdot f_i}) \neq \frac{1}{c} [1 + \frac{q \cdot c}{q(b + of_{avg})} \cdot (b + of_i)] \neq \frac{of_i}{q \cdot of_{avg}} \neq \frac{of_i}{\sum_{i=1}^q of_i} \quad (4.8)$$

这说明, 在这种适应度变换的意义下, BCRC 算法是符合生境规则的。

#### 7.4.4 BCRC 算法的稳定性

下面讨论在稳态时的动力学性态, 为此计算  $F_k(\dot{x})$  在  $x \neq x^s$  处的导数, 可以得到:

$$\left. \frac{\partial F_k(\dot{x})}{\partial x_k} \right|_{x \neq x^s} \neq 1 \neq \frac{(f_k \cdot (1 + c) \cdot f_n) \cdot x_k}{f_k \cdot (1 + c \cdot x_k)} \quad (4.9)$$

若使稳态点成为吸引不动点, 要求  $\left. \frac{\partial F_k(\dot{x})}{\partial x_k} \right|_{x \neq x^s} \neq 1$  , 即  $c \neq 1 + \frac{f_n}{f_k}$  。

按照前述的稳态点的存在条件, 各个适应度值之间的差距不大, 因此上述条件都能满足。

应该注意到, 这里得到的条件只是必要条件, 而非充分条件。充分条件的获得尚待进一步的研究

### 7.5 实验测试

对于广为使用的测试函数

$$F(x) \neq \exp(-2 \ln 2 \cdot (\frac{x + 0.1}{0.8})^2) \cdot \sin^6(5\pi x), x \in [0, 1) \quad [22, 84, 88]$$

曲线所示, 此函数具有 5 个极值点, 列表如下:

N	X	F(x)	1/F(x)	$m(n) = \sum_{i=1}^n 1/f(x_i)$	$F(x_n) = m(n)$	M(n)
1	0.1	1	1	1	1	0
2	0.3	0.917	1.091	2.091	1.917	0.083
3	0.5	0.707	1.414	3.505	2.478	0.521
4	0.7	0.459	2.181	5.686	2.607	1.393
5	0.9	0.25	4	9.686	2.421	2.579

表 7.1 测试函数的公式值

从上表中可以看出，若取  $c=0.8$ ，只有当  $n \leq 3$  时， $M(n)$  才能满足(4.6)，也就是说，使用 BCRC 算法，只有前三个极值点才能在稳态时存在。它们各自所占的比例为  $x^s = \{0.466, 0.394, 0.140\}$ 。若取  $c=0.5$ ，则第三个极值点也不能稳定存在。

在测试时使用如下参数：

群体容量：200

基因长度：10

预定义聚类数目：8

交叉因子：1.0

变异因子：0.01

最小聚类距离：0.025

迭代次数：40

距离函数：表现型距离函数， $d(x_1, x_2) = |x_1 - x_2|$

每次计算随机均匀初始化，执行 50 次进行统计。



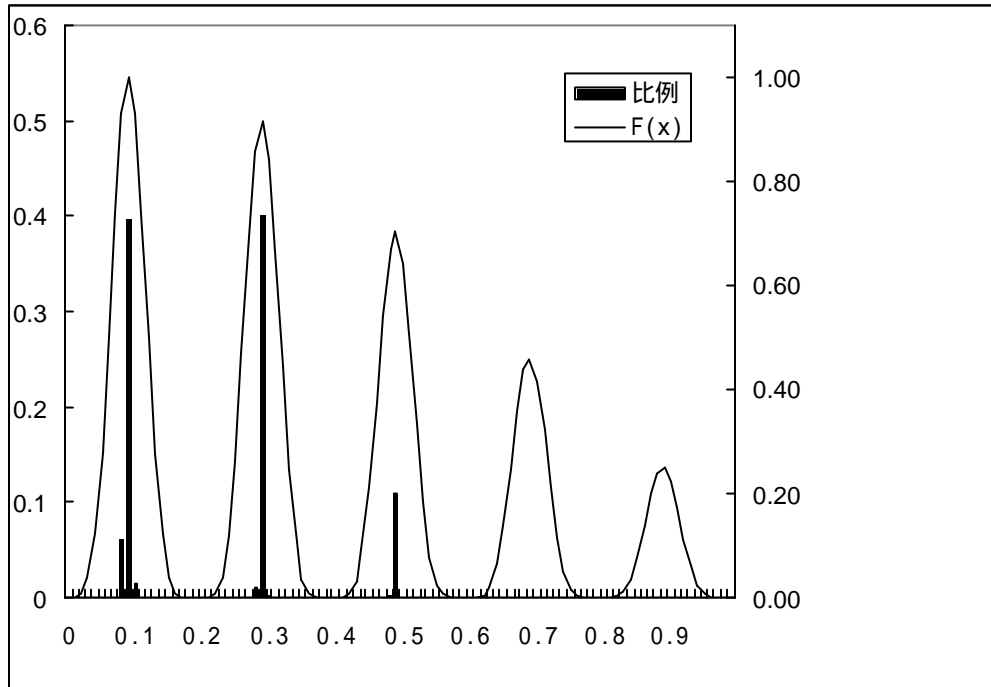


图 7.1 测试函数的极值点与种群分布

当拥挤因子  $c=0.8$  时，在 50 次执行中，有 42 次收敛到三个生境共存的状态，占 84%，其余 8 次收敛到两个生境共存的状态。这是因为第三个生境所占的稳态比例比较小，因此基因漂移的影响较大，容易消失。在下表中“整体”栏指统计所有 50 次运行的数据；“3 生境”栏只统计存在 3 个生境的 42 次运行。如图 7.1 中的柱状图所示。

生境	理论比例	整体		3 生境	
		比例	标准差	比例	标准差
1	0.466	0.473	0.043	0.461	0.038
2	0.394	0.413	0.041	0.402	0.036
3	0.140	0.113	0.050	0.137	0.031

表 7.2 生境的比例 1

若取  $c=0.5$ ，有 10 次执行  $x_3$  仍然存在，但所占比例很小，数据如下：

生境	理论比例	整体	
		比例	标准差
1	0.565	0.552	0.052
2	0.435	0.440	0.050
3	0	0.008	0.012

表 7.3 生境的比例 2

## 7.6 小结

前面已经提到，共享方法的问题在于难以确定生境半径。如果生境半径取得过大，则有可能将多个极值点包含在一个生境中，从而找不到真正的极值点；小的生境半径导致局部极值点的数目急剧增加，使得最优解（和准最优解）淹没在局部极值点中，即使是对于多模函数，如此多的局部极值解也不是我们所期望的。更加复杂的是，实际问题中，各个生境的生境半径大小并不相同，有可能区别很大，使得生境半径的选取无所适从。

