中南大学

博士学位论文

进化策略的变异算子与仿真平台研究

姓名：王湘中

申请学位级别：博士

专业：控制理论与控制工程

指导教师：喻寿益

20050901

摘 要

论文研究进化策略中变异算子的改进。现有的变异算子都使用全基因变异，本文提出单基因变异，通过对变异成功概率、局部收敛速度、全局收敛性能、变异步长控制、计算开销、多种群技术系统地分析比较两种变异方式的优劣，建立基于递减型变异步长单基因变异算子的单种群和多种群进化策略，最后论述进化算法仿真试验平台的构建及其应用。

对进化策略及其变异算子的研究源于变异算子在进化计算中起主导作用的认识。通过对简单遗传算法的改进试验，对交叉算子作用机理的分析，证明了变异算子对交叉算子的局部和全局搜索功能的可替代性，变异算子在算法中起主导作用。

借鉴生物进化理论和基因突变思想的进化策略，其变异算子使用所有基因同时变异的全基因变异方式，本文提出了一次只随机选择其中一个基因发生变异的单基因变异方式。理论分析和仿真试验证明，对于多维优化问题，当变异步长较大时，单基因变异的成功概率大于全基因变异的成功概率，全基因变异存在进化停顿现象，而且计算开销较大，对于高维优化问题尤其突出。

理论分析证明使用 Gauss 分布的单基因变异算子时，保持成功概率为 0.445 可以获得最优的局部收敛速度，并提出相应的递减型变异步长控制策略。为了直观分析、比较进化算子的性能，提出了横向仿真技术，并用于单基因变异和全基因变异的局部收敛速度的比较研究。试验证明虽然当变异步长合适时，全基因变异算子的局部收敛速度大于单基因变异，但全基因变异算子要求变异步长较小并且范围很小，而单基因变异算子可以在变异步长较大、且在一个较大的范围内获得良好的局部收敛速度，说明单基因变异算子对变异步长具有良好的鲁棒性。通过两个反例说明 Gauss 分布递减型步长控制单基因变异算子全局搜索能力的不足，提出了 Gauss 分布递减型步长单基因变异与均匀分布变异相结合的改进进化策略 $(\mu+\lambda+\kappa)-ES$，通过一组 100 维典型测试函数的仿真试验，说明了 $(\mu+\lambda+\kappa)-ES$ 良好的局部和全局搜索能力、较少的计算开销。

为了增强 $(\mu+\lambda+\kappa)-ES$ 的全局搜索能力、在解多模态优化问题时

获得多个全局最优解和局部最优解，在$(\mu+\lambda+\kappa)-ES$基础上提出了多种群进化策略$m\times(\mu+\lambda+\kappa)-ES$。在基于排挤和（/或）共享的多种群技术中，共享半径是一个难以确定而又非常关键的参数，本文提出了山谷探索法，避免了共享半径的确定。建立了在给定精度下以一定的可信度判断子种群是否收敛的判据，使子种群的收敛性判别有了实用的定量依据。以一组典型多模态优化问题测试函数仿真验证了$m\times(\mu+\lambda+\kappa)-ES$准确求解全部极值点的能力。

进化算法仿真试验研究与应用平台是进化计算重要工具，本文论述了基于面向对象程序设计技术和 Visual C++设计仿真计算平台的基本思路、基本结构，构建了一个通用的仿真计算平台，通过两个实例——复杂机电系统速度控制参数的优化和足球机器人的最优控制说明了平台和算法的应用。

最后给出了未来的研究方向。

关键词 进化策略，变异算子，变异步长，多种群,收敛判据

ABSTRACT

Improvements of the mutation operator of the evolution strategies (ES) are studied in this paper. All-gene mutation is applied to current mutation operators, single-gene mutation is proposed in the paper. Based on systemic analysis and comparison of the two types of mutation operators in the success probability, local convergence velocity, global convergence performance, mutation step-size control, computation costs, multi-population technique, single-population ES and multi-population ES based on single-gene mutation with descending mutation step-size control strategy are proposed respectively, finally establishment of a platform for evolutionary computation (EC) simulation and the application of the platform are discussed.

The study on the ES and its mutation operators is originated from the knowledge that mutation operators play a dominant role in EC. With the experimentation on an improvement of the simple genetic algorithm (SGA) and analysis on the mechanism of crossover operators, it is proven that the local and global searching functions of crossover operators can be replaced by the mutation operators and the later plays a dominant role in EC.

Originating from evolutionary theories and gene mutation ideas, the mutation operators of the ES take a manner of all-gene mutation, i.e. all the genes of an individual (or a chromosome) are varied at a time, single-gene mutation in which only one randomly selected gene is varied at a time is proposed in the paper. It is proven through theoretical analysis and simulation that single-mutation has a larger success probability than all-gene mutation under large mutation step-size for high-dimensional optimization, that the all-gene mutation suffers a stagnation of evolution and costs more especially for high-dimensional optimization.

It is proven theoretically that when the success probability is about 0.445, the Gaussian distribution single-gene mutation can get the best local searching velocity, and a descending mutation step-size control

strategy is correspondingly proposed. In order to intuitively analyze and compare the performances of different evolutionary operators, a novel method named as transverse simulation is proposed and is applied to the comparative study on the local convergence velocity of the single-gene mutation and all-gene mutation. It is proven through simulation that all-gene mutation under an appropriate mutation step-size and a small varying range has a lager local convergence velocity than single-gene mutation, and that the single-gene mutation can get a promising convergence velocity under a large mutation step-size and in a large varying range, and that the single-gene mutation operator shares an excellent robustness for the mutation step-size. Two counter examples demonstrate that Gaussian-distribution based single-gene mutation operator with descending step-size control has no enough global searching capability, a uniform-distribution mutation operator is introduced and the both operators are combined in ES, so a novel ES named as $(\mu+\lambda+\kappa)-ES$ is proposed. Simulations under a set of typical 100-dimensional benchmark demonstrate that $(\mu+\lambda+\kappa)-ES$ has promising local and global searching capability and costs less.

In order to promote the global searching capability of the $(\mu+\lambda+\kappa)-ES$ and to solve the multi-modal function optimization (MFO) with multiple optima, a novel multi-population ES $m\times(\mu+\lambda+\kappa)-ES$ based on the $(\mu+\lambda+\kappa)-ES$ is proposed. Sharing-radius is a difficult-determining and key parameter in crowding and (/or) sharing multi-population EC, hill-valley searching method is proposed and determination of the sharing-radius is avoided. A criterion is established to determine the convergence of a sub-population under a given accuracy and confidence, which is an applicable quantitative criterion. Simulations on a set of MFOs demonstrate that $m\times(\mu+\lambda+\kappa)-ES$ can properly find all the optima.

A platform for simulation and application is an important tool for EC, the basic idea and fundamental architecture of the platform is described based on the object-oriented programming (OOP) and Visual C++. A platform is established and two actual examples –parameters

optimization of an electromechanical system's speed-control subsystem and optimal control of soccer robots are given to demonstrate the application of the platform and the algorithms proposed in this paper. Finally the future studying topics are given.

KEY WORDS evolution strategies, mutation operator, mutation step-size, multi-population, convergence criterion

原 创 性 声 明

本人声明，所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了论文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中南大学或其他单位的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在在论文中作了明确的说明。

作者签名：[signature]　日期：2005年12月14日

关于学位论文使用授权说明

本人了解中南大学有关保留、使用学位论文的规定，即：学校有权保留学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用复印、缩印或其它手段保存学位论文；学校可根据国家或湖南省有关部门规定送交学位论文。

作者签名：[signature]　导师签名[signature]　日期：2005年12月14日

第一章 绪 论

优化是一个在科学技术与工程领域中应用极为广泛的理论与实践问题，很多问题都可以用优化问题建模或者转化为优化问题，如作者所在课题组承担的国家自然科学基金重点资助项目“复杂机电系统耦合分析与解耦控制”（项目号：59835170）、“智能仿生机电人工腿关键应用基础研究”（项目号：50275150）等都包含优化问题，前者需要实现复杂机电传动速度控制系统参数的优化、解耦控制，后者需要设计基于进化算法的 PID 参数自整定模糊控制器；作者承担的机器人足球比赛项目中也需要对足球机器人进行最优控制、参数辨识等。

随着问题复杂性的增加（如变量维数增多、不连续、不可微、甚至无法用数学表达式描述），传统的基于导数（或梯度）的方法遇到了极大的困难，人们借鉴、学习、模拟自然界的进化现象提出的进化算法，为求解优化问题开辟了新的思路、注入了新的活力，吸引了众多研究者的目光。收敛速度慢、存在早熟收敛等是进化算法需要解决的主要问题。

本文的工作就是在这些背景下展开的，主要研究进化算法（主要是进化策略）的改进、建立进化算法仿真试验研究平台并应用于优化计算。

本章主要包括以下内容：简述进化算法的发展、进化算法的基本框架，综述进化策略的演化过程、研究内容和现状，介绍本文的研究思路和主要研究内容。

1.1 进化算法概述

进化算法的研究起源于人们对自然界生物进化过程的学习、借鉴和模仿。经典的 Darwin 进化理论与 Weismann 选择和 Mendel 遗传学说相结合，构成了已被广泛接受的新 Darwin 主义（Neo-Darwinism）。新 Darwin 主义认为，只用种群和物种内的少量统计过程就可以充分地解释大多数生命历史，这些过程就是繁殖、变异、竞争、选择。繁殖是所有生命的共同特征；变异保证了任何生命系统能连续地繁殖自己；对于限制在有限范围内的不断扩展的种群，竞争和选择不可避免。于是，生物在这四个互相作用的随机过程中一代一代地进化[1][2]。今天，进化思想已不再局限于对生命研究，进化是一个可以通过计算机或者其他方式模拟的优化过程。人们在生产、科学实践中面对大量用传统方法难以求解

的优化问题，借鉴、模仿自然界的进化规律，提出了以遗传算法 GA（Genetic Algorithm）、进化策略 ES（Evolution Strategies）、进化规划 EP（Evolution Programming）等为典型代表的进化算法。

进化算法可以追溯到 20 世纪 30 年代的通过仿真生物进化过程进行机器学习的研究[3]。早在 1932 年，Cannon 就把自然进化想象为一个学习过程，只是当时没有使用种群的概念，而是对一个单一个体反复进行随机试验；1950 年，Turing 认识到机器学习与进化之间存在着明显的关系。Friedman 试图用变异和选择的仿真设计“思想机器”； Cambell 猜想，在导致知识扩张的过程中，都有涉及“盲目－变化－选择－幸存”的过程；Friedberg 作了自动程序设计试验。

Bremermann[4]是现代进化计算的早期研究者之一，早在 1958 年，他就开始了模拟生物进化的优化计算试验，他把生物的进化看作是一个优化的过程。为了使进化过程易于分析，Bremermann 用简单的函数进行了试验，在他的算法试验中，以解线性方程组 $\mathbf{Ax}-\mathbf{b}=\mathbf{0}$（其中 $\mathbf{A}$ 是 $n\times n$ 矩阵，$\mathbf{b}$ 是 $n\times 1$ 列矢量）为研究对象，把其求解过程转换为求函数 $f(\mathbf{x})=\|\mathbf{Ax}-\mathbf{b}\|_2$ 最小值。在他的试验中，每次以固定步长变异一个基因，优化过程很快使适应值陷入某个停滞点；减小步长，优化过程得以改善，然后陷入新的停滞点。Bremermann 猜测，有性交配可以克服进化停滞，通过线性叠加或随机组合，Bremermann 尝试了成对、成组的各种交配方案，但结果甚微，使 Bremermann 相当失望，后人把他的试验称为“Bremermann 失望”（Bremermann's Disappointment）[5]。尽管 Bremermann 的结果让他“失望”，但他的计算方法已经具备了遗传算法的雏形，包含了种群、重组、变异、选择这些基本要素。

现代进化算法的形成与发展要归功于 Holland、Fogel、Rochenberg 和 Schwefel 等人的成果，他们分别为遗传算法、进化规划和进化策略的建立做出了开创性的贡献。

在二十世纪六、七十年代由美国 Michigan 大学 J.H.Holland 教授及其学生和同事创立了遗传算法[6]。二十世纪六十年代，Holland 注意到生物的自然遗传现象与人工自适应系统行为的相似性，他认为不仅要研究自适应的系统，还要研究与系统相关的环境，因此他提出在研究和设计人工自适应系统时，可以借鉴自然界生物遗传的基本原理，模仿生物遗传的基本方法。1967 年，他的学生 J.D.Bagley 在其博士论文中首次提出了“遗传算法（Genetic Algorithm，GA）”这一术语。1975 年，Holland 出版了系统论述遗传算法的第一本专著《Adaptation in Nature and Artificial Systems》[7]，建立了以“模式定理（Schema Theorem）”为基础的遗传算法基本定理，标志着遗传算法的诞生。

几乎在同一时代， L. J. Fogel [8]在人工智能研究中，借鉴自然界生物进化的思想，提出了一种随机优化方法，称之为进化规划（Evolution Programming：EP），并首先成功地应用于有限状态机的进化中。1966 年，Fogel 等出版了《 Artificial intelligence through simulated evolution》 [9]，系统阐述了进化规划的基本原理。

遗传算法和进化规划都诞生于美国，而进化策略则起源于德国。1963 年，当时是柏林工业大学学生的 I. Rechenberg 和 H. P. Schwefel[10][11]在试验处理流体动力学问题，如弯管形状优化试验研究中，考虑到传统的导数优化方法不能适于解决这类问题，提出了按照自然突变和自然选择的生物进化思想，设计了一组弯管形状优化试验规则，并取得了良好的效果，他们把这组规则称之为进化策略，并成功应用于其他实值优化问题。

遗传算法、进化策略、进化规划是不同领域研究人员分别提出的，在各自的领域取得了成功，但直到二十世纪九十年代，这三种算法才有了相互的交流[12]，人们发现这些算法的基本思想都来源于自然界的生物进化思想，具有惊人的相似之处，可谓异曲同工、殊途同归。人们把这类算法统称为进化计算 EC（Evolutionary Computation），相应的算法统称为进化算法 EA（Evolutionary Algorithms），标志着进化算法的诞生。随着《Evolutionary Computation》 (MIT Press)，《IEEE Transactions on Evolutionary Computation》 (IEEE Press)， 《Soft Computing》 (Springer Verlag)， 《Applied Soft Computing》 (Elsevier Science) 和《Evolutionary Optimization—An International Journal on the Internet》等专业学术期刊的相继创刊，一年一度的国际进化计算、机器学习与进化大会等学术活动有力地推动了进化算法的研究与应用，使进化计算成为自 90 年代以来最活跃的研究领域之一。

今天，进化算法的理论研究不断深入、应用日益广泛，已逐步成为集生命科学、系统优化、统计学、人工智能和计算机科学于一体的交叉学科。同时，借鉴、学习自然规律、模仿人脑智能的新算法，如模拟退火、量子搜索算法[13]、微粒子群算法[14]、免疫算法[15]、蚁群算法[16]、心脑计算[17]等等也相继涌现，进化计算具有了更为广泛的内涵。

1.2 进化算法基本框架

进化算法 EA 是模拟自然界的“自然选择，物竞天演”等生物进化机制的一类随机搜索型的计算方法，它与传统的导数类方法不同在于，EA 通过包含若干个体的种群同时维持着一组候选解，每个个体代表了一个候选解，其迭代过程

是一个一代接着一代的繁殖、选择进化过程，新的个体（后代）的产生通过交叉（crossover）或重组（recombination）、变异（mutation）、评价（evaluation）、选择（selection）等遗传算子（genetic Operators）完成，具有一定的随机性。

考虑含有 n 个变量求最大值的优化问题：

$$J=\max(f(\mathbf{x})) \tag{1-1}$$

$\mathbf{x}\in S^n$，$f(\mathbf{x})$ 为适应值函数。进化算法可以由 Algorithm1.1 描述[18][19]，算法中，$P(t)$ 表示在 t 代包含 μ 个个体（individual）或染色体（chromosome）的父种群（parental population），子种群（offspring）$P''(t)$ 包含 λ（$\lambda\geq\mu$）个个体，由 $P(t)$ 经过重组和变异产生，Q 根据选择方式的不同，有不同的含义：当使用 $(\mu+\lambda)$ 选择方式时，$Q=P(t)$，即父代和后代同时竞争，使用 (μ,λ) 选择方式时，$Q=\varnothing$，即父代不参加竞争。

Algorithm 1.1 ： evolutionary algorithm

```
begin
    t = 0;
    initialize P(t);
    evaluate P(t);
    while (not terminate-condition) do
    begin
        P'(t) =recombine P(t);
        P''(t)=mutate P'(t);
        evaluate P''(t);
        P(t+1)=select (P''(t)∪Q);
        t = t+1;
    end;
end;
```

由算法的描述可知，进化算法一般包含以下主要部分：

1、表示：对于每个进化计算方法（EC），存在一个非空集合 S，称为 EC 的个体空间。当将 EC 用于优化搜索时，每个个体（或者染色体）$a\in S$ 代表这个优化问题的候选解，它与目标变量 $\mathbf{x}$ 之间的关系有不同的表示方法，常用的有二进制编码、实数编码、Grey 编码、树结构编码等[19]，

这些编码方式对于不同类型的优化问题各有其优点。本文考虑连续函数的优化问题，使用实数编码方式，这样每个个体的基因对应一个变量x_i，无需编码和解码，编码精度高，表示范围大，便于设计和实现有关进化算子。

2、评价或适应值函数：优化过程中，适应值函数是评价解质量的指标。从生物生存的角度看，它代表了生物个体对环境的适应性，决定了个体在竞争和选择过程中的生存能力。为了控制选择压，一些算法中使用了适应值变换技术，如非负化、线性静态变换、线性动态变换、对数变换、指数变换等等。对于连续函数求最大值优化问题，在一定的选择方式下，用待优化的函数$f(\mathbf{x})$作为适应值函数，简单、直接。

3、交叉（重组）：交叉算子模拟了自然界生物的有性（Sexual）繁殖，同源染色体通过交叉实现基因重组，生成新的个体或物种。T.Back[21]把进化算法中最常用的重组算子归纳为一点交叉、多点交叉、均匀交叉、离散重组、算术重组等等。

4、变异：在标准遗传算法中，变异是作为一个“背景”算子提出的，其作用是为了在种群中引入缺失的基因，以实现解空间的全局搜索，在二进制编码的算法中，变异以小的概率p_m对每个二进制位进行翻转操作。在进化策略中，变异算子是主要的进化算子，在实数编码表示时，常用Gauss变异、Cauchy变异、均匀变异。

5、选择：选择是一个择优的过程，指从父代种群中挑选个体进入下一代的操作，选择的依据是个体的适应值。一般而言，适应值较大的个体（对于求最大值的优化问题，下同）应该具有较大的生存概率，以有更多的机会繁殖后代。选择是生物实现进化的驱动力，在进化计算中，选择是算法实现定向搜索的唯一方法，个体之间的选择概率差称为选择压（selection pressure），强烈地影响着算法的搜索性能，大的选择压可以加速局部收敛速度，但也容易降低种群多样性和全局搜索能力，可能导致早熟收敛（premature convergence），而弱的选择压使收敛速度放慢。常用的选择方法有轮盘赌选择（roulette wheel selection）或比例（proportional）选择、排序（rank-based）选择、联赛（tournament）选择、截断（truncation）选择[22][23]。根据是否确保父代种群中最佳个体进入下一代，把选择方式分为精英保留（elitist reservation）和非精英保留（non-elitist reservation）。

6、终止条件：常用的终止条件判据有两类，其一为资源使用类判据，以最

大计算代数、最大计算时间、适应值函数最多计算次数等为终止条件；其二为收敛性判据，根据适应值、目标变量、策略参数与给定值比较，作为终止条件。

交叉算子、变异算子统称为遗传算子（genetic operator）或进化算子（evolutionary operator）。

1.3 进化策略研究综述

1.3.1 ES 的演变

ES 经历了从 $(1+1)-ES$ 到 $(\mu+1)-ES$，再到 $(\mu+\lambda)-ES$ 和 $(\mu,\lambda)-ES$，$(\mu/\rho+\lambda)-ES$ 和 $(\mu/\rho,\lambda)-ES$ 的演变发展过程。

1.（1+1）—ES

进化策略的研究始于 1964 年，当初主要用于试验处理流体动力学问题，如弯管形状的优化。事实上，最初的进化策略 ES 是一组优化试验规则：

规则 1：Change all variables at a time, mostly slightly and random;

规则 2：If the new set of variables does not diminish the goodness of the device, keep it, otherwise return to the old status.

规则 1 类似于自然界的变异，规则 2 模拟了“适者生存”，与 Darwin 自然选择原理相对应。由于在每次迭代只有一个旧的个体（父个体）产生一个新的个体（子个体），选择发生在这两个体，这种形式的 ES 后来称为两成员 ES（two-membered）或 $(1+1)-ES$。最初的变异是阶梯式的变量调整（stepwise variable adjustments），服从二项分布。H.P. Schwefel（1965）在其学位论文中研究发现进化过程会出现停滞现象，当进化过程停滞在某个极值点时，需要更大粒度的变异，二项分布很难得到这种大粒度的变异，导致了连续变量、Gauss 分布的变异算子产生。

设父代 $\mathbf{x}$ 产生后代 $\mathbf{x}'$，变异算子表示为

$$\mathbf{x}' = \mathbf{x} + \mathbf{z} \tag{1-2}$$

其中，$\mathbf{z} = (z_1, z_2, ..., z_n)$，每个分量 $z_i, i = 1,2,...,n$ 是各自独立的随机数。当使用 Gauss 分布变异时，$z_i = \sigma N(0,1)$ 是均值为 0，标准差为 σ 的正态分布（即 Gauss 分布）随机数，σ 是 Gauss 分布随机变量的标准差（或称变异步长、策略参数、变异强度）。

Rochenberg[10]通过两个差异很大的实数值函数——超球模型（hypersphere）和走廊模型（rectangular corridor model），研究了基于 Gauss 变异的 $(1+1)-ES$，

表明：收敛速度（每一代向极值点移动的期望距离）反比于变量数；合适的变异强度可以得到线性阶的收敛速度；当成功变异的概率约为 1/5 时相应的变异强度是较优的，此即为著名的 1/5 成功准则。

2.（μ＋1）－ES

Rochenberg（1975）提出了第一个多个体的 ES—$(\mu+1)-ES$ 或称稳态 ES（steady-state ES），通过仿真研究得出以下结论：

1）如果以每一代产生的进化而非每次适应值函数计算产生的进化衡量进化速度，交叉可以显著地加速进化；

2）如果把变异步长嵌入进化过程，作为内在（endogenous）策略参数，而非外部的控制，种群可以在进化过程中通过自学习调节进化步长。

尽管 $(\mu+1)-ES$ 没有得到广泛应用，但这是第一个基于种群的 ES，常用于多处理机下的异步并行计算[24]，并引入了交叉算子，把变异步长以内在参数（策略参数）形式作为个体基因的一部分参与进化。

3.（μ＋λ）－ES、（μ，λ）－ES

和其他经典的优化方法一样，ES 的性能很大程度上依赖于算法内部参数，尤其是变异步长的调整。鉴于 $(\mu+1)-ES$ 具有减小步长的内在趋势（尽管不一定合适），Schwefel（1975，1977）[1][25]提出了更进一步的多成员 ES 版本，即 $(\mu+\lambda)-ES$ 和 $(\mu,\lambda)-ES$，算法由 μ 个个体组成父个体种群，产生 λ 个子个体作为子种群。$(\mu+\lambda)-ES$ 表示父代与子代个体同时竞争选择 μ 个作为下一代，属于精英保留型；$(\mu,\lambda)-ES$ 只在子代中选择 μ 个进入下一代，无论其父代个体多么优秀，都被“遗忘”。

4.（μ/ρ＋λ）－ES、（μ/ρ，λ）－ES

具有重组算子的 ES，$\rho\in\{1,2,...,\mu\}$ 表示参加重组的个体数量，$\rho=1$ 表示不重组。ES 的重组有离散重组（discrete combination，也称为优势重组，dominant recombination）和中间重组（intermediate recombination）。$(\mu/\rho+\lambda)-ES$ 由于表示的灵活性，可作为基本 ES 的通用表示方法称为 CES（classical evolution strategies）。

本文主要研究 ES 的变异算子，不考虑重组算子，以“＋”号型的 $(\mu+\lambda)-ES$ 为研究对象，以下将以 $(\mu+\lambda)-ES$ 泛指基本 ES。

1.3.2 基本 ES 描述

对于式(1-1)描述的优化问题，$(\mu/\rho+\lambda)-ES$、$(\mu/\rho,\lambda)-ES$ 的伪代码描述如 Algorithm 1.2 所示。

ES 操作在含有若干个体 $\mathbf{a}$ 的种群 P。每个个体 $\mathbf{a}_k$ 不仅包括优化问题所对应

的目标变量（向量）$\mathbf{x}_k$，一组内在策略参数（endogenous strategy parameters）$\mathbf{s}_k$，也可以把相应的目标函数值（适应值）$F_k = f(\mathbf{x}_k)$作为个体的一部分，但它只表征个体的属性，不能作为基因参加变异、重组等遗传操作，即

$\mathbf{a}_k = (\mathbf{x}_k, \mathbf{s}_k, F_k)$

内在策略参数$\mathbf{s}_k$控制了遗传算子，特别是变异算子的某些统计特性，在自适应进化策略 SA-ES（self-adaptive ES）中，它作为个体的一部分参加进化。

在 ES 的每一代t，μ个父个体$\mathbf{a}_m$组成父种群$P_p^{(t)} = \{\mathbf{a}_m, m = 1,2,...,\mu\}$产生$\lambda$个后代个体$\tilde{\mathbf{a}}_l$组成子种群$P_o^{(t)} = \{\tilde{\mathbf{a}}_l, l = 1,2,...,\lambda\}$。$\mu, \lambda, \rho$称为 ES 的外在策略参数

Algorithm 1.2: $(\mu/\rho,\lambda)-ES$、$(\mu/\rho,\lambda)-ES$

```
begin
    t = 0;
    initialize(P_p^(0) = {a_m = (x_m^(0), s_m^(0), F(x_m^(0))), m = 1,2,...,μ};//父种群初始化
    while(not terminate condition) do      //主循环
    begin
      for l = 0 to λ do                    //产生λ个子个体
      begin
        E_l = marriage(P_p^(t), ρ);        //随机选择ρ个组成家庭
        s_l = s_recombination(E_l);        //策略参数重组
        x_l = x_recombination(E_l);        //目标变量重组
        s̃_l = s_mutation(s_l);             //策略参数变异
        x̃_l = x_mutation(x_l, s̃_l);        //目标变量变异
        F̃_l = F(x̃_l);
      end;
      P_o^(t) = {ã_l = (x̃_l, s̃_l, F̃_l), l = 1,2,...,λ};      //组成子种群
      if (μ/ρ+λ)-ES:  P_p^(t+1) = selection(P_o^(t), P_p^(t), μ);
       //  "+" 竞争选择
      if (μ/ρ,λ)-ES:  P_p^(t+1) = selection(P_o^(t), μ);  //  "," 竞争选择
      t = t+1;
    end;
end;
```

（exogenous strategy parameter），一般在进化过程中保持不变。ρ 为重组过程中个体的数量。

Algorithm 1.2 描述表明，$(\mu/\rho+\lambda)-ES$ 具有进化算法的一般结构，以下主要说明其特有之处。

1．选择

每种进化算法都需要面向目标的选择算子，以指导搜索朝向目标参数空间期望区域，使进化具有方向性。在 ES 中，选择就像动植物育种，只有那些具有所期望的属性（对于求最大值优化问题，具有较大的适应值）的个体得到繁殖的机会。ES 的选择是一个确定性的过程，在选择池中，只有 μ 个最佳的个体能成为下一代的父个体。这种选择方式也称为截断（truncation）或育种（breeding）选择，可表示为

$$P_p^{(t+1)}=\{\mathbf{a}_{1:\gamma},\mathbf{a}_{2:\gamma},...,\mathbf{a}_{\mu:\gamma}\} \tag{1-3}$$

其中 $\mathbf{a}_{m:\gamma}$ 表示从 γ 个中选择排名为 m 的最佳个体。在 ES 中有两者选择模式，“＋”号选择 $(\mu+\lambda)$ 和“，”号选择 (μ,λ)。

(μ,λ) 选择模式中，只有新生的 λ 个个体，即 $P_o^{(t)}$ 种群进入选择池，从中选择 μ 个组成下一代父种群 $P_p^{(t+1)}$，$\gamma=\lambda$。显然必须 $\mu<\lambda$，才能实现选择功能，一种极端情况，$\mu=\lambda$，将无选择功能，因为所有子个体都直接进入下一代，此时，选择不能提供有关搜索信息，种群的进化过程是在搜索空间随机漫游。

$(\mu+\lambda)$ 选择模式考虑了父代个体，父代个体和新生个体同时进入选择池，$\gamma=\mu+\lambda$。在此，μ 和 λ 的取值没有限制，特殊地，$(\mu+1)$ 为稳态 ES。$(\mu+\lambda)$ 确保了当前最佳个体的保存，是一种精英（elitist）保留型的选择方式（elitism）。

$(\mu+\lambda)$ 容易确保全局收敛性，而 (μ,λ) 可能发散，它常用于适应值函数是时变的情形和多处理机并行计算。

2．变异

对于实数搜索空间，变异算子常用式（1-2）表示，变异矢量 $\mathbf{z}=(z_1,z_2,...,z_n)$，每个分量 $z_i,i=1,2,...,n$ 是各自独立、服从某种分布随机变量，如 Gauss 分布、Cauchy 分布、均匀分布等，将在第三章详细论述。

在二进制搜索空间 $\mathbf{x}\in B^n$，变异是对 $\mathbf{x}$ 每一个二进制位（bit）随机地翻转，变异概率（即位翻转概率）为 p_m，变异算子的无偏性体现在每一位由 1 变 0 和由 0 变 1 的概率一致，变异概率 p_m 即变异步长。

最简单的情形是 $\mathbf{x}$ 的每一位各自独立地变异，变异概率的理论和经验值[21]为 $p_m=1/n$，p_m 是可以控制的，但这种控制的实用性还没有令人满意的实际应

用[80]。Jason 等[120]在一个特殊设计的伪布尔型适应值函数证实，使用 $p_m = f(t)$，p_m 在 $1/N \le p_m \le 1/2$ 范围内周期性地变化可以改进 $(1+1)-ES$ 的性能。也可以参照 GA 的方式，改变变异概率。除了按位变异外，Mübhlenbein、Mahnig[121]提出了所谓因数分解分布算法（ factorized distribution algorithm）、Pelikan 等[122]提出了贝叶斯优化算法（Bayesian optimization algorithm）等多位相关变异方式。

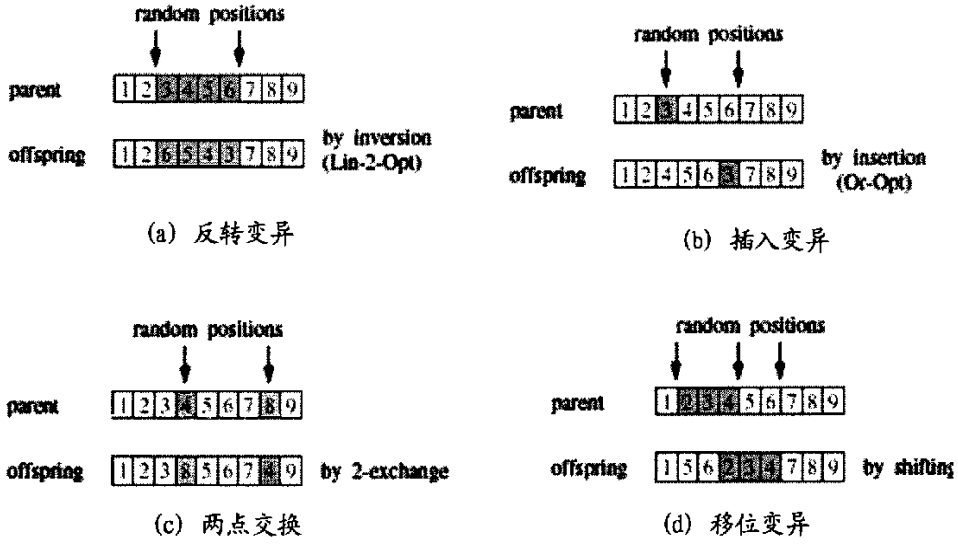
组合优化问题应用极为普遍，如著名的旅行商问题 TSP（traveling salesman problem），其个体的适应值是由基因的排列顺序决定的。变异通过改变亲本基因的排列完成，常称为排列变异算子(per-mutation operator)。常用的变异方式有 4 种类型，如图 1-1(a)~(d)所示。

反转变异(inversion)：随机地选择两个基因位置，这两个位置之间的基因反转排列。

插入变异(insertion)：随机选择一个基因，再随机选择一个插入位置，将所选择基因插到该位置。

两点交换(exchange)：随机选择两个基因，它们的位置互相交换。

移位变异(shifting)：随机选择一段基因，移动到随机选择的新基因位。



(a) 反转变异　(b) 插入变异

(c) 两点交换　(d) 移位变异

图 1-1 排列变异的四种常用方式

3. 重组

标准 ES 重组算子有离散重组和中间重组，一般认为，搜索过程中，变异基

于一个父个体的信息，而重组共享了ρ个父个体的信息[11]。与 GA 的交叉算子不同，在GA中两个父个体交叉产生两个子个体，而在ES中ρ个父个体重组后只产生一个个体。

设父代个体种群$\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_\mu$，其中$\mathbf{a}_i=(a_{i1},a_{i2},...,a_{iD})$，$i=1,2,...,\mu$（个体中可同时包括目标变量和策略参数，共$D$个基因），从中随机选择$\rho$个作为父个体，重组产生的个体$\mathbf{r}=(r_1,r_2,...,r_D)$。

离散重组：$\mathbf{r}$的每个元素r_k，$(k=1,2,...,D)$是随机地从ρ个父个体中选择相应的元素，即

$$r_k=a_{m_k k},\quad m_k=random\{1,2,...,\rho\} \tag{1-4}$$

中间重组：中间重组实际上是对每个父基因求平均，可表示为

$$r_k=\frac{1}{\rho}\sum_{m=1}^{\rho}a_{mk} \tag{1-5}$$

1.3.3 ES的改进研究

ES 从$(1+1)-ES$到$(\mu/\rho+\lambda)-ES$、$(\mu/\rho,\lambda)-ES$的演变过程伴随着 ES 的改进过程，除此之外，人们从其他方面对 ES 进行了改进研究。文献[26]研究了重组对收敛速度、收敛可靠性的影响，并把n_σ也纳入自适应的参数，甚至设想把学习率参数τ和τ'也作为遗传基因参加进化。文献[27]试验了$\rho=2$和$\rho=\mu$时的中间重组效果，通过一组测试函数试验，表明这两种情况得出的结果并不一致，只是比无重组的ES提高了收敛速度。文献[28]用球模型函数试验研究了有σ自适应（σSA：σ self-adaptive）和无自适应$(\mu/\mu,\lambda)-ES$，表明σSA并不是总能够引导搜索进程指向所希望的区域，而进化性能对学习率参数τ和τ'比较灵敏。文献[29][30]对离散重组和中间重组作了大量对比试验，结果表明对在大多数情况下，目标变量和策略参数同时使用离散重组，可以获得鲁棒性。重组算子的作用机理严重缺乏理论指导，而相关的试验结果并不相互一致，可推广性也不强。有两种理论支持重组算子的使用，其一是积木块假设BBH(building block hypothesis)[31][32]，认为不同父个体的优良基因（积木块）在一起组合，产生的后代也将同时具有这些优良特性，但 BBH 本身也颇受争议；其二为遗传修复GR（genetic repair）假设[33][34]，这一理论与BBH有点相反，认为重组并不使不同父个体的不同（所期望的）特征组合到子代，而是提取它们共同的特征，所以，重组是从父代提取它们相似的部分，对于中间重组算子（式(1-5)）的遗传修复功能尤其明显。

于一个父个体的信息，而重组共享了ρ个父个体的信息[11]。与 GA 的交叉算子不同，在 GA 中两个父个体交叉产生两个子个体，而在 ES 中ρ个父个体重组后只产生一个个体。

设父代个体种群$\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_\mu$，其中$\mathbf{a}_i=(a_{i1},a_{i2},...,a_{iD})$，$i=1,2,...,\mu$（个体中可同时包括目标变量和策略参数，共$D$个基因），从中随机选择$\rho$个作为父个体，重组产生的个体$\mathbf{r}=(r_1,r_2,...,r_D)$。

离散重组：$\mathbf{r}$的每个元素r_k，$(k=1,2,...,D)$是随机地从ρ个父个体中选择相应的元素，即

$$r_k=a_{m_k k},\quad m_k=random\{1,2,...,\rho\} \tag{1-4}$$

中间重组：中间重组实际上是对每个父基因求平均，可表示为

$$r_k=\frac{1}{\rho}\sum_{m=1}^{\rho}a_{mk} \tag{1-5}$$

1.3.3 ES 的改进研究

ES 从$(1+1)-ES$到$(\mu/\rho+\lambda)-ES$、$(\mu/\rho,\lambda)-ES$的演变过程伴随着 ES 的改进过程，除此之外，人们从其他方面对 ES 进行了改进研究。文献[26]研究了重组对收敛速度、收敛可靠性的影响，并把n_σ也纳入自适应的参数，甚至设想把学习率参数τ和τ'也作为遗传基因参加进化。文献[27]试验了$\rho=2$和$\rho=\mu$时的中间重组效果，通过一组测试函数试验，表明这两种情况得出的结果并不一致，只是比无重组的 ES 提高了收敛速度。文献[28]用球模型函数试验研究了有σ自适应（σSA：σ self-adaptive）和无自适应$(\mu/\mu,\lambda)-ES$，表明σSA并不是总能够引导搜索进程指向所希望的区域，而进化性能对学习率参数τ和τ'比较灵敏。文献[29][30]对离散重组和中间重组作了大量对比试验，结果表明对在大多数情况下，目标变量和策略参数同时使用离散重组，可以获得鲁棒性。重组算子的作用机理严重缺乏理论指导，而相关的试验结果并不相互一致，可推广性也不强。有两种理论支持重组算子的使用，其一是积木块假设BBH(building block hypothesis)[31][32]，认为不同父个体的优良基因（积木块）在一起组合，产生的后代也将同时具有这些优良特性，但 BBH 本身也颇受争议；其二为遗传修复GR（genetic repair）假设[33][34]，这一理论与BBH有点相反，认为重组并不使不同父个体的不同（所期望的）特征组合到子代，而是提取它们共同的特征，所以，重组是从父代提取它们相似的部分，对于中间重组算子（式(1-5)）的遗传修复功能尤其明显。
修复功能尤其明显。

