分类号 ……………………　　　　密级 ……………………

U D C ……………………　　　　编号 ……………………

中南大學

CENTRAL SOUTH UNIVERSITY

硕 士 学 位 论 文

论 文 题 目　多目标进化算法及其在约束优化中的应用研究

学科、专业　模式识别与智能系统

研究生姓名　曾威

导师姓名及专业技术职务　蔡自兴 教授

日　　期

MS THESIS

Research on Multiobjective Evolutionary Computation and Its application on Constrained Optimization

Specialty： Pattern Recognition and Intelligent System

Master Degree Candidate： Wei Zeng

Supervisor： Prof. Cai-Zixing

College of Information Science & Engineering

Central South University

ChangSha Hunan P.R.C

摘 要

科学和工程中的很多优化问题都是多目标问题，因此，对其进行研究非常具有实际意义和科研价值。多目标优化问题中各目标之间往往相互制约，对其中一个目标优化必须以其它目标作为代价。80 年代中期以来，进化算法开始应用于该问题，并行成了最近的一个热门领域，对多目标进化算法的研究也是近年来求解多目标优化的重点。

本文首先对多目标优化问题的一般定义、Pareto 最优解集和 Pareto 最优边界等进行了讨论，对多目标优化进化算法的国内外研究现状进行了回顾，然后提出了两种多目标优化进化算法，并将多目标优化的思想用来求解约束优化问题。主要内容如下：

1、将多目标属性决策方法中的 ELECTRE 法引入到多目标优化进化算法中，提出了一种新的多目标优化算法。构造出一种新的超序关系对个体进行排序，并证明了该超序关系比 Pareto 优劣关系弱，利用此超序关系能增加进化过程中的选择压，加快收敛速度。通过数据试验，表明算法能很好地收敛到 Pareto 最优，有效保持解的多样性。

2、提出了一种基于差异进化的多目标优化算法，利用ε优超关系来保存进化中的非劣个体 ，并对ε-MOEA 算法中个体存档准则进行了改进。将ε-水平比较准则引入到算法中，通过对一系列标准的测试函数进行实验，实验结果表明该算法在保持解集分布性和收敛性方面非常有效。

3、将多目标优化思想引入到约束优化中，提出了非劣个体替换准则。为了有效利用不可行解，提出了一种不可行解保存和替换机制，对 6 个测试函数进行了测试，测试结果表明该算法优于其它算法。

最后，提出了多目标进化算法中值得进一步研究的问题。

关键词 多目标进化算法, 约束优化, ELECTRE 法, 差异进化

ABSTRACT

Many optimization problems encountered in the disciplines of science and engineering are multi-objective problems (MOPs), therefore, to do research on MOPs has a great value. Evolutionary algorithms of Artificial Intelligence were initially applied to those problems from the mid-eighties. Thus a popular area of research has formed recently. Research on multiobjective evolutionary algorithms for multiobjective optimization problems is still very important

This dissertation is starts with the discussions of general definition of multi-objective optimal problem, Pareto optimal Solutions, Pareto optimal Front and reviewing the advances of MOEAs(Multi-Objective Optimization Evolutionary Algorithms) at home and abroad. Then, two MOEAs and one COPs(Constrained Optimization Problems) based on Multi-objective optimization are proposed by author. The main contribution and work are described as following:

1、Proposed a new mutiobjective evolutionary algorithm. based on ELECTRE method. The proposed algorithm uses a secondary population in order to retain the non-dominated solutions found during the evolutionary process and adopts the same fitness assignment strategy as SPEA-II to get well distributed solutions. Additionally, a novel outranking relationship is constructed, and proved to be weaker than Pareto dominance relation. Experiment results show that this algorithm can converge to true Pareto Front well and effectively maintain diversity of the solutions.

2、Investigate a new multi-objective evolutionary algorithm based on differential evolution. The proposed approach adopts a secondary population in order to retain the non-dominated solutions found during the evolutionary process. Additionally, the approach also incorporates the concept of ε-dominance to get a good distribution of the solutions retained. We adopted standard test functions and performance measures reported in the specialized literature to validate our proposal. Our results are compared with respect to two approaches that are representative of the state-of-the-art in the area: the NSGA-II and ε-MOEA.

3、Incorporated the idea of mulitobjective optimization into Constrained Optimization problems. Proposed the non-dominated individual replacement scheme. Additionally an infeasible solutions archiving and replacement mechanism is introduced to effectively exploit infeasible solutions. The algorithm is tested on six benchmark functions, and the result shows that this algorithm is outperforms others compared with some other state-of-the-art algorithms.

Finally, this dissertation points out some directions that are worthy to be researched further in this area.

