东北大学

硕士学位论文

进化策略及其在神经网络优化中的应用

姓名：李影

申请学位级别：硕士

专业：运筹学与控制论

指导教师：邢伟

20061201

进化策略及其在神经网络优化中的应用

摘 要

进化计算是借鉴生物自然选择和遗传机制而产生的一类随机搜索算法，主要包括遗传算法、进化规划、进化策略。人工神经网络是源于人脑神经系统的一类模型，是模拟人类智能的一条重要途径，具有模拟人的部分形象思维的能力。虽然进化计算和神经网络都是借鉴生物个体或生物界的某些行为特征和结构属性而发展起来的人工智能方法，但两者的侧重点有所不同。神经网络偏重于对生物个体学习智能的描述，进化计算则是对生物遗传进化的模拟。神经网络与进化计算相结合，将使神经网络同时具备学习和进化的特征，从而表现出更加完备的智能特性。

作者在论文期间的工作主要集中在以下几个方面：

介绍了进化策略的基本理论与技术。阐述了最初的进化策略，它的发展，以及进化策略经常采用的操作算子。

如何保持种群的多样性一直是进化算法的一个研究主题。基本的进化计算在实际应用中常常会遇到过早收敛或仅得到局部最优解的问题，这是因为一般的选择操作使得种群中的优秀个体大量繁殖后代，致使大量相似的个体在种群中占有极大的比例，导致种群缺乏多样性。为了克服进化计算存在的这一缺陷，文中对基本的进化算法作以改进，使劣质个体与优秀个体能够共同参与到进化过程中，从而使个体的有利信息能够传给后代，增强种群的多样性。将改进的进化策略用于求解有约束和无约束函数优化问题，实例仿真表明改进的进化策略具有很好的可行性和实用性。

人工神经网络是越来越得到广泛应用的新兴学科，但目前主要采用的 BP 算法极易陷入局部最优，使其的应用受到很大的限制。本文利用进化策略同时优化神经网络的结构与连接权，得到了很好的效果。

关键词：进化计算；进化策略；操作算子；神经网络；网络结构；连接权

Evolutionary Strategy and its Application in the Optimization of Neural Networks

Abstract

The evolutionary computation is a stochastic search algorithm which uses for reference to the biological natural selection and the heredity mechanism. It mainly includes genetic algorithms, evolutionary programming and evolutionary strategy. Artificial neural networks derived from human brain nervous system. It is an important way to simulate the human intelligence, and it has simulated human's part thinking in images ability. Although evolutionary computation and neural network are both the artificial intelligence method that simulate certain behavior characteristic and the structure attribute of the biological individual or biosphere, their emphasis point is different. Neural network stresses in describing the study intelligence of biological individual. Evolutionary computation is to simulate the biological heredity evolution. Therefore, combining neural network and evolutionary computation will make the neural network to have the study and evolution characteristic simultaneously, and display the more complete intelligent characteristic.

This thesis is focused on the following several aspects:

Introduce the basic theory and technology of the evolutionary strategy. Elaborate the initial evolutionary strategy and its development, as well as the operation operator that evolutionary strategy used frequently.

How to maintain the multiplicity of the population is always a researching subject about evolutionary algorithm. The basic evolutionary computation can meet premature or only obtain the local optimal solution in the practical application. The cause of this is the general select operator causing the outstanding individual massive reproduction descendant. So the massive similar individuals hold the enormous proportion in the population, cause the population to lack the multiplicity. In order to overcome this flaw, do some modification to the basic evolution algorithm. Both the inferior individual and the excellent individual can participate in the evolution process together. Thus enables the advantageous information of individual to generate to the descendants, enhances population's multiplicity. The simulation indicated the feasibility and the validity of the algorithm.

Artificial neural network is a novel subject, and its application is widely. But the BP

algorithm is extremely easy to fall into local extremum. This thesis used the evolutionary strategy to optimize the structure and the weights of the neural network simultaneously, obtained the very good effect.

Key words: evolutionary computation; evolutionary strategy; evolutionary operator; neural network; network architecture; weights

独创性声明

本人声明，所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他学位而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：李影

日　　期：2007年1月17日

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流。

（如作者和导师不同意网上交流，请在下方签名；否则视为同意。）

学位论文作者签名：　　导师签名：

签字日期：　　签字日期：

第一章 绪 论

1.1 生物进化进程

根据科学家的估算，在已诞生百余亿年的茫茫宇宙间，确知唯一有生命存在的只有我们共同生活的地球。数十亿年来，生命体由小到大，由简单到复杂，由低级到高级不断地进化，构成了种类繁多、千差万别的动植物群体。

在数十亿年的进化过程中，生物个体为了适应环境生存和繁衍更好的后代，在自身的形态、习性和能力等多方面不断地进行进化。例如，我国著名风景区峨眉山麓伏虎寺一带，生活着一种稀少而珍贵的枯叶蝶，它的翅长约7厘米，翅膀的颜色枯黄，翅膀的形状如叶片，叶脉清晰可见。当它栖息在树枝或树干上时，宛如一片飘零的枯叶。这样，枯叶蝶用这种拟态就可以躲避天敌而求得生存。这里值得注意且受到启发的是，这种拟态的本领不是某一个枯叶蝶生存期间所能学到并实现的，而是在千百万年进化的种群级上完成的。相对来说，那些模拟枯叶较逼真的蝴蝶有较长的生存期，繁殖具有同样特征后代的概率也就大些。

人类从生物的进化过程中得到了很多有益的启示，即：

(1) 进化是自然界“物竞天演，适者生存，不适者淘汰”的优胜劣汰，是一个巧妙的适应环境的优化过程；

(2) 这种优化不是由单个个体来实现的，而是一代又一代的种群同时进行。实质上它是一个大规模并行优化求解过程，在搜索高维求解空间中的合适解点时，大大降低了时间的复杂性；

(3) 生物在进化过程中，通过基因的重组和变异，能在更大的范围内生成品质更好、更能适应环境变化的后代，也就是搜索可能解的空间大大地扩大了；

(4) 生物进化并不是在一个因子众多、结构复杂的时变动态系统中追求某一个全局最优解，而只是在环境的约束条件下寻找一个更为满意的解的子空间。

鉴于以上原因，模拟生物群体进化过程的进化计算(Evolutionary Computation，简记为EC)产生了，并开始得到逐步地发展和应用。

19世纪以来，随着科学技术的进步和社会生产的不断发展，人们通过农牧业的杂交育种，给遗传学的产生和发展积累了丰富的经验和资料。奥地利植物学家 Mendel 系统

地概括和总结前人的经验，于1866年发表了著名的论文《植物杂种实验》，阐明了生物的遗传规律，是历史上公认的实验遗传学创始人。

19世纪中叶，著名的英国生物学家达尔文(Darwin)根据对世界各地区生物的考察和人工选择实验，在1859年发表了《物种起源》巨著，提出了“物竞天演，适者生存，不适者淘汰”的自然选择为基础的生物进化论学说。根据这种进化理论，生物的发展和进化主要有三种形态，即遗传、变异和选择。

此后Vries提出的突变论和Johannse提出的基因概念、纯系理论等，都给进化论的发展以有力的支持。本世纪初期，美国Morgen通过遗传学与细胞学的结合奠定了基因理论。1952年Hershey等通过实验，成功地证明DNA(Desoxyribonucleic Acid脱氧核糖核酸)是遗传物质。于是，遗传学由个体水平、细胞水平进入到以核酸、蛋白质等生物分子为研究对象的基因工程阶段。

1.2 进化计算的发展及特点

1.2.1 进化计算的发展

进化计算是自60年代开始发展起来的一门新兴学科。它是仿照生物进化过程，按优胜劣汰的自然选择优化规律和方法，来解决科技领域中难以用传统优化方法解决的优化计算问题。尽管迄今为止，已提出了大量的进化计算规则和方法，但是从基本上说，可以归纳为三大类：即遗传算法(Genetic Algorithm，简记为GA)[8]、进化规划(Evolutionary Programming，简记为EP)和进化策略(Evolutionary Strategies，简记为ES)。

进化计算的思想可追溯到20世纪50年代。遗传算法是由美国Michigan大学John Holland教授提出的；美国科学家Fogel等人提出了进化规划；德国科学家Rechenberg和Schwefel提出了进化策略。他们分别用不同的进化模式模拟生物进化的过程，从而形成了三种具有普通影响的模拟进化优化计算方法。

60年代初，Holland认识到生物的遗传和进化现象与人工自适应系统的相似性，提出在研究和设计人工自适应系统时，可以借鉴生物遗传的机制，以群体的方法进行自适应搜索。他的学生Bagley于1967年在其博士论文中首次提出了“遗传算法”一词。1975年，Holland出版了第一本系统论述遗传算法和人工自适应系统的专著《Adaptation in Natural and Artificial Systems》，提出了模式定理(Schema Theorem)，为遗传算法奠定了理论基础。

进化规划是60年代中期Fogel等人为有限状态机的演化而提出的一类进化算法。与

遗传算法的不同在于，进化规划注重父代与子代的表现行为而不是遗传细节。1966 年，Fogel 出版了《Artificial Intelligence Through Simulated Evolution》，系统阐述了进化规划的思想。但当时学术界对在人工智能领域采用进化规划表示怀疑，直到九十年代初才逐步得到学术界的重视。

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

目前，进化计算引起包括数学、物理、化学、生物学、计算机科学等领域的科学家的极大兴趣。进化计算的研究领域和内容十分广泛，如进化计算的设计与分析，进化计算的理论基础及其在各个领域的应用等。随着理论的深入研究和应用领域的不断拓展，进化计算必将取得更大的发展。

1.2.2 进化计算的特点

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

(1) 进化计算在搜索过程中使用的是基于目标函数值的评价信息，而不是传统方法主要采用的目标函数导数信息或待求解问题领域内知识。进化算法的这一特点使其成为具有良好普适性和可规模化的优化方法。

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

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

(4) 进化算法具有较强的鲁棒性，即在存在的情况下，对同一问题用进化算法多次求解得到的结果是相似的。进化算法的鲁棒性在大量的应用实例中得到了充分的验证。

1.3 进化计算的研究方向及应用

进化计算为求解复杂系统优化问题提供一个通用的框架，它不依赖于问题的具体领域，因此被广泛应用于很多领域之中。其主要应用有：

(1) 组合优化

组合优化是进化算法最基本最重要的研究和应用领域，复杂的组合优化问题通常带

有大量的局部极值，目标往往具有不可微、不连续、多维多目标、有约束、高度非线性等特征，因此，精确求解组合优化问题的全局最优解是不可能的。进化算法作为随机搜索方法已在组合优化中得到相当广泛的应用，并在解决某些典型的优化问题时显示良好的效果。特别是对有约束、多目标的优化，已经发展成为进化计算的一个分支。

(2) 函数优化

函数优化是进化算法的经典应用领域，也是对进化算法进行性能评价的常用算例。很多人构造出各种各样复杂的测试函数，有连续函数也有离散函数，有凸函数也有凹函数，有低维函数也有高维函数，有确定函数也有随机函数，有单峰函数也有多峰函数等，用这些几何特性各具特色的函数来评价进化算法的性能，更能反映算法的本质效果。而对于一些非线性、多模型、多目标的函数优化问题，用其它优化方法较难求解，而进化算法却可以方便地得到较好的结果。

(3) 人工生命

人工生命是用计算机、机械等人工媒体模拟或构造出的具有自然生物系统特有行为的人造系统。自组织能力和自学习能力是人工生命的两大主要特征。人工生命与进化算法有着密切的关系。基于进化算法的进化模型是研究人工生命现象的重要基础理论，虽然人工生命的研究尚处于启蒙阶段，但进化算法已在其进化模型、学习模型、行为模型、自组织模型等方面显示出初步的应用能力，并且必将得到更为深入的应用和发展。人工生命与进化算法相辅相成，进化算法为人工生命的研究提供一个有效的工具，人工生命的研究也必将促进进化算法的进一步发展。

(4) 图像处理

图像处理是计算机视觉中的一个重要研究领域。在图像处理过程中，如扫描、特征提取、图像分割等不可避免地会存在一些误差，从而影响图像的效果。如何使这些误差最小是使计算机视觉达到实用化的重要要求。进化算法在图像处理中的优化计算方面找到了用武之地，目前已在模式识别(包括汉字识别)、图像恢复、图像边缘特征提取等方面得到了应用。

