

浙江大学

博士学位论文

进化策略学习、收敛和逃逸能力的研究及应用

姓名：阎岭

申请学位级别：博士

专业：控制理论与控制工程

指导教师：蒋静坪

20050615

## 中 文 摘 要

进化计算是一种受生物进化论启发而建立的优化算法,拓展了传统的计算模式,为复杂问题的求解提供了新的解决办法。将进化策略和学习算法相结合是当前的发展方向之一,但是理论研究相对较少,限制了它的进一步发展。本文首先分析了基于拉马克主义的进化学习策略(简称LELS)和基于达尔文主义的进化学习策略(简称DELS)的异同;利用马尔可夫链理论证明了此类算法的收敛性,并且在理论上分析了DELS具有更强的局部逃逸能力,算法的运行速度也明显得到了提高。

接下来本文分别将均匀变异算子引入进化策略和进化学习策略中,使得该类算法结构更简单,理论分析更方便,局部逃逸能力较好,并对收敛性进行了证明,最后的仿真实验表明了该方法的有效性。

模糊系统的可解释性明显优于其他人工智能方法,却长期没有引起人们足够的注意。本文对模糊系统的可解释性作了深入的分析,定义了最简约模糊划分、模糊划分的完备-清晰性、模糊规则的完备性、紧凑性和一致性,并将其加入到进化策略的适值函数中,用于优化模糊系统。即使在先验知识较少的情况下,该方法依然可以设计出具有较好系统响应性能和较高可解释性的模糊系统。

针对*K-means*聚类算法易陷入局部极小和*K*值选取的问题,本文提出一类基于进化策略的聚类算法,可以有效的搜索最优聚类中心和聚类个数*K*;还提出了确定*K*值范围经验公式,以减小搜索空间,提高搜索效率,并给出理论分析。

传统的观点认为进化策略是一种全局随机多点搜索技术,LMS算法是一种基于梯度法的单点搜索策略,这两种算法是两种截然不同的。本章对此进行了深入的探讨,认为这两者之间的差异更多的是基于我们平时的一些“想当然”的观念上的,进化策略可以看作是一种“概率”意义上的梯度搜索技术,而LMS算法也可以看作一种多点搜索技术。

关键词:进化策略 进化学习策略 拉马克主义 达尔文主义 LELS  
DELS 后天获得性遗传 模糊系统可解释性 最简约模糊划分 完备-清晰性  
一致性 *K-means* 聚类

### Abstract

The Lamarckian evolution learning strategy (LELS) and Darwinian evolution learning strategy (DELS) are presented in this paper. Both their similarities and differences in learning implementation are discussed. The former is based on Inheritance of acquired character, i.e., both phenotype and genotype can be optimized through learning. While the later only optimizes phenotype based on Darwinian selection. The convergence of ELS was proved using Markov chain theory. And we also theoretically demonstrated that DELS has stronger escaping capacities. These algorithms are applied to 8 standard test functions. Simulation results show that LELS and DELS yield faster convergence and better global optimization ability than standard evolution strategies; moreover, DELS also leads to better escaping capacity. Finally, the limitations of the work are discussed, as well as the future study.

The minimal fuzzy partition, completeness and distinguishability of fuzzy partition, and consistency of rule are defined, with further analysis on Interpretability. In an attempt to develop interpretable fuzzy systems, a systematic design paradigm is proposed using evolution strategy for mixed parameter optimization. The proposed approach is applied to the design of fuzzy controller for a nonlinear plant. It is verified that the interpretable fuzzy system works very well, especially when the priori Knowledge are insufficient.

*K-means* clustering has two disadvantages, one is easily trapped in local minimum, and the other is difficult to determine the number of clusters  $K$ . To address the problems, this paper proposes 3 new  $K$ -means algorithms based on Evolution Strategy. They own the easier coding scheme and the smaller population than Genetic Algorithm. These algorithms are applied to cluster Fisher's iris data set and work very well, especially when the priori Knowledge are insufficient.

**Keywords:** ES, ELS, Lamarckian, Darwinian, LELS, DELS, Inheritance of acquired character, Fuzzy System, Interpretability, Minimal Fuzzy Partition, Completeness and Distinguishability, Consistency,  $K$ -means clustering, global optimum

## 第一章 绪 论

### 1.1 进化计算的发展概况

20 世纪 40 年代以来,科学家不断努力从生物学中寻求用于计算科学和人工系统的新思想、新方法,比如早期的自动机理论就试图采用类似神经元的元素建造一种新型的思维机器。很多学者对关于从生物进化和遗传的机理中发展出适合于现实世界复杂适应系统 (complex adaptive systems) 研究的计算技术——自然进化系统计算模型 (computational models of natural evolutionary systems) 和模拟进化算法 (simulated evolutionary algorithm) 等,进行了开拓性的长期探索和研究[1~7]。在进化计算发展初期,人们曾对它寄予了厚望,希望它在不久的将来能全面达到或超过人的智力水平。该领域吸引了大量的研究人员,从而使得该领域得以蓬勃地发展。但在同时,我们也认识到了这一学科发展的困难和局限[8, 9]。

进化计算的思想可追溯到 20 世纪 50 年代[1~3, 10~13]。一般认为,进化计算 (evolutionary computation, EC) 包括三个组成部分[11~13]: ①由美国密歇根大学 John H. Holland 教授提出的遗传算法[1, 3]; ②由美国科学家 Lawrence J. Fogel 等人提出的进化规划 (evolutionary Programming, 简称 EP) [2, 12]; ③由德国科学家 Ingo Rechenberg 和 Hans-Paul Schwefel 提出的进化策略 (evolutionary strategy, 简称 ES) [12, 14]。他们用不同的进化模式模拟了生物进化过程,从而形成了三种具有普遍影响的模拟进化的优化计算方法。这三种方法也统称为进化算法 (evolutionary algorithms, 简称 EA)。

60 年代初, Holland 认识到了生物的遗传和进化现象与人工自适应系统的相似性,提出在研究和设计人工自适应系统时,可以借鉴生物遗传的机制,以群体的方法进行自适应搜索。他的学生 Bagley 于 1967 年在其博

士论文中首次提出了“遗传算法”一词[15]。1975年, Holland 出版了第一本系统论述遗传算法和人工自适应系统的专著《Adaptation in Natural and Artificial Systems》[16], 提出了模式定理 (Schema Theorem), 为遗传算法奠定了理论基础。

进化规划是 60 年代中期 Fogel 等人为有限状态机的演化而提出的一类进化算法。与遗传算法的不同在于, 进化规划注重父代与子代的表现行为而不是遗传细节[17]。1966 年, Fogel 出版了《Artificial Intelligence through Simulated Evolution》[18], 系统阐述了进化规划的思想。但当时学术界对在人工智能领域采用进化规划表示怀疑, 直到九十年代初才逐步被学术界重视。

进化策略的研究始于 1964 年。当初主要用于试验流体动力学问题, 如弯管形状优化[19]。由于当时现有的一些优化算法不适于解决这类问题, Rechenberg 提出按照自然突变和自然选择的生物进化思想, 对物体的外形参数进行随机变化并尝试其结果。后来, Schwefel 系统地推广了 Rechenberg 的二元进化策略 (又称(1+1)-ES), 建立了多元进化策略, 即  $(\mu + \lambda)$ -ES 和  $(\mu, \lambda)$ -ES。

有关进化计算的研究主要集中在以下几个方面[9, 84]:

## 1. 算法的基础理论研究

### 1.1 随机模型理论

对于有限的编码空间和有限的群体, 遗传算法的搜索过程可以表示为离散时间的马尔可夫链模型 (Markov chain model), 从而可以采用已有的随机过程理论进行严密分析。Goldberg 和 Segrest[20]运用有限马尔可夫链理论对遗传算法进行了收敛性分析, Eiben 等人证明了一类抽象遗传算法在精英选择情况下的概率收敛情况[21], Rudolph 用齐次有限马尔可夫链证

明了带有选择、交叉、和变异操作的标准遗传算法收敛不到全局最优解，但是如果让每一代群体中的最佳个体不参加交叉与变异操作而直接保留到子代，那么遗传算法是收敛的[22]。Qi 和 Palmieri 对实数编码的遗传算法，在基于连续空间中群体规模为无穷大这一假设下进行了严密的数学分析[23]。Fogel 和 Suzuki 从进化计算的角度对 GA 收敛问题进行了研究[24~26]。李书全等采用泛函分析理论证明了 GA 的收敛性[27]。

Vose, Nix, Liepins 等采用统计动力学方法，分析了无穷群体下 GA 的搜索轨迹和不动点收敛性[7]。Cerfdeng 采用摄动理论和马尔可夫链，对遗传算法的渐进收敛特性进行分析，得出了遗传算法运行及全局收敛性的一般性结论[28]。Bäck[29]和 Muhlenbein[30]等研究了到达全局最优解的遗传算法的时间复杂性问题，恽为民等基于马尔可夫链对此进行了进一步的分析[31]。

## 1.2 进化动力学理论

进化动力学理论的基本内容是对遗传算法的进化行为进行分析，其基本定理即为 Holland 提出的模式定理，但是模式定理仅仅描述了模式的生存模型，没有反映模式的重组过程，所以有限群体下的模式定理不保证遗传搜索的全局收敛性。建筑模块假说描述了遗传算子的重组功能，与模式定理一起构成了求解优化问题时，遗传算法具备发现全局最优解的充分条件，也是分析遗传算法进化行为的基本理论，统称为模式理论。

## 2. 遗传策略的研究

遗传策略研究与设计的主要内容是决定遗传算法采取何种适宜的运算形式，来维持群体可进化性并最终搜索到问题的全局最优解。遗传算法应用于优化问题求解，可以视为一种随机化搜索过程。在该搜索过程中，

GA 不仅需要搜索解空间上的全局最优解 (exploration), 而且应当充分利用已获得的解空间信息逼近当前局部最优解(exploitation), 我们分别称之为 GA 的求泛和求精(reforming and refining)能力。但是这两种能力并非可以同时获得, 对于任何一种算法来讲它们构成了一对矛盾, 求精能力好的算法往往不具备良好的解空间上的探索能力, 反之亦然。因此, 遗传策略研究与设计是一个重要的研究方向[3~7, 32~61], 我们可以将之分为微观遗传策略 (micro genetic strategy) 和宏观遗传策略 (macro genetic strategy)。

微观遗传策略主要讨论群体规模、遗传算子的形式和参数设计, 及其对 GA 求解能力的影响。微观遗传策略的理论研究一直集中于遗传算子的适应性的控制, 即进化过程中遗传算子参数的适应性调整, 进而达到预期的搜索目标[4, 62~64], 并针对一组函数进行了详细测试。进化策略是最早引入策略参数的适应性设计的[12~14, 65]。同时, 在遗传算法、进化规划等算法中, 也比较注重策略参数的适应性设计[65]。Whitley 采用微分方程和马尔可夫链理论, 对 GA 与特定步数 Baldwin 或者 Lamarck 局部搜索相结合的进化过程进行了建模分析[66]。

遗传算法的宏观策略主要讨论关于通过对 GA 流程的再设计改变 GA 的宏观特征, 或者以 GA 流程为基础, 引入其他算法构成混合 GA (hybrid genetic algorithms, H-GA) [63, 67~69], 以期提高 GA 求解问题全局最优解的能力。

另外, 针对多模态函数优化问题 (multi-modal function optimization), 如何构造一种优化算法, 使之能够搜索到全部或者尽量多的全局最优解和有意义的局部最优解, 已经成为一个重要的研究领域[61, 70]。De Jong 等人提出了排挤模型 (crowding model) [16, 71, 72]、概率排挤模型 (deterministic crowding with probabilistic replacement) 等[73]。Goldberg 和 Richardson 提出了适应值共享模型 (fitness-sharing model) [41, 61, 70], 基于适应值共享机制的小生境技术 (niching technology) [41, 74], 通过定义群体中个体的共享度, 调整个体的适应值, 使得群体保持多个高阶模式。

### 3. 进化计算的编码方式

早期的遗传算法大多采用二进制编码。Goldberg 和 Smith 用动态背包问题进行了比较研究, 试验表明双倍体比单倍体的动态跟踪能力强[4]。进化策略则一开始就采用实数编码。实数编码具有精度高、便于大空间搜索的优点, 越来越受到重视, Michalewicz 比较了两种编码方法的优缺点[75, 76], Qi 和 Plmieri 对实数编码的遗传算法进行了严密的数学分析[77]。Vose 等扩展了 Holland 的模式概念, 揭示了不同编码之间的同构性[78]。

从整体上讲, 二进制编码的进化层次是基因, 而实数编码的进化层次是个体。同时对于非二进制编码, 可以结合具体问题领域的知识, 设计合适的遗传算子。

### 4. 进化计算的其他问题

一、no free lunch (NFL) 定理[79]。NFL 定理的主要结论为: 对任意的表现度量, 当对所有可能的目标函数作平均时, 所有搜索算法的表现是完全一样的。因此, 一个特定的优化方法只能对于某个特定领域的问题, 亦即所有目标函数的一个子集来讲, 优于另一个算法。在实践中, 对一个特定算法如何确定它的适用函数子集, 或者对于具体的优化问题如何选择和设计适宜的算法, 有着重大的意义。

二、实际应用中涉及的问题大多数是带有约束条件的, 求解约束化问题是对进化计算的巨大挑战[12, 13, 80]。能否处理好约束问题, 是能否成功应用进化计算的一个非常关键的问题。

三、遗传算法的并行化研究 (parallel GA, 简称 PGA) [81]。设计各种并行执行策略、建立相应的并行化算法的数学基础, 对提高进化算法的效率有着重要意义。PGA 主要有细粒度和粗粒度两种计算模型, 具体实现



的方法又同步主从式、异步并发式和网络分布式等三种[82, 83]。

## 1.2 进化计算原理与特点

进化算法是一种基于自然选择和遗传变异等生物进化机制的全局性概率搜索算法。与基于导数的解析方法和其他启发式搜索方法（如模拟退火方法，单纯形法等）一样，进化算法在形式上也是一种迭代方法。它从选定的初始解出发，通过不断迭代逐步改进当前解，直至最后搜索到最优解或满意解。在进化计算中，迭代计算过程采用了模拟生物体的进化机制，从一组解（群体）出发，采用类似于自然选择和有性繁殖的方式，在继承原有优良基因的基础上，生成具有更好性能指标的下一代解的群体。

优化问题采用进化计算求解的一般过程包括以下步骤：

- ①随机给定一组初始解；
- ②评价当前这组解的性能；
- ③根据②的评价结果，从当前解中选择一定数量的解作为基因操作对象；
- ④对所选择的解进行基因操作（交叉、变异），得到一组新解；
- ⑤返回②，对该组新的解进行评价；
- ⑥若当前解满足要求或进化过程达到一定的代数，计算结束，否则转向③；

进化算法是一种随机化搜索方法，在初始解生成以及选择、交叉与变异等遗传操作过程中，大多采用了随机处理方法。与其他搜索技术相比，进化算法具有以下特点：

1. 进化计算在搜索过程中使用的是基于目标函数值的评价信息，而不是传统方法主要采用的目标函数导数信息或待求问题领域内知识。进化

算法的这一特点使其成为具有良好普适性和可规模化的优化方法。

2. 进化计算具有显著的隐式并行性 (implicit parallelism)。进化算法虽然在每一代只对有限解个体进行操作, 但处理的信息量为群体规模的高次方。

3. 进化算法在形式上简单明了, 不仅便于与其他方法相结合, 而且非常适合在大规模并行计算机上运行, 因此可以有效地用于解决复杂的适应性系统模拟和优化问题。

4. 进化算法具有较强的鲁棒性, 即在存在噪声的情况下, 对同一问题的进化算法的多次求解中得到的结果是相似的。进化算法的鲁棒性在大量的应用实例中的到了充分的验证。

### 1.3 进化策略简介

进化策略(ES, Evolution Strategies)是一类仿效自然界进化规律以解决参数优化问题的方法。它是 60 年代在德国发展起来的[12, 14]。如文献[58]中所描述的:

“在 1963 年, 两个德国柏林技术大学的学生碰到一起, 并不久合作于流体工程学院的风洞实验。在流动中优化特体形状的研究中, 一开始孕育的思想是策略性的。但试图用坐标和简单的梯度法没有成功的。然后, 其中一个学生, Ingo Rechenberg, 现在为仿生学和进化工程(Bionics and Evolutionary Engineering)的教授, 偶然想到一个主意, 即试着按照自然变异的实例随机改变定义形状的参数。进化策略于是就诞生了。”(第二个学生是 Hans-Paul Schwefel, 现在为计算机科学教授及系统分析主席。)

早期的进化策略可以被看成使用浮点数表达, 只使用变异作为其遗传算子的演化程序。它们已经应用于各种连续可变参数的优化问题。只是在近来, 它们被扩展到各种问题上[110, 111]。

### 1.3.1 二元进化策略

进化策略可以分成二元进化策略(two-membered evolution strategies)和多元进化策略(multi-membered evolution strategies)。我们先从二元进化策略开始讨论。

二元进化策略是进化策略中最简单的一种形式。描述如下：

1. 初始化： 随机生成 1 个个体，构成第 1 代群体的父代。
2. 变异： 父代个体将自己的基因经过变异后产生 1 个子代，种群数为 2。  
子代个体与父代个体在基因上有所不同。
3. 选择： 每个个体由于对环境的适应性不同而表现出不同的生命力或适应度，它由遗传基因决定。2 个个体中只有生命力较强的个体成为下一代的父辈个体。如果终止条件满足，则算法结束。否则回到步骤(2)。

以下用  $N(0,1)$  表示一个服从期望为 0，标准差为 1 的一维正态分布的随机变量的实现， $N_i(0,1)$  表示对于下标  $i$  的每个值随机变量要重新抽样。一个服从期望为 0，标准差为  $\sigma$  的一维正态分布随机变量的实现就可以由  $\sigma * N(0,1)$  给出。

二元进化策略的编译操作是在原个体上加上一个满足正态分布随机变量来实现，如下所示：

$$\mathbf{I}' = \mathbf{I} + \sigma * \mathbf{N}(0,1) \quad (1.1)$$

其中  $\sigma$  称为步长。步长的选择非常重要，

为了提高搜索效率，Rechenberg 建议用一个“1/5 成功法则”：

对所有的变异，成功变异的比率应该是 1/5。如果大于 1/5，增加变异算子的方差；否则，减少其方差。

Schwefel 在实际运算中建议如下：

$$\sigma' = \begin{cases} c_d * \sigma & \text{if } \varphi(k) < 1/5 \\ c_i * \sigma & \text{if } \varphi(k) > 1/5 \\ \sigma & \text{if } \varphi(k) = 1/5 \end{cases} \quad (1.2)$$

其中,  $\varphi(k)$  是最近  $k$  代中解变好的次数与  $k$  的比值;  $c_d=0.82$ ,  $c_i=1.22=1/0.82$ 。

### 1.3.2 多元进化策略

多元进化策略简记为  $(\mu, \lambda)-ES$  和  $(\mu + \lambda)-ES$  描述如下:

1. **初始化:** 随机生成  $\mu$  个个体, 构成第 1 代群体的父代。其中每个个体的不同是由个体的基因型的不同而决定的。
2. **变异:** 每个父代个体将自己的基因经过变异后产生  $\lambda/\mu$  个子代, 总共生成  $\lambda$  个子代个体。此  $\lambda$  个子代个体与其各自的父代个体在基因上有所不同。
3. **选择:** 每个个体由于对环境的适应性不同而表现出不同的生命力或适应度, 它由遗传基因决定。在  $\lambda$  个个体中  $(\mu, \lambda)-ES$  或者  $\mu + \lambda$  个个体中  $(\mu + \lambda)-ES$  选择生命力较强的  $\mu$  个个体成为下一代的父辈个体。如果终止条件满足, 则算法结束。否则回到步骤 (2)。

$(\mu, \lambda)-ES$  有较大的优越性, 在实际应用中取得了巨大的成功, 因此, 也是本文的主要研究对象。为方便起见, 本文以后除特别声明, 进化策略均特指  $(\mu, \lambda)-ES$ 。其特点将在 1.1.3 节中详细介绍。

### 1.3.3 进化策略的特点 [9, 106]

ES 是最早引入自适应机制的算法[58]。它与最广为人知的遗传算法相比, 在编码、种群规模、交叉、变异和选择等诸多方面有着许多的不同。

## 1. 编码

在进化策略中，目标参数和策略参数都需要编码到染色体中。目标参数是指直接涉及适应值计算的参数[107]。策略参数即应用于进化算法中的控制参数，例如种群规模、突变步长、突变频率、交叉位置、交叉频率等等。一般而言，策略参数的选取对进化算法的性能有着直接的影响。因此，人们希望策略参数和目标参数在进化的过程中可以同时得到优化，这就是“自适应”的由来。

与经典遗传算法采用二进制编码不同，进化策略采用实数编码将目标变量直接编码到染色体中。它的另一个重要特征是将策略参数也编码到染色体中。因此，一个目标向量  $X$  的染色体可以表示为：

$$X = [x_1 x_2 \cdots x_n \sigma_1 \sigma_2 \cdots \sigma_m] \quad (1.3)$$

其中  $x_i, i=1,2,\cdots,n$  是目标参数，而  $\sigma_j, j=1,2,\cdots,m$  是策略参数。策略参数的数目在不同的进化策略中是不同的。一般来说，遗传算法的编码方式比进化策略有着更大的灵活性，这或许是遗传算法应用更为广泛的原因之一。随着各种进化算法相互影响和融合，也有许多学者致力于实数编码的遗传算法的研究[75~77]。

## 2. 变异算子

与遗传算法不同，变异算子是进化策略的主要算子。在经典进化策略中，变异是通过给目标参数上加上一个服从正态分布的随机数来实现的。

进化策略目标参数和步长参数的变异公式如下：

$$\sigma'_i = \sigma_i \cdot \exp(\tau_1 \cdot N(0,1) + \tau_2 \cdot N_i(0,1)) \quad (1.4)$$

$$I'_i = I_i + \sigma'_i \cdot N_i(0,1) \quad (1.5)$$

其中， $\tau_1$  是标准差向量  $\sigma$  的整体步长参数， $\tau_2$  是  $\sigma$  的每个分量  $\sigma_i$  的步

长参数。因此  $\tau_1$  和  $\tau_2$  是给定的外部参数, Schwefel 建议如下设置[58]:

$$\tau_1 \propto (\sqrt{2\sqrt{n}})^{-1}, \quad \tau_2 \propto (\sqrt{2n})^{-1} \quad (1.6)$$

其中  $\propto$  表示成正比。通常  $\tau_1$  和  $\tau_2$  的比例因子取为 1。

这一特别的变异机制使得 ES 能够进化本身的策略参数, 在搜索的过程中, 开发利用了适宜的内在模式和好的适应值之间的一个隐含的关联, 由此所导致的根据适应值曲面的拓扑特性而使策略参数得到进化的机制, 被称为自适应机制。

### 3. 交叉算子

交叉操作是提供两个不同个体之间交换信息的机制。与遗传算法不同, 交叉算子在进化策略中并不是必需的。文献[58]中介绍了以下四种交叉操作:

$$x'_i = \begin{cases} x_{a,i} \text{ or } x_{b,i} & \text{discrete-recombination} \\ \frac{1}{2}(x_{a,i} + x_{b,i}) & \text{intermediate-recombination} \\ x_{a,i} \text{ or } x_{b,i} & \text{global-discrete-recombination} \\ \frac{1}{2}(x_{a,i} + x_{b,i}) & \text{global-intermediate-recombination} \end{cases} \quad (1.7)$$

其中  $a, b, b_j$  是父代个体,  $i$  是向量  $x$  第  $i$  个元素。通常来说, 在一代群体中, 所有的父代个体具有相同的交叉概率, 即参加交叉的父代是通过均匀随机数来确定的。在离散交叉情况下, 子代的成员向量的每一维都是随机继承某个父代, 即  $x_{a,i}$  或  $x_{b,i}$  的对应部分, 相当于遗传算法中交叉点数量可变的交叉算子。在中间交叉情况下, 子代的成员向量是父代成员向量的平均值。在全局交叉情况下, 通过选取另外的父代  $x_{b_j,i}$  使子代继承群体中更多的遗传信息。文献[108]认为交叉操作可以有效的提高进化策略的性能, 但是出于方便和效率方面的考虑, 许多应用中均不采用交叉算子。