BCRC 算法克服了以上的缺点。首先，生境的确认是通过聚类分析得到的，而不是由生境半径决定的。这就不必预先定义生境半径，使得各种大小形状不同的生境可以被同时发现。（虽然在改进的聚类算法进行聚类合并时使用了参数：最小聚类距离，但这里对参数的要求很宽松）。其次，由前面分析的稳态比例的特点，只有那些适应度足够接近于最优适应度的生境才能稳定存在，这就限制了算法找到的局部极值点的数目。算法中的参数  $c$  就是一个决定能够入选的解与最优解之间差距的标准。在进行聚类分析时就可以确定一个相对大的可能生境数目，由于改进的  $k$ -means 算法可以对聚类进行合并，以及算法本身特点，最后得到一个更加合理的生境的数量、位置和相互比例。

随后的数学分析说明了算法的运行机理，实验测试数据验证了此方法的有效性。在以后的工作中，将对算法进行进一步的分析，以证明算法的收敛性和稳定性，同时将此算法应用到更加复杂的问题中。

## 第八章 结论和进一步工作

目前在很多领域，大规模的优化问题以及其他复杂问题的广泛出现，对传统的优化计算方法和人工智能技术提出了强烈的挑战。人们基于自然界生物进化的思想而提出了一类新的计算智能方法：进化算法。这一类算法，使用简单的编码方式，对由一组个体组成的群体施以重组、变异等遗传操作，并按照优胜劣汰的选择方式确定群体的搜索方向，从而获得可行的优化解。进化算法具有不依赖于领域知识，不受搜索空间的限制以及内在并行性等优势，这些特点使其在各个领域得到了越来越多的应用。在进化算法之中，遗传算法具有典型代表性，使用的也最为广泛。

虽然进化算法已经得到了很多成功的应用，但由于它内在的并行性、随机性和进化算子的非线性作用，使得进化算法的理论分析具有很大的困难。目前，进化算法的理论分析还很不完善，还没有对算法的收敛性、运行机理以及适用性等根本问题得到妥善的解答。对于近年来理论分析的热点问题：遗传算法，虽然从提出之日就有理论上的说明，但由于其内在的复杂性，尤其是交叉算子的非线性，仍存在很多问题。

为分析遗传算法，人们提出了很多数学模型。早期的理论分析方法是模式理论。目前，人们逐渐意识到模式理论不能分析算法的极限行为。使用 Markov 链的方法，人们对传统算法和精英算法，获得了极限收敛结论，但和实际应用距离尚远。使用动力系统的研究方法，虽然针对的是无限群体，但已有结论表明，无限群体模型与有限群体模型之间的关联性。由于这种方法的数学工具比较成熟，复杂度也较 Markov 链方法小，因此最近很多人对此进行了深入研究。对于在遗传算法中起到重要作用的交叉算子，由于其内在的非线性，更加难以对其建模和分析。

在本文中，在忽略变异算子时，通过对标准遗传算法进行动力学建模，获得了包括交叉算子和选择算子的算法系统模型。为分析此模型在平凡不动点（也就是某个基因型充斥整个群体的状态）处的吸引力，需要计算模型在此处的 Jacobin 矩阵的特征值。由于选择算子和交叉算子的可分解性，可以分别计算这两个算子对应的矩阵的特征值。选择算子较为简单，而交

叉算子就比较复杂。因此引入了一种依照 Hamming 距离对基因型重新排序的置换,在此置换下,交叉算子的 Jacobin 矩阵简化为上三角矩阵。这样,计算出整个模型的所有特征值。

根据特征值是否全部小于 1,可以将平凡不动点区分为可收敛的局部极值点和不可收敛到的两种类型。局部极值点这个概念,以往文献中经常提到的,却又没有给出明确定义。通过上面的分析,本文中给出了精确的数学定义。这种分析方法对于常见的各种算子的改进都是适用的。因此,本文又分析了基于排名的和基于锦标赛的选择算子,以及其他类型的交叉算子,都获得了类似的结论。在分析选择算子的同时,又发现锦标赛选择与线性排名的极限行为的一致性。使用这些结论,可以对遗传算法的改进提供理论上的指导作用。

要了解遗传算法的运行机理,不仅需要知道它的极限行为,而且需要了解算法在未收敛时的运行形态。简单地说,就是非平凡不动点的存在性和收敛性。本文针对一个具有代表性的简化问题,详细分析了遗传算法的全局动力学形态。由于在建模时加入了算法的控制参数,可以分析当这些参数变化时,算法动力学形态的稳定性。从这些分析中可以看出,对于这个简化问题,模型的全局动力学形态还是比较简单的,算法能够收敛到哪一个极值点完全由初始条件决定,并且对初值不敏感。当求解问题存在两个局部极值点时,还可以计算出每个点的收敛区域。当算法的变异概率很小时,模型与没有变异算子相类似。当变异算子足够大时,模型的动力学形态随着变异概率的增加发生了突变。上述结论有助于加深对于 GA 的动力学形态的理解。对于参数的选取具有一定的参考作用。

在评价单个选择算子的性能,也就是选择压力时,经常使用取代时间的概念。目前的研究中,大部分的定量分析结果都是建立在某种特定的适应度函数以及均匀的初始分布这种假设基础上,没有取得一般性的结论。在本文中,对取代时间这个概念给出严格定义。对于基于比例的选择算子,详细地分析了算子的取代时间,证明了取代时间的阶不依赖于适应度函数的选取和初始群体的分布,都是  $O(\ln N)$  阶。同时,本文提出了取代时间系数的概念,这样,各种不同类型的适应度函数及其变换将只影响取代时间系数。在此基础上,本文讨论了常见的适应度函数变换方式对选择压力的