(5) 数据挖掘

数据挖掘是近几年出现的数据库技术，它能够从大型数据库中提取隐含的、先前未知的、有潜在应用价值的知识和规则。许多数据挖掘问题可看成是搜索问题，数据库看作是搜索空间，挖掘算法看作是搜索策略。因此进化算法在数据库中进行搜索，对随机产生的一组规则进行进化，直到数据库能被该组规则覆盖，从而挖掘出隐含在数据库中的规则。Sunil 已成功地开发了一个基于遗传算法的数据挖掘工具，利用该工具对两个

飞机失事的真实数据库进行了数据挖掘实验，结果表明进化算法是进行数据挖掘的有效方法之一。

此外，进化算法在生产调度、并行分布处理、复杂系统分析、自适应控制、自动程序设计、演化硬件等领域都有一定的应用。由于进化算法是对自然进化的一个粗糙简化，其完整的数学基础有待深入研究，借以时日，随着它不断地完善其应用会更加广泛。

1.4 论文研究的问题及其意义

进化计算和神经网络都是借鉴生物个体或生物界的某些行为特征和结构属性而发展起来的人工智能方法。因此，将神经网络与进化计算相结合，不仅使神经网络同时具备学习和进化的特征，而且会表现出更加完备的智能特性。近年来，进化计算与神经网络日益融合，形成神经网络一个新的研究课题，并在实际应用中显示出以前神经网络所没有的优越性，足以说明进化计算与神经网络结合的有效性和必要性。

应用进化计算对网络的输入特征向量和学习样本进行进化选取，以获得最优的输入特征向量和样本；用神经网络对控制系统模拟建模，同时利用进化计算优化控制系统的参数；利用进化计算从神经网络中获取知识和提取规则等，它们都是神经网络和进化计算共同解决的问题。而进化计算与神经网络的另一个结合点就是应用进化计算对神经网络优化。由于对神经网络的优化一直都没有得到较好地解决，因此在进行网络结构设计和权值学习时需要一定的经验和反复试验，而且很难找到较优的网络结构和权值。在一定程度上制约着神经网络的发展和应用，而进化计算群体寻优的优越性，为神经网络的设计和实现提供了一条有效的途径。

目前，基于进化计算的神经网络设计和实现已成为神经网络领域一个新的研究热点，国内外许多学者已经做过大量的工作，初步取得一些显著的成就，从实践方面说明进化算法与神经网络结合的可行性。但总体上讲，进化神经网络处于初期发展阶段，理论方法有待于完善，应用研究有待于进一步深入。

1.5 作者的主要工作和论文的结构

作者在硕士学习期间，初步掌握最优化、进化计算、神经网络方面的基本理论和方法，为毕业论文的设计打下一定的基础。

作者在论文中的工作主要集中在：

(1) 进化策略；

(2) 对神经网络的权值和结构的优化学习；

论文可以分为两大部分：

第一部分介绍进化策略的理论和方法；

第二部分介绍神经网络的理论及利用进化策略对神经网络优化设计。这部分首先讲述了与论文相关的神经网络理论基础知识，随后具体的介绍使用进化计算设计神经网络的方法和步骤，通过实例对提出的方法进行验证。

由于进化计算是多个学科交叉的一门新兴的智能技术，尚处于发展阶段，还不是十分的成熟，特别是它的数学和生物学基础本身就比较薄弱，许多方面有待于进一步地研究和探讨。加上作者本人的知识上的局限性，难免有错漏之处，敬请批评指正。

第二章 进化策略

进化策略[4-7]是一类仿效自然界进化规则来解决函数优化问题的方法，它是60年代在德国发展起来的。早期的进化策略可以看作使用浮点数进行表达，仅使用变异作其遗传算子的演化程序，主要应用于各种连续可变参数的优化问题。近些年，它也逐渐被应用于多种不同的领域。

2.1 问题的表达

进化策略不同于遗传算法，它主要采用传统的十进制实数表达问题。为了与突变操作相适应，进化策略主要采用两种表达方式。

(1) 二元表达方式

这种表达方式中个体由目标变量 X 和标准差 σ 两部分组成，每部分又可以有 n 个分量，即：

$$(X,\sigma)=((x_1,x_2,\cdots,x_i,\cdots x_n),(\sigma_1,\sigma_2,\cdots,\sigma_i,\cdots\sigma_n))$$

X 和 σ 之间的关系是：

$$\begin{cases}\sigma_i'=\sigma_i\cdot\exp(\tau'\cdot N(0,1)+\tau\cdot N_i(0,1))\\ x_i'=x_i+\sigma_i'\cdot N_i(0,1)\end{cases} \tag{2.1}$$

式中：(x_i,σ_i)—父代个体的第 i 个分量；

(x_i',σ_i')—子代新个体的第 i 个分量；

$N(0,1)$—服从标准正态分布的随机数；

$N_i(0,1)$—针对第 i 分量重新产生的一个符合标准正态分布的随机数；

τ'—全局系数，常取 1；

τ—局部系数，常取 1。

上式表明：新个体是在旧个体基础上随机变化而来的。

二元表达方式简单易行，应用十分广泛。

(2) 三元表达方式

为了改善进化策略的收敛速度，Schwefel 在二元表达的基础上引入第三个因子——坐标旋转角度 α。个体的描述扩展为 (X,σ,α)，即：

$$(X,\sigma,\alpha)=((x_1,x_2,\cdots,x_i,\cdots x_n),(\sigma_1,\sigma_2,\cdots,\sigma_i,\cdots\sigma_n),(\alpha_1,\alpha_2,\cdots,\alpha_j,\cdots\alpha_m))$$

三者的关系为：

$$\begin{cases}\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha_j' = \alpha_j + \beta \cdot N_j(0,1) \\ x_i' = x_i + z_j\end{cases} \tag{2.2}$$

式中：α_j—父代个体i分量与j分量间坐标的旋转角度；

α_j'—子代个体i分量与j分量间坐标的旋转角度；

β—系数，常取0.0873；

z_j—取决于σ'和α'的正态分布随机数。

其余符号同式(2.1)。

旋转角度α_j表面上是分量i与j间坐标的旋转角度，实质上它是分量i与分量j之间协方差的体现。

随机数z_j服从正态分布，其数学期望为零，其方差σ_a^2取决于突变方差σ'^2及旋转角度α，即：

$$\sigma_a = (\prod_{i=1}^{n-1} \prod_{j=i+1}^{n} R(r_{ij})) \cdot \sigma' \tag{2.3}$$

式中矩阵$R(r_{ij})$各元素按下式计算：

$$\begin{aligned} r_{ii} &= r_{jj} = \cos\alpha_{ij} \\ r_{ij} &= -r_{ji} = -\sin\alpha_{ij} \end{aligned} \tag{2.4}$$

因此，随机数z_i可表示为$N(0,\sigma_a)$。

2.2 进化策略的分类

进化策略分为最初的(1+1)-ES和逐渐发展的$(\mu+1)$-ES、$(\mu+\lambda)$-ES及(μ,λ)-ES。其中(1+1)-ES与$(\mu+1)$-ES为单子代进化策略，$(\mu+\lambda)$-ES及(μ,λ)-ES为多子代进化策略。

2.2.1 单子代进化策略

1963年，德国柏林技术大学的Rechenberg和Schwefel为了研究风洞中的流体力学问题，提出进化策略。当时提出的这种优化方法只有一个个体，并由此衍生同样仅有一个的下一代新个体，故称为(1+1)-ES。

进化策略中的个体用传统的十进制实数型数表示，即：

$$X^{t+1} = X^t + N(0,\sigma) \tag{2.5}$$

式中：X^t—第t代个体的数值；

$N(0,\sigma)$—服从正态分布的随机数，其均值为零，标准差为σ。

因此，进化策略中的个体含有两个变量，为二元组(X,σ)。新个体的X^{t+1}是在旧个体X^t的基础上添加一个独立的随机变量$N(0,\sigma)$。假若新个体的适应度优于旧个体，则用新个体代替旧个体；否则，舍弃性能欠佳的新个体，重新产生下一代新个体。在进化策略中，个体这种进化方式称为突变。很明显，突变产生的新个体与旧个体的差异不大，这符合生物进化的基本状况：生物的微小变化多于急剧变化。

下面用具体实例说明(1+1)-ES的执行过程。设目标函数为：

$$\max f(x_1,x_2)=21.5+x_1\sin(4\pi x_1)+x_2\sin(20\pi x_2)$$

约束条件为：

$$-3.0\le x_1\le 12.1$$

$$4.1\le x_2\le 5.8$$

在此优化问题中，$X=(x_1,x_2)$，令$\sigma=(\sigma_1,\sigma_2)$，即：

$$x_1^{t+1}=x_1^t+N_1(0,\sigma_1)$$

$$x_2^{t+1}=x_2^t+N_2(0,\sigma_2)$$

假设在第t代，有：

$$(X,\sigma)=((5.3,4.9),(1.0,1.0))$$

则突变后的新个体为：

$$x_1^{t+1}=x_1^t+N_1(0,1.0)=5.3+0.4=5.7$$

$$x_2^{t+1}=x_2^t+N_2(0,1.0)=4.9-0.3=4.6$$

式中$N(0,1.0)$产生两个服从正态分布的随机数0.4及−0.3。由于

$$f^{(t)}=f(5.3,4.9)=18.383705$$

$$f^{(t+1)}=f(5.7,4.6)=24.849532>f^{(t)}$$

因此x_1^{t+1}及x_2^{t+1}被接纳，用新个体X^{t+1}替换旧的个体X^t。

为了控制收敛速度，Rechenberg还提出了著名的“1/5成功定律”。他用φ表示突变次数中成功突变(新个体被采纳)的比率。他认为φ应为1/5，若φ大于1/5，适当加大σ；反之，减少σ，即：

$$\sigma^{t+1}=\begin{cases}C_d\cdot\sigma^t & \varphi<1/5\\ \sigma^t & \varphi=1/5\\ C_i\cdot\sigma^t & \varphi>1/5\end{cases}\qquad(2.6)$$

式中：φ—经历k次迭代后突变成功的次数与总突变次数之比，通常$k>10$；

C_d—小于 1 的系数；

C_i—大于 1 的系数。

(1+1)－ES 仅仅使用一个个体，进化操作只有突变一种，亦即用独立的随机变量修正旧个体，以求提高个体素质。显然，这是最简单的进化策略。

早期的(1+1)－ES，没有体现群体的作用，只有单个个体在进化，具有明显的局限性。随后，Rechenberg 又提出了$(\mu+1)$－ES，在这种进化策略中，父代有μ个个体$(\mu>1)$，并且引入重组(Recombination)算子，使父代个体组合出新的个体。在执行重组时，从μ个父代个体中用随机的方法任选两个个体：

$$(X^1,\sigma^1)=((x_1^1,x_2^1,\cdots,x_n^1),(\sigma_1^1,\sigma_2^1,\cdots,\sigma_n^1))$$

$$(X^2,\sigma^2)=((x_1^2,x_2^2,\cdots,x_n^2),(\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2))$$

然后从这两个共同中组合出如下新个体：

$$(X,\sigma)=((x_1^{q_1},x_2^{q_2},\cdots,x_n^{q_n}),(\sigma_1^{q_1},\sigma_2^{q_2},\cdots,\sigma_n^{q_n}))$$

式中q_i=1 或 2，它以相同的概率针对$i=1,2,\cdots,n$随机选取。

对重组产生的新个体执行突变操作，突变方式及σ的调整与(1+1)－ES 相同。将突变后产生的个体与父代的μ个个体相比较，若优于父代最差的个体，则代替后者成为下一代μ个个体新成员；否则，重新执行重组和突变产生另一新个体。

$(\mu+1)$－ES 和(1+1)－ES 具有相同的策略：只产生一个新个体。$(\mu+1)$－ES 的特点在于：

(1) 采用群体，其中包括μ个个体；

(2) 增添重组算子，它相当于遗传算法的交换算子，从父代继承信息构成新个体。

显然，$(\mu+1)$－ES 比(1+1)－ES 有了明显的改进，为进化策略这种新的进化算法奠定了良好的基础。

2.2.2 多子代进化策略

1975 年，Schwefel 首先提出$(\mu+\lambda)$－ES，随后又提出了(μ,λ)－ES。这两种进化策略都是用含有μ个个体的父代群体，通过重组和突变产生λ个新个体。它们的差别仅仅在于下一代群体的组成上。$(\mu+\lambda)$－ES 是在原有μ个个体及新产生的λ个新个体中(共$\mu+\lambda$个个体)再择优选择μ个个体作为下一代群体。(μ,λ)－ES 则是只在新产生的λ个新个体中择优选择μ个个体作为下一代群体，这时要求$\lambda>\mu$。总之，在选择了代新个体时，若需要有父代个体的参与，则使用“＋”记号，如(1+1)、$(\mu+1)$及$(\mu+\lambda)$；否则，改用逗号分隔，如(μ,λ)。