#### 4. 选择方法

ES 中的选择方式是完全确定性的。 $(\mu, \lambda)$ -ES 是从  $\lambda$  个子代个体中选择  $\mu$  个最好的个体作为下一代的父代； $(\mu + \lambda)$ -ES 是从  $\lambda$  个子代个体和  $\mu$  个父代中个体选择  $\mu$  个最好的个体作为下一代的父代，即精英选择方法，从而保证了性能的改进是单调的，但是该选择方法无法适应变动的环境，不利于实施策略参数的自适应机制。因此， $(\mu, \lambda)$ -ES 在今天更受到推崇。文献[58]认为，比率  $\mu/\lambda \approx 1/7$  是最优的。

##### 1.3.4 进化策略的基本流程

ES 的基本流程可以描述如下：

Begin

    给定各个参数值；

    随机生成  $\mu$  个个体的初始化群体  $P(0) = \{x_{0,1}, x_{0,1}, \dots, x_{0,\mu}\}$ ；

    计算每个个体的适应值；

    while (中止条件不满足) do

$t = t + 1$ ;

        对  $P(t-1)$  中每一个父代个体按照公式(1.4)、(1.5)平均产生  $\lambda/\mu$  个子代个体，整个群体共产生  $\lambda$  个个体，构成群体

$P'(t)$ ;

        选择  $(\mu, \lambda)$ -ES：计算  $P'(t)$  中个体的适应值，选出最好的个个体构成下一代群体  $P(t) = \{x_{t,1}, x_{t,1}, \dots, x_{t,\mu}\}$ ；

        选择  $(\mu + \lambda)$ -ES：计算  $P'(t)$  中个体的适应值，从  $P(t-1) \cup P'(t)$  中选出最好的个个体构成下一代群体  $P(t)$

$= \{x_{t,1}, x_{t,1}, \dots, x_{t,\mu}\}$ ；

    end

end

## 1.4 进化论的发展概况及对进化计算的影响

进化论是进化计算诞生的土壤，但是进化论并非铁板一块，它产生过众多的学派，各学派的观点针锋相对，而且进化论本身也是在不断“进化”，涌现出更多新的理论和流派，这为我们提供了新的角度来重新认识进化论和进化计算。

### 1.4.1 进化没有方向没有终点[85]

以往的进化论，被增加了许多不恰当的修饰成分。例如，1999年版的《辞海》这样定义“进化”：“生物逐渐演变，由低级到高级、由简单到复杂、种类由少到多的发展过程。”这也是引用最多、最广为人知的定义。而现代进化论却认为：进化是生物群体的遗传组成或其比例的不可逆的改变，而并不必然伴随着进步、改善，或复杂化，也不一定是逐渐的，但进化必定是趋向适应的。

比如，在有些时期，有些物种的进化是逐渐的；而在另一些时期，对另一些物种，其进化则可能是突发、跳跃的。被美国《纽约时报》列为20世纪最惊人的科学发现之一的我国云南的澄江动物群就是爆发性跃进的进化动物群典型。类似的爆发性跃进现象，在生命演化史上至少还有“埃迪亚卡拉大爆发”、“三叠大爆发”、“早第三纪大爆发”等。

生物的进化归根结蒂是基因突变引起的，而一个基因的微小突变就有可能引起表型的重大改变，要在鱼鳍中生出骨头（进而变成脚，进化成两栖类），只要一个或几个基因发生突变就可以一步到位，并不需要十分之一骨头，十分之二骨头等等一系列的过渡型。



鉴于大量事实，美国埃尔德雷奇和古尔德于 1972 年提出了一个全新的生物进化理论——“间断平衡论”。该理论指出，生物的进化不像达尔文所强调的那种连续渐进的进化过程、线性进化模式和缓慢变异积累的新种形成作用，而是渐进与跃进交替的进化过程、间断平衡的进化模式，以及基因突变或地理隔离的成种作用。

进化也未必是由低级到高级的发展过程。所谓高级、低级，不同的标准可以有不同的结果。或许可以简单的认为，一个种类对环境的适应程度，标志着它就高级与否。但是要记住环境不断变化这个前提。在不同的环境下比较物种的适应性可以说没什么意义。正如系统论和控制论中指出的那样，如果目标变化，最优控制可能会变成最差的。例如在六千五百万年前，环境的急剧变化（目前最令人信服的解释是陨石撞击地球），导致了恐龙的灭绝，而哺乳动物却顽强地生存了下来，其中的一个因素是当时的哺乳动物都只有老鼠那么大小甚至更小，靠吃种子、啃树根熬过了这场全球性大饥荒。但是在此之前，体形小显然不是什么高级的特征，哺乳动物实际上已被体形庞大的恐龙压抑了一亿年才等到了这个幸运的一击，从此代替恐龙成为地球的主宰，而且也进化出了象、鲸这样的庞然大物！在同时存在的生物种类之间作低级、高级的比较也是没有意义的，因为进化大树有无数的分枝，很难说哪个枝头更高一些。生物学上有所谓低等生物与高等生物的划分，不过是说低等生物比高等生物出现的时间更早，保留共同祖先的特征更多，因而在进化上更“原始”。这也许是一种省事的分类法，但是我们不能因此认为在进化上越“原始”对环境的适应就越不成功。事实可能正好相反。原核生物（细菌）要比真核生物更原始，但它们的种类之多、数目之巨、对环境适应之强却是真核生物所望尘莫及的。在哺乳类大家庭中，人类所属的灵长类是相当原始的一个分支，但是人类无疑是哺乳类中最成功的物种。

进化也未必是一个由简单到复杂的发展过程。总体上似乎如此，晚出现的真核生物要比早出现的原核生物复杂，多细胞生物要比单细胞生物复

杂，脊椎动物要比无脊椎动物复杂。但是请不要忘记，原核生物、单细胞生物、无脊椎动物并没有被真核生物、多细胞生物、脊椎动物所取代，它们同时也在不同的途径上进化。这些途径，可能是由简单到复杂，但也可能是由简单到简单，甚至是由复杂到简单！比如对于寄生物，它们的细胞、组织、器官逐步退化，形态结构就是越来越简单，病毒甚至于退化到只剩下最少量的遗传物质和一个外壳。也就是说，在进化上，复杂的必然由简单的演变而来，但简单的并不一定会变成复杂，复杂的反而可能变得简单。

进化也未必是种类由少到多的过程。如果拿今天的几百万个物种与生命起源之初只有一个或几个物种相比，似乎可以下这个结论。但是这个结论忽视了这么一个事实：从生命起源到今天的三十亿年内，不断地有新物种产生，同时也不断地有旧物种灭绝；在各个时期，种类时多时少，并非累积递增。比如在著名的“寒武纪物种大爆发”中，在短短的五百万年内产生的动物类群就要比今天的多。寒武纪之后的各动物类群不过是这次大爆发后的幸存者的进一步进化而已。而最近的研究表明，即使是在特定的某一类群内，其种类也不是不断扩增的，比如哺乳类，在两亿年的进化史中，其种类时增时减，始终保持在大约 90 个属。可见生物进化既是种类由少到多的过程，也是种类由多到少的过程。在发生物种大灭绝时，后一过程表现得更为明显，而物种大灭绝平均两千六百万年发生一次。另外一句题外话：今天我们人类正以平均每天两个物种的速度消灭野生动植物，而新的物种并没有产生。

所有生存过的物种中，百分之九十九以上已灭绝了。这个悲惨的事实说明了一点：进化是一个试错的过程，试验，失败，再试验，再失败。它不是由低级到高级，也不是由简单到复杂，更不是种类由少到多。进化没有方向，也没有终点。人们往往把进化树画成存在着一个从单细胞生物进化到人类的主干，把人类作为进化的最终产物置于进化树的最顶端，这实际上不过是人类的主观上的自慰而已。客观地说，人类的产生是无数偶然

事件的结果。如果在寒武纪的物种大爆发中，最初的弱小的脊椎动物没能幸存下来，则今天或许根本不会有脊椎动物；如果一种不起眼的鱼类不是偶然在鱼鳍中长出了骨头，则脊椎动物或许根本不可能登上陆地；如果不是一个偶然事件导致恐龙的灭绝，则小小的哺乳动物可能根本没有机会主宰地球；而如果在大约四百万年前非洲大草原的气候不变干燥，迫使南方古猿的祖先放弃丛林生活下地直立行走，则所谓的人类也许不过是另一类猩猩。

弗洛伊德曾经说过，科学上的重大发现往往带来对人类尊严的重大打击。哥白尼的日心说让我们的家园从宇宙的中心“迁”到了一个微不足道的角落，达尔文的进化论让我们知道我们的祖先并不高贵，不过是一种猿，而弗洛伊德的精神分析说——他毫不谦逊地说——使我们人类再也无法认为自己是理性的动物。而现在我们可以说：地球上产生生命也许是必然的，但进化出人类却是十分偶然的。我们人类不过是进化偶然的产物。

#### 1.4.2 “Evolution”译成“进化”或“演化”的思考[86]

现代进化论一般认为，进化（Evolution）是生物群体的遗传组成或其比例的不可逆的改变，它只是趋向适应的，而并不必然伴随着进步、改善，或复杂化。分子进化学者甚至将蛋白质中一个氨基酸的置换或核酸中一个碱基的置换也视为进化。鉴于 Evolution 一词具有其特定的内涵，故将它译成进化或演化，在学术界一直存在争议。

主张翻译成“演化”的学者认为，“演”字与“进”字比较，在语意上，“演”字既含有连续的意味且随机性大些；而“进”字则多带有进步和由低等向高等直线发展的含意。事实上，在生物进化的过程中，既有由简到繁的变化，也有由繁到简的退化。这种退化也是“进化”。汉语中“进”与“退”是代表截然相反过程的两个字，说退化也是进化在逻辑上似有不妥。但把 Evolution 译成演化就不存在这个问题了。

英语 Evolution 一词源于拉丁语的 *evolvere*, 表示将卷着的东西铺展开来之意。Evolution 最早是被用于哲学上的。17 世纪后半叶, 新柏拉图主义者首先使用了 Evolution 一词, 含意是“历史的展开”。不久博物学界(当时还无生物学这一概念)胚胎学的前定论者(*preformationist*)引入 evolution, 定义为“生命的展开”。

英国地质学家莱伊尔(*Charles Lyell, 1797-1875*)在《地质学原理》中提出了地质渐变的观点即均变论(*Uniformitarianism*)。均变论的核心思想是认为地质纪元中的一切地质变化都是逐渐发生的, 并且认为“现在是鉴别过去的钥匙”。莱伊尔的这部著作对达尔文后来形成的进化理论产生了深刻的影响。但是, 莱伊尔对待生物世界却是长期坚持特创论的立场, 反对法国学者拉马克提出的生物进化理论, 并将其称之为“*Evolution Theory*”。显然, Evolution 在此是贬意的。

达尔文用 *Transmutation* 和 *Descent with modification* 词组来理喻他对生物进化的认识, 而未使用 Evolution。*Transmutation* 是演变、衍变、变形、变质的意思。*Mutation* 是变化的意思, 现在作为遗传学的专门用语意为突变。而 *trans* 则有贯通、转移、超越之意。因此达尔文选用 *Transmutation* 一词旨在体现他的连续进化和渐进进化的思想。

*Descent with modification* 是伴有变更(或后天变异)的世代传承的意思。世代传承强调生物具有连续性、保守性、遗传性的特征。而这种特征又不是一成不变的, 每一代又伴有 *modification* (变更、后天变异)。这个词组更明确地表达出达尔文关于生物渐进进化的思想。

达尔文未采用 Evolution 一词来表述生物进化有以下 3 个原因[86]:

1. Evolution 表示将卷着的现成的东西展开, 欠缺变更或增添新质的含意。
2. Evolution 一词具有并不光彩的历史。唯心主义宗教哲学家剑桥新柏拉图主义者用过它; 反科学的前定论者也用它表述过其主张。
3. 达尔文为了表示自己的进化理论有别于拉马克的进化理论, 因为

莱伊尔曾不怀好意地称拉马克的理论为 Evolution Theory。

19 世纪 60 年代以后, 英国斯宾塞(Hebert Spencer, 1826~1903), 在其巨著《综合哲学体系》中, 详细介绍了达尔文的进化理论, 用 Evolution 一词取代了达尔文原来使用的 Transmutation 和 Descent with modification。从那以后, 学术界普遍接受了斯宾塞的用法, 也以 Evolution 一词表示生物进化现象。

日本明治维新(1868~1873)之后, 积极地学习西方, 科学教育文化尤其率先。在进化论引入日本时, Evolution 一词被日本著名翻译家井上哲次郎(原名船越哲次郎, 1855~1944)译为“进化”。甲午战争(1894—1895)战败后, 清朝政府认识到学习西方的必要性。于是在 1896 年(光绪 22 年)春, 多次派遣留日学生, 由此“进化”和“进化论”等词汇开始传入我国。

但是早在留日学生将达尔文进化理论介绍到中国来之前, 我国清末民初著名的启蒙思想家严复已将达尔文的进化理论介绍给国人。严复(1853~1921), 曾访问日本的长崎、横滨, 并于 1876 年被派赴英国留学。

甲午战争后, 严复将英国著名生物学者、达尔文主义的忠诚拥护者赫胥黎(Thomas Huxley, 1825~1895)著的《进化与伦理学》翻译成中文, 即《天演论》。该书为赫胥黎 1894 年所著, 是一本介绍达尔文进化学说的通俗读物。严复取英文原书名的第一个词“Evolution”作为中译本的书名——《天演论》。“天”在古汉语中可代表大自然, “演”代表演变。“天演”乃生物藉自然之造化, 即藉自然选择之力而发生演变之意。言简意赅, 寓意深刻, 符合“信、达、雅”的要求。

由此可见, “进化”一词来源于日本, 而“演化”一词来源于严复的《天演论》。本文也认为“演化”比“进化”更为准确一些。但是由于历史的原因, 时代已经赋予“进化”新的内涵, 我们没有必要再改为“演化”, 正如“Evolution”一词最初也不能表述“达尔文进化论”, 但是今天却最为通用一样。然而我们应该清楚“进化”与“演化”的区别以及来源, 这对于我们学习进化论以及以新的角度认识进化计算, 有着重要的意义。

### 1.4.3 进化论的新发展对进化计算的影响

进化论发展到今天，产生了许多新的观点和不同的流派。例如，现在的主流学派越来越倾向于认为，进化只是生物群体的遗传组成或其比例的不可逆的改变，而并不必然伴随着进步、改善，或复杂化，但进化必定是趋向适应的；接受达尔文的自然选择，抛弃拉马克的用进废退；接受古尔德渐进与跃进交替的“间断平衡论”，而非达尔文所强调的那种连续渐进的进化过程。此外还有木村的“中性进化假说”等等。

进化计算(EC)是以进化论为生物学背景，将生物进化过程简化并模拟的一大类优化算法，在多个领域中取得了成功。进化论的新发展，必然为我们重新认识和改造进化计算提供了新的角度。

例如，当前有一种较为流行的错误观点，认为进化计算的三个主要分支：遗传算法(GA)、进化策略(ES)和进化规划(EP)，只是各有侧重，而并无根本区别，并且三者之间的差异越来越小，有着趋向统一的趋势。本文认为，这三种算法虽然都是受到进化论启发而发展起来的，但是他们却有着不同的生物学依据和理论基础。例如遗传算法依据的是达尔文所强调的“连续渐进”的进化理论，并不十分看重“变异”的作用；而进化策略和进化规划更主要的是受到古尔德“间断平衡论”的影响，主张渐进与跃进交替，因此“变异”的作用十分重要，而“交叉”，即有性生殖，并不十分看重。所以遗传算法侧重交叉算子，而进化策略侧重变异算子，进化策略发展到后期，才引入交叉算子，但是考虑到计算的复杂度以及实时性问题，相当多的场合是不采用交叉算子的；进化规划更是只用变异算子，完全放弃交叉算子。长期以来各学派努力不止，争论也不休。比如文[87]认为与变异相比较，交叉没有任何竞争优势；文[88]则批评了这一论断，认为对某些问题仅有变异算子是不够的。而文[89]的结论是将连续参数优化

问题表示成二进制位串并且允许用交叉来做搜索的做法并不总是最好的。Spears 从定义杂交和变异的两个潜在作用——分裂(disruption)和构造(construction)出发,考虑它们在执行这两个作用上的差异,研究结果表明:对于分裂,变异比杂交有效,虽然它缺少杂交保留个体共同等位基因的能力;然而,对于构造,杂交比变异更有效[90]。

受进化论发展的启发而对经典进化计算进行改造,也是目前的研究方向之一。例如将传统优化算法作为学习算子引入到进化计算中,是近年来的研究热点之一。文献[91, 92]将遗传算法与单纯形法相结合,即后代个体利用单纯形法进行局部搜索,以提高适应度;[93]明确提出将单纯形法作为学习算子;[94]提出是基于拉马克主义的,即后天获得性遗传;[95]给出了混和遗传算法的收敛性证明;此外[96~98]是将遗传算法与梯度法或其他局部优化算法相结合,进行相关的研究。文献[99, 100]详细分析了基于拉马克主义和达尔文主义的学习算子的异同,证明了该算法的收敛性,并在理论上作了局部逃逸能力的分析。

R. H. Macarthur [101] 认为在稳定的环境中,动物的繁衍有可能接近环境容纳量,即接近逻辑斯谛方程中的饱和密度  $k$ 。受此启发,杨震等人[102]提出一种新型的混合遗传算法,将生物的进化过程建模成一个在表现型与基因型两个层面上同时进行的生命过程。“地理隔离”在生物进化的过程中起到了很重要的作用,因此“小生境”[103]技术在进化计算中也是一个研究热点。

日本学者木村资生提出的分子进化的中性学说[104],得到了学术界日益广泛的承认。文献[105]提出一种基于中性进化的遗传算法。

总之,对于进化论的进一步研究,必将导致进化计算的新发展。

## 1.5 本文研究的主要内容

论文的第二章分析了基于拉马克主义的进化学习策略 (Lamarckian Evolution Learning Strategy, 简称 LELS) 和基于达尔文主义的进化学习策略 (Darwinian Evolution Learning Strategy, 简称 DELS) 的异同, 利用马尔可夫链理论证明了此类算法的收敛性, 在理论上分析了 DELS 具有更强的局部极值逃逸能力; 最后针对 8 个常用的测试函数给出了仿真试验的结果。

第三章阐述作者对基于均匀分布变异算子的进化策略的研究结果, 探讨了均匀分布变异算子的作用, 并证明了此类算法的收敛性。

第四章在上述两章的基础上, 提出基于均匀变异算子的进化学习策略, 并对收敛性和局部逃逸能力进行了理论分析, 最后的仿真实验表明了该方法的有效性。

第五章首先给出最简约模糊划分的定义; 提出了模糊划分的完备—清晰性的概念, 并给出计算公式; 进一步讨论了模糊规则的完备性—紧凑性; 以及模糊规则的一致性。在此基础上, 应用进化策略, 对模糊系统进行寻优, 使其获得较好系统响应性能和较高可解释性。针对一个典型非线性对象和经典的 Box-Jenkins 煤气炉作为仿真实验的对象, 验证了该方法的有效性。

第六章针对 *K-means* 聚类算法易陷入局部极小以及  $K$  值选取的问题, 提出一类基于进化策略的聚类算法, 可以有效的搜索最优聚类中心和聚类个数  $K$ ; 还提出了确定  $K$  值范围的经验公式, 以减小搜索空间, 提高搜索效率, 并给出理论分析。相对遗传算法而言, 本方法编码简单, 种群较小。对 Fisher's iris 数据集的仿真实验表明, 该方法得到最优解的可能性比经典算法大得多。

第七章探讨了引入全局中间交叉算子的进化策略与 LMS 算法的异同。



LMS 算法是一种基于梯度法的单点搜索策略。而进化策略是一种受生物进化论启发而建立的优化算法，传统的观点一般认为它是一种全局随机多点搜索技术，与 LMS 有着完全不同的特点，是两种截然不同的优化技术。本章对此进行了深入的探讨，认为这两者之间的差异更多的是基于我们平时的一些“想当然”的观念上的，进化策略可以看作是一种“概率”意义上的梯度搜索技术，而 LMS 算法也可以看作一种多点搜索技术。

第八章是对全文做的一个总结。

## 第二章 进化学习策略收敛性和逃逸能力

进化策略(Evolution Strategies), 是进化计算(Evolution Computation)的主要分支之一[1], 曾经取得过很大的成功, 但是近年来没有获得根本性的发展, 其瓶颈在于: 它只是生物进化过程的一个极其简单的模拟。例如, 现有的进化策略没有考虑生物后天学习的情况。显然, 一个生物个体, 要在生存竞争中获胜, 除了要具有良好的遗传基因以外, 还应该进行较好的学习, 否则就不能充分发挥其优良基因的潜力。因此, 在进化策略中以适当的方式加入学习的环节, 就有可能在本质上提高其性能和效率。

### 2.1 问题的提出

进化策略(Evolution Strategy)与传统优化算法相结合, 是近年来的研究热点之一。相关的文献有, [91,92]将遗传算法与单纯形法相结合, 即后代个体利用单纯形法进行局部搜索, 以提高适应度; [93]明确提出将单纯形法作为学习算子; [94]提出是基于拉马克主义的, 即后天获得性遗传; [95]给出了混和遗传算法的收敛性证明; 此外[96~98]是将遗传算法与梯度法或其他局部优化算法相结合, 进行相关的研究。

但是目前的工作, 多是将一些新的混和进化算法(常属作者自己提出的新方法)与经典进化算法就某些数字仿真例子进行比较, 以此验证新算法的性能或效率更优, 而理论分析则非常有限, 尤其是有关收敛性、逃逸能力以及收敛速度的研究大大滞后于其在工程中的应用[112~114]。而且现有的学习方式大多基于拉马克主义(虽然许多文献并未明确标出), 基于达尔文主义的尚不多见, 两者的区别以及理论分析也未做足够的研究。针对进化策略的研究也较少见。此类方法的名称也较混杂, 早期的工作大多称之为“混和进化算法(Hybrid EA)”, 但是本文认为这不足以体现学习

的概念，因此建议改为“进化学习算法”；将进化策略与学习算法相结合，称之为“进化学习策略”。

基于上述考虑，本文分析了基于拉马克主义的进化学习策略（Lamarckian Evolution Learning Strategy，简称 LEELS）和基于达尔文主义的进化学习策略（Darwinian Evolution Learning Strategy，简称 DELS）的异同，证明了此类算法的收敛性，在理论上分析了 DELS 具有更强的局部极值逃逸能力；最后给出了仿真试验的结果。

## 2.2 预备知识

### 2.2.1 学习、基因型、表现型与适应度之间的关系

上述 4 者的关系如图 2.1 所示。一个生物个体，要在生存竞争中获胜，除了要具有良好的遗传基因以外，还应该进行较好的学习，以获得较好的表现型，否则就不能充分发挥其优良基因的潜力。个体的适应度，是由其表现型决定的，与基因型没有必然的关系。选择只能根据适应度针对表现型来进行，而无法发生在基因型这一层面上。学习是沟通基因型与表现型之间的桥梁。如果图 2.1 中虚线成立，即表现型在后天学习中也会影响基因型，