影响。

简单的遗传算法难以处理多模函数优化问题。为此已经提出了各种改进方法，包括基于共享和基于拥挤等机制。这些机制提出的目的是为了在算法的运行过程中形成生境，以维持局部极值点的多样性。本文提出了一种基于聚类分析的资源竞争算法，可以不必预先确定生境的分布和生境半径，并对小适应度的局部极值点具有自适应性的排除能力。这种算法的计算量也远小于基于共享的方式。从初步的数学分析和测试来看，算法完全可以胜任多模函数的优化问题。

本文使用动力系统的方法，对遗传算法的收敛性和全局动力学形态进行了一定的分析，深入的探讨了选择算子的取代时间，并提出了一种新的生境遗传算法。在以后的工作中，将深入分析遗传算法的全局动力形态，对其中可能存在的复杂的动力学行为，如非全局吸引子、极限环和混沌行为等进行探讨，从而得到影响算法的收敛目标和收敛速度的各种条件，以及遗传算法适用的函数特性等结果。同时，将深入分析无限群体模型和有限群体模型之间的关系，从而将这些结论推广到实际使用的有限群体算法上。对于生境遗传算法，将试图对其进行进一步的理论分析，拓展其使用范围和使用效果。

---

## 致 谢

衷心感谢导师刘大有教授和周春光教授在博士论文期间对我的精心指导。两位导师严谨的治学作风和积极的工作态度使我受益匪浅。从他们那里我学到了很多科研方法，尤其是在治学态度上的严格要求，对我完成博士论文帮助很大。这些在我以后的工作中都将起到巨大的作用。

感谢我的家人在工作和生活上的支持。

对论文期间给予我很多帮助的老师 and 同学们表示衷心感谢。

---

## 在读期间完成的主要研究工作

1. 郭东伟, 周春光, 刘大有。遗传算法取代时间的分析。计算机研究与发展 Vol. 38(10), 2001. Pages 1211-1216
2. 郭东伟, 刘大有, 周春光, 张仲明: 遗传算法收敛性的动力学分析及其应用。计算机研究与发展 Vol. 39(2), 2002. Pages 225-230
4. Guo Dongwei, Dynamic Analysis on GA's Constringency at Fixed Point. Proceedings of the sixth International Conference for Young Computer Scientist. ICYCS2001. Volume II, 2001. Pages 745-748.
3. 吕慧英, 郭东伟, 周春光, 梁艳春。氨基酸序列中同源序列的识别。小型微型计算机系统。Vol.21, No.6, pp.646-649, Jue.2000.
5. 郭东伟, 周春光, 刘大有。基于聚类分析和资源竞争模型的生境遗传算法。计算机研究与发展 (已投稿)。
6. 郭东伟, 周春光, 吕慧英, 张仲明。参数选取对遗传算法动力学形态的影响。小型微型计算机 (已录用)。
7. 郭东伟, 周春光, 张仲明, 胡明 遗传算法的全局动力学形态分析。计算机研究与发展 (已投稿)。
8. 国家自然科学基金项目: 基于计算智能的振动系统仿真建模的研究。批准号: 19872027。(主要参加人)
9. 国家自然科学基金项目: 基于计算智能方法计算机味觉的研究。批准号: 60175024。(主要参加人)
10. 吉林大学创新基金项目: 并行遗传算法在分子遗传学中的应用。批准号: 2000B02。(负责人)

## 参考文献

- [1] Ackley, D.H. (1987). A Connectionist Machine for Genetic Hill-climbing. Boston, MA. Kluwer Academic Publisher.
- [2] Agapie, A.(1999). Genetic Algorithms as Random Systems with Complete Connections, In [12], Volume I, Pages 770.
- [3] Alliot, J.M., Lutton, E., Randal, E., Schoenauer, M. and Snyers, D. (Eds.), *Artificial Evolution, Volume 1063 of LNCS*. Springer-Verlag, 1996.
- [4] Altenberg, L.(1995). The Schema Theorem And Price's Theorem. In [133], Pages 23-49.
- [5] Ankenbrandt, C.A.(1991). An Extension the Theory of Convergence and a Proof of Time Complexity of Genetic Algorithms. In [107], Pages 53-68.
- [6] Antonisse, J.(1989). A New Interpretation of Schema Notation That Overturns the Binary Encoding Constraint. In [117], Pages 86-91.
- [7] Arona, S., Rabani, Y., and Vazirani, U. (1994). Simulating Quadratic Dynamical Systems is PSPACE-Complete. In *Proceeding of the 26th Annual ACM Symposium on Theory of Computing (STOC'94)*, Pages 459-467. ACM Press, New York
- [8] Baker, J.E.(1985), Adaptive Selection Methods for Genetic Algorithms. In [64], Pages 110-111.
- [9] Baker, J.E.(1987), Reducing Bias and Inefficiency in the Selection Algorithm. In[47], Pages 14-21.
- [10] Baluja, S.(1995). An Empirical Comparison of Seven Iterative and Evolutionary Function Optimization Heuristics. Technical Report CMU-CS-95-193, Carnegie Mellon University.
- [11] Banzhaf, W. and Reeves, C. (Eds.), *Foundations of Genetic Algorithms 5*. Amsterdam, 1998, Morgan Kaufmann, 1999.
- [12] Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., and Simth, R.E., (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference 1999(GECCO-99)* .Morgan Kaufmann. San Francisco, CA, 1999.
- [13] Beasley, D., Bull, D.R., and Martin, R.R. (1993). A Sequential Niche Technique for Multimodal Function Optimization, *Evolutionary Computation*, Vol.1, No.2, Pages:101-125.
- [14] Belew, R.K. and Booker, L.B., (Eds.), *Proceedings of the 4th International Conference on Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, 1991.