变异算子是 ES 的主要算子，ES 的演变过程，很大程度上是变异算子的演变、改进。文献[35]为了提高全局搜索能力，增大跳出局部极值点的概率，引入所谓“灾难理论”（theory of disaster），每隔一定的正常进化代（如 100 代）后，用显著增大的变异步长进化若干代(3 代)。Kappler[36]受快速模拟退火(Simulating annealing）原理启发，引入了 Cauchy 变异代替 $(1+1)-ES$ 中的 Gauss 变异，发现该方法在处理一维函数优化问题时具有较强的鲁棒性；姚新等[37]把 Cauchy 变异推广到 $(\mu,\lambda)-ES$，用一组测试函数做对比试验研究，结果表明，Cauchy 变异具有较强的全局搜索能力，而局部搜索能力有所下降；林丹等[38]从理论上分析比较了 Gauss 变异、Cauchy 变异、均匀变异（uniform mutation）的局部和全局搜索能力，证明了 Gauss 变异局部搜索能力最强，均匀变异的全局搜索能力最强，而 Cauchy 变异的局部和全局搜索能力都居中。王云诚等[39]用均匀变异算子求解单峰函数，研究了其步长控制策略。刘若辰等[40]用基于细胞克隆选择学说的克隆算子和 Cauchy 变异算子改进了 ES。σSA －ES 的步长不能很好地跟踪进化进程[28][41]，王战权等[42]试验了变异步长使用对数 Cauchy 分布。文献[43]提出以步长－种群年龄函数控制的变异步长，但对不同函数，往往需要不同的变异步长控制方法，使这种方法缺乏灵活性。文献 [44][45]提出通过强化学习实现步长的自适应。文献[46][47][48]通过比较变异成功与否，确定变异的方向，引入非对称概率密度函数实现有向变异。文献[49]提出二方式变异法（two-way mutation），通过试探，改变变异方向和变异步长，取得了良好的效果。

在进化算法中，维持种群多样性是保证全局搜索能力的重要前提，种群规模对 ES 的多样性具有直接的影响，种群规模大，容易维持多样性，但也会增加个体排序的时间消耗。文献[50]根据进化进程而动态改变种群规模 λ，减小了计算消耗，[51]通过试验研究 λ 维持不变时，λ/μ 比率对 ES 进化效率的关系，提出自适应改变父种群规模 μ 的策略，[52]研究了在动态环境下，种群规模的自适应方法及对种群多样性的影响。

维持种群多样性和足够的选择压是相互矛盾的两个方面[2]，由于在进化进程中，遗传漂移的作用，使种群朝某个方向漂移而丧失多样性，多种群（multi-population、subpopulation）技术（或小生境技术: niching ）是抑制遗传漂移、维持种群多样性的有效方法[53]。与多种群 GA 比较，多种群 ES 的研究要薄弱得多，很多原理都是借鉴 GA 提出的，如适应值共享、排挤等。在 ES 中引入多种群技术的研究大都仅仅是为了增强全局搜索能力[54][55][56][57][58]，用于求多模态函数多个全局最优解和局部解的 ES 文献较少[59]。

1.3.4 ES 理论研究进展

与 GA 相比，ES 的理论研究显得很不足[60]，由于 ES 本身的复杂性，理论研究很复杂，很多理论成果是在简化的对象模型（如使用球模型函数）和简化的算法模型下得到的，普适性受到怀疑。试验结果也说明，很多研究结论并不一致，这些都给理论研究带来了困难。

1. 收敛性问题 和任何迭代型算法一样，收敛性是算法必须回答的首要问题。进化算法是一类随机搜索型的求解方法，其收敛性常表示为“能否以概率 1 收敛？”[61]。ES 提出近 40 年间，其收敛性理论主要以简单函数为对象，研究算法是否收敛于极值点[62][63]，其中球模型函数最常用[64][65]，尽管模型简单，然而收敛性的解析分析也很困难[66]。Rudolph 使用称为“诀窍理论”（martingale theory）的数学工具证明了 $(1+\lambda)-ES$ 的收敛性[67][68]。Yin 等用随机逼近（stochastic approximation）分析了 $(1,\lambda)-ES$ 的收敛性[69][70]。

Beyer 建立了较全面的收敛性理论[64][71][72]，其中包括全局收敛性和收敛性的单步测度（one-step measures），然而，Beyer 的方法使用了若干近似，如用一阶近似考虑 ES 的平均行为而非随机行为，用 Gauss 噪声给性能波动问题建模，从数学角度看,这些近似不是很严格[66]。

从应用的角度看，对于精英保留 EA，只要能保证全局遍历（或者每一状态可达），当 $t\to\infty$ 时就可以以概率 1 收敛[73][74][75][76]，郭观七[77]提出了收敛性分析的统一方法。在进化过程中，随着自适应变异步长的减小，全局遍历性不一定能满足，常出现早熟收敛，Rudolph 给出了一个这样的反例[78]，并通过理论分析，提出了防止早熟收敛的自适应策略[79]。

2. 收敛速度 全局收敛性往往建立计算时间趋于无穷或者种群规模无穷这一前提下，从理论上讲，以概率 1 保证收敛就足够了，但在实际应用时，计算时间是有限的，收敛速度成为人们关心的重要问题，到达极值点的期望搜索代数、适应值函数的计算次数、计算时间等是评价算法效率的常用定量指标[80]。通过对收敛速度的研究，可以初步估算算法需要的计算时间或进化代数，也可以得到有关算法参数对收敛性能的影响，指导算法的改进。

早在 1973 年，Rochenberg[10]以走廊模型和球模型函数为优化对象，定义了收敛速度（改进率：progress rate）：

$$\varphi = E[\|\mathbf{x}^*-\mathbf{x}^{*(t-1)}\|-\|\mathbf{x}^*-\mathbf{x}^{*(t)}\|] \tag{1-6}$$

其中，$\mathbf{x}^{*(t)}$ 为当前（第 t 代）最优解，$\mathbf{x}^{*(t-1)}$ 为上一代最优解，$\mathbf{x}^*$ 为真实最优解。Rochenberg 分析了 $(1+1)-ES$ 变异步长与收敛速度的关系，提出了进化窗（evolution window）概念，证明了当成功变异的概率为 1/5 时，可以得到最

大的期望收敛速度，由此提出了著名的 1/5 成功率准则，其变异步长为：

$$\sigma=\begin{cases}\sigma/\alpha & if\ \ P_s>1/5\\ \sigma\cdot\alpha & if\ \ P_s<1/5\\ \sigma & if\ \ P_s=1/5\end{cases} \tag{1-7}$$

其中成功变异的概率 $P_s=G_s/G$，G_s 为 G 代中成功变异的代数。系数 α 取决于待优化的目标函数、搜索空间的维数 n 和 G。当维数足够大 $n\geq 30$ 时，$G=n$，$0.85\leq\alpha<1$[11]。

文献[81]提出了以平均截止代数和截止代数分布熵的概念，并以此平面测度作为评价准则。文献[82]分析了不同变异步长控制策略下的性能，[83]分析了含有噪声的适应值函数的性能。文献[84][85]应用次序统计学[86][87]分析了使用 Cauchy 变异算子、Cauchy 与 Gauss 组合变异算子 EA 的局部收敛速度。

应该指出的是，即使对 EA 做很多简化，并使用简单的适应值函数，如球模型函数、绝对值函数 $f(x)=|x|$，收敛速度的分析也还是非常复杂的。

1.3.5 应用

ES 最初用于连续函数优化，并表现出较好的性能，现在， ES 在离散组合优化[88][89][90][91][92]、多目标优化[93][94]、约束优化[95][96]、含噪声的优化[97]、并行计算[98][99][100]等方面也得到了应用。在 GA 有应用的领域，几乎都可以看到 ES 的应用。

1.3.6 现状与趋势

在进化算法的三个分支 GA、ES、EP 中，ES 在求解连续函数优化问题具有良好的性能，其收敛速度较快、精度较高，但从发表的论文数量统计，如在 IEEE、SCI 等数据库，有关 GA 的论文都在二千篇以上，而 ES 不到 500 篇，EP 更少，可见，它们受重视的程度有很大差别，GA 研究最为广泛、深入，ES 次之，EP 研究较少，在 ES 的发源地欧洲，研究成果相对较多。在国内，人们对遗传算法的偏爱也很明显，从中国期刊网(www.cnki.net)统计，1999～2005 年内，有关 GA 的论文数量超过 1500 篇，而 ES 方面的论文不到 50 篇。对于求多模态函数优化问题，基于 GA 的研究成果很多，但有关 ES 的成果较少。这种现象与很多有关介绍进化算法的教材有很大关系，很多教材一般以 GA 为主，ES 只是其中的一个章节，占的篇幅也较少[2][22][23][101]，另外很多进化算法综述也主要以 GA 为主。

另一方面，GA、ES、EP 等进化算法正呈现相互融合、渗透之势[2][22][23]，它们的差别逐渐减小。在 ES 中，变异算子是主要的进化算子，其重要性自不待

说。对于GA，是否真正需要交叉算子是GA研究中一个颇具争议的问题。实际上，有的生物学家[102]认为变异是进化的主要根源，Schaffer等认为[103]，只有选择和变异的“简单进化”（naïve evolution）就可以完成爬山搜索，交叉算子可以加快进化，但变异可以求得更好的解。Spears [104]进一步建议适当修改的变异算子可以做交叉算子所做的一切。Falco[105]试验了无交叉算子的GA，其有效性与有交叉算子几乎相同。文献[106]证明了无交叉算子可以简化搜索，减少计算时间，而且与有交叉的GA效果差别不显著。文献[107]对使用一致交叉算子的GA与只有纯变异算子的GA做了试验对比，也得出结论认为它们的计算效果基本相同。现在，EA研究者们越来越重视变异的作用，对它的研究兴趣在不断增加，一般认为，交叉算子视所处理的问题而定，如果根据问题性质，交叉算子确有意义则可以考虑使用，而另一方面，只使用变异算子也足以求得最优解[108]。

1.4 论文主要研究内容和结构

正如本章开始部分所述，本文的研究工作起源于科研与工程实际中求解优化问题的需要，而这些问题往往难以用传统的优化方法求解，进化算法为求解这类问题提供了新的方法，但进化算法收敛速度慢、存在早熟收敛是其严重的不足。进化策略是当前主要的进化算法之一，在求解连续函数优化问题时，具有相对较好的性能，但仍然存在着上述不足，相对遗传算法，其理论研究更为薄弱。

进化策略中，广泛使用所有基因同时变异的全基因变异方式，如果一次只随机选择其中一个基因发生变异（本文称之为单基因变异），对算法的全局搜索能力、局部搜索能力产生哪些影响？相应的变异步长如何控制？多种群算法如何实现？本文将通过理论和仿真计算，与全基因变异ES进行对比分析和研究，并论述仿真试验研究平台的构建及应用。本文研究路线和组织如下：

进化算法是一类借鉴、学习、模拟自然界进化规律的随机型搜索算法，但也存在过于模仿之嫌，如在GA中，父个体的选择、交叉位置的确定、是否发生变异等都是随机确定的，只有父个体的选择过程中，考虑了个体适应值的大小，其他是完全随机的，这样势必使搜索过程沦为一种半盲目的状态，降低搜索的效率。基于这一认识，作者和众多启发式进化算法研究者一样，试图减小GA过多的随机性，加强对进化进程的引导，提出了一种强化引导型的遗传算法IEGA（induction-enhanced GA），本文第二章论述了这种算法的基本原理。IEGA使用了“保留最优、调节中间、淘汰最差”策略，并提出了一种基因调节算子，在

仿真计算试验中，发现在基因调节算子基础上加入了变焦微调后，对进化起主要作用的是变焦微调，而基因调节基本失效，通过对交叉算子的作用机理分析，说明交叉算子局部和全局搜索功能的可替代性，变异算子在进化计算中起主导作用。由于没有基因调节的 IEGA 事实上已经演变为一种进化策略，作者的研究重点也转向进化策略。

按照现有进化理论，自然界中的变异同时发生在生物的每个基因，以小幅度变化为主，大幅度变化较少。进化策略模拟了自然界的这种变异思想，其典型的变异算子使用 Gauss 分布，基本特点是所有基因同时变异（全基因变异），现有的有关 ES 理论，如收敛性、变异步长控制等都建立在这种变异方式的基础上。凭直觉，全基因变异的成功概率应该小于一次只有一个基因的变异（单基因变异），因为每个基因变异的成功概率小于 0.5，在第三章，通过理论分析和仿真计算证明了在相同变异步长，且变异步长较大时，全基因变异的成功概率低于单基因变异。

当前有关 ES 的局部和全局搜索能力及收敛性的研究、变异步长的控制策略都建立在全基因变异基础上，单基因变异 ES 的搜索能力、全局收敛性怎样？应该使用什么样的步长控制策略？在第四章，将对这些问题作研究，通过两个反例说明单基因变异全局搜索能力的不足，并提出改进的措施，引入均匀分布变异以弥补 Gauss 分布变异全局搜索能力的不足，建立$(\mu+\lambda+\kappa)-ES$；通过理论分析单基因变异局部收敛速度与变异步长之间的关系，建立相应的步长控制策略。仿真是研究进化算法的重要手段，作者提出了一种横向仿真方法以研究、改进算法的遗传算子，第四章将论述这一方法，并用于分析比较在不同变异步长时单基因变异、全基因变异算子的局部搜索能力、进化窗，然后，通过一组典型 100 维测试函数分析、比较不同算法的全局收敛性能、收敛速度、计算开销，说明单基因变异、递减型变异步长$(\mu+\lambda+\kappa)-ES$具有良好的性能。

由于遗传漂移现象的存在，单种群的进化算法很难维持种群多样性，导致算法的全局搜索能力不足，也很难同时求出多模态优化问题的多个最优解，多种群（或小生境）技术是解决这些问题的有效手段。第五章将论述基于$(\mu+\lambda+\kappa)-ES$的多种群进化策略$m\times(\mu+\lambda+\kappa)-ES$，说明算法的构造思想、实现方法。在基于适应值共享、排挤的多种群技术中，共享半径（或小生境半径、峰半径）是一个重要而复杂、难以确定的参数，作者提出了山谷探索法，直接判别同峰极值点与异峰极值点，第五章将论述这一方法的基本原理和与单基因变异相应的子种群收敛判据，通过一组多模态测试函数验证$m\times(\mu+\lambda+\kappa)-ES$准确求出所有局部最优解和全局最优解的能力。

进化算法的研究需要在理论研究的基础上做大量的仿真计算试验，在理论指导下，通过仿真计算分析、比较、验证不同策略、不同算法参数对不同优化对象的计算效果，一个方便、适用的计算试验平台是仿真计算研究的重要前提。作者在算法研究过程中，建立了一个通用性较强的开放式仿真计算试验平台，第六章将论述这一平台的构建思想，并通过复杂机电传动速度控制系统参数的优化、足球机器人的最优控制两个应用实例说明平台和算法的应用。

总之，本文内容可以概括为：变异算子的主导作用的认识（第二章）；单基因变异 ES 的成功变异概率分析（第三章）；单基因变异 ES 的步长控制策略、全局收敛性、局部搜索能力分析与改进（第四章）；基于单基因变异 ES 的多种群技术（第五章）；仿真计算平台的构建（第六章）；最后，是本文的结论。

第二章 变异算子的主导作用

作者在研究基本遗传算法 SGA (simple genetic algorithm)的过程中，觉得SGA 具有太多的随机性；使 SGA 沦为一种半盲目的搜索状态，提出了一种“保留最优、调节中间、淘汰最差”的强化引导型的遗传算法 IEGA(Induction-enhanced genetic algorithm)[110][111]，在 IEGA 中使用一种基因调节算子作为算术交叉算子，旨在加强对搜索过程的引导，但在对高维优化问题的仿真计算试验中，发现交叉算子基本失效，认识到变异算子在算法中起主导作用，并开始对进化策略及其变异算子的研究。作为本文的引子，反映了作者的研究路线。

本章首先简要介绍 SGA 的基本算子，然后论述 IEGA 的思路和改进过程，分析交叉算子的作用机理，变异算子起主导作用的原因，最后构造一种无交叉算子的 IEGA——似进化策略。

2.1 基本遗传算法

SGA 具有 Algorithm1.1 所描述的算法基本结构，在有关进化算法的教科书中已有介绍，GA 的遗传算子多种多样，为了叙述方便，在此，以实数编码 SGA 常用的算术交叉算子、非均匀变异算子和轮盘赌选择算子为例说明 GA 算子的基本特征。

轮盘赌选择

轮盘赌选择 RWS（roulette wheel selection）是 SGA 中最先使用的一种简单的选择方法，它根据赌盘原理设计，设种群规模为 N，其 N 个个体为 $\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N \in R^n$，在选择前，先累计种群中个体的总适应值（适应值需要确保都为正）：

$$f_{sum}=\sum_{i=1}^{N} f(\mathbf{x}_i)$$

根据每个个体的适应值大小计算其选择概率：

$$p_i=\frac{f(\mathbf{x}_i)}{f_{sum}},\quad i=1,2,...,N$$

得到每个个体的累积概率区间，它们依次为：$0\sim p_1$，$p_1\sim p_1+p_2$，…

仿照轮盘赌原理，产生（0，1）的随机数，该随机数落在某个个体的概率

区间则选择该个体。

算术交叉

首先从父种群中用 RWS 方法（或其他选择方法）选择两个个体 $\mathbf{x}_i, \mathbf{x}_j$，设交叉后产生的个体 $\mathbf{x}=(x_1, x_2, ..., x_k, ..., x_n)$，常用的算术交叉算子为[2][23]：

$$x_k = \alpha x_{ik} + (1-\alpha)x_{jk} = x_{jk} + \alpha(x_{ik} - x_{jk}), k = 1, 2, ..., n \tag{2-1}$$

其中 α 是在（0，1）内均匀分布的随机数。

非均匀变异

以非均匀变异[19]为例，设个体 $\mathbf{x}$ 是亲本，随机选择其中的基因 x_k 变异，得到 $\mathbf{x}'=(x_1, ..., x'_k, ..., x_n)$，其中

$$x'_k = \begin{cases} x_k + \Delta(t, UB_k - x_k) & \text{if random}\{0,1\}=0 \\ x_k - \Delta(t, x_k - LB_k) & \text{if random}\{0,1\}=1 \end{cases}$$

UB_k, LB_k 为变量 x_k 的上、下限。函数 $\Delta(t, y)$ 返回区间 $[0, y]$ 的一个值，以便当代数 t 增加时，$\Delta(t, y)$ 接近于 0 的概率增加，使变异的幅度随着代数 t 增加而减小，如 $\Delta(t, y)$ 可使用下面的函数：

$$\Delta(t, y) = y(1 - r^{(1-\frac{t}{T})^b})$$

其中 $r = random(0,1)$，T 是最大代数，b 是确定对迭代代数依赖程度的系统参数，可取 $b=5$。

从以上几种算子的原理可见，驱动 SGA 朝优化方向前进的主要动力是适应性好的个体具有更多的生存和繁殖后代的机会，SGA 在选择、交叉、变异过程中，都包含了随机性成分，是一种随机搜索型的优化算法。

收敛速度慢、存在早熟收敛是 SGA 存在的两个主要问题，它们与 GA 中最重要的两个因素是“种群多样性（diversity）”和“选择压力（selection pressure）[112]”直接相关。选择压力指选择机制给适应值较低的个体造成的生存压力，种群多样性指种群之间个体的差异性，这是一对需要相互协调的矛盾，它们决定了算法勘探（exploitation）和开采（exploration）能力[113]。选择压力大有利于使优秀个体得到更多的繁殖机会，提高收敛速度，但也是导致早熟收敛的一个重要因素，而太小的选择压力将使 GA 陷入随机漫游，丧失寻优能力。

2.2 一种强化引导型的遗传算法

SGA 收敛速度慢、存在早熟收敛。作者认为，影响遗传算法收敛速度的一

个重要原因在于遗传算法过多地模拟自然界的概率选择法则，以上节介绍的几种典型遗传算子为例，如个体的生存是按照其适应值的高低以不同的概率选择的，进行交叉的个体是随机选择的，交叉点的位置选择是随机确定的，变异基因是随机选择的，变异的量也带有随机性，这样，寻优过程沦为“半盲目”（semi-blind）的状态，使 GA 勘探与开采能力难以协调，减缓了收敛速度。作者试图通过减少这种“自然型”的随机性，使用确定型选择算子代替随机型选择算子，参照基因工程和育种原理，提出一种“保留最优、调节中间、淘汰最差”的强化引导型的遗传算法 IEGA，其中使用基因调节算子代替算术交叉算子，以加强对进化方向的引导，提高搜索效率，对优秀个体确定性地予以保存，而对于劣质个体，确定性地淘汰。

2.2.1 IEGA 基本原理

编码方案

遗传算法主要有浮点数编码和二进制编码方案，浮点数编码方案相对于二进制编码具有以下优点：1）基因与变量一一对应，减少编、解码计算的时间开销；2）避免了“Hamming 悬崖”现象的影响 [2]。另外在解决连续优化问题时，浮点数编码具有较高的精度，IEGA 的基因调节和变焦微调算子用浮点数编码更自然方便，故 IEGA 使用了浮点数编码方案，对于多变量问题，每一个变量对应一个基因。

选择策略

影响遗传算法两个最重要的因素是 “种群多样性”和“选择压力” [112][114]，它们往往互相矛盾，按适应值大小确定选择概率的轮盘赌等概率型选择方式，很难兼顾种群多样性和选择压力，常用的解决方案有两种：1）修改采样机制（选择算子）；2）对适应值函数进行尺度变换。

IEGA 使用确定性和随机性相结合的选择策略，设种群规模为 N，除初始种群全部随机产生外，其他每一代种群由三部分组成：保留种群、繁殖种群、随机种群，它们分别包含 N_1、N_2、N_3 个个体，$N=N_1+N_2+N_3$。对于求最大值的优化问题，父种群个体按其适应值从大到小排序，保留其中最好的 N_1 个个体组成保留种群，直接进入下一代；对父种群前 N_2+1 个个体，使用基因调节方式产生 N_2 个个体，组成繁殖种群；随机产生 N_3 个个体，组成随机种群，替换父种群中被淘汰的 N_3 个最差个体，用这种淘汰替代算子代替常规的变异算子。这样构成了 IEGA 的“保留最优、调节中间、淘汰最差”选择策略。

相对于概率型选择方式（如轮盘赌等），IEGA 引入的确定型选择法则有以下特点：不论父代中某个个体相对于其他个体多么突出，该个体最多也只有一

份进入下一代，这样有利于克服早熟收敛，同时保持了足够的选择压力；在种群中始终有N_3个随机产生的新个体，在整个可行解区间均匀分布，与常规的变异算子相比，它更有利于维持种群的多样性，保持全局搜索能力，实现勘探与开采操作的协调与平衡[114]。另外，这种排序选择方式不要求适应值函数为正，也不需要进行适应值函数的尺度变换。

基因调节原理

对于式(1-1)所描述的n维多变量求最大值优化问题，设$\mathbf{x}_1=(x_{11},x_{12},...,x_{1n})$，$\mathbf{x}_2=(x_{21},x_{22},...,x_{2n})$是两个体，且$f(\mathbf{x}_1)>f(\mathbf{x}_2)$，则新个体$\mathbf{x}_3=(x_{31},x_{32},...,x_{3n})$的每一个基因由以下基因调节公式产生：

$$x_{3i}=x_{1i}+k(x_{1i}-x_{2i}),i=1,2,...,n \tag{2-2}$$

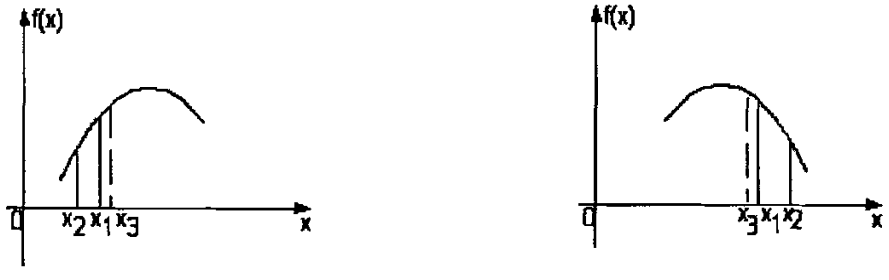
其中$k>0$称为调节系数，它决定调节的幅度，取得太大，容易导致计算结果不收敛，太小则收敛速度慢，根据经验，k取0.1。

以下用单变量、求最大值问题说明基因调节原理。

设两个体的基因分别为x_1、x_2，适应值为$f(x_1)$、$f(x_2)$，且$f(x_1)>f(x_2)$，但都不是极值点，若a)：$x_1>x_2$则可以期望，对于适当小的正数k，由式(2-3)得到的$x_3>x_1$满足$f(x_3)>f(x_1)$。反之，若b)：$x_1<x_2$，则可以期望由式(2-3)得到的$x_3<x_1$满足$f(x_3)>f(x_1)$。(a)、(b)两种情况分别示于图2-1（a)、(b)。

$$x_3=x_1+k(x_1-x_2) \tag{2-3}$$

在算法实现时，由于已经对个体按适应值从大到小进行了排序，只要从父代的前N_2+1个个体中依次取两个体按式(2-2)即可生成N_2个子个体，构成繁殖



(a) $x_1>x_2$　　(b) $x_1<x_2$

图2-1　基因调节原理示意图

种群。在数学意义上，基因调节是外推，它利用两个体的适应值大小差异，引导了外推（进化）的方向。

按照式(2-1)所述的常规实数编码算术交叉算子，x_{3i}为

$$x_{3i}=\alpha x_{1i}+(1-\alpha)x_{2i}=x_{2i}+\alpha(x_{1i}-x_{2i}),i=1,2,...,n \tag{2-4}$$

式中α是在[0，1]中均匀分布的随机数，比较基因调节公式(2-2)与(2-4)，虽然它们在形式上式相同，但存在本质的差别，式(2-4)的父代个体是根据适应值大小按不同概率随机选择的，α也是随机的，方向性较弱，无自适应性。式(2-2)的父代个体是排序后依次选择的，$\mathbf{x}_1$总是适应值较大的个体基因，$\mathbf{x}_2$是适应值较小的个体基因，它们的适应值相隔较近，k作为调节系数（步长）是确定的，具有较强的方向性。以这种基因调节方式代替常规的交叉操作，在基因调节过程中，根据适应值的大小引导基因的调节方向，以提高搜索效率。

另外，式(2-2)本身具有一定的自适应特性，一般而言，在进化的开始阶段，个体比较分散，距最优解比较远，基因之间的差异$|x_{1i}-x_{2i}|$较大，用式(2-2)产生的基因调节幅度也大；而在进化后期，种群中的优秀个体逐步靠近，基因差异$|x_{1i}-x_{2i}|$较小，基因调节幅度也减小，有利于得到更高精度的基因。

变异算子

IEGA 的变异是通过淘汰最差个体，用新产生个体代替实现变异功能，它在全局范围内均匀分布。

2.2.2 算法描述

综上所述，IEGA 的伪代码描述如 Algorithm 2.1 所示。

每一代进化计算中，IEGA 除了需要计算$N-N_1$个新个体的适应值外，所需的额外计算量很少，有利于提高计算速度。另外，对于求最小值的优化问题，只要在个体排序时从小到大排序，或者适应值的计算结果符号取反，转化为求最大值问题即可。

2.2.3 仿真计算

为了验证 IEGA 的效果，选择以下函数进行仿真计算：

1) 球函数：

$$f_1(\mathbf{x})=\sum_{i=1}^{n}x_i^2 \tag{2-5}$$

变量范围为[－20，30]；它是常用的所谓球模型函数，是连续、强凸单峰函数的代表，$\mathbf{x}^*=(0,...,0)$时，理论全局最小值为$f_1(\mathbf{x}^*)=0$。

2) Ackley 函数：

$$f_2(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e \tag{2-6}$$

其变量范围为[-20，30]，Ackley 函数是拓扑结构的主体为指数超曲面，余弦函数在指数超曲面上形成大量的局部极小值点，搜索过程容易陷入局部极小，$\mathbf{x}^* = (0,...,0)$ 时，理论全局最小值为 $f_2(\mathbf{x}^*) = 0$。

3) Rosenbrock 函数：

$$f_3(\mathbf{x}) = \sum_{i=1}^{n-1}[100(x_i - x_{i+1})^2 + (x_i - 1)^2] \tag{2-7}$$

Algorithm 2.1： IEGA-1

Begin

$t = 0$；

随机产生 N 个个体；

评价 N 个个体;

按适应值从大到小排序；

while （不满足终止条件时）

Begin

产生新一代种群：

Begin

保留最好的 N_1 个个体；

用基因调节算子(2-2)产生 N_2 个个体；

随机产生 N_3 个新个体代替最差的 N_3 个劣等个体；

End

评价 N 个个体；

按适应值从大到小排序；

(报告产生的新种群的适应值)；

$t = t + 1$；

End

End

其变量范围为[-5.12，5.12]， Rosenbrock 函数是不可分的具有互异特征值的二次函数，全局最小值位于底部平坦的峡谷中，$\mathbf{x}^*=(1,...,1)$ 时，理论全局最小值为 $f_3(\mathbf{x}^*)=0$。

它们的二维适应值函数拓扑曲面分别如图 2-2（a)、(b)、(c）所示。

图 2-3 给出了 n 为 2、4、10、30 时的 IEGA-1 的计算结果，使用的参数为 $N_1=2,N_2=8,N_3=5$，其终止条件为最大进化代数为 1000，从图可见，IEGA-1 具有一定的寻优能力，但存在收敛速度慢和早熟收敛现象。

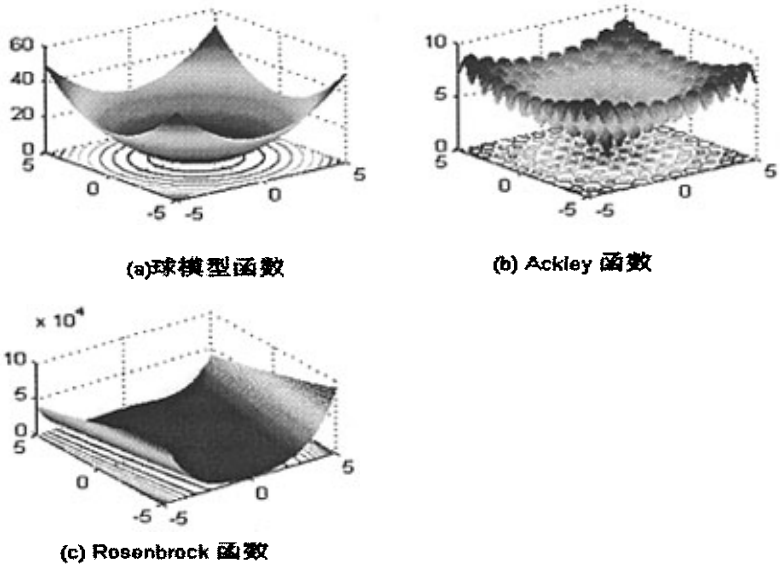
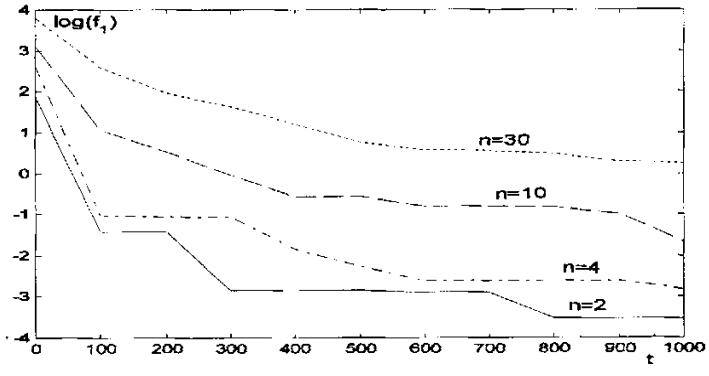
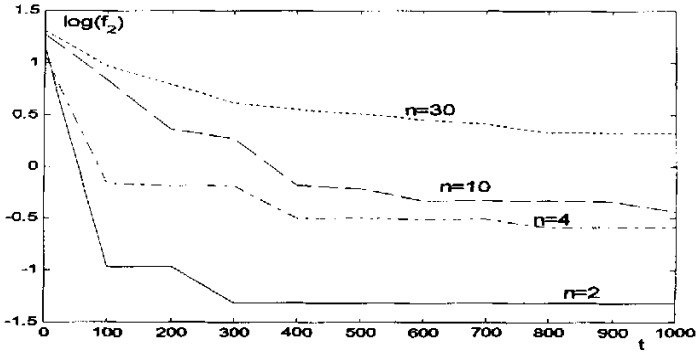


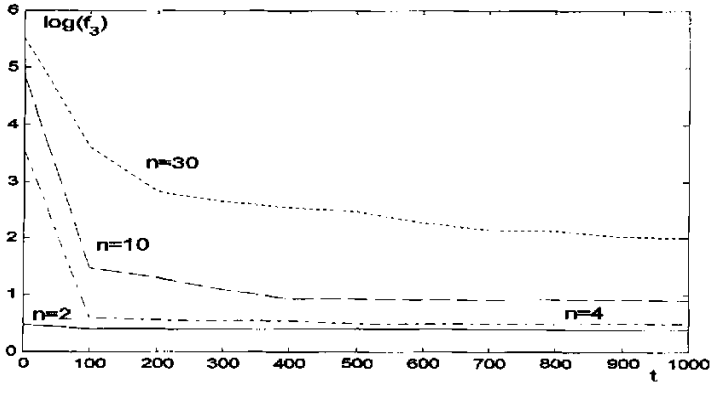
图 2-2 二维测试函数的适应值拓扑曲面图



（a）球函数



(b) Ackley 函数



(c) Rosenbrock 函数

图 2.3 适应值变化曲线

2.3 IEGA 的进一步改进

引入变焦微调

仿真计算结果表明，算法 IEGA-1 寻优效果不理想，其根源在于作为交叉操作的基因调节算子由于以下原因导致局部搜索能力不强。

原因之一：式(2-2)所示基因调节算子的基因调节作用依赖于两个个体基因的差异，由于参加调节的是种群中优秀的保留种群部分，在进化后期，它们的基因差异很小，致使基因调节作用很微小，局部搜索能力降低。

原因之二：当个体变量数较多时，个体的基因也多，在基因调节过程中，很容易出现相同或很相近的基因，这时，使用式(2-2)进行基因调节时，相同或相近的基因失去调节作用。

为此，当个体的两个基因相差很小时，在基因调节公式(2-2)的基础上，增加人为的扰动，这种扰动也可以看作是一种微调，其实现的方式如下：

$$x_{3i} = x_{1i} + k(x_{1i} - x_{2i}) + \zeta \tag{2-8}$$

$$\zeta = \begin{cases} 0 & if\ |x_{1i} - x_{2i}| \geq \varepsilon \\ random(-\zeta_m, \zeta_m) & else \end{cases} \tag{2-9}$$

其中$i = 1,2,...,n$，ζ为微调算子，ζ_m为微调幅度。引入微调算子后的仿真试验发现，对于某个固定的ζ_m，进化进程得到改善，但进化到一定阶段，又陷入停顿状态，减小ζ_m，又可以继续进化，随后又陷入停顿，这样，不断减小ζ_m，进化过程逐步逼近最优解。这一结果说明，微调幅度需要跟踪进化进程，不断减小。于是，引入了变焦微调，即在进化过程中，如果进化停顿，则减小微调幅度。这种变焦微调过程与日常工作中人工调节仪器仪表的过程相似，刚开始时粗调，调节的幅度可以大点，当调到接近目标时，调节幅度逐步减小，使用细调、微调，具有变焦微调功能的 IEGA-2 描述如 Algorithm 2.2 所示。

Algorithm 2.2：IEGA-2

Begin

$t = 0$;

随机产生N个个体;

评价N个个体;

按适应值从大到小排序;

while （不满足终止条件时）

Begin

产生新一代种群:

Begin

保留最好的N_1个个体;

用基因调节算子(2-8)、(2-9)产生N_2个个体;

随机产生N_3个新个体代替最差的N_3个劣等个体;

Algorithm 2.2(续)

End

评价 N 个个体；

按适应值从大到小排序；

如果 t_m 代没有进化，则 $\zeta_m = 0.75\zeta_m$，否则，ζ_m 不变。

(报告产生的新种群的适应值)；

$t = t+1$；

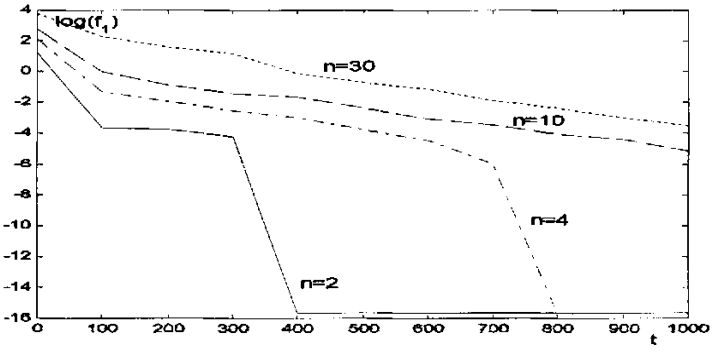
End

End

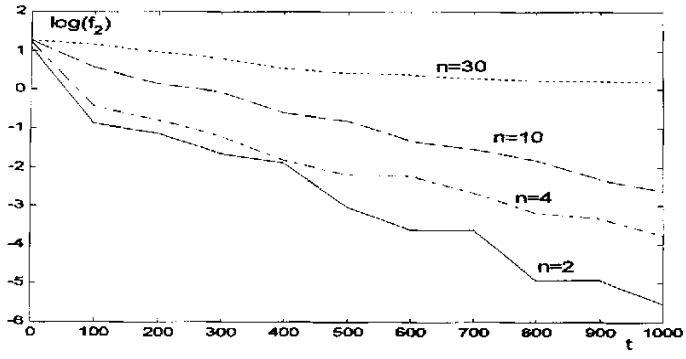
IEGA-2 的仿真结果如图 2-4 所示，由图可见，即使对于高维优化函数，IEGA-2 也具有良好的寻优效果。图 2-5 给出了用 IEGA-1，IEGA-2 分别计算 100 维球函数和 Ackley 函数的直接比较结果，说明变焦微调的引入显著提高了 IEGA-2 的局部搜索能力。

基因调节的效果

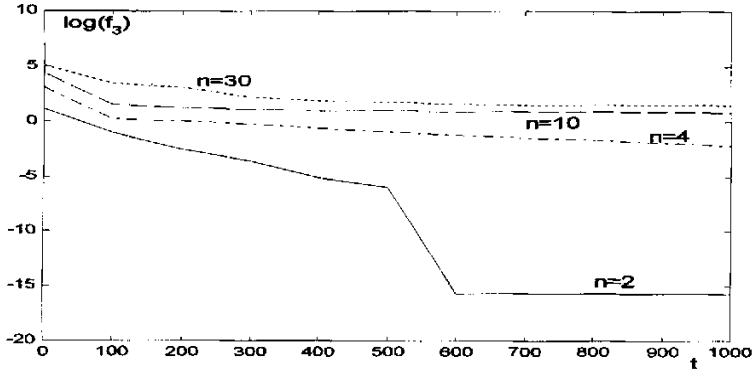
为了获得合理的基因调节系数，k 取不同的值，考察进化效果，得到的结果令人惊异：在引入变焦微调以后，当 $k=0.0$ 时得到的结果甚至比 $k=0.1,0.2,0.5,1.0$ 都好！图 2-6 为 100 维球模型函数在不同 k 值时的典型计算结果，k 取值太大，使基因调节效果变差容易理解，而 $k=0.0$ 意味着没有基因调节，对于以基因调节为交叉算子的式(2-8)，当 $k=0.0$ 时，实际上是一种变异步长 ζ_m 不断递减的均匀变异算子，也就是说，IEGA-2 可以没有交叉算子，只需要变异