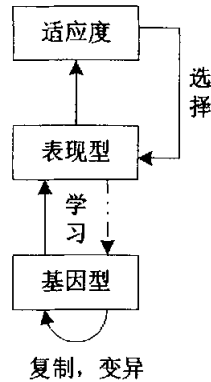


图 2.1 学习、基因型、表现型与适应度

就是拉马克主义；否则就是达尔文主义，这将在 2.2.2 节中详细论述。

早期的进化算法是生物进化过程的一个极其简单的模拟，并没有考虑生物后天学习的情况。因此，在进化算法中以适当的方式加入学习的环节，就有可能在本质上提高其性能和效率。

### 2.2.2 拉马克的用进废退和达尔文的自然选择

拉马克主义最重要的一个观点就是“后天获得性遗传”，即所谓“用进废退”说，最著名的例子就是“长颈鹿”：长颈鹿的祖先经常伸长了脖子去吃树枝高处的叶子，脖子受到了锻炼，变长了，而这一点可以遗传，因此其后代就要比父母的脖子长一些，经过一代又一代的进化，脖子就会变得越来越长。概括起来讲，就是生物在后天的学习中获得了表现型的改变（长颈鹿的脖子在锻炼中拉长），可以直接影响基因型的改变，进而导致了进化的产生。

达尔文对此有着截然不同的解释，他认为是自然选择在起作用。在长颈鹿繁殖的过程中，会有变异的情况产生，有的后代脖子会变长，有的会变短，当然还有的长度不变。这种变异是随机的，即“上帝是掷骰子的”。但是，只有脖子长的鹿，才有可能获得更多的食物，存活下来的可能性也就越大。这种变异和选择的过程不断发展，使得长脖子的性状在群体中扩散开来，就会进化出长颈鹿这样一个新的物种。脖子长度改变这样一种变异，对于每一只鹿来说只是一个偶然事件，但是在自然选择的作用下，在其种群的层面上来看，却呈现出向长颈鹿进化的方向。通过后天学习而获得的表现型的改变（长颈鹿的脖子在锻炼中拉长），虽然可以提高其存活的可能性，但是不会导致基因型的改变，更不会直接遗传给后代[85]。

自 20 世纪以来，基因论确立并与自然选择学说完美的结合在一起之后，拉马克主义也就被彻底的抛弃了。但是由于拉马克主义可以在一定程度上解释一定层次上的某些进化问题，所以也可以为我们改进进化策略提

供一些有益的启发。本文分别以拉马克主义和达尔文主义为依据，并不意味着作者认为拉马克主义在生物进化界也是正确无误的，只是以该理论为启发来讨论算法问题。

## 2.3 进化学习策略描述

### 2.3.1 基于达尔文主义的进化学习策略（DELS）算法流程

本算法中父代个体的数目用  $\mu$  表示，子代个体数目用  $\lambda$  表示，简记为  $(\mu, \lambda)$ -DELS 策略，描述如下：

1. 初始化：随机生成  $\mu$  个个体，构成第 1 代群体的父代。其中每个个体的不同是由个体的基因型的不同而决定的。
2. 变异：每个父代个体将自己的基因经过变异后产生  $\lambda/\mu$  个子代，总共生成  $\lambda$  个子代个体。此  $\lambda$  个子代个体与其各自的父代个体在基因上有所不同。
3. 学习： $\lambda$  个子代个体各自进行学习。这一步可以采用不同的学习算法，本文采用的是简单有效的单纯形法。学习后得到的个体，表现型被优化，适应度提高，但是基因型不会发生变化。
4. 选择：每个个体由于对环境的适应性不同而表现出不同的生命力或适应度，它由遗传基因和后天学习的情况共同决定。 $\lambda$  个个体中只有生命力较强的  $\mu$  个个体成为下一代的父辈个体。如果终止条件满足，则算法结束。否则回到步骤（2）。在进行新一轮的繁殖时，只有基因型经过变异遗传给下一代，表现型在此不起作用。

在 DELS 算法中，基因型与表现型没有必然的关系，即基因型好的个体，表现型不一定好；表现型好的个体，基因型不一定好。

### 2.3.2 基于拉马克主义的进化学习策略 (LELS)

**假设 2.3.1** 在 LELS 算法中, 表现型等同于基因型。

一般来说, 基因型与表现型之间的关系较为复杂, 但是由于 LELS 算法采用拉马克主义, 表现型的改善会直接导致基因型的改善, 因此两者之间的函数关系是单调递增的。本文为简单起见, 在 LELS 算法中, 将表现型等同于基因型, 以此来体现“后天获得性遗传”的观点。

LELS 在学习和变异操作上, 与 DELS 有所不同。学习后得到的个体, 表现型被优化, 适应度提高。由于采用拉马克主义, 以及假设 2.3.1, 所以基因型也同样被优化。 $\lambda$  个个体中表现型较好, 适应度较高的  $\mu$  个个体成为下一代的父代。需要注意的是, 在父代个体将自己的基因遗传给下一代时, 已经包括了后天学习中获得的基因型 (表现型) 的改变。

## 2.4 收敛性证明和局部极值逃逸能力分析

收敛性是进化计算的一个重要性质, 对于理论研究和实际应用都具有深远的影响。对于进化学习算法来说, 已有的工作很少涉及收敛性分析, 严重的制约了该算法的进一步发展。本文在 2.4.1 小节应用马尔可夫链理论, 证明了该类算法的收敛性, 并在后面的仿真实验中进行了验证。

局部极值逃逸能力 (escaping capacity) [115] 是指跳出局部极值点的足够大邻域的能力, 在理论研究和实际应用中, 尤其对于多模态优化问题, 具有重要的作用, 本文在 2.4.2 小节研究了进化学习算法的逃逸能力, 指出 DELS 算法的逃逸能力比 LELS 强。

### 2.4.1 进化学习策略收敛性证明

本节所指的进化学习策略 (ELS)，包括 LEELS 和 DEELS 算法。假设 ELS 在第  $t(t \geq 0)$  代的种群为  $X(t) = (x_{t,1}, x_{t,2}, \dots, x_{t,N}), x_{t,i} \in S$ ，种群规模为  $N < \infty$ ， $F_t = \max\{f(x_{t,k}) : k = 1, 2, \dots, N\}$  是第  $t$  代种群中最佳个体适应值。 $\{X(t)\}_{t \geq 0}$  构成一个离散时间的随机过程，其状态空间为

$E = S^N, t \in \overline{Z^+} = (0, 1, \dots, n, \dots)$ 。由于新种群  $X(t+1)$  仅以一种概率方式依赖于当前种群  $X(t)$ ，因此随机过程  $\{X(t)\}_{t \geq 0}$  构成一个离散时间的马尔可夫链。

**定义 2.4.1** 对于 ELS 求解数值优化问题，设随机变量  $T = \min\{t \in \overline{Z^+} : F_t = f^*\}$  表示首次击中全局极值点的时间，如果  $P\{T < \infty\} = 1$  且与初始种群无关，则称 ELS 以概率 1 在有限时间内访问到全局极值点。函数值  $f^* = f(x^*) < +\infty$  称为一个全局极大值，当且仅当

$$\forall x \in S \Rightarrow f(x) \leq f(x^*) \text{ 成立时， } x^* \in S \text{ 被称为一个全局极大值点。} [9]$$

**假设 2.4.1** 在每一进化代  $t$ ，对父代种群  $X(t)$ 、变异算子作用一次后得到的后代种群  $\hat{X}(t)$  以及任意的  $y \in S$ ，如果  $y \notin X(t)$ ，则存在  $p(t) > 0$ ，使得  $y \in \hat{X}(t)$  的概率  $\geq p(t)$ 。

这个假设保证了在进化的每一代中，每一个个体可以以严格大于 0 的概率通过一次变异成为搜索空间中其他任意一个个体。对于绝大多数实际情况，该假设都是成立的。

**定义 2.4.2** 考察连续参数优化问题 (continuous parameter optimization problem)，即给定搜索空间为非空  $n$  维欧氏空间  $R^n$  中的有界集合

$S = \prod_{i=1}^n [a_i, b_i]$ ，其中  $a_i < b_i, i = 1, 2, \dots, n$ 。定义  $S_1$  为全局极大点子空间， $S_2$  为非全局极大点子空间，当且仅当  $S_1 \cap S_2 = \Phi$ ， $S_1 \cup S_2 = S$ ，全局极大点  $x^* \in S_1$ ，且梯度  $\nabla f(x) \neq 0$ ， $\forall x \in S_1, x \neq x^*$ 。显然  $S_1$  为一凸集，与此对应的函数  $f(\omega), \omega \in S_1$  为凸函数。

**定理 1** 如果 ELS 满足假设 2.4.1，则其在有限次进化后以概率 1 访问

到全局极大点，亦即  $P\{T < \infty\} = 1$ ，且与初始分布无关。

证明：由于可以证明单纯形算法对于凸函数全局极值点搜索的收敛性，因此，根据定义 2.4.2，只要证明在任一进化代  $t(t < \infty)$ ，若父代个体  $X(t) \notin S_1$ ，而通过变异算子变异后得到的任一子代个体  $\hat{x}_i \in S_1$ ，则通过学习算子的作用，必可以得到最优个体  $\hat{x}_i = x^*$ ，即以概率 1 访问到全局极大点。其中  $\hat{x}_i$  为学习后的个体。

由假设 2.4.1 知，在任一进化代  $t$ ，通过变异算子的作用可以以一个下界为  $p(t)$  的概率将  $X(t)$  中任一个体变异为搜索空间中任意一个其他个体。因此，在变异过程中对任一个体  $x_i \notin S_1$ ，通过变异算子作用后变为  $S_1$  子空间中任意一点的概率都不小于  $p(t)$ ，则落入  $S_1$  子空间的概率不小于  $\int_{S_1} p(t)$ ，且与个体的具体实例无关。因此，在  $t$  代进化以后，还未有任一个体落入  $S_1$  子空间的概率  $P_{not}(t)$  满足：

$$P_{not}(t) = \prod_{i=1}^t (1 - \int_{S_1} p(i)) \quad (2.1)$$

$$\text{则} \quad P\{T < \infty\} = 1 - P_{not}(t) \quad (2.2)$$

在上式两端令  $t \rightarrow \infty$ ，可得：

$$P\{T < \infty\} = 1 - \prod_{i=1}^{\infty} (1 - \int_{S_1} p(i)) \quad (2.3)$$

$$\because 0 < \int_{S_1} p(i) < 1 \quad (2.4)$$

$$\therefore 0 < 1 - \int_{S_1} p(i) < 1 \quad (2.5)$$

$$\therefore \prod_{i=1}^{\infty} (1 - \int_{S_1} p(i)) = 0 \quad (2.6)$$

$$\text{即可得到：} \quad P\{T < \infty\} = 1 \quad (2.7)$$

由证明过程可以看出，结论的成立与初始群体是无关的。 [证毕]



## 2.4.2 局部极值逃逸能力分析

逃逸能力 (escaping capacity)[115]是指跳出局部极值点的足够大邻域的能力。文献[14]对高斯变异算子采用精英和非精英选择两种情况下的逃逸能力进行比较, 受此启发, 本文用类似的方法分析 DELS 和 LELS 的逃逸能力。

由于 ELS 的复杂性, 本文采用两种简化的方式来产生新的搜索点, 以分别代替 LELS 和 DELS 来进行讨论。假设当前点  $x_0$  是局部最优点, 方法一是重复方式, 即产生  $k$  个新的搜索点  $\{x_0 + N_i(0,1), i=1,2,\dots,k\}$ , 其中  $N_i(0,1)$  重复独立产生。由于 LELS 的表现型的改进直接引起基因型的改进, 所以陷入局部极值点的个体, 在变异后产生的下一代, 如果仍然在局部极值点邻域内, 必将经过学习操作又收敛于该极值点。因此与方法一相类似。方法二产生新的搜索点为:  $\{x_i + N_{i+1}(0,1), i=0,1,\dots,k-1\}$ , 其中  $N_{i+1}(0,1)$  也是重复独立产生的, 即对于每一个新产生的点, 都接受它并用它产生新的搜索点。由于 DELS 的表现型的改进不能引起基因型的改变, 所以基因型不受学习的影响而只经过变异操作遗传给下一代, 因此与方法二相类似。

对于  $x_0$  的某个邻域  $S(R) = \{x: |x - x_0| < \varepsilon\}$ ,  $\varepsilon$  为事先给定的一个小数, 考虑在产生  $k$  个个体中, 至少有一个个体跳出该邻域的概率为  $P_{esp}$ 。

对于方法一, 定义:

$$x_{\max} = \max_i \{|x_0 - x_i|\} \quad (2.8)$$

显然有:

$$P_{esp1} = 1 - [P(x_{\max} < R)]^k \quad (2.9)$$

$$= 1 - [P(|N(0,1)| < R)]^k \quad (2.10)$$

$$= 1 - \left[ \int_{-R}^R \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right) dt \right]^k \quad (2.11)$$

$$= 1 - (\operatorname{erf}(\frac{\sqrt{2}}{2} R))^k \quad (2.12)$$

对于方法二，有：

$$P_{esp2} = 1 - \prod_{i=1}^k P(|x_i - x_0| < R) \quad (2.13)$$

$$= 1 - \left[ \int_R^{\infty} \frac{1}{\sqrt{2\pi k}} \exp(-\frac{1}{2k} t^2) dt \right] \quad (2.14)$$

$$= 1 - \operatorname{erf}(\frac{R}{\sqrt{2k}}) \quad (2.15)$$

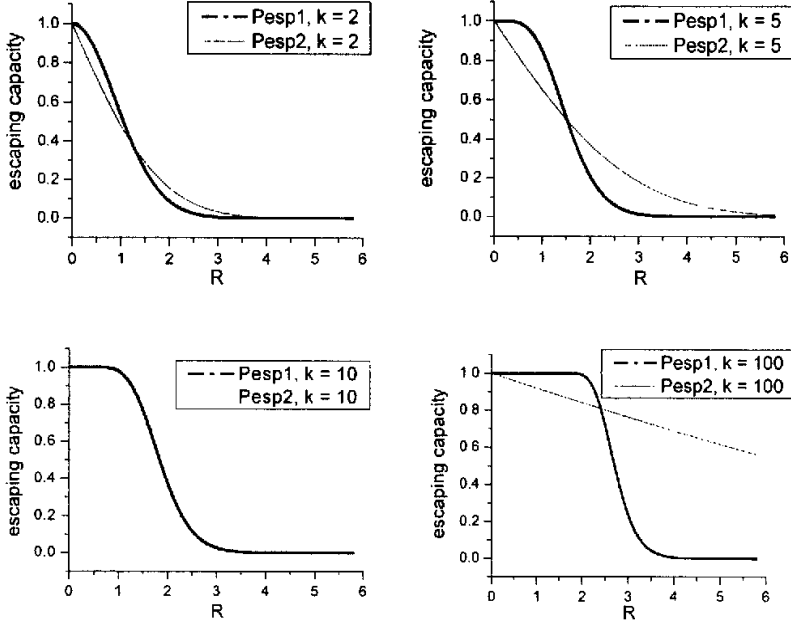


图 2.2 LELS 和 DELS 逃逸能力比较

根据上式对  $P_{esp1}$  和  $P_{esp2}$  作图，分别取  $k=2, 5, 10, 100$ ，如图 2.2 所示。

分析图 2.2 可以看出, 局部极值点的邻域越大, 跳出局部极值点的可能性就越小; 进化代数越多, 跳出局部极值点的可能性就越大, 两种方法都符合该规律; 当  $R$  较小时, 两种方法跳出局部极值点的可能性都较大, 但是拉马克主义更大一些; 而当  $R$  较大时, 拉马克主义迅速衰减, 而达尔文主义仍保持较高的可能性, 尤其在  $k$  较大的时候。

## 2.5 仿真实例

表 2.2 LELS/DELS 算法和 ES 算法的性能比较

测试函数	优化算法	进化代数	运行时间 (秒)	解的平均值	平均每代运行时间 (秒)
$F_1$	(2,14)-LELS	1	0.0141	2E-05	0.0141
	(2,14)-DELS	1	0.0109	4E-05	0.0109
	(2,14)-ES	20.5	0.0164	0.00042	0.0008
$F_2$	(2,14)-LELS	7.9	0.0172	0.00060	0.00218
	(2,14)-DELS	25	0.0408	0.00040	0.00163
	(2,14)-ES	776.2	0.1812	0.00046	0.00023
$F_3$	(2,14)-LELS	43.2	0.0469	186.7306	0.00109
	(2,14)-DELS	31.5	0.0440	186.7304	0.00140
	(2,14)-ES	5523.4	2.3749	186.7301	0.00040
$F_4$	(2,14)-LELS	20.3	0.0265	1.03127	0.00131
	(2,14)-DELS	18.8	0.0124	1.03122	0.00066
	(2,14)-ES	114	0.0314	1.03128	0.00028
$F_5$	(2,14)-LELS	352.2	0.1610	837.966	0.00046
	(2,14)-DELS	136.6	0.0907	837.966	0.00066
	(2,14)-ES	473.4	0.1092	837.965	0.00023
$F_6$	(2,14)-LELS	614.1	0.3766	0.00088	0.00061
	(2,14)-DELS	339.9	0.1655	0.00087	0.00049
	(2,14)-ES	795.8	0.1670	0.00090	0.00021
$F_7$	(2,14)-LELS	19.3	0.0234	0.99963	0.00121
	(2,14)-DELS	13.2	0.0171	0.99963	0.00130
	(2,14)-ES	11089.3	4.6188	0.99944	0.00042
$F_8$	(2,14)-LELS	36.1	0.0267	0.0004	0.00074
	(2,14)-DELS	29.5	0.0125	0.0005	0.00042
	(2,14)-ES	160.3	0.0390	0.0005	0.00024

本节采用 8 个标准测试函数来全面测试 LEELS/DELS 的性能, 这 8 个标准测试函数如附录 1 所示[115][84]。其中进化算法的父代个数为 2, 子代个数为 14, 父代均不参与下一代的竞争。LEELS/DELS 算法的学习操作应用单纯形法实现, 设置其最大迭代次数为 100。每一次变异和学习之后, 产生的非法后代直接抛弃。对每个函数独立运行 50 次, 并记录(1) 平均进化代数, (2) 运行时间(秒), (3)解的平均值, (4)平均每代运行时间(秒)。测试平台: AMD XP 2200+, DDR256M/ WinXP/ Matlab6.5。由于循环是 Matlab 运行速度的瓶颈[116], 而学习子程序(单纯形法)会用到大量的循环, 所以将 LEELS/DELS 算法与 ES 算法做速度方面的比较时, 会有失偏颇。因此本文将学习子程序用 C 语言编制 MEX 文件实现, 提高了学习速度; 而 LEELS/DELS 算法的其他部分采用 Matlab 实现, 与 ES 算法完全一样。又由于 Matlab 一般只是用于仿真, 系统实施时一般采用 C/C++等其他语言实现, 没有因循环而导致的速度瓶颈, 因此本文的做法是较公正的。

仿真结果如表 2.2 所示。分析该表可以看出, LEELS/DELS 所需的进化代数较少, 约为 ES 的  $1/840 \sim 1/2$ ; 搜索精度也略好于 ES。而为此付出的代价只是每代多做若干次单纯形迭代, 以及平均每代花费的时间是经典算法的约 2~3 倍。但是由于进化代数远远小于经典算法, 所以总体运行时间仍然比经典算法少。其中最突出的是  $F_7$  函数, DELS 的进化代数仅是 ES 算法的约  $1/840$ , 总体运行时间约为  $1/270$ ; LEELS 算法相应的结果分别是  $1/574$  和  $1/197$ 。

大多数情况下, DELS 比 LEELS 所需要的进化代数少, 总体运行时间更短。例如对于  $F_6$  函数, DELS 的进化代数是 LEELS 算法的约  $1/2.58$ , 总体运行时间约为  $1/1.78$ ; 相应的  $F_6$  结果是  $1/1.81$  和  $1/2.28$ 。只有对于  $F_1$  和  $F_2$  函数例外。其中对于  $F_1$ , DELS/LEELS 都只需要一代就可以搜索到最优值, 并且 LEELS 搜索精度更高一些, 这是因为  $F_1$  函数只有一个极值点; 对于  $F_2$  函数 DELS 的进化代数和总体运行时间是 LEELS 的 3 倍左右, 这主要是因为  $F_2$  函数的局部极值较少, 更适合 LEELS 发挥其后天获得性遗传的优势。

这说明局部极值较少的时候, LELS 性能更好一些; 而局部极值较多时, DELS 更好一些。这也印证了本文提出的 DELS 算法局部极值逃逸能力强于 LELS 的观点。

众所周知, 实时性是进化算法的一个瓶颈, DELS/LELS 算法在此做了一个较好的突破。

## 2.6 本章小结

自然环境是根据适应度针对表现型进行选择, 表现型的好坏, 一部分取决于基因是否优良, 另一部分则取决于个体在后天学习中是否充分发挥了优良基因的潜力。因此进化学习算法是当今的研究热点之一。

本文针对已有工作大都缺乏收敛性证明和逃逸能力分析的缺点, 讨论基于拉马克主义的进化学习策略(简称 LELS)和基于达尔文主义的进化学习策略(简称 DELS)在学习和变异操作上的异同, 证明了此类算法的收敛性, 并且进行了局部极值逃逸能力的研究。仿真试验表明, 此类算法具有较好的全局优化能力和较高的实时性。在测试函数局部极值较少的情况下, LELS 算法的收敛速度较快, 而测试函数局部极值较多时, DELS 算法的收敛速度较快, 这也与本文提出的 DELS 具有更强的局部极值逃逸能力的观点相吻合。该类算法的收敛速度也要比经典进化策略快, 为提高进化算法的实时性, 扩大应用范围, 提供了一个较好的思路。

虽然本文证明了进化学习策略的收敛性, 在仿真中也得到了该算法收敛速度快的结果, 但是无法给出收敛速度的证明, 这也是本文今后的努力方向。

### 第三章 基于均匀分布变异算子的进化策略

#### 3.1 问题的提出

经典进化策略(Evolution Strategy)采用的变异算子通常是正态分布算子[58]。作为 ES 中变异的一个扰动,在缺少有关问题的先验知识的情况下,可以认为是由许多微小因素共同作用的结果,因此可以认为它服从正态分布。受模拟退火算法的启发,文献[117, 118]将满足柯西分布的柯西变异算子引入到 ES 中。为了发挥正态分布算子局部搜索能力强,收敛速度快的优点,又能利用柯西算子产生大的变异的特点,文献[119]提出了一个折衷的方案,将柯西变异算子和正态变异算子结合起来,定义一个新的变异算子,称之为“平均变异算子”。其结果是一维的情况,平均变异算子的局部搜索能力比单纯的柯西变异算子高,在产生很小变异时也超过了正态变异算子。但是在产生大的变异方面,平均变异算子的能力并不高,甚至不如正态变异算子。

均匀分布形式简单,便于分析和处理。本章受上述文献的启发,将满足均匀分布的均匀变异算子引入到 ES 中,并对收敛性和局部逃逸能力进行了理论分析。

#### 3.2 基于均匀变异算子的进化策略描述

##### 3.2.1 算法流程

本算法中父代个体的数目用  $\mu$  表示,子代个体数目用  $\lambda$  表示,简记为

$(\mu, \lambda)$ -UES, 描述如下:

1. 初始化: 随机生成  $\mu$  个个体, 构成第 1 代群体的父代。其中每个个体的不同是由个体的基因型的不同而决定的。
2. 变异: 每个父代个体将自己的基因经过变异后产生  $\lambda/\mu$  个子代, 总共生成  $\lambda$  个子代个体。此  $\lambda$  个子代个体与其各自的父代个体在基因上有所不同。变异算子服从均匀分布。
3. 选择: 每个个体由于对环境的适应性不同而表现出不同的生命力或适应度。 $\lambda$  个个体中只有生命力较强的  $\mu$  个个体成为下一代的父辈个体。如果终止条件满足, 则算法结束。否则回到步骤 (2)。