- 
- [15] Belew, R.K. and Vose, M.D., (Eds.), *Foundations of Genetic Algorithms 4* Morgan Kaufmann, San Francisco, CA, 1997.
  - [16] Berny, A. (1999). Statistical Machine Learning and Combinatorial Optimization. Springer. In *Proceedings of the Second Evonet Summer School on the Theoretical Aspects of Evolutionary Computing*.
  - [17] Bethke, A.D. (1980). Genetic Algorithms as Function Optimizers. PhD Thesis, University of Michigan.
  - [18] Beyer, H.G. (1994). Toward a Theory of Evolution Strategies: the  $(\mu, ?)$  Theory. *Evolutionary Computation*, 2(4):381-407.
  - [19] Bezdek, J. C., Boggavarapu, S., Hall, L. O., & Bensaid, A.(1994) Genetic Algorithm Guided Clustering. In [90], Volume 2, Pages 34--39.
  - [20] Blicke, T. and Thiele L.(1995). A Mathematical Analysis of Tournament Selection. In [45], Pages 9-16..
  - [21] Booker, L.B. (1993). Recombination Distributions for Genetic Algorithms. In [132], Pages 29-44.
  - [22] Brad, L.M., Michael, J.S. (1996). Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. *IEEE International Conference on Evolutionary Computation*, Pages 786-791.
  - [23] Bäck, T.(1992). The Interaction of Mutation Rate, Selection and Self-adaptation in Genetic Algorithm. In [92], Pages 85-94.
  - [24] Bäck, T.(1994). Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In [90], Pages 57-62
  - [25] Bäck, T.(1995). Generalized Convergence Models for Tournament- and  $(\mu, ?)$ -Selection. In [45], Pages 2-8
  - [26] Bäck, T.(1995b). *Evolutionary Algorithms in Theory and Practice*. New-York: Oxford University Press.
  - [27] Bäck, T., (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann, San Francisco, CA, 1997
  - [28] Cerf, R.(1996). An Asymptotic Theory of Genetic Algorithms. In [3]. Pages 11-18.
  - [29] Chakraborty, U.K. and Dastidar, D.G..(1993). Using Reliability Analysis to

Estimate the Number of Generations to Converge in Genetic Algorithms. Information Processing Letters, Vol(46), Pages 199-209.

[30] Chakraborty, U.K., Deb, K., and Chakraborty, M. (1996). Analysis of Selection Algorithms: A Markov Chain Approach. Evolutionary Computation, 4(2), Pages 133-167.

[31] 陈培德 《随机数学引论》 科学出版社，北京，2001

[32] Christopher Ronnewinkel, Claus O. Wilke, Thomas Martinetz.(2000) Molecular Evolution in Time-Dependent Environments. In [33], Pages 417-421

[33] Dario Floreano Jean-Daniel Nicoud, Francesco Mondada (Eds.): *Advances in Artificial Life, 5th European Conference, ECAL'99*. 1999.

[34] Das, R. and Whitley, D.(1991). The Only Challenging Problems Are Deceptive: Global Search by Solving Order-1 Hyperplanes. In [14], Pages 166-173.

[35] David, H. and Hornik, K.(1996) The Dynamic of Genetic Algorithms in Interactive Environments. Journal of Network and Computer Applications, 19, Pages 5-19.

[36] Davis, L. (1991a). Towards an Exploration of the Simulated Annealing Exploration Theory onto the Simple Genetic Algorithm. PhD Thesis, University of Florida, Gainesville, FL.

[37] Davis, T.E. and Principe, J.C.(1991b). A Simulated Annealing Like Convergence Theory for the Simple Genetic Algorithms. In [14], Pages 174-181.

[38] De Jong, K.A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral dissertation, University of Michigan.

[39] De Jong, K.A.(1992) Are Genetic Algorithms Function Optimizers?. In [92], Pages 3-13.

[40] Douglas, I.B., and Jamie, R.S. (1999). Evolution Determined by Trajectory of Expected Populations: Sufficient Conditions with Application to Crossover. Evolutionary Computation, 7(2). Pages 151-171.

[41] Eiben A.E., Aarts E.H. and Van H.K.M.(1991). Global Convergence of Genetic Algorithms: An Infinite Markov Chain Analysis. In [118], Pages 4-12.

[42] Eiben, A.E. and Rudolph G (1999). Theory of Evolutionary Algorithms: A

Birds Eye View. Theoretical Computer Science 229(1):3-9, 1999.

[43] Electrical, I. and Engineers, E. (Eds.), *Proceedings of the Fifth IEEE International Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 1998.

[44] Erik van Nimwegen, James P. Crutchfield and Melanie Mitchell. (1997). Finite Populations Induce Metastability in Evolutionary Search. Physics Letters A, 229(2). Pages 144-150.

[45] Eschelman L.J. (Ed.). *Proceedings of the 6th International Conference on Genetic Algorithms*. Morgan Kaufman. San Francisco, CA, 1995.

[46] Fogel, D.B. (1992) An Analysis of Evolutionary Programming. In [47], Pages 43-51.

[47] Fogel, D.B. and Atmar, W., (Eds.), *Proceedings of the 1st Annual Conference on Evolutionary Programming*. Evolutionary Programming Society, La Jolla, CA, 1992.

[48] Fogel, L.J.(1962) Autonomous Automata. Industrial Research, 4. Pages 14-19.

[49] Fogel, L.J., Owens, A.J. and Walsh,(1966) M.J. Artificial Intelligence through Simulated Evolution. John Wiley, New York.

[50] Forest, S., (Ed.), *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, 1993.

[51] Forrest, S. and Mitchell, M.(1993). What Makes a Problem Hard for a Genetic Algorithms? Machine Learning, 13(2/3), Pages 285-319.

[52] Francois, O. (1998). An Evolutionary Strategy for Global Minimization and its Markov Chain Analysis. IEEE Transactions on Evolutionary Computation. Vol. 2, 3, Pages: 77-91.

[53] Garnier, J., Kallel, L. and Schoenauer, M. (1999). Rigorous Hitting Times for Binary Mutations. Evolutionary Computation, 7(2), Pages 173-203.

[54] Goldberg, D.E.(1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley.

[55] Goldberg, D.E.(1990). Real-Coded Genetic Algorithms, Virtual Alphabets,

and Blocking, University of Illinois at Urbana-Champaign, Technical Report No.90001.

[56] Goldberg, D.E. (1998). The Race, the Hurdle and the Sweet Spot: Lessons from Genetic Algorithms for the Automation of Design Innovation and Creativity. Technical Report 98007, Illinois University.

[57] Goldberg, D.E. (2000a). A Meditation on Computational Intelligence and Its Future. A Foreword to the Proceeding of the 2000 International Symposium on Computational Intelligence in Slovakia. Technical Report 2000019, Illinois University.

[58] Goldberg, D.E. and Bridges C.L.(1990). An Analysis of a Reordering Operator on a GA-Hard Problem. Biological Cybernetics. 62, Pages397-405.

[59] Goldberg, D.E. and Deb, K.(1991). A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In [107], Pages 69-93.

[60] Goldberg, D.E., Deb, K. and Thierens, D. (1993). Toward a Better Understanding of Mixing in Genetic Algorithms. Journal of the Society for Instrumentation and Control Engineers(SICE), 32(1). Pages 10-16.