(a) 球函数

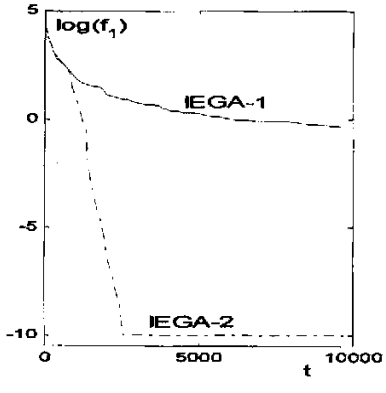


(b) Ackley 函数

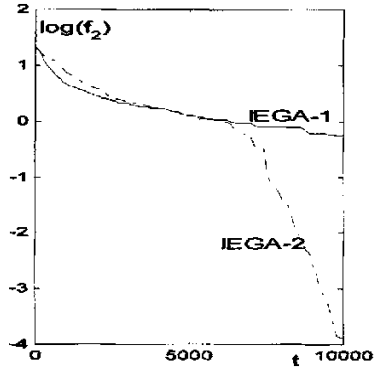


(c) Rosenbrock 函数

图 2-4 IEGA-2 计算适应值变化曲线



(a) 100 维球函数



(b) 100 维 Ackley 函数

图 2-5 IEGA-1、IEGA-2 计算结果比较

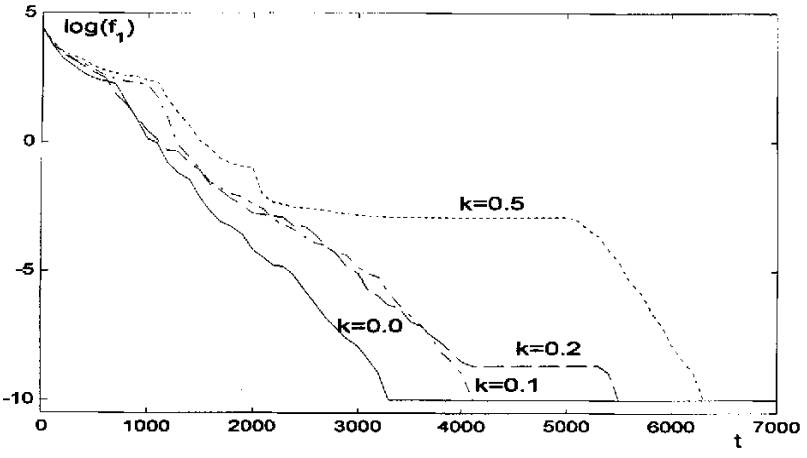


图 2-6 不同基因调节步长时，100 维球模型函数的进化效果比较

算子，而无交叉算子的 IEGA-2 几乎就是一种使用变异步长递减均匀变异算子的进化策略。

这一结果促使作者重新认识进化算法中交叉算子的作用机理，并把重心转向进化策略及其变异算子的研究。

2.4 交叉算子作用机理分析

本节将分析算术交叉算子和本文提出的基因调节算子的作用机理，为了叙述方便，先定义以下术语。

定义 2.1　成功操作　设 $\mathbf{x}_1$、$\mathbf{x}_2$ 是两个体，它们经遗传算子操作后，得到新个体 $\mathbf{x}'$，如果 1）$\mathbf{x}'$ 与极值点个体 $\mathbf{x}^*$ 的欧氏距离满足：

$$\|\mathbf{x}'-\mathbf{x}^*\|<\min\{\|\mathbf{x}_1-\mathbf{x}^*\|,\|\mathbf{x}_2-\mathbf{x}^*\|\}$$

则 $\mathbf{x}'$ 在 $\mathbf{x}_1$、$\mathbf{x}_2$ 的基础上得到了改进，为成功操作；

2）$\mathbf{x}'$ 的适应值在 $\mathbf{x}_1$、$\mathbf{x}_2$ 基础上得到了改进，即对于最大值优化问题，有：

$$f(\mathbf{x}')>\max\{f(\mathbf{x}_1),f(\mathbf{x}_2)\}$$

则为成功操作。

1)、2）两种成功操作定义不完全等价，但在算法研究过程中，为了分析处理方便，常不严格加以区别，都称为 $\mathbf{x}'$ 得到了进化。在很多文献中，以球模型函数为研究对象，这时，两者等价。

算子成功操作的能力体现了其局部寻优能力，某个算子成功操作机会越多

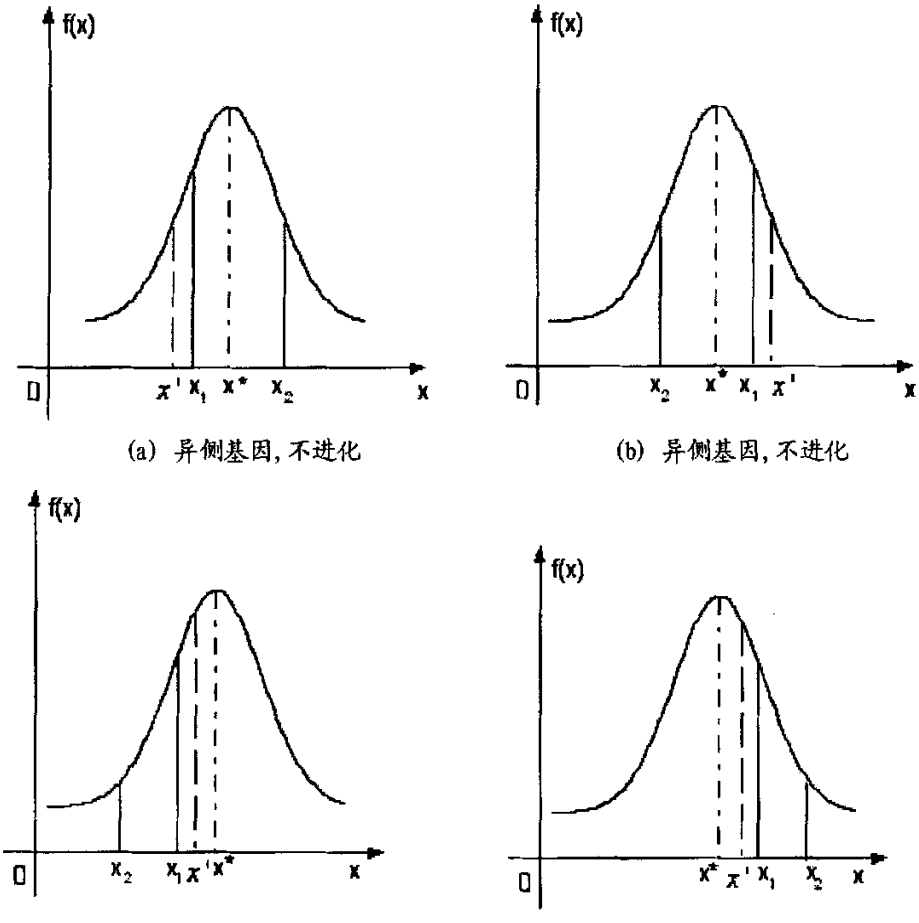
（概率越大），它的局部搜索能力越强。

定义 2.2 同侧基因与异侧基因 设x_1、x_2是两个体基因，x^*是极值点基因，如果$(x_1-x^*)(x_2-x^*)>0$，则称x_1、x_2互为同侧基因，否则，称x_1、x_2互为异侧基因。

2.4.1 一维时交叉算子的作用

基因调节算子的作用机理

设$|x_1-x^*|=\delta$为x_1与极值点x^*的距离，当x_1、x_2为异侧基因时，如图 2-7(a)、(b) 所示，由式(2-2)得到的新基因x'不进化，　x'与x_1同侧；当x_1、x_2为同侧基因时，如图 2-7 (c)、(d)所示，x'得到进化的概率为：



(a) 异侧基因，不进化　　(b) 异侧基因，不进化

(c) 同侧基因，以概率进化　　(d) 同侧基因，概率进化

图 2-7　基因调节的进化作用分析

$$p_{ev1} = P\{k \mid x_1 - x_2 \mid < 2\delta\} \tag{2-10}$$

算术交叉算子的作用机理

用算术交叉算子式(2-4)对x_1、x_2进行交叉操作时，得到两个新基因x_1'、x_2'：

$$x_1' = x_1 + \alpha(x_2 - x_1) \tag{2-11a}$$

$$x_2' = x_2 + \alpha(x_1 - x_2) \tag{2-11b}$$

其中x_1、x_2是以某种选择策略随机选择的个体，α是[0，1]内均匀分布的随机数。式(2-11)实际是随机内插算式，为了叙述方便，假设：

$$x_1 = \min\{x_1, x_2\}, x_2 = \max\{x_1, x_2\}$$

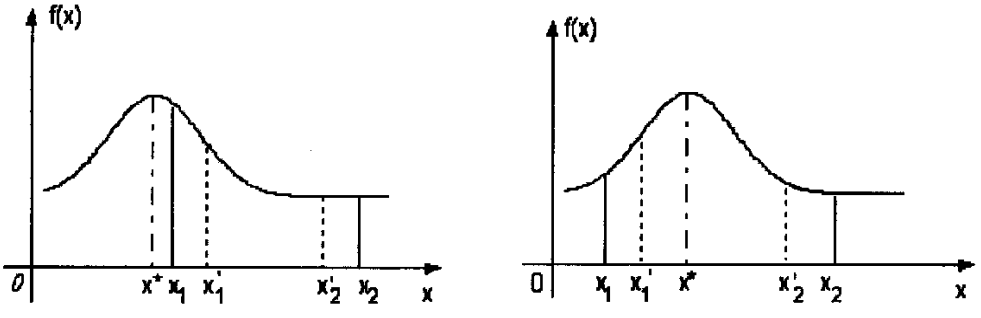
$$\delta_{\min} = \min\{|x_1 - x^*|, |x_2 - x^*|\}, \delta_{\max} = \max\{|x_1 - x^*|, |x_2 - x^*|\}$$

x_1'、x_2'将始终分布在$[x_1, x_2]$之内。

如图 2-8（a）所示，当x_1、x_2是同侧基因时，产生的新基因x_1'、x_2'，不能进化，x_1'、x_2'仍为同侧基因。当x_1、x_2为异侧基因时，如图 2-8(b)所示，x_1'、x_2'进化的概率为：

$$p_{ev2} = P\{\alpha(x_2 - x_1) < 2\delta_{\min} \cup [\delta_{\max} - \alpha(x_2 - x_1)] < \delta_{\min}\} \tag{2-12}$$

从以上两种算子的作用过程可见，对于基因调节算子，它需要参加操作的两个基因是同侧的，才以概率p_{ev1}进化；而对于算术交叉算子，它需要参加操作



(a) 同侧基因交叉，不进化　　(b) 异侧基因交叉，以概率进化

图 2-8　算术交叉算子的进化效果分析

的两个基因是异侧基因，才以概率 p_{ev2} 进化。比较式(2-11)和式(2-12)，由于基因调节系数 k 小，p_{ev1} 比较大，而式(2-12)中含有随机数 α，使 p_{ev2} 较小。在没有先验知识的前提下，参加操作的两个基因是同侧或异侧的机会是相同的，所以，这两种算子操作成功机会分别为 $0.5p_{ev1}$ 和 $0.5p_{ev2}$，都小于 0.5。

从另一个角度考虑，如果在式(2-12)中，令 $k=0.0$，取消基因调节功能，它变为一个均匀变异算子，对 x_1 作变异操作：

$x'=x_1+\zeta$

$\zeta=random(-\zeta_m,\zeta_m)$，只要有合适的幅度 ζ_m，其操作成功的概率完全可以与 $0.5p_{ev1}$ 或 $0.5p_{ev2}$ 相当，说明变异算子可以替代交叉算子的局部搜索功能。

2.4.2 高维时交叉算子的作用

在高维情况下，个体基因较多，对两个具有不同适应值的个体 $\mathbf{x}_1=(x_{11},x_{12},...,x_{1n})$ 和 $\mathbf{x}_2=(x_{21},x_{22},...,x_{2n})$，不妨设 $f(\mathbf{x}_1)>f(\mathbf{x}_2)$，在没有适应值函数的梯度信息的情况下，很难断定是哪些基因的变化将使个体的适应值增大还是减小，基因调节算子都按照从 $\mathbf{x}_2$ 到 $\mathbf{x}_1$ 方向调节基因，操作成功的概率将大幅度下降；另外，这些基因中，同侧基因和异侧基因混杂，而异侧基因经基因调节作用后产生的基因一定变差，这也将降低基因调节算子的成功率；再者，如 2.3 节分析，高维情况下，由于参加调节的个体基因很容易出现相近的现象，这时，对于以基因差别作为调节作用的调节算子失去调节功能，类似的分析也适用于算术交叉算子。以上原因导致交叉算子的作用非常有限。

2.4.3 交叉算子的可替代性

对于 GA，是否真正需要交叉算子是 GA 研究中一个颇具争议的问题。实际上，有的生物学家[102]认为变异是进化的主要根源，Schaffer 等认为[103]，只有选择和变异的“简单进化”（naïve evolution）就可以完成爬山搜索，交叉算子可以加快进化，但变异可以求得更好的解。Spears [104]进一步建议适当修改的变异算子可以做交叉算子所做的一切。Falco[105]试验了无交叉算子的 GA，其有效性与有交叉算子几乎相同。文献[106]证明了无交叉算子可以简化搜索，减少计算时间，而且与有交叉的 GA 效果差别不显著。文献[107]对使用一致交叉算子的 GA 与只有纯变异算子的 GA 做了试验对比，也得出计算效果基本相同的结论。

现在，EA 研究者们越来越重视变异的作用，对它的研究兴趣在不断增加，一般认为，交叉算子的使用视所处理的问题而定，如果根据问题性质，交叉算子确有意义则可以考虑使用，而另一方面，只使用变异算子也足以求得最优解[108]。交叉算子和重组算子的作用机制和理论分析是进化算法研究中较薄弱的环节之一。遗传算法学派基于模式定理和积木块假设认为交叉、重组算子可以

提取不同个体的优秀基因，而进化策略学派从基因修复的角度应用交叉、重组算子，认为交叉、重组可以修复个体中缺失的基因，这两种理论本身有一定矛盾。

作者从 IEGA 的改进过程也认识到交叉算子的局部搜索功能可以用变异算子等价替代。而对于交叉算子的全局搜索能力，如果变异算子中使用大幅度的变异，如在 IEGA 中的通过淘汰最差的个体，以均匀分布重新产生新个体，也将具有等价的全局搜索作用，因此，交叉算子的局部和全局搜索功能可以由变异算子替代，在 IEGA-2 中，变异算子起主导作用。

2.4.4 无交叉算子的 IEGA-2 与进化策略

没有交叉算子后的 IEGA-2，与进化策略比较，它们非常的相似：

IEGA-2 从 N 个个体中保留 N_1 个最优秀的个体，式(2-8)所示的基因调节算子当 $k=0$ 时实际就是均匀变异，它产生了 N_2 个个体，而对最差的 N_3 个淘汰后重新初始化产生新个体的操作也是一种变异，所以，与 $(\mu+\lambda)-ES$ 相比，IEGA-2 就是 $\mu=N_1$、$\lambda=N_2+N_3$、使用均匀变异的进化策略。

基于这一认识，作者开始了进化策略研究，主要研究变异算子的改进。

2.5 小结

本章论述了对 SGA 的一种改进试验，提出了一种强化引导型的遗传算法 IEGA，在算法中使用“保留最优，调节中间，淘汰最差”的选择策略，使用基因调节算子作为交叉算子，淘汰与更新算子作为变异算子，在仿真计算试验中，发现了基因调节算子的不足，引入变焦微调进一步改进了基因调节算子。

引入变焦微调后的仿真试验表明，IEGA-2 可以没有基因调节算子，说明交叉算子的局部和全局搜索功能可以被变异算子所替代，为此，进一步分析了交叉算子的作用机理，说明了交叉算子的可替代性，变异算子在算法中起主导作用。没有交叉算子的 IEGA-2 事实上是一种使用均匀变异算子的进化策略，作者将研究重点转向对进化策略及其变异算子的研究。

第三章 单基因变异进化策略

进化策略最初的变异思想来源于进化论，通过变异算子产生子代时，子代个体的每个基因都是从父个体基因变异产生的，本文把这种变异方式称为全基因变异(All-gene mutation)。凭直觉，所有基因同时变异，变异产生的进化效果很容易相互抵消，使变异产生进化（成功变异）的概率下降，影响变异算子的局部搜索能力，为此，作者提出了单基因变异（single-gene mutation）方式，即一次只从父个体中随机选择一个基因发生变异，其他基因直接从父个体继承、复制，本章将从理论上分析这两种变异方式的成功概率。

本章首先综述进化策略中的变异算子，然后从理论上分析比较全基因变异与单基因变异的成功概率，并通过仿真实例比较二者的进化效果，证明单基因变异具有较强的局部搜索能力。

3.1 进化策略中的变异算子

3.1.1 变异算子设计的一般原则

在 ES 中，变异算子通常作为基本的遗传操作算子，它是遗传变化的主要根源。变异算子的设计一般与问题有关，但现在还没有建立通用的变异算子设计方法学，Beyer[115]分析了几种比较成功的变异算子，并从理论角度考虑，提出了变异算子设计的三条原则，分别为可达性(Reachability)、无偏性（unbiasedness）、可标度（scalability）。

可达性 可达性指对于任一给定的父个体状态 $\mathbf{a}_p=(\mathbf{x}_p,\mathbf{s}_p)\in I$（此处，把个体对应的基因看作是个体空间的状态），变异算子能在有限的变异次数或进化代数内达到任意的其他状态 $\tilde{\mathbf{a}}=(\tilde{\mathbf{x}}_p,\tilde{\mathbf{s}}_p)\in I$。这一条件使算法具有全局搜索能力或遍历性，是证明全局收敛性的必要条件。

无偏性 这一要求来源于 Darwin 进化论。在进化算法中，选择和变异是两个不同而且有点矛盾的作用。选择起到勘探（exploit）适应值信息，以指导搜索过程朝向有潜力的搜索空间，而变异用于对搜索空间的探索（explore），它不应该使用任何适应值信息，只能使用源于父种群的搜索空间信息。按照 Beyer[115]的观点，变异算子不应该对所选择的父个体有任何偏好，不应该引入偏移。对于给定的父个体，尽量无偏是设计变化算子的原则。根据这一思想，自然会想

到“最大熵原理”，应用这一原理，在无约束的实数搜索空间 R^n，使用 Gauss 分布（即正态分布）是合适的，而在整数搜索空间 Z^n 可以是几何分布[116]。

可标度 可标度指变异步长或者每一次变异的平均长度可以调节以适用适应值函数拓扑曲面的特性和进化进程，以确保可以连续地“进化”。

值得说明的是，这三条原则仅仅是一般性的，根据所处理的问题不同，它们的重要性也有差别，与这一原则相冲突的变异算子在特定的问题中不一定失败。特别是无偏性原则，在启发式的 EA 中，往往通过强化学习[44][45]、或者利用问题相关的领域知识，使用定向变异[46][47][48]可以得到更好的性能。

3.1.2 实数搜索空间的变异算子

对于实数搜索空间 R^n，CES（classical ES ）的变异算子可表示为

$$\mathbf{x}' = \mathbf{x} + \mathbf{z} \tag{3-1}$$

$\mathbf{z} = (z_1, z_2, ..., z_n)$ 为 n 维随机变量。CES 使用 Gauss 分布的随机变量，目标变量的各个分量（基因）同时变异，变异步长（标准差）σ 既可以以内在参数的形式作为个体基因与个体目标变量基因一起进化，也可以作为外在参数形式以某种控制方式，如 1/5 成功规则，随进化过程改变变异步长。

当变异步长作为外在参数时，所有分量都为相同的变异步长 σ，常称为适应型步长（Adaptive σ）控制策略，使用 Gauss 分布随机变量时：

$$\mathbf{z} = \sigma(N_1(0,1), N_2(0,1), ..., N_n(0,1)) \tag{3-2}$$

$z_i \sim N_i(0,\sigma)$，$i = 1,2,...,n$ 是互相独立的随机变量。$N(0,\sigma)$ 的概率密度函数 *pdf*（probability density function）为：

$$f_N(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2}\frac{z^2}{\sigma^2}) \tag{3-3}$$

如图 3-1（a）所示，变异步长 σ（标准差）控制了变异增量的分散程度，由于每个分量使用相同的变异步长，采样点 $\mathbf{z}$ 在各个方向的分布相同，称为各向同性（isotropic）变异，$n = 2$ 时，$\mathbf{z}$ 的分布如图 3-1（b）所示。这种适应型步长变异算子最初用于 $(1+1)-ES$，使用 1/5 成功规则控制变异步长，如式（1-7）所示。

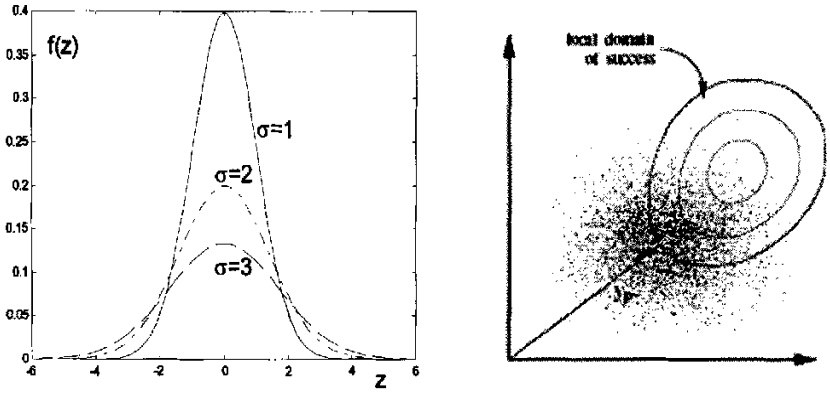
当变异步长作为内在参数时，常称为自适应型步长（self-adaptive σ，SA-σ）控制策略。种群中的个体 $\mathbf{a}$ 由目标变量 $\mathbf{x} = (x_1, x_2, ..., x_n)$ 和称为策略参数的标准差 $\boldsymbol{\sigma} \in R_+^{n_\sigma}$、协方差 $\boldsymbol{\alpha} \in [-\pi, \pi]^{n_\alpha}$ 组成，$\mathbf{a} = (\mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\alpha}) = (\mathbf{x}, \mathbf{s})$，$n_\sigma \in \{1,2,...,n\}$，$n_\alpha \in \{0,1,...,(2n - n_\sigma)(n_\sigma - 1)/2\}$，$\mathbf{s} = (\boldsymbol{\sigma}, \boldsymbol{\alpha})$ 统称为策略参数。个体的变异分为目标变量和策略参数变异两部分，根据 n_σ 和 n_α 的取值，可以分为以下四种情形：

1、$n_\sigma=1,n_\alpha=0$：即目标变量的每一个分量使用相同的标准差，甚至在一个种群中所有目标变量的每个分量都用同一个标准差（前者σ称为个体级策略参数，后者σ是种群级策略参数，此时σ只在代间发生变化。），变异算子表示为

$$\sigma'=\sigma\cdot\exp(\tau_0\cdot N(0,1)) \tag{3-4}$$

$$x'_i=x_i+\sigma'\cdot N(0,1) \tag{3-5}$$

其中$i=1,2,...,n$，$\tau_0\propto(\sqrt{n})^{-1}$为学习率参数。这种变异方式也属于各向同性变异，只是步长控制策略不同，$n=2$时，z的分布如图3-1（b）所示。



（a）Gauss 分布 pdf　　（b）2 维变异采样点分布

图 3-1 Gauss 分布与各向同性变异

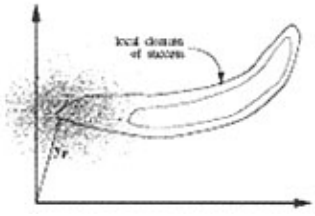
2、$n_\sigma=n,n_\alpha=0$：σ_i是基因级的策略参数，每个个体目标变量的每个分量x_i都有各自的标准差σ_i，变异算子表示为

$$\sigma_i'=\sigma_i\cdot\exp(\tau'\cdot N(0,1)+\tau\cdot N_i(0,1)) \tag{3-6}$$

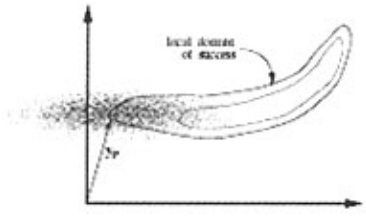
$$x_i'=x_i+\sigma_i'\cdot N(0,1) \tag{3-7}$$

其中$i=1,2,...,n$，$\tau'\propto(\sqrt{2n})^{-1}$，$\tau\propto(\sqrt{2\sqrt{n}})^{-1}$。当适应值函数的拓扑曲面如图3-2（a）所示时，各向同性变异的搜索效率不很高，如果能根据拓扑曲面的形状使用各不相同的σ_i，使z的分布如图3-2（b）所示，则可以期望得到更高的搜索效率果。变异算子式(3-6)、(3-7)就是基于这一思路设计的，它属于各向异性变异，但基因之间的变异是互不相关的。

3、$n_\sigma=n,n_\alpha=n\cdot(n-1)/2$：对于图3-3(a)所示的情况，如果使用如图3-3(b)所示的轴线旋转了的超椭球结构的变异曲面，则可以获得更高的变异效率，基于这一思想，Hansen 等[117]提出了旋转型的超椭球变异算子，此时，$\mathbf{a}=(\mathbf{x},\mathbf{s})\in I=R^n\times A_s=R^n\times R_+^{n_\sigma}\times[-\pi,\pi]^{n_\alpha}$，策略参数除了变异步长$\sigma\in R_+^{n_\sigma}$外，还

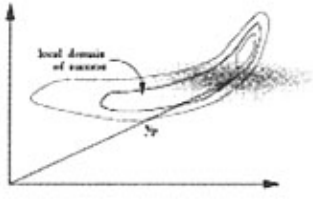


（a）各向同性变异

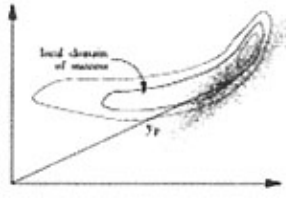


（b）各向异性变异

图 3-2 2 维各向同性与异性变异



（a）各向异性不相关变异



（b）各向异性相关变异

图 3-3 旋转型的椭球变异（相关变异）

有旋转方向参数$\alpha \in [-\pi,\pi]^{n_\alpha}$，$\sigma_i$，$\alpha_i$作为基因级策略参数，每个目标变量的每个分量都有自己的标准差σ_i和α_i旋转角，α_i表示了变异算子的超椭球面与坐标轴间的角度，它使目标变量各分量的变异互相关联，这种变异算子也称为相关变异算子，而前两种称为不相关变异算子。变异算子表示为

$$\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$$

$$\alpha'_j = \alpha_j + \beta \cdot N_j(0,1) \tag{3-8}$$

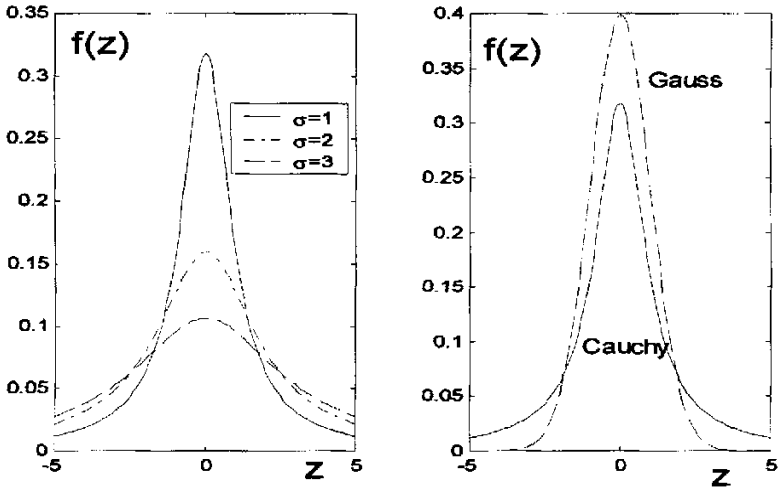
$$\mathbf{x}' = \mathbf{x} + N(\mathbf{0}, \mathbf{C}(\boldsymbol{\sigma}', \boldsymbol{\alpha}')) \tag{3-9}$$

其中$i = 1,2,...,n$，$j = 1,2,...,n_\alpha$，$\beta \approx 0.0873$。$\mathbf{z}$的pdf可表示为：

$$p(\mathbf{z}) = \frac{1}{(\sqrt{2\pi})^n} \frac{1}{\sqrt{\det \mathbf{C}}} \exp(-\frac{1}{2}\mathbf{z}'\mathbf{C}^{-1}\mathbf{z})$$

由于这种变异算子参数众多，计算量大，在算法执行过程中，一般没有适应值函数的先验知识，实际使用时很难“碰巧”得到所需要的变异方向，使用较少。

4、$1<n_\sigma<n$：与$n_\sigma=n,n_\alpha=0$类同。



(a) Cauchy 分布 *pdf*　　(b) Cauchy 与 Gauss 分布 *pdf* 比较

图 3-4 Gauss 分布、Cauchy 分布 *pdf*

Gauss 分布比较集中在均值附近，不利于跳出局部极值点。姚新[37]提出了 Cauchy 变异，由于 Cauchy 分布具有较强的分散性，其 *pdf* 具有较大的拖尾，如图 3-4（a）所示，有利于提高算法的全局搜索能力。

$$\mathbf{z}=\sigma(C_1(0,1),C_2(0,1),...,C_n(0,1))$$

$z_i\sim C_i(0,\sigma)$是相互独立的 Cauchy 分布随机变量，$C(0,\sigma)$的 *pdf* 为：

$$f_C(z)=\frac{1}{\pi}\frac{\sigma}{\sigma^2+z^2} \tag{3-10}$$

图 3-4（b）比较了 Gauss 分布与 Cauchy 分布的 *pdf*，可见，Gauss 分布比较集中于均值附近，而 Cauchy 分布具有较强的分散性。

王云诚[39]提出了均匀变异：

$$\mathbf{z}=\sigma(U_1(-1,1),U_2(-1,1),...,U_n(-1,1))$$

$z_i\sim\sigma U_i(-1,1)$是相互独立的均匀（uniform）分布随机变量，$U(-1,1)$的 *pdf* 为：

$$f_U(z)=\begin{cases}\dfrac{1}{2} & -1\le x\le 1\\ 0 & others\end{cases} \tag{3-11}$$

这几种变异方式互相结合可以构成新的变异算子，如文献[117]提出了平均变异算子（mean mutation operator）

$$z_i=\frac{\sigma}{2}(C(0,1)+N(0,1)) \tag{3-12}$$

文献[119]提出了组合变异算子（combined mutation operator）：

$$z_i=\begin{cases}\sigma N(0,1) & for\ \ 0\le k\le\alpha\\ \sigma C(0,1) & for\ \ \alpha\le k\le 1\end{cases} \tag{3-13}$$

其中，$k=random(0,1)$，α 是选择 Gauss 变异与 Cauchy 变异的概率因子。

林丹等[38]系统地分析了 Gauss 分布、Cauchy 分布、均匀分布变异算子的全局搜索能力和局部搜索能力，理论和试验表明，Gauss 分布算子具有较强的局部搜索能力，而均匀分布变异具有较强的全局搜索能力，Cauchy 分布算子的局部搜索能力和全局搜索能力居中。

3.2 单基因变异与全基因变异

如上所述，在实数搜索空间，ES 的变异算子都使用了全基因变异方式，即在繁殖子代时，子代个体的每个目标变量基因从父个体基因变异产生。设 $\mathbf{x}=(x_1,x_2,...,x_n)$ 为父个体，其子个体 $\mathbf{x}'=(x_1',x_2',...,x_n')$，$\sigma_i'$ 是变异步长。CES 的全基因变异方式为

$$x_i'=x_i+\sigma_i'\cdot N(0,1) \tag{3-14}$$

$i=1,2,...,n$，σ_i' 可以是所有基因相同，也可以是各个基因互异。

与全基因变异相对应，作者提出了一种单基因变异方式[134]，在通过变异繁殖子代时，只从父个体目标变量基因中，随机选择其中一个发生变异，其他基因直接从父个体基因继承、复制，表示如下：

$k=Random(1,n)$

$$x_i'=\begin{cases}x_i+\sigma'\cdot N(0,1) & if\ i=k\\ x_i & if\ i\ne k\end{cases} \tag{3-15}$$

本文的余下部分将对单基因变异作分析，建立相应的算法。本章以下部分

将分析单基因变异与全基因变异的成功概率，以比较它们的局部搜索能力。下一章将结合变异步长、收敛速度进一步分析单基因变异算子的局部搜索能力和全局搜索能力，建立合适的步长控制策略和相应的算法。

3.2.1 成功变异的概率分析

使用 Gauss 变异算子时，考察式(3-14)、(3-15)，父个体 $\mathbf{x}=[x_1,x_2,...,x_n]$，经变异产生的子个体 $\mathbf{x}'=[x'_1,x'_2,...,x'_n]=[x_1+z_1,x_2+z_2,...,x_n+z_n]$，变异矢量 $\mathbf{z}=[z_1,z_2,...,z_n]$ 是零均值且方差为 $\mathbf{C}=\sigma^2\mathbf{I}$ 的 Gauss 正态分布.

定义 3.1　成功变异　如果变异后产生的子个体与父个体相比得到了改进，则称为成功变异。成功变异的概率简称为成功概率。

在这里，“改进”可以有两个方面，其一为 $\mathbf{x}'$ 与极值点 $\mathbf{x}^*$ 的距离减小；其二为个体适应值得到了改善，对于求最大值的优化问题，$f(\mathbf{x}')>f(\mathbf{x})$。这两者虽然不完全等价，但当分析局部搜索能力时，可以粗略地认为它们是等价的。

单基因变异的成功概率 p_s

设父个体 $\mathbf{x}=[x_1,x_2,...,x_n]$ 的所有基因都是可进化的（即没有收敛到极值点），不失一般性，为了叙述方便，对单基因变异，设随机选择了 x_1 作为变异基因，其他基因不变，即 $z_2=...=z_n=0$，考虑图 3-5（a），单基因变异后，$f(\mathbf{x}')>f(\mathbf{x})$ 的概率（称之为成功概率）为：

$$P_s=P(f(\mathbf{x}')>f(\mathbf{x}))=P(0<z_1<\delta_1)) \tag{3-16}$$

由于 z 是零均值的正态分布随机变量，$P_s<0.5$。

全基因变异的成功概率 p_a

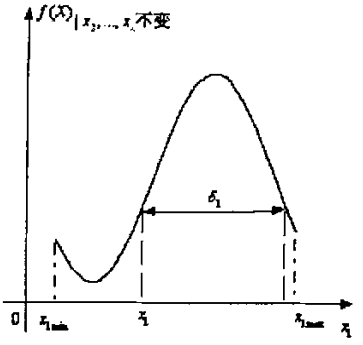
对于全基因变异，把全变异产生的适应值函数变化分解为两部分，即

$$\begin{aligned} f(\mathbf{x}') &= f(x_1+z_1,x_2+z_2,...,x_n+z_n) \\ &= f(x_1+z_1,x_2,...,x_n)+\Delta f \end{aligned}$$

Δf 是由于 $x_2,...,x_n$ 变异产生的适应值变化，考虑图 3-5（b）和（c），全基因变异的成功概率 p_a 为：

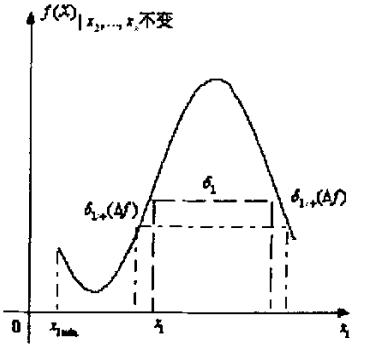
$$\begin{aligned} P_a &= P(f(\mathbf{x}')>f(\mathbf{x})) \\ &= P(\Delta f>0)\cdot P(-\delta_{1l+}(\Delta f)<z_1<\delta_1+\delta_{1r+}(\Delta f)) \\ &\quad + P(\Delta f\le 0)\cdot P(\delta_{1l-}(\Delta f)<z_1<\delta_1-\delta_{1r-}(\Delta f)) \\ &= P(\Delta f>0)\cdot(P_s+\Delta P_+)+P(\Delta f\le 0)\cdot(P_s-\Delta P_-) \end{aligned} \tag{3-17}$$

上式中：



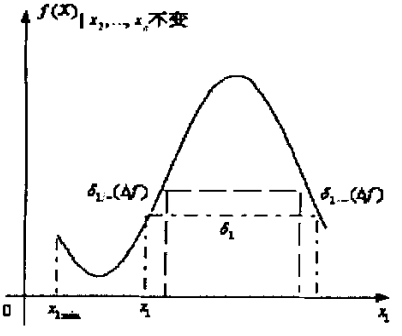
(a)单基因变异的进化区间为

$$(x_1, x_1+\delta_1)$$



（b）全基因变异且$\Delta f > 0$时，进化区间为

$$(x_1-\delta_{1l+}(\Delta f), x_1+\delta_1+\delta_{1r+}(\Delta f))$$



（c）全基因变异且$\Delta f < 0$时，进化区间为

$$(x_1+\delta_{1l-}(\Delta f), x_1+\delta_1-\delta_{1r-}(\Delta f))$$

图 3-5 不同变异方式的进化区间分析

$$\Delta P_+ = P(-\delta_{1l+}(\Delta f) < z_1 \le 0 \,\|\, \delta_1 \le z_1 < \delta_1 + \delta_{1r+}(\Delta f))$$

$$\Delta P_- = P(0 < z_1 \le \delta_{1l-}(\Delta f) \,\|\, \delta_1 - \delta_{1r-}(\Delta f) \le z_1 < \delta_1)$$

显然，$\Delta P_- \le P_s$，当变量数n较大时，$n-1$个变量变异时的成功概率与n个变量变异时的成功概率相同，即$P(\Delta f > 0) = P(f(\mathbf{x}') > f(\mathbf{x})) = P_a$，所以，$P(\Delta f \le 0) = 1 - P_a$，代入式(3-17)式整理可得：

$$P_a = \frac{P_s - \Delta P_-}{1 - \Delta P_+ - \Delta P_-} \tag{3-18}$$

为了直观起见，可设$\Delta P_+ = \Delta P_- = \Delta P$，考察

$$dP = P_a - P_s = \frac{\Delta P(2P_s - 1)}{1 - 2\Delta P} \tag{3-19}$$

由于 $P_s < 0.5$ 和 $\Delta P < 0.5$，所以，$dP < 0$，即单基因变异的成功概率大于全基因变异的成功概率。

另外，从式(3-18)可见，当 $\Delta P_- \to P_s$ 时，$P_a \to 0$，全基因变异不能进化，将发生进化停顿。

以上理论分析的结果可以推广到所有变异算子为零均值概率分布的情形，如均匀分布、Cauchy 分布。可以得到如下定理：

定理 3.1 在相同条件下，当使用零均值概率分布的变异算子时，单基因变异的成功概率不小于全基因变异的成功概率。

推论 3.1 当单基因变异的成功概率比较小，$P_s \to \Delta P_-$ 时，全基因变异的成功概率趋于 0，将出现进化停顿。

3.2.2 局部搜索能力仿真分析

CES 使用全基因变异，一般认为，这样有利于从各个方向搜索，事实上，当问题规模比较大（$n \geq 10$）时，由于变异方向的随机性，往往使变异的效果互相抵消，很难得到进化了的后代个体，导致 CES 在求解大规模问题时的收敛速度很低，甚至停顿，而且这种变异的计算开销也大.