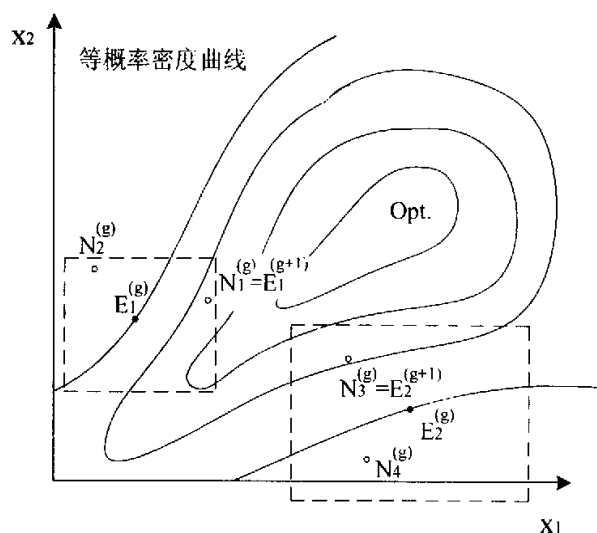


图 3.1 (2,4)-UES 的一次变异过程

### 3.2.2 变异过程

以下用  $U(1)$  表示一个在区间  $[-1, 1]$  上服从一维均匀分布的随机变量的实现,  $U_i(1)$  对于下标  $i$  的每个随机变量要重新抽样。一个在区间  $[-a, a]$  上服从一维均匀分布的随机变量的实现就可以由  $a \times U(1)$  给出。

则 UES 变异公式如下:

$$a'_i = a_i \cdot \exp(\tau' U(1) + \tau \cdot U_i(1)), i = 1, 2, \dots, Q \quad (3.1)$$

$$x'_i = x_i + a'_i * U_i(1), i = 1, 2, \dots, Q \quad (3.2)$$

则  $n$  维总体  $X = [x_1, x_2, \dots, x_n]$  的联合分布密度为:

$$P(X) = \begin{cases} \frac{1}{\prod_{i=1}^n (2a_i)}, & -a_i \leq x_i \leq a_i, i = 1, 2, \dots, n \\ 0, & \text{else} \end{cases} \quad (3.3)$$

其中  $\prod_{i=1}^n (2a_i)$  表示的是一个超长方体的体积。在此域内, 概率处处相等。

由此, 我们知道了在方程(3.3)所限定的条件下, 变异过程的几何意义如下: 对于给定的一个解  $x$  所做的变异就是在该解的邻域内选取的另一个新解; 而在该邻域内所有解被选择的机会都相等。当  $n=2$  时, 一次变异过程的几何意义如图 3.1 所示。

### 3.3 收敛性证明

假设 ELS 在第  $t(t \geq 0)$  代的种群为  $X(t) = (x_{t,1}, x_{t,2}, \dots, x_{t,N}), x_{t,k} \in S$ , 种群规模为  $N < \infty$ ,  $F_t = \max\{f(x_{t,k}) : k = 1, 2, \dots, N\}$  是第  $t$  代种群中最佳个体适应值。 $\{X(t)\}_{t \geq 0}$  构成一个离散时间的随机过程, 其状态空间为  $E = S^N$ ,  $t \in \overline{\mathbb{Z}^+} = (0, 1, \dots, n, \dots)$ 。由于新种群  $X(t+1)$  仅以一种概率方式依赖于当前种



群  $X(t)$ ，因此随机过程  $\{X(t)\}_{t \geq 0}$  构成一个离散时间的马尔可夫链[120]。

**假设 3.3.1** 在每一进化代  $t$ ，对父代种群  $X(t)$ 、变异算子作用一次后得到的后代种群  $\hat{X}(t)$  以及任意的  $y \in S$ ，如果  $y \notin X(t)$ ，则存在  $p(t) > 0$ ，使得  $y \in \hat{X}(t)$  的概率  $\geq p(t)$ 。

这个假设保证了再进化的每一代中，每一个个体可以以严格大于 0 的概率通过一次变异成为搜索空间中其他任意一个个体。对于绝大多数实际情况，该假设都是成立的。

**定理 3.1** 如果 UES 满足假设 3.3.1，则其在有限次进化后以概率 1 访问到全局极大点，亦即  $P\{T < \infty\} = 1$ ，且与初始分布无关。

**证明：**由假设 3.3.1 知，在进化代  $t$ ， $X(t)$  中任意一个个体通过变异算子的作用变异为搜索空间中任意一个其他个体的概率不小于  $p(t)$ ，且与个体的具体实例无关。因此，在  $t$  代进化以后，还未有任一个体访问到整体极值点的概率  $P_{not}(t)$  满足：

$$P_{not}(t) = \prod_{i=1}^t (1 - p(i)) \quad (3.4)$$

$$\text{则} \quad P\{T < \infty\} = 1 - P_{not}(t) \quad (3.5)$$

在上式两端令  $t \rightarrow \infty$ ，可得：

$$P\{T < \infty\} = 1 - \prod_{i=1}^{\infty} (1 - p(i)) \quad (3.6)$$

$$\because 0 < p(i) \leq 1 \quad (3.7)$$

$$\therefore 0 \leq 1 - p(i) < 1 \quad (3.8)$$

$$\therefore \prod_{i=1}^{\infty} (1 - p(i)) = 0 \quad (3.9)$$

$$\text{即可得到:} \quad P\{T < \infty\} = 1 \quad (3.10)$$

由证明过程可以看出，结论的成立与初始群体是无关的。 [证毕]

### 3.4 局部极值逃逸能力分析

逃逸能力(escaping capacity)[115]是指跳出局部极值点的足够大邻域的能力。文献[115]对高斯变异算子采用精英和非精英选择两种情况下的逃逸能力进行比较,受此启发,本章用类似的方法分析 UES 和 ES 的逃逸能力。

本章只考虑非精英选择策略。由于进化策略的复杂性,本章采用简化的方式来产生新的搜索点,以分别代替 ES 和 UES 来进行讨论。假设当前点  $x_0$  是局部最优点,ES 产生  $k$  个新的搜索点为:  $\{x_i + N_{i+1}(0,1), i = 0,1,\dots,k-1\}$ , 其中  $N_{i+1}(0,1)$  是重复独立产生的,即对于每一个新产生的点,都接受它并用它产生新的搜索点;UES 产生新的搜索点为:  $\{x_i + U_{i+1}(1), i = 0,1,\dots,k-1\}$ , 其中  $U_{i+1}(1)$  也是重复独立产生的。

对于  $x_0$  的某个邻域  $S(R) = \{x: |x - x_0| < \varepsilon\}$ ,  $\varepsilon$  为事先给定的一个小数,考虑在产生  $k$  个个体中,至少有一个个体跳出该邻域的概率为  $P_{esp}$ 。为简单起见,本章只考虑考虑一维的情况。

设 ES 的逃逸能力为  $P_{esp1}$ , 显然有:

$$P_{esp1} = 1 - \prod_{i=1}^k P(|x_i - x_0| < R) \quad (3.11)$$

$$= 1 - \left[ \int_R^{\infty} \frac{1}{\sqrt{2\pi k}} \exp\left(-\frac{1}{2k} t^2\right) dt \right] \quad (3.12)$$

$$= 1 - \operatorname{erf}\left(\frac{R}{\sqrt{2k}}\right) \quad (3.13)$$

设 UES 的逃逸能力为  $P_{esp2}$ , 则有:

$$P_{esp2} = 1 - P(|x_k - x_0| < R) \quad (3.14)$$

$$= 1 - P(U(k) < R) \quad (3.15)$$

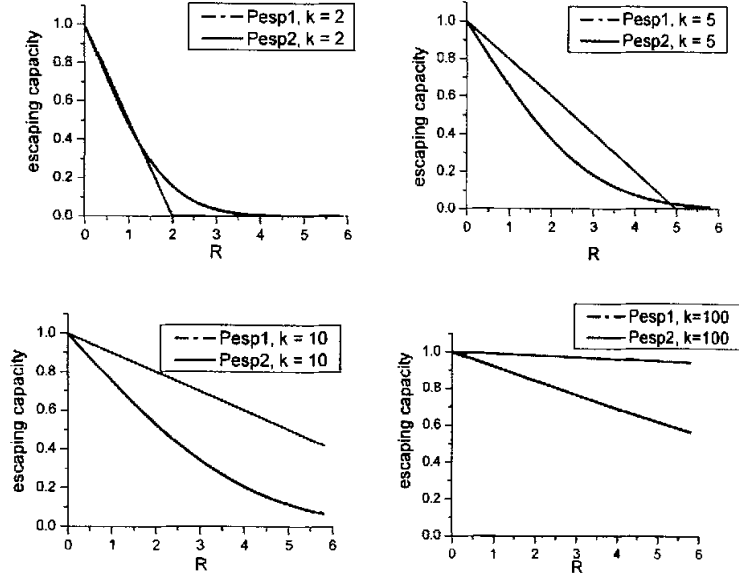


图 3.2 UES 和 ES 逃逸能力比较

$$= \begin{cases} 1 - \frac{R}{k}, & \text{if } (k > R) \\ 0, & \text{else} \end{cases} \quad (3.16)$$

根据上式对  $P_{esp1}$  和  $P_{esp2}$  作图，分别取  $k=2, 5, 10, 100$ ，如图 3.2 所示。

分析图 3.2 可以看出，局部极值点的邻域越大，跳出局部极值点的可能性就越小；进化代数越多，跳出局部极值点的可能性就越大，两种方法都符合该规律；但是 UES 的可能性更大一些；而且当  $R$  较大时，ES 迅速衰减，而 UES 仍保持较高的可能性，尤其在  $k$  较大的时候。但是当  $k < R$  时，UES 的局部逃逸能力衰减为 0，也就是说，局部极值点的邻域大，而进化代数却不够多时，UES 无法逃出局部极值点。

### 3.5 仿真实例

本节采用 8 个标准测试函数来测试 UES/ES 的性能, 即附录中的  $F_1$ 、 $F_2$ 、 $F_3$ 、 $F_4$ 、 $F_5$ 、 $F_6$ 、 $F_8$  和  $F_{10}$ 。对每个函数独立运行 50 次, 并记录(1) 平均进化代数, (2) 运行时间(秒), (3)解的平均值, (4)平均每代运行时间(秒)。

测试平台: AMD XP 2200+, DDR256M/ WinXP / Matlab6.5。

仿真结果如表 3.1 所示。分析该表可以看出, 大多数情况下, UES 比 ES 所需要的进化代数少, 总体运行时间更短。例如对于  $F_1$  函数, UES 的进化代数是 ES 算法的约 1/1.23, 总体运行时间约为 1/3.03; 相应的  $F_2$  结果是 1/1.56 和 1/3.51。只有对于  $F_6$  和  $F_{10}$  函数例外, ES 的性能要好一些。

表 3.1 UES 算法和 ES 算法的性能比较

测试函数	优化算法	进化代数	解的平均值	运行时间 (秒)	平均每代运行 时间(秒)
$F_1$	(2,14)-ES	20.5	-0.00042	0.00940	0.00046
	(2,14)-UES	16.6	-0.00036	0.00310	0.00019
$F_2$	(2,14)-ES	776.2	-0.00046	0.18120	0.00023
	(2,14)-UES	303.7	-0.00057	0.05160	0.00017
$F_3$	(2,14)-ES	5523.4	186.73010	2.37490	0.00040
	(2,14)-UES	4394.5	186.73035	1.42980	0.00033
$F_4$	(2,14)-ES	114	1.03128	0.03140	0.00028
	(2,14)-UES	75.1	1.03134	0.02050	0.00027
$F_5$	(2,14)-ES	473.4	837.96530	0.10920	0.00023
	(2,14)-UES	283.3	837.96523	0.04830	0.00017
$F_6$	(2,14)-ES	795.8	0.00090	0.16700	0.00021
	(2,14)-UES	876.7	-0.00085	0.20010	0.00023
$F_8$	(2,14)-ES	160.3	0.00050	0.03900	0.00024
	(2,14)-UES	153.9	-0.00045	0.03450	0.00022
$F_{10}$	(2,14)-ES	504.8	38.84974	0.09520	0.00019
	(2,14)-UES	340.6	38.85029	0.10140	0.00030

### 3.6 本章小结

均匀分布形式简单，便于分析和处理。本章将满足均匀分布的均匀变异算子引入到 ES 中，提出了 UES 算法，描述了该算法的流程，分析了变异的过程，并对收敛性和局部逃逸能力进行了理论分析。仿真实验表明了该算法的有效性。

## 第四章 基于均匀分布变异算子的进化学习策略

### 4.1 问题的提出

本文第二章分析了基于拉马克主义的进化学习策略（Lamarckian Evolution Learning Strategy, 简称 LELS）和基于达尔文主义的进化学习策略（Darwinian Evolution Learning Strategy, 简称 DELS）的异同，证明了此类算法的收敛性，在理论上分析了 DELS 具有更强的局部极值逃逸能力；本文第三章将服从均匀分布的均匀变异算子引入到 ES 中，并对收敛性和局部逃逸能力进行了理论分析。

在上述两章的基础上，本章提出基于均匀变异算子的进化学习策略，并对收敛性和局部逃逸能力进行了理论分析，最后的仿真实验表明了该方法的有效性。

### 4.2 均匀进化学习策略描述

与本文第二章相似，均匀进化学习策略也分为基于拉马克主义的均匀进化学习策略（简称 LUELS）和基于达尔文主义的均匀进化学习策略（简称 DUELS）的异同，下面分别叙述。

#### 4.2.1 基于达尔文主义的均匀进化学习策略（DUELS）

本算法中父代个体的数目用  $\mu$  表示，子代个体数目用  $\lambda$  表示，简记为

$(\mu, \lambda)$ -DUELS 策略, 描述如下:

1. **初始化:** 随机生成  $\mu$  个个体, 构成第 1 代群体的父代。其中每个个体的不同是由个体的基因型的不同而决定的。
2. **变异:** 每个父代个体将自己的基因经过变异后产生  $\lambda/\mu$  个子代, 总共生成  $\lambda$  个子代个体。此  $\lambda$  个子代个体与其各自的父代个体在基因上有所不同。
3. **学习:**  $\lambda$  个子代个体各自进行学习。这一步可以采用不同的学习算法, 本章采用的是简单有效的单纯形法。学习后得到的个体, 表现型被优化, 适应度提高, 但是基因型不会发生变化。
4. **选择:** 每个个体由于对环境的适应性不同而表现出不同的生命力或适应度, 它由遗传基因和后天学习的情况共同决定。 $\lambda$  个个体中只有生命力较强的  $\mu$  个个体成为下一代的父辈个体。如果终止条件满足, 则算法结束。否则回到步骤(2)。在进行新一轮的繁殖时, 只有基因型经过变异遗传给下一代, 表现型在此不起作用。

在 DUELS 算法中, 基因型与表现型没有必然的关系, 即基因型好的个体, 表现型不一定好; 表现型好的个体, 基因型不一定好。

#### 4.2.2 基于拉马克主义的进化学习策略 (LUELS)

**假设 4.2.1** 在 LUELS 算法中, 表现型等同于基因型。

一般来说, 基因型与表现型之间的关系较为复杂, 但是由于 LUELS 算法采用拉马克主义, 表现型的改善会直接导致基因型的改善, 因此两者之间的函数关系是单调递增的。本章为简单起见, 在 LUELS 算法中, 将表现型等同于基因型, 以此来体现“后天获得性遗传”的观点。

LUELS 在学习和变异操作上, 与 DUELS 有所不同。学习后得到的个体, 表现型被优化, 适应度提高。由于采用拉马克主义, 以及假设 4.2.1, 所以基因型也同样被优化。 $\lambda$  个个体中表现型较好, 适应度较高的  $\mu$  个

体成为下一代的父代。需要注意的是，在父代个体将自己的基因遗传给下一代时，已经包括了后天学习中获得的基因型（表现型）的改变。

#### 4.2.3 变异和学习的几何意义

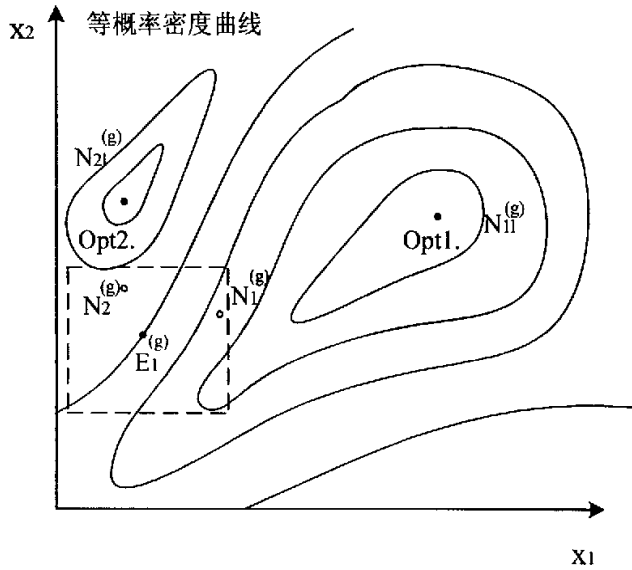


图 4.1 (1, 2) 进化学习策略相邻两次迭代示意图

变异和学习过程的几何意义如下：对于给定的一个解  $x$  所做的变异就是在该解的一个超正方体域内选取的另一个新解；而在该邻域内所有解被选择的机会都相等。每个新解按照一定的学习算法进行学习，搜索到该邻域附近的极值点。当  $n=1$  时，一次变异过程的几何意义如图 4.1 所示。

其中  $E_1^{(g)}$  是第  $g$  代的父代个体，经过变异操作，产生两个子代个体  $N_1^{(g)}$  和  $N_2^{(g)}$ ，经过学习操作后，变为  $N_{1l}^{(g)}$  和  $N_{2l}^{(g)}$ ，即分别搜索到两个局部极值点 opt.1 和 opt.2。然后，在  $N_{1l}^{(g)}$  和  $N_{2l}^{(g)}$  中选择适应值高的一个。如果是 LUELS 算法， $N_{1l}^{(g)}$  和  $N_{2l}^{(g)}$  中适应值高的一个直接作为  $g+1$  代的父代；如果是 DUELS 算法，则由其对应的  $N_1^{(g)}$  和  $N_2^{(g)}$  作为第  $g+1$  代的父代。



## 4.3 收敛性证明和局部极值逃逸能力分析

### 4.3.1 均匀进化学习策略收敛性证明

均匀进化学习策略收敛性证明的证明，与第二章第四节类似，本章不再赘述。

### 4.3.2 局部极值逃逸能力分析

逃逸能力(escaping capacity)[115]是指跳出局部极值点的足够大邻域的能力。第三章对高斯变异算子和均匀变异算子的逃逸能力进行比较，本章用类似的方法分析 DUELS 和 LUELS 的逃逸能力。

假设当前点  $x_0$  是局部最优点，方法一是重复方式，即产生  $k$  个新的搜索点  $\{x_0 + U_i(l), i = 1, 2, \dots, k\}$ ，其中  $U_i(l)$  重复独立产生。由于 LUELS 的表现型的改进直接引起基因型的改进，所以陷入局部极值点的个体，在变异后产生的下一代，如果仍然在局部极值点邻域内，必将经过学习操作又收敛于该极值点。因此与方法一相类似。方法二产生新的搜索点为：

$\{x_i + U_{i+1}(l), i = 0, 1, \dots, k-1\}$ ，其中  $U_{i+1}(l)$  也是重复独立产生的，即对于每一个新产生的点，都接受它并用它产生新的搜索点。由于 DUELS 的表现型的改进不能引起基因型的改变，所以基因型不受学习的影响而只经过变异操作遗传给下一代，因此与方法二相类似。

对于  $x_0$  的某个邻域  $S(R) = \{x : |x - x_0| < \varepsilon\}$ ， $\varepsilon$  为事先给定的一个小数，考虑在产生  $k$  个个体中，至少有一个个体跳出该邻域的概率为  $P_{esp}$ 。

设 DUELS 的逃逸能力为  $P_{esp1}$ ，显然有：

$$P_{esp1} = 1 - \prod_{i=1}^k P(|x_i - x_0| < R) \quad (4.1)$$

$$= 1 - [P(|x_i - x_0| < R)]^k \quad (4.2)$$

$$= 1 - [P(|U(1)| < R)]^k \quad (4.3)$$

$$= \begin{cases} 1 - R^k, & \text{if } (R < 1) \\ 0, & \text{else} \end{cases} \quad (4.4)$$

设 LUELS 的逃逸能力为  $P_{esp2}$ ，则有：

$$P_{esp2} = 1 - P(|x_k - x_0| < R) \quad (4.5)$$

$$= 1 - P(U(k) < R) \quad (4.6)$$

$$= \begin{cases} 1 - \frac{R}{k}, & \text{if } (k > R) \\ 0, & \text{else} \end{cases} \quad (4.7)$$

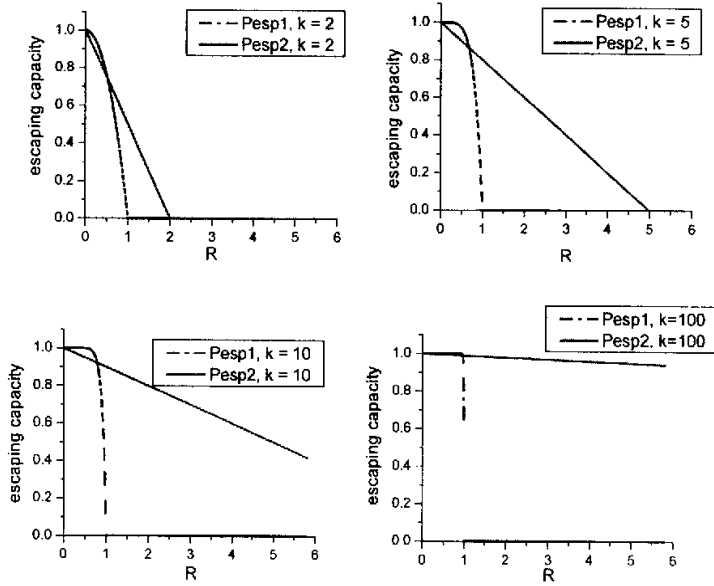


图 4.2 LUELS 和 DUELS 逃逸能力比较

根据上式对  $P_{esp1}$  和  $P_{esp2}$  作图，分别取  $k=2, 5, 10, 100$ ，如图 4.2 所示。

分析图 4.2 可以看出，局部极值点的邻域越大，跳出局部极值点的可能性就越小；进化代数越多，跳出局部极值点的可能性就越大，两种方法都符合该规律；当  $R$  接近 1 时，拉马克主义迅速衰减，当  $R=1$  时，拉马克主义衰减为 0，而达尔文主义仍保持较高的可能性，尤其在  $k$  较大的时候。

#### 4.4 仿真实例

本节采用 8 个标准测试函数来全面测试 LUELS/DUELS 的性能，这 8 个标准测试函数如附录所示。其中 LUELS/DUELS 算法的学习操作应用单纯形法实现，设置其最大迭代次数为 100。对每个函数独立运行 50 次，并记录(1) 平均进化代数，(2) 运行时间（秒），(3)解的平均值，(4)平均每代运行时间(秒)。测试平台：AMD XP 2200+，DDR256M/ WinXP / Matlab6.5。由于循环是 Matlab 运行速度的瓶颈[16]，而学习子程序（单纯形法）会用到大量的循环，所以将 LUELS/DUELS 算法与 ES 算法做速度方面的比较时，会有失偏颇。因此本章将学习子程序用 C 语言编制 MEX 文件实现，提高了学习速度；而 LUELS/DUELS 算法的其他部分采用 Matlab 实现，与 ES 算法完全一样。又由于 Matlab 一般只是用于仿真，系统实施时一般采用 C/C++等其他语言实现，没有因循环而导致的速度瓶颈，因此本文的做法是较公正的。

仿真结果如表 4.1 所示。大多数情况下，DUELS 比 LUELS 所需要的进化代数少，总体运行时间更短。只有对于  $F_1$  函数例外，DUELS/LUELS 都只需要一代就可以搜索到最优值，并且 LUELS 搜索精度更高一些，这是因为  $F_1$  函数只有一个极值点。这说明局部极值较少的时候，LUELS 性能