[61] Goldberg, D. E. and Richardson, J.J. (1987). Genetic Algorithms with Sharing for Multimodal Function Optimization. In [65], Pages 41-49.

[62] Goldberg, D.E. and Thomas, A.L.(1986). Genetic Algorithms: A Bibliography 1962-1986. TCGA Report No. 86001. Tuscaloosa: University of Alabama.

[63] Goldberg, D.E. etc. Genetic Algorithms: A Bibliography. IlliGAL Report No. 2000037. Illinois Genetic Algorithms Laboratory, University of Illinois. 2000b.

[64] Grefenstette, J.J., (Ed.), *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[65] Grefenstette, J.J., (Ed.), *Proceedings of the 2nd International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[66] Grefenstette, J.J.(1991) Conditions for Implicit Parallelism. In [107], Pages 252-261.

[67] Hao, J.K., Lutton, E., Ranald, E., Schoenauer, M. and Snyers, D. (Eds.),

*Artificial Evolution'97*, Nîmes, France, LNCS. Springer Verlag, 1997.

[68] Hart, W.E. and Belew, R.K.(1991). Optimizing an Arbitrary Function is Hard for Genetic Algorithm. In [14], Pages 190-195.

[69] Hartl, R.F. and Belew, R.K.(1990). A Global Convergence Proof for a Class of Genetic Algorithms. Technical report. Technische Universitat Wien.

[70] Holland, J.H.(1962) Outline for a Logical Theory of Adaptive Systems. Journal of the association of computing machinery, 3.

[71] Holland, J.H.(1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI.

[72] Horn, J. (1993). Finite Markov Chain Analysis of Genetic Algorithms with Niching. In [50], Pages 110-117.

[73] Horn, J., Goldberg, D. E., and Deb, K. Implicit Niching in a Learning Classifier System: Nature's Way. *Evolutionary Computation*, 2(1), 37--66. 1994

[74] Jonathan, E.R. (1999). Finding Attractors for Periodic Fitness Functions. In [12].

[75] Jonathan, E.R. (2000a). Cyclic Attractors and Quasi-species Adaptability. In [78]

[76] Jonathan, E. R. (2000b). The Dynamical Systems Model of the Simple Genetic Algorithm. In [78].

[77] Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992

[78] Leila Kallel, Bart Naudts and Alex Rogers. (Eds). *Theoretical Aspects of Evolutionary Computing*. Springer Verlag, 2000.

[79] Liang Yanchun, Wang Zhaishen and Zhou Chunguang. A Study on Convergence of Canonical Genetic Algorithms with Natural Selection. *Chinese Journal of Electronics*. 1997, 6(3): 71-75.

[80] 梁艳春、周春光、王在申. 基于扩展串的等效遗传算法的收敛性. *计算机学报*. 1997, 20(8): 686-694.

[81] 梁艳春、王在申、周春光. 选择和变异操作下遗传算法的收敛性研究. *计算机研究与发展*, 1998, 35(7): 655-660.

- [82] Liepins, G.E. and Vose, M.D.(1991). Deceptiveness and Genetic Algorithm Dynamics. In [107], Pages 36-50.
- [83] Lobo, F. G., Goldberg, D.E. and Pelikan, M. Time Complexity of Genetic Algorithms on Exponentially Scaled Problems. IlliGAL Report 2000015, 2000
- [84] Mahfoud, S. W. (1992). Crowding and Preselection Revisited. In [92]. Pages 27-36.
- [85] Mahfoud, S.W. (1993). Finite Markov Chain Model for an Alternative Selection Strategy for the Genetic Algorithms. *Complex Systems*, 7(2), Pages 155-170.
- [86] Mahfoud, S.W. (1995). Niching Methods For Genetic Algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA.
- [87] Mathias, K.E. and Whitley, L.D(1994). Transforming the Search Space with Gray Coding. In [90], Pages 519-524.
- [88] Mengshoel, O.J. and Goldberg, D.E. (1999). Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement. In [12], Pages 409-416.
- [89] Michalewicz, Z.(1992). *Genetic Algorithms + Data Structure = Evolution Programs*. Springer-Verlag, Berlin,
- [90] Michalewicz, Z., Schaffer, J.D., Schwefel, H.P., Fogel, D.B. and Kitano, H. (Eds.), *Proceedings of the First IEEE International Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 1994.
- [91] Munetomo, M. and Goldberg, D.E. (1999). Identifying Linkage Groups by Nonlinearity/ Non-monotonicity Detection. In [12], Pages 433-440.
- [92] Männer, R. and Manderick, B. (Eds.), *Proceedings of the 2nd International Conference on Parallel Problems Solving form Nature*. Elsevier Science. North-Holland, Amsterdam, 1992.
- [93] Mühlenbein, H.(1991). Evolution in Space and Time. In [107]. Pages 316-338.
- [94] Mühlenbein, H.(1992). How Genetic Algorithms Really Work: I. Mutation and Hill-Climbing. In [92], Pages 15-25.
- [95] Mühlenbein, H.(1997). The Equation for the Response to Selection and Its use for Prediction. *Evolutionary Computation*, 5(3), Pages 303-346.