这两种进化策略中，都采用突变、选择两种算子，也可使用重组算子。其中重组算

子类似于$(\mu+1)-ES$，而突变算子有了新的发展，标准差σ既不是固定的常数，也不是按1/5成功规则的确定型变化，而是可自适应地调整，即：

$$\begin{cases}\sigma' = \sigma \cdot e^{N(0,\Delta\sigma)} \\ X' = X + N(0,\sigma')\end{cases} \tag{2.7}$$

式中：(X,σ)—父代个体；

(X',σ')—子代新个体；

$\Delta\sigma$—参数；

$N(0,\sigma')$—独立的服从正态分布的随机变量，其均值为0，标准差为σ'。

近年来，$(\mu,\lambda)-ES$得到广泛的应用，这是由于这种进化策略使每个个体的寿命只有一代，更新进化很快，特别适合于目标函数有噪声干扰或优化程度明显受迭代次数影响的课题。

2.3 进化策略的基本技术

2.3.1 初始种群的产生

进化策略中初始群体由μ个个体组成，每个个体(X,σ,α)内又可以包含n个x_i、σ_i分量及$n(n-1)/2$个α_j分量。产生初始个体的方法是随机生成。为了便于和传统的方法比较，可以从某一初始点$(X(0),\sigma(0),\alpha(0))$出发，通过多次突变产生$\mu$个初始个体，该初始点从可行域中用随机方法选取。

初始个体的标准差$\sigma(0)$，可按下式计算：

$$\sigma(0) = \frac{\Delta X}{\sqrt{n}} \tag{2.8}$$

式中：ΔX—初始点与最优点的距离；

n—个体中所含分量的个数。

由于ΔX在初始时不便确定，可取$\sigma(0)=3.0$。$\sigma(0)$不宜取的太大，若$\sigma(0)$太大而且μ也大时，选择力度不够，易使群体过于分散。尽管$\sigma(0)$较小，但在进化过程中通过个体的自适应调整仍可使搜索点很快散步到整个可行域内。

2.3.2 适应度函数的计算

适应度是衡量个体优劣的尺度。寻优问题可归结为求目标函数的极小值或极大值问题，用进化算法寻优，可将目标函数进行变换，得到相应的f_i，但必须满足两个条件：

(1) 变换要保证$f_i \geq 0$；

(2) 目标函数的优化方向，对应于适应度增大的方向。

由于进化策略采用实数表达问题，因此适应度的计算更加直观、简便。相对于遗传算法和进化规划，进化策略中的适应度计算更易于执行。

下面介绍一种适应度函数的变换方法：

(1) 极小值问题

若设目标函数为 $z(x)$，则可设

$$f=\begin{cases}c_{max}-z(x) & z(x)<c_{max}\\ 0 & \text{其它}\end{cases}$$

c_{max} 或是输入的参数，或是理论上 $z(x)$ 的最大值，但一般理论上 $z(x)$ 的最大值是未知的，通常，对于要求解的问题，可取迭代过程中出现 $z(x)$ 的最大值，这样 c_{max} 将随迭代次数而有所变化。

(2) 极大值问题

目标函数为 $z(x)$，则可设：

$$f=\begin{cases}c_{max}+z(x) & z(x)+c_{max}>0\\ 0 & \text{其它}\end{cases}$$

c_{max} 或是输入的参数，或是迭代过程中出现的 $z(x)$ 的最小值，也是可变化的。

由于适应度是指导搜索的关键，为了避免早期收敛对适应度函数应作适当的调整，这里介绍两种调整方法。

(1) 线性调整

$$f'=af+b$$

式中：f、f'—分别为调整前、调整后个体的适应度；

a、b—常系数。

调整前后的关系应为：

$$\begin{cases}\overline{f'}=\overline{f}\\ f'_{max}=c\overline{f}\end{cases}$$

式中：c 在1.2～2之间选取。

(2) 乘幂调整

$$f_i'=f_i^k$$

式中：k 为常数，$k=1.005$ 较适宜。

进化策略中对于约束条件的处理，主要是采用重复试凑法。每当新个体生成，将其代入约束条件中检验是否满足约束条件。若满足，则接纳新个体；否则，舍弃该新个体，

借助重组、突变再产生另一个新个体。由于进化策略采用实数编码，这种检验比较直观和简单易行。

应该指出，当采用$(\mu,\lambda)-ES$时，旧群体不参加选择，因此初始个体可以略去适应度计算，直接执行重组、突变等操作，然后再计算新群体的适应度。而对于$(\mu+\lambda)-ES$，旧群体参与选择，则需要计算初始群体的适应度。

2.3.3 进化算子

(1) 变异算子

与遗传算法不同，变异算子是进化策略的主算子。在基本进化策略中，变异是通过给目标参数加上一个服从正态分布的随机数来实现的。

进化策略目标参数和步长参数的变异公式如下：

$$\begin{cases}\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ x_i' = x_i + \sigma_i' \cdot N_i(0,1)\end{cases} \tag{2.9}$$

式中n为个体中所含分量数目，系数τ'，τ按 Schwefel 建议，可按下式计算：

$$\begin{aligned}\tau &= (\sqrt{2\sqrt{n}})^{-1} \\ \tau' &= (\sqrt{2n})^{-1}\end{aligned} \tag{2.10}$$

这一特别的变异机制使得进化策略本身的策略参数，在搜索的过程中，开发利用了适宜的内在模式和好的适应值之间的一个隐含的关联，由此所导致的根据适应值曲面的拓扑特性而使策略参数得到进化的机制，被称为自适应机制。而这一变异方式也就是我们常常使用的高斯变异。高斯变异是对变异个体的附近区域进行重点搜索的一种变异算子。

Yao X 和 Liu Y 在高斯变异的基础上提出一种新的变异算子，即柯西变异算子。

一维柯西密度函数集中在原点附近，其定义为：

$$f_t(x) = \frac{1}{\pi}\cdot\frac{t}{t^2+x^2} \quad , \quad -\infty < x < +\infty$$

式中$t>0$为比例参数，相应的分布函数定义为：

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(\frac{x}{t})$$

密度函数$f_t(x)$类似于高斯密度函数，其差异主要表现在：柯西分布在垂直方向略小于高斯分布，而柯西分布在水平方向上越接近水平轴，变得越缓慢，因此柯西分布可以看作是无限的。

柯西变异按下式进行：

$$\begin{cases}\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot C(0,1) + \tau \cdot C_i(0,1)) \\ x_i' = x_i + \sigma_i' \cdot C_i(0,1)\end{cases} \tag{2.11}$$

由此可见，柯西变异算子就是用柯西变异替换原有进化策略中的高斯变异。根据柯西分布和高斯分布的相似性和柯西分布具有较高的两翼概率特性，柯西分布容易产生一个远离原点的随机数，它比高斯变异产生的随机数分布的范围大，如果用柯西变异替换原来进化策略中的高斯变异来产生后代，这就意味着柯西变异有可能很快跳出极小的区域，但柯西分布较小的中央部分却是它的弱点。

尽管高斯变异与柯西变异均存在一定的不足，但目前为止，进化策略算法还是主要依赖这两种变异操作。

(2) 重组算子

随着进化策略的逐渐发展，进化操作的算子不再依赖于唯一的变异操作，重组算子也被慢慢的引入到其中。进化策略中的重组算子相当于遗传算法的交叉操作，它们都是以两个父代个体为基础进行信息交换。进化策略中的重组方式主要有三种：

(a) 离散重组　先随即选择两个父代个体

$$\begin{aligned}(X^1,\sigma^1) &= ((x_1^1,x_2^1,\cdots,x_n^1),(\sigma_1^1,\sigma_2^1,\cdots,\sigma_n^1)) \\ (X^2,\sigma^2) &= ((x_1^2,x_2^2,\cdots,x_n^2),(\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2))\end{aligned} \tag{2.12}$$

然后将其分量进行随机交换，构成子代新个体的各个分量，从而得出如下新个体：

$$(X,\sigma) = ((x_1^{q_1},x_2^{q_2},\cdots,x_n^{q_n}),(\sigma_1^{q_1},\sigma_2^{q_2},\cdots,\sigma_n^{q_n}))$$

式中q_i=1 或 2，新个体的分量是从两个父代个体随机选取，而且x_i分量的q_i不一定要等于σ_i分量的q_i。

(b) 中值重组　这种重组方式也是先随机选择两个父代个体如式(2.12)，然后将父代个体各分量的平均值作为子代新个体的分量，构成新个体为：

$$\begin{aligned}(X,\sigma) = (&((x_1^1+x_1^2)/2,(x_2^1+x_2^2)/2,\cdots,(x_n^1+x_n^2)/2), \\ &((\sigma_1^1+\sigma_1^2)/2,(\sigma_2^1+\sigma_2^2)/2,\cdots,(\sigma_n^1+\sigma_n^2)/2))\end{aligned}$$

这时，新个体的各个分量兼容两个父代个体的信息，而在离散重组中则只含有某一个父代个体的因子。

(c) 混杂重组　这种重组方式的特点在于父代个体的选择上。混杂重组时先随机选择一个固定的父代个体，然后针对子代个体每个分量再从父代种群中随机选择第二个父代个体。也就是说，第二个父代个体是经常变化的。至于父代两个个体的组合方式，既可以采用离散方式，也可以采用中值方式，甚至可以把中值重组中的 1/2 改为[0,1]之间的任一权值。

将上述三组方法排列组合，进化策略的重组方式有以下各种：

$$x_i'=\begin{cases}x_{S,i} & \text{无重组}\\ x_{S,i}\text{或}x_{T,i} & \text{离散型}\\ x_{S,i}\text{或}x_{T_i,i} & \text{混杂离散型}\\ x_{S,i}+(x_{T,i}-x_{S,i})/2 & \text{中值型}\\ x_{S,i}+(x_{T_i,i}-x_{S,i})/2 & \text{混杂中值型}\\ x_{S,i}+\eta(x_{T,i}-x_{S,i}) & \text{广义中值型}\\ x_{S,i}+\eta(x_{T_i,i}-x_{S,i}) & \text{广义混杂中值型}\end{cases} \tag{2.13}$$

式中：x_i'—新个体目标变量 X 的第 i 个分量；

$x_{S,i}$—固定的父代个体的目标变量 X 的第 i 个分量；

$x_{T,i}$—离散重组时另一固定父代个体的目标变量 X 的第 i 个分量；

$x_{T_i,i}$—随机选择的另一个父代个体的第 i 个分量；

η—权值，在[0,1]区间内。

上式为简单起见，只列举 X 因子。事实上对 σ、α 因子都可以采用上述 7 种重组方式，这样进化策略的重组算子便有 $7^3=343$ 种。

研究表明，进化策略采用重组后，明显增加算法的收敛速度。Schwefel 建议，对于目标变量 X 宜采用离散重组，对于策略因子 σ 及 α 宜用中值重组或混杂中值重组。

(3) 选择算子

进化策略的选择方式是完全确定的。$(\mu,\lambda)-$ES 是从 λ 个子代个体中选择 μ 个最好的个体作为下一代的父代；$(\mu+\lambda)-$ES 是从 λ 个子代个体和 μ 个父代个体中选择 μ 个最好的个体作为下一代的父代，即精英选择方法，从而保证了性能的改进是单调的，但是该选择方法无法适应变动的环境，不利于实施策略参数的自适应机制。因此，$(\mu,\lambda)-$ES 在今天更受到推崇。一般认为，比率 $\mu/\lambda\approx 1/7$ 是最优的。

除了上述基于进化策略的基本选择机制外，进化策略开始采用与其它选择方法相结合的选择机制，比如在选择较优个体时融入 $q-$竞争策略，在 $(\mu,\lambda)-$ES 的选择过程中加入精英选择策略。