表 4.1 LUELS 算法和 DUELS 算法的性能比较

测试函数	优化算法	进化代数	运行时间 (秒)	解的平均值	平均每代运行时间 (秒)
F <sub>1</sub>	(2,14)-LUELS	1	0.0137	2E-05	0.0137
	(2,14)-DUELS	1	0.0119	4E-05	0.0119
F <sub>2</sub>	(2,14)-LUELS	20.7	0.0344	-0.00048	0.00166
	(2,14)-DUELS	10.4	0.0124	-0.00041	0.00119
F <sub>3</sub>	(2,14)-LUELS	14.5	0.026	186.731	0.00183
	(2,14)-DUELS	14.1	0.0234	186.731	0.00166
F <sub>4</sub>	(2,14)-LUELS	237.2	0.0968	1.03116	0.00041
	(2,14)-DUELS	13.2	0.017	1.03113	0.00129
F <sub>5</sub>	(2,14)-LUELS	407.5	0.172	837.966	0.00042
	(2,14)-DUELS	148.3	0.0901	837.966	0.00061
F <sub>6</sub>	(2,14)-LUELS	597.4	0.3705	0.00087	0.00062
	(2,14)-DUELS	317.6	0.1584	0.00088	0.00050
F <sub>7</sub>	(2,14)-LUELS	19.7	0.0264	0.99963	0.00134
	(2,14)-DUELS	12.2	0.0151	0.99963	0.00124
F <sub>8</sub>	(2,14)-LUELS	38.1	0.0286	0.0005	0.00075
	(2,14)-DUELS	27.9	0.0126	0.0004	0.00045

更好一些；而局部极值较多时，DUELS 更好一些。这也印证了本章提出的 DUELS 算法局部极值逃逸能力强于 LUELS 的观点。

4.5 本章小结

在前两章的基础上，本章提出基于均匀变异算子的进化学习策略，并对收敛性和局部逃逸能力进行了理论分析，最后的仿真实验表明了该方法的有效性。

## 第五章 基于进化策略生成可解释性模糊系统

模糊系统的可解释性明显优于其他人工智能方法，却长期没有引起人们足够的注意。本章对模糊系统的可解释性作了深入的分析，定义了最简约模糊划分、模糊划分的完备-清晰性、模糊规则的完备性、紧凑性和一致性，并将其加入到进化策略的适值函数中，用于优化模糊系统。仿真试验表明，即使在先验知识较少的情况下，该方法依然可以设计出具有较好系统响应性能和较高可解释性的模糊系统。

### 5.1 引言

模糊系统理论与其他人工智能方法（如神经网络、进化计算）相比较，一个突出的优点是，在可解释性方面，具有显著的优越性，主要体现在，其结构和参数可以有明显的物理意义，这为人们理解蕴涵其中的知识提供了便利[121]；缺点是不具备学习的能力。以自适应模糊系统为代表的一大类方法[122]，将神经网络、进化计算等各种学习算法与模糊系统相结合，可以从数据中提取模糊规则，并将其优化，较好的解决了上述缺点，但却无法保证可解释性，甚至使之受到不同程度的破坏。因为这些算法的目标是获得最优的系统响应性能，而对于模糊集合划分个数、隶属度函数参数的选取却缺乏有效的指导。显然，在系统性能相差无几的情况下，模糊划分个数越少（进而直接导致更小的规则库），模糊划分物理意义越明显，也就更容易为人们理解。因此，如何既能得到最优的系统响应，又能同时保证较高的可解释性，成为一个重要的问题。

迄今为止，模糊系统的可解释性，尚无一般性的定义。已有的工作，主要是围绕着模糊子集相似性、完备性、清晰性、一致性以及紧凑性展开

[122~125]。文[106, 126, 127]提出了可解释性的概念, 阐述了完备性、清晰性、一致性以及紧凑性的定义, 并将其用于系统的优化。但是, 目前的工作, 没有考虑模糊划分的最简约性; 没有考虑模糊划分的完备性与清晰性之间的关系, 且没有给出相应的性能判别公式; 模糊划分完备性和模糊规则完备性的公式较复杂, 运算量大, 不利于工程实现。

进化计算(EC), 包括遗传算法(GA)、进化策略(ES)、进化规划(EP), 是一种模拟自然随机优化算法, 不依赖对象的数学模型, 具有全局搜索能力, 可用于模糊系统参数与结构的寻优。目前的工作主要集中于应用遗传算法优化模糊系统[128]。但是, 遗传编码的选择是一个较难解决的问题。如果选用常规的二进制编码, 染色体的长度会随着输入输出变量和模糊划分数目的增加而显著增加。如果选用其他的编码方式, 会导致算法的复杂。采用进化策略(ES)设计模糊系统的研究还相对较少。由于 ES 采用实数直接编码, 以及处理约束的简单性, 使得 ES 优化模糊系统更为简单有效[127, 106, 129]。

本章首先给出最简约模糊划分的定义; 提出了模糊划分的完备-清晰性的概念, 并给出计算公式; 进一步讨论了模糊规则的完备性-紧凑性; 以及模糊规则的一致性。在此基础上, 应用进化策略, 对模糊系统进行寻优, 使其获得较好系统响应性能和较高可解释性。

## 5.2 模糊系统的可解释性

知识是可以被人和机器理解以及重复应用的数据或信息, 但是, 机器可以理解的知识, 人不一定容易理解, 反之亦然。例如, 神经网络可以通过学习从数据中提取知识, 在这个过程中, 知识被表达为神经网络的连接方式和权值, 可以被机器理解, 但是人无法理解。因此, 将知识表达为容易被人们理解的方式, 是一个很重要的工作。模糊系统的可解释性, 就是

指模糊系统所蕴涵的知识被人们理解的难易程度。

模糊系统的可解释性，包括最简约模糊划分、完备-清晰性、模糊规则的完备性、紧凑性和一致性，分述如下。

不失一般性，本章讨论的均为 MISO 模糊系统，其输入变量  $x_j, j=1,2,\dots,M$ ，划分成  $MI_j$  个模糊子集  $A_{ij}$ ，其中  $i=1,2,\dots,MI_j, j=1,2,\dots,M$ ，隶属度函数为高斯函数。

### 5.2.1 最简约模糊划分

考察上述 MISO 模糊系统，输入输出变量划分模糊子集有多种不同的划分方案，假定其中第  $n$  种方案是，将输入变量  $x_j, j=1,2,\dots,M$ ，划分成  $MI_n$  个模糊子集  $A_{ni}(x_j), i=1,2,\dots,MI_n$ ；输出变量  $y$  划分成  $MO_n$  个模糊子集  $B_{nl}(y), l=1,2,\dots,MO_n$ ，其隶属度函数  $\mu_{A_{ni}}$  和  $\mu_{B_{nl}}$  的各项参数均可任意整定，那么，该模糊划分方案被称为最简约模糊划分，当且仅当存在  $f_{En} \leq f_E$ ， $f_{Em} \leq f_E$ ， $|f_{En} - f_{Em}| < \varepsilon$ ，满足：

$$MO_n + \prod_{j=1}^M MI_{nj} \leq MO_m + \prod_{j=1}^M MI_{mj} \quad (5.1)$$

其中： $m$  指其他任意一种划分方案； $f_E$  是给定的系统响应性能指标； $f_{En}$  和  $f_{Em}$  分别是第  $n$  种和第  $m$  种方案的系统响应性能； $\varepsilon > 0$ 。

显然，在系统性能可以满足的情况下，输入输出的模糊子集划分少的系统，其可解释性，明显优于模糊子集划分多的系统。

### 5.2.2 模糊划分的完备-清晰性

模糊划分的完备-清晰性是模糊系统可解释性的重要组成部分之一。如图 5.1 所示，划分为 5 个完备性和清晰性都很好的模糊子集，可以定义为 {NL, NS, ZE, PS, PL}。然而，图 5.2 中的模糊子集不完备，即有的输入/输出没有相对应的模糊子集；而图 5.3 中的模糊子集不清晰，我们很难分清各模糊子集的区别。

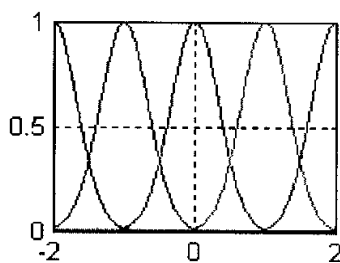


图 5.1 完备且清晰的模糊划分

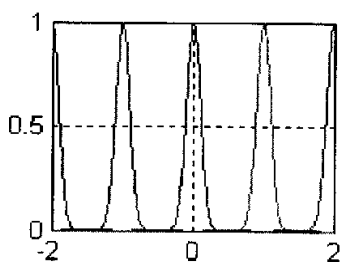


图 5.2 清晰但不完备的模糊划分

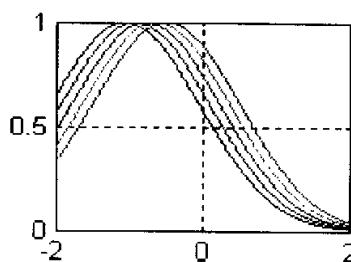


图 5.3 完备但不清晰的模糊划分

文[106]从所有的输入都应该有相对应的模糊子集出发，给出了模糊划分完备性的定义，但是要将所有的输入量  $x$  代入该式计算，运算量大，不



利于工程实现；本章考虑到模糊划分的完备性和清晰性是一对密不可分的性质，是互相联系互相制约的，所以放在一起讨论。本章从两相邻模糊子集之间的距离出发，给出了模糊划分的完备-清晰性的定义：

考察上述 MISO 模糊系统模糊划分的完备-清晰性为：

$$Compl\_Distin_{\mu} = \frac{(\omega_{i+1} + \omega_i) - (C_{i+1} - C_i)}{\omega_{i+1} + \omega_i} - 0.5 \quad (5.2)$$

其中  $i=1,2,\dots,MI_j$ ,  $j=1,2,\dots,M$ ,  $\omega$  是隶属度函数的标准差； $C$  是均值。

如果  $|Compl\_Distin(i)| \leq \varepsilon$  ( $\varepsilon > 0$ )，即同时具有较好的完备性和清晰性。

如果  $Compl\_Distin(i) > \varepsilon$  ( $\varepsilon > 0$ )，即完备性较好，但清晰性较差。

如果  $Compl\_Distin(i) < -\varepsilon$  ( $\varepsilon > 0$ )，即清晰性较好，但完备性较差。其中  $\varepsilon$  为给定的正数。

### 5.2.3 模糊规则的完备性与紧凑性

模糊规则的完备性，简单的讲，就是每一输入的每一个模糊子集都应该在模糊规则的前件中至少出现一次。基于此，给出如下定义：

考察上述 MISO 模糊系统的模糊规则是完备的，当且仅当满足：

$$sum_{premise}(A_{\mu}) \geq 1 \quad (5.3)$$

其中  $sum_{premise}(A_{\mu})$  是任意一个模糊子集在模糊前件中出现的次数。

模糊规则的紧凑性，是在模糊划分确定的情况下，评价模糊规则库规模的指标。定义如下：

考察上述 MISO 模糊系统，其模糊规则库是紧凑的，当且仅当存在

$f_{En\_rule} < f_E$ ,  $f_{Em} < f_F$ ,  $|f_{En\_rule} - f_{Em}| < \varepsilon$ , 满足：

$$n\_rule = \min(m), m \in \{0,1,\dots,\prod_{j=1}^M MI_j\} \quad (5.4)$$

其中  $n\_rule$  是模糊规则数。

### 5.2.4 模糊规则的一致性

模糊规则的一致性，是指相似的输入，应该得到相似的输出，否则会导致系统性能的下降，也必然影响系统的可解释性。本章引用文[127]中的定义，简述如下：

两条模糊规则的一致性  $Cons(R_i, R_k)$  为：

$$Cons(R_i, R_k) = \exp\left\{-\frac{\left(\frac{SRP(R_i, R_k)}{SRC(R_i, R_k)} - 1.0\right)^2}{\left(\frac{1}{SRC(R_i, R_k)}\right)^2}\right\} \quad (5.5)$$

其中  $SRP(R_i, R_k)$  为模糊规则前件的相似性， $SRC(R_i, R_k)$  为模糊规则后件的相似性。

## 5.3 基于进化策略生成可解释性模糊系统

模糊逻辑在系统控制领域取得了巨大的成功，尤其在控制系统难于建模、存在大的不确定性和强非线性时。近年来工业领域的成功应用和理论上的不断突破[109]都极大的促进了模糊系统的发展。但是模糊系统也有缺点。文[133]曾批评模糊控制规则没有提取深层次知识的能力，因为它们通常是建立在人类经验和直觉的基础之上的。

随着软计算，或者称之为计算智能的出现，模糊控制获得了新的发展动力。文[134]将 MPL 网络、RBF 网络、混合 pi-sigma 网络以及其他神经网络模型应用于模糊隶属函数和后件参数的自适应。上述工作的成功之处在于无需事先精确确定模糊规则和模糊划分的参数。

进化计算也被应用于模糊系统的设计之中。自从文[128]首次尝试应用遗传算法调节模糊规则库的某些参数，许多学者致力于研究应用遗传算

法设计模糊系统。GA 不仅可以同时优化参数和规则数目，而且还可以优化模糊规则的结构以获得一个紧凑的模糊系统。

应用 ES 设计模糊系统的工作还相对较少。本章应用 ES 而不是 GA 主要是基于两点考虑。其一是 ES 是实数直接编码的，因此目标向量的长度仅随着待优化变量数目的增长而线性的增长。由于要同时考虑隶属度函数参数和模糊规则的结构，待优化变量的数目会显著增加。如果采用常规的 GA 二进制编码，目标向量的长度会增加到难以忍受的程度。其二是 ES 处理约束问题比较容易，可以保证该算法能够搜索更大的参数空间。

### 5.3.1 采用进化策略的混和参数优化

进化策略(ES)可以用于模糊系统的参数和结构的优化。由于待优化变量包括实数和整数，所以我们采用了一种稍加变化的  $(\mu, \lambda)$ -ES 算法 [106,130]。描述如下：

$$(\mu, \lambda) - ES = (I, \mu, \lambda; m, s, \sigma; f, g) \quad (5.6)$$

其中， $I$  是由实数和整数组成的向量，代表一代种群中的一个个体； $\mu$  和  $\lambda$  分别表示父代和子代的个数； $m$  是变异算子，即 ES 中的主要算子； $s$  代表选择的方法，在这里下一代的父代，只能从  $\lambda$  个子代中选取； $\sigma$  是变异步长，在 ES 算法中，不仅仅变量变异，变异步长  $\sigma$  也要进行变异； $f$  是目标函数； $g$  是约束条件。目标参数和步长参数的变异公式如下：

$$\sigma'_i = \sigma_i \cdot \exp(\tau_1 \cdot N(0,1) + \tau_2 \cdot N_i(0,1)), i = 1, 2, \dots, Q \quad (5.7)$$

$$I'_i = I_i + \sigma'_i \cdot N_i(0,1), i = 1, 2, \dots, Q_1 \quad (5.8)$$

$$I'_i = I_i + \lfloor \sigma'_i \cdot N_i(0,1) \rfloor, i = Q_1 + 1, Q_1 + 2, \dots, Q \quad (5.9)$$

其中， $N(0,1)$  和  $N_i(0,1)$  是 0 均值，标准差为 1 的正态分布随机数； $Q$  是目标参数的总数量； $Q_1$  是实数值目标参数的个数； $Q - Q_1$  是整数值目标参数的个数， $\lfloor \cdot \rfloor$  是取整符号。 $\tau_1$  是标准差向量  $\sigma$  的整体步长参数， $\tau_2$  是  $\sigma$

的每个分量  $\sigma_i$  的步长参数。在本章中，待优化的参数是实数编码的隶属度函数的参数，和整数编码的模糊规则结构参数。

为了更好的获得全局最优解，当  $\sigma$  变得非常小时，给它重新赋一个初值 1，使它获得跳出局部极小的能力。同时，在每一次进化过程中，记录迄今最优的个体，但不参加下一代的竞争。

### 5.3.2 模糊系统编码和适值函数的建立

对于模糊子集的参数本章采用直接编码的方式。不失一般性，隶属度函数使用高斯函数：

$$A(x) = \exp\left\{-\frac{(x-c)^2}{\omega^2}\right\} \quad (5.10)$$

因此每一个隶属度函数有两个参数，即中心  $c$  和宽度  $\omega$ 。

模糊规则结构的编码很重要，也较复杂。考虑前述 MISO 模糊系统，其任意一个变量  $x_j$  的模糊子集数是  $MI_j$ ，那么规则库最多有  $N = \prod_{j=1}^M MI_j$  条

规则。这样，模糊规则前件可以编码成如下矩阵：

$$\mathbf{Struc}_{premise} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{bmatrix} \quad (5.11)$$

其中  $a_{ij} \in \{0, 1, 2, \dots, m_j\}$ ,  $i = 1, 2, \dots, N$ ;  $j = 1, 2, \dots, M$ 。整数  $1, 2, \dots, m_j$  代表  $x_j$  相应的模糊子集；当  $a_{ij} = 0$  的时候表示变量  $x_j$  没有出现在第  $i$  条规则前件里面；当所有的  $x_j$  都不出现在第  $i$  条规则前件里的时候，认为该规则不存在。相应的，假定输出变量最多有  $K$  个模糊子集，则模糊规则后件可以编码成由正整数组成的向量：

$$\mathbf{Struc}_{consequent} = [c_1, c_2, \dots, c_N]^T \quad (5.12)$$

其中  $c_i \in \{0, 1, 2, \dots, K\}, i = 1, 2, \dots, N$ 。当  $c_i = 0$  的时候表示着第  $i$  条规则不存在。该方法适用于 Mamdani 规则。如果应用 Takagi-Sugeno 规则，就不需要对后件编码。

如前所述，模糊系统不仅应该具有较好响应性能，还应该具备较好的可解释性。因此，进化策略的适值函数应该包括可解释性性能指标。其中模糊划分的完备-清晰性指标，根据(5.2)式，得到：

$$f_{\text{Compl\_Distn}} = \sum_{j=1}^M \sum_{i=1}^{M_j} |\text{Compl\_Distn}_j| \quad (5.13)$$

显然，模糊划分的完备性和清晰性越高，(5.13)式的值越小。请注意，优化的任务在此是搜索更小的适值函数，因此，适值函数越小，该个体生存的可能性就越大。

模糊规则的一致性指标[106,127]为：

$$f_{\text{Incons}} = \sum_{i=1}^N \sum_{\substack{1 \leq k \leq N \\ k \neq i}} [1.0 - \text{Cons}(R_i, R_k)] \quad (14)$$

模糊规则的完备性和紧凑性无法直接写到适值函数中，需要稍加改动。我们用  $f_{\text{incomp\_R}}$  表示模糊规则的完备性指标，将其作为一个惩罚因子，当检验模糊规则不满足式(5.3)时， $f_{\text{incomp\_R}}$  就被赋予一个很大的值，使得该个体不可能被保存下来。

至于模糊规则的紧凑性，没有在适值函数中显式的给出，但是进化策略更倾向于选择出一个紧凑的系统[106]。例如，一致性  $f_{\text{Incons}}$  是随着模糊规则数单调递增的，模糊规则数多的子代， $f_{\text{Incons}}$  就有可能更大一些，当其他性能指标都很接近时，其保存下来的可能性，就会大大降低。

### 5.3.3 优化过程

为了保证生成一个具有可解释性的模糊系统，优化过程分为两阶段来

做，每一阶段的目的、步骤以及适值函数的选取等等都有所不同，分述如下：

第一阶段，目的是生成具有最简约模糊划分的模糊系统。本章考虑在先验知识较少的情况下，无法确定将输入输出的论域划分为几个区间较好，所以先将其划分成较多的区间，在进化寻优的过程中，在保证系统响应性能的情况下，逐步减小划分的个数，最终获得最简约模糊划分。步骤如下：

1. 初始化：划分成  $m$  个区间 ( $m$  较大)，适值函数只选用系统响应性能指标。本章考虑到兼顾控制系统的动态和静态性能，采用 ITAE，其离散形式为：

$$f_E = \sum_{n=1}^N nT_s |e(n)|T_s ; \quad (5.15)$$

2. 应用进化策略对模糊系统寻优，使得适值函数达到给定的性能指标，即  $f_{Em} \leq f_E$ ，将此时的最优和次优个体记录备份；
3. 考察输入/输出模糊子集划分的完备-清晰性，找到完备性最好即清晰性最差的两个模糊子集，任意剔除其中一个，即相当于将输入和输出统一划分成  $m = m - 1$  个区间；
4. 考察改变后的适值函数是否满足下式：  $f_{Em-1} < f_E$ ，  $|f_{Em-1} - f_{Em}| < \varepsilon$ ，如果满足，重复第 3 步；如果不满足转到第 2 步；
5. 直到划分  $m$  个区间可以满足性能指标，而  $m = m - 1$  无法满足时，确定将输入和输出划分成  $m$  个区间是最简约模糊划分。

第二阶段，目的是最终生成具有可解释性的模糊系统。适值函数选用

$$f = f_E + \xi_1 * f_{Incons} + \xi_2 * f_{Cmpl\_Distm} + f_{Incomp\_R} \quad (5.16)$$

其中， $f_{Incons}$  表示模糊规则的一致性指标， $f_{Cmpl\_Distm}$  表示模糊划分的完备-清晰性， $f_{Incomp\_R}$  表示模糊规则的完备-清晰性。 $\xi_1$  和  $\xi_2$  为权重系数。以前一阶段产生最优和次优个体做为父代，继续应用进化策略对模糊系统寻

优, 使得该适值函数达到给定的综合性能指标。

#### 5.3.4 仿真实例及分析

本章采用两个仿真实例来验证该算法的有效性。在第一个例子里, 被控对象选用一个非线性系统[131], 以差分方程的形式给出:

$$y_p(k+1) = \frac{y_p(k)}{1+y_p^2(k)} + u^3(k) \quad (5.17)$$

该模糊控制器是一个典型的以误差  $e$  和误差变化量  $\Delta e$  为输入, 控制量  $u$  为输出的 MISO 模糊系统, 将输入输出初步划分为 7 个模糊区间, 隶属度函数选为均值  $c = \{-3, -2, -1, 0, 1, 2, 3\}$ , 标准差  $\omega = 0.5$  的高斯函数, 并随机确定模糊规则。

为了便于比较, 采用 3 种基于进化策略的优化方案。一种是适值函数只选用系统响应性能指标, 即不考虑可解释性。第二种是应用文[7]中的方法; 第三种是应用本章的方法。

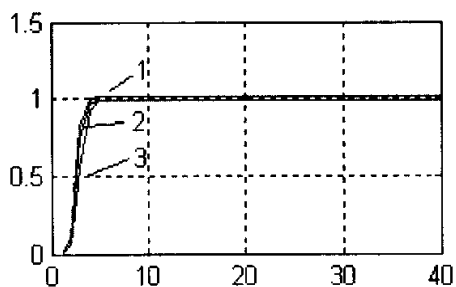


图 5.4 阶跃响应

三种方案得到的阶跃响应如图 5.4 所示。可见三种方案都能设计出响应性能满意的控制器, 其 ITAE 指标相差不大。但是其隶属度函数和模糊规则有较大的差别。

其中采用第一种方案时, 模糊子集分布如图 5.5 所示。不难看出, 在

不检验可解释性的情况下，应用进化策略对模糊系统寻优，其系统响应性能良好，但是模糊划分的清晰性很差。以图 5.5(a)为例，我们很难分清“负大”和“负中”以及“正大”和“正中”。规则数为 47 条，略小于最多的 49 条。