- [96] Mühlenbein, H. and Schlierkamp-Vose, D.(1993). Predictive Models for the Breeder Genetic Algorithm. *Evolutionary Computation*, 1(1), Pages 25-49.
- [97] Nix, A. and Vose M.D.(1992). Modeling Genetic Algorithms with Markov Chains. *Annals of Math. And AI*, 5, Pages 79-88.
- [98] 潘正君、康立山、陈毓屏著《演化计算》清华大学出版社，1998
- [99] Peck, C.C. and Dhawan, A.P.(1995) Genetic Algorithms as Global Random Search Methods: An Alternative Perspective. *Evolutionary Computation*, 3(1), Pages 39-80.
- [100] Pelikan, M., Goldberg, D. E., and Cantú-Paz, E. (2000a). Hierarchical Problem Solving by the Bayesian Optimization Algorithm. IlliGAL Report No. 2000002. Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory. 2000
- [101] Pelikan, M. and Goldberg, D. E. (2000b) Genetic Algorithms, Clustering, and the Braking of Symmetry. IlliGAL Report 2000013.
- [102] Prügel-Bennett, A. and Shapiro, J.L. (1994). An Analysis of Genetic Algorithms Using Statistical Mechanics. *Physical Review Letter.*, 72(9), Pages 1305-1309.
- [103] Prügel-Bennett, A. (1997a). Modeling Evolving Populations. *Theoretical Biology*.185, Pages 81-95, 1997
- [104] Prügel-Bennett, A. and Shapiro, J.L.(1997b). The Dynamics of a Genetic Algorithms for Simple Random Ising Systems. *Physical D*, 104, Pages 75-111.
- [105] Radcliffe, N.J.(1992). Nonlinear Genetic Representations. In [92], Pages 259-268.
- [106] Rattay, M. (1996). Modeling the Dynamics of Genetic Algorithms Using Statistical Mechanics. PhD thesis, Manchester University, UK.
- [107] Rawlins, G..J.E., (Ed.), *Foundations of Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, 1991.
- [108] Rechenberg, I.(1973a). Bionik, Evolution und optimierung. *Naturwissenschaftliche Rundschau*, 26(11), Pages 465-472.
- [109] Rechenberg, I. (1973b). Evolutionsstrategie optimierung technischer

systeme nach prinzipien der biologischem evolution. Stuttgart- Bad Cannstatt: Friedrich Frommann Verlag.

[110] Rogers, A. and Prügel-Bennett, A. (1998). Genetic Drift in Genetic Algorithm Selection Schemes. IEEE Transactions on Evolutionary Computation, Vol3(4), Pages:298-303.

[111] Rogers, A. and Prügel-Bennett, A. (1999). Modeling the Dynamics of a Steady State Genetic Algorithm. In [11], Pages 57-68.

[112] Rudnic, M.(1992) Genetic Algorithms and Fitness Variance with an Application to the Automated Design of Artificial Neural Networks. [Doctoral dissertation]. Oregon Graduate Institute of Science and Technology. 1992.

[113] Rudolph, G.(1994a). Convergence Analysis of Canonical Genetic Algorithm. IEEE Transactions on Neural Networks. 5(1); Pages 96-101.

[114] Rudolph, G. (1994b). Convergence of Non-elitist Strategies. In [90], Pages 63-66.

[115] Rudolph, G. (1997). Asymptotical Convergence Rates of Simple Evolutionary Algorithms Under Factorizing Mutation Distributions. In [67], Pages 223-233.

[116] Rudolph, G. (1997b). Convergence Properties of Evolutionary Algorithms. Kovac, Hamburg, 1997.

[117] Schaffer, J.D., (Ed.), *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, 1989.

[118] Schwefel H.P. and Manner R., (Eds.). *Proceddings of the 1st International Conference on Parallel Problem Solving from Nature*. Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1991.

[119] Shapiro, J.L. and Prügel-Bennett, A.(1997). Genetic Algorithm Dynamics in a Two Well Potential. In [15], Pages 101-115.

[120] Stephens, C. and Waelbroeck, H. (1997). Effective Degree of Freedom in Genetic Algorithms and the Block Hypothesis. In [27], Pages 34-40.

[121] Stephens, C. and Waelbroeck, H. (1999). Schemata Evolution and Building Blocks. Evolutionary Computation, 7(2): Pages 109-124.



- [122] Suzuji, J. (1997) . A Further Result on the Markov Chain Model of Genetic Algorithms and its Application to a Simulated Annealing-like Strategy. In [15], Pages 53-72.
- [123] Thierens, D. and Goldberg, D.E. (1993). Mixing in Genetic Algorithms. In [50], Pages 38-55.
- [124] Thierens, D., Goldberg, D.E. and Pereira, A.G. (1998). Domino Convergence, Drift, and the Temporal-salience Structure of Problems. In [43], Pages:535-540.
- [125] Troune, A. (1996). Cycle Decomposition and Simulated Annealing. SIAM Journal Control and Optimisation, 34(3), Pages 966-986.
- [126] Vose, M.D.(1991). Generalizing The Notion of Schemata in Genetic Algorithms. Artificial Intelligence, 50(3), Pages 385-396.
- [127] Vose, M.D.(1993). Modeling Simple Genetic Algorithms. In [132], Pages 63-73.
- [128] Vose, M.D. (1999). The Simple Genetic Algorithm: Foundations and Theory. MIT Press.
- [129] Vose, M.D. and Liepins, G.E. (1991). Punctuated Equilibria in Genetic Search. Complex Systems, 5(1), Pages 31-44. 1991
- [130] Whitley, D. (1989). The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trial is Best. In [117], Pages 116-121
- [131] Whitley, D.(1993) An Executable Model of a Simple Genetic Algorithm. In [132], Pages 45-62.
- [132] Whitley, L.D. (Ed.), *Foundations of Genetic Algorithms 2*. Morgan Kaufman., San Mateo, CA, 1993.
- [133] Whitley, L.D. and Vose, M.D. (Eds.), *Foundations of Genetic Algorithms 3*. Morgan Kaufmann, San Mateo, CA, 1995.
- [134] Wilson, S.W. (1991). GA-easy Does Not Imply Steepest-Ascent Optimizable. In [14], Pages 85-89.
- [135] Wright, A.H. and Bidwell, G.L. (1997) A Search for Counter Examples to Two Conjectures on the Simple Genetic Algorithm. In [15], Pages 73-84.

- [136] Wright A.H., Alexandru Agapie.(1999). Cyclic and Chaotic Behavior in Genetic Algorithms. In [12], Pages 718-724.
- [137] 张筑生 《微分动力系统原理》 科学出版社，北京 1997
- [138] Zhigljavsky, A.A.(1992) . Theory of Global Random Search. Mathematics and its Applications. Kluwer Academic Press. Kluwer, Dordrecht, 1992.