$q-$竞争策略是源于进化规划的。假设在进化规划中父代有 μ 个个体，经过突变操作又产生 μ 个新个体，利用 $q-$竞争选择，要从 2μ 个个体中选出 μ 个子代个体。为此，将 2μ 个个体按适应度大小顺序排列，其中优胜者置于前面，即：

$$x_1,x_2,\cdots,x_\mu,x_{\mu+1},\cdots,x_{2\mu}$$

且

$$f(x_i)\geq f(x_{i+1})$$

式中：x_i—第i个个体；

$f(x_i)$—第i个个体的适应度函数值。

为确定某一个个体i的优劣，从2μ个个体组成的种群中任选q个个体的适应度进行比较，记录个体i优于或等于测试群体中各个个体的次数，此数即为个体的得分w_i，即：

$$w_i = \sum_{j=1}^{q} \begin{cases} 1 & f_i \geq f_j \\ 0 & \text{其它} \end{cases} \tag{2.14}$$

最后，根据个体得分选择分值高的μ个个体组成下一代群体。q-竞争具有的特点是：种群中的最优个体，由于它的得分总是最高的，因此最优个体有很大的概率被保存下来，劣质个体也有存活的机会。

精英保留选择机制[4]的思想是把种群中适应度最高的个体不进行配对交叉而直接复制到下一代中。此种选择操作又称复制，此方法一般与其他选择方法结合使用。

2.4 进化策略算法的一般流程

进化策略的基本流程如下：

Begin

{

给定初始参数；

随机生成u个个体的初始化群体$P(0)$；

计算个体的适应度；

while(终止条件不满足)

do

{

$t=t+1$；

对$P(t-1)$中的每个个体实施进化操作，平均产生λ/μ个子代个体，整个群体共产生λ个个体，构成群体$P'(t)$；

计算$P'(t)$中个体的适应值，根据选择策略，选择最好的个体，构成下一代种群$P(t)$；

}

}

End

2.5 进化策略与其它进化算法的对比

遗传算法、进化规划、进化策略都体现生物进化过程的“物竞天择、优胜劣汰”这一自然选择机制，从随机产生的初始解出发，经过进化择优，直至最后产生最优解。三种算法的实质相同，均借鉴生物进化和自然遗传的思想，通过群体中个体间相互竞争和信息交换进化出适应度高的个体，且都以种群作为搜索的起点，而不是单一的个体，搜索在解空间中同时向多个方向进行，所以在很大程度上是相似的。但由于它们模拟不同的生物进化机制，因此在实施每种方法时也有很大的差异。

(1) 编码方式

遗传算法主要采用二进制编码方式，简单易行，常用来处理离散型问题。而随着研究发现，二进制编码要不断进行编码，译码操作，增大的工作的复杂性，浪费大量的时间，且它的运算精度相对也较低。所以，很多学者在研究遗传算法时，也逐渐开始采用十进制编码，但编码中没有引进控制变异步长的控制因子。而进化规划与进化策略最初就采用十进制编码，不仅提高了算法的精度，而且加快搜索进程，常用来处理连续的优化计算课题。

(2) 重组算子

重组算子又称交叉算子。遗传算法和进化策略中都可以采用重组算子，但进化规划中不采用重组算子。

(3) 变异算子

遗传算法采用随机方式对二进制串变化，变异步长不受控制因子的控制，突变是遗传算法的次要算子。进化规划与进化策略都是在个体的基础上，通过控制因子产生一个正态分布的随机数形成新个体。突变是进化规划和进化策略的主要手段。

(4) 选择算子

遗传算法和进化规划都是从父代种群中进行随机性选择，优良个体尽可能被选中，并且允许少数劣质个体入选；进化策略用适应度排序的方法只允许优良个体进入新群体，劣质个体不被选中，进化策略也可以从子代群体或父代群体中进行选择。

通过对三个算法的对比分析，可以充分利用各个算法的优点，相互借鉴不断完善。在问题的表达方式上，遗传算法从原来的二进制编码方式扩充到实数表达问题。而进化策略重组算子的使用是对遗传算法的借鉴。进化策略中参数的自适应调整的特点在遗传算法中有所体现，如群体规模及交叉变异算子的概率不再是常数，而是可以动态的调整。实数编码遗传算法中的交叉和变异算子在某些方面都借鉴进化策略的思想。

对于进化算法中操作算子变异和交叉的重要性，一直是进化算法研究领域有争议的问题之一。进化规划和进化策略都强调变异算子的重要性，并将它作为主要的遗传算子，近年的研究证实变异算子非常有效。而遗传算法中交叉是主要的操作算子，变异是辅助算子。从更高的层次来看，变异用于群体中产生随机多样性，而交叉相当于一个加速器，由部分加速构成整体行为。原来的问题就转化为多样性和构造的相对重要性。对于遗传算法，就是探测和开发之间的均衡问题。多样性和构造的相对重要性是解答遗传算法与进化规划之间差异的关键。对于构造，交叉算子更为有效；而对于多样性，变异比交叉有效。出于生物学的局限性，交叉和变异在生物的进化中所起的作用并不十分清楚。交叉和变异都不应该轻易的舍弃，它们在搜索过程中所起的作用不同，因此，需要具体问题具体分析。

尽管进化算法能解决很多问题，但是还存在着某些不足。首先，算法的适应性评估函数是预先指定固定不变的，而在生物系统中个体的适应性应该是局部的，是它在与环境作生存斗争自然形成的并随着环境不断变化的，因此算法的选择机制充其量不过是人工选择，而非自然选择。其次，算法中的个体主要以竞争交互为主，而没有考虑个体之间的协作关系，更接近生物系统的模拟应该是竞争和合作并存的协同演化；算法中所有个体的结构都是形式预先定义好的字符串或树状结构，因此不能实现无穷无尽的演化，系统所能发生的一切都在设计者的掌握之中；最后，算法中的复制和交叉过于简单，复制只是一个精确的拷贝，而杂交则是取被杂交个体的一部分拼凑出下一代。当然，随着对生物遗传进化的进一步的认识，有望设计出逼近真实生物系统的算法。

综上所述，遗传算法、进化规划、进化策略在利用生物进化机制提高计算机求解问题能力的目标和基本思路上是一致的，但在具体的做法上则是有很大差别的，甚至对于一些基本问题有着根本观点上的分歧。

2.6 本章小结

进化策略是一种模拟生物进化机制的算法，主要用来求解数值优化问题。本章主要介绍了进化策略的基本原理、进化策略种类的发展及其基本技术，给出了进化计算求解优化问题的基本流程，最后给出进化策略与遗传算法、进化规划之间的对比。

第三章 求解优化问题的进化策略及其仿真

基本进化计算在实际应用中常常会遇到过早收敛或仅得到局部极值点的问题，这是因为一般的选择操作使得种群中的优秀个体大量繁殖后代，致使大量相似的个体在种群中占有极大的比例，导致种群缺乏多样性。但这些占种群大多数的个体并不一定是全局最优的，进化早期的有些劣质个体也有可能发展为更好的解，然而，基本的进化计算不利于保护这样的个体，导致算法过早收敛到一个局部最优解。

为了克服进化计算存在的这一缺陷，许多学者纷纷提出不同的进化算法，美国密歇州立大学遗传算法研究应用学会提出了一种新的算法模型——分等级公平竞争模型，戴春妮、姚萌等提出了种群竞争消亡算法，主要都是利用小生境的思想，但是一旦劣质个体构成的种群被完全淘汰，所有的劣质个体仍然没有机会参与到进化中去。本文对基本的进化算法作以改进，使劣质个体与优秀个体在选择算子的作用下，能够共同参与到进化过程中，使个体的有利信息能够传给后代。本章主要针对用进化策略解决函数优化问题，提出一种新的改进算法。

3.1 一种求解有约束优化问题的进化策略

3.1.1 问题描述

需进行优化的带约束问题一般可表述如下：

$$\min f(X);\quad X=(x_1,x_2,\cdots x_n)\in R^n$$
$$st.\quad g_k(X)\le 0,\ k=1,2,\cdots,p$$
$$h_k(X)=0,\ k=1,2,\cdots,q$$

其中 $f:R^n\to R^1$，$g:R^n\to R^p$，$h:R^n\to R^q$，目标函数定义的搜索空间为 $S\subseteq R^n$，$(p+q)$ 个约束条件定义的可行域为 $F\subseteq S$。$\forall X$，如果 $X\in F$，那么称 X 为一个可行解，否则，称 X 为一个非可行解。求解该类问题的关键是，在可行域内找到一个 X^*，使其满足

$$f(X^*)\le f(X)$$

3.1.2 改进的进化策略

近年来，利用进化算法解决带约束优化问题得到了学术界的极大关注，随之，不同

求解方法也相继出现。由于约束优化问题的解分为：可行解与非可行解，用进化算法求解该问题时，如何处理可行个体与非可行个体之间的关系将会直接影响最后的结果。

基本进化计算在实际应用中常使可行个体大量繁殖，但这些可行个体并不一定是全局最优的，进化早期的某些非可行个体也很有可能发展为更好的解，然而进化计算不利于保护这样的个体，算法可能由于淘汰所有的非可行个体而过早收敛到一个局部最优解。为克服进化计算存在的这一缺陷，对基本的进化算法作以改进。

(1) 新的选择算子

在调整可行个体与非可行个体之间关系上前人曾做了大量的工作，许多学者曾对可行个体和非可行个体采用不同的评判标准，本文从如果存在非可行个体，进化过程中便有非可行个体参与这一思想出发，对原有进化算法的选择算子给以改进。采用进化算法解有约束优化问题时，很直观地，可行个体被认为比非可行个体有较好的适应值，有更大的繁衍机会，所有的约束条件对于可行解都已经满足，剩下的任务就是在其中寻找一个 X^*，使其满足 $f(X^*)$ 为最小。但是，这种观点忽略了很重要的一点，进化算法是一种概率方法，在某些进化过程中，非可行个体可能带有比可行个体更好的信息，而且，一个非可行个体可能跳出非可行域而很快达到最优点。

传统的选择方法却使那些非可行个体几乎没有机会参与到以后的进化中去，这使得非可行个体的一些优良特性不能在以后的后代中得到继续传播。文献[11]将种群分为多个小的群体，使个体在各自能生存的环境中继续存活下去，最后在这些种群中进行竞争，淘汰劣势群体，保留优良群体，但这种做法也可能使劣质群体全军覆没。这里对每代中的群体，根据个体的可行性分成两组[12]，可行个体组和非可行个体组。对这两组个体独立的进行评判和排列，然后分别选择可行组和非可行组中的较优个体一起作为父代个体。父代可行个体和非可行个体的数目根据调整参数 S_p 进行调整。这样确保一定数量的非可行个体能够参与到进化中，避免可行个体独自控制进化进程。

(2) 父代个体的评判与选取

利用改进的进化算法求解多约束优化问题，应注意以下两个方面：一是如何对可行个体和非可行个体进行评价和排列；二是如何确定可行父代和非可行父代的数目。对于第一方面，可以对可行组和非可行组采用不同的评判和排列策略。对于可行组，目标函数可直接作为适应度函数，对于非可行组，可采用任何一种先前用于评价和排列整个种群的方法，例如：作为整个种群适应度函数的动态惩罚法。对于第二方面，通常情况下作为父代的个体数和种群个体数是预先给定的常数，如(20,100)－ES指的是父代个体数为20，种群中个体的总数为100，由于新算法中两组个体的评价和排列是相互独立的，我们需要决定父代个体中可行个体和不可行个体的数目。因此，提出了一种父代选择策

略，这里对文献[12]的策略进行改进，父代的个体数与可行个体总数存在着由一个正数 S_p 决定的比例关系，可行与非可行父代个体数通过下面的规则给出：

① $FeaPar = \text{round}(FeaInd / S_p)$ (3.1)

if $FeaPar \geq Par$

$$FeaPar = \text{round}((0.75 + t/(4T)) \cdot Par) \tag{3.2}$$

② $InFeaPar = Par - FeaPar$ (3.3)

$FeaPar$ 表示作为父代的可行个体数，$FeaInd$ 表示可行个体的数目，Par 表示父代个体的总数，$InFeaPar$ 表示作为父代的非可行个体数。(3.2)式不仅保证了父代的可行个体数不能超过父代总的个体数，且保证了进化过程中如果存在非可行个体，则进化过程中会始终有非可行个体参与。进化后期非可行个体数量会不断减少，若计算出父代非可行个体数目大于非可行个体的总数，那么，不足的部分用余下的可行个体补充。一旦父代的可行个体数目和非可行个体数确定，我们就可以从各自的组中选择较好的个体进行繁殖。