Keywords multi-objective optimization evolutionary algorithm, constrained optimization, ELECTRE method, differential evolution

目 录

第一章 绪论

1.1 多目标优化概述

在实际应用中，人们经常遇到需要使多个目标在给定区域上均尽可能最佳的优化问题。例如在设计新产品时，既要考虑使产品具有较好的性能，又要考虑使其制造成本最低，同时还要考虑产品的可制造性、可靠性、可维修性等，这些设计目标的改善可能相互抵触，譬如好的可维修性会引起可靠性的降低，因此须在这些设计目标之间取一折衷结果。再如投资问题，一般希望所投入的资金量少，风险最小，且所获得的收益最大。这种多于一个的数值目标在给定区域上的最优化问题一般就称为多目标优化问题(MOPs: Multiobjective Optimization Problems)[1]。多目标有时也称多准则、多属性或多指标，目标分为总目标和子目标，这里所谓的多目标优化是指对多个子目标同时实施最优化。这些实际问题非常复杂、困难，要很好地解决这类优化问题需要科研人员投入更多的精力。因此，解决多目标优化问题是一个非常有实际意义和科研价值的课题。

1.2 进化算法及多目标进化算法

进化算法 EA:（Evolutionary Algorithm）是一种模拟自然进化过程的随机优化方法。EAs 的起源可追溯到 50 年代末 70 年代以来产生了几种进化方法如：遗传算法 GA:（Genetic Algorithm）、进化规划 EP:（Evolutionary Programming）、和进化策略 ES:（Evolution Strategy）。进化算法是模拟由个体组成的群体的集体学习过程，其中每个个体表示给定问题搜索空间中的一点，进化算法从任一初始的群体出发，通过随机选择（在某些算法中是确定的）变异和交叉（有时被完全省去）过程使群体进化到搜索空间中越来越好的区域[2,3]。

差异进化算法由 Storn 和 Price[4]提出，它是一种基于种群的随机搜索算法，对实数值、多模式目标函数优化具有很好的寻优效果。差异进化算法的特征是：简单，高效，采用浮点数编码而非二进制编码，对初始值无要求，收敛速度快，适合求解非线性问题。同其他进化算法类似，差异算法也是先随机产生初始种群，然后采用选择，变异交叉等操作，对种群进行不断的优化。在本文第四章中有对差异进化算法的详细介绍。

最近大量事例和迹象表明 EAs 的机理最适合求解多目标优化,因为它们可在单轮模拟过程中找到多个 Pareto 最优解，通过逐代组合寻找具有某些特征的

个体。甚至有的学者认为，在多目标优化领域 EAs 要优于其它盲目搜索方法[5, 6]。虽然这种提法与最优化领域中的没有免费的午餐（no free lunch）定理[7]不太吻合，但迄今为止还没有找到其它方法比 EAs 更能有效地解决 MOPs[8]。多目标进化算法（MOEA: Multiobjective Evolutionary Algorithm）在国外的许多应用案例和掀起的研究热潮已成为事实。

进化算法应用于多目标优化领域，始于 80 年代中期[9,10]。在 1991—1994 年期间，有几种 MOEAs 被提出来[11,12,13]，后来这些方法及其变种在实际多目标优化问题中得到了成功的应用[14,15]。近年来一些学者集中研究了基于进化算法的多目标搜索中的某些专题，如 Pareto 最优前端的收敛性[16]、小生境技术[17]、精英策略[18,19]。同时有关权威人士在不同时间对 MOEAs 分别进行了总结和综述[20,21]。

1.3 约束优化进化算法

约束优化问题（constrained optimization problems，简称 COPs）是一类广泛存在实际工程中但又较难求解的问题，传统的求解这类问题的方法通常是基于梯度的搜索方法，如可行方向法、投影梯度法、简约梯度法、各类外点罚函数法和内点法、Lagrangian 法和二次规划法等。这些方法存在的主要问题是求解需要设置很好的初值点和需要函数的梯度信息，他们对于不可导、可行域不连通、甚至根本没有显式数学表达式等问题无能为力。同传统的优化方法相比，进化算法更适合于求解约束优化问题，已有许多学者进行了广泛的研究，并且提出了大量的约束优化进化算法[22,23,24,25,26]。

进化算法在约束优化问题中的成功应用，取决于以下几个主要原因：

1 进化算法从一个群体即多个点而不是一个点开始搜索，这使它能以较大概率找到全局最优解；

2 同传统优化方法相比，进化算法对所优化问题的特征不敏感；

3 进化算法很容易执行和使用。

1.4 本论文的主要工作

在吸取已有研究成果的基础上，本文设计和讨论了两个多目标优化算法，并将多目标优化的思想应用到约束优化中。1）针对单纯 Pareto 进化算法很难解决目标数目很多的优化问题，本文在多个指标之间引入偏好信息，提出一种求解高维多目标优化的新型进化算法，使进化群体按协调模型进行偏好排序而不是单纯的基于 Pareto 优于关系来比较个体优劣。2）将 ε 水平比较准则引入到多目标进化算法中，提出一种新的基于差异进化的多目标进化算法。

后续章节安排如下：

第二章 给出了多目标优化问题的一般定义，对具有代表性的多目标优化进化算法进行了介绍并分析了各算法的优点以及不足之处，对国内外的研究现状进行了介绍。

第三章 首先分析了基于 Pareto 比较准则在求解高维多目标优化问题的缺陷，然后对多目标决策中的 ELECTRE 进行了介绍，将其引入到多目标优化进化算法中，定义了一种新的关系，提出了一个新的多目标优化算法，最后，用实验方法验证了本文提出的算法的有效性和正确性。