## 摘 要

目前在很多领域,大规模的优化问题以及其他复杂的计算问题,对传统的优化计算方法和人工智能技术提出了强烈的挑战。人们基于自然界生物进化过程中重组、变异以及优胜劣汰的思想,提出了一类新的计算智能方法:进化算法。进化算法具有不依赖于领域知识,不受搜索空间的限制以及内在并行性等优势。因此进化算法在各个领域得到了越来越多的应用。在进化算法之中,遗传算法具有典型代表性,使用的也最广泛。

虽然进化算法已经得到了很多成功应用,但算法的理论分析还很不完善。为分析遗传算法的收敛性、运行机理以及适用性等根本问题,人们提出了各种理论。但由于它内在的并行性、随机性、特别是交叉算子的非线性,迄今为止仍没有一个一致的、完备的、实用的理论分析结果。

为分析遗传算法,人们提出了很多数学模型。早期提出的模式定理在分析算法的极限性能时失去了作用。针对传统算法和精英算法,人们使用 Markov 链方法获得了极限收敛结论,但和实际应用距离尚远。其中虽然讨论了交叉算子的作用,但只是将其作为一个模型中起状态转换作用的算子,最终结论的取得需要依赖于变异算子的不可约性,未体现出交叉算子在引导收敛目标中的作用。近年来,很多研究成果基于动力系统的方法,虽然这种方法针对的是无限群体,但已有研究表明,无限群体模型与有限群体模型之间具有很强的关联性。由于交叉算子的存在,使得传统的线性系统的分析工具难以对所有算子的交互作用统一建模和分析。

在忽略变异算子时,通过对标准遗传算法进行动力学建模,可以获得包括交叉算子和选择算子的算法系统模型。为分析此模型在平凡不动点(也就是某个基因型充斥整个群体的状态)处的吸引性,需要计算模型在此处的 Jacobin 矩阵的特征值。由于选择算子和交叉算子的可分解性,可以通过分别计算这两个算子对应的矩阵而得到整个的矩阵。选择算子的 Jacobin 矩阵较为简单,是一个对角矩阵。而交叉算子就比较复杂。这也是以往研究工作遇到的主要难点之一。本文引入了一种依照 Hamming 距离对基因型重新排序的置换方法,在此置换下,交叉算子的 Jacobin 矩阵可以转换为上三角矩阵。这样,就能够计算出整个模型 Jacobin 矩阵的所有特征值。根据特征值

是否全部小于 1，可以将平凡不动点区分为可收敛的局部极值点和不可收敛到的排斥点两种类型。虽然以往的文献经常提到局部极值点概念，但却没给出明确的定义。基于上面的分析，本文给出了局部极值点的精确数学定义。

在剖析了上述分析成立的条件之后，可以发现，这种方法对于各种其他常见类型的进化算子都是适用的。基于此，经过详细分析基于排名和基于锦标赛的选择算子，以及其他类型的交叉算子，对这些常见类型的进化算子都获得了类似结论。在分析选择算子的同时，又发现锦标赛选择与线性排名的极限行为相一致。使用这些结论，可以对改进遗传算法提供理论上的指导作用。

要了解遗传算法的运行机理，不仅需要知道它的极限行为，而且需要了解算法在未收敛时的运行形态。简单地说，就是非平凡不动点的存在性和收敛性。本文针对一个具有代表性的简化问题，详细分析了遗传算法的全局动力学形态。由于在建模时加入了算法的控制参数，可以分析当这些参数变化时，算法动力学形态的稳定性。从这些分析中可以看出，对于这个问题，模型的全局动力学形态是比较确定的，模型由局部极值点和可能存在的鞍点构成。算法能够收敛到哪一个极值点完全由初始条件决定，并且对初值不敏感。当存在两个局部极值点时，还可以计算出每个点的收敛区域。当算法的变异概率很小时，模型与没有变异算子相类似。当变异算子足够大时，模型的动力学形态随着变异概率的增加发生了突变。上述结论有助于加深对 GA 动力学形态的理解，对于参数的选取具有一定的参考作用。

在评价单个选择算子的性能（即选择压力）时，经常使用取代时间的概念。在目前的研究中，对于取代时间的定义，混淆了群体规模和基因型种类这两个因素，因此本文重新定义了这个概念。由于大部分的定量分析结果都建立在某种特定的适应度函数以及均匀初始分布的假设基础上，所以没有取得一般性的结论。在本文中，详细地分析了基于比例的选择算子的取代时间，证明了取代时间的阶都是  $O(\ln N)$ ，不依赖于适应度函数的选取和初始群体的分布。据此，本文提出了取代时间系数的概念。这样，各种不同类型的适应度函数及其变换只影响取代时间系数，从而可以更加清晰地反映选择算子的效果。在此基础上，本文讨论了常见的适应度函数变换方式对选择压力的影响。

简单的遗传算法难以处理多模函数优化问题。为此，人们提出了各种改进方法，包括基于共享和基于拥挤等机制的生境遗传算法。这些机制提出的目的是为了在算法的运行过程中形成生境，以维持局部极值点的多样性。本文提出了一种基于聚类分析的资源竞争算法，可以不必预先确定生境的分布和生境半径，并对小适应度的局部极值点具有自适应性的排除能力。这种算法的计算量远小于基于共享的算法的计算量。从初步的数学分析和测试来看，该算法完全可以胜任多模函数的优化问题。

综上所述，在本文中，对于进化算法，特别是遗传算法的运行机理进行了一定程度的分析，得到了以下主要结论：

1. 定理：在使用基于适应度比例选择的标准 GA 中，对于任意  $i \in [0, m], i \neq k$ ，如果某个基因型  $k$  满足  $f_k > (inh(k, i))^{1/c} f_i$ ，则称在搜索空间中此基因型频率为 1 的点为局部极值点，GA 在局部极值点的邻域内收敛到此基因型频率为 1 的稳定状态。

2. 对于使用线性排名、指数排名和锦标赛选择等选择算子的 GA，存在类似上述定理的结论。

3. 对于 2-bit 问题，一般来说，在其状态空间内，存在两个局部极值点和一个鞍点，在鞍点处经过的不稳定流形，将状态空间分割为两个收敛区域，且每一部分一致收敛到其中的极值点。

4. 对于 2-bit 问题，当变异概率很小时，模型的动力学形态只有微小变化；当变异概率足够大时，模型突变为只有一个吸引子的全局动力学形态。

5. 基于适应度比例选择的选择算子的取代时间为  $O(\ln N)$ ，与适应度函数的具体形式无关和群体初始分布无关。

6. 使用基于聚类分析和资源竞争的遗传算法，可以达到对多模函数优化的目的。当局部极值点的适应度无法满足  $n \sum_{i=1}^n \frac{1}{f_i} f_n > c > 1$  时，这种算法可以自动将其排除。

基于本文得出的上述结论，可以更加深入地理解遗传算法的运行机理，设计新的算子和算法结构，调整算法的参数，以提高遗传算法的运行效率和使用效果。

## **ABSTRACT**

In many realms, large-scale optimization problems and other complicated problems have presented a serious challenge to the traditional optimization technology and artificial intelligence theory recently. A set of algorithms named as evolutionary algorithms (ECs) are brought forward for solving these problems. Evolutionary algorithms imitate thought of recombination, mutation and the survival of the fittest in nature evolution. These algorithms have several advantages: they don't depend on domain knowledge; they are not restricted by the limited factor of search space and they have the internal parallel character. Thus, the ECs are used in many fields successfully. In evolutionary algorithms, the genetic algorithm (GA) is typical, and used very widely.