下面具体解释一下这一选择策略，以(20,100)－ES为例，如果取 $S_p = 3$，$FeaInd = 15$，我们得到 $FeaPar = 5$，那么，$InFeaPar = 20 - 5 = 15$。因此，我们从可行组中选择5个较好的个体，从非可行组中选择15个较好的个体。若取 $S_p = 4$，$FeaInd = 92$，设最大进化代数为500，第1代我们通过(3.1)式得到 $FeaPar = 23$，超过了作为父代个体的最大数目20，则按(3.2)式计算 $FeaPar = 15$，那么求得父代个体中非可行个体数为5。S_p 的选取要根据实际而定，根据一定的经验，S_p 过小，非可行个体参加繁衍的机会变小，而 S_p 过大时，参与繁衍的可行个体会明显减少。通常，种群规模比较大时，S_p 会选的大些，否则会小些。

(3) 变异算子

进化策略主要采用的操作算子是变异算子，所以选择合适的变异算子也是相当重要的。最初，多数学者采用如下的变异算子[13-14]对个体进行修正：

$$X' = X + N(0,1) \tag{3.4}$$

$N(0,1)$ 为标准正态分布，式中对每个个体的所有分量均采用了同一个随机量。本文采用如下的变换策略：

$$X'_{ij} = X_{ij} + \gamma \cdot N_j(0,1) \tag{3.5}$$

其中 $N_j(0,1)$ 为与个体 X 的第 j 分量相关的服从正态分布的随机量，$\gamma = (1 - t/T)^b$，b 为控制参数，这样的修正方法使变异产生的偏移量随着进化代数的增长从大范围逐渐到小范围。

3.1.3 算法的基本流程

图3.1给出了上述算法的基本流程：

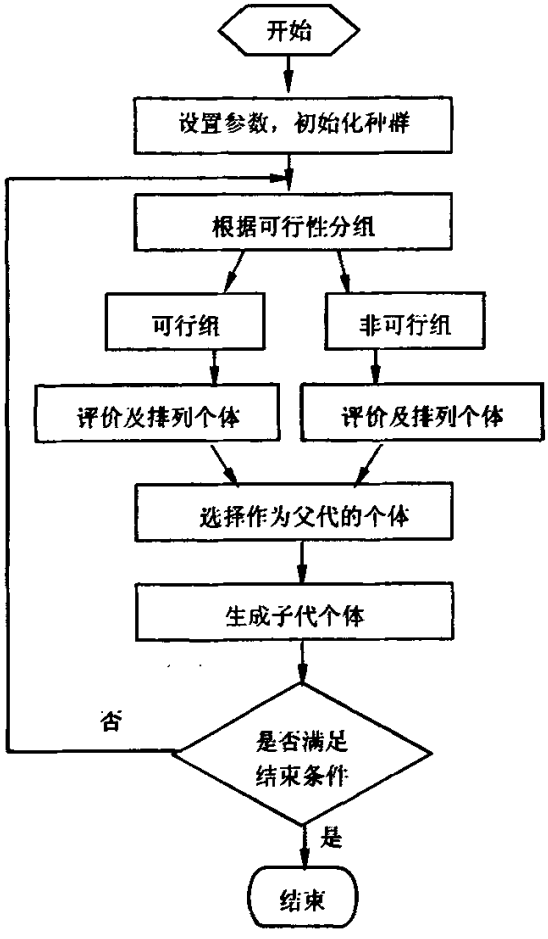

图3.1 算法的流程图

Fig. 3.1 Flow chart of algorithm

3.1.4 实例仿真

采用下面的函数测试该算法的性能。该函数选自文献[12]。

$$\max \quad f(X)=\sin^3(2\pi x_1)\sin(2\pi x_2)/(x_1^3(x_1+x_2))$$

$$s.t. \quad g_1(X)=x_1^2-x_2+1\le 0$$

$$g_2(X)=1-x_1+(x_2-4)^2\le 0$$

$$-10\le x_1,x_2\le 10$$

图3.2为该测试函数在可行域附近的局部图像，从图像中可以看出在这部分区域里，函数是多峰的，这样的函数能够有效地检测算法跳出局部最优的能力。

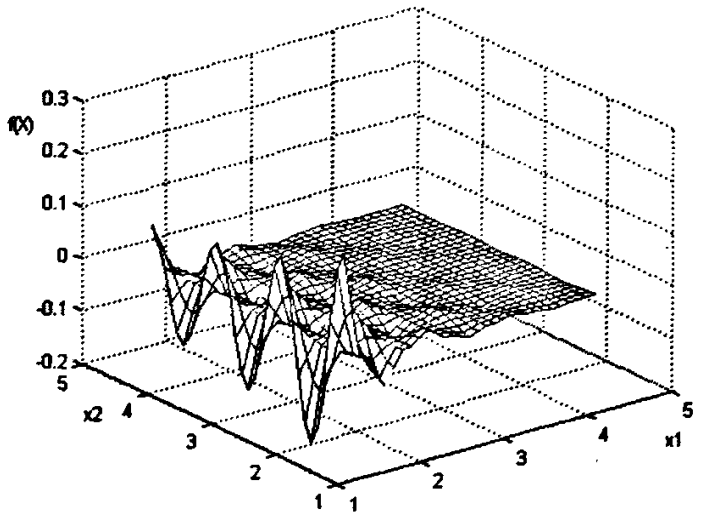

图3.2 局部函数图像

Fig. 3.2 Local image of function

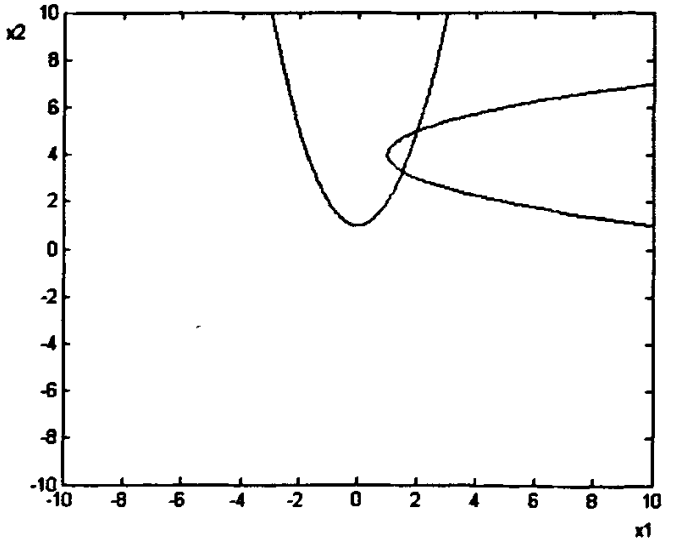

图3.3 问题的可行域

Fig. 3.3 Feasible area of problem

图3.3给出问题的可行域，图中两约束函数图像相交的部分。相对于整个搜索空间来说可行域是极其小的一部分。由于可行域基本实际可求，为了加快搜索进度，初始化种群时使一小部分个体在可行域内产生，本文取种群的1/10。

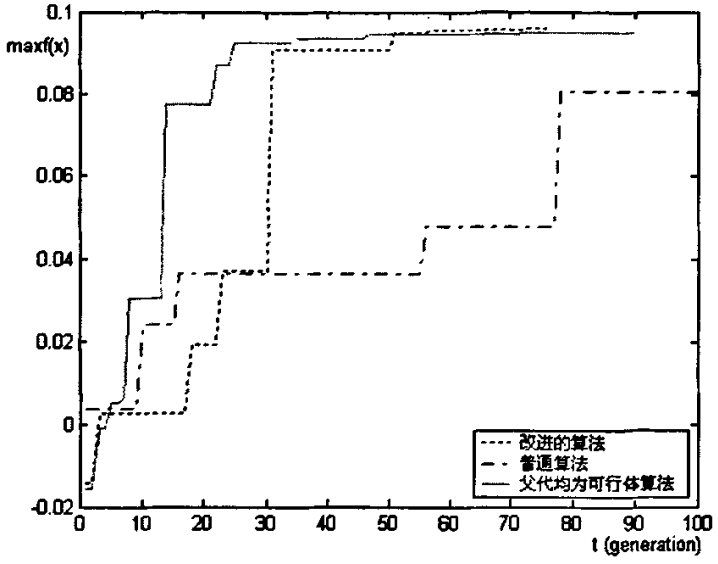

图 3.4 三种算法的结果比较

Fig. 3.4 Compare between results of three algorithms

图3.4给出了该函数基于改进进化策略求得的最优值随进化代数变化的曲线与普通进化策略及父代均为可行体的进化策略求得的曲线图的比较。这里选用(10,20)－ES，变异规则采用式(3.5)的方式，随着进化代数的增长，变异由大范围到小范围，参数$b=3$，$S_p=2$，搜索精度为10^{-6}，当进化过程中某代的结果满足该精度，记录当前进化代数，进化进程结束。通过多次计算，收敛代数基本在75代左右，比原来的算法要早15代左右。在种群规模不大且可行域较小的情况下，这样的收敛效果是不错的。

文中根据可行性将种群分为两部分，使得非可行个体参与到进化中来，对基本进化策略的父代选择策略和变异策略的改进，通过函数实例的测试，说明改进的算法在收敛速度和精度上有一定的优越性。

3.2 求解无约束优化问题的进化策略

这里所解决的无约束问题指可以只含有定义域而没有其它的约束条件。对于无约束优化问题，不再存在可行域，问题的解也不再区分为可行解与非可行解，所以若想将种

群进行分组进化只能寻找一个新的分组标准。一般的，种群在进化的每一代中，个体的适应度是大小不一的，但可以找到一个均值——平均适应度，这里取其作为分组的标准。适应度大于该值的个体称为优秀个体，否则，称为劣质个体，据此，将种群分为优秀个体组和劣质个体组。

3.2.1 算法的改进

父代优秀个体和劣质个体的数目同样根据调整参数S_p进行调整。这样保证一定数量的劣质个体能够有效地参与到进化中去，避免优秀个体独自控制进化进程。

(1) 父代个体的选取

利用改进的算法求解无约束优化问题时，同样应注意如何确定优秀父代和劣质父代个体的数目。优秀父代个体与劣质父代个体数通过与前一节相同的规则给出。只是分组不再依赖于可行性，而是以每代的平均适应度将种群分为优秀个体组和劣质个体组。

(2) 变异算子

对于无约束问题采用如下的变换策略：

$$x'_{ij} = x_{ij} + \gamma \cdot N_j(0,1) \tag{3.6}$$

其中，$N_j(0,1)$为一个与个体X的第j个分量相关的服从标准正态分布的随机量，γ为一个值小于等于1的变量，这里：

$$\gamma = \frac{\min(x_{max} - x_{ij}, x_{ij} - x_{min})}{(x_{max} - x_{min})}$$

其值随着个体的不同而改变。

3.2.2 算法的基本流程

进化策略是一种随机的搜索方法，下面给出该改进算法的基本流程：

Step1：设置参数，初始化种群个体$X(i)$，$i = 1,2,\cdots\mu$；

Step2：评价种群中个体的适应度$f(i)$及种群的平均适应度f_{ave}；

Step3：根据当代种群的平均适应度将种群分为两组。如果$f(i) \geq fave$，则$X(i)$属于优秀个体组，否则$X(i)$属于劣质个体组；

Step4：根据父代选择策略选择作为父代的个体，然后按式(3.6)的方式每个个体生成λ/μ个个体；

Step5：判断是否满足结束条件(最大进化代数或事先给定的计算精度)，满足则停止进化，否则转Step2。

3.2.3 实例仿真

本文采用下面两个较为典型的多峰函数[13]，通过对这两个函数求极值来测试算法的可行性及有效性。

函数1：　$F(x,y)=100(x-y^2)^2+(1-y^2)$

函数2：　$F(x,y)=0.5-\dfrac{\sin^2(x^2+y^2)-0.5}{(1+0.01(x^2+y^2))^2}$

其中函数1侧重于测试算法收敛速度，而函数2则侧重于测试算法克服早熟现象的能力。函数的自变量取值范围、优化模式及相应的全局极值如表3.1所示。

表3.1 函数的优化模式

Table 3.1 Optimized mode of function

函数	函数 1	函数 2
自变量取值范围	[−2.048，2.048]	[−2.048，2.048]
优化模式	极大值	极大值
极值	3905.926227	1.0