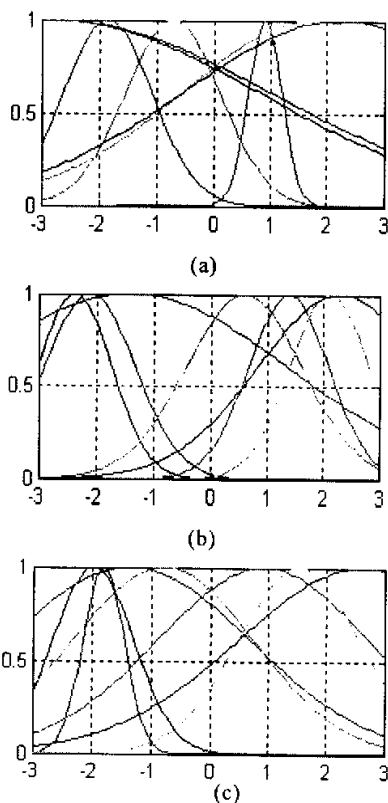


图 5.5 隶属度函数（不检验可解释性）：  
(a)误差  $e$ ；(b)误差变化量  $\Delta e$ ；(c)控制量  $u$



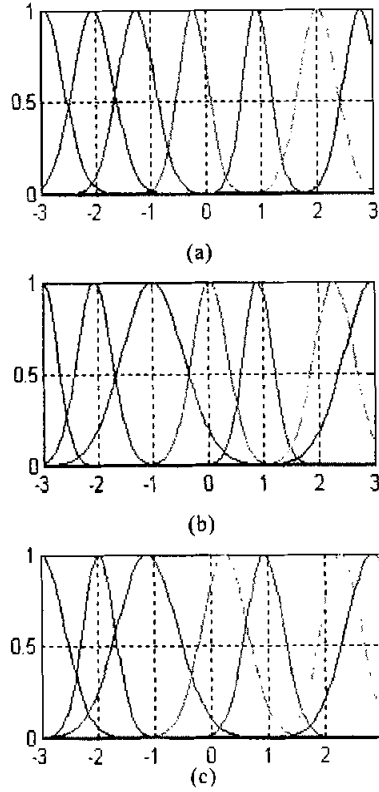


图 5.6 隶属度函数（检验可解释性）：

(a)误差  $e$ ；(b)误差变化量  $\Delta e$ ；(c)控制量  $u$

我们再以文[127]的方法产生模糊控制器，得到的模糊子集的隶属度函数分布如图 5.6 所示。可以看出，在检验可解释性的情况下，应用进化策略对模糊系统寻优，其系统响应性能良好，而且模糊划分的完备-清晰性也很好。规则数为 38 条，小于最多的 49 条，紧凑性也得到了较好的保证。

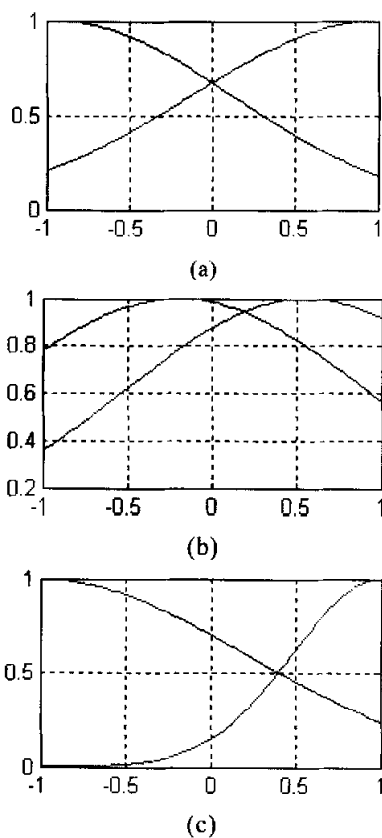


图 5.7 隶属度函数（本文方法）：  
(a)误差  $e$ ；(b)误差变化量  $\Delta e$ ；(c)控制量  $u$

我们再以本章提出的方法产生模糊控制器，得到的结果如图 5.7 所示。可见系统响应性能良好，而且输入输出只有 2 个模糊子集，是最简约模糊划分；其完备-清晰性也很好；模糊规则数为 3，小于最多的 4 条，保证了规则的完备性和紧凑性。

本章的第二个仿真实例是采用经典的 Box-Jenkins 煤气炉[132]作为仿真实验的对象，其数据集包括 296 组数据，其中  $u(t)$  为输入的煤气流量， $y(t)$  为释放出的  $CO_2$  浓度，本章采用  $u(t-4)$  和  $y(t-1)$  作为模糊系统的输入，将输入输出初步划分为 7 个模糊区间，隶属度函数选为均值  $c = \{-3, -2, -1, 0, 1, 2, 3\}$ ，标准差  $\omega = 0.5$  的高斯函数，并随机生成模糊规则。

表 5.1 各方法的比较

模型	输入个数	规则数	RMSE
Tong'77	2	19	0.684
Pedrycz'84	2	81	0.565
Xu'87	2	25	0.572
Peng'88	2	49	0.548
Sugeno'91	6	2	0.261
Sugeno'93	3	6	0.435
Wang'96	2	5	0.397
本文方法	2	3	0.472

该算法的结果是，均方根误差为 0.472，输入输出只有 2 个模糊子集，是最简约模糊划分；模糊规则为 3，小于最多的 4 条，保证了规则的完备性和紧凑性。从图 5.8 可以看出，模糊划分的完备-清晰性也很好，因此该系统表达的知识就很容易被人理解。表 5.1 给出了不同方法的比较。

## 5.4 本章小结

本章首先分析了模糊系统的可解释性，定义了最简约模糊划分、模糊划分的完备-清晰性、模糊规则的完备与紧凑性和模糊规则的一致性。然后应用进化策略对模糊系统寻优，在适值函数加入上述指标。应用该方法，不仅模糊系统的性能被优化，而且可解释性也大大提高。针对典型非线性控制和 Box-Jenkins 煤气炉数据的仿真实验表明了该方法的有效性。

## 第六章 基于进化策略的*K-means*聚类算法

针对 *K-means* 聚类算法易陷入局部极小以及 *K* 值选取的问题, 本文提出一类基于进化策略的聚类算法, 可以有效的搜索最优聚类中心和聚类个数 *K*; 还提出了确定 *K* 值范围的经验公式, 以减小搜索空间, 提高搜索效率, 并给出理论分析。相对遗传算法而言, 本方法编码简单, 种群较小。对 Fisher's iris 数据集的仿真实验表明, 该方法得到最优解的可能性比经典算法大得多。

### 6.1 问题的提出

*K-means* 聚类算法简单有效, 已经应用到许多领域, 包括模式识别, 图像和语音数据压缩, 径向基神经网络学习算法等等。但是, 该算法本质上是一种局部搜索技术, 它采用所谓的爬山法 (hill-climbing) 来搜寻最优解, 因此必然会存在局部极小的问题。而且, 目前没有较好的方法来确定 *K* 值, 这对聚类效果会产生较大的影响。

进化计算 (EC), 包括遗传算法 (GA)、进化策略 (ES)、进化规划 (EP), 是一种模拟自然随机优化算法, 不依赖对象的数学模型, 具有全局搜索能力, 已经成功应用于多种优化问题 [130]。文 [135] 应用遗传算法优化聚类算法, 效果较好, 但是仍然需要试凑 *K* 值; 而且遗传编码的选择也是一个较难解决的问题。如果选用常规的二进制编码, 染色体的长度会随着维数的增加或精度的提高而显著增加。如果选用其他的编码方式, 会导致算法的复杂。这无疑会影响 GA 的有效性。采用进化策略 (ES) 进行聚类分析的研究还相对较少。由于 ES 采用实数直接编码, 以及处理约束的简单性, 使得 ES 优化聚类算法更为简单有效。

本文提出了基于进化策略的 *K-means* 聚类算法, 可以自动搜索最佳的聚类数  $K$ 。进化策略可以全局搜索, 具备较强的局部极值逃逸能力; 而且, 对于初始值不敏感。本文还提出了确定  $K$  取值范围的经验公式, 可以有效减小搜索空间, 提高搜索效率。

## 6.2 *K-means* 聚类算法

设  $R^p$  是  $p$  维实数集,  $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$  是任一有限数据集, *K-means* 聚类算法的目标函数是

$$J_m = \sum_{j=1}^n \sum_{i=1}^k u_{ji} d_{ji}^2 \quad (6.1)$$

其中:  $d_{ji} = \|x_j - c_i\|$  为第  $j$  个数据点到第  $i$  个聚类中心的距离;  $c_1, c_2, \dots, c_k$  是  $K$  个聚类中心,  $c_i \in R^p$ ,  $u_{ji} \in \{0, 1\}$ , 表示第  $j$  个数据属于 ( $u_{ji} = 1$ ) 或不属于 ( $u_{ji} = 0$ ) 第  $i$  类。

其步骤为: 任意选定一组  $K$  个初始聚类中心, 按欧氏距离把  $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$  分到  $K$  类中去; 重新计算每一类的均值, 即聚类中心, 直到中心不再发生改变。显然, 该算法无法确定  $K$  值, 存在局部极小的问题, 而且与初始聚类中心的选择有极大的相关性。

## 6.3 基于进化策略的聚类算法

本文应用 ES 而不是 GA 主要是基于两点考虑。其一是 ES 是实数直接编码的, 因此目标向量的长度仅随着待优化变量数目的增长而线性的增长。如果采用常规的 GA 二进制编码, 目标向量的长度会增加到难以忍受

的程度。其二是 ES 处理约束问题比较容易，可以保证该算法能够搜索更大的参数空间。但是这并不意味着本文认为进化策略优于遗传算法或者相反，正如 NFL 定理指出的那样，无法设计出一种算法，在解决所有优化问题的时候，都优于其他算法，而只能认为该算法在此领域占有优势[9, 79]。

### 6.3.1 进化策略简介

进化策略(ES)是一种简单高效的启发式多点随机搜索算法，它更倾向于应用进化论中变异和选择的概念，通过不断的变异和选择，使群体的适应度不断提高，最终得到最优的个体。由于本文的中待优化变量包括实数和整数，所以我们采用了一种稍加变化的  $(\mu, \lambda)$ -ES 算法[106]。简述如下：

$$(\mu, \lambda) - ES = (\mathbf{Ind}, \mu, \lambda; m, s, \sigma; f, g) \quad (6.2)$$

其中， $\mathbf{Ind}$  是由实数和整数组成的向量，代表种群中的一个个体； $\mu$  和  $\lambda$  分别表示父代和子代的个数； $m$  是变异算子，即 ES 中的主要算子； $s$  代表选择的方法，在这里下一代的父代，只能从  $\lambda$  个子代中选取； $\sigma$  是变异步长，在 ES 算法中，不仅仅变量变异，变异步长  $\sigma$  也要进行变异； $f$  是目标函数； $g$  是约束条件。目标参数和步长参数的变异公式如下：

$$\sigma_i' = \sigma_i \cdot \exp(\tau_1 \cdot N(0,1) + \tau_2 \cdot N_i(0,1)), i = 1, 2, \dots, Q \quad (6.3)$$

$$\mathbf{Ind}_i' = \mathbf{Ind}_i + \sigma_i' \cdot N_i(0,1), i = 1, 2, \dots, Q_1 \quad (6.4)$$

$$\mathbf{Ind}_i' = \mathbf{Ind}_i + \lfloor \sigma_i' \cdot N_i(0,1) \rfloor, i = Q_1 + 1, Q_1 + 2, \dots, Q \quad (6.5)$$

其中， $N(0,1)$  和  $N_i(0,1)$  是 0 均值，标准差为 1 的正态分布随机数； $Q_1$  是实数值目标参数的个数； $Q - Q_1$  是整数数值目标参数的个数， $\lfloor \cdot \rfloor$  是取整符号。 $\tau_1$  是标准差向量  $\sigma$  的整体步长参数， $\tau_2$  是  $\sigma$  的每个分量  $\sigma_i$  的步长参数。

为了更好的获得全局最优解，当  $\sigma$  变得非常小时，给它重新赋一个初值 1，增强它跳出局部极小的能力。同时，在每一次进化过程中，记录迄今最优的个体，但不参加下一代的竞争。

### 6.3.2 给定 $K$ 值时的进化编码和适值函数的建立

为简单起见,本文首先考虑给定 $K$ 值时的进化聚类算法。对于聚类的参数的编码,有两种方式。在第一种方式中,设 $n$ 个数据要分成 $K$ 类,每个个体 $Ind_i = \{ind_{i1}, ind_{i2}, \dots, ind_{in}\}$ 是一个整数串,表示一种分类结果,其中 $ind_{in} \in \{1, 2, \dots, K\}$ ,表示第 $n$ 个数据属于第 $ind_{in}$ 类。变异产生的子代,表示新产生的分类组合。计算每一类的均值,得到聚类中心。当达到给定的性能指标时,算法结束。在第二种方案中,每个个体 $Ind_f = \{ind_{f1}, ind_{f2}, \dots, ind_{fn}\}$ 是一个实数串,表示 $K$ 个数据聚类中心。变异产生的子代,表示新的聚类中心。

对于两种方案,均选用相同的适值函数:

$$F = \frac{\frac{1}{n} \sum_{p=1}^K \sum_{i=1}^n \|x_i - c_p\|^2}{\frac{1}{K-1} \sum_{p \neq h} \|c_p - c_h\|^2} \quad (6.6)$$

(6.6)式的意义是:同一类的输入样本与聚类中心的欧式距离越近,不同聚类中心的欧式距离越远,则聚类结果的合理性越好。进化策略的任务是搜索适值函数 $F$ 的最小值。进化策略的流程如图 6.1 所示。

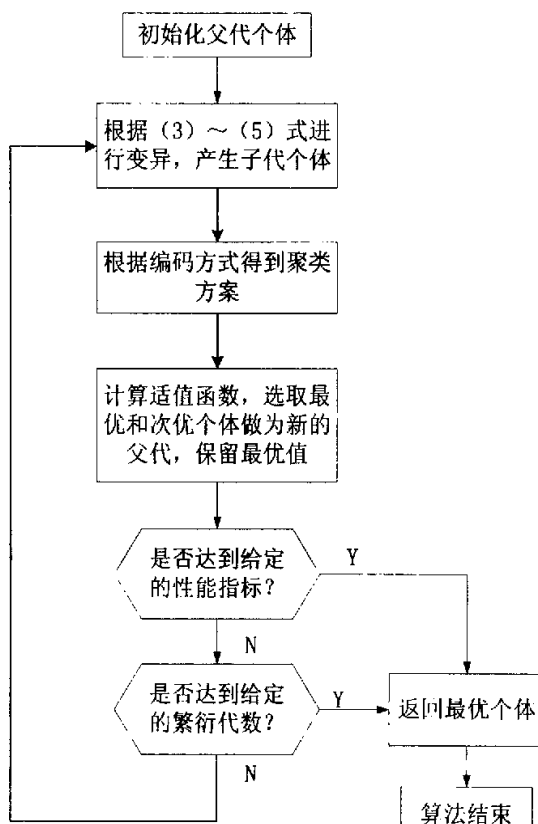


图 6.1 进化聚类的流程图

Figure 6.1: A Flowchart of Evolution Clustering

### 6.3.3 未给定 $K$ 值时的进化编码和适值函数的建立

在实际应用中, 尤其在先验知识不足的情况下, 很难得到一个较好的  $K$  值, 传统的  $K$  均值聚类只能用试凑的方法人为的指定  $K$  值, 这一点对聚类结果的精度影响很大。因此, 本文提出, 在搜索聚类中心的同时, 也应



该搜索  $K$  值。其编码方式为：在第一种聚类方案的基础上，在每个个体  $Ind_i$  中增加一个分量  $ind_{i0}$ ，即

$$Ind_i = \{ind_{i0}, ind_{i1}, ind_{i2}, \dots, ind_{im}\} \quad (6.7)$$

$Ind_i$  是一个整数串，表示一种分类结果，其中  $ind_{i0} = k$ ，表示聚类个数；其余各个分量与未给定  $K$  值编码方式的含义相同，只是其取值范围会根据  $K$  值的变化而相应的改变。搜索空间为  $\sum_{k=1}^n k^n$  个点。

### 6.3.4 提高搜索效率的经验公式

众所周知，进化算法是一种随机搜索算法，效率较低。如果能缩小搜索空间，就能够有效地提高搜索效率。因此本文提出如下经验公式：

$$2 \leq K \leq \left\lfloor \frac{n}{2} \right\rfloor \quad (6.8)$$

上式的意义是，对大多数实际问题来讲，最少分为 2 类，否则就失去了聚类的意义；最多分为  $\left\lfloor \frac{n}{2} \right\rfloor$  类，即平均最少每 1 类有 2 个数据，否则无实际意义。下面对于(6.8) 式的作用给出理论分析：

不失一般性，只考虑样本数  $n$  为偶数的情况， $n$  为奇数时推导过程类似。首先介绍一个引理：

$$\text{引理：若 } a_i \geq 0 \text{ 且 } b_i > 0 (i = 1, 2, \dots, n), \text{ 则满足 } \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \leq \sum_{i=1}^n \frac{a_i}{b_i}$$

证明：

$$\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} = \frac{a_1 + a_2 + \dots + a_n}{b_1 + b_2 + \dots + b_n}$$

$$= \frac{a_1}{b_1 + b_2 + \dots + b_n} + \frac{a_2}{b_1 + b_2 + \dots + b_n} + \dots + \frac{a_n}{b_1 + b_2 + \dots + b_n}$$

$$\because a_i \geq 0 \text{ 且 } b_i > 0 (i=1,2,\dots,n)$$

$$\therefore \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \leq \frac{a_1}{b_1} + \frac{a_2}{b_2} + \dots + \frac{a_n}{b_n} = \sum_{i=1}^n \frac{a_i}{b_i} \quad \text{证毕。}$$

令  $m = \frac{n}{2}$ , 则根据(6.8) 式给定  $K$  值范围后, 搜索空间有  $\sum_{k=2}^m k^n$  个点。其

与  $\sum_{k=1}^n k^n$  的比值为:

$$\frac{\sum_{k=2}^m k^n}{\sum_{k=1}^n k^n} = \frac{\sum_{k=1}^m k^n - 1}{\sum_{k=1}^m [k^n + (k+m)^n]} \quad (6.9)$$

$$\leq \frac{\sum_{k=1}^m k^n}{\sum_{k=1}^m [k^n + (k+m)^n]} \quad (6.10)$$

根据引理和(6.10) 式得到:

$$\frac{\sum_{k=1}^m k^n}{\sum_{k=1}^m [k^n + (k+m)^n]} \leq \sum_{k=1}^m \frac{k^n}{k^n + (k+m)^n} \quad (6.11)$$

$$\leq \sum_{k=1}^m \frac{k^n}{k^n + (2k)^n} = \sum_{k=1}^m \frac{1}{1 + 2^n} = \frac{n(n+2)}{8(1+2^n)} \quad (6.12)$$

该式表明, 利用(6.8) 式, 搜索空间会显著减小, 尤其是当  $n$  较大的时候。例如, 设数据样本数为 20, 代入上式可得缩小范围后搜索空间的点数小于原来的 0.0052%。显然, 当  $n$  更大时, 这个比例会更小。因此, (6.8)

式可以有效的提高搜索效率。实际上(6.8)式估计的仍然过于保守，尤其对于  $n$  较大的情况来说，该范围可以进一步缩小，以减小搜索空间，提高算法收敛的速度。

当然，(6.8)式只是根据众多实际情况归纳出的一个经验公式，理论上存在不满足(6.8)式的可能。因此本文建议，进行运算时，首先应用(6.8)式确定  $K$  值的范围，然后搜索最优个体；如果无法收敛，则去掉(6.8)式， $K$  值的范围确定为  $1 \sim n$ ，重新搜索最优个体。

6.4 仿真实例及分析

表 6.1 仿真结果

基于进化策略聚类						K 均值聚类		
给定 K 值 I		给定 K 值 II		未给定 K 值				
进化代数	性能指标	进化代数	性能指标	进化代数	性能指标	迭代次数	性能指标	
732	3.7614	566	3.7600	5764	3.7362	30	3.7614	
514	3.7614	1248	3.7343	4512	3.7343	54	3.7614	
457	3.7614	995	3.7614	6308	3.7614	32	11.6049	
1133	3.7516	851	3.7438	4363	3.7575	31	3.7575	
505	3.7575	745	3.7362	13750	3.7456	38	3.7614	
583	3.7392	822	3.7614	11092	3.7614	39	3.7438	
764	3.7456	677	3.7438	5556	3.7614	39	10.0468	
693	3.7614	956	3.7392	3609	3.7438	27	3.7362	
581	3.7438	1539	3.7614	7570	3.7392	42	3.7614	
525	3.7362	764	3.7438	18205	3.7343	32	3.7614	
平均	648.7	3.7519	916.3	3.7485	8072.9	3.7475	36.4	5.1696

1920 年左右, 植物学家收集了 3 种虹膜 (鸢尾属植物), 每种 50 个, 共计 150 个样本的数据, 包括萼片长度、宽度和花瓣的长度, 即著名的 Fisher's iris 数据集。这些数据的分类结果是已知的, 有助于各种方法之间进行比较。

为了便于比较, 本文分别采用传统的  $K$  均值聚类和上述 3 种基于进化策略的聚类方法对 Fisher's iris 数据集各做了 10 次仿真, 仿真结果如表 6.1 所示。其中采用进化策略的地方, 父代个数为 2, 子代个数为 14, 远远小于一般遗传算法的种群规模。

从表 6.1 中可以看出, 进化策略可以搜索出全局最优解, 结果比较稳定。而传统的聚类方法则有两次落入局部极小。但是进化策略所需要的进化代数较多, 其中, 在给定  $K$  值的情况下, 分别是传统聚类方法的迭代次数的 17.82 和 25.17 倍。在未给定  $K$  值的情况下, 第三种方法可以自动搜索出最佳的  $K=3$ , 进化代数是传统方法迭代次数的 221.78 倍。但是为了得到一个可靠的结果, 在实时性要求不强的情况下, 该方法还是具有较大的优越性。

## 6.5 本章小结

本章首先简述了  $K$ -means 聚类的不足, 提出了两种给定  $K$  值时的基于进化策略的聚类算法, 较好的解决了局部极小的问题; 然后针对  $K$  值选取的问题, 提出了可同时寻优  $K$  值的基于进化策略的聚类算法, 可搜索到最优  $K$  值, 使得类内距最小, 类外距最大。并且给出确定  $K$  取值范围的经验公式, 可有效减小搜索空间, 提高搜索效率。相对于遗传算法, 本方法也具有编码简单, 种群小的特点。通过对 Fisher's iris 数据集的仿真实验表明该算法是有效的。但是该方法也存在一些不足, 所需要的进化代数较多,

只适用于对实时性要求不高的场合。

## 第七章 引入全局中间交叉算子的进化策略与LMS算法异同的探讨

### 7.1 引言

进化策略是一种受生物进化论启发而建立的优化算法[12],经典的进化策略是不采用交叉算子的,虽然在后来发展中与遗传算法相互影响和融合,引入了交叉算子[58],但是基于效率和复杂程度等方面的考虑,大多数实际应用中均不采用交叉算子。遗传算法侧重交叉算子,而进化策略侧重变异算子,进化规划更是只用变异算子,完全放弃交叉算子。长期以来各学派努力不止,争论也不休。比如文[87]认为与变异相比较,交叉没有任何竞争优势;文[88]则批评了这一论断,认为对某些问题仅有变异算子是不够的。而文[89]的结论是将连续参数优化问题表示成二进制位串并且允许用交叉来做搜索的做法并不总是最好的。Spears 从定义杂交和变异的两个潜在作用——分裂(disruption)和构造(construction)出发,考虑它们在执行这两个作用上的差异,研究结果表明:对于分裂,变异比杂交有效,虽然它缺少杂交保留个体共同等位基因的能力;然而,对于构造,杂交比变异更有效[90]。

LMS 算法是一种基于梯度法的单点搜索策略,在白适应滤波器和 Adaline 单神经元等方面有着重要的应用[136~139]。传统的观点一般认为进化策略是一种全局随机多点搜索技术,与 LMS 有着完全不同的特点,是两种截然不同的优化技术。但是本章的研究表明,当进化策略引入全局中间交叉算子之后,可以认为这两者之间的差异更多的是基于我们平时的一些“想当然”的观念上的,进化策略可以看作是一种“概率”意义上的梯度搜索技术,而 LMS 算法也可以看作一种多点搜索技术。

## 7.2 基础知识

### 7.2.1 交叉算子简介

文献[58]中介绍了以下四种交叉操作：

$$x_i = \begin{cases} x_{a,i} \text{ or } x_{b,i} & \text{discrete-recombination} \\ \frac{1}{2}(x_{a,i} + x_{b,i}) & \text{intermediate-recombination} \\ x_{a,i} \text{ or } x_{b,i} & \text{global-discrete-recombination} \\ \frac{1}{2}(x_{a,i} + x_{b,i}) & \text{global-intermediate-recombination} \end{cases} \quad (7.1)$$