The basis of GA's theory is not perfect compared to its obvious biological background. A series of theories are brought out to analyze some essential problems, such as convergence, convergence rate and applicability. As GA is parallel, stochastic, especially the crossover operator is nonlinear, so a consistent, completed, and practical theory analysis frame is still not constructed up to now.

In order to analyzing genetic algorithm, a lot of mathematics models are put forward. The early scheme theory doesn't function meeting with limit conditions. The limiting convergence conclusion is drawn for traditional GA and elite GA applying Markov chain method which depends on the unreduced character of mutation operator. Making practical application is a long distance to go. In course of proving, the effect of crossover operator is discussed, but it is just one factor acting as model status exchange operator, the function of crossover operator leading to final convergence goal is not put into consideration. In recent, some results are proved by

using dynamical system method on the population of unlimited individuals. It is shown that unlimited individuals population has close relationship to the limited. The traditional linear analysis tools become invalid because the crossover operator is nonlinear, so it is hard to modeling and analyzing the interaction of all evolutionary operators.

The model of standard GA, in which selection and crossover operator are included but mutation is not included, can be constructed using dynamical system method. The eigenvalues of Jacobin matrix must be calculated at each normal fixed point (the status that the population is filled with only one genotype) for analyzing the attraction character of this model. Because of the resolvability of model, the Jacobin matrix can be solved through the matrix of each operator. It is easy to determine the matrix of selection. This matrix is diagonal. It is the main obstruction former research meeting with for the matrix of crossover is difficult to calculate. However, the matrix can be converted to an upper triangle matrix under a new permutation based Hamming distance. Thus all eigenvalues of completed model can be calculated. The normal fixed point can be distinguished between repulsive point and local peak toward which model can be converged depending on whether each absolute value of eigenvalue is less than 1. The exact mathematics definition of local peak, which is usually mentioned in publications, is given basing on analysis.

After checking the conditions of above proven theory, it is clear that the theory is applicable for other familiar evolutionary operator. The similar conclusions are proved for other selection operators which are based on ranking and tournament. By the way, it is found that the limiting behavior are equivalent between linear ranking selection and tournament selection whose scale is 2.

The mechanism of GA includes not only limiting behavior but also the global running shape before convergence, in one word, the existence and convergence of non-normal fixed points. The dynamical shape of GA, which is used to solve for a representative simple problem, is exhibited. The stability of this shape is discussed on when control parameters are changed. From these analysis, the shape is almost certain for this problem when the mutation operator is omitted. There are local peaks and saddle points (if perhaps) in the dynamical shape. The arrived peak is determined by initial distribution. It is also not sensitive to the shift of initial values. When there are two local peaks, the convergent domain can be calculated. The model is not changed when mutation probability is small enough. When this probability beyond a boundary, the dynamic shape generates saltation. These conclusions is helpful to select appropriate parameters.

The conception of takeover time is usually used to estimate the performance of selection operator, that is to say, selection pressure. Many papers confuse the number of population with the number of genotypes, thus, the definition of takeover time should be re-defined. Moreover, a lot of quantitative results are based on some specific fitness function and uniformity initial distribution. In this paper, the takeover time of selection operator based on fitness function proportion is analyzed. It is proved that the rank of takeover time is  $O(\ln N)$ , and it does not depend on specific form of fitness function or initial population distribution. The takeover time coefficient is given to measure the performance of different selection operators. Each type of fitness function and its transform will only influence on this coefficient. The effect that some familiar transforms of fitness function have on selection pressure is discussed.

It is difficult to solve the multimodal function optimization



problem using standard genetic algorithm. Some improved algorithms are brought out, including niche GA which is based on sharing or crowding. The purpose of these algorithms is to maintain the diversity of local peaks through the creation of niche in running progress. A new niche GA that based on clustering and resource competition (BCRC) is approved, which is composed of clustering, sharing and crowding technology. A multimodal function can be optimized efficiently through BCRC without assurance of the count of niche and the value of niche radius in advance. This algorithm has the ability to exclude self-adaptively the local peaks whose fitness is small enough. The computation complexity is much less than the algorithm based on sharing. This algorithm is competent for multimodal function optimization from primary mathematical analysis and experiment test.

In all, the running mechanism of genetic algorithm is analyzed in this paper. There are some conclusions as follows:

1. In the standard GA that uses fitness function proportion selection operator, for each  $i \in [0, m], i \neq k$ , if any genotype  $k$  satisfies  $f_k \geq (inh(k, i) \cdot f_i) / f_i$ , the point in search space that the frequency of corresponding genotype is equals to 1 is called local peak. In the neighborhood of local peak, GA will converge toward the stable status that the population is filled with this genotype.

2. The above proof is also adaptable to other selection operators, such as linear ranking, exponential ranking and tournament selection.

3. There are two local peaks and one saddle point in the dynamic shape of GA for the 2-bit problem. The state space is divided into two domains by the unstable manifold that passes through the saddle point. All points in each domain converge entirely toward the local peak belongs to this domain.

4. For 2-bit problem, if mutation probability is small enough, the dynamical shape of model will have a little bit change. If mutation probability is big enough, the model will generate saltation, and have only a global attractor in state space.

5. The rank of takeover time is  $O(\ln N)$  for selection operator based on fitness function proportion, and it does not depend on specific form of fitness function or initial population distribution.

6. The niche GA that based on clustering and resource competition (BCRC) is competent for multimodal function optimization. If the fitness of a local peak can not satisfy  $f_n \geq \frac{1}{n} \sum_{i=1}^n f_i + c$ , this peak will be discarded automatically.

These conclusions are helpful for us to understand the running mechanism of GA deeply, to design new evolutionary operator and new algorithm structure, to improve the efficiency and effect of genetic algorithm.