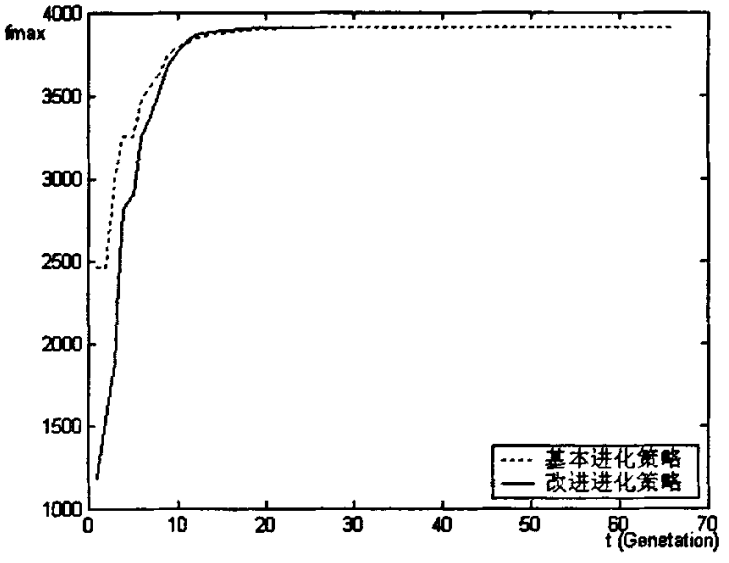

图3.5 函数1的最优值随进化代数变化的曲线

Fig.3.5 Curve of value changing with generation of function 1

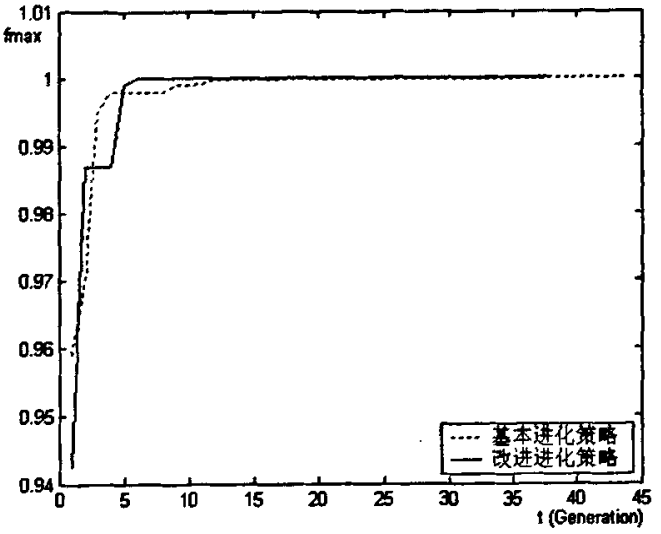

图3.6 函数2的最优值随进化代数变化的曲线

Fig.3.6 Curve of value changing with generation of function 2

图3.5、图3.6分别给出上述两个函数利用改进的进化策略与基本进化策略求得的最优值随进化代数变化曲线的比较。这里选用的是(10,20)-ES，搜索空间设为[-2.048, 2.048]，变异规则采用式(3.6)的方式，参数$S_p=2$，最大进化代数设为100代，搜索精度为10^{-6}，在进化过程中，当最优解与理论极值的差值小于给定的搜索精度时，认为已经找到最优解。跟踪得到全局最优解时的收敛代数，记录此时的进化代数，以此作为衡量算法收敛速度的指标。由于进化计算具有随机性，所以通过多次的计算，在种群规模不大的情况下，得到这样的收敛速度和收敛效果是不错的。

3.3 本章小结

本章主要是考虑在使用进化策略解决实际问题的过程中，基本进化策略可能使劣质个体过早的淘汰，从而它们所携带的有利信息不能很好的留给后代，从这一角度出发，对原有的进化策略作以改进，使得在进化过程中，优良个体与劣质个体都能够参与到进化中。第一部分提出一种求解带约束问题的进化策略算法，第二部分将算法推广到求解无约束优化问题。文中给出了算法的主要操作算子以及算法的实现步骤。通过对具体算例的仿真实验，表明新算法在搜索精度与收敛速度上具有很好的优越性。

第四章 基于进化策略的神经网络优化

4.1 前馈神经网络的结构和学习算法

神经网络系统理论是近年来人工智能的一个前沿研究领域，与基本符号机制的 Von Neumann 计算机理论完全不同，神经网络是基于连接机制的大规模并行处理和分布式的信息存储。它依靠大量神经元的连接以及由连接所引起的神经元的不同兴奋状态来表现系统的总体行为。与 Von Neumann 计算机相比，更加接近人脑的信息处理模式。从 20 世纪 40 年代，由 McCulleh 和 Ptts 提出的 MP 神经元模型以来，人工神经网络的研究几经沧桑。直到 80 年代，由于新的神经计算模型和学习算法的提出，再度掀起神经网络的研究和应用热潮。然而，对于一个具体的应用如何设计神经网络没有理论上的指导。目前各种神经网络应用系统的设计主要是根据设计者的经验和反复试验来进行的。因此设计工作的效率很低，而且不能保证设计出的网络结构和连接权等参数的组合是最优的，造成资源的大量浪费和网络性能的低下，极大地限制神经网络的应用和发展。

80 年代中期，Rumelhart 和 McClelland 提出多层前馈网络(Multilayer Feedforward Neural Networks)的反向传播(Back Propagation)学习算法，简称 BP 算法，是一种有导师的学习，它是梯度下降法在多层前馈网络中的应用。许多学者在不同时期从不同角度探讨过 BP 算法和采用该算法的前馈型神经网络(简称 BP 网络)。日本的 Amari 早在 1967 年即用梯度下降法训练含单个隐单元的 BP 网络，Bryson 和 Ho 于 1969 年也提出一种非常类似于误差逆向传播的算法用于非线性自适应控制，Werbos1974 年在他的博士论文中也独立地提出了误差逆向传播算法和它的几种变形。1986 年 PDP(Parallel Distributed Processing)研究小组的科学家 Rumelhart、Hinton 和 Williams 对 BP 算法给出了详细的数学推导，对其能力和潜力进行广泛深入的探讨，系统地解决了多层前馈网络的连接权学习调整问题，为推动人工神经网络技术的发展做出重要贡献。

典型的多层前馈神经网络模型一般采取三层结构，即网络输入层、隐含层和输出层。各层之间采用全连接，同一层中的节点没有连接。显然，它只是一个数学模型而不是生理模型，其拓扑结构如图 4.1 所示。学习能力是人工神经系统的一个重要特征，学习机制和学习算法始终是受到极大关注和深入研究的课题，通过学习算法使网络自动调整权值以达到某种期望的目的。误差反向传播算法(BP 算法)其基本工作原理是将学习样本对

加在网络的输入层，神经元的激活值从输入层经过隐含层向输出层传播，在输出层的各神经元获得网络的输入响应。然后将网络的期望输出和实际输出的差从输出层向输入层逆向传播，同时调整各连接权值。下面讨论 BP 学习算法的具体过程。

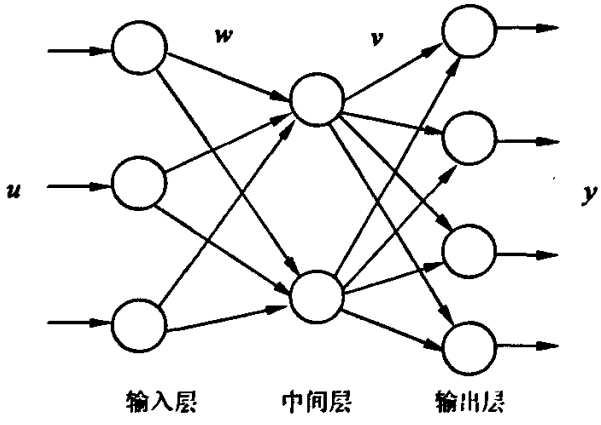

图 4.1 多层前馈网络

Fig.4.1 Three layers feedforward network

BP 学习算法的步骤：

(1) 设置初始权值 W(0)为较小的随机非零值。

(2) 给定输入/输出样本对，计算网络的输出。

设第 p 组样本输入、输出分别为：

$$u_p=(u_{1p},u_{2p},\ldots u_{np})$$

$$d_p=(d_{1p},d_{2p},\ldots d_{kp}),\quad p=1,2,\cdots,L$$

节点 i 在第 p 组样本输入时，输出为：

$$y_{ip}(t)=f[x_{ip}(t)]=f[\sum_j w_{ij}(t)I_{jp}] \tag{4.1}$$

式中：I_{jp}—在第 p 组样本输入时，节点 i 的第 j 个输入；

$f(\cdot)$—取可微分的 S 型作用函数。如：

$$f(x)=\frac{1}{1+e^{-x}} \tag{4.2}$$

$$f(x)=\frac{1-e^{-x}}{1+e^{-x}} \tag{4.3}$$

可由输入层经隐含层至输出层，求得网络输出层节点的输出。

(3) 计算网络的目标函数 J：

设E_p为在第p组样本输入时网络的目标函数，取L_2范数，则

$$E_p(t)=\frac{1}{2}\|d_p-y_p(t)\|_2^2=\frac{1}{2}\sum_k[d_{kp}-y_{kp}(t)]^2=\frac{1}{2}\sum_k e_{kp}^2(t) \tag{4.4}$$

式中$y_{kp}(t)$—在第p组样本输入时，经t次权值调整网络的输出，k是输出层第k个节点。

网络的总目标函数为：

$$J(t)=\sum_p E_p(t) \tag{4.5}$$

作为对网络学习状况的评价。

(4) 判别：

若

$$J(t)\le\varepsilon \tag{4.6}$$

则算法结束；否则，至步骤(5)。

式中ε—预先确定的，$\varepsilon>0$。

(5) 反向传播算法：

由输出层，依据J按“梯度下降法”反向计算，逐层调整权值。

若取步长为常数值，可得到神经元j到神经元i的连接权$t+1$次调整算式：

$$w_{ij}(t+1)=w_{ij}(t)-\eta\frac{\partial J(t)}{\partial w_{ij}(t)}=w_{ij}(t)-\eta\sum_p\frac{\partial E_p(t)}{\partial w_{ij}(t)}=w_{ij}(t)+\Delta w_{ij}(t) \tag{4.7}$$

式中：η—步长，在此称学习算子。

具体算法如下：

$$\frac{\partial E_p}{\partial w_{ij}}=\frac{\partial E_p}{\partial x_{ip}}\cdot\frac{\partial x_{ip}}{\partial w_{ij}} \tag{4.8}$$

设

$$\delta_{ip}=\frac{\partial E_p}{\partial x_{ip}} \tag{4.9}$$

式中：δ_{ip}—第i个节点的状态x_{ip}对E_p的灵敏度(第p组样本输入时)。

由式(4.8)、(4.9)可得

$$\frac{\partial E_p}{\partial w_{ij}}=\delta_{ip}I_{jp} \tag{4.10}$$

δ_{ip}可分以下两种情况进行计算。

(a) 若i为输出节点，即$i=k$。

由式(4.4)和(4.9)，可得

$$\delta_{ip} = \delta_{kp} = \frac{\partial E_p}{\partial x_{kp}} = \frac{\partial E_p}{\partial y_{kp}} \cdot \frac{\partial y_{kp}}{\partial x_{kp}} = -e_{kp} f'(x_{kp}) \tag{4.11}$$

将式(4.11)代入式(4.8)，则

$$\frac{\partial E_p}{\partial w_{ij}} = -e_{kp} f'(x_{kp}) I_{jp} \tag{4.12}$$

(b) 若i不是输出节点，即$i \neq k$。

此时式(4.9)为：

$$\delta_{ip} = \frac{\partial E_p}{\partial x_{ip}} = \frac{\partial E_p}{\partial y_{ip}} \cdot \frac{\partial y_{ip}}{\partial x_{ip}} = \frac{\partial E_p}{\partial y_{ip}} f'(x_{ip}) \tag{4.13}$$

其中

$$\frac{\partial E_p}{\partial y_{ip}} = \sum_m \frac{\partial E_p}{\partial x_{mp}} \frac{\partial x_{mp}}{\partial y_{ip}} = \sum_m \frac{\partial E_p}{\partial x_{mp}} \frac{\partial}{\partial y_{ip}} \sum_j w_{mj} I^*_{jp} = \sum_m \frac{\partial E_p}{\partial x_{mp}} w_{mi} = \sum_m \delta_{mp} w_{mi} \tag{4.14}$$