本节通过仿真实例比较全基因变异与单基因变异的局部搜索能力.

为了验证定理 3.1，以最常用的球模型函数：

$$f(\mathbf{x}) = -\sum_{i=1}^{n} x_i^2$$

求最大值为例进行了仿真计算，计算条件为：使用 $N(0,1)$ 恒标准差 Gauss 变异

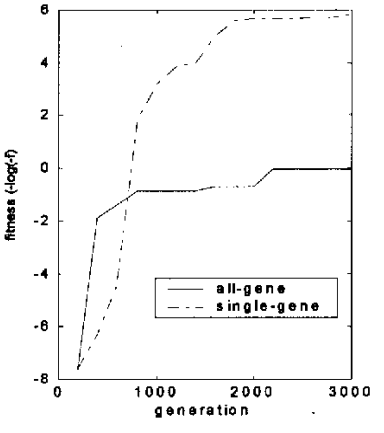
表 3-1 不同变异方式，$n = 10$ 时，适应值随进化代数的变化

进化代数	全基因变异	单基因变异
200	-2024.4926	-1964.4423
400	-6.678599	-562.37433
600	-3.985328	-86.178448
800	-2.345735	-0.1542389
1000	-2.345735	-0.0400843
1200	-2.345735	-0.0206491
1400	-2.345735	-0.0183496
1600	-1.995233	-0.0066030
1800	-1.995233	-0.0037267
2000	-1.995233	-0.0034335
2200	-1.029373	-0.0034335
2400	-1.029373	-0.0034335
2600	-1.029373	-0.0032608
2800	-1.029373	-0.0032608
3000	-1.029373	-0.0029491

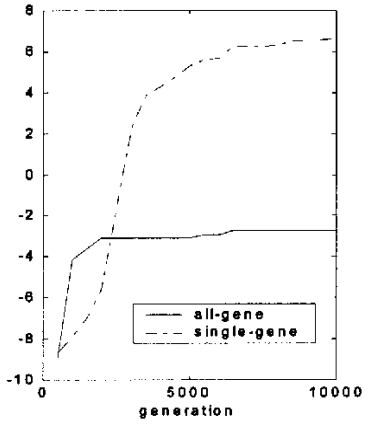
算子，$(1+1)-ES$。仿真计算分两方面，其一分别计算$n=10,30,50$不同变异方式时适应值的变化；其二为$n=10$在不同进化阶段，发生 10 000 次变异，统计进化次数，得到成功概率。图 3-6（a）、（b）、（c）分别为$n=10,30,50$时的适应值一进化代数变化曲线，图中，纵坐标（适应值）做了取对数处理$y=-\log(-f(\mathbf{x}))$，以方便显示；图 3-6（d）为使用 4.2.4 节介绍的横向仿真方法，在不同进化阶段（代）分别作 10000 次全基因和单基因变异，统计变异成功次数，得到时成功概率一进化代数曲线（$n=10$）。为了更清楚地考察适应值的变化，表 3-1 列出了$n=10$时适应值的变化。

从图 3-6（a）～（c）中，可以明显的观察到，除了在进化的初始阶段，全基因变异的适应值优于单基因变异外，在进化的中后期，单基因变异的适应值优于全基因变异。从图 3-6（d）可以看到，随着进化过程的延续，适应值增大，全基因变异的成功概率显著下降，接近于 0，最终导致进化停顿，而单基因变异，不出现进化停顿现象。当然，可以通过减小变异步长（标准差），从而增大P_s和P_a，防止$P_a \to 0$。

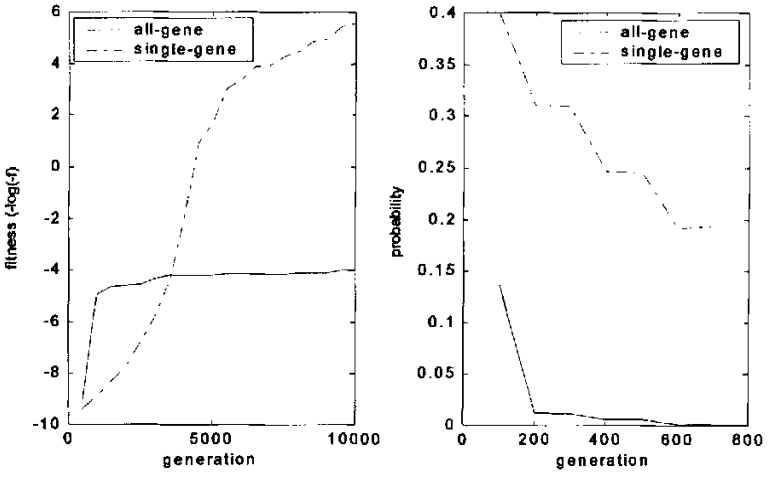
需要指出的是，在进化起始阶段，当变异步长比较“合适”时，全基因变异的进化速度大于单基因变异，但单基因变异在适应值变化的很大范围内都可以获得良好的进化速度，说明单基因变异对变异步长的要求相对较低，而全基因变异要求步长足够小，才能得到良好的进化性能，在第四章将论述步长控制策略，进一步分析不同步长时的进化效率。



(a)适应值变化曲线（$n=10$）



（b）适应值变化曲线（$n=30$）



(c) 适应值变化曲线（$n=50$）　　(d) 进化概率变化曲线

图 3-6　全基因变异与单基因变异的局部搜索能力比较

3.3 关于变异方式的进一步探讨

3.3.1 计算开销比较

从式(3-14)、(3-15)可以看出，全基因变异方式下，每个目标基因都变异，需要调用n次 Gauss 概率分布函数，Gauss 函数$N(\mu,\sigma)$实现的一般方法[123]为

$$x_1=\mu+\sigma\sqrt{-2\ln r_1}\cos(2\pi r_2) \tag{3-20}$$

$$x_2=\mu+\sigma\sqrt{-2\ln r_1}\sin(2\pi r_2) \tag{3-21}$$

其中，r_1,r_2为（0，1）间均匀分布随机数，x_1,x_2为互相独立的$N(\mu,\sigma)$分布随机变量。其 C 语言实现典型代码[124]如 Algorithm 3.2 所示。

完成一次全基因变异需要调用n次 Gauss 函数，所需要的计算量相当于调用一次中等复杂程度的适应值函数，而单基因变异只需要调用一次 Gauss 函数，计算开销明显小于全基因变异，n越大，单基因变异节省计算开销的效果越显著。

单基因变异成功概率大而且计算开销小，与全基因变异相比具有明显的优势，为什么一直以来，没有人提出单基因变异方式（至少，作者所查的文献没有关于单基因变异的研究）？这让人感到困惑，促使作者对变异方式作进一步探讨，以下试图从 H.Bremermann 早期研究、进化策略最初的变异思想解释。

Algorithm 3.2： $N(\mu,\sigma)$ 的 C 语言实现

```
double Gauss(double u,double sigma)
{
 double t,rndx1,r1,r2;
 static double rndx2;
 static int flag=1;
 if(flag)
 {
     r1=(double)(rand()+1)/RandMax;
     r2=(double)(rand()+1)/RandMax;
     rndx1=sqrt(-2*log(r1));
     t=2*Pi*r2;
      rndx2=rndx1*sin(t);
     flag=0;
      return u+sigma*rndx1*cos(t);
 }
 else
 {
     flag=1;
     return u+sigma*rndx2;
 }
}
```

3.3.2 H.Bremermann 的早期研究

从上世纪 50 年代末到 1996 年去世，H. Bremermann 对进化计算做出了很大的贡献，他是进化计算的早期开拓者之一。早在 1958 年，他就开始了模拟生物进化的优化计算试验，他把生物的进化看作是一个优化的过程，并在个体学习与进化学习之间建立了联系，并寻求以个体环境模型解释神经系统。这个基本模型使用了简单的变化和选择模式，与单亲本、单后代进化算法完全相同，Bremermann 发表了题为“The evolution of intelligence”的技术报告（H.J.Bremermann. The evolution of intelligence. The nervous system as a model of its environment. Technical report, no.1, contract no. 477(17), Dept. Mathematics,

University Washington, Seattle, July,1958）。1962 年，Bremermann[4]把这一模型扩展为包含个体种群，并研究求函数极小值的优化计算问题，在算法试验中，为了使进化过程易于分析，Bremermann 以解线性方程组 $\mathbf{Ax}-\mathbf{b}=\mathbf{0}$（其中 $\mathbf{A}$ 是 $n\times n$ 矩阵，$\mathbf{b}$ 是 $n\times 1$ 列矢量）为研究对象，把解方程转化为求函数 $f(\mathbf{x})=\|\mathbf{Ax}-\mathbf{b}\|_2$ 最小值的优化计算问题。

在 Bremermann 的试验中，首先把变量离散化，每次只有其中一个元素（基因）变异，改变增量（步长）Δx_j。对于条件数较好的（最大特征值与最小特征值之比接近于 1），并且具有变量范围的某些预先知识和 n 不大（不大于 10）方程组，此方法在几分钟内得到了近似解，但对于坏条件数方程组，进化过程陷入停滞点，减小变异步长，可以进一步进化，但又陷入新的停滞点。

当时，Bremermann 用图 3-7 解释了停滞的原因。他认为，一次改变一个变量，相当于沿坐标轴移动，对于特定的步长，进化到某个位置后，陷入进化停滞状态；而同时改变所有基因，将陷入组合爆炸；以概率改变每个基因，他认为对克服停滞效果不佳。

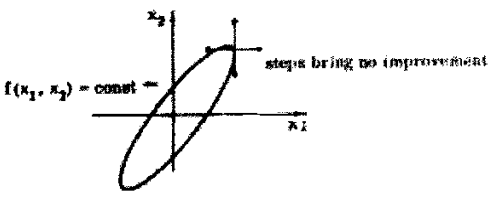


图 3-7　沿坐标轴改变特定长度，不能改进结果

Bremermann 的试验结果使他很失望，Atmar[5][125]称之为“Bremermann 失望”（Bremermann's disappointment）。

Bremermann 的试验，一次只有一个基因变异，可以说是最早的单基因变异方式，与 CES 的变异步长和变异方式不同，他的变异是基因按照一定的变异步长直接改变，使他得到了失望的结果，我想，这也为后人研究单基因变异设置了一道鸿沟，以后的遗传算法使用了以一定的变异概率改变所有基因的变异方式。

3.3.3 进化策略的变异思想

Rochenberg 和 Schwefel 提出进化策略 ES 的原始思想是借鉴生物进化原理实现技术系统的优化[10][11][65][126]。按照现有生物进化理论和基因突变原理，在生物界，变异同时发生在个体所有的基因，而不会仅仅变异其中的某个基因，所有的基因同时变异似乎最符合自然界的进化规律，ES 遵循了这一规律。ES 重视基因的表现型特性（phenotype）而非染色体的基因型（genotype）特性，以个体的形式对待目标变量 $\mathbf{x}$，在父本繁殖子代时所有参数同时改变.

在以后的 ES 研究、改进过程中，人们对如何提高进化的效率即局部搜索能

力非常重视，如果能引导搜索过程直接朝目标点方向移动，自然搜索效率最高。在 T. Bäck 的著作《Evolutionary Algorithms in Theory and Practice》([21],p70)中，描述了理想的变异图形。如图 3-1（b）所示，当所有基因使用相同的变异步长时，即 $n_\sigma=1,n_\alpha=0$，按式(3-4)、(3-5)变异时，等概率密度线是超球面，搜索效率不高；当基因的变异步长不等，$n_\sigma=n,n_\alpha=0$，按式(3-6)、(3-7)变异时，等概率密度线是轴线平行于坐标轴的超椭球面；更理想的变异方式是轴线旋转了的超椭球面，这时，$n_\sigma=n,n_\alpha=n\cdot(n-1)/2$，按式(3-6)、(3-8)、(3-9)变异，这种改进变异算子的思维方式在 ES 研究中占据着统治地位。

诚然，如果能按照这样的超椭球面变异，进化的效率将是很高的，但问题是如果没有有关适应值函数拓扑结构的先验知识，能有多少机会按照这样的理想超椭球面变异？所以，作者的思路是退而求其次，使用单基因变异方式，力求提高变异成功的概率，积跬步以至千里吧。

3.3.4 有关仿真计算结果的评价

最后需要说明的是，如果取合适的变异步长，全基因变异的效率（平均每次变异使适应值函数值的改进量）确实会高于单基因变异，从图 3-6（a）、（b）、（c）可以得到证实，在进化初期，由于变异步长 $\sigma=1$，相对较小，全基因变异方式时，适应值的改进速度高于单基因变异，从表 3-1 可以更清楚地看出，在 600 代以前，全基因变异的适应值大于单基因变异的适应值。在第四章，将给出进一步的仿真结果，对于一些较“简单”的适应值函数，如球模型函数，由于容易找到合适的变异步长，全基因变异时，收敛速度高于单基因变异。这也许是全基因变异受到重视，而单基因变异无人研究的另一个原因，因为很多研究特别是理论研究都以球模型函数为研究对象。

3.4 小结

本章提出了一种新的变异方式——单基因变异。通过理论分析证明，对于相同的变异步长，使用相同分布的零均值随机变量为变异量时，单基因变异的成功概率不小于全基因变异成功概率，同时也证明了全基因变异可能出现进化停顿。仿真计算试验表明，对于同一变异步长，单基因变异的局部搜索能力强于全基因变异，能进化到更接近于极值点，验证了理论分析的正确性。就计算开销而言，单基因变异小于全基因变异。从分析变异方式研究的历史，探索了单基因变异方式被忽视的原因。

当变异步长一定，随着进化过程的延续，种群中的优秀个体逐步接近极值

点，会出现进化减缓，甚至进化停顿，导致进化效率下降，需要及时修正变异步长，以跟踪进化进程，提高进化效率。单基因变异由于一次只有一个基因发生变异，其全局搜索性能怎样？应该采用怎样的变异步长控制策略？将在下一章分析。

第四章　单基因变异 ES 的步长控制

变异步长对成功变异的概率（简称为成功概率）有直接的影响，随着进化进程的延续，优秀的个体逐步逼近极值点，成功概率也逐渐减小，此时，需要适当地减小变异步长，以跟踪进化进程；另一方面，过小的变异步长虽然可以增大成功概率，使之接近 0.5，但也会使进化步伐很小，进化效率很低，合理的变异步长控制可以有效地提高算法的效率；变异步长也会对算法的全局收敛性产生影响，不合理的步长可能导致局部早熟收敛。

本章首先概述 ES 的步长控制策略，分析步长与局部搜索能力的关系、步长控制策略对全局收敛性的影响，提出适应于单基因变异的步长控制策略，并分析全局收敛性，以及增强全局搜索能力的措施，建立基于单基因变异的进化策略 $(\mu+\lambda+\kappa)-ES$，最后，给出一组高维优化测试函数（100 维）的仿真计算结果。

4.1 变异步长控制概述

进化策略 ES 中，子代的产生主要依靠变异算子，影响变异算子性能的主要因素有随机变量的分布形式（如 Gauss 分布、Cauchy 分布、均匀分布及它们的组合）、变异步长的控制方式、本文第三章提出的变异方式（单基因变异或全基因变异）等。变异步长是变异算子中最重要的参数之一，对算法的性能有直接的影响。

研究者们为与算法本身有关的策略参数建立了几种分类方法。Angeline[127] 根据参数的作用范围和控制机制进行分类，算法的参数按其作用范围分为三级：种群级、个体级、基因级（component level、gene level），如变异步长，当种群中所有个体使用同一个变异步长时，这个变异步长就是种群级的，如果每个个体都使用各自不同的变异步长，则为个体级，而在各向异性的变异算子中，个体的每个目标变量（基因）都有各自的变异步长，此时，变异步长参数就是基因级的。Angeline 把控制参数的机制分为绝对型（absolute）和经验型（empirical），前者使用预先确定的控制方式，如变异步长随进化代数以某种函数关系变化，后者通过个体在生存竞争中按照适者生存的原则，在竞争中确定，是一种自适应型机制。Hinterding[128] 扩展了 Angeline 的分类方法，增加了环境级（environment

level)，而把参数的更新机制分为确定型（deterministic）、适应型（adaptive）、自适应型（self-adaptive），如种群规模有关的参数 μ,λ 在算法执行前就已经确定，是确定型参数，而 1/5 成功准则中步长的控制策略，根据变异成功率反馈修改变异步长，是适应型控制方式，当变异步长作为个体基因的一部分参加变异、进化时，这种参数控制方法为自适应型。Smith 和 Fogarty[129][130]根据控制的参数、作用范围、控制机制提出了分类更复杂方法。Eiben [82]在 Hinterding 的基础上，根据参数控制的机制和所控制的参数的作用范围提出了较为简单、实用的分类方法。根据参数的更新机制，按图 4-1 分类，而根据参数的作用范围分为种群级、

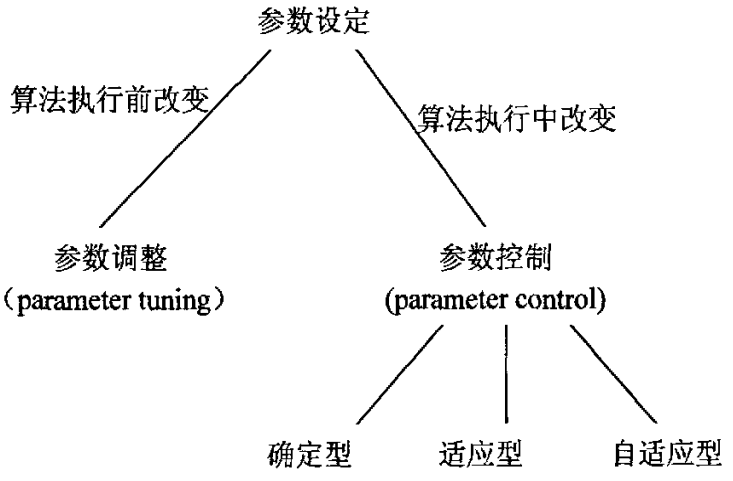


图 4-1　算法参数设定方式分类

个体级、次个体级（sub-individual）。

按 1/5 成功规则的适应型变异步长控制策略[10]、自适应变异步长的控制方式[65][21]是 ES 的经典方式，除此以外，递减型[39][131]控制方式也是一种常用的适应型步长控制方式，这种方式根据变异成功概率不断地缩减步长，避免了 1/5 规则中步长的摆动。

4.2 变异步长与局部搜索性能的关系

在 ES 研究历史上，Rochenberg 的 1/5 成功率原则是在全基因变异 $(1+1)-ES$ 的基础上提出的，在进化过程中，变异步长常常上下摆动，降低了算法的局部搜索能力，自适应进化策略将变异步长等策略参数作为个体基因的一部分参加

进化，由于变异步长具有一定的随机性，给进化过程带来很多不确定性，如σ按式（3-6）计算，并限定在（0.00001，3）的范围内，作 2000 次计算试验，得到的步长如图 4-2 所示，可见σ具有较大的随机性，并不能随着进化进程变化，只

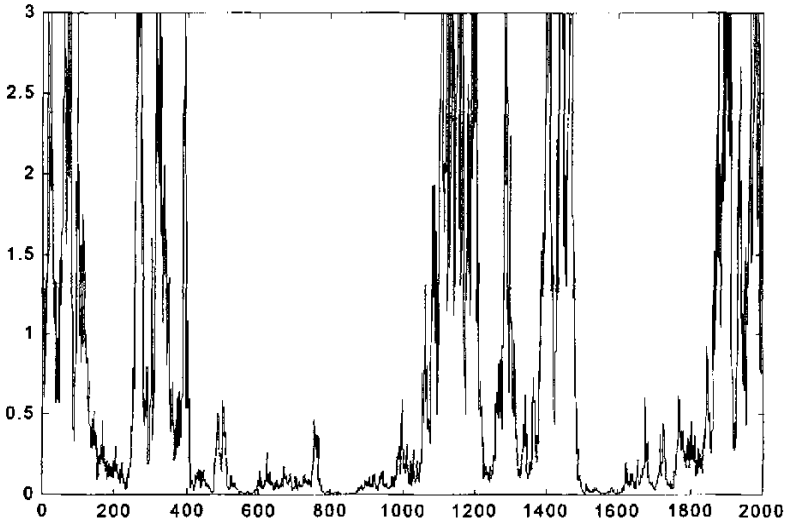


图 4-2 自适应型σ的变化曲线

是其中常出现大的值，有利于增强全局搜索能力。很多实际计算表明[131]，变异步长不能很好地跟踪进化进程，降低了 ES 的计算效率。

以下将研究单基因变异方式下，变异步长与进化效率（局部搜索能力）之间的关系，建立相应的步长控制策略。

4.2.1 理论分析

为了分析单基因变异 ES 的变异步长与进化效率之间的关系，以$(1+1)-ES$为分析对象。

定义 4. 1　改进率（progress rate）　改进率φ为变异一代后，目标变量与极值点距离的减小值的数学期望。

由于此处分析单基于变异的改进率，所以，用单变量问题分析。参考图 3-5（a），单基因变异时成功概率如式(3-6)所示，当使用 Gauss 变异时，$z \sim N(0,\sigma)$，改进率为：

$$\varphi = \int_0^{\delta} z f_Z(z)dz + \int_{\delta}^{2\delta} (2\delta - z) f_Z(z)dz$$

$$= \int_0^{\delta} z\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{1}{2}\frac{z^2}{\sigma^2})dz + 2\delta\int_{\delta}^{2\delta}\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{1}{2}\frac{z^2}{\sigma^2})dz$$

$$-\int_{\delta}^{2\delta} z\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{1}{2}\frac{z^2}{\sigma^2})dz$$

$$= \frac{\sigma}{\sqrt{2\pi}}(1-2\exp(-\frac{\delta^2}{2\sigma^2})+\exp(-\frac{2\delta^2}{\sigma^2}))+2\delta\int_{\delta}^{2\delta} f_Z(z)dz \tag{4-1}$$

上式中 $f_Z(z)$ 为随机变量 z 的 *pdf*，它为 Gauss 正态分布时，$z\sim N(0,\sigma)$，对不同的 δ 值，作 $\varphi-\sigma$ 曲线，如图 4-3 所示，由图可见，对于不同的 δ，存在相应的 $\sigma=\sigma_{opt}$，使 φ 最大。如果使用 $d\varphi/d\sigma=0$ 求 $\varphi_{\max}$ 将需要解超越方程，不能得到解析解，为此，给定一系列 δ 值，分别求 σ_{opt}，得到 $\sigma_{opt}-\delta$ 关系曲线，如图 4-4 所示，近似为一条经过原点的直线，用线性回归方法，得到

$$\sigma_{opt}\approx 1.25\delta \tag{4-2}$$

$$\varphi_{\max}\approx 0.288\sigma_{opt} \tag{4-3}$$

当 $\sigma=\sigma_{opt}=1.25\delta$ 时，成功概率为

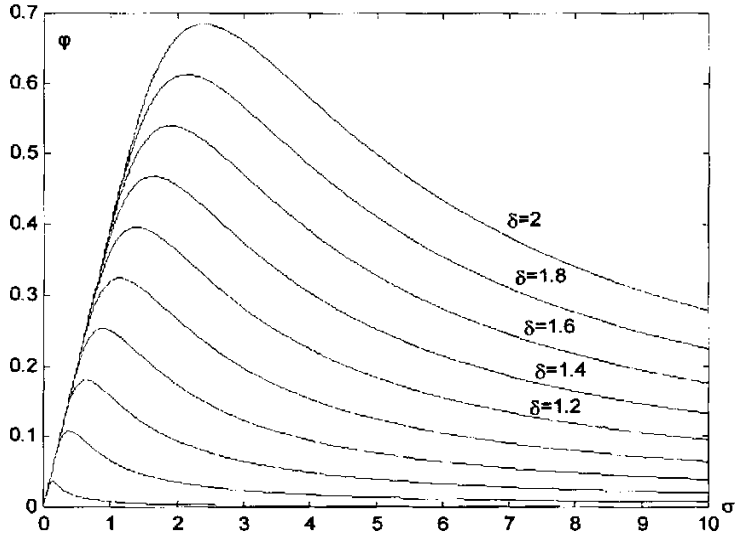


图 4-3 进化期望与 Gauss 分布变异步长关系曲线（$\varphi-\sigma$ 曲线）

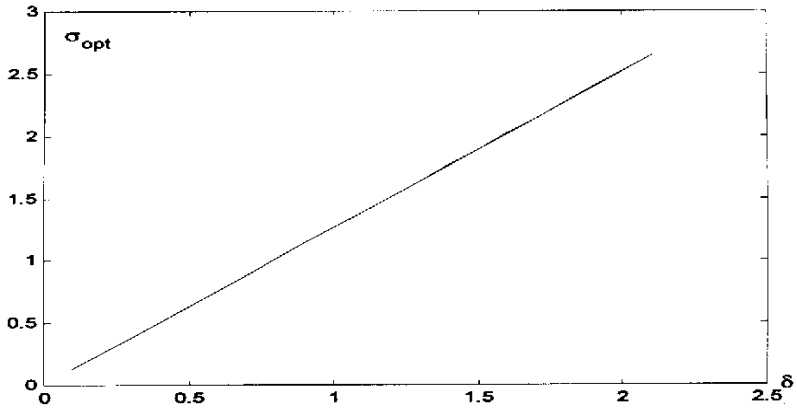


图 4-4 最优变异步长σ_{opt}与δ关系（$\sigma_{opt}-\delta$）

$$p_e = P(0 < z < 2\delta)$$

$$= \int_0^{2\delta} \frac{1}{\sigma_{opt}\sqrt{2\pi}} \exp(-\frac{1}{2}\frac{z^2}{\sigma_{opt}^2})dz$$

$$= \int_0^{\frac{2\sigma_{opt}}{1.25}} \frac{1}{\sigma_{opt}\sqrt{2\pi}} \exp(-\frac{1}{2}\frac{z^2}{\sigma_{opt}^2})dz$$

$$= \int_0^{1.6\sigma_{opt}} \frac{1}{\sigma_{opt}\sqrt{2\pi}} \exp(-\frac{1}{2}\frac{z^2}{\sigma_{opt}^2})dz$$

$$= \Phi(1.6) - \Phi(0)$$

$$\approx 0.445 \qquad (4\text{-}4)$$

上式中：

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}z^2)dz$$

为标准正态分布的累积概率分布函数 *cdf* 。

对图 4-3 作归一化处理，$\varphi^* = \varphi/\varphi_{\max}$，$\sigma^* = \sigma/\sigma_{opt}$，可得到图 4-5 所示的不同$\delta$时，$\varphi^*-\sigma^*$、$p_e-\sigma^*$关系曲线族，由图可见，在不同的$\delta$时，$\varphi^*-\sigma^*$关系曲线非常地相似，几乎重合。

由以上分析可见，当成功概率控制在 0.445 左右时，可以得到最大的进化期望。取进化窗在$p_e = 0.3 \sim 0.48$范围内，进化期望将大于$0.8\varphi_{\max}$，可以考虑当成功概率低于 0.3 时，减小变异步长为原来的一半。另外，由图 4-3 或图 4-5 可见，

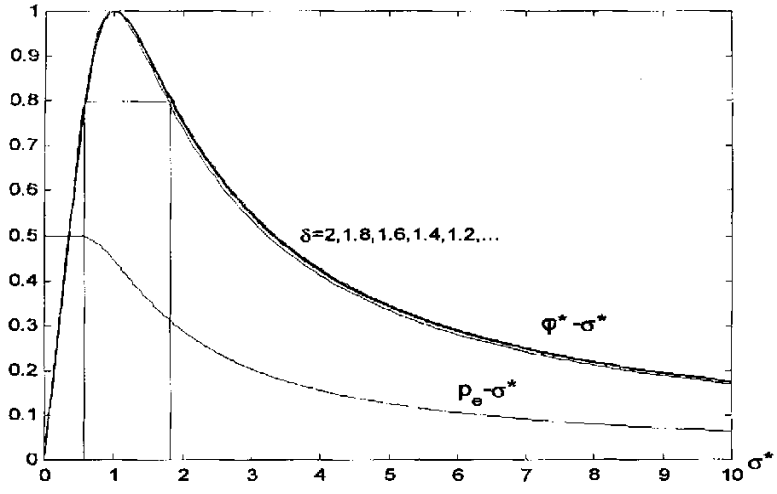


图 4-5　Gauss 变异算子，不同 δ 时的 $\varphi^*-\sigma^*$、$p_e-\sigma^*$ 关系曲线

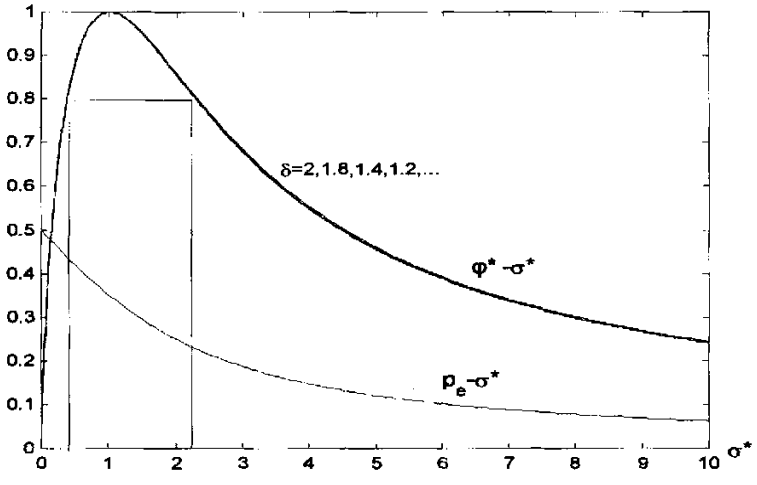


图 4-6　Cauchy 变异算子，不同 δ 时，$\varphi^*-\sigma^*$、$p_e-\sigma^*$ 关系曲线

当 σ 在大于 σ_{opt} 的较大范围内，进化期望比较大，而当 σ 在小于 σ_{opt}，进化期望值下降比较陡峭，所以，在实际计算过程中，尽管很难得到 σ_{opt}，即使比 σ_{opt} 大点，也能得到良好的进化期望。

当使用 Cauchy 分布变异时用类似的方法得到

$$\left.\begin{aligned}&\sigma_{opt}\approx\delta\\&\varphi_{\max}\approx 0.256\sigma_{opt}\\&p_e\approx 0.352\end{aligned}\right\}\tag{4-5}$$

分析图 4-6，可以取进化窗为 $p_e=0.24\sim0.44$，当成功概率小于 0.24 时，减小变异步长为原来的一半。式(4-5)与式(4-2)、(4-3)、(4-4)比较，Cauchy 分布变异需要较小的 σ，进化期望和成功概率都比 Gauss 分布时小，说明 Cauchy 分布的局部搜索能力较 Gauss 差，所以，在本文的单基因变异算子中，使用了 Gauss 分布变异。

由图 4-5 和图 4-6 可见，当 $\sigma^*=0.5\sim2$ 时，$\varphi^*>0.7$，即 $\sigma=(0.5\sim2)\sigma_{opt}$ 时，$\varphi>0.7\varphi_{\max}$，说明在单基因变异时，$\sigma$ 的取值在较大范围内，都有良好的进化性能，具有良好的鲁棒性。

4.2.2 横向仿真分析

在进化算法的仿真研究中，为了检验一种进化算子的有效性，常以一组典型测试函数为优化对象，观察算法作用在这组测试对象时的进化代数、收敛性能、计算精度等，这是一种从进化计算的全过程考察算法效果的方法，可以把这种仿真方法称为纵向仿真方法。进化算法在执行过程中，由于参数多、随机性强，仅从纵向考察某个算子的作用，往往由于众多随机因素和其他参数的影响，使结果不够准确。作者提出了一种横向仿真分析法，作为考察某种进化算子的补充，与纵向仿真相结合，容易得到较为直观的结果。

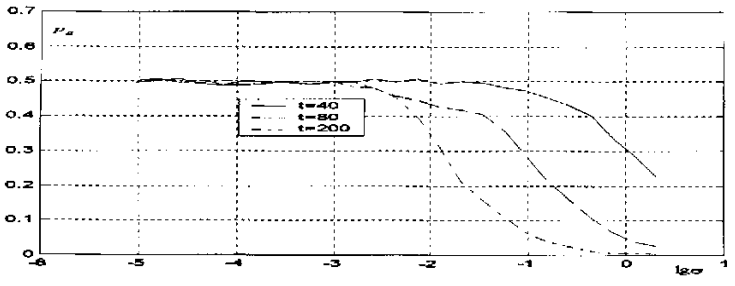
所谓横向仿真，是在进化进程的不同阶段（代），嵌入某个仿真子进程，以测试某个算子的作用效果，研究该算子的参数设置，它对进化进程的多个横断面进行考察、仿真、分析，故称为横向仿真。

在本节，将应用横向仿真方技术研究变异步长与局部搜索能力的关系，作为对前一小节结果的验证，并与全基因变异比较。实现方法如下：

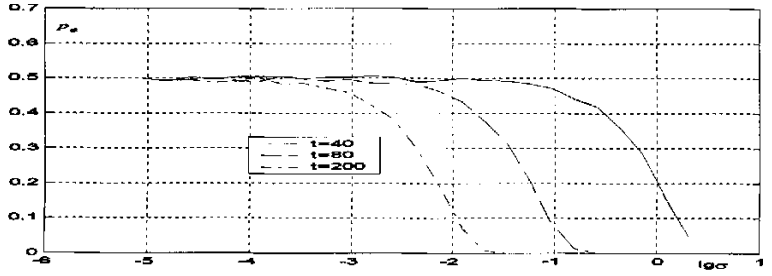
以单峰球模型函数 $f_3(\mathbf{x})$ 和多峰函数 $f_6(\mathbf{x})$（见 4.5.2）求最大值问题作仿真计算，在进化的不同阶段（代），以不同的变异步长分别作 10000 次单基因变异和全基因变异，统计分析、比较这两种变异方式的变异成功概率、适应值的累计进化幅度与变异步长的关系。

成功概率与变异步长关系

以 CES 为原型，仿真 10 维球模型函数在不同进化阶段（$t=40,80,200$）时，分别试验 σ 从 2.0 不断减小，直到 0.000001，每一个 σ 值变异 10000 次，统计计算发生进化的概率，得到成功概率 $p_e-\sigma$ 关系曲线，如图 4-7 所示，由图可见，



(a)单基因变异



(b)全基因变异

图 4-7 变异成功概率与变异步长的关系

σ 不断减小时，成功概率逐步接近 0.5；比较图 4-7 (a)、(b) 可见，当 σ 较大时，相同 σ 下全基因变异的成功概率小于单基因变异的成功概率（当 σ 很小时，它们的成功概率都为 0.5）。而随着进化代数的增大（所得到的最优解与理论最优解之间的距离也不断减小），需要不断减小 σ 才能得到较大的成功变异的概率。

进化窗

虽然 σ 值越小，成功概率越大，接近 0.5，但进化的效果并不是 σ 值越小越好，图 4-8 显示了在不同 σ 时适应值累计改进幅度，由图可见，在进化代数不同时（因而适应值也不同，与极值点的距离也不同）最佳 σ 值也不同，而且此最佳的 σ 值随进化过程不断减小。进一步仿真计算表明，与此最佳的 σ 相对应的成功概率为 0.30~0.48 之间，与上一节理论研究的结果一致。

图 4-9 比较了全基因变异和单基因变异的进化窗，其中图（a）是 100 维球模型函数在进化到 1500 代（此时，函数适应值为 45.45574）时的横向仿真结果，用横向仿真法作 10000 次变异试验，得到适应值函数累计改进幅度，归一化处理后得到的 $\varphi-\sigma$ 曲线。(b) 是函数 f_6（100 维）在进化 500 代（此时，函数适应值为 2.47639）时的横向仿真结果，试验方法同（a），由于全基因变异时，最

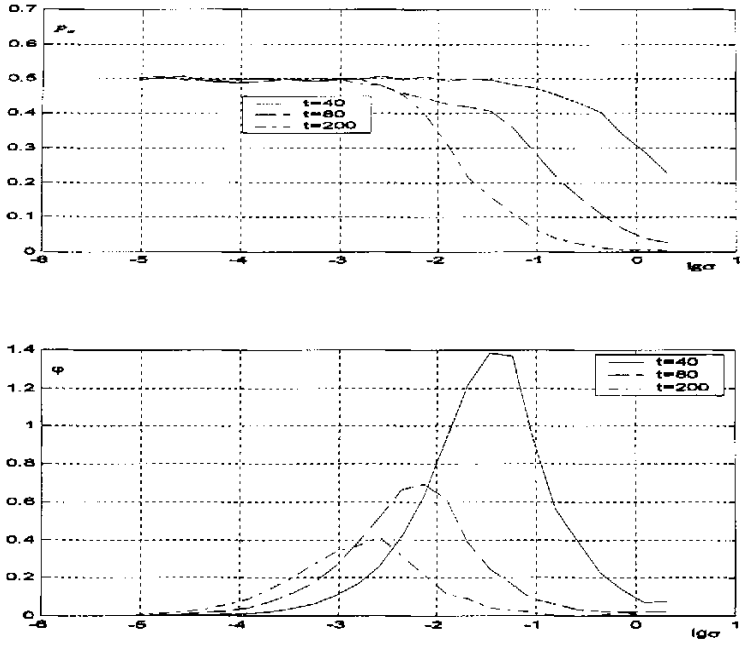
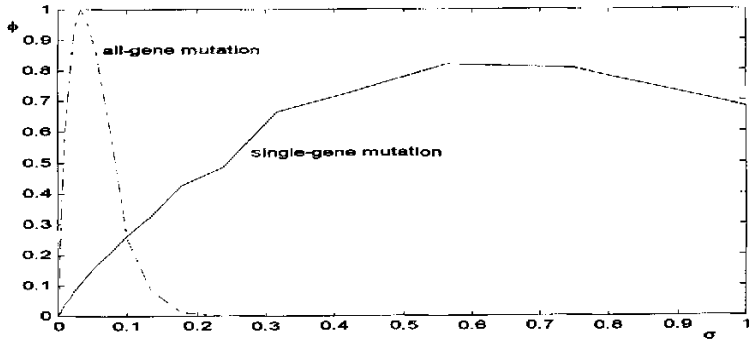