其中  $a, b, b_j$  是父代个体,  $i$  是向量  $x$  第  $i$  个元素。通常来说,在一代群体中,所有的父代个体具有相同的交叉概率,即参加交叉的父代是通过均匀随机数来确定的。在离散交叉情况下,子代的成员向量的每一维都是随机继承某个父代,即  $x_{a,i}$  或  $x_{b,i}$  的对应部分,相当于遗传算法中交叉点数量可变的交叉算子。在中间交叉情况下,子代的成员向量是父代成员向量的平均值。在全局交叉情况下,通过选取另外的父代  $x_{b,i}$  使子代继承群体中更多的遗传信息。

### 7.2.2 LMS算法简介

LMS 算法最早应用于自适应滤波器和 Adaline 单神经元等。自适应滤波器如图 7.1 所示[138]。它是由可变加权系数的抽头延迟线、一个加法器和一个自适应过程所构成的。这些权系数的输入信号就是在各延迟线抽头上的信号,用一个加法器将这些加权后的信号相加,在由一个自适应过程自动搜寻调节这些权系数以获得一个最优响应。

自适应滤波器的训练方法是基于梯度法的LMS算法。梯度法是使得在权向量上的每一步变化都正比于梯度向量的负值:

$$W_{k+1} = W_k + \mu(-\nabla_k) \quad (7.1)$$

LMS算法是利用梯度估计的最速下降的一种实现

$$W_{k+1} = W_k + \mu(-\hat{\nabla}_k) \quad (7.2)$$

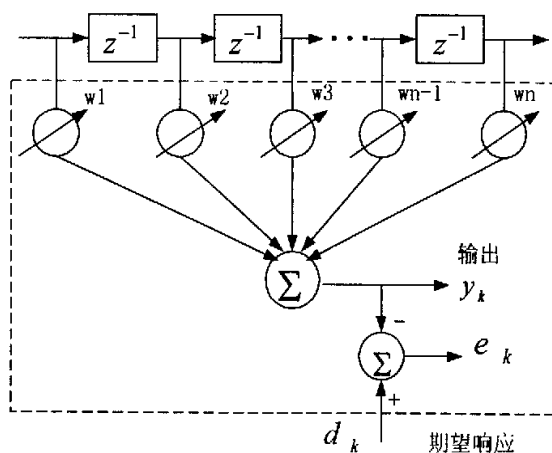


图 7.1 自适应滤波器

$$\text{梯度估计} \quad \hat{\nabla}_k = \begin{bmatrix} \frac{\partial e_k^2}{\partial w_1} \\ \vdots \\ \frac{\partial e_k^2}{\partial w_n} \end{bmatrix} = 2e_k \begin{bmatrix} \frac{\partial e_k}{\partial w_1} \\ \vdots \\ \frac{\partial e_k}{\partial w_n} \end{bmatrix} = -2e_k X_k \quad (7.3)$$

将 (7.3) 式代入 (7.2) 式得到LMS算法:

$$W_{k+1} = W_k + \text{delta} * e_k * X_k \quad (7.4)$$

其中  $\text{delta} = 2\mu$  是学习速率。

### 7.3 进化策略 (ES) 和LMS算法的比较

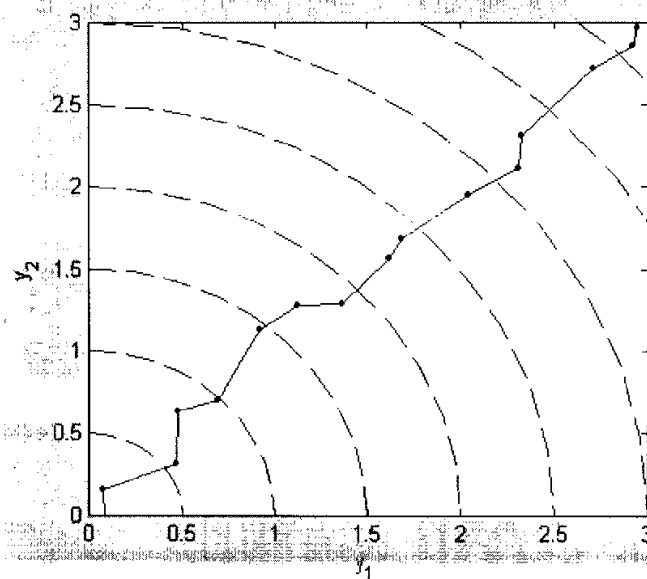
本节将在以下几个方面比较 ES 和 LMS 算法: 搜索路径是否沿梯度方向, 多点搜索和单点搜索策略; 以及逃出局部极值点的可能性。



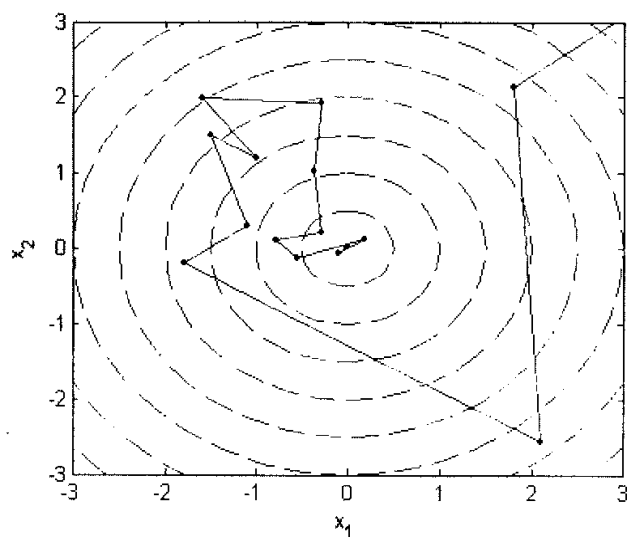
为了方便讨论上述问题，本节将  $n$  维搜索空间映射为 2 维空间。文献 [140, 141] 中应用了下列变换：

$$y_1 = \sqrt{\sum_{i=1}^{n/2} x_i^2}, \quad y_2 = \sqrt{\sum_{i=n/2+1}^n x_i^2} \quad (7.5)$$

ES 对目标函数是二次型函数时的寻优轨迹如图 7.2 (a) 所示。其中横坐标是  $y_1$ ，纵坐标是  $y_2$ ，设置问题的维数  $N=100$ ，根据公式(7.5)将其映射维 2 维空间。进化策略采用(2,14)-ES。显然从 2 维的观点来看，它是沿梯度负方向搜索的。这也意味着，ES 被局部极值点吸引，并且存在陷入局部极值点的可能。然而如果仍然从维数  $n=100$  的角度考虑，任意抽取两个参数  $x_i$  和  $x_j$ ，（本文取的是  $x_{10}$  和  $x_{80}$ ）情况会发生巨大的变化。图 7.2(b) 显示 [142]， $x_i$  和  $x_j$  的搜索路径是“混乱”的，虽然  $y_1$  和  $y_2$  是沿梯度负方向搜索的。也就是说，ES 微观层次上“杂乱无章”的、随机的行为，在宏观上却体现出确定的行为。在分别设置  $n=100, 50, 20$  相关试验中，可以观察到 ES( $y_1$  和  $y_2$ ) 几乎完全是沿梯度路径搜索的。然而，如果变异矢量的长度大于当前点到最优点的长度，ES 会暂时偏离梯度路径，由于搜索步长的自适应机制，这种情况时有发生。在这种情况下，大多数的子代都或多或少的接近平均适应度，因此选择操作就显得有较大的随机性。所以个体  $x_i$  就表现出频繁偏离梯度路径的情况，如图 7.2 (b)所示。



(a)



(b)

图 7.2 ES 对二次型函数的寻优轨迹：  
(a)  $y_1$  和  $y_2$  的轨迹；(b)  $x_1$  和  $x_2$  的轨迹

第二个问题是有关多点搜索方面的。传统的观点一般认为，进化算法由于多点搜索策略，所以局部极值逃逸能力较高，可以找到全局最优解，以及大量的经验式的证据[143, 144, 145]，似乎支持着这个假说。基于这样一个理念，大多数的进化策略使用者都把参数设定为  $\mu > 1$ ， $\lambda > 1$ 。

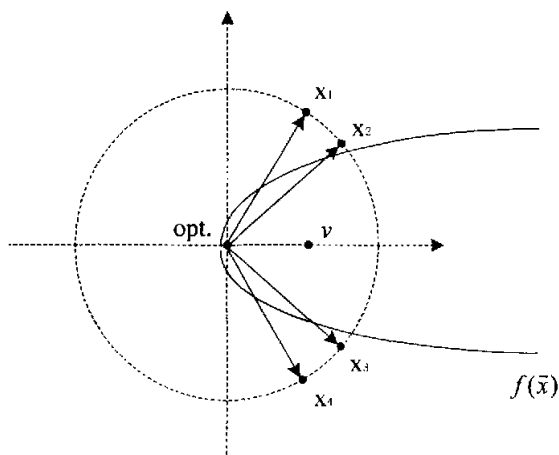


图 7.3  $(\mu/\mu_t, \lambda)$ -ES 的全局中间交叉运算

文献[142, 145]在理论上分析了  $(\mu/\mu_t, \lambda)$ -ES (其中  $\mu_t$  是全局中间交叉算子， $\mu$  个父代个体在生成  $\lambda$  个子代个体之前先进行此交叉操作)。如图 7.3 所示， $(\mu/\mu_t, \lambda)$ -ES 的特点之一就是全局中间交叉运算，生成一个虚拟的父代个体  $\bar{v} = \frac{1}{\mu} \sum_{i=1}^{\mu} \bar{x}_i$ ，做为第  $t$  代  $\mu$  个父代个体的中心。换言之， $(\mu/\mu_t, \lambda)$ -ES 用一个虚拟的父代个体  $\bar{v}$  替代了整个父代群体。研究表明它有较好的求精(refining)能力，充分利用已获得的解空间信息逼近当前局部最优解(exploitation) 的能力，即但求泛 (reforming)能力，搜索解空间上的全局最优解 (exploration)略差一些。而且，生成一个虚拟的父代个体，似

乎与多点搜索策略的理念相抵触。

相反的, LMS 算法看上去似乎是单点搜索策略, 但是, 该算法要求至少  $n$  个不同方向的信息, 以此来计算梯度。由于一般难以获得目标函数的解析表达式或其梯度, 所以 LMS 算法采用梯度的估计值, 如公式 7.2、7.4 所示。在这种情况下, 将 LMS 算法视作单点搜索策略就不是十分恰当的了。而且计算该梯度的估计值可以被认为是  $n$  个体呈标准正交分布且相互之间距离无限小的一个种群。

梯度的估计需要  $n$  个独立的信息, 这将导致下面关于局部逃逸能力的讨论。当我们从给定的目标函数直接计算梯度时, 步长最后将趋于无穷小, 即  $\Delta x \rightarrow 0$ , 这意味着该算法最后将被局部极值点点牢牢吸引, 与此相对应地是, 在 ES 中, 由于步长  $\sigma$  的存在, 父代和子代之间总是存在着一定的距离。然而, 由于 LMS 算法采用的是梯度的估计值, 该值的计算需要  $n$  次独立取值, 且沿各坐标轴有  $\Delta x$  的距离。考虑到这一点, ES 和 LMS 的局部逃逸能力就没有什么差异了。

## 7.4 本章小结

本章首先将  $n$  维搜索空间映射为 2 维空间, 目标函数选为二次型函数, 对其进行寻优, 发现 ES 是沿梯度路径进行搜索的。而任意抽取两个个体, 却发现他们的搜索路径是“混乱”的, 即 ES 微观层次上“杂乱无章”的行为, 在宏观上却体现出确定的行为。当 ES 引入全局中间交叉算子时, 需要生成一个虚拟的父代个体, 这也可以看作是一种单点搜索; LMS 算法要求至少  $n$  个不同方向的信息, 以此来估计梯度, 这也可以看作一种多点搜索策略。

总之, 全局中间交叉算子的 ES 是否被认为是一种梯度算法, 或者是一种多点随机搜索算法, 很大程度上依赖于我们如何看待它。

## 第八章 全文总结

进化算法是一种基于自然选择和遗传变异等生物进化机制的全局性概率搜索算法。与基于导数的解析方法和其他启发式搜索方法（如模拟退火方法，单纯形法等）一样，进化算法在形式上也是一种迭代方法。它从选定的初始解出发，通过不断迭代逐步改进当前解，直至最后搜索到最优解或满意解。在进化计算中，迭代计算过程采用了模拟生物体的进化机制，从一组解（群体）出发，采用类似于自然选择和有性繁殖的方式，在继承原有优良基因的基础上，生成具有更好性能指标的下一代解的群体。进化论发展到今天，产生了许多新的观点和不同的流派。例如，现在的主流学派越来越倾向于认为，进化只是生物群体的遗传组成或其比例的不可逆的改变，而并不必然伴随着进步、改善，或复杂化，但进化必定是趋向适应的；接受达尔文的自然选择，抛弃拉马克的用进废退；接受古尔德渐进与跃进交替的“间断平衡论”，而非达尔文所强调的那种连续渐进的进化过程。此外还有木村的“中性进化假说”等等。进化计算是以进化论为生物学背景，将生物进化过程简化并模拟的一大类优化算法，在多个领域中取得了成功。进化论的新发展，必然为我们重新认识和改造进化计算提供了新的角度。

基于上述考虑，本文分析了基于拉马克主义的进化学习策略（Lamarckian Evolution Learning Strategy，简称 LEELS）和基于达尔文主义的进化学习策略（Darwinian Evolution Learning Strategy，简称 DELS）的异同，利用马尔可夫链理论证明了此类算法的收敛性，在理论上分析了 DELS 具有更强的局部极值逃逸能力；然后针对 8 个常用的测试函数给出了仿真试验的结果。

第三章阐述作者对基于均匀分布变异算子的进化策略的研究结果，探讨了均匀分布变异算子的作用，并证明了此类算法的收敛性。

第四章在上述两章的基础上，提出基于均匀变异算子的进化学习策略，并对收敛性和局部逃逸能力进行了理论分析，最后的仿真实验表明了该方法的有效性。

第五章首先给出最简约模糊划分的定义；提出了模糊划分的完备—清晰性的概念，并给出计算公式；进一步讨论了模糊规则的完备性—紧凑性；以及模糊规则的一致性。在此基础上，应用进化策略，对模糊系统进行寻优，使其获得较好系统响应性能和较高可解释性。针对一个典型非线性对象和经典的 Box-Jenkins 煤气炉作为仿真实验的对象，验证了该方法的有效性。

第六章针对 *K-means* 聚类算法易陷入局部极小以及  $K$  值选取的问题，提出一类基于进化策略的聚类算法，可以有效的搜索最优聚类中心和聚类个数  $K$ ；还提出了确定  $K$  值范围的经验公式，以减小搜索空间，提高搜索效率，并给出理论分析。相对遗传算法而言，本方法编码简单，种群较小。对 Fisher's iris 数据集的仿真实验表明，该方法得到最优解的可能性比经典算法大得多。

第七章探讨了引入全局交叉算子的进化策略与 LMS 算法的异同。LMS 算法是一种基于梯度法的单点搜索策略。而进化策略是一种受生物进化论启发而建立的优化算法，传统的观点一般认为它是一种全局随机多点搜索技术，与 LMS 有着完全不同的特点，是两种截然不同的优化技术。本章对此进行了深入的探讨，认为这两者之间的差异更多的是基于我们平时的一些“想当然”的观念上的，进化策略可以看作是一种“概率”意义上的梯度搜索技术，而 LMS 算法也可以看作一种多点搜索技术。

当然，本文存在许多需要继续研究的问题，如：进化学习策略收敛速度的分析等等，这些工作留待以后去完成。

## 主要参考文献

- [1] Holland, J. H. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, 1962, 9(3): 297~314
- [2] Fogel, L. J., Owens, A. J., and Walsh, M. J. *Artificial intelligence through simulated evolution*. New York: John Wiley & Sons, 1966
- [3] Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. 1<sup>st</sup> edition, Ann Arbor, MI: The University of Michigan Press, 1975; 2<sup>nd</sup> edition, Cambridge, MA: MIT Press, 1992
- [4] Goldberg, D. E. *Genetic algorithms in search, optimization and machine learning*. MA: Addison-Wesley Publishing Company, 1989
- [5] Whitley, L. D. (ed.) *Foundations of genetic algorithms 2*. San Mateo, CA: Morgan Kaufmann, 1993
- [6] Whitley, L. D., and Vose, M. D. (eds.) *Foundations of genetic algorithms 3*. San Mateo, CA: Morgan Kaufmann, 1995
- [7] Melanie Mitchell. *An introduction to genetic algorithms*. Cambridge, MA: The MIT Press, 1996
- [8] 阎平凡, 张长水。神经网络与模拟进化计算[M]。北京: 清华大学出版社, 2000年11月。
- [9] 李敏强, 寇纪淞, 林丹, 李书全, 《遗传算法的基本理论与应用》, 科学出版社, 2002。
- [10] Cavicchio, D. J. *Adaptive search using simulated evolution*(Ph. D Dissertation). University of Michigan, Ann Arbor, 1970.
- [11] Hoffmeister F., and Bäck T. Genetic algorithms and evolution strategies: similarities and differences. In *Proceedings of the First International Conference on Parallel Problem Solving from Nature (PPNS-1)*, Schwefel, H. -P. and Männer, R. (eds.), Berlin: Springer-Verlag, *Lecture Notes in Computer Science*, Vol. 496, 1991:455~470.

- [12] Fogel, D. B. *Evolutionary computation: toward a new philosophy of machine intelligence*. IEEE Neural Networks Council. New York: IEEE Press, 1995
- [13] Miettinen, K., Neittaanmaki, P., and Periaux J. (eds.). *Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, evolution strategies, evolutionary programming*. New York: John Wiley & Sons, June 1999.
- [14] Rechenberg, I. *Evolutionstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Stuttgart: Frommann-Holzboog Verlag, 1973.
- [15] J. D. Bagley, *The Behavior of Adaptive Systems Which Employ Genetic and Correlation Algorithms*. Dissertation Abstracts International. 1967, 28(12), 5106B(Univ. Microfilms No.68~7556).
- [16] J. H. Holland, *Adaptation in Nature and Artificial System*. Cambridge, MA: MIT Press, 1975.
- [17] J. Heitkotter and D. Beasley, *The Hitch-hiker's Guide to Evolutionary Computation*. FAQ in Comp. AI Genetic, 1995.
- [18] L. J. Fogel, A. J. Owens, and M. T. Walsh, *Artificial Intelligence Through Simulated Evolution*. New York: Wiley, 1966.
- [19] I. Rechenberg, "Cybernetic solution path of an experimental problem," Royal Aircraft Establishment, Library Translation no. 1122, 1965.
- [20] Goldberg, D. E., and Segrest, P. Finite Markov chain analysis of algorithm. In *Proceedings of the Second International Conference on Genetic Algorithm (ICGA 2)*, Grefenstette, J. J. (ed.), Hillsdale, NJ: Lawrence Erlbaum Associates, 1987:1~8
- [21] Eiben, A. E., Arts, F. H., and Van Hee, K. M. Global convergence of genetic algorithm: an Infinite Markov chain analysis. In *Proceedings of the First International Conference on Parallel Problem Solving from Nature (PPNS-1)*, Schwefel, H. -P. and Männer, R. (eds.), Berlin: Springer-Verlag, Lecture Notes in Computer Science, Vol. 496, 1991:4~12.
- [22] Rudolph, G. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 1994, 5(1): 86~101.
- [23] Qi, X. F. and Palmieri, F. Theoretical analysis of evolution algorithms with an



- infinite population size in continuous space, Part 1, 2: basic properties of selection and mutation. *IEEE Transactions on Neural Network*, 1994, 5(1): 102~119, 120~129.
- [24] Fogel, D. B. Asymptotic convergence properties of genetic algorithms and evolutionary programming: analysis and experiments. *Cybernetics and systems*, 1994, 25(3): 389~407
- [25] Fogel, D. B. *Evolutionary computation: the fossil record*. Piscataway, NJ: IEEE Press, 1998.
- [26] Suzuki, J. A markov chain analysis on simple genetic algorithms. *IEEE Transactions on System, Man, and Cybernetics*, 1995, 25(4): 650~659.
- [27] 李书全、寇纪淞、李敏强。遗传算法的随机泛函分析。系统工程学报, 1998, 13(1): 97~101.
- [28] Cerf, R. Asymptotic convergence of genetic algorithms. Doctorial Dissertation, 1993.
- [29] Bäck, T., and Schwefel, H. -P. Evolution strategies I: variants and their computational implementation. In winter, G., P' eriaux, J., Gal' an, M., and Cuesta, P. (eds.), *Genetic algorithms in engineering and computer science*. Chichester: John Wiley & Sons, 1995: 111~126.
- [30] Mühlenbein, H., How genetic algorithms really work. I: mutation and hillclimbing. In *Parallel Problem Solving from Nature, (PPSN 2)*, Männer, R., and Manderick, B. (eds.), North-Holland, Elsevier Science Publishers, Amsterdam, 1992: 15~25.
- [31] 恽为民, 席裕庚。遗传算法的全局收敛性和计算效率分析。控制理论与应用, 1996, 13 (4) : 454~460
- [32] Davis, L., (ed.). *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York, 1991.
- [33] Grefenstette, J. J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on System, Man, and Cybernetics*, 1986, 16(1): 122~128.
- [34] Louis, S., and Rawlins, G. J. E. Syntactic analysis of convergence in genetic algorithms. *Foundations of genetic algorithms 2*. San Mateo, CA: Morgan Kaufmann, 1993

- [35] Spears, W. M., and De Jong, K. A. Dining with Gas: operator lunch theorem. Foundations of genetic algorithms 4. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1997.
- [36] Goldberg, D. E. Optimizing Global-Local Search Hybrids. In Proceedings of the Genetic and Evolutionary Computation Conference 1999(GECCO-99), Banzhaf, W., et al (eds.), San Francisco, CA: Morgan Kaufmann, 1999.
- [37] Eshelman, L. J., et al. Preventing premature convergence in genetic algorithms by preventing incest. In Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA 4), Belew, R. K., and Booker, L. B. (eds.), San Mateo, CA: Morgan Kaufmann Publishers, 1991.
- [38] Syswerda, G. Uniform crossover in genetic algorithms. In Proceedings of the Third International Conference on Genetic Algorithms (ICGA 3), Schaffer, J. D. (ed.), San Mateo, CA: Morgan Kaufmann Publisher, 1989: 2~9.
- [39] Eshelman, L. J., et al. Biases in the crossover landscape. In Proceedings of the Third International Conference on Genetic Algorithms (ICGA 3), Schaffer, J. D. (ed.), San Mateo, CA: Morgan Kaufmann Publisher, 1989.
- [40] Goldberg, D. E. Sizing populations for serial and parallel genetic algorithms. In Proceedings of the Third International Conference on Genetic Algorithms (ICGA 3), Schaffer, J. D. (ed.), San Mateo, CA: Morgan Kaufmann Publisher, 1989: 70~79.
- [41] Goldberg, D. E., and Richardson, J. Genetic algorithms with sharing for multi-model function optimization. In Proceedings of the Second International Conference on Genetic Algorithms (ICGA 2), Grefenstette, J. J. (ed.), Hillsdale, NJ: Lawrence Erlbaum Associates, 1987: 41~49.
- [42] Goldberg, D. E., and Rudnick, M. Genetic algorithms and the variance of fitness. Complex Systems, 1991, 5: 265~278.
- [43] Schaffer, J. D., Eshelman, L. J., and Offutt, D. Spurious correlations and premature convergence. In Foundations of Genetic Algorithms, Rowlins, G. (Ed), San Mateo, CA: Morgan Kaufmann, 1991: 102~112.
- [44] Feller, W. An introduction to probability theory and its applications, vol. 2, New

York: John Wiley & Sons Inc., 1971.