式中：m—节点i后边一层的第m个节点；

I^*_{jp}—节点m的第j个输入(第p组样本输入时)。

当$i = j$时，

$$y_{ip} = I^*_{jp}$$

将式(4.13)、(4.14)代入式(4.8)，有

$$\frac{\partial E_p}{\partial w_{ij}} = f'(x_{ip}) I_{jp} \sum_m \frac{\partial E_p}{\partial x_{mp}} w_{mi} = f'(x_{ip}) I_{jp} \sum_m \delta_{mp} w_{mi} \tag{4.15}$$

可总结为由式(4.12)和(4.15)，即可进行式(4.7)的权值调整计算。

BP 网络通过若干简单的非线性处理单元的复合映射，可获得复杂的非线性处理能力。BP 算法用了优化算法中的梯度下降法，从数学角度看，它把一组样本的 I/O 问题变为非线性优化问题，经迭代运算，求解权值，使误差信号达到要求的程度。隐含层的作用是使优化问题的可调参数增加，使解更精确。

然而算法自身具有无法克服的缺陷，因为算法采用误差函数梯度下降的方式进行迭代，收敛速度慢，并且可能会陷入局部极值，而且具体的极值位置与权值的初始化数值密切相关；网络隐含层的层数及隐含层单元数等网络参数的选取没有理论的指导，算法只调整网络权值而对网络结构无法进行调整；学习样本的数量和质量，是影响学习效果和学习速度的重要因素，而对学习样本的选择需要一定的经验。BP 算法的缺点主要是它只利用误差函数的一阶偏导数信息，现在已经提出几个使用二阶导数信息的算法，主要是牛顿法、共轭梯度法和高斯牛顿法，但还是无法避免局部极小问题的出现。

目前，神经网络在控制中的应用，面临的两大问题是，神经网络拓扑结构的优化设计和高效的学习算法。进化算法的群体寻优、天然的增强式学习能力，使其具有全局性、并行性。快速性和自适应性，已成为解决上述两大问题的有力工具。

这里主要介绍利用进化策略优化神经网络，下面将详细介绍具体的操作方法。

4.2 基于进化策略的神经网络优化

进化策略是20世纪60年代初,柏林工业大学的Rechenberg和Schwefel等在进行风洞实验时提出的。它利用生物变异的思想，随机地改变参数值，获得了较好的结果。随后，他们便对这种算法进行深入的研究和发展，形成进化计算的另一个分支——进化策略。进化策略主要用于求解数值优化问题，近年来，进化策略除了用于函数优化外，在控制领域也有了广泛的应用，这里主要介绍进化策略用来优化神经网络。

神经网络的优化就是寻找一个逼近能力高且结构简单的网络。目前对网络优化多数学者采用结构优化和权值优化相分离的方法[3]，对网络结构的设计多采用增构法和减构法[4]。增构法就是网络训练从一个规模尽可能小的网络开始，训练过程中根据一定的判别标准增加网络的层数、结点及连接；减构法与之相反。但这两种方法极易陷入局部极小[5]。这样不能充分利用网络的逼近能力和泛化能力对网络进行整体评价。基于进化策略的人工神经网络优化的基本原理是用进化策略对神经网络的结构和连接权同时进行优化学习，利用进化策略的寻优能力来获取最佳的网络。

4.2.1 个体编码策略

编码是应用进化计算时应解决的首要问题，也是设计进化算法的一个关键。每种编码方式有其独特的特点，染色体直接采用实值表示，不需要编码和解码，这样提高算法的执行效率。

网络采用中间层内无连接，层外全连接的方式。由于神经网络由输入层、中间层(又称隐含层)和输出层构成，设输入层结点个数为n_i，阈值存在时将其作为一个输入，中间层结点个数为n_m，输出层结点个数为n_o。编码方式如图4.2：

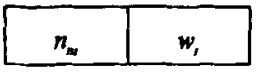

图4.2 编码机制

Fig 4.2 Mechanism of coding

其中，$i=1,2,\cdots,n_m$，w_i为与中间层第i个结点(神经元)相连的所有权值，权值排列自上向下，自左向右，将与同一结点相连的所有权值放在一起便于增加或删除结点操作。

初始隐含层单元数不是随意猜测的，在许多涉及BP网络的理论研究和实际应用中，都是考虑结构比较简单的三层BP网。考虑到待解决问题的性质，网络的输入层和输出层的单元数是确定的，因此如何确定隐含层单元数目，成为众所关心的重要技术课题。

文献[21]中曾经介绍过三种选择隐含层单元数n_m的公式如下：

$$k<\sum_{i=0}^{n_i}C\begin{pmatrix}n_m\\i\end{pmatrix} \tag{4.16}$$

式中：k为样本数，n_i为输入层单元数，当$i>n_m$时，取$C\begin{pmatrix}n_m\\i\end{pmatrix}=0$。

$$n_m=\sqrt{n_i+n_o}+l \tag{4.17}$$

式中：n_o为输出层单元数，常数$l=1\sim10$。

$$n_m=\log_2^{n_i} \tag{4.18}$$

比较上面三个式子不难看出，它们不仅数值不同，而且数量级不同。按(4.16)式，隐单元数与样本数k和输入层单元数n_i有关；按(4.17)式则与输入层单元数n_i和输出层单元数n_o有关；按(4.18)式则只与输入层单元数n_i有关。

此外，文献[21]的研究表明，要使多层网络具有泛化能力，网络的可调连接权总数W和必要的训练样本数N(假定样本集为高斯分布)之间的关系可以近似地表达为：

$$N\approx\frac{W}{\varepsilon}$$

式中ε取10左右。该式表明：隐含层单元数与训练样本数呈线性比例关系。

由于人工神经网络是一个极复杂的非线性动态系统，确定隐含层单元数时应注意：

(1) 隐含层单元过少，无法产生足够的连接权组合数来满足若干样本对的学习；隐含层单元过多，则学习以后网络的泛化能力变差；

(2) 如要求逼近的样函数变化剧烈、波动很大，则要求可调整的连接权数多，从而隐含层单元数也应该多一些；

(3) 如果规定的逼近精度高，则隐含层单元数也应该多些；

(4) 可考虑开始时放入较少的隐含层单元，学习一定的次数后如未学习成功，再逐步增加隐含层单元数。也可以先加足够多的隐含层单元，而后把学习中不太起作用的连接权和隐含层单元删去。

本文中间层结点数的初始选取根据经验公式[2]：

$$n_m=\sqrt{n_i+n_o}+l$$

其中，l在[1,10]之间随机选取，结点个数为n_m经过四舍五入后的整数。连接权一部分采用连接权值取值范围内的随机数，另一部分用零均值的正态分布产生，这样使所有的状态都得到遍历。

4.2.2 适应度函数的选取

适应度函数的选取要遵循自然界优胜劣汰的原则，适应能力越强的个体适应度要越大。对于一个网络，适应度函数的选取不仅要考虑网络的逼近能力，还要考虑它的泛化能力，鉴于此，选择适应度函数如下：

$$fitness=\frac{1}{E+\alpha\cdot n_m} \tag{4.19}$$

其中：E为样本的绝对误差，α为自定义的网络控制参数。

4.2.3 选择算子

进化初期，劣质的个体也有可能发展为后期的最优解，这里将种群进行分组[9]，选取父代个体时按一定的比例进行选择，使得劣质个体也有机会参与到进化中。父代优秀个体与劣质个体数采取前面第 3 章中给出的规则：

① $BestPar=\text{round}(BestInd/S_p)$ (4.20)

If $BestPar \geq Par$

$BestPar=\text{round}((0.75+t/(4T))\cdot Par)$ (4.21)

② $WorstPar=Par-BestPar$ (4.22)

$BestPar$表示作为父代的优秀个体数，$BestInd$表示优秀个体的数目，Par表示父代个体的数目，$WorstPar$表示作为父代的劣质个体数，S_p决定父代优秀个体与所有优秀个体的比例关系，t表示进化代数，T表示预先设定的最大进化代数。(4.21)式保证父代的优秀个体数不会超过父代总的个体数，且保证进化过程中若存在劣质个体，则进化过程中便有劣质个体的参与。进化后期劣质个体数量会不断减少，若计算出父代劣质个体数目大于实际存在的劣质个体的总数，那么，不足的劣质个体用余下的优秀个体补充。

4.2.4 变异算子

本文采用如下的变异方式：

$$x_{ij}'=x_{ij}+\gamma\cdot N_j(0,1) \tag{4.23}$$

其中$N_j(0,1)$为一个与个体X的第j分量相关的服从正态分布的随机量，$\gamma=(1-t/T)^b$，b为控制参数，这样的修正方法使变异产生的偏移随着进化代数的增长从

大范围逐渐到小范围。

变异后对个体根据中间层的结点数整理各个个体的码串长度，长则舍弃，不足的用权值取值范围内的随机数予以补充。

4.3 算法的基本步骤

Step1：设置进化代数计数器，确定所有的参数，随机产生初始种群，根据隐层结点数处理染色体长度；

Step2：计算初始种群中个体的输出、均方误差、适应度；

Step3：选择作为父代的个体，然后进行变异操作生成中间种群；

Step4：计算新种群中个体的输出、适应度；

Step5：判断是否满足进化结束条件，满足退出进化，否则转 Step6；

Step6：通过选择机制选择作为父代的个体，$t=t+1$；转 Step3。

4.4 实例仿真

非线性曲线拟合问题：

$$y = 0.4\sin(4\pi x) + 0.5 \qquad x \in [0,1] \text{的连续值。}$$

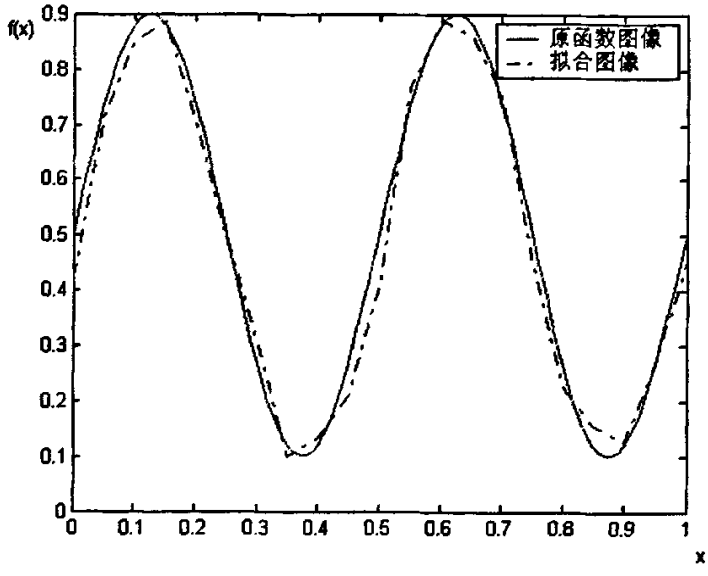

图 4.3 函数拟合效果图

Fig. 4.3 Chart of function drawing up

由于本算例中函数自变量和变量均为一维的，所以解决该问题的神经网络应该是单输入、单输出的。这里仍然采用的是(20,100)-ES，$S_p=2$。利用本文的算法得到进化神经网络的结构为 1—6—1，该网络不仅具有很好的泛化能力，而且也具有很好的逼近能力。图 4.3 为取 20 个样本点时，对非线性函数的拟合效果。

4.5 本章小结

本章首先介绍了人工神经网络的基础知识，重点介绍了多层前馈神经网络及其 BP 算法，在描述 BP 算法的同时，指出了它的不足之处。从而引出需要找到一种新的算法来解决这一问题。由于进化算法具有群体寻优的特点，且能够同时进化神经网络的网络结构及连接权，成为设计神经网络的首选算法。文中具体给出了利用进化策略进行人工神经网络优化的具体算子、算法的进化流程。最后，利用 MATLAB 给出算法最终实现得到的函数拟合效果图。仿真的结果表明利用进化策略设计的神经网络不仅具有很好的泛化能力且有很好的逼近能力。