图 4-8　变异成功概率、累计改进幅度与步长的关系

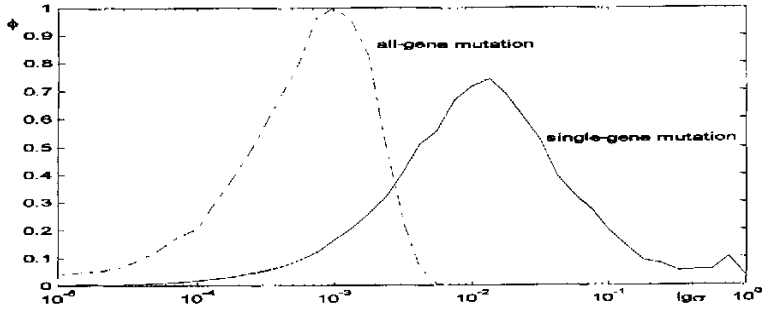
大累计改进幅度出现在σ很小时，在图（b）中，横坐标使用的是对数坐标。从图可见，全基因变异的累计改进幅度大于单基因变异，有利于提高收敛速度，但单基因变异在σ很大变化范围内都具有较大的改进幅度，使σ的适应范围很广，增强了算法的鲁棒性，另外，由于单基因变异时，较大的σ也可以获得良好的改进幅度，因此在进化过程中，σ能够维持在较大的值，有利于跳出局部极值点，增强全局搜索能力。

为了更进一步比较单基因变异与全基因变异的变异成功概率和累计改进幅度，表 4-1 列出了使用自适应型变异步长，在进化代数为 40、80、200 代时，单基因变异、全基因变异各 10000 次，得到的统计结果。从表 4-1 可知，由于使用自适应型变异步长时，σ具有较强的随机性，其进化的概率也具有一定的随机性，但单基因变异成功的概率显著优于全基因变异，累计进化幅度也大于全基因变异。

基于以上理论分析和横向仿真结果，使用 Gauss 分布单基因变异时，变异步长作为种群级别的参数，使用逐步递减的适应型步长控制策略，即一个种群使用同一个变异步长，当进化过程中变异成功概率小于 0.3 时，则σ减半，否则，维持不变，具有较好的局部搜索能力，以下将这种步长控制策略称为递减型步长控制策略。



(a) 球模型函数



(b) 函数 f_6

图 4-9 全基因变异和单基因变异的 $\phi-\sigma$ 比较

表 4-1　CES 变异方式在不同阶段的进化概率和进化效果

进化代数	当前最优适应值	进化概率		累计进化幅度	
		单基因变异	全基因变异	单基因变异	全基因变异
40	-30.71543	0.289	0.0702	0.28455	0.020502
80	-0.09868	0.187	0.0552	0.14190	0.020274
200	-0.00141	0.362	0.0533	0.01718	0.005230

4.3 全局收敛性分析

4.3.1 概率意义下的收敛性

定义 4. 2 [2][133] **收敛性** 设 Z_t 为 t 代时种群中所有个体适应值的最大值，f^* 为适应值函数 $f(\mathbf{x})$ 在可行解区间 S 的最大值，若 Z_t 满足

$$\lim_{t\to\infty} P\{Z_t = f^*\} = 1 \tag{4-6}$$

则称算法收敛到最优解 f^*。

当种群中的最优个体处在某个极值点的吸引域且没有收敛到该极值点时，由 3.3 节和 4.2 节的分析可知，只要步长合适，算法以大于 0 的成功概率不断向极值点逼近。问题是，这个极值点不一定是全局最优解，算法能否以概率 1 进入全局极值点所在吸引域成为保证全局收敛的关键。在众多的有关进化算法收敛性研究[73~77][133]的结论中，都可以归结为“当算法使用精英保留策略，并具有全局遍历性时，算法以概率 1 收敛到全局极值点”，但在实际应用这一结论时，算法的全局遍历性能否始终得到保证？Rudolph[78]提出了一个这样的反例，并证明自适应变异可能导致早熟收敛[79]。使用递减型步长控制策略的单基因变异进化策略 SM—ES（single-gene-mutation-based ES）能否保证全局收敛性？本节将以两个反例说明。先给出能否跳出局部极值点的判据。

考察图 4-10，S 是可行解区间，$P \subset S$ 为局部极值点区间，对于 $\forall \mathbf{x} \in P$，$f(\mathbf{x}) = \varepsilon > 0$；$Q \subset S$ 为全局极值点区间，对于 $\forall \mathbf{x} \in Q$，$f(\mathbf{x}) < \varepsilon$；对于 $\forall \mathbf{x} \in S$ 且 $\mathbf{x} \notin P \cup Q$，$f(\mathbf{x}) > \varepsilon$。

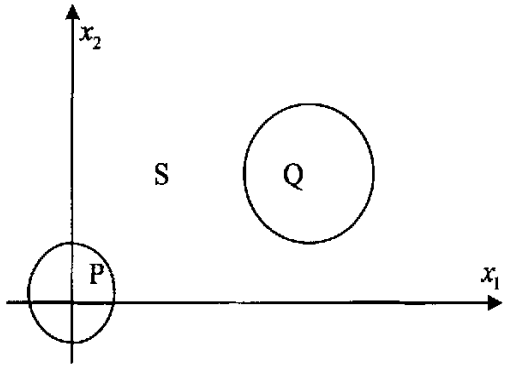


图 4-10 能否跳出局部极值点的判据

当算法具有局部收敛性，即能以概率 1 在极值点所在吸引域收敛到极值点时，是否一定能够以概率 1 保证全局收敛，可以等价地表示为是否一定能成功跳出局部极值点区域 P。

一定能成功跳出局部极值点的判据

以 $(1+1)-ES$ 分析，此时执行一代进化计算就是一次变异，如果第 $k\geq 1$ 代变异跳出局部极值点的概率 p_k 满足

$$p_k = P\{P \to Q \ at \ step \ k\} \geq q_k \tag{4-7}$$

则经过 t 代能成功跳出局部极值点的下限概率为

$$1-\prod_{k=1}^{t}(1-p_k) \geq 1-\prod_{k=1}^{t}(1-q_k)$$

能以概率 1 成功跳出局部极值点的充分条件为

$$\lim_{t\to\infty}\prod_{k=1}^{t}(1-q_k)=0 \quad \Leftrightarrow \quad \lim_{t\to\infty}\sum_{k=1}^{t}\log(\frac{1}{1-q_k})=\infty \tag{4-8}$$

不一定能成功跳出局部极值点的判据

如果 p_k 满足：

$$p_k = P\{P \to Q \ at \ step \ k\} \leq q'_k$$

则经过 t 代进化，能成功跳出局部极值点的上限概率为

$$1-\prod_{k=1}^{t}(1-p_k) \leq 1-\prod_{k=1}^{t}(1-q'_k)$$

不能以概率 1 成功跳出局部极值点的充分条件为

$$\lim_{t\to\infty}\prod_{k=1}^{t}(1-q'_k)>0 \quad \Leftrightarrow \quad \lim_{t\to\infty}\sum_{k=1}^{t}\log(\frac{1}{1-q'_k})<\infty \tag{4-9}$$

此时，算法不能以概率 1 全局收敛。为了后面分析方便，先给出两个引理。

引理 4.1[78]　如果 $x\in(0,1)$，则

$$x<\log(\frac{1}{1-x})<\frac{1}{1-x}。$$

引理 4.2 [78]　如果序列 $\lim_{k\to\infty}|a_k|^{1/k}=\alpha<1$，则级数 $\sum_{k=1}^{\infty}a_k$ 收敛于有限值。

4.3.2 反例 1　变异步长减小导致早熟收敛

此例是仿照文献[78]构造的,由于 SM-ES 使用了单基因变异，所以，以一个求一维函数最小值的优化问题说明比较直接。设适应值函数

$$f(x)=\begin{cases}\frac{3}{4}\varepsilon x^2/\delta^2+\varepsilon & if\ \ x\le 2\delta\\ 4\varepsilon & if\ \ 2\delta<x<l-2\delta\\ \varepsilon(x-l)^2/\delta^2 & if\ \ x\ge l-2\delta\end{cases}$$

其中 $\varepsilon>0,\delta>0,l\ge 6\delta>0$，其函数图形如图 4-11 所示。

使用 (1+1) SM－ES，并略作简化，当经过一代进化而适应值没有改善时，则减小变异步长，即

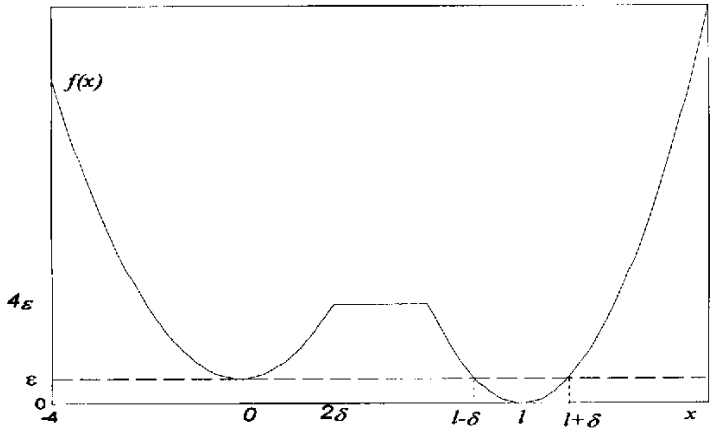


图 4-11 反例 1 的适应值函数图形

$$x_{t+1}=\begin{cases}x_t+\sigma_t z_t & if\ \ f(x_t+\sigma_t z_t)<f(x_t)\\ x_t & otherwise\end{cases}\tag{4-10}$$

$$\sigma_{t+1}=\begin{cases}\sigma_t & if\ \ f(x_t+\sigma_t z_t)<f(x_t)\\ \gamma\sigma_t & otherwise\end{cases}\tag{4-11}$$

$\gamma\in(0,1)$ 为步长缩减因子。根据收敛性定义 4. 2，如果算法全局收敛，则不论在何种初始状态，$t\to\infty$ 时，都应该以概率 1 收敛于 $x^*=l$。设初始状态在极值点 $x_0=0$ 所在的吸引域，并已经收敛到该极值点，即式(4-10)、(4-11)中 $x_0=0,\sigma_0>0$ 为某个正值，此时，只有当随机变量 $\sigma_t z_t$ 的值落在 $(l-\delta,l+\delta)$ 范围内，才能跳出局部极值点 $x_0=0$，进入全局极值点 x^* 所在吸引域，所以，以概率 1 全局收敛的必要条件是：$t\to\infty$ 时，$P\{\sigma_t z_t\in(l-\delta,l+\delta)\}=1$。

当使用 Gauss 变异时，$\sigma_t z_t$ 为 $N(0,\sigma_t)$ 分布的随机变量，其累积概率分布 $F_Z(z)$，概率分布密度函数为

$$f_Z(z)=\frac{1}{\sigma_t\sqrt{2\pi}}\exp(-\frac{z^2}{2\sigma_t^2}) \tag{4-12}$$

一次变异跳出局部极值点的概率

$$\begin{aligned} p_k &= P\{0\to(l-\delta,l+\delta)\} \\ &= P\{l-\delta<z<l+\delta\} \\ &= F_Z(l+\delta)-F_Z(l-\delta) \\ &= 2\delta f_Z(l-\delta+\theta\cdot 2\delta) \quad \text{（根据中值定理，} 0<\theta<1\text{）} \end{aligned}$$

由于$l>\delta>0$，有

$$2\delta f_Z(l+\delta)\le p_k\le 2\delta f_Z(l-\delta) \tag{4-13}$$

对于单基因 Gauss 变异，由概率分布函数式(4-12)和式(4-13)得

$$\begin{aligned} p_k &\le q'_k = 2\delta f_{Z_k}(l-\delta) \\ &= \delta\sqrt{\frac{2}{\pi}}\frac{1}{\sigma_k}\exp(-\frac{(l-\delta)^2}{2\sigma_k^2}) \\ &= A\eta_k\exp(-B\eta_k^2) \end{aligned} \tag{4-14}$$

其中$A=\delta\sqrt{\frac{2}{\pi}}$，$B=\frac{(l-\delta)^2}{2}$，$\eta_k=\frac{1}{\sigma_k}$。由引理 4.1 有

$$\sum_{k=1}^{\infty}\log(\frac{1}{1-q'_k})<\sum_{k=1}^{\infty}\frac{1}{1-q'_k}\le\frac{1}{1-q'_1}\sum_{k=1}^{\infty}q'_k \tag{4-15}$$

由式(4-11)及$\eta_k=\frac{1}{\sigma_k}$，有$\eta_k=\frac{1}{\sigma_k}=\frac{1}{\gamma^k\sigma_0}=\eta_0\beta^k$，其中$\beta=1/\gamma>1$。

由式(4-14)和式(4-15)可以得到

$$\begin{aligned} \frac{1}{1-q'_1}\sum_{k=1}^{\infty}q'_k &= \frac{A}{1-q'_1}\sum_{k=1}^{\infty}\frac{\eta_k}{\exp(-B\eta_k^2)} \\ &= \frac{A\eta_0}{1-q'_1}\sum_{k=1}^{\infty}\frac{\beta^k}{\exp(-B\eta_0^2\beta^{2k})} \end{aligned}$$

$$=\frac{A\eta_0}{1-q'_1}\sum_{k=1}^{\infty}a_k \tag{4-16}$$

由于$\lim_{k\to\infty}|a_k|^{1/k}=\lim_{k\to\infty}\frac{\beta}{\exp(B\eta_0^2\beta^{2k}/k)}=0$，故式(4-16)收敛，考虑式(4-15)即有

$\sum_{k=1}^{\infty}\log(\frac{1}{1-q'_k})<\infty$。

由式(4-9)，可以得出结论，递减变异步长 SM−ES 不能保证跳出局部极值点，因而，有可能出现早熟收敛，或不能以概率 1 全局收敛。

结论 4.1　递减步长 Gauss 变异将可能出现早熟收敛。

类似方法，也可以得出递减步长 Cauchy 变异也可能出现早熟收敛。直观的说，出现这种早熟收敛的原因就是当算法收敛于一个极值点后，由于成功变异的概率不断减小，变异步长也不断减小，导致跳出局部极值点的概率不断减小，最终接近于 0。

4.3.3 反例 2　单基因变异导致早熟收敛

考虑二维函数：

$$f(x_1,x_2)=\frac{1}{x_1^2+x_2^2+0.1}+\frac{1}{(x_1-1)^2+(x_2-1)^2+0.05}$$

其等高线如图 4-12 所示，该函数在$\mathbf{x}_1^*=(0.99943219,0.99943219)$附近有全局最

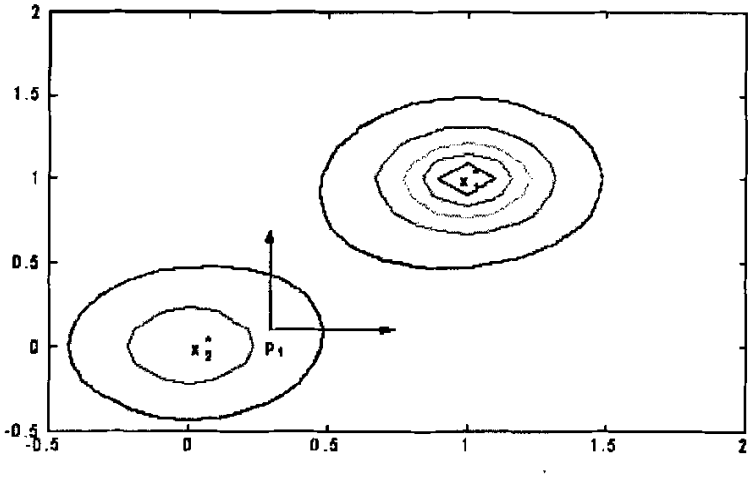


图 4-12　反例 2 适应值函数等高线

大值 20.47644798，在 $\mathbf{x}_2^*=(0.0023967,0.0023967)$ 有局部极大值 10.48894535。当使用单基因变异时，一旦当前最优解进入 $\mathbf{x}_2^*$ 所在吸引域，由于一次只有一个变量发生变异，很难跳到另一个极值点的吸引域，即使有足够大的变异步长。

在不同的种群参数下，对该函数做 100 次重复计算，统计收敛到全局最优解的次数，如表 4-2 所示。

可见，当种群规模较小时，单基因变异方式找到全局最优解的次数明显少于全基因变异方式，说明单基因变异比较容易导致早熟收敛。另外，从表也可以看出，当种群规模比较小时，递减型变异步长 ES 的收敛效果优于 $SA-\sigma$ 型 ES。

表 4-2 中可见，使用单基因变异时，需要比较大的种群规模，以增加进入全局最优解吸引域的概率，才能获得较好的全局收敛性。

表 4-2　不同变异方式和种群参数时的函数收敛次数

算　法	全基因变异	单基因变异
(30+200) $SA-\sigma$	100	100
(30+200) 递减型变异步长	100	100
(2+8)　$SA-\sigma$	66	41
(2+8)　递减型变异步长	100	49

4.4 均匀变异算子的引入

如果直观地解释 4.3.2 小节反例 1 早熟收敛的原因，就是因为当陷入局部极值点后，由于跳出极值点的概率比较小，而跳出不成功导致变异步长进一步减小，使跳出概率趋于 0。如果不使用变异步长递减策略，又将导致进化停顿，为此，可以在变异算子中，既使用含变异步长递减的变异算子产生部分后代，同时使用不改变变异步长的变异算子产生部分后代，两者结合，即可同时兼顾局部搜索能力和全局搜索能力。不改变变异步长的变异最容易实现的就是使用均匀变异，为此，在 SM－ES 中，除了单基因 Gauss 变异，还引入均匀变异，把这种算法称为 $(\mu+\lambda+\kappa)-ES$，其中 λ 是 Gauss 变异产生后代的数量，κ 是均匀变异产生后代的数量。为了说明 $(\mu+\lambda+\kappa)-ES$ 以概率 1 跳出局部极值（或者说以概率 1 跳到全局收敛域），以 $(1+1+1)-ES$ 证明。

设 $A=[a_1,a_2]^n\subset Q$，变异由 Gauss 变异和均匀变异两部分组成，在 k 代时，一代变异从 P 跳到 Q 的概率，不妨设 $\mathbf{x}^{(k)}=\mathbf{0}\in P$

$$
\begin{aligned}
p_k &= P\{\tilde{\mathbf{x}}^{(k+1)} \in Q \mid \mathbf{x}^{(k)} \in P\} \\
&= \int_Q f_{Z_G^{(k)}}(Z)dZ + \int_Q f_{Z_U}(Z)dZ - \int_Q f_{Z_G^{(k)}}(Z)dZ \cdot \int_Q f_{Z_U}(Z)dZ \\
&\geq \int_Q f_{Z_U}(Z)dZ \\
&\geq \int_A f_{Z_U}(Z)dZ \qquad (\because A \subset Q, f_{Z_U}(Z) > 0) \\
&= \prod_{i=1}^{n} \frac{a_2 - a_1}{s_{i2} - s_{i1}} > r^n = q_k
\end{aligned}
$$

其中，$r = \min\{\frac{a_2 - a_1}{s_{i2} - s_{i1}}, i = 1,2,...,n\}$。

而 t 代能到达 Q 的概率：

$$1 - \prod_k^t (1 - p_k) \geq 1 - \prod_{k=1}^t (1 - q_k) = 1 - (1 - r^n)^t$$

当 $t \to \infty$ 时，$(1-r^n)^t \to 0$，所以，由式(4-8)得到 $(1+1+1)-ES$ 以概率 1 到达全局极值点区间 Q。

4.5 （$\mu + \lambda + \kappa$）-ES

4.5.1 算法描述

根据以上分析，在单基因变异的基础上，使用递减型策略参数，并引入均匀变异算子，构造了 $(\mu+\lambda+\kappa)-ES$，如 Algorithm 4.1 所示。

4.5.2 仿真计算

测试函数

为了全面地分析单基因变异与全基因变异、自适应型与递减型变异步长控制策略、引入均匀变异的效果，作者选择了一组 10 个 100 维的测试函数：

1. $f_1 = \sum_{i=1}^{n} \sin(x_i) \sin^{20}(\frac{i \times x_i^2}{\pi})$

变量范围[0,3.14]，据[132]介绍，当 $n=100$ 时最大值为 99.2784,作者的计算结果为 99.61630365；此函数随着 n 增大，复杂性显著增加，峰的数量大幅增大。

Algorithm 4.1：　$(\mu+\lambda+\kappa)-ES$

Procedure $(\mu+\lambda+\kappa)-ES$

Begin

产生$(\mu+\lambda+\kappa)$个个体组成的初始种群$\mathbf{x}_i, i=1,...,\mu+\lambda+\kappa$，$t=1$；

计算所有个体的适应值$f(\mathbf{x}_i), i=1,...,(\mu+\lambda+\kappa)$，保留其中优秀的$\mu$个优秀个体；

while（终止条件不满足）

Begin

由当前最优父代个体$\mathbf{x}_0=\max\{\mathbf{x}_i, i=1,...,\mu+\lambda+\kappa\}$，产生$\lambda$个单基因 Gauss 变异子个体：

$$x'_k(j)=x_0(j)+\sigma_t N(0,1),\quad j=random(1,n)$$

$$x_k'(i)=x_0(i), k=1,2,...,\lambda,\quad i=1,2,...,n, i\neq j$$

再产生κ个均匀变异子个体：

$$x_k'(j)=x_0(j)+\zeta_j,\quad j=random(1,n)$$

$$x_k'(i)=x_0(i), k=1,2,...,\kappa,\quad i=1,2,...,n, i\neq j$$

计算$(\lambda+\kappa)$个新个体的适应值，从$(\mu+\lambda+\kappa)$个个体中选择μ个个体作为下一代的父个体；

如果一个进化时段（如 30 代）的个体进化统计概率小于 0.3，则$\sigma_{t+1}=0.5\sigma_t$，否则$\sigma_{t+1}=\sigma_t$；

$t=t+1$；

End

End

2. $$f_2=\frac{-1}{10}\{(\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+$$

$$(x_n-1)^2[1+\sin^2(3\pi x_n)\}-\sum_{i=1}^{n}u(x_i,5,100,4)$$

其中：

$$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\ 0, \quad -a\le x_i\le a\\ k(-x_i-a)^m, x_i<a\end{cases}$$，变量范围为[-50,50]，理论最大值为 0。

3. $f_3 = -\sum_{i=1}^{n} x_i^2$

球模型函数，变量范围为[-20,30]，其二维函数图形如图 2-2（a）所示，理论最大值为 0。

4. $f_4 = 20\exp(-0.2\sqrt{0.1\sum_{i=1}^{n} x_i^2}) + \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) - 20 - e$

Ackley 函数，变量范围为[-20,30]，其二维函数图形如图 2-2（b）所示，理论最大值为 0。

5. $f_5 = -\sum_{i=1}^{n-1}[100(x_i - x_{i+1})^2 + (x_i - 1)^2]$

此函数称为 Rosenbrock 函数，变量范围为[-5.12,5.12]，其二维函数图形如图 2-2（c）所示，理论最大值为 0。

6. $f_6 = -\sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$

变量范围为[-5.12,5.12]，理论最大值为 0。

7. $f_7 = \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$

变量范围为[-512，512]，理论最大值为 41898.288。

8. $f_8 = \frac{-1}{4000}\sum_{i=1}^{n} x_i^2 + \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) - 1$

变量范围为[-600,600]，理论最大值为 0。

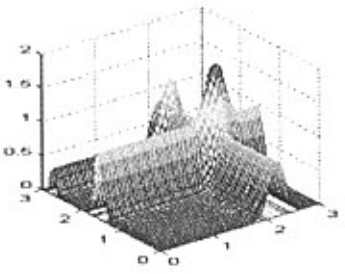
9. $f_9 = \frac{-1}{N}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$

变量范围为[-5,5]，理论最大值为 78.3326。

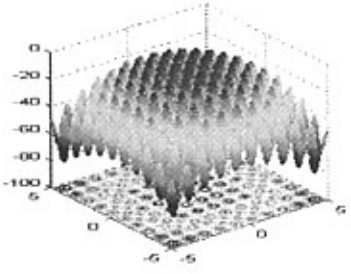
10. $f_{10} = -(\sum_{i=1}^{n}|x_i| + \prod_{i=1}^{n}|x_i|)$

变量范围为[-10,10]，理论最大值为 0。

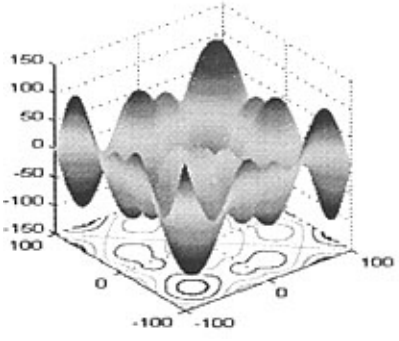
部分函数当维数为 2 时的图形如图 4-13 所示。



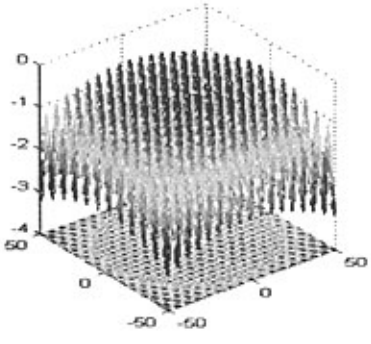
（a）2 维 f_1 函数



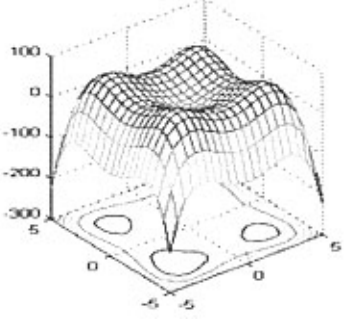
（b）2 维 f_6 函数



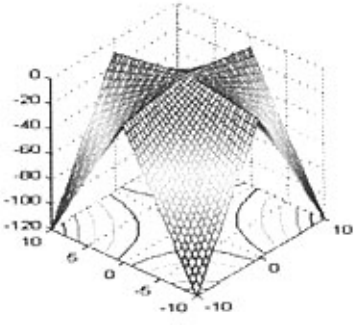
（c）2 维 f_7 函数



（d）2 维 f_8 函数



（e）2 维 f_9 函数



（f）2 维 f_{10} 函数

图 4-13 部分测试函数维数为 2 维时的 3D 图形

仿真结果

为了全面地分析比较不同步长控制策略、不同变异方式时 ES 的计算性能，作者从三方面做了仿真计算：1） $SA-\sigma$ 全基因变异 $(\mu+\lambda)-ES$（即 CES）与单基因变异 $(\mu+\lambda)-ES$ 比较，种群参数使用（30+200）[18]，停止条件为：最大计算代数 10000 代或连续 1000 代没有进化或收敛到给定适应值；2）递减型变异步长（2+8）全基因与单基因变异，最大进化代数为 100000 代；3）递减型变异步长（2+8+2）全基因与单基因变异，最大进化代数为 100000 代。所有全基因变异重复计算 10 次，单基因变异重复计算 50 次。计算结果分别如表 4-3~4-5 所示。

表 4-3 平均适应值统计

函数号	（30+200）$SA-\sigma$		（2+8）递减型变异步长		（2+8+2）递减型变异步长	
	全基因	单基因	全基因	单基因	全基因	单基因
1	26.02777	99.00497	45.43576	97.26129	50.46779	98.71543
2	-0.50202	-0.00009	-41.93301	-0.00009	-51.63603	-0.00009
3	-0.00007	-0.00009	-0.00009	-0.00009	-0.00009	-0.00009
4	-14.46476	-0.00193	-18.67009	-0.00009	-18.52045	-0.00009
5	-99.71810	-88.17848	-98.53956	-1.86377	-25.81481	-0.00014
6	-349.8261	-0.002639	-559.66148	-1.82542	-630.20336	-0.00010
7	26526.81	25975.17	24465.868	23738.69	26353.989	41898.28
8	-0.09928	-0.30464	-0.00494	-0.28283	-0.00494	-0.20961
9	65.0008	74.92820	66.03338	72.91513	67.19259	78.33226
10	-35.76674	-0.03920	-72.20080	-12.2861	-454.68850	-0.00001

表 4-3 统计了不同变异步长策略和变异方式下，平均适应值结果统计，从表可见，在步长控制策略相同时，单基因变异 ES 的适应值结果优于全基因变异 ES。从收敛次数统计，如表 4-4 所示，单基因变异收敛性能明显优于全基因变异，引入均匀变异后的（2+8+2）单基因变异 ES，其解的质量、收敛性能得到了改善。表 4-5 为计算时间统计，单基因变异的计算时间明显少于全基因变异，这是由于全基因变异算子本身的计算开销大、而且收敛速度慢所致。从种群规模比较，（2+8+2）ES 比（2+8）ES 稍大，另外由于前者收敛精度较高，所需计算代数较多，计算时间开销略大。

以上计算结果中，计算 Rosenbrock 函数 f_5 时，递减型变异步长 ES 的种群规模分别为（2+80）、（2+80+2）。对于容易陷入局部极值点的函数，种群规

表 4-4　收敛次数统计

函数号	(30+200)自适应型变异步长		(2+8)递减型变异步长		(2+8+2)递减型变异步长	
	全基因	单基因	全基因	单基因	全基因	单基因
1	0	50	0	50	0	50
2	10	50	2	50	3	50
3	50	50	50	50	50	50
4	0	50	0	50	0	50
5	0	0	0	48	0	50
6	0	47	0	2	0	50
7	0	0	0	0	0	50
8	50	50	50	50	50	50
9	0	0	0	0	0	50
10	0	50	0	38	0	50

表 4-5　计算时间统计

函数号	(30+200)自适应型变异步长		(2+8)递减型变异步长		(2+8+2)递减型变异步长	
	全基因	单基因	全基因	单基因	全基因	单基因
1	183.100	76.280	47.100	7.400	76.300	13.980
2	485.75	44.240	73.500	2.6400	96.900	7.000
3	255.900	17.960	2.600	0.360	2.100	0.360
4	257.800	72.400	3.100	1.740	3.500	1.520
5	2488.00	86.300	566.000	64.600	538.250	42.500
6	305.300	73.620	5.100	1.940	4.500	2.300
7	404.500	62.700	5.400	3.460	6.500	3.180
8	394.200	177.400	4.100	5.300	4.600	2.220
9	224.660	33.240	2.900	1.820	3.600	1.380
10	531.800	65.940	60.125	6.9200	4.095	1.620

模大点，有利于获得较强的全局收敛性，表 4-2 也说明了这一结果。

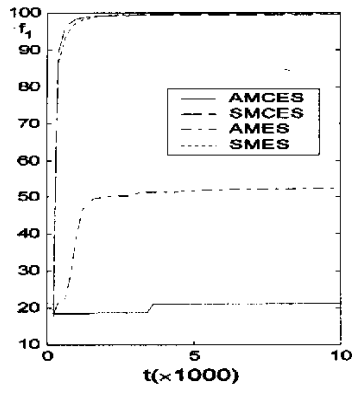
在上述这些函数中，球模型函数 f_3 和函数 f_8 在以上几种变异方式和策略参数、步长控制策略下都得到了比较好的适应值结果，甚至全基因变异的精度还略高，只是计算时间开销稍大，而球模型函数是 ES 研究中常常使用的测试函数，我想，这也是导致单基因变异一直以来无人重视的原因。

为了比较不同变异方式和不同步长控制策略下，算法的动态性能，对以上函数使用相同种群规模（2+80）-CES、（2+78+2）-ES 做进化计算，得到适应值动态变化曲线，如图 4-14 所示。图中，AMCES 表示使用全基因变异、自适应型变异步长；SMCES 表示单基因变异、自适应型变异步长；AMES 表示全基因变异、递减型变异步长；SMES 表示单基因变异、递减型变异步长。从图可见，AMCES 的性能都比较差，单基因变异总的性能都比较好，而 SMES 的总体性能比较 SMCES 好，即使用递减型变异步长比自适应变异步长好。有些函数，如 f_4、f_5、f_8、f_{10}，AMES 性能比 SMES 稍好，但对于其他函数，如 f_1、f_6、f_7、f_9，则性能较差，甚至不能收敛，说明单基因变异的鲁棒性好，适用范围更广。

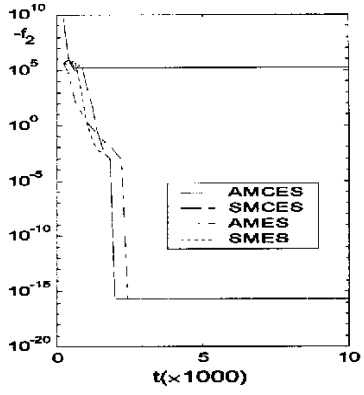
表 4-6 列出了相同种群规模和计算代数时不同算法的计算时间开销比较、相同精度下适应值函数计算次数比较。由于 AMCES 需要每个基因都按式(3-6)计算变异步长、按式(3-7)做变异操作，所以，其时间开销最大；AMES 的所有基因使用相同的变异步长，只需要对每个基因做变异操作，计算时间开销次之；SMCES 每次繁殖，计算一次变异步长、做一次变异操作，因而，时间开销比 SMES 大；SMES 每一代繁殖才计算一次步长，每次繁殖变异一个基因，故计算时间开销最少。SMES 的时间开销不到 AMES 的 50%，当适应值函数不太复杂且维数越高时，如球模型函数 f_3，时间开销差距越大。在相同精度下，SMES 的适应值函数调用次数明显少于 OGA/Q[132]。

表 4-6 计算开销比较

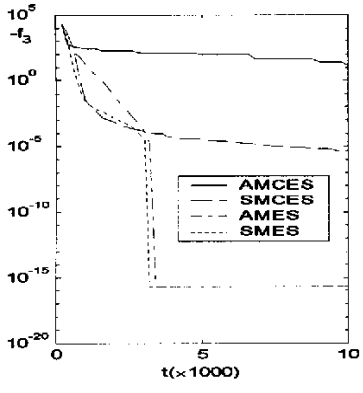
函数号	相同种群规模和计算代数时计算时间（s）比较				相同精度下适应值计算次数比较	
	AMCES	SMCES	AMES	SMES	SMES	OGA/Q[132]
1	221.0	107.0	142.0	105.0	19657	302773
2	183.0	49.0	91.0	46.0	21089	134143
3	144.0	10.0	54.0	8.0	54294	112559
4	158.0	26.0	70.0	24.0	93565	112421
5	143.0	8.0	52.0	6.0	88145	167863
6	158.0	27.0	71.0	25.0	9635	302166
7	177.0	39.0	84.0	37.0	20313	134556
8	170.0	32.0	80.0	31.0	29044	245930
9	177.0	45.0	86.0	44.0	44361	112612
10	154.0	25.0	66.0	21.0	94608	112576



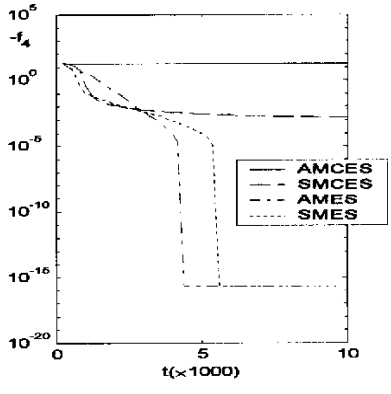
(a) f_1



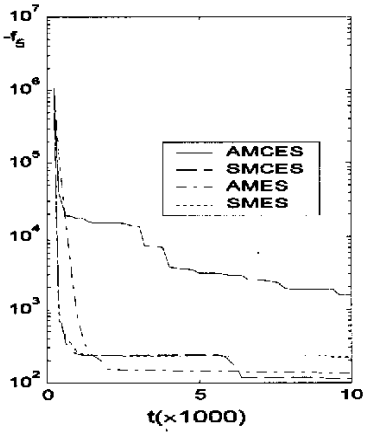
(b) f_2



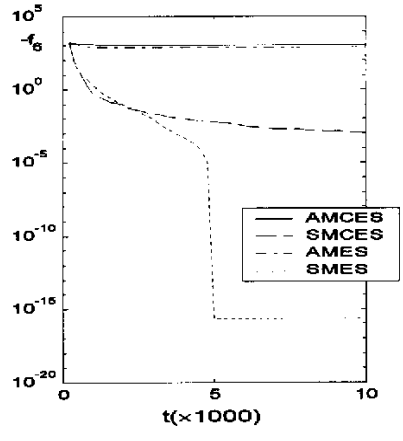
(c) f_3



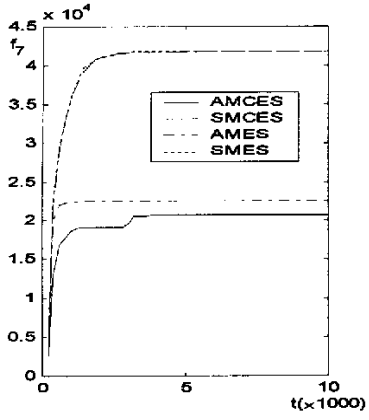
(d) f_4



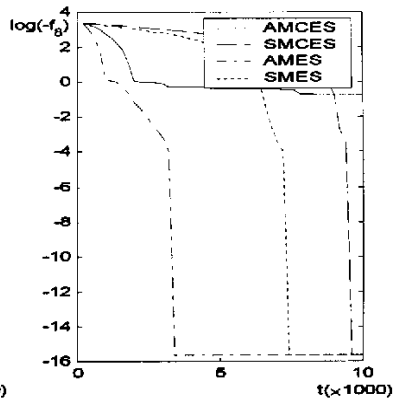
(e) f_5



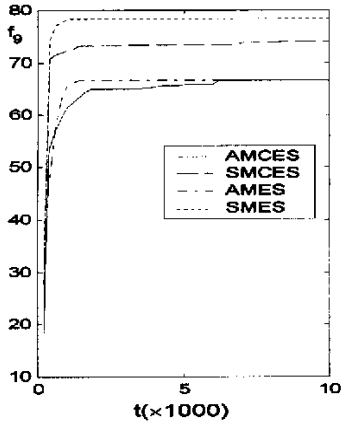
(f) f_6



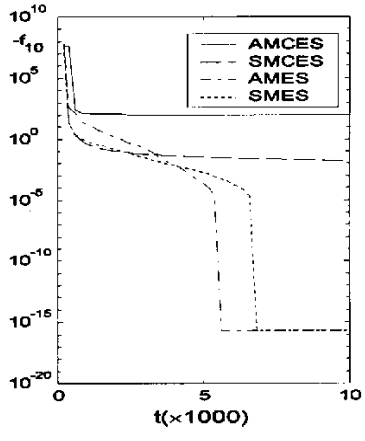
（g）f_7



（h）f_8



（i）f_9



（j）f_{10}

图 4-14　不同变异步长策略和变异方式时，适应值变化曲线

4.6 小结

本章在单基因变异 ES 的基础上，通过理论和仿真分析提出了与单基因变异相适应的递减型步长控制策略，并提出了一种横向仿真方法计算分析遗传算子的作用效果。当使用 Gauss 分布的单基因变异算子时，保持变异成功率在 0.3~0.48 范围内，具有较好的局部搜索性能，并提出了递减型的变异步长控制策略。通过两个反例说明了这种基于步长递减型单基因 Gauss 变异的 ES 全局搜索能力的