- [45] Goldberg, D. E., Deb, K., and Clark, J. H. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 1992, 6: 333~362.
- [46] Miller, B., and Goldberg, D. E. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary computation*, 1996, 4(2): 25~49.
- [47] Harik, G. R., Cantu-Paz, E., Goldberg, D. E. and Miller, B. L. The gambler's ruin problem, genetic algorithms, and the sizing of populations. In *Proceedings of the 1997 IEEE Conference on Evolutionary Computation*, Bäck T. (ed.), New York: IEEE Press, 1997: 7~12.
- [48] Goldberg, D. E., and Deb, K. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, Belew, R. K., and Booker, L. B. (eds.), San Mateo, CA: Morgan Kaufmann Publishers, 1991: 69~93.
- [49] Baker, J. E. Reducing bias and inefficiency in the selection algorithms. In *proceedings of the First International Conference on Genetic Algorithms (ICGA 1)*, Grefenstette, J. J. (ed.), Hillsdale, NJ: Lawrence Erlbaum Associates, 1985: 101~111.
- [50] Baker, J. E. Reducing bias and inefficiency in the selection algorithms. In *proceedings of the*
- [51] Harik, G. Finding multimodal solutions using restricted tournament selection. In *proceedings of the sixth International Conference on Genetic Algorithms (ICGA 6)*, Eshelman, L. J. (ed.), San Francisco, CA: Morgan Kaufmann, 1995: 24~31.
- [52] English, T. Evaluating evolutionary operators. In *proceedings of the Fifth Annual Conference on Evolutionary Computation*, Fogel, L. J., Angeline, P. J., and Bäck, T. (eds.), Cambridge, MA: The MIT Press, 1996, 421~440.
- [53] Culberson, J. C. On the futility of blind search. *Journal of Evolutionary Computation*, 1998, 6(2): 109~128.
- [54] De Jong, K. A., and Spears, W. M. A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 1992, 5(1): 1~26.

- [55] Spears, W. M. The role of mutation and recombination in evolutionary algorithms. PH. D. thesis, George Mason University, Fairfax, VA, 1998.
- [56] Spears, W. M. and De Jong, K a. On the virtues of parameterized uniform crossover. In proceedings of the Fourth International Conference on Genetic Algorithms (ICGA 4), Belew, R. K., and Booker, L. B. (eds.), San Mateo, CA: Morgan Kaufmann Publishers, 1991: 230~236.
- [57] Bäck, T., Hoffmeister, F., and Schwefel, H-P. A survey of evolution strategies. In proceedings of the Fourth International Conference on Genetic Algorithms (ICGA 4), Belew, R. K., and Booker, L. B. (eds.), San Mateo, CA: Morgan Kaufmann Publishers, 1991: 2~9.
- [58] Schwefel, H-P. Numerical optimization of computer models. New York: John Wiley & Sons, 1981.
- [59] Hesser, J., and Manner, R. Towards an optimal mutation probability for genetic algorithms. In proceedings of the first workshop on Parallel Problem Solving From Nature (PPSN 1), Schwefel, H. -P., and Manner, R. (eds.), Heidelberg Berlin: Springer- Verlag, 1990: 23~32.
- [60] Atmar, J. W. The philosophical errors that plague both evolutionary theory and simulated evolutionary programming. In proceedings of the first Annual Conference on Evolutionary Programming, Fogel, D. B., and Atmar, J. W. (eds.), San Diego, CA: Evolutionary Programming Society, 1992: 27~34.
- [61] Belew, R. K., and Vose, M. D. Foundations of genetic algorithms 4. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1997.
- [62] Davis, L. Adapting operator probabilities in genetic algorithms. In Proceeding of the Third International Conference on Genetic Algorithms (ICGA 3), Schaffer, J. D. (ed.), San Mateo, CA: Morgan Kaufmann Publishers, 1989: 61~69.
- [63] Lobo, F. G., and Goldberg, D. E. Decision making in a hybrid genetic algorithm. In Proceeding of the 1997 IEEE Conference on Evolutionary Computation, New York: IEEE Press, 1997: 121~125.
- [64] Shaefer, C. G. The ARGOT strategy: adaptive representation genetic optimizer

- technique. In *Proceeding of the Second International Conference on Genetic Algorithms (ICGA 2)*, Grefenstette, J. J. (ed.), Hillsdale, NJ: Lawrence Erlbaum Associates, 1987: 50~55.
- [65] Ibaraki, T. Combinations with other optimization methods. In Bäck, T., Fogel, D. B., and Michalewicz, Z. (eds.), *Handbook of Evolutionary Computation (Section D3)*. New York: Oxford University Press, 1997.
- [66] Whitley, D. Modeling hybrid genetic algorithms. In Winter, G., P'eriaux, J., Gal' an, M., and Cuesta, P. (eds.) *Genetic algorithms in engineering and computer science*. Chi Chester: John Wiley & Sons. 1995: 191~201.
- [67] Schaffer, J. D., Whitley L. D., and Eshelman, L. J. Combinations of genetic algorithms and neural networks: a survey of the state of the art. In *Proceeding of the International Workshop on Combinations of Genetic Algorithms and Neural Networks*, Baltimore, June, 1992: 1~37.
- [68] Kadaba, N., Nygard, K. E., and Juell, P. J. Integration of adaptive machine learning and knowledge-based systems for routing and scheduling applications. *Expert Systems with Applications*, Vol 2, No.1, 1991: 15~27.
- [69] Attar, A. A hybrid GA-heuristic search strategy. *AI Expert*, Vol. 9, No. 9, 1994:34~37.
- [70] Miller, B. L., and Shaw, M. J. Genetic Algorithms with dynamic niche sharing for multi- modal function optimization. In *IEEE International Conference on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1996: 786~791.
- [71] De Jong, K. A. Genetic Algorithms: a 25-year perspective. In *Proceedings of the fifth International Conference on Genetic Algorithms (ICGA 5)*, Whitley, L. D. (ed.), San Mateo, CA: Morgan Kaufmann, 1993: 169~177.
- [72] Mahfound, S. W. Crowding and pre-selection revisited. In *Parallel Problem Solving from Nature (PPSN 2)*, Manner, R., and Manderick, B. (eds.), North-Holland, Elsevier Science Publishers, Amsterdam, 1992.
- [73] Mengshoel, O. J., and Goldberg, D. E. Probabilistic crowding: deterministic crowding with probabilistic replacement. In *Proceedings of the Genetic and*

- Evolutionary Computation Conference 1999(GECCO-99), Banzhaf, W., et al (eds.), San Francisco, CA: Morgan Kaufmann, 1999.
- [74] Goldberg, D. E., and Wang, L. Adaptive niching via co-evolutionary sharing. In Genetic Algorithms in Engineering and Computation, 1993, 1 (2): 101~125.
- [75] Michalewicz, Z., Janikow, C., and Krawczyk, J. A modified genetic algorithm for optimal control problems.
- [76] Michalewicz, Z., Krawczyk, J., Kazemi, M., and Janikow, C. Genetic algorithms and optimal control problems. In Proceedings of 29th IEEE Conference on Decision and Control, 1990: 1664~1666.
- [77] Qi, X. F., and Palmieri, F. Adaptive mutation in the genetic algorithm. In the Proceeding of the Second Annual Conference on Evolutionary Programming. Fogel, D. B., Atmar, W., La Jolla(eds.), CA: Evolutionary Programming Society, 1993: 192~196.
- [78] Vose, M. D. Generalizing the notion of schema in genetic algorithms. Artificial Intelligence, 1991,50: 385~396.
- [79] Wolpert, D. H., and Macready, W. G. No free lunch theorems for optimization. IEEE Transaction on Evolutionary Computation, 1997, 1(1): 67~82.
- [80] Michalewicz, Z. Genetic algorithms: an overview. Machine Learning, 1988, 3: 121~138.
- [81] Stender, J. (Ed), Parallel genetic algorithms: theory and application. Amsterdam: IOS Press, 1993.
- [82] 陈国良、王煦法、庄镇泉等编著。遗传算法及其应用，北京：人民邮电出版社，1996。
- [83] Cantu-Paz, E. A survey parallel genetic algorithms. Calculateurs Paralleles. Vol. 10, No. 2. Paris: Hermes, 1998.
- [84] 杨启文，《计算智能及其工程应用》，浙江大学博士学位论文，2001。
- [85] 方舟子，《进化新解说》，香港天地图书出版公司，1997年。
- [86] 庚镇城，“Evolution”一词译成“进化”及其传入中国的经过，科技导报（北京），2000(12)。

- [87] Fogel, D.B., and Atmar, J.W. Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems. *Biological Cybernetics* 63,1990: 111~114
- [88] Schaffer, J. D., Eshelman, L. J. On crossover as an evolutionarily viable strategy. In *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA 4)*, Belew, R. K., and Booker, L. B. (eds.), San Mateo, CA: Morgan Kaufmann Publishers, 1991.
- [89] Fogel, D.B., and Stayton, L. C. On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems*, 1994, 32(3): 171~182.
- [90] Spears, W. M. Crossover or mutation? In: *Foundations of Genetic Algorithms-2*. Whitley L. D. (ed.) Morgan Kaufmann Publishers, Inc., 1993.
- [91] John Yen, Bogju Lee, A Simplex Genetic Algorithm Hybrid. *Evolutionary Computation*, IEEE International Conference on, 13-16 April 1997: 175 – 180.
- [92] 邓建军, 徐立鸿, 吴启迪, 一种求解非线性函数优化问题的混和遗传算法。同济大学学报, 第 29 卷第 11 期, 2001 年 11 月。
- [93] Seront, G., Bersini, H., Simplex GA and hybrid methods, *Evolutionary Computation*, 1996., *Proceedings of IEEE International Conference on*, 05/20/1996 - 05/22/1996: 845 – 848.
- [94] Mashhadi, H.R.; Shanechi, H.M.; Lucas, C.; A new genetic algorithm with Lamarckian individual learning for generation scheduling, *Power Systems*, IEEE Transactions on, Volume: 18, Issue: 3, Aug. 2003: 1181 – 1186.
- [95] 向丽, 顾培亮, 一种快速收敛的混和遗传算法。控制与决策, 第 17 卷第 1 期, 2002 年 1 月。
- [96] Jian Xun Peng, Thompson, S., Kang Li; A gradient-guided niching method in genetic algorithm for solving continuous optimization problems, *Intelligent Control and Automation*, 2002. *Proceedings of the 4th World Congress on*, 10-14 June 2002: 3333 – 3338, vol.4.
- [97] Kazarlis, S.A., Papadakis, S.E., Theocharis, J.B., Petridis, V.; Microgenetic algorithms as generalized hill-climbing operators for GA optimization,

- Evolutionary Computation, IEEE Transactions on, June 2001: 204 – 217, Volume: 5, Issue: 3.
- [98] Magyar, G., Johnsson, M., Nevalainen, O., An adaptive hybrid genetic algorithm for the three-matching problem, Evolutionary Computation, IEEE Transactions on, July 2000: 135 – 146, Volume: 4, Issue: 2
- [98] Magyar, G., Johnsson, M., Nevalainen, O., An adaptive hybrid genetic algorithm for the three-matching problem, Evolutionary Computation, IEEE Transactions on, July 2000: 135 – 146, Volume: 4, Issue: 2
- [99] Yanling, Jiang jing-ping, Local Escaping Capacity of Darwinian Evolution Learning Strategy, 2005 ACC accepted.
- [100] 阎岭, 蒋静坪, 进化学习策略收敛性和逃逸能力的研究, 自动化学报, 2004 年 10 月录用。
- [101] 李博主编, 生态学[M]. 北京, 高等教育出版社, 2000, 76~83。
- [102] 杨震, 放发良, 一种新型的混合遗传算法的研究, 数据采集与处理, 2003 年 12 月, 第 18 卷第 4 期。
- [103] 李智勇, 童调生. 基于多种群进化小生境遗传算法的神经网络进化设计方法研究. 控制与决策. 2003 年 9 月, 第 18 卷第 5 期。
- [104] 木村资生. 分子进化的中性学说[M]. 陈建华译. 成都: 科技大学出版社, 1993.
- [105] 王向军, 林春生, 龚沈光. 基于中性进化、自组织和自然选择的进化算法. 数据采集与处理, 2003 年 12 月, 第 18 卷第 4 期。
- [106] Yaochu, Jin. Knowledge in Evolutionary Learning Systems [M], Shaker Verlag, Aachen 2002.
- [107] Back, T., Fogel D. B. and Michalewicz, T. (eds.), Evolutionary Computation 1: Basic Algorithms and Operators. Britol: Institute of Physics Publishing, 2000.
- [108] Aemeyer, T., and Ebeling, W. Unified description of evolutionary strategies over continuous parameter space. BioSystems, 1997, 41(3): 167~178.
- [109] 原抓纲, 蒋静坪, “基于 T-S 模糊模型的电力传动系统的非线性 PD 状态反馈控制器”【英文】, 中国电机工程学报, 24(12): 198-206, 2004.
- [110] Fogel, D. B. (Editor), IEEE Transactions on Neural Networks. Special issue on



- Evolutionary Computation. Vol. 5. No. 1. 1994.
- [111] Herdy, M., Application of the Evolution Strategy to Discrete Optimization Problems. In proceedings of the first workshop on Parallel Problem Solving From Nature (PPSN 1), Schwefel, H. -P., and Manner, R. (eds.), Heidelberg Berlin: Springer- Verlag, 1990: 23~32. pp: 188-192.
- [112] Tang, W.K.S.; Wong, E.W.M.; Chan, S.; Ko, K. -T.; Optimal video placement scheme for batching VOD services, Broadcasting, IEEE Transactions on, Volume: 50, Issue: 1, March 2004.
- [113] Liu, X.; Lee, B.; Optimal Design of Fiber Raman Amplifier Based on Hybrid Genetic Algorithm, Photonics Technology Letters, IEEE, Volume: 16, Issue: 2, Feb. 2004.
- [114] Balakrishnan, P.V.; Gupta, R.; Jacob, V.S.; Development of Hybrid Genetic Algorithms for Product Line Designs, Systems, Man and Cybernetics, Part B, IEEE Transactions on, Volume: 34, Issue: 1, Feb. 2004.
- [115] Beyer, G. -H., and Fogel, D. B. A Note on the Escape Probabilities for Two Alternative Methods of Selection under Gaussian Mutation. In Proceedings of the sixth Annual Conference on Evolutionary Programming, Angeliene, P. J. (ed.), Berlin: Springer-Verlag, 1997: 264~274.
- [116] 刘志俭 等, 《matlab 应用程序接口用户指南》, 科学出版社, 2000。
- [117] Yao, X., and Liu, Y. Fast Evolutionary Programming. In Proceedings of the Fifth International Conference on Evolutionary Computation, Fogel, L. J., Angeline, P. J., and Bäck, T. (eds.), Cambridge, MA: The MIT Press, 1996: 441~450.
- [118] Shaffer, J. D., and Morishima, A. An adaptive crossover distribution mechanism for genetic algorithms. In Proceedings of the Second International Conference on Genetic Algorithm (ICGA 2), Grefenstette, J. J. (ed.), Hillsdale, NJ: Lawrence Erlbaum Associates, 1987:36~40
- [119] Gehlhaar, D. K., and Fogel, D. B. Two new mutation operators for enhanced search optimization in evolutionary programming. In the Proceedings of SPIE, Bosacchi, B., Bezdek, J. C., and Fogel, D. B. (eds.), 1997: 260~269.
- [120] [美]钟开荣. 概率论教程. 刘文译. 上海科学技术出版社, 1989.
- [121] 王立新. 模糊系统挑战与机遇并存——十年研究之感悟[J]. 自动化学报. 2001, 27(4): 585-590.

- [122] Lin, C and C. Lee. Real-time supervised structure/ parameter learning for fuzzy neural networks [A]. In Proceedings of IEEE International Conference on Fuzzy System [C], San Diego, CA, Mar. 1992: 1283-1290.
- [123] Koczy, L. and K. Hirota. Size reduction by interpolation in fuzzy rules [J]. IEEE Transactions on Systems, Man and Cybernetics, 1997, 27(1): 14-25.
- [124] Zhang, Q and J. G. amd M. Kreutz. Optimization of a fuzzy controller for a diver assistant system [A]. In Proceedings of the Fuzzy-Neuro Systems [C], Nieuwe Hemweg, Amsterdam, 1998: 376-382.
- [125] 诸静 等.模糊控制原理与应用[M].北京: 机械工业出版社, 1995.
- [126] Jin, Y. Fuzzy modeling of high-dimensional system: Complexity reduction and interpretability improvement [J]. IEEE Transactions on Fuzzy Systems, 2000, 8(2): 212-221.
- [127] Jin, Y., W. von Seelen, and B. Sendhoff. On generating FC3 fuzzy rule systems from data using evolution strategies [J]. IEEE Transactions on Systems, Man and Cybernetics, 1999, 29(6): 829-845.
- [128] Karr, C. Applying genetics to fuzzy logic [J]. IEEE AI Expert 1992,6(2): 26-33.
- [129] Wienholt, W. Improving a fuzzy inference system by means of evolution strategy [A]. In B. Reusch, Hrsg., Fuzzy Logic - Theorie und Praxis [C], Springer, Berlin, 1994: 186-195.
- [130] Bäck, T. Parallel optimization of evolution algorithms [A]. In Parallel Problem Solving from Nature III [C]. Springer, Berlin, 1994: 418-427.
- [131] 张乃尧, 阎平凡.神经网络与模糊控制[M]. 北京: 清华大学出版社, 1998.
- [132] Stolpmann, A. and Dooley, L.S. About the use of fuzzy clustering for texture classification, Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American, 10-12 June 1999.
- [133] Elkan, C. The paradoxical success of fuzzy logic. In Proceedings of the Eleventh National Conference on Artificial Intelligence, 1993: 196- 203.
- [134] Jin, Y., J. Jiang, and J. Zhu. Neural network based fuzzy identification with application to modeling and control of complex systems. IEEE Transactions on Systems, Man, and Cybernetics 25(6), 1995: 990-997.
- [135] 刘健庄, 谢维信, 黄建军 等, 聚类分析的遗传算法方法, 电子学报, 第 23 卷, 第 11 期, 1995 年 11 月, 81-83。
- [136] B.Widrow, and M.E.HOFF." Adaptive switching circuits," in IRE WESCON Conv. Rec., Pt.4, 1960,pp.96-104.

- [137] B.Widrow, "Generalization and information storage in networks of Adaline 'neurons'," in M.C.Yovitz, G.Goldstein, (eds.), *Self-Organizing System* 1962(Washington, DC: Spartan Books, 1962), pp.435-461.
- [138] B.Widrow, and M.A.LEHR,"30years of adaptive neural networks: Perceptron, Madaline and backpropagation," *Proc. IEEE* Vol.78, No.9 (September 1990). Pp.1415-1441.
- [139] B.Widrow, E. Walach 著, 刘树棠, 韩崇昭 译, 自适应逆控制, 西安交大出版社, 2000 年 5 月。
- [140] Rechenberg, *Evolutionssstrategie'94*. Stuttgart: Frommann-Holzboog, 1994.
- [141] Ralf Salomon, Evolutionary algorithms and gradient search similarities and differences. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 2, NO. 2, JULY 1998.
- [142] H. -G. Beyer, "On the 'explorative' power of ES/EP-like algorithms," in *Proc. 7th Annual Conf. Evolutionary Programming EP'98*, V. W. Porto, N. Saravanan, D. E. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer-Verlag, 1998.
- [143] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1-23, 1993.
- [144] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm I," *Evol. Comput.*, vol. 1, no. 1, pp. 25-50, 1993.
- [145] H.-G. Beyer, "An alternative explanation for the manner in which genetic algorithms operate," *BioSystems*, vol. 41, pp. 1-15, 1997.

## 附录 本文采用的10个标准测试函数

测试函数	最值
$F_1 = \sum_{i=1}^2 x_i^2, (-5.12 < x_i < 5.12)$	0
$F_2 = \sum_{i=1}^{M-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2], (-5.12 < x < 5.12, i = 1, 2, \dots, M)$	0
$F_3 = \left\{ \sum_{i=1}^5 i \cos[(i+1)x + i] \right\} \left\{ \sum_{i=1}^5 i \cos[(i+1)y + i] \right\}, (-10 < x, y < 10)$	186.7309
$F_4 = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2, (-100 < x, y < 100)$	1.0316
$F_5 = -\sum_{i=1}^M x_i \sin(\sqrt{ x_i }), (-500 < x_i < 500, i = 1, 2, \dots, M)$	837.966
$F_6 = (x_1^2 + x_2^2)^{0.25} [\sin^2(50 \times (x_1^2 + x_2^2)^{0.1}) + 1.0], -10 \leq x_1, x_2 \leq 10$	0
$F_7 = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, -10 \leq x_1, x_2 \leq 10$	0.999
$F_8 = MA + \sum_{i=1}^M (x_i^2 - A \cos(2\pi x_i)), (-5.12 < x < 5.12, i = 1, 2, \dots, M)$	0
$F_9 = x \times \sin(\sqrt{ y+1-x }) \times \cos(\sqrt{ y+x+1 }),$ $+ (y+1) \times \cos(\sqrt{ y+1-x }) \times \sin(\sqrt{ y+x+1 })$ $x, y \in [-512, 512]$	-511.7
$F_{10} = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2),$ $(-3.0 < x < 12.1, 4.1 < x < 5.8)$	38.8503

## 攻读博士学位期间发表的主要学术论文

### A: 在国内外核心学术刊物上发表和录用的论文

1. 第一作者, 进化学习策略收敛性和逃逸能力的研究, 自动化学报, 2004 年 10 月录用。
2. 第一作者, 基于进化策略生成可解释性模糊系统, 电子学报 2005 年 1 月。
3. 第一作者, Optimization of Fuzzy Partition Based on Evolution Strategy, 中国电机工程学报, 第 24 卷, 第 9 期, 2004 年 9 月。
4. 第一作者, Convergence and Local Escaping Capacity of Darwinian Evolution Learning Strategy, submitted to IEEE Transactions on Evolutionary Computation.

### B: 在国际学术会议上发表和录用的论文

1. 第一作者, Application of Evolution Strategy in Cluster Analysis, Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P. R. China, 2004: 2197-2199.
2. 第一作者, Local Escaping Capacity of Darwinian Evolution Learning Strategy, 2005 ACC accepted.

## 致 谢

衷心感谢导师蒋静坪教授对本人的精心指导。他的言传身教将使我终生受益。蒋老师严谨的治学态度,渊博的学识以及豁达的人生观,无时无刻不对我产生积极的影响。在我博士课题研究过程中,蒋老师给了我不少启发性的建议,开拓了我的思维。蒋老师在各个方面给予我极大的信任,鼓励我学术上的创新,使得我能够最大限度发挥自己能力来完成课题;同时,蒋老师时时刻刻在生活上对我关心和照顾,使我能够从容克服学业和生活中遇到的困难。这篇论文的顺利完成,凝聚了蒋老师的心血。在此,谨向我的导师蒋静坪教授表示由衷的感谢!

感谢师兄郑洪涛博士,他给了我很多及时的指导和无私的帮助;我还要感谢何衍博士、李金厚博士、丁滢颖博士、原抓纲硕士、曹光植硕士等多位师兄师弟,同他们的交流不仅扩大了我学术上的视野,还为我的求学生活带来了不少乐趣。

在我的同门中,李玉榕博士、乔宾博士、工雷博士、李广海硕士、郑海东硕士、叶勇硕士、刘兴硕士、龙晓林博士、王柱峰博士、董扬斌博士、向旭博士、陈少斌博士等多位师兄师弟在我的学术研究以及博士论文撰写过程中,给予了极大的帮助。在此表示对他(她)们衷心地感谢,并原谅我无法一一列出他(她)们的姓名。

最后,我要深深地感谢我的妻子李璟博士,没有她的理解、鼓励和支持,我不可能完成本文的研究工作。她为支持我的学业作出了巨大的牺牲,这份付出是无以回报的。

感谢所有努力试图听懂我的想法的人们!感谢所有帮助过我的人们!

文章千古事,得失寸心知。

阎 岭

二零零五年六月