第五章 总结与展望

5.1 全文总结

进化计算包括遗传算法、进化规划、进化策略三种基本的算法，它是一类基于自然选择和遗传变异等生物进化机制的全局概率搜索算法，是近年来智能计算领域的研究热点。与基于导数的解析方法和其它启发式搜索方法(如模拟退火法、单纯形法等)一样，进化计算在形式上也是一种迭代方法。它采用简单的编码来表示各种复杂的结构，并通过对一组编码表示进行简单的进化操作和优胜劣汰的自然选择，从而来指导学习和确定搜索空间。其采用的种群并行搜索方式使进化计算特别适合于解决在复杂情况下的大规模寻优问题。

随着进化算法的深入研究，进化策略已经作为它的一个分支得到了很多学者的关注，而且应用也越来越广泛。本论文在对进化策略理论与技术作以一般论述的基础上，对基本的进化策略进行适当的改进，将改进的进化策略用于求解有约束优化问题，并将其推广到求解一般的无约束优化问题上。文中最后一章给出了基于进化策略的人工神经网络优化设计，具体的叙述了利用进化策略进行神经网络优化设计的进化算子和算法实现的步骤。通过实例仿真证明了改进进化策略的实用性及有效性。

5.2 进化策略的研究现状及展望

虽然近年来进化算法得到越来越多的学者的关注，且它的应用领域也愈加的广泛，但是进化算法还只是简单的模拟生物界的进化机制，它的生物理论基础和数学理论基础还不是很完善，进化策略的应用研究已经远远的超过了它的理论研究。进化策略作为进化计算的一个分支，同样不可避免的也存在这方面的缺陷。所以，在以后的研究中，进化策略的理论分析(如编码表示方法、遗传算子、种群收敛性和多样性等)应该得到更加深入的探讨。当然，进化策略在其它的未知领域的应用也仍是今后进化策略研究的主导方向。

神经网络和进化计算都是借鉴生物进化机制而发展起来的人工智能理论方法。两者的结合使神经网络具有更优的性能。目前，基于进化计算的神经网络设计和实现已经成为神经网络领域的一个重要的研究和发展方向，国内外诸多学者在这方面已做了大量工

作。但从总体看，这方面研究还处于初级阶段，理论方法有待于加强提高。如目前的研究多基于具体事例，还没形成一般性的方法体系；计算量较大，进化进程需进一步提高，尤其是并行进化计算方法应受到重视，另外神经网络与进化计算相结合的其它方式也有待于进一步研究和挖掘。

但总的来说，对于进化策略的研究还是很有意义的，由于它在求解过程中不需要函数的梯度信息以及所要解决问题领域内的具体知识，进化算法的出现，给很多以往难以解决的问题带来了新的生机。因此，对进化策略的研究是很有必要的，当然对它的研究还要继续深入，发现它的不足，改进它的缺陷，提高算法的利用效率，使其在今后的社会发展中发挥更大的作用。

参考文献

1. Yao Xin. Evolving Artificial Neural Networks[J], Proceedings of the IEEE, 1999, 87(9): 1423－1447.
2. 何永勇，褚福磊，钟秉林. 基于进化计算的神经网络设计与实现[J]，控制与决策，2001，16(3)：257-262.
3. Peter J. Angeline, Gregory M, and Jordan B. Pollack. "An Evolutionary Algorithm that Constructs Recurrent Neural Networks"[J], IEEE Transactions on Neural Networks, 1994, 5(1): 54-64.
4. 邢文训，谢金星. 现代优化计算算法[M]，北京：清华大学出版社，1999.
5. Yao Xin, Liu Yong. Fast evolution strategies[J], Control and Cybernetics, 1997, 26(3): 467-496.
6. 王正志，薄涛. 进化计算[M]，湖南：国防科技大学出版社，2000.
7. 姚新，陈国良，徐惠敏，刘勇. 进化算法研究进展[J]，计算机学报，1995，18(9)：694-706.
8. 席裕庚，柴天佑，恽为民. 遗传算法综述[J]，控制理论与应用，1996，13(6)：697-708.
9. 吴庆洪，张纪会，徐心和. 一种有效的进化规划算法[J]，系统仿真学报，1999，11(6)：409-411.
10. 陆荣双，曾璐，杨文龙. 模拟进化计算理论及设计方法[J]，矿山机械，2005，33(2)：75-76.
11. 戴春妮，姚萌，谢祝捷，陈春宏. 一种新的进化计算算法模型——种群竞争消亡算法[J]，计算机应用，2005，25(1)：224-225.
12. Ming Yuchi, Jong Hwan Kim. Evolutionary algorithm using feasibility-based grouping for numerical constrained optimization problems[J], Applied Mathematics and Computation, 2006, 175(2): 1298-1319.
13. 王云诚，唐焕文. 单峰函数最优化问题的一个快速收敛的进化策略[J]，小型微型计算机系统，2002，23(11)：1390-1392.
14. 商允伟，裘聿皇. 一种求解数值优化问题的快速进化规划算法[J]，系统仿真学报，2004，16(6)：1190-1192.
15. 皮亦鸣，付毓生，黄顺吉. 采用进化计算的 BP 神经网络学习算法研究[J]，信号处理，2002，18(3)：261-264.
16. 柯晶，姜静，李歧强. 改进进化策略及其在神经网络训练中的应用[J]，计算机工程与应用，2006，19(4)：68-70.

17. Liu Yong and Yao Xin. "Evolutionary design of artificial neural networks with different nodes" in Proc. 1996 IEEE[J], Int. Conf. Evolutionary Computation (ICEC'96), Nagoya, Japan, 1996, 9(3): 670–675.
18. Greenwood G W. Training partially recurrent neural networks using evolutionary strategies[J], IEEE Trans on Speech Audio Proc, 1997, 5(1): 192-194.
19. Yao Xin, Liu Yong. A New Evolutionary System for Evolving Artificial Neural Networks[J], IEEE Transactions on Neural Networks, 1997, 8(3): 694-713.
20. 叶立生，何奉道. 基于进化规划的 BP 神经网络学习[J]，西南交通大学学报，2001，36(5)：545-548.
21. 靳蕃. 神经计算智能基础原理方法[M]，成都：西南交通大学出版社，2000.
22. 陈得宝，赵春霞. 一种改进的自适应免疫进化规划方法及其应用[J]，系统仿真学报，2006，18(5)：1147-1150.
23. 阎岭，蒋静坪. 进化学习策略收敛性和逃逸能力的研究[J]，自动化学报，2005，31(6)：873-880.
24. 刘芳，李人厚. 基于自适应变异规则的一种有效的进化规划[J]，控制与决策，2002，17(2)：148-150.
25. 陈明. 基于进化遗传算法的优化计算[J]，软件学报，1998，9(11)：876-879.
26. 罗文辉. 遗传算法在神经网络优化中的应用[J]，控制工程，2003，10(5)：401-403.
27. 云庆夏. 进化算法[M]，北京：冶金工业出版社，2000.
28. 王旭，王宏，王文辉. 人工神经元网络原理与应用[M]，沈阳：东北大学出版社，2000.
29. 徐丽娜. 神经网络控制[M]，哈尔滨：哈尔滨工业大学出版社，1999.
30. Vittorio Maniezzo. Genetic Evolution of the Topology and Weight Distribution of Neural Networks[J], IEEE Transactions on Neural Networks, 1994, 5(1): 39-53.
31. 武妍，张立明. 神经网络的泛化能力与结构优化算法研究[J]，计算机应用研究，2002，19(6)：21-25.
32. 张纪会，徐心和. 模拟进化算法研究进展[J]，系统工程与电子技术，1998，20(8)：44-47.
33. 李孝安，康继昌，蔡小斌，戴冠中. 进化神经网络研究进展[J]，控制与决策，1998，13(6)：617-623.
34. 陈荣，徐用懋，兰鸿森. 多层前向网络的研究——遗传 BP 算法和结构优化策略[J]，自动化学报，1997，23(1)：43-49.
35. 何述东，瞿坦，黄献宵，黄心汉. 多层前向神经网络结构的研究进展[J]，控制理论与应用，1998，15(3)：313-319.
36. 王战权，唐春安，云庆夏. 进化规划和进化策略中变异算子的改进研究[J]，华东冶金学院学报，1999，16(4)：295-300.

37. 王战权，赵朝义，云庆夏，唐春安. 进化策略中变异算子的改进研究[J]，计算机仿真，1997，16(3)：8-11.

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

39. Ji-Hui Zhang, Xin-He Xu.An efficient evolutionary programming algorithm[J], Computers & Operations Research, 1999, 26(7): 645-663.

40. Zong-Ben Xu, Kwong-Sak Leung, Yong Liang, Yee Leung. Efficiency speed-up strategies for evolutionary computation: fundamentals and fast-Gas[J], Applied Mathematics and Computation, 2003, 142(2): 341 - 388.

41. Yao Xin, Liu Yong. Towards designing artificial neural networks by evolution[J], Applied Mathematics and Computation, 1998, 91(1): 83-90.

42. 阎平凡，张长水. 神经网络与模拟进化计算[M]，北京：清华大学出版社，2004.

43. 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[J]，天津大学学报，2000，33(5)：627-630.

44. Ajay K. Gupta, Garrison W. Greenwood. Static task allocation using (μ, λ) evolutionary strategies[J], Information Sciences, 1996, 94(4): 141－150.

45. Christian Igel, Martin Kreutz. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks[J], Computers & Mathematics with Applications, 2003, 46(10-11): 1493-1509.

46. R. Berlich, M. Kunze. Parallel evolutionary algorithms[J], Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2003, 502(2-3): 467-470.

47. Mehrdad Salami, Tim Hendtlass. A fast evaluation strategy for evolutionary algorithms[J], Applied Soft Computing, 2003, 2(3): 156-173.

48. Nikolaos D. Lagaros, Manolis Papadrakakis, George Kokossalakis. Structural optimization using evolutionary algorithms[J], Computers & Structures, 2002, 80(7-8): 571-589.

49. M. Mandischer. A comparison of evolution strategies and back propagation for neural network training[J], Neurocomputing, 2002, 42(1-4): 87-117.

50. 侯进军，彭宏. 一种新的快速进化策略的设计与实现[J]，吉首大学学报，1999，20(2)：31-34.

51. 王湘中，喻寿益. 多模态函数优化的多种群进化策略[J]，控制与决策，2006，21(3)：285-288.

52. 史天运，贾利民. 基于进化策略的动态递归神经网络建模与辨识[J]，控制与决策，2000，15(4)：439-442.

53. 俞健. 基于进化计算的神经网络设计方法[D]，浙江：浙江大学，1998.
54. 阎平凡. 对多层前向神经网络研究的几点看法[J]，自动化学报，1997，23(1)：129-135.
55. 焦李成. 进化计算与进化神经网络——计算智能的新方向[J]，电子学报，1995，3(1)：9-19.
56. 谢金星. 进化计算简要综述[J]，控制与决策，1997，12(1)：1-7.
57. Ko-Hsin Liang, Xin Yao, Charles S. Newton. Adapting Self-Adaptive Parameters in Evolutionary Algorithms[J], Applied Intelligence, 2001, 15(3): 171-180.
58. W. Gutkowski, Z. Iwanow, J. Bauer. Controlled mutation in evolutionary structural optimization[J], Structural and Multidisciplinary Optimization, 2001, 21(5): 355-360.

致 谢

首先感谢我的导师邢伟教授，从论文题目的选定到论文的完成，老师给了我悉心的指导，在生活上给了我无微不至的关怀。邢老师严谨的治学态度、渊博的知识、踏实的工作作风以及勤奋的工作精神，是永远值得我学习的。这种影响对我来讲是巨大的，有益而持久的！

感谢系统科学研究所的各位老师在攻读硕士学位期间对我生活和学习上的关心和帮助！

感谢我的室友杨芳、王新、蔡娜、白云姣、徐涛，因为有了大家的互相鼓励与帮助，我愉快的度过了两年多的硕士学习生涯！

感谢我的父母，他们在生活上给了我莫大的关怀，因为有了他们的理解和支持，我的学业才得以更加顺利，他们的情感永远是我上进的动力源泉！

感谢所有关心，理解和鼓励我的亲人和朋友！

最后对参加论文评审和答辩的各位老师表示衷心的感谢！

硕士期间发表的论文

1. 李影，徐涛，邢伟. 基于进化遗传算法的神经网络优化，长春理工大学学报，2006，29(3)：48-50.
2. 李影，邢伟. 一种改进的进化算法，控制工程. (已录用)
3. 李影，邢伟. 进化计算理论及其应用，宁夏大学学报. (已投稿)
4. 李影，邢伟. 基于适应度分组的进化策略，系统仿真学报. (已投稿)