不足，可能导致早熟收敛，为此，在 Gauss 变异的基础上，引入了均匀变异，构造了变异步长递减单基因 Gauss 变异与均匀变异相结合的 $(\mu+\lambda+\kappa)-ES$，通过一组典型测试函数说明了单基因变异比全基因变异具有更好的局部搜索能力，递减型变异步长比随机型变异步长具有更好的局部搜索性能，而均匀变异算子的引入提高了全局搜索能力。

由于遗传漂移的作用，种群中的个体即使在初期具有良好的多样性，但很快就会集中到某一个优秀个体，以后产生的后代主要是该个体的，使种群失去多样性，此时，尽管算法可以在 $t\to\infty$ 时以概率 1 跳出局部极值点，但每一代跳出的概率取决于水平集 $L(\mathbf{x}^*(t))=\{\mathbf{x}\mid f(\mathbf{x})\geq f(\mathbf{x}^*(t)\}$ 的尺寸（其中 $\mathbf{x}^*(t)$ 为 t 代时的最优个体），而随着进化代数的增加，$\mathbf{x}^*(t)$ 不断增大，$L(\mathbf{x}^*(t))$ 的尺寸不断减小，使跳出局部极值点的概率越来越小，导致全局搜索能力显著不足。为了增强全局搜索能力，也为了求解多模态函数多解问题，作者在 $(\mu+\lambda+\kappa)-ES$ 基础上，引入了多种群技术，在本文下一章将论述基于 $(\mu+\lambda+\kappa)-ES$ 的多种群技术。

第五章　基于($\mu+\lambda+\kappa$)-ES的多种群技术

5.1 引言

基于以下原因，在$(\mu+\lambda+\kappa)-ES$基础上引入多种群技术：

1、遗传漂移　在单一种群的进化算法中，由于重组和选择机制将导致遗传漂移[53]，即使在进化初期，种群中的个体满足多样性要求，但随着遗传操作和选择竞争，种群中的个体将逐步由单一的某个优秀个体及其后代所占据，这时，种群丧失多样性。多种群技术(multi-population)、小生境技术(niching)是一种形成和维持多个子种群、抑制遗传漂移、保持种群多样性的有效手段。

2、全局搜索能力　当种群中的个体是单一类型的个体时，如果这种类型的个体在某个局部极值点的吸引域内，这时，尽管算法具有全局搜索能力，但新产生的在其他吸引域的个体适应值必须大于当前最优个体$\mathbf{x}^*(t)$的适应值才可能得以生存、繁殖、进化，也就是说，新生个体必须在水平集$L(\mathbf{x}^*(t))=\{\mathbf{x}\,|\,f(\mathbf{x})>f(\mathbf{x}^*(t))\}$内才能具有足够的竞争力得以生存，而随着进化的进行，$\mathbf{x}^*(t)$不断增大，$L(\mathbf{x}^*(t))$的尺寸不断减小，在$L(\mathbf{x}^*(t))$内产生新生个体的机会越来越小，导致全局搜索能力严重不足，出现早熟收敛。使用多种群技术，把生存竞争限制在各自的子种群，在子种群的个体即使在整个种群中没有竞争能力，但只要它所在子种群没有充分进化，还是能在子群中得以生存、繁殖、进化，这时，新种群中个体的生存与否取决于是否在该个体所在极值点的吸引域是否已有子种群存在。

3、多解问题　在多模态函数优化问题中，适应值函数存在多个极值点，常常需要求出所有的这些极值点。在单一种群的算法中，即使种群规模很大，由于遗传漂移，很难同时维持多个不同类的个体，因而难以求出多个极值点，多种群技术由于可以维持多个不同类的子种群，是同时求出多个极值点的有效方法。

另外，在多处理机、分布式并行计算中，多种群更易于实现并行计算。

本章在综述多种群进化算法的基础上，提出基于$(\mu+\lambda+\kappa)-ES$的多种群算法$m\times(\mu+\lambda+\kappa)-ES$[134]，论述其实现思想，确定子种群消亡与再生的判据，提出一种避免确定峰半径（或小生境半径，下同）的山谷探索法，峰半径的确定问题是当前多种群技术中普遍存在的难点问题，最后，将给出仿真计算结果。为便于叙述，先定义几个有关的术语。

定义 5.1 类 类是具有某种共同属性或特征的元素集合。

定义 5.2 极值点 在高等数学中，已有极值点的定义，即如果$\mathbf{x}^* \in S$，存在一个$\mathbf{x}^*$的邻域，对于所有在这个邻域内的$\mathbf{x}$，$f(\mathbf{x}) < f(\mathbf{x}^*)$成立，则$\mathbf{x}^*$是一个极大值。在此，对极值点的概念稍作推广，考虑山脊型函数，如果$\mathbf{x}^*$是极值点，在包含$\mathbf{x}^*$的连续子空间内的所有$\mathbf{x}'^*$满足$f(\mathbf{x}'^*) = f(\mathbf{x}^*)$，则把这个子空间作为一个广义的极值点。

定义 5.3 吸引域 对于问题空间S，吸引域A是包含于S的最大子空间，在A中有且只有一个极大值点$\mathbf{x}^*$，A称为极大值$\mathbf{x}^*$的吸引域，表示为$A(\mathbf{x}^*)$。如果$\mathbf{x}^*$是A中的一个极小值点，$A(\mathbf{x}^*)$是极小值点$\mathbf{x}^*$的吸引域。如果$\mathbf{x}_1^* \neq \mathbf{x}_2^*$是两极大值点，则$A(\mathbf{x}_1^*) \cup A(\mathbf{x}_2^*) = \varnothing$。图 5-1 为一个一维函数按极大值划分吸引域示意图。

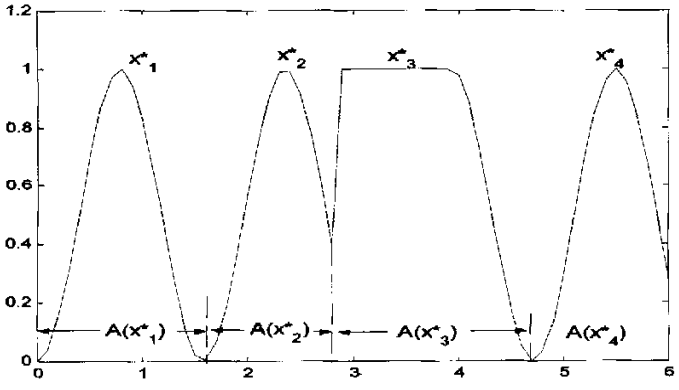


图 5-1 按极大值划分吸引域示意图

对种群中的个体按所在吸引域分类，属于同一个吸引域的个体称为同类个体（这些个体互为等价类）。

定义 5.4 多样度 对于问题空间S，可以分解为有限个互不相同的吸引域$A_1,\ldots,A_{MAX}$，其个数为MAX，对种群P中的个体按照它们所在吸引域分类，设可以分为$numc$类，则种群多样度定义为

$$diversity(P) \triangleq \frac{numc}{MAX} \tag{5-1}$$

种群多样度是多样性的直接数字度量。

按种群中个体所在吸引域分类是一种比较理想的分类方法，但由于适应值函数拓扑曲面性质的复杂性，在缺乏先验知识的情况下，考虑到算法实现的复杂性和计算开销等因素，很多算法常采样其他方法如适应值的差别、个体间的

距离作为分类依据。

定义 5.5 同峰与异峰极值点 由于计算精度、有限字长的限制，在实际进化计算过程中，所求得的极值并不一定是精确的理论值。如果$\mathbf{x}_1^*$和$\mathbf{x}_2^*$都是同一个极值点$\mathbf{x}^*$的近似值，则称$\mathbf{x}_1^*$和$\mathbf{x}_2^*$是同峰极值点；如果$\mathbf{x}_1^*$和$\mathbf{x}_2^*$是不同极值点的近似值，则$\mathbf{x}_1^*$和$\mathbf{x}_2^*$是异峰极值点；同峰与异峰极值点也分别称为同类极值点与异类极值点。

5.2 多种群技术综述

多种群技术是把一个大的种群分为若干个小的子种群，同类个体占据一个子种群，个体的生存竞争、繁殖都在各自的子种群内进行，通过不同的子种群维持大种群的总体多样性，种群之间可以通过迁徙（migration）等相互交流信息。在相关多种群技术文献中，与子种群（subpopulation）相类似的术语有小生境（niche）、同类群（deme）、子群体（subgroup）、物种（species）等，在实现的过程中，存在有一定的差别，但其基本思想相同，本文所说的子种群是一种泛指。

在多种群进化算法研究中，人们对基于遗传算法的多种群技术研究较多，基于进化策略的多种群技术文献较少，但它们都面临如何避免同类个体过度繁殖以维持种群多样性的问题，而与多样性直接相关的就是判别、区分同类个体，所以，多种群 ES 可以学习、借鉴多种群 GA 的有关方法，以下首先综述多种群 GA，再综述多种群 ES。

5.2.1 多种群 GA

在进化算法中，当种群规模一定时，如果某个类的个体过度繁殖，占据数量过大，必然会使其他类个体丧失生存机会，导致种群中某些类个体（或者基因）缺失，失去多样性。维持种群多样性的方法就是限制、避免同类个体的过度繁殖，在 GA 中，避免同类个体过度繁殖的方法主要有排挤和共享等。

排挤

早在 1970 年，Cavicchio[135]就提出了通过预选择实现排挤的思想，在他的预选择方法中，先从父代中选出与其最相似的个体，新生的子代个体只与这些相似的个体进行生存竞争。Cavicchio 的预选择建立了多峰问题遗传优化的里程碑。De Jong[136]在 Cavicchio 的基础上，提出了标准排挤算法，他以匹配的等位基因数量（Hamming 距离）作为相似性的度量，对每一个子代个体，从父代种群中随机地选择 CF（称为排挤系数）个个体，新生个体替代其中最相似的个体。

但随机选择CF个个体的过程中采样误差会导致替换错误，所以，这种算法虽然改进了种群多样性，但在多峰函数优化问题中仅能够表现有限的小生态维持能力[137]，Stdnyk[138]和 Sedbrook[139]分别作了某些局部的改进，如按适应值反比例地随机选择CF个个体，但没能有效地改进标准排挤技术。

Mahfoud[140][141]提出了以减少替换错误为主要目标的确定性排挤技术（DC: Deterministic Crowding）。DC将规模为μ的种群随机地配成$\mu/2$对，每一对父代个体通过交叉和变异生成两个子代个体，每一个子代个体和一个父代个体通过联赛竞争成为下一代种群的成员。联赛可采用两种分组方式，即父代1对子代1和父代2对子代2，或父代1对子代2和父代2对子代1。实际应用中采用一组能产生最大相似性替换的分组，相似性采用父子之间的欧几里德距离测度。DC已成功地应用于多峰问题的优化，表现了很强的小生态维持能力。郭观七等[53][142]提出了以个体间的欧氏距离测度个体间的相似性，相似个体概率替换的联赛选择小生态技术，应用期望比例分析建立小生态生长的动态方程，解析地证明了该技术能有效地维持稳定子种群，实现多物种平衡。确定性排挤DC可能导致适应值低的小生境丢失，Mengshoel[143]提出了概率排挤（PC: probability crowding）子种群技术，子代个体与相似的父代个体进行概率联赛竞争，二者以与其适应值成正比的概率竞争成为下一代种群的成员，因此引入低适应值子种群的恢复压。

共享

Goldberg[144]提出了适应值共享（fitness sharing）小生境技术，其基本思想是降低种群中相似个体的适应值，使其他个体得到较多的生存机会，从而维持种群多样性。Goldberg 提出了共享适应值概念，个体的选择以共享适应值为依据，以限制相似个体在种群中的过度繁殖。个体i的共享适应值f_i'定义为

$$f_i' \triangleq \frac{f_i}{m_i} \tag{5-2}$$

其中，f_i为个体i的原始适应值，m_i称为个体i的共享度，用于测度种群中全体个体与个体i的相似程度，m_i定义为

$$m_i \triangleq \sum_{j=1}^{\mu} sh(d_{ij}) \tag{5-3}$$

μ表示种群规模，d_{ij}表示个体i与j之间的距离，$sh(d_{ij})$称为共享函数，表示个个体i与j之间的相似性水平。最常用的共享函数定义为

$$sh(d_{ij})=\begin{cases}1-\left(\dfrac{d_{ij}}{\sigma_{share}}\right)^{\alpha}, & \text{if } d_{ij}<\sigma_{share}\\ 0 & \text{其它}\end{cases} \tag{5-4}$$

式中，σ_{share}称为小生态半径，是算法实现时由人工设置的一个表示个体之间相似性水平的阈值，α是决定共享函数形状的常数，典型值取1。

从式(5-2)、(5-3)、(5-4)可见，对于种群中某个个体i，如果存在与其距离小于σ_{share}的个体数量越多，距离越近，则共享度m_i越大，共享适应值越小，从而对它的生存起到抑制作用。Goldberg的适应值共享技术已成为多种群GA最流行的技术，称为标准共享，但共享存在以下不足之处：1)计算共享适应值需要$O(\mu^2)$的距离测度开销，还需要按式(5-2)、(5-3)、(5-4)作一系列的额外计算；2）由式(5-4)可见，σ_{share}是其中一个非常重要而且关键的参数，其确定需要适应值曲面的先验知识，这是适应值共享方法的难点；3）使用相同的σ_{share}是基于极值点的均匀分布，但实际上，极值点不一定均匀分布。

鉴于在标准共享的不足，人们提出了多种改进方法。动态适应值共享算法[145]在计算共享适应值前先将种群按照适应值降序排序，然后，根据个体之间的距离是否小于共享半径判断个体是否属于同一个小生境，应用聚类分析将种群分割成q个小生境。共享适应值的计算只在小生境内部进行，使计算成本近似为$O(q\mu)$，在一定程度上加快了算法的运行速度，提高了共享技术的效率。清除技术[146]、标签技术[147][148]、合作共享[149]、GAS[150]等都是以某种方式划分小生境，使共享在小生境内部进行。

于歆杰[151]用一种新型的保存策略和对峰值个体的大概率变异，改进了顺序生境遗传算法，并提出了一种自适应峰半径控制方法[152]，将适应值共享遗传算法中每个个体的峰半径当作决策变量，对其编码，使之参与优化的全过程，从仿真计算结果看，仍然存在极值点遗漏现象；李敏强[153]在其协同多群体GA中提出了自动确定共享半径的方法，但其仿真计算结果表明，在求解多模态函数优化问题时，仍然存在求出多余的假极值点现象，说明依靠共享半径不能准确地判别同峰还是异峰极值点，因为极值点可能分布不均匀。黎明等[154]提出了以自适应模糊Hamming神经网络通过遗传个体分类和学习，将不同的遗传群体分配在搜索空间的不同位置，以动态地调整遗传个体的搜索区域或建立新的遗传群体，以保持遗传群体的个体多样性、抑制早熟收敛、增强全局搜索能力，但其仿真计算结果表明，其全局收敛性有待增强。

动态小生境聚类（DNC：dynamic niche clustering）算法[155]在进化过程中，

应用模糊聚类分析动态地合并、分解和生成小生境，在每个小生境内部计算共享适应值。DNC 应用山谷函数[157]分析适应值曲面拓扑结构，提高了小生态分类的准确性。

Yin [156]提出了聚类分析方法确定共享半径，Lin[157]提出了一种小生境辨识技术（NIT: Niche identification technique），NIT 根据个体适应值曲面拓扑信息确定小生境半径和中心。

其他

Ursem[157]在其多国家进化算法中应用山谷函数（hill-valley function）对适应值曲面拓扑进行分析，按适应值曲面拓扑将种群分割为多个子种群，尽可能精确地使子种群能够表示多峰函数的吸引域，避开了共享技术中共享半径的确定等缺点。

mGA（messy GA）[159]通过限制非相似性个体之间的竞争和交叉来维持稳定的子种群，其基本原理是利用个体之间的距离测度衡量相似性，在竞争过程中，从种群中随机选择的个体只与相似的个体作联赛选择，由相似个体通过交叉和变异繁殖子代。文献[160]将种群中的个体按适应值大小排序，按序配对，以减小 mGA 中按相似性随机配对的计算开销，实际是以适应值作为相似性的测度。

将一个种群中分为多个子种群，使每个子种群对应一个极值点吸引域的多种群方法，当极值点数量增大时，也需要子群数量同步增大，将导致种群规模很大，更重要的是，各个子种群的进化步伐不会同步，有的子种群先收敛到局部极值点，而有的可能还需要漫长的进化过程，如果强制性地同步计算，已收敛的子种群的进化计算是浪费，而且可能因为变异，跳出该极值点，造成极值点丢失，前功尽弃。序列小生境技术[157][161]（sequential niche technique）对同一目标问题反复计算，每次运行结束时保持已经找到的最优解，在下次运行时，用类似计算共享适应值的方法降低已经找到的最优解的σ_{share}范围内的所有解的适应值，以减少重复访问的概率。

5.2.2 多种群 ES

与多种群 GA 比较，多种群 ES 的文献少得多，而用于多模态函数优化问题的更少，这不能不说是 ES 研究薄弱的地方。

Izumi[54]在 ES 中引入了相互竞争的子种群，以平衡 ES 开采与探索能力，改进全局搜索性能。Porter[55]受 Wright 的漂移平衡理论启发，提出了小生境进化策略 NES（niche ES），算法由多个小生境各自独立地同时进化，每隔一定进化代数，使劣质种群消亡、重生，小生境中的最优个体重组产生新的个体，NES 的目的主要是为了并行计算的需要和求单一的全局最优解。Zhang[58]在 ES 和 EP

中，应用小生境技术，直接借用GA的共享和清除方法，其原理是在个体周围给定的半径（清除半径，clearing radius）范围内，只允许最多κ个个体保留，其他个体清除，对个体周围给定半径（共享半径，sharing radius）内的个体使用共享，以降低适应值，共享半径和清除半径的确定需要适应值函数的有关先验知识，如极值点的数量，并假定极值点均匀分布。

Kim[59]提出了基于ES解多模态函数优化问题的新算法，在算法中，提出了限制进化（restricted evolution）概念，个体只在其进化范围内进化，而进化范围是在进化过程中动态改变的，以此代替GA中的小生境半径，在算法中使用精英集合（elite set）保留已经找到的、相互之间有一定距离的优秀解，但距离的确定还是需要有关适应值函数的先验知识。

Beyer给出了一种多种群ES的通用描述形式，称为Meta－ES[80]（或称嵌套式ES，nested ES；分层ES，hierarchically organized ES）：

$$[\mu'/\rho'+\lambda'(\mu/\rho+\lambda)^{\gamma}]-ES \tag{5-5}$$

其中的“＋”号也可以是“，”号。这是一种内外双层进化策略，内层使用普通的$(\mu/\rho+\lambda)$策略，外层由μ'个使用$(\mu/\rho+\lambda)$策略的父种群产生λ'个子$(\mu/\rho+\lambda)$策略，在γ代内，子策略之间没有任何联系，每隔γ代，从λ'个子策略中选择μ'个策略作为下一代的父策略。$[\mu'/\rho'+\lambda'(\mu/\rho+\lambda)^{\gamma}]-ES$已有的应用方式为：1）内层ES完成局部搜索，求出极值点集合，外层策略对内层求出的极值点集作新的重组、进化操作[163][164]；2）对于混合搜索空间，如混合整数空间、具有连续权值的组合优化（如神经网络设计），这时，内层ES完成连续空间搜索，外层完成结构进化[165]。3）由外层ES优化内层ES的性能[166][167]。

5.2.3 多种群技术中有待解决的问题

多种群技术通过适应值共享、排挤等方法限制相似个体的过度繁殖、维持种群多样性、增加全局搜索能力取得了一定的成功，但还存在着一些需要进一步研究解决的问题：1）个体的相似性判别问题，大多数算法判别个体相似性是以个体间的欧氏距离或Hamming距离或适应值为依据的，没有考虑到适应值曲面的拓扑结构，与此相关的共享技术中，共享半径已成为其中一个复杂而且关键的参数，不合适的共享半径是导致同峰与异峰极值点的判别错误、出现极值点遗漏或产生多余极值点的主要原因；2）子种群的收敛性判别，对于多种群的进化算法，各个子种群的进化一般不会同步，如何判别子种群是否收敛，以确定子种群的消亡与再生，防止已收敛的子种群“过度”而无效地进化也是多种群技术中需要解决的问题。

以下将结合$(\mu+\lambda+\kappa)-ES$提出一种多种群的进化策略$m\times(\mu+\lambda+\kappa)-ES$，其中主要论述同峰与异峰极值点的判别方法、子种群的收敛判据，并提出山谷探索法隐式地解决相似性的判别问题。

5.3 多种群进化策略 $m\times(\mu+\lambda+\kappa)$-ES

5.3.1 算法结构

$m\times(\mu+\lambda+\kappa)-ES$建立在使用递减型变异步长、单基因变异方式的$(\mu+\lambda+\kappa)-ES$基础上，既可以求解多模态问题的多个极值点，也可以求单个全局最优解问题。

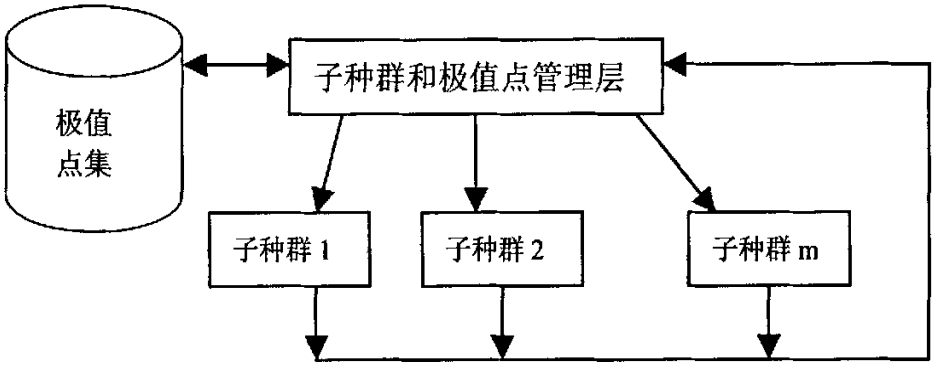


图 5-2　$m\times(\mu+\lambda+\kappa)-ES$结构

$m\times(\mu+\lambda+\kappa)-ES$算法结构如图5-2所示，它由三大部分组成，其一为$m$个子种群，每个子种群同时搜索不同的局部极值点，通过多个子种群实现全局搜索；其二为极值点集，它负责保存找到的极值点；其三为子种群和极值点管理层，当某个子种群收敛到一个极值点而不再进化时，通过管理层计算新加入的极值点与极值点集合中已有的极值点之间的距离，如果该新极值点与某个极值点距离比较小时，则用山谷探索法判别它们是同峰极值点还是异峰极值点，如果是同峰极值点则舍弃该新极值点，否则，加入到极值点集合中，并对极值点集合中的个体进行排序，该子种群重新初始化。当连续若干次没有找到新的极值点时，则认为已找出全部极值点，而极值点集合中最优者即为全局最优解。

5.3.2 算法参数的设置

$m\times(\mu+\lambda+\kappa)-ES$是一种通用的表示形式，根据所求问题不同种群参数稍有不同。

对于求单个全局最优解的优化问题，可以按Schwefel[11]推荐，μ和λ都可取较大的值，如分别为30，200。取较大值的初衷是为了实现多点同时搜索，增强全局搜索能力，事实上，只在进化初期，有利于得到较强的全局搜索能力，而在进化后期，由于遗传漂移的影响，种群中的所有优秀个体基本上是同类个体的后代，与$\mu=1$时效果基本相同。另外，在$m\times(\mu+\lambda+\kappa)-ES$中主要依靠多个子种群及其消亡与再生实现全局搜索，故在$m\times(\mu+\lambda+\kappa)-ES$中，$\mu=1$即可。$\kappa\neq 0$时，具有强全局搜索能力的均匀变异和多种群技术同时使用，可以获得更强的全局搜索能力。

对需要求多模态函数所有局部极值点和全局极值点的问题（简称为MFO，multi-modal function optimization），每个子种群的作用是在其所在极值点的吸引域内完成局部搜索，因而无需使用均匀变异增强全局搜索能力，故取$\kappa=0$，初始变异步长取得较小，使每个子种群只在其局部的区域搜索，而全局搜索功能由各个子种群的消亡、再生实现，此时，$m\times(\mu+\lambda+\kappa)-ES$实际为$m\times(1+\lambda)-ES$。以下主要针对模态函数优化问题论述$m\times(\mu+\lambda+\kappa)-ES$。

5.3.3 子种群的消亡与再生

$m\times(\mu+\lambda+\kappa)-ES$的每个子种群各自以$(\mu+\lambda+\kappa)-ES$执行进化计算，一个子种群在两种情况下需要消亡、再生，其一是当子种群已经收敛到极值点时；其二是已有同类更优秀的子种群时。为此，需要判别子种群的是否收敛和对子种群分类。

子种群收敛性的判别

对于多解优化问题，当$m\times(1+\lambda)-ES$中某一子种群收敛到一个极值点时，应及时停止进化，减少不必要的计算，也防止丢失该极值点。

对于式（1-1）所描述的最大值优化问题，设$\mathbf{x}^*=(x_1^*,x_2^*,...,x_n^*)$是一个极大值点，它可以是全局的或局部的。对于给定的误差$\varepsilon>0$，如果$|x_i-x_i^*|<\varepsilon$，则称变量x_i已收敛。如果对于$\forall i\in\{1,2,...,n\}$，满足$|x_i-x_i^*|<\varepsilon$，则称$\mathbf{x}=(x_1,x_2,...,x_n)$已收敛到$\mathbf{x}^*$。

对于变异步长不断缩减的$(\mu+\lambda+\kappa)-ES$，每一个子种群在进化过程中，σ不断减小，当$\sigma=\sigma_{\min}$时如果连续k代没有进化，则认为该子种群已收敛到极值点，为了保证子种群以足够的可信度收敛到某个极值点，而又不致过多的计算，需要合理地确定k。为此，我们研究了$(1+\lambda)-ES$的进化概率和停止条件。

对于求最大值问题，如果父代$\mathbf{x}$产生的后代$\mathbf{x}'$满足$f(\mathbf{x}')>f(\mathbf{x})$，则称$\mathbf{x}'$得到了进化。

引理 5.1 对于单基因 Gauss 变异算子 $z\sim N(0,\sigma)$，当 $\sigma=\sigma_{\min}=\varepsilon/2$ 且 $\exists i\in N=\{1,2,...,n\}$ 满足 $|x_i-x_i^*|>\varepsilon$ 时，$(1+\lambda)-ES$ 一次变异产生进化的最小概率 $p_e\approx 1/(2n)$。

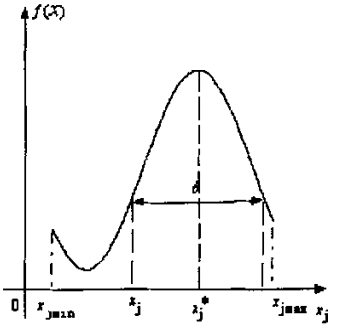


图 5-3 单基因变异的进化区间为 $(x_j,x_j+\delta)$

证明：显然，对于单基因变异算子 $z\sim N(0,\sigma)$，当 n 个变量中，$n-1$ 个变量已收敛，只有一个变量没有收敛时，$(1+\lambda)-ES$ 经一次变异产生进化的概率最小，不妨设 $|x_j-x_j^*|>\varepsilon$，$x_i=x_i^*,i,j\in N,i\neq j$，如图 5-3 所示，进化概率

$$p_e=\frac{1}{n}P(0<z<\delta) \tag{5-6}$$

考虑到在进化过程中，如果 g 代没有进化，则以某种法则减小 σ，直到 $\sigma\le\sigma_{\min}$ 时不再减小，则对于 $z\sim N(0,\sigma)$，当 $\sigma=\sigma_{\min}=\frac{1}{2}\varepsilon$，如果 $|x_j-x_j^*|>\varepsilon$，则

$$0.5\ge P(0<z<\delta)\ge P(0<z<2\varepsilon)=P(0<z<4\sigma_{\min}) \tag{5-7}$$

对于 $N(0,\sigma)$，由于 $P(-4\sigma\le z\le 4\sigma)=0.9999683\approx 1.0$，故：

$P(0\le z\le 4\sigma_{\min})\approx 0.5$，$P(0<z<\delta)\approx 0.5$。

$p_e\approx 1/(2n)$。　　引理 5.1 得证。

定理 5.1 对于单基因 Gauss 变异算子 z，当：

$z\sim N(0,\sigma)=N(0,\sigma_{\min})=N(0,\varepsilon/2)$ 时，$(1+\lambda)-ES$ 如果连续 $k\ge\frac{\ln(1-p_g)}{\lambda\ln\frac{2n-1}{2n}}$ 代没有进化，则 $\mathbf{x}=(x_1,x_2,...,x_n)$ 已收敛到 $\mathbf{x}^*$ 的可信度 $c\ge p_g$。

证明：由引理 5.1，当 $z\sim N(0,\sigma)=N(0,\sigma_{\min})=N(0,\varepsilon/2)$，$\exists|x_i-x_i^*|>\varepsilon$ 时，经连续 k 代进化计算至少有一次产生进化的最小概率为 $p_{ek}=1-(1-\frac{1}{2n})^{k\lambda}$。令 $p_{ek}=1-(1-\frac{1}{2n})^{k\lambda}\ge p_g$ 得

$$k\lambda\ln(1-\frac{1}{2n})\le\ln(1-p_g)$$

考虑到$1-\frac{1}{2n}<1,\ln(1-\frac{1}{2n})<0$有：

$$k\ge\frac{\ln(1-p_g)}{\lambda\ln\frac{2n-1}{2n}}。$$

即$\exists|x_i-x_i^*|>\varepsilon$时，经$k\ge\frac{\ln(1-p_g)}{\lambda\ln\frac{2n-1}{2n}}$代进化计算，至少有一次进化的概率$p_{ek}>p_g$。换句话说，如果经$k\ge\frac{\ln(1-p_g)}{\lambda\ln\frac{2n-1}{2n}}$代计算没有进化，则对$\forall i\in N$，$P(|x_i-x_i^*|\le\varepsilon)\ge p_g$。　　证毕。

定理 5.1 的证明借用了质量控制中定性抽样的思想[123]，把是否收敛看作产品是否合格，而每次变异相当于在无限大样本空间的一次抽样试验，如果变异成功（适应值改进），则相当于抽样试验产品不合格。要以一定的可信度保证产品合格（子种群收敛），需要有相应次数的抽样试验全部合格（不成功变异）。

引理 5.1 隐含了$\sigma_{\min}$的确定，取$\sigma_{\min}=\frac{1}{2}\varepsilon$。定理 5.1 可以指导我们在进化计算时，设计合适的停止准则，确定需要经多少代没有进化，就可以以可信度p_g保证所有的x_i满足$|x_i-x_i^*|\le\varepsilon$。例如对于含 50 个变量的优化问题，如果使用$(1+20)-ES$，要求所有$x_i$满足$P(|x_i-x_i^*|\le0.001)\ge0.9999$，则应在$\sigma_{\min}=\frac{1}{2}\varepsilon=0.0005$时，连续$k\ge\frac{\ln(1-p_g)}{\lambda\ln\frac{2n-1}{2n}}=45.8$代没有进化，换言之，当$\sigma_{\min}=\frac{1}{2}\varepsilon=0.0005$时，如果连续$k\ge45.8$代没有进化，则所有$x_i$满足$|x_i-x_i^*|<\varepsilon$的可信度大于99.99%。

同类个体的判别——区域型山谷探索法 HVS1

同类个体的判别是多解优化问题的关键，是子种群分类和同峰与异峰极值点判别的基础。作者在文献[134]中提出了山谷探索法 HVS（hill-valley searching）对个体按所在吸引域分类。如图 5-4 所示，当 A、B 是两个不同吸引域内的个体时，在它们所界定的区域内必然包含山谷，而 C、D 在同一个吸引域时，在它们

界定的区域则不包含山谷.

HVS的基本原理就是探索两个个体之间是否存在山谷，为此，以这两个个体的变量值为边界域，在这个小区域内用$(1+1)-ES$进行求最小值（对于求最大值的优化问题而言）的进化计算，经过若干代（如100代），如果能找到比这两个个体的函数适应值小的个体，则说明这两个个体之间存在“山谷”，它们属于不同极值点的吸引域，是不同类的个体，否则，它们是同类个体。在上述探索过程中，使用了$(1+1)-ES$这种小种群的进化策略，其计算成本是比较小的，因为搜索空间很小，而且并不要找出最低的谷点，只要判定是否存在山谷.

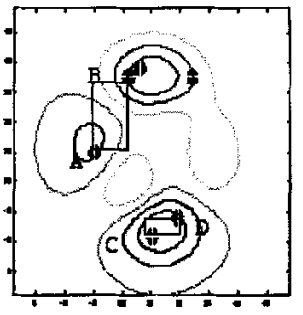


图 5-4　山谷探索法判别同峰极值点原理

这种HVS是通过搜索两个个体所界定的区域是否包含有山谷，故称区域型山谷探索法HVS1。

同类个体的判别——连线型山谷探索法HVS2

区域型HVS1以两个个体$\mathbf{x}_1,\mathbf{x}_2$所界定的区域作为搜索区间，与适应值函数的维数$n$相同，尽管计算成本不很大，但在一个区间探索计算量还是较多，为此，对区域型HVS做了改进，提出了连线型山谷探索法HVS2，其基本原理是只在$\mathbf{x}_1,\mathbf{x}_2$的连线范围内探索是否存在山谷，这样把探索空间从$n$维转化为一维空间，使探索效率得到进一步提高。此时，后代$\mathbf{x}$由下式产生：

$$\mathbf{x}=\mathbf{x}_1+\alpha(\mathbf{x}_2-\mathbf{x}_1) \tag{5-8}$$

其中$\alpha=random(0,1)$是均匀分布的随机变量，也可以直接取$\alpha=0.1,0.2,...,0.9$。设适应值函数为$f(\mathbf{x})$，如果存在$\exists\mathbf{x}$满足：

$$f(\mathbf{x})<f(\mathbf{x}_1)-\xi \quad \text{and} \quad f(\mathbf{x})<f(\mathbf{x}_2)-\xi \tag{5-9}$$

则说明$\mathbf{x}_1,\mathbf{x}_2$之间存在山谷，它们是不同类个体。式(5-9)中，$\xi$是控制山谷深度的阀值，可取$\xi=(10\sim100)\varepsilon$，以避免计算噪声的干扰。这样，探索山谷的过程之间作若干次试验即可，不需要使用$(1+1)-ES$。

在使用HVS进行类判别前，可以设置一个距离阀值σ_{share}（也可以称为峰半径或共享半径），只有当两个个体的距离小于该阀值时，才启用HVS判别，否则就认为它们是异类个体。这个距离阀值与常规的共享半径相比，对它的准确性要求很低，对于距离较大的个体，就认为它们是非同类的，以减小计算量，如

可以设置为$\sigma_{share}=(10\sim100)\varepsilon$，而$\alpha$的取值间隔$\Delta\alpha$也可以与之相关，如$\Delta\alpha=\varepsilon/\sigma_{share}$。

同类子种群的判别

由于存在遗传漂移，经过进化若干代，种群中的个体将逐步由其中的最优个体组成，当子种群为$(1+\lambda)-ES$时更是如此，同时考虑到减小计算开销，在判别两个子种群是否同类时，就根据它们的最优个体是否同类为依据，如果这两个个体同类，则认为这两个子种群亦同类，个体的同类判别方法使用上述山谷探索法。

由于个体是以均匀分布的方式产生的，难免存在同类个体和同类子种群；同类子种群中较差的（以它们的最优个体的适应值为依据）消亡，重新初始化实现再生；另外子种群中最优个体也需要与已找到的极值点作同类判别，是同类时，该子种群也将重新初始化。

同类极值点的判别

当一个子种群收敛后，它所求出的新极值点在放入极值点集合之前，先要判别在极值点集中是否已有同类极值点，这时，只要取新极值点逐个与极值点集合中的每个极值点用HVS即可。当已有同类极值点时，该新极值点舍弃，否则，加入到极值点集合中。

这样，同类子种群的判别以及同类极值点都可以通过同类个体的判别完成。

5.3.4 算法实现

根据图5-2所描述的$m\times(\mu+\lambda+\kappa)-ES$算法结构及上述有关算法的论述，$m\times(\mu+\lambda+\kappa)-ES$的伪代码描述分别如Algorithm 5.1～5.5所示。Algorithm 5.1为算法的管理层，它控制各个子种群的进化计算；对已收敛的子种群最优个体加入极值点集合；每隔若干代（T_1），对子种群做同类判别，使同类子种群中的劣质子种群消亡，重新初始化。Algorithm 5.2为子种群的一代进行计算，通过变异产生新的后代个体，对它们评价、选择，修改变异步长σ，做收敛性判断。Algorithm 5.3实现了HVS2；Algorithm 5.4实现了两个子种群是否同类的判别；Algorithm 5.5完成新增极值点的处理。

5.3.5 仿真计算

为了验证$m\times(\mu+\lambda+\kappa)-ES$同时求解多模态函数多个极值点的能力，选择一组典型的测试函数进行了仿真计算，在计算过程中，所有计算参数为：使用5×(1+5)-ES，当$\sigma=\sigma_{\min}$时连续50次没有找到新的极值点则停止计算。计算结果

Algorithm 5.1：$m\times(\mu+\lambda+\kappa)-ES$

所有子种群初始化；

while（停止条件不满足）

{

for i=1 to m　{ Evolve（）；} //所有子种群执行一代进化

对于已收敛的子种群 popi，调用 AddOptima（popi.indbest）把最优个体保存到极值点集，popi 初始化；

每隔T_1代，调用 Cluster（）判别同类子种群，同类子种群中劣质子种群初始化；

}

Algorithm 5.2 ：Evolve（）　　//子种群执行一代进化

产生λ个 Gauss 变异子个体；

产生κ个均匀变异子个体；

计算$(\lambda+\kappa)$个子个体的适应值；

从$(\mu+\lambda+\kappa)$个个体中选择μ个保存；

如果连续T_σ代没有进化，则缩减σ；

当$\sigma=\sigma_{\min}$时，如果连续T_k代没有进化，则设置“已收敛”标志；

Algorithm 5.3 : bool HVS2(ind1,ind2)　//同类个体判别

计算个体 ind1 与 ind2 的欧氏距离 dis；

if　$dis>\sigma_{share}$　return false；//返回 false 表示是异类个体

$\alpha=\varepsilon$

while（$\alpha<\sigma_{share}$）

{

$\mathbf{x}=\mathbf{x}_1+\alpha(\mathbf{x}_2-\mathbf{x}_1)$；

if　（$f(\mathbf{x})<f(\mathbf{x}_1)-\xi$ *and* $f(\mathbf{x})<f(\mathbf{x}_2)-\xi$）　return true；

$\alpha=\alpha+\varepsilon$；

}

return false;

Algorithm 5.4： bool Cluster(pop1,pop2) //同类子种群判别

Return HVS2(pop1.indbest,pop2.indbest);

//其中，pop1.indbest、pop2.indbest分别为子种群pop1、pop2的最优个体

Algorithm 5.5： AddOptima（ind）//新增极值点的管理

For i=1 to opti　　//opti为已有的极值点数量

{ If HVS2(ind,indi) return; }　//indi为第i个极值点个体

ind加到极值点集合；

opti＝opti＋1；

为每个函数重复50次计算后的平均值。

1）Himmelbau函数

$f_1(x,y)=660-(x^2+y-11)^2-(x+y^2-7)$，$x,y\in[-6,6]$，此函数为线性不可分的等高、非等距多峰函数，有4个极值点。在重复的50次计算中，每次都找到了4个极值点，且没有多余的极值点，计算结果如表5-1所示。

表5-1 函数1计算结果

序号	变量		函数值
	x	y	
1	-2.80511602	3.13127953	659.99999996
2	3.58442682	-1.84807122	659.99999996
3	-3.77932909	-3.28315895	659.99999993
4	2.99995064	2.00001894	659.99999992

2） Shekel’s Foxholes函数

$$\min\frac{1}{f_2(x_1,x_2)}=0.02+\sum_{j=1}^{25}\frac{1}{c_j+\sum_{i=1}^{2}(x_i-a_{ij})^2},\quad x_1,x_2\in[-65.536,65.536]$$

其中：

$$a_{ij}=\left(a_{ij}{}^0 a_{ij}{}^1 a_{ij}{}^2 a_{ij}{}^3 a_{ij}{}^4\right)$$

$$a_{ij}^{k}=\begin{pmatrix}-32 & -16 & 0 & 16 & 32\\ -32+16k & -32+16k & -32+16k & -32+16k & -32+16k\end{pmatrix}$$

$i=1,2;j=1,2,...,25;k=1,2,3,4$。此函数是典型的非等高多峰函数，共有 25 个峰且分布密度大。重复 50 次计算结果每次都找到表 5-2 所示的 25 个极值点，且没有出现多余的虚假极值点。

表 5-2　函数 2 计算结果

序号	x_1	x_2	函数值	序号	x_1	x_2	函数值
1	-31.97833369	-31.9783347	0.9980038	14	15.96614790	0.02884596	13.6186089
2	-15.98638872	-31.9703368	1.9920309	15	31.93341164	0.02960112	14.5630541
3	0.01321760	-31.9650893	2.9821051	16	-31.93168635	15.96563807	15.5038168
4	15.98158993	-31.9608358	3.9682501	17	-15.96335572	15.96335658	16.4409073
5	31.95868715	-31.9586886	4.9504912	18	0.03269635	15.96210514	17.3744065
6	-31.95391987	-15.9779369	5.9288451	19	15.96146720	15.96146860	18.3043095
7	-15.97531816	-15.9753318	6.9033356	20	31.92527655	15.96193316	19.2306781
8	-0.02179324	-15.9735858	7.8739929	21	-31.92644553	31.92644475	20.1534869
9	15.97242790	-15.9724282	8.8408359	22	-15.96022650	31.92235829	21.0726875
10	31.94343284	-15.9722478	9.8038979	23	-0.03666417	31.92093514	21.9884075
11	-31.94121032	0.02535000	10.763180	24	15.95860134	31.91959428	22.9006340
12	-15.96838701	-0.02670663	11.718699	25	31.92109612	31.92109605	23.8094344
13	-0.02777494	-0.02780687	12.670505				

3）函数 3：

$$f_3(x)=(1-x)\sin^4\left(\frac{5\pi}{(1+x)^4}\right),\quad x\in[0,1]。$$

此函数有非均匀分布且相距很近的 5 个极大值点。其图形如图 5-5 所示。50 次重复计算每次都准确找到如表 5-3 所示的 5 个极大值点。

4）函数 4：

$$f_4(x)=\frac{1}{100(x-5)^2+0.1}+\frac{1}{100(x-5.3)^2+0.6}+\frac{1}{100(x-8)^2+0.5}+\frac{1}{100(x+3)^2+0.11}$$

$x\in[-4,10]$。这是作者自己设计的测试函数，此函数有 4 个极大值点，且其中 2

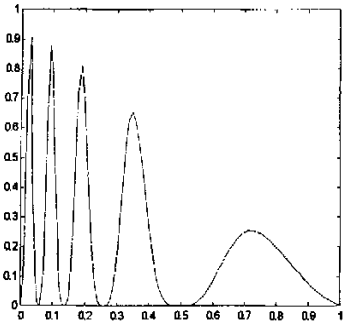


图 5-5 $f_3(x)$的图形

表 5-3 $f_3(x)$ 计算结果

序号	x	函数值
1	0.02660549	0.97335222
2	0.09309499	0.90681998
3	0.18876678	0.81101329
4	0.34924787	0.64978036
5	0.72240545	0.25312889

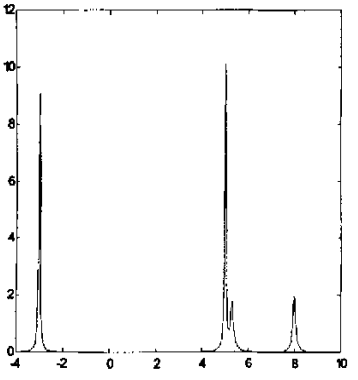


图 5-6 $f_4(x)$的图形

表 5-4 $f_4(x)$计算结果

序号	x	函数值
1	5.00003260	10.10544403
2	-2.99999999	9.09129313
3	7.99999779	2.00256425
4	5.29867969	1.77855025

个靠得很近，如图 5-6 所示。50 次计算结果全部找到表 5-4 所示的 4 个极大值点。

以上仿真计算表明，$m\times(\mu+\lambda+\kappa)-ES$ 具有准确地找到所有极值点的能力，而且不需要与峰半径或小生境半径有关的先验知识，在以上计算中，对不同问题即使使用相同计算参数也可以准确求出所有极值点，只影响计算量，表明 $m\times(\mu+\lambda+\kappa)-ES$ 具有很强的鲁棒性。

5.4 小结

为了增强$(\mu+\lambda+\kappa)-ES$的全局搜索能力以及求解多模态函数优化问题，在 $(\mu+\lambda+\kappa)-ES$ 基础上建立了 $m\times(\mu+\lambda+\kappa)-ES$ 。其基本思想是多个子种群同

时进化，当子种群收敛时，消亡、再生，子种群找到的极值点由极值点管理层管理，完成同峰极值点与异峰极值点的判别。

在多种群进化算法中，共享半径或者峰半径是进行聚类分析、同类个体、同类种群和同类极值点判别的关键参数，共享半径的确定需要有关适应值函数的先验知识，而且往往很难得到，且由于极值点的不均匀分布，使用共享半径难以准确判别同类与异类极值点。本章提出了山谷探索法，该方法直接利用适应值函数曲面的拓扑性质，无需共享半径，适应性强，具有准确地寻找所有极值点的能力，也可以应用于其他多模态函数优化算法。建立在单基因变异基础上的子种群收敛性判据、停止条件确定方法可以确保在给定的精度和可信度下收敛于极值点。

在多极值点的搜索过程中，如何避免极值点的重复搜索而又能确保不丢失极值点，是需要进一步研究和解决的问题。

第六章　进化算法仿真平台的研究

6.1 平台现状

进化算法的研究需要理论与仿真计算试验紧密结合，一些有关进化计算的参考书也列出了一些基本的算法程序，如文献[19]、[22] 和文献[65]分别列出了用 C 语言和 Fortran 语言编写的实例程序，这些程序为初学者提供了良好的示范例程。在很多互联网站、BBS，如 http://www.geatbx.com/ 也可以找到一些基于 Matlab 的遗传算法工具箱，如 gaot，为进化算法的研究、应用起到了很好的推动作用。然而，在算法仿真试验和应用研究过程中，由于进化算法参数多，需要试验多种不同类型测试函数，对算法运行过程做仔细的观察研究，对算法的参数做深入的对比分析，深感缺乏一个方便的试验平台，给算法的研究、相互交流带来诸多不便。作者在算法研究过程中，对算法仿真与研究平台的构建作了一些探索，并开发了仿真计算平台，为算法仿真和应用提供了很多方便，本章将介绍平台的设计思想，并以两个应用实例说明算法及仿真平台的应用。

6.2 算法平台的构建

6.2.1 基本要求

作为一个通用型的算法仿真研究与应用平台，需要考虑以下主要方面：

开放性　进化算法本身也在不断地发展和“进化”中，平台要能够方便地嵌入其他算法，对算法开放；另一方面，仿真试验和应用需要面向各种各样的适应值函数，这些不同的适应值函数及相关的领域知识能方便“无缝”地连接到平台，所以平台需要对应用对象（适应值函数）开放。

界面友好　由于进化算法需要多种策略参数，需要多次试验，某些参数可能对某类适应值函数更有效，对另外的适应值函数效果较差，这些参数的修改需要友好的用户界面。在算法进行过程中，需要对计算进程作仔细观察研究，也需要良好的界面，便于选择观察感兴趣的数据。

便于数据保存、统计、处理　要设计一个完整的数据处理、图形显示程序需要付出很大的努力，比较可行的方法是把这部分功能交给 MatLab©实现，平台只负责数据的保存和一些计算过程中的特定数据处理，平台与 MatLab©之间通过数据文件互相连接。

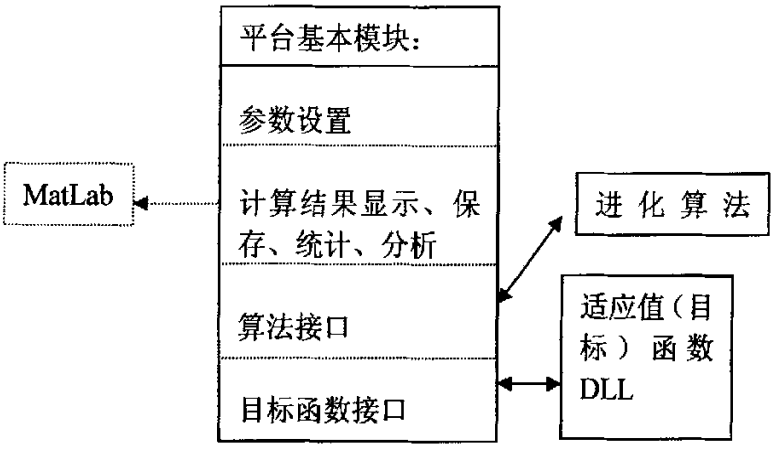


图 6-1 仿真计算平台模块结构

现代软件技术的发展为以上技术的实现提供了有效的手段，如动态连接库DLL（dynamic link library）、组件对象模型 COM（component object model）、各种控件技术如 Automation、ActiveX 等使不同程序、进程之间可以实现方便的连接[168][169]；Windows©、MatLab©，Linux 等软件系统为开发友好的软件界面提供了良好的保证，通过对话框、图形界面等可以方面地修改、调整算法参数。作者构建的平台框架如图 6-1 所示，它包括三大模块：平台基本模块、目标函数 DLL 模块、算法 DLL 模块组成。

基本模块负责算法参数的设置、算法的启动、暂停、终止控制、计算结果的显示、保存、统计、分析，它也包含了与进化算法 DLL 和适应值函数 DLL 的接口；进化算法 DLL 通常作为一个子进程（thread），由平台基本模块控制其启动、暂停、终止，它每隔一定的进化代（由平台设置）向平台报告当前种群、个体的有关状态信息，供观察、分析、处理；适应值函数 DLL 完成适应值的计算，它通过算法 DLL 调用，返回函数适应值。

6.2.2 程序设计

进化算法的程序设计主要有两类，一类以 MatLab©为基础平台，开发算法工具箱[170][171]；另一类利用 C、C＋＋等程序设计语言直接开发算法程序[172]。作者使用了后一种方式，在 Windows 环境下，用 Visual C＋＋实现，C＋＋程序可以很方便地与 MatLab©连接[173][174]。

虽然进化算法有多种程序实现方式，但作者认为，进化算法与现代程序设

计中的面向对象程序设计（OOP：object-oriented programming）方法、智能体（Agent）思想具有自然的联系，用 OOP 方法、Agent 思想实现进化算法，程序结构清晰、模块性强，可维护性好。程序设计和 Agent 技术不是本文的主要内容，本文只作一些初步的探索，以下只介绍在算法平台构建过程中的基本思想。

在面向过程的结构化程序设计中，数据和过程（函数）分离，数据是过程处理的对象，面向对象程序设计 OOP 则把客观事物的属性（数据）和行为（过程）抽象为一个整体对象，把属性和行为通过类（class）封装（encapsulation）在一起，形成一个完整的整体——类（class）、类对象（object），类属性（成员数据）表征对象的属性，类成员函数（member function）是处理属性数据的功能模块，是对象的行为。OOP 的中心是围绕三个主要概念：封装、继承（inheritance）和多态性（polymorphism），通过继承、派生（derivation）可以构成一个多层次的类结构，实现从上到下的瀑布式程序结构。多态性允许使用相同名称的函数以不同的实现方法对不同对象进行处理。

从 Agent 的角度看，类就是 Agent 的程序实现，一个类对象就是一个 Agent，Agent 的行为对应类的成员函数，Agent 的属性就是类的成员数据[176]。

虽然 OOP 主要是为了大型程序设计提出的，但由于它结构清晰，模块化强，易于维护，在进化算法程序设计中，把个体、种群抽象为类对象，是一种非常自然的程序模型[172]。

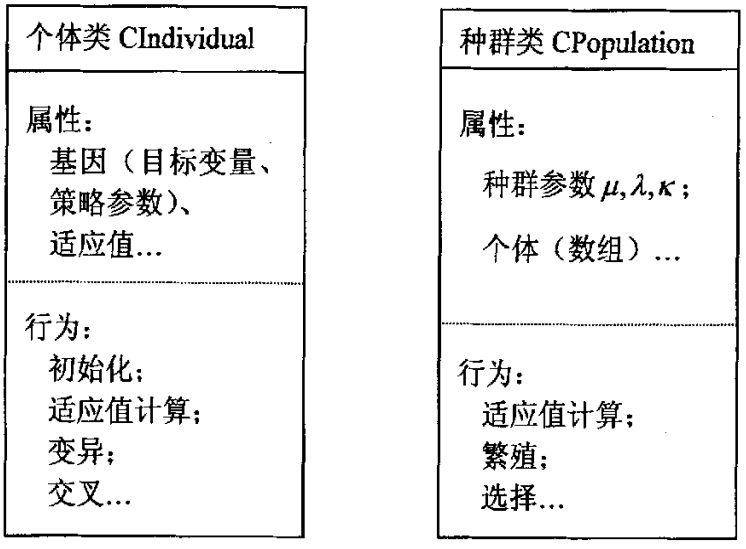


图 6-2　个体、种群类结构

图 6-2 为个体和种群的类结构图，个体类 CIndividual 的成员数据包括目标变量、策略参数组成的基因数组、适应值等，个体公共数据以类的静态成员数据表示，如变量范围，适应值函数指针等，个体的行为包括适应值函数计算，变异，与其他个体的交叉等。种群由个体组成，种群类 CPopulation 的数据成员包括个体数组，种群的参数如 μ,λ,κ 等，公共数据包括繁殖策略、选择策略等参数，放在类的静态数据成员中，种群类的成员函数包括繁殖、选择、适应值计算（循环调用每个个体的适应值函数）。

平台程序在 Windows 下用 Microsoft Visual C++（VC）设计，使用 VC 的 MFC（Microsoft foundation class，微软基础类）和 Doc/View 单文档结构[168][169]。所以，程序中除了与算法直接有关的 CIndividual、CPopulation 类以外，还包括 MFC Wizard 生成的框架类、文档类、视图类、应用类等。框架类是 Doc/View 结构中的容器类，管理文档类、视图类。文档类负责保存、输入算法的有关参数、保存计算结果；视图类是平台的界面，算法参数的设置、修改、算法启动、停止和计算结果的显示等都在视图类实现；应用类是平台的入口。仿真平台的主要对话框、窗口如图 6-3 至 6-6 所示。

图 6-3 是仿真计算平台主窗口，包含 Windows 窗口的常用部分，菜单、工具条、客户区、状态栏。进化计算过程的起动、暂停、停止控制，打开参数设置对话框等操作都可以通过菜单或工具条完成，客户区负责显示计算结果，状态栏显示算法的进展等附加状态数据。图 6-4 为算法参数的总体设置对话框，其中包括算法选择、变异方式选择、繁殖方式选择、子种群数量、计算结果的保存控制、停止准则的设置、重复计算次数等。图 6-5 为子种群参数的设置对话框，用于设置 μ,λ,κ，变异步长的初始值、最小值，缩减系数，缩减间隔，子种群停止准则。图 6-6 适应值函数设置对话框，用于选择适应值函数，设置变量个数、变量的变化范围。平台内部已经嵌入了近 40 个典型测试函数，除此之外，还可以使用外部动态连接库 DLL，这样，计算新的适应值函数无须修改平台程序，有利于平台程序的维护和知识产权的保护。

所有设置的参数都通过参数文件保存。除以上主要对话框以外，还有若干辅助对话框，如数据显示、统计分析设置等。

为了说明算法仿真平台的应用方法和操作步骤，以下给出两个应用实例，它们都在相关的科研项目中得到验证，证明了仿真平台的实用性和有效性。

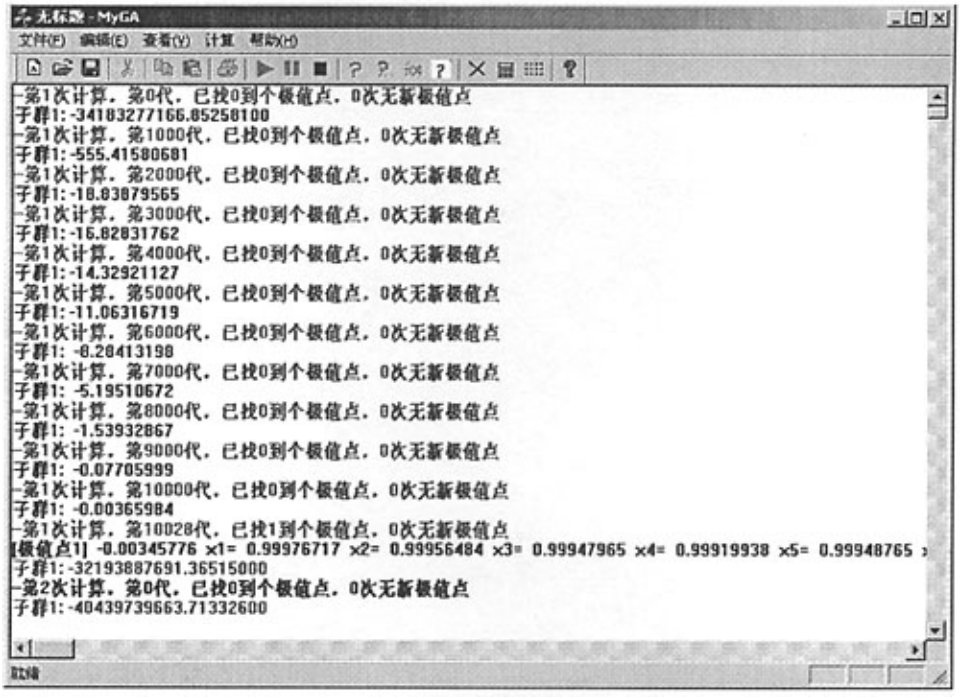


图 6-3 仿真计算平台主窗口

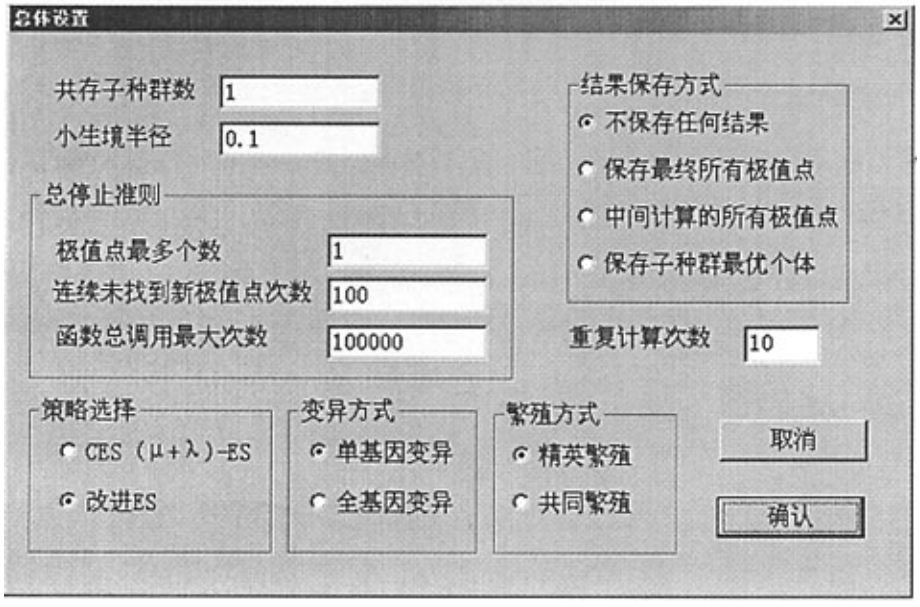


图 6-4 算法参数总体设置界面

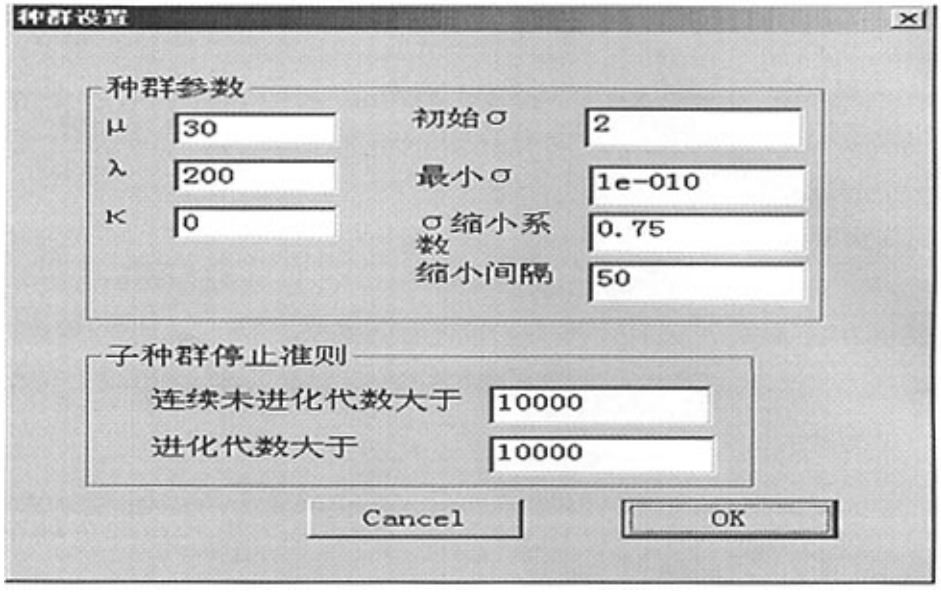


图 6-5 子种群参数设置界面

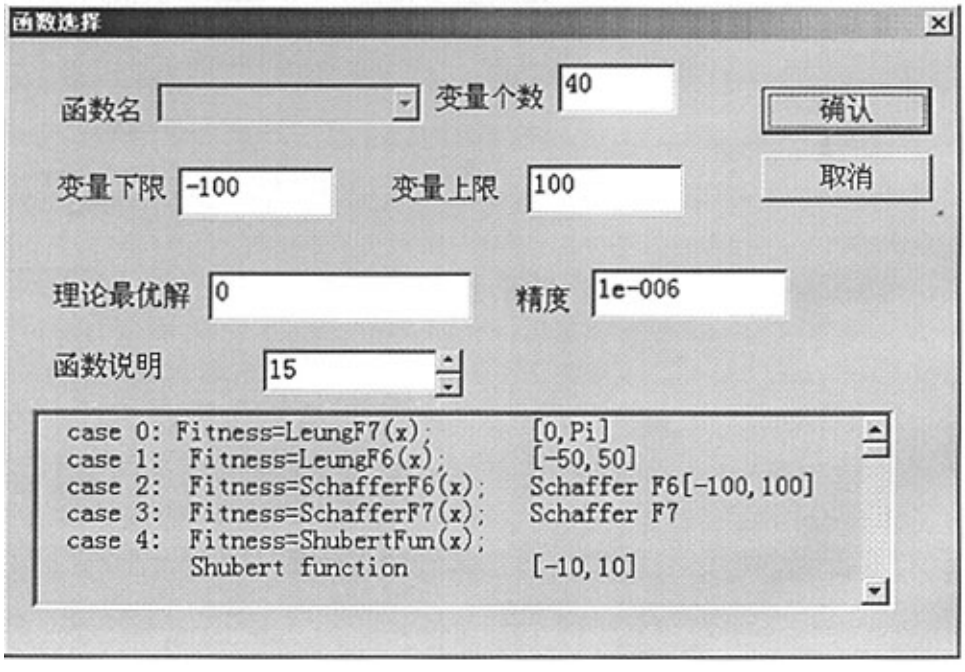


图 6-6 适应值函数选择、设置界面

6.3 应用 1——复杂机电传动控制系统参数的优化组合

平整机主传动速度控制系统参数的优化组合是国家自然科学基金重点资助项目“复杂机电系统耦合分析与解耦控制”（项目号：59835170）的一个子课题。平整机是轧制汽车板、镀锡板、彩色板等高附加值钢带板材的关键设备之一，是一种典型的复杂机电系统。由于平整机在轧制过程中需要进行频繁地启、停操作（一卷带材的轧制时间约为 4～5 分钟），因此，主传动速度控制系统的快速性、稳定性对产品的产量和质量有着直接的影响，如果系统参数调整不好，造成工作辊的速度超调量太大、调整时间过长、鲁棒性差，在轧制过程中容易被偶发因素激发系统振荡，带材的前后张力发生波动，轧件表面出现“振痕”。

6.3.1 系统结构与数学模型

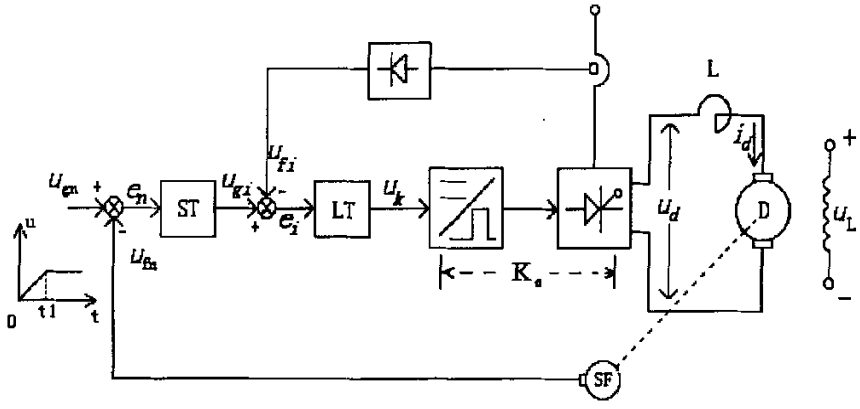


图 6-7 平整机主传动速度控制系统原理图

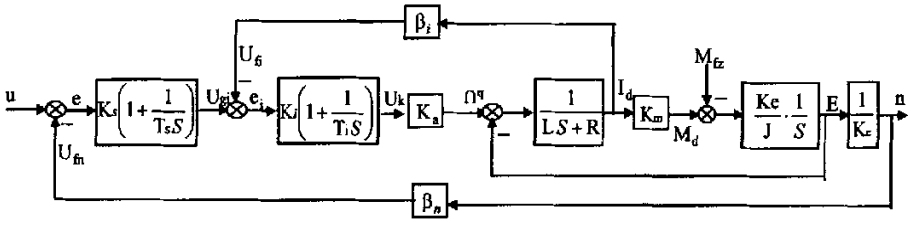


图 6-8 主传动速度控制系统传递函数框图

平整机主传动速度控制系统原理图[177]如图 6-7 所示，相应传递函数框图如图 6-8 所示。该系统是一个典型的双闭环直流调速传动系统，电流调节器（LT）和速度调节器（ST）为模拟比例积分环节，LT 的比例系数 K_i 和积分时间常数

T_i(sec)、ST 的比例系数 K_s 和积分时间常数 T_s(sec) 都是待优化的可调参数，其调节范围为：

$$\begin{cases} K_i \in [5,92] \\ T_i \in [0.1,4.7] \\ K_s \in [5,92] \\ T_s \in [0.1,4.7] \end{cases} \tag{6-1}$$

其他参数为不可调参数，其中整流系数 $K_a=125.8$、电流反馈系数 $\beta_i=10/2048$(v/A)、速度反馈系数 $\beta_n=10/984$(v/r.min^{-1})、电动机电势系数 $K_e=1.03$(v/r.min^{-1})、电磁转矩系数 $K_m=9.5$(Nm/A)、电枢电阻 $R=0.035\Omega$、电枢电感 $L=6(mL)$、折算到电动机轴上的总转动惯量 $J=1.952105/375$(Nm^2)。u、M_{fz}、n 分别为系统的速度控制量、负载转矩，电动机转速。

图 6-8 从左到右依次取各积分器的输出为状态变量 x_1、x_2、x_3 和 x_4，则可以写出系统的状态空间方程：

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}u + \mathbf{D} \\ n = \mathbf{C}\mathbf{X} \end{cases} \tag{6-2}$$

上式中，$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 \end{bmatrix}^{\mathrm{T}}$，　$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & -\frac{\beta_n}{J} \\ \frac{K_s}{T_s} & 0 & -\frac{\beta_n}{L} & -\frac{\beta_n k_s}{J} \\ \frac{K_i K_s K_a}{T_s} & \frac{K_a K_i}{T_i} & -\frac{R+\beta_i K_a K_i}{L} & -\frac{K_e + K_n K_i \beta_n}{J} \\ 0 & 0 & \frac{K_m}{L} & 0 \end{bmatrix}$$

$\mathbf{B} = \begin{bmatrix} 1 & K_s & K_a K_s K_i & 0 \end{bmatrix}^{\mathrm{T}}$，$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1/J \end{bmatrix}$，

$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & -\mathrm{M}_{fz} \end{bmatrix}^{\mathrm{T}}$。

6.3.2 适应值函数

根据平整机对速度控制系统的工艺要求，综合考虑速度最大超调量、系统的响应的快速性和精确性，系统的优化目标可表示为求以下目标函数的最小值：

$$f(\mathbf{y}) = \int_0^\infty t e^2(\mathbf{y},t)dt \tag{6-3}$$

其中，$\mathbf{y} = (K_i, T_i, K_s, T_s)$ 为目标变量（此处 $\mathbf{x}$ 已用于表示状态变量，为区别

起见，改用$\mathbf{y}$表示目标变量），其取值范围如式(6-1)所示。t为时间，$e(\mathbf{y},t)$为系统控制输入$u(t)$与输出$\beta_n n(\mathbf{y},t)$之间的误差，如式(6-4)所示

$$e(\mathbf{y},t)=u(t)-\beta_n n(\mathbf{y},t) \tag{6-4}$$

实际计算时，把$e(\mathbf{y},t)$、$u(t)$、$\beta_n n(\mathbf{y},t)$、整个控制系统（式(6-2)）离散化，设采样周期为T_{samp}(sec)，t对应kT_{samp}，简写为k，适应值函数转化为求和表达式，如式(6-5)、(6-6)所示：

$$e(\mathbf{y},k)=u(k)-\beta_n n(\mathbf{y},k) \tag{6-5}$$

$$f(\mathbf{y})=\sum_{k=1}^{N} k\cdot[u(k)-\beta_n n(\mathbf{y},k)]^2 \tag{6-6}$$

理论上，$N\to\infty$，实际计算时，取4～5sec对应的采样次数（$4\sim5\text{sec}/T_{samp}$）即可。优化过程中，通常$u(k)$为单位阶跃函数

$$u(k)=\begin{cases}1 & k=0,1,2,...\\ 0 & else\end{cases} \tag{6-7}$$

$n(\mathbf{y},k)$通过解系统式(6-2)离散化后的状态方程求得。

6.3.3 计算过程与结果

在仿真计算平台作优化计算时，只要根据式(6-6)编写适应值函数程序，编译、联接生成动态连接库DLL文件，在仿真计算平台设置相应的进化算法参数，如变量个数、计算精度要求等，即可进行优化计算。经计算得到一组优化参数$\mathbf{y}^*=(24.4455,4.1294,91.8165,0.1842)$，实际应用表明，这组参数获得了满意的控制效果，图6-9为设定工作辊速度为584r/min时主传动速度控制曲线，其稳态误差为零，上升时间、调节时间和超调量均非常理想。

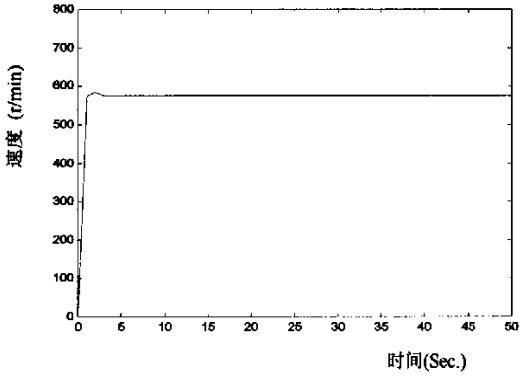


图6-9 主传动最优速度控制曲线

6.4 应用 2——足球机器人的最优控制

机器人足球是近几年发展起来的机器人研究与竞赛项目，RoboCup 和 FIRA 等国际性组织每年都举办国际性和区域性比赛，吸引了世界各地数百所大学和研究机构参加。机器人足球已成为研究人工智能、MAS（multi-agent system）等的一个理想平台，并促进了新兴的电子竞技产业的发展。

机器人足球赛是作者主持的学校资助项目。机器人足球赛中，每个球队由若干个（如 3Vs3、5Vs5,11Vs11 分别为 3、5、11 个）足球机器人（以下简称为机器人）队员组成，它们在各自设计的比赛策略程序控制下，队员相互配合完成相应的动作，如守门、拦截、射门、防守、进攻等。当两个参赛队机器人物理性能参数相同的情况下，机器人的速度主要由控制策略决定。显然，队员的速度对比赛起着很重要的作用，如在争球过程中，能在最短时间内到达球所处位置，则可以“先人一步”控球，掌握主动权，否则，对方球员先得到球，陷入被动。在此，最优控制的目标为机器人到达指定位置的时间（到达时间）t 最短。机器人的运动及其运动轨迹由左、右轮的速度 v_l 和 v_r 控制，到达时间 t 由 v_l、v_r 和运动轨迹决定，但 t 与 v_l、v_r 很难用直接的函数关系表达，用传统的求解方法（如导数方法）难以求解。

6.4.1 机器人运动方程

考虑到对机器人的控制是通过每隔一定时间间隔发出控制指令完成的，是一个 “天然”的离散时间控制系统，所以，可以直接用离散时间系统方程描述机器人运动，如图 6-10 所示，设机器人在时刻 k 的位姿 $[x_k, y_k, \theta_k]^T$，$P_k = (x_k, y_k)$ 为当前位置，θ_k 为方向，速度向量 s 定义为[178]：

$$s = \begin{bmatrix} v_k \\ \omega \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ -1/L & 1/\mathrm{L} \end{bmatrix} \cdot \begin{bmatrix} v_{lk} \\ v_{rk} \end{bmatrix} \tag{6-8}$$

其中，v_k 和 ω_k 分别是 k 时刻机器人质心的瞬时线速度和角速度，v_{lk} 和 v_{rk} 分别为机器人实际的左、右轮速度，L 是机器人两轮之间的距离（轮距）。机器人在下一个采样时刻的位姿为（T_{samp} 为采样控制周期）：

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} T_{samp}\cos\theta_k & 0 \\ T_{samp}\sin\theta_k & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \tag{6-9}$$

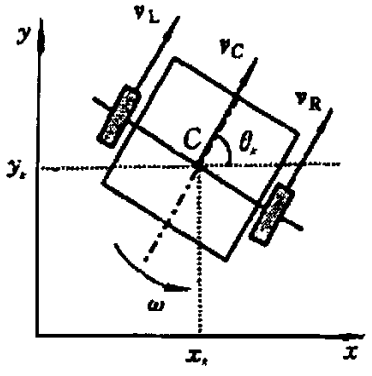


图 6-10 足球机器人模型

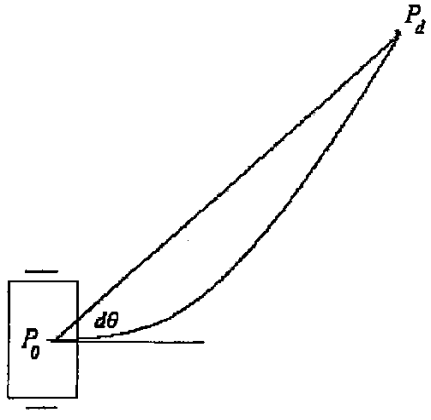


图 6-11 机器人运动控制

6.4.2 机器人轮速的动力学方程

经过试验和模式识别表明，机器人的轮速可以用一阶系统描述：

$$v_{i(k+1)} = av_{ik} + bu_{ik} \tag{6-10}$$

其中，$i = l,r$，a,b为机器人的动力学参数，u_{ik}为给定控制量（即左右轮的给定控制速度），如在 Simurosot 5Vs5 平台，经过系统辨识，得到 $a \approx 0.94, b \approx 0.0026$，$u_i = [u_{i\min}, u_{i\max}] = [-125,125]$。

6.4.3 最优控制的描述与实现

由于在实战中，每秒钟需要对机器人发出约 60 次控制指令，球队有 5 个（5Vs5）甚至 11 个（11Vs11）队员需要控制，在这么短的时间内不可能用进化算法完成实时最优控制所需的计算，可行的办法是在离线状态下，得到最优控制参数，再应用到实战中。

如图 6-11 所示，设目标位置$P_d = (x_d, y_d)$，机器人初始位姿$[x_0, y_0, \theta_0]^T$，零初始状态，即$[v_{l0}, v_{r0}, \omega_0] = [0,0,0]$。$\overline{P_0P_d}$的方向角$\theta_d$与$\theta_0$之差$d\theta = \theta_d - \theta_0$，为了尽快到达目标位置，根据$d\theta$的符号，控制方式可以设计为如下所示：

$$if \quad d\theta > 0 \quad \begin{cases} u_r = u_{r\max} \\ u_l = u_{l\max} - \alpha \cdot d\theta \end{cases}$$

$$else \quad \begin{cases} u_l = u_{l\max} \\ u_r = u_{r\max} - \alpha \cdot d\theta \end{cases}$$

α 是待优化的控制参数。最优控制优化问题描述为：确定合适的 α，使机器人第一次满足：

$$dis=\sqrt{(x_d-x_k)^2+(y_d-y_k)^2}<\varepsilon \tag{6-12}$$

时的 k 最小，其中，$\varepsilon>0$ 为位置误差，$P_k=(x_k,y_k)$ 由式(6-9)、(6-10)、(6-11)确定，尽管这个优化问题的变量只有一个 α，但由于指标与 α 无直接的解析表达式，只能用程序描述，伪代码如 Algorithm 6.1 所示。这种优化问题用传统方法难以求解，用进化算法求解则比较方便。

在算法仿真平台上，只要对 Algorithm 6.1 所示的适应值函数编译、联接生成 DLL 后，设置好有关参数，即可完成优化计算，得到最优控制参数，按此参数控制机器人的运动，试验结果表明，机器人的响应速度、跟踪目标的能力得到明显提高，应用于比赛中取得了满意的成绩[179]。

Algorithm 6.1　适应值函数的实现

```
function(α)
{
    k = 0
    do {
        求 dθ_k = d_d − θ_k ;
        由(6-11)计算 u_lk, u_rk ;
        由式(6-10)计算 v_lk+1, v_rk+1 ;
        由式(6-9)计算 [x_k+1, y_k+1, θ_k+1]^T
        由式(6-12)计算 dis ;
        k = k + 1;
      } while(dis > ε);
    return k;
}
```

6.6 小结

本章论述了进化算法仿真平台的研究和构建，以及平台应用计算实例。仿真平台的构建应满足开放性、实用性、基本功能需求，用面向对象程序设计方

法设计了个体、种群类，用C++语言设计了 Windows 下的计算平台。

通过两个应用实例——复杂机电传动控制系统的参数优化和足球机器人优化控制说明了平台的应用过程和单基因变异进化策略的实际应用。

第七章　总结、创新与展望

本文从对遗传算法的改进研究开始，通过一种“保留最优、调节中间、淘汰最差”的强化引导型遗传算法及其基因调节算子的研究，认识到交叉算子在进化算法中的可替代性，变异算子在进化算法中的主导作用，进而转向进化策略及其变异算子的研究。通过对进化策略中变异算子的变异方式、步长控制策略、局部和全局搜索能力、多种群技术的研究得出以下结论：

通过对交叉算子的作用机理分析、仿真，说明在进化算法中，交叉算子的局部和全局搜索功能可以由变异算子替代，在进化算法中变异算子起到了主导作用。

在进化策略中，当变异步长较大时，与全基因变异方式比较，单基因变异具有较大的成功概率，而且当变异步长继续增大时，全基因变异的成功概率可能趋于0，出现进化停顿，这也是导致全基因变异进化策略出现早熟收敛的一个主要原因。单基因变异比全基因变异的计算开销小，尤其对于高维优化问题，差别更为显著。

通过理论分析和横向仿真，证明使用Gauss分布的单基因变异时，保持变异成功率为0.3~0.48，具有较好的局部搜索性能。递减型的变异步长控制策略是一种比较合适的步长控制方法，通过理论分析和实例说明，单纯的递减型变异步长单基因变异算子全局搜索能力不足，通过引入均匀变异可以提高进化策略的全局搜索能力，Gauss变异与均匀变异相结合的$(\mu+\lambda+\kappa)-ES$既有良好的局部搜索能力，又有良好的全局搜索能力。

通过比较全基因变异算子与单基因变异算子的局部收敛速度、进化窗，说明在合适的变异步长时，全基因变异可以得到较快的局部收敛速度，但单基因变异对变异步长的适应范围较大。在没有有关变异步长的先验知识时，单基因变异使算法具有更强的鲁棒性。

遗传漂移是影响进化算法全局搜索能力的重要因素，引入多种群技术可以有效地克服遗传漂移产生的负面影响，保持种群多样性，获得求解多模态函数所有局部最优解的能力。在基于排挤和（/或）共享的多种群进化算法中，共享半径是一个关键而又难以确定的参数，它直接影响种群中个体的分类、同类极值点的判别，本文提出的山谷探索法，可以有效地克服这一困难，准确地区分同类与异类极值点。对于使用变异步长递减型的单基因变异算子，可以在给定

的精度下以一定的可信度判断子种群的收敛性。

本文的主要创新点：

1. 提出了基因调节算子，建立了同侧基因与异侧基因的概念，并以此分析交叉算子的作用机理，论述了交叉算子的可替代性和变异算子在进化算法中的主导作用。

2. 提出了单基因变异算子，从理论上分析和比较了单基因变异与现有全基因变异算子的成功概率，证明了单基因变异具有较大的成功概率，全基因变异可能出现进化停顿，建立了一种新的进化策略$(\mu+\lambda+\kappa)-ES$。

3. 提出了一种新的分析遗传算子性能的仿真试验方法——横向仿真技术，并用于分析比较单基因变异与全基因变异的进化效果和进化窗，证明了单基因变异对变异步长具有更好的适应性。

4. 建立了基于单基因变异的多种群进化策略$m\times(\mu+\lambda+\kappa)-ES$，提出了相应的子种群收敛判据，这一判据根据给定计算精度以一定可信度判断子种群是否收敛，使实际计算中收敛性判断有了定量依据。

5. 应用山谷探索法对个体、种群作同类判别和同峰与异峰极值点的判别，有效地避免了基于共享、排挤的多种群技术中，共享半径这一重要参数的很难确定问题。

进化策略中变异方式的研究及单基因变异方式的提出，是变异算子理论研究的重要补充，对于高维优化问题，可以大幅减小计算开销，提高收敛速度，具有重要的实用价值。本文通过理论分析、仿真试验与对比研究，较系统地研究了单基因变异算子的成功概率、步长控制策略、全局搜索能力、多种群技术。

本文提出的单基因变异算子并不排斥全基因变异算子的使用，如果能充分挖掘与问题相关的领域知识或者通过学习算法获得有关适应值函数的拓扑性质，使用全基因变异可以获得更高的搜索效率，所以，进化算法与数据挖掘方法、学习算法等智能技术相结合将是下一步研究的内容，此外，本文的主要研究的是“＋”号型算法，“，”号型算法是待研究的问题，在多种群技术中如何建立预警机制以避免重复搜索也是需要进一步研究的问题。

参考文献

[1] D.B. Fogel. Evolutionary computation: a new transaction. IEEE Trans. On Evolutionary Computation，1997，1(1):1~2

[2] 王正志，薄涛著. 进化计算. 长沙：国防科技大学出版社，2000 年

[3] T.Bäck，U.Hammel and H.P. Schwefel. Evolutionary computation: comments on the history and current state. IEEE Trans. On Evolutionary computation. 1997.1(1):3~17

[4] H.J.Bremermann. Optimization through evolution and recombination. Self~organizing Systems，M.C. Yovits，G.T.Jacobi，G.D.Goldstine，eds. Startan Books，Washington D.C. 1962:93~106

[5] J.W.Atmar. Speculation on the evolution of intelligence and its possible realization in machine learning. Doctoral dissertation，New Mexico State Univ.，Las Cruces，1976

[6] J.H. Holland. Concerning efficient adaptive systems. In M.C. Yovits, eds，Self~Organizing Systems，Spartan Books，Washington，D.C. 1962

[7] J.H. Holland，Adaptation in Natural and Artificial Systems. Cambridge，MA: MIT press，1975

[8] L.J. Fogel. On the organization of intellect. Doctoral Dissertation，University of California，Los Angeles，1964

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

[10] I. Rechenberg. Evolutionsstrategie: Optimierung texhnischer Systeme nach Prinzipien der biologischen Evolution. Stuttgart，Germany: Frogmann Holzboog，1973

[11] H.P. Schwefel. Evolutionsstrategie und numerische Optimierung. Ph. D Dissertation，Technische Universität Berlin，Germany，1975

[12] 谢金星. 进化计算简要综述. 控制与决策，1997，12（1）：1~7

[13] 孙吉贵，何雨果. 量子搜索算法. 软件学报，2003，14（3）：334~344

[14] 谢晓锋，张文俊，杨之廉. 微粒群算法综述. 控制与决策，2003，18（2）：129~134

[15] 李艳君. 拟生态系统算法及其在工业过程控制中的应用：[博士学位论文].

杭州：浙江大学，2001

[16] Dorigo M. Optimization， learning and natural algorithms:[Ph.D Thesis]. Dipartimento Di Electronica，Politecnico di Milano，Italy，1992

[17] 周昌乐编著. 心脑计算举要. 北京：清华大学出版社，2003

[18] T.Bäck and H.P. Schwefel. Evolutionary computation: an overview. In Proc. of the 1996 IEEE Int'l Conf. on Evolutionary Computation (ICEC1996) Nagoya Japan，IEEE Press，New York，NY，1996: 20~29

[19] D.Whitley. An overview of evolutionary algorithms: practical issues and common pitfalls. Information and Software Technology，43(2001):817~831

[20] Z. Michalewicz. Genetic Algorithms + Data Structure = Evolution Programs，Artificial Intelligence，Berlin: Springer，1992

[21] T. Bäck. Evolutionary Algorithms in Theory and Practice， New York: Oxford University Press，1996

[22] 王小平，曹立明著. 遗传算法——理论、应用与软件实现. 西安：西安交通大学出版社，2002

[23] 潘正君，康立山，陈毓平. 演化计算. 北京：清华大学出版社，1998

[24] C. Kappler， T.Bäck， J.Heistermann， etc. Refueling of nuclear power plant: comparison of a naïve and a specialized mutation operator. H.~M Voigt，W.Ebeling， I. Rochenberg eds. parallel problem solving from nature--PPSN IV， Int'l conf. Evolutionary computation，1996:829~838

[25] H.P. Schwefel. Evolutionsstrategie und numerische Optimierung. Ph. D Dissertation，Technische Universität Berlin，Germany，1975

[26] Kursawe， F. Towards self~adapting evolution strategies.Evolutionary Computation， 1995，IEEE International Conference，Volume: 1 ，29 Nov.~1 Dec. 1995:283~288

[27] Back，T.; Eiben，A.E. Generalizations of intermediate recombination in evolution strategies.Evolutionary Computation，1999. CEC 99. Proceedings of the 1999 Congress on，Volume: 2， 6~9 July 1999: 1566~1573

[28] L.Gruenz，H.G Beyer. Some observations on the interaction of recombination and self~adaptation in evolution strategies. Evolutionary Computation，1999. CEC 99. Proceedings of the 1999 Congress，Volume: 1：6~9

[29] Ming Chang; Ohkura，K.; Ueda，K.; Some experimental observations of (μ/μ，λ)~evolution strategies. Evolutionary Computation， 2001. Proceedings of the

2001 Congress，Volume: 1，2001:663 ~670

[30] Y. Matsumura，K.Ohkura，K.Ueda. Advantages of global discrete recombination in (μ/μ，λ，)~evolution strategies. Evolutionary Computation，2002. CEC '02. Proceedings of the 2002 Congress，Volume: 2，2002:1848~1853

[31] D.Goldberg. Genetic algorithms in search，optimization，and machine learning. Addison Wesley，MA，1989

[32] J.H. Holland. Hidden order: How adaptation builds complexity. Addison Wesley，MA，1995

[33] H.G. Beyer. Toward a theory of evolution strategies: on the benefits of sex—the $(\mu/\mu,\lambda)$ theory. Evolutionary computation, 1995，5(3):218~235

[34] H.G. Beyer. An alternative explanation for the manner in which genetic algorithms operates. Bio System，41(1997):1~15

[35] C.A.Magele，K.Preis，W.Renhart，etc. Higher order evolution strategies for the global optimization of electromagnetic devices. IEEE Trans. On Magnetics，1993，29(2):1775~1778

[36] C.Kappler. "Are evolutionary algorithms improved by large mutation?"，in Parallel Problem Solving from Nature (PPSN)IV (H.M Voigt，W.Ebeling，I.Rochenberg，and H.P.Schwefel eds.Vol.1141 of Lecture Notes in Computer Science，(Berlin)，Springer-Verlag，1996:346~355

[37] X.Yao and Y.Liu. "Fast evolution strategies"，Proc.4th，IEEE conf in Evolutionary Programming VI，P.J Angeline，R.Reynolds，J.McDonnell，and R.Eberthart，eds. Berlin，Germany Springer~Verlag，1997: 151~161

[38] 林丹，李敏强，寇纪淞. 进化规划和进化策略中变异算子的若干研究，天津大学学报，2000，33(5):627~630

[39] 王云诚，唐焕文. 单峰函数最优化问题的一个快速收敛进化策略，小型微型计算机系统. 2002，23（11）：1390~1392

[40] 刘若辰，杜海峰，焦李成. 基于柯西变异的免疫单克隆策略，西安电子科技大学学报，2004，31（4）：551~556

[41] 王战权，赵朝义. 进化策略中变异算子的改进研究，计算机仿真，1999，16（3）：8~11

[42] 王战权，赵朝义，夏云庆. 进化策略中基于柯西分布的变异算子改进探讨，系统工程，1999，17（4）：49~54

[43] Hashem， M.M.A.; Watanabe， M.; Izumi，K. Evolution strategy a new time~variant mutation for fine local tuning SICE '97. Proceedings of the 36th SICE Annual Conference. International Session Papers， 1997:1099 – 1104

[44] Sang Hwan Lee, Hyo Byung Jun， Kwee Bo Sim. Performance improvement of evolution strategies using reinforcement learning.Fuzzy Systems Conference Proceedings， 1999. IEEE '99. 1999 IEEE International，Volume: 2，22~25

[45] S.D.Müller， N.N. Schraudlph，P.D.Koumoutsakos. Step size adaptation in evolution strategies using reinforcement learning. Evolutionary Computation， 2002. CEC '02. Proceedings of the 2002 Congress， Volume: 1 , 2002:151 ~156

[46] L.Hildebrand ; B.Reusch, Fathi， M.; Directed mutation--a new self-adaptation for evolution strategies. Evolutionary Computation， 1999. CEC 99. Proceedings of the 1999 Congress on，Volume: 2，1999: 1550~1557

[47] Q. Zhou and Yanda Li. Directed variation in evolution strategies. IEEE trans. On Evolutionary computation，2003，7(4):356~366

[48] Yu Yongquan; Huang Ying; Zeng Bi; Chen Xianchu; Learning fuzzy model for nonlinear system using evolution strategies with adaptive direction mutation. Fuzzy Systems，2003. FUZZ '03. The 12th IEEE International Conference， Volume: 1，2003:237 ~ 241

[49] Yong Liang， Kwong Sak Leung. Two-way mutation evolution strategies. CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation， Volume: 1 ，2002 ：789 ~794

[50] N.Hansen，A.Gawelczyk; A.Ostermeier. Sizing the population with respect to the local progress in (1，λ)~evolution strategies~a theoretical analysis. Evolutionary Computation， 1995.， IEEE International Conference，Volume: 1 ,1995:80~85

[51] Tao Yuan Huang; Yung Yaw Chen. Parental population sizing in evolutionary strategies. Evolutionary Computation， 2001. Proceedings of the 2001 Congress，Volume: 2 , 2001:1351 —~1358

[52] L.Schonemann. The impact of population sizes and diversity on the adaptability of evolution strategies in dynamic environments. evolutionary Computation， CEC2004 Congress，Volume: 2, 2004:1270 ~1277

[53] 郭观七. 进化计算的遗传漂移分析与抑制技术：[博士学位论文]. 长沙：中南大学，2003

[54] K.Izumi， M.Hashem，M.A. Watanabe. An evolution strategy with competing subpopulations. Computational Intelligence in Robotics and Automation，1997 :306~311

[55] B.Porter, F.Xue. Niche evolution strategy for global optimization. Evolutionary Computation， 2001. Proceedings of the 2001 Congress， Volume: 2 , 2001: 1086 ~1092

[56] H.Fuger，G.Stein，U.Stilla. Multi~population evolution strategies for structural image analysis. Evolutionary Computation， 1994. IEEE World Congress on Computational Intelligence，Proceedings of the First IEEE Conference，1994: 229 ~234

[57] O.Cordon，F.Herrera. Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule--based system. Fuzzy sets and systems,118(2001):235~255

[58] J.Zhang， X.J.Yuan， Z.X. Zeng，etc. Niching in an ES/EP context. In Proc. Congress. Evolutionary computation，Vol.2，1999:1426~1433

[59] C.H. Kim， H.K. Jung， K.C. Choi. An algorithm for multimodal function optimization based on evolution strategy. IEEE Trans. On magnetics. 2004，40(2):1224~1227

[60] 徐宗本，陈志平，章祥荪. 遗传算法基础理论研究的新近发展. 数学进展，2000，2（29）：97－114

[61] A.E. Eiben，G..Rudolph. Theory of evolutionary algorithms: a bird's eye view. Theoretical Computer Science，229(1999)，3~9

[62] A.Auger. Convergence results for $(1,\lambda)-SA-ES$ using the theory of $\varphi-$irreducible Markov chains. Theoretical computer science. 2005，In press

[63] T. Bäck， H.P.Schwefel. An overview of evolutionary for parameters optimization. Evolutionary computation.1(1993):1~24

[64] H.G. Beyer. The theory of evolution strategies. Natural computing series，Springer， Heidelberg，2001

[65] H.P. Schwefel. Evolution and Optimum Seeking. New York: John Wiley，1995

[66] A.Bienvenüe， O.François. Global convergence for evolution strategies in spherical problems: some simple proofs and difficulties. Theoretical computer science. 306(2003): 269~289

[67] G. Rudolph. Convergence properties of evolutionary algorithms， Kovac，

Hamburh，1997

[68] G. Rudolph. Finite Markov chain results in evolutionary computation: a tour d'horizon. Fund. Inform. 35(1998):1~22

[69] G.Yin，G.Rodulph，H.P. Schwefel. Establishing connections between evolutionary algorithms and stochastic approximation. Ingormatica，1995，6(1) :93~116

[70] G.Yin，G.Rodulph，H.P.Schwefel. Analyzing evolution $(1+\lambda)$ strategy via stochastic approximation methods. Evolutionary computation. 1996，3(4) :473~489.

[71] H.G. Beyer. Toward a theory of evolution strategies: some asymptotical results from the $(1,+\lambda)-$theory. Evolutionary computation. 1993，1(2):165~188

[72] H.G. Beyer. The theory of evolution strategies. Natural computing series

[73] 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）：455~460

[74] 何琳，王科俊，李国斌等. 关于“遗传算法的全局收敛性和计算效率”一文的商榷. 控制理论与应用，2001，18（1）：142~145

[75] 李宏，唐焕文，郭崇慧. 一类进化策略的收敛性分析. 运筹学学报，1999，4（3）：79~83

[76] 郭崇慧，唐焕文. 演化策略的全局收敛性分析. 计算数学，2001，1（23）：105－110

[77] 郭观七，喻寿益. 遗传算法收敛性分析的统一方法. 控制理论与应用，2001，18（6）：443~446

[78] G. Rudolph. Global convergence and self~adaptation: a counter~example. In proceedings of the 1999 congress of evolutionary computation. Piscataway，NJ:IEEE press，1999，Vol.1:646~651

[79] G. Rudolph. Self~adaptive mutations may lead to premature convergence. IEEE. Trans. On Evolutionary Computation，2001，5(4):410~414

[80] H.G. Beyer，H.P. Schwefel. Evolution strategies – A comprehensive introduction. Natural computing 2002 (1):3~52

[81] 孙瑞祥，屈梁生. 遗传算法优化效率的定量评价. 自动化学报，2000，26（4）：552~556

[82] A.E. Eiben，R.Hinterding，Z. Michalewicz. Parameter control in Evolutionary Algorithms. IEEE. Trans. On Evolutionary computation. 1999，2(3):124~141

[83] D.V. Arnold， H.G. Beyer. Performance analysis of evolutionary optimization with cumulative step length adaptation，IEEE Trans. On Automatic Control，2004，49(4):617~622

[84] G. Rudolph，Local convergence rates of simple evolutionary algorithms with Cauchy mutations， IEEE Trans. On Evolutionary Computation，1997，4(1):249~258

[85] N.G. Kim,J.M.Won, J.S.Lee, etc. Local convergence rate of evolutionary algorithm with combined mutation operator. Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on Volume 1, 2002: 261~ 266

[86] 陈希孺著. 数理统计引论. 北京：科学出版社 1981

[87] H.A. David. Order statistics. New York：Wiley1981

[88] O.François. An evolutionary strategy for global minimization and its Markov chain analysis. IEEE Trans. on evolutionary computation，1998，2(3):77~90

[89] Jianbo Cai， G.Thierauf. Evolution strategies for solving discrete optimization problems. Advances in engineering software，25(1996):177~183

[90] Jianbo Cai，G.Thierauf. A parallel evolution strategy for solving discrete structural optimization. Advances in engineering software，27(1996):91~96

[91] G.Thierauf，Jianbo Cai. A parallel evolution strategy for solving discrete structural optimization. Advances in engineering software，4（1997）：318－324

[92] C.Ebenau，J.Rottschäfer，G. Thierauf. An advanced evolutionary strategy with an adaptive penalty function for mixed~discrete structural optimization. Advances in engineering software. 2005，36（1）：29－38

[93] L.Costa， P.Oliveira. An evolution strategy for multiobjective optimization. Evolutionary Computation， 2002. CEC '02. Proceedings of the 2002 Congress，Volume: 1，2002:97 – 102

[94] S.M. Yang，D.G. Shao，Y.J. Luo. A novel evolution strategy for multiobjective optimization problem. Applied mathematics and computation.2005，In press.

[95] Z.Michalewicz， D.Dasgupta， R.G. Le and M.Schoenauner. Evolutionary algorithms for constrained engineering problems. Computer Ind. Engng. 1996，30(4):851~870

[96] E.M. Montes and C.A.C.Coello. A simple multi~membered evolution strategy to solve constrained optimization problems. IEEE Trans. On Evolutionary computation.2005，9(1):1~17

[97] D.V.Arnold and H.G. Beyer. Investigation of the (μ, λ)~ES in the presence of noise. Evolutionary Computation, 2001. Proceedings of the 2001 Congress, Volume: 1, 2001:332 – 339

[98] R.Berlich, M.Kunze. Parallel evolutionary algorithms. Nuclear instruments & methods in physics research. 502(2003):467~470

[99] Guo Guanqi and Yu Shouyi. Evolutionary parallel local search for function optimization. IEEE trans. On System, man, cybernetics—Part B, 2003, 33(6):864~876

[100] V.Rupela, G.Dozier. Parallel and distributed evolutionary computations for multimodal function optimization World Automation Congress, 2002. Proceedings of the 5th Biannual Volume 13, 9~13 June 2002:307 – 312

[101] 陈国良，王熙发等. 遗传算法及其应用. 北京：人民邮电出版社，1996

[102] D.L. Hartl. A primer of population genetics. Sinauer Associates Inc., 1988

[103] J.D. Schaffer, R.A. Caruna, L.J.Eshelman etc. A study of control parameters affecting online performance of genetic algorithms for function optimization. In J.D. Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, 1989：51~60

[104] W. M. Spears. Crossover or mutation? In L. Darrell Whitley, editor, Foundations of Genetic Algorithms, 2, Morgan Kaufmann. 1993：221~237

[105] I.D.Falco, A.D.Coippa, E.Tarantino. Mutation--based genetic algorithm: performance evaluation. Applied Soft Computing, 1(2002):285~299

[106] Dai Xiaoming. Zou Runmin, Sun Rong etc. Convergence properties of non-crossover genetic algorithm. Proceedings of the 4th World Congress on Intelligent control and automation, Shanghai, China, 2002:1882~1826

[107] Falkenauer, E.; The worth of the uniform, Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, Volume: 1, 6~9 July 1999 : 776~782

[108] A. E. Eiben, M. Schoenauner. Evolutionary computing. Information processing letters.82(2002):1~6

[109] R.Salomon. The curse of high-dimensional search spaces observing premature convergence in unimodal functions, IEEE Trans. On Evolutionary computation.2005, 9(1):in press

[110] 王湘中，喻寿益，贺素良等.一种强引导进化型遗传算法. 控制与决策，2004，

19（7）：795－798

[111]王湘中，喻寿益. 基于人工育种策略的进化算法. 计算机科学，2002，29(10)，176~178

[112]D.Whitley. The GENITOR algorithm and selection pressure: why rank~based allocation reproduction trials is best. J.Schaffer eds. Proceeding of the 3rd Int'l conf. On genetic algorithms， Morgan Kaufmann Publishers， Los Altos，CA，1989:133~140

[113]Z.B. Xu，K.S. Leung，Y. Liang，etc. Efficiency speed~up strategies for evolutionary computation: fundamentals and fast~Gas. Applied mathematics and computation. 142(2003):341~388

[114]江瑞，罗予频，胡动成等.一种协调勘探和开采的遗传算法：收敛性及性能分析. 计算机学报，2001，24(12):1233~1241

[115]H.G. Beyer. The theory of evolution strategies. Natural computing series. Springer， Heidelberg，2001

[116]G. Rudolph. An evolutionary algorithm for integer programming， Int'l conf. 1994 in Parallel problem solving from nature (Y. David， R. Männer and H.P. Schwefel eds. Vol.3: 139~148

[117]N.Hansen， A.Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. Evolutionary Computation， 1996.， Proceedings of IEEE International Conference，1996 :312~317

[118]D.K. Gehlhaar，D.B. Fogel. Two new mutation operators for enhanced search and optimization in evolutionary programming. B.Biosacchi， J.Bezdek and D.B. Fogel， eds. Proceedings of SPIE.Piscataway，N.J. IEEE press，1996，3165，260~269

[119]K.Chellapilla. Combining mutation operators in evolutionary programming. IEEE trans. On Evolutionary Computation.1998，2(1):91~96

[120]T.Jason and I.Wegener. On the choice of the mutation probability for the (1+1)EA. M. Schoemauer eds. Parallel problem solving from nature，Springer，Heidelberg，2000，vol. 6: 89~98

[121]H.Mübhlenbein and T.Mahnig. FDA: A scalable evolutionary algorithm for the optimization of additively decomposed functions. Evolutionary computation，1999， 7(4):353~376

[122]M.Pelikan，D.Goldberg and E.Cantu~Paz. BOA: the Bayesian optimization algorithm. W.Banzhaf，J.Daida，A.Eiben etc. eds. GECCO~99:Proceedings of the genetic and evolutionary computation conference， Morgan Kaufmann，San Francisco，1999:525~532

[123]黄克中，毛善培编著. 随机方法与模糊数学应用.上海：同济大学出版社，1987 年

[124]William H. Press 等著. 傅祖芸等译.C 数值算法（第二版），北京：电子工业出版社，2004

[125]D.B.Fogel. Simulated evolution: a 30~year perspective. Signals， Systems and Computers， 1990. 1990 Conference Record Twenty~Fourth Asilomar Conference，Volume 2 : 5~7

[126]H.P. Schwefel. Numerical Optimization of Computer Models， Chichester: Wiley， 1981

[127]P.J.Angeline. Adaptive and self~adaptive evolutionary computation. In Computation intelligence: a dynamic system perspective. M. Palaniswami，Y.Attikiouzel etc. eds. New York: IEEE Press，1995:152~161

[128]R.Hinterding，Z.Michalewicz and A.E.Eiben. Adaptation in evolutionary computation. In Proc. 2nd IEEE Conf. Evolutionary computation. Piscataway，N.J: IEEE Press，1997:65~69

[129]J.Smith. Self-adaptation in evolutionary algorithm. Ph.D. dissertation， Univ. of the West of England,Bristol,1997

[130]J.Smith. Operator and parameter adaptation in genetic algorithms. Soft computing.1997，1(2):81~87

[131]彭宏，杨立洪，郑咸义等. 计算工程优化问题的进化策略. 华南理工大学学报，1997，25(12)，17~21

[132]Yiu Wing Leung and Yuping Wang. An orthogonal genetic algorithm with quanitization for global numerical optimization. IEEE Trans. On Evolutionary Computation，2001，1(5):41~53

[133]G. Rudolph. Convergence analysis of canonical genetic algorithms. IEEE Trans. On neural networks，1994，5(1):96~101.

[134]王湘中，喻寿益. 多模态函数优化的多种群进化策略. 控制与决策，（已录用）

[135]D.J. Cavicchio. Adaptive search using simulated evolution:[Doctoral

Dissertation]. Michigan :University of Michigan，Ann Arbor，1970

[136]K.A. DeJong. An analysis of the behavior of a class of genetic adaptive systems: [Doctoral Dissertation]. Michigan: University of Michigan，Ann Arbor，1975

[137]K. Deb and D.E. Goldberg， An investigation of niche and species formation in genetic algorithms， In Proc of the 3rd International Conference on Genetic Algorithms，San Mateo， CA: Morgan Kaufman，1989: 42~50

[138]Stadnyk， Schema recombination in a pattern recognition problem， In Proc of the Second International Conference on Genetic Algorithms， Hillsdale NJ: Lawrence Erlbaum Associates，1987: 27~35

[139]T.A. Sedbrook， H. Wright， and R. Wright， Application of genetic classifier for patient triage， In Proc of the 4th International Conference on Genetic Algorithms， San Mateo， CA: Morgan Kaufman， 1991: 334~338

[140]S.W. Mahfoud， Crowding and pre-selection revisited， In Parallel Problem Solving from Nature， 2， Amsterdam: Elsevier， 1992: 27~36

[141]S. W. Mahfoud. Niching Methods for genetic algorithms:[Doctoral Dissertation]. Illinois : University of Illinois，Urbana~Champaign，1995

[142]郭观七，喻寿益，贺素良. 自适应小生态遗传算法的理论分析和加速技术. 计算机学报，2003，26（6）：753~758

[143]O.J. Mengshoel and D.E. Goldberg， Probabilistic crowding: deterministic crowding with probabilistic replacement， Illi GAL Report No.99004，1999

[144]D.E. Goldberg and J. Richardson， Genetic algorithms with sharing for multimodal function optimization， In Proc of the 2nd International Conference on Genetic Algorithms， Hillsdale NJ: Lawrence Erlbaum，1987: 41~49

[145]B.L. Miller and M.J. Shaw，Genetic algorithms with dynamic niche sharing for multimodal function optimization， In Proc of 1996 IEEE International Conference on Evolutionary Computation，Nagoya，Japan，1996: 786~791

[146]A. Pétrowski， A clearing procedure as a niching method for genetic algorithms， In Proc of 1996 IEEE International Conference on Evolutionary Computation，Nagoya，Japan，1996: 798~803

[147]W. M. Spears， Simple subpopulation scheme， In Proc of the 3rd Conference on Evolutionary Programming， San Diego， CA， 1994: 296~307

[148]W. Spears ， Evolutionary Algorithms: The Role of Mutation and Recombination， Berlin: Springer Verlag， 2000: 28~34

[149]D.E. Goldberg and L. Wang, Adaptive niching via co-evolutionary sharing, In Genetic Algorithms in Engineering and Computer Science, John Wiley and Sons, Ltd. 1997: 21~38

[150]M. Jelasity. GAS, a concept on modeling species in genetic algorithms, Artificial Intelligence, Vol.99, 1998: 1~19

[151]于歆杰，王赞基. 适用于多峰函数优化的改进顺序生境遗传算法. 清华大学学报，2001，41（3）：17~20

[152]于歆杰，王赞基. 自适应调整峰半径的适应值共享遗传算法. 自动化学报，2002，28（5）：816~820

[153]李敏强，寇纪淞. 多模态函数优化的协同多群体遗传算法. 自动化学报，2002，28（4）：497~504

[154]黎明，杨小芹，周琳霞. 动态区域性多群体搜索的遗传算法(英文). 自动化学报，2003，29（3）：212~218

[155]J. Gan and K. Warwick, Dynamic niche clustering: a fuzzy variable radius niching technique for multimodal optimization in GAs, In Proc of 2001 IEEE International Conference on Evolutionary Computation, Piscataway, NJ: IEEE Press, 2001: 215~222

[156]X.Yin, N.Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods on multimodal function optimization. Proceedings of the Int'l Conf. On artificial neural networks and genetic algorithms. 1993:450~457

[157]C.Y. Lin, W.H.Wu. Niche identification techniques in multimodal genetic search with sharing scheme. Advances in engineering software, 33(2002):779~791

[158]R. Ursem. Multinational evolutionary algorithms, In Proc of 1999 IEEE International Conference on Evolutionary Computation, Piscataway, NJ: IEEE Press, 1999: 1633~1640

[159]D.E. Goldberg, K. Deb and B. Korb, Messy genetic algorithms revisited: studies in mixed size and scale, Complex System, 4, 1990: 415~444

[160]喻寿益，郭观七. 一种改进遗传算法搜索性能的小生镜技术. 信息与控制，30(6)，2001:526~530

[161]D. Beasley, D.R. Bull and R.R. Martin. A sequential niche technique for multimodal function optimization, Evolutionary Computation, 1(2), 1993: 101~125

[162]S. Mikami, M. Wada, and T.C Fogarty. Learning to achieve co-operation by

temporal~spatial fitness sharing， In Proc of IEEE International Conference on Evolutionary Computation，Piscataway，NJ: IEEE Service Center，1995: 803~807

[163]R.Lohmann. Structure evolution and incomplete induction. In: R.Matousek and B.Manderich eds. Parallel problem solving from nature，1992，Elsevier，Amsterdam:175~185

[164]I.Rochenberg. Evolutionsstrategie'94. Frommann~ Holzboog Verlag ，Stuttgart,1994

[165]X.Yao. Evolving artificial neural networks. Proceedings of the IEEE 87(9):1423~1447

[166]L.Grünz and H.G. Beyer. Some observations on the interaction of recombination and self~adaptation in evolution strategies. In: P. Angeline (ed.) Proceedings of the CEC'99 Conf. IEEE，Piscataway，NJ，1999:9639~645

[167]何劲生，郑浩然，方潜生等. 在进化策略中引入策略进化. 小型微型计算机系统，2003，24（9）：1715~1717

[168]Jeffrey Richter. Windows 高级编程指南. 北京：清华大学出版社，1999

[169]Jeff Prosise. MFC Windows 程序设计. 北京：清华大学出版社，2001

[170]刘国华，包宏，李文超. 用 MATLAB 实现遗传算法程序. 计算机应用研究，2001，80~82

[171]殷铭，张兴华，戴先中. 基于 MATLAB 的遗传算法实现. 计算机应用，2000（1）：9~11

[172]周济，罗应立，张建华等. 面向对象的遗传算法及其在神经网络辅助设计中的应用. 计算机工程与应用，2002，: 54~56

[173]潘孝勇，姜伟，杨继隆. Visual C 与 Matlab 的混合编程. 计算机仿真，2004，21（3）：140~143.

[174]李天昀，葛临东. 综述 MATLAB 与 VC 的交互编程. 计算机仿真，2004，21（9）：193~196

[175]曾宏坤，沈德耀. 基于多智能体的新型遗传算法及其在复杂系统中的应用研究. 信息与控制，2003，32（3）：277~280

[176]S. Russell， P.Norvig，Artificial Intelligence: A Modern Approach. 北京：人民邮电出版社，2002

[177]贺建军. 复杂机电系统机电耦合分析与解耦控制技术：[博士学位论文]. 长沙：中南大学，2003

[178]张春晖，侯祥林，徐心和. 足球机器人系统仿真中的数学模型. 东北大学学报，2001，22（5）：493~496

[179]王湘中，喻寿益. 基于改进进化策略的足球机器人最优控制. 哈尔滨工业大学学报，2005，37(7):969~970

附录1 作者在攻博期间发表的论文

1. 王湘中，喻寿益，贺素良。一种强引导进化型遗传算法，控制与决策，2004，19（7）：795~798（EI收录）
2. 王湘中，喻寿益，龙永红。机器人足球赛中队员角色的动态分配策略，哈尔滨工业大学学报，2004，36（7）：943~945（EI收录）
3. 王湘中，喻寿益。适用于高维优化问题的改进进化策略，控制理论与应用，（已录用）
4. 王湘中，喻寿益。多模态函数优化的多种群进化策略，控制与决策，（已录用）
5. 王湘中，喻寿益。一种基于人工育种策略的进化算法。计算机科学，2002，29（10）：176~178
6. 王湘中，喻寿益。进化策略中变异算子的仿真研究与改进，系统仿真学报，（已投稿）
7. 王湘中，喻寿益。Single-gene mutation based evolution strategies, IEEE Trans. On Evolutionary Computation,(已投稿)
8. 王湘中，喻寿益。基于改进进化策略的足球机器人最优控制，哈尔滨工业大学学报，2005，37（7）：969~970（EI收录）
9. 王湘中，谭敦强。微量CO对HCO带材性能的影响，湖南有色金属学报，2003
10. 王湘中，罗伟成。网络连通性问题的快速解法，计算机工程，2002，28（7）
11. 夏利锋，王湘中，喻寿益。π网络法石英晶片电参数计算机测量系统，中南大学学报，2004，35（1）：101~105（EI收录）
12. 王湘中，黎晓兰。电气装置的计算机测试诊断技术，湖南大学学报，2002，29（3）：58~61
13. 王湘中，肖阳伟。变电站电压无功综合控制的策略研究与实现，湖南大学学报，2002，29（3）：81~83
14. 贺素良，王湘中，喻寿益。遗传算法及模糊、神经网络如何技术的研究，计算机工程，2003，29（7）：17~19

附录 2 作者在攻博期间参加的科研项目

1. 主持学校资助的机器人足球比赛项目，设计的机器人足球 5Vs5 仿真比赛程序获“2003 年中国机器人足球大赛” 亚军和 2004 年第九届 FIRA 世界杯亚军
2. 主持完成机车压缩空气风源净化装置的研制（横向科研课题）
3. 主持完成学校资助的项目“复杂机电系统测试诊断平台的研制”
4. 参加国家自然科学基金资助的重点项目“复杂机电系统耦合与解耦理论与方法（项目号：59835170）
5. 参加国家自然科学基金资助的重点项目“智能仿生机电人工腿关键应用基础研究”（项目号：50275150）
6. 参加石英晶片参数测量与分拣系统的研制（横向科研课题）
7. 2003 年指导参加全国大学生电子设计制作大赛，获湖南赛区 3 等奖

致　　谢

本文是在导师喻寿益教授的精心指导下完成的，从论文选题、研究路线、研究方向到论文的写作无不凝聚着导师的心血，导师高尚的师德、敏锐的思维、渊博的学识和严谨的治学态度将使我终身受益。在读期间，导师和师母在生活、学习等多方面的给予了关怀，在此谨向导师和师母表示最诚挚的感谢！

我同样要感谢桂卫华教授、吴敏教授、阳春华教授、王随平教授、王雅琳博士的诸多关怀和指导，在此深表感谢！

和师兄郭观七博士多次非常有益的讨论、思想火花的碰撞都使我受益匪浅，和其他同门吴舒辞博士、贺素良博士、周旋博士、秦斌博士、邹恩博士、李祥飞博士在学术上的交流和探索，营造了良好的学术氛围，受益良多，在此深表感谢！

最后，我要感谢我的家人和亲友们，他们在我攻博期间给予多方面的关心、鼓励和支持。特别是我的妻子和儿子，是他们的大力支持和默默的奉献使我能全身心投入到学习和研究中，顺利完成学业。

王湘中 谨识

2005-8-28.