第四章 首先对差异进化算法进行了介绍，提出了一种新算法，在处理约束条件时，将ε水平比较准则引入到算法中，利用ε优劣关系来保存外部种群个体，提高分布的均匀性，最后对一些测试函数进行了测试，同目前流行的算法进行了比较。

第五章 首先对约束优化问题作了简单的介绍，对目前存在的算法进行了分类。接着，提出了一种基于多目标优化技术的约束优化进化算法。该方法的主要特点是利用多目标优化思想比较和选择个体，并将基于群体的算法发生器模型与一个不可行解存档和替换机制相结合。最后，通过 6 个标准的测试函数对新算法的优化性能进行了测试。

第六章 对于本论文的研究进行了总结，并对有待进一步研究的问题进行了分析和展望。

第二章 多目标进化算法综述

进化算法（EA：Evolutionary Algorithm）以其擅长求解高度复杂的非线性问题而得到了非常广泛的应用，同时它又具有较好的通用性。在解决只有单个目标的复杂系统优化问题时，充分展现了进化算法的优势。但是，现实世界中的优化通常是由多个目标组成的，如一个国家的最优良性发展，涉及到经济的快速增长、社会秩序的稳定等多个方面。在这里，经济的快速增长和社会秩序稳定这两个优化目标是相辅相成的，互相促进的，通常称之为一致的。与此同时，有一些优化问题，如产品质量与生产成本，也是多目标优化问题，但其被优化的多个目标之间是相互冲突的（站在生产者的角度来看）。当我们使用进化算法来解决多个目标的优化问题时，就显得有点力不从心了。我们为了寻找总目标的最优化，对于相互冲突的子目标通常需要综合考虑，即对各个子目标进行折衷（tradeoffs），这样就存在多个解。针对多目标的优化问题，出现了多目标进化算法（MOEA：Multi Objective Evolutionary Algorithm）.

无论在科学或工程上，多目标优化问题都是非常重要的研究课题。这不仅仅是因为许多现实世界中的问题涉及到多目标优化，还有一些与多目标优化有关的问题也是难以回答的，如最优解，它不同于单目标的优化，通常由多个；对于多个最优解，究竟哪个是我们要找的呢？这取决于决策者。

早在 1967 年 Rosenberg 就建议采用基于进化的搜索来处理多目标优化问题，但他没有具体实现。1984 年，David Schaffer 首次在机器学习中实现了向量评估遗传算法（VEGA：Vector Evaluated Genetic Algorithm）。此后，多目标进化算法的研究进入了低谷，直至 1989 年 David Goldberg 在其著作《Genetic algorithms for search, optimization, and machine learning》中，提出了用进化算法实现多目标的优化技术，使多目标进化算法的研究出现了重大转折。近 17 年来，多目标进化算法（或多目标遗传算法，MOGA: Multi Objective Genetic Algorithm）引起了许多研究者的广泛关注，并涌现出了大量的研究成果。

2.1 问题的描述

为了方便阐述多目标进化算法的研究现状，我们在此先讨论一下多目标优化问题的一般定义。值得注意的是，有关多目标优化问题及其最优解的定义在许多专著或论文上都有介绍，此处讨论的定义只是其中的一种。

给定决策向量 $X=[x_1,x_2,...x_n]^T$，它满足下列约束[21]：

$$g_i(X) \geq 0 \quad i = 1, 2, ..., k \tag{2-1}$$

$$h_i(X) \geq 0 \quad i = 1, 2, ..., l \tag{2-2}$$

设有 r 个优化目标，且这 r 个优化目标可能是相互冲突的，优化目标可表示为：

$$f(X) = [f_1(X), f_2(X), ... f_r(X)]^T \tag{2-3}$$

寻求 $X^* = [x_1^*, x_2^*, ... x_n^*]^T$，使 $f(X^*)$ 在满足约束（2-1）和（2-2）的同时达到最优。

这里所讲的最优，是由 Vilfredo Pareto 在 1896 年提出的，故称之为 Pareto 最优解（Pareto Optimum Solution），定义如下：

称 $X^* \in F$ 是最优解（即 Pareto Optimal Solution），若 $\forall X \in F$ 满足：

或者 $$\bigcap_{i \in I} (f_i(X) = f_i(X^*)) \tag{2-4}$$

或者至少存在一个 j，使：

$$f_j(X) > f_j(X^*) \tag{2-5}$$

其中 F 是满足式（1-1）和式（1-2）的可行解集，即：

$$F = \{X \in R^n \mid g_i(X) \geq 0, i = 1, 2, ...k; h_j(X) = 0, j = 1, 2, ...l\} \tag{2-6}$$

通常情况下，最优解不只一个，而是一个最优解集（Pareto Optimal Solutions）。所需做的工作时，构造非支配集（non-dominated solutions 或 non-inferior solutions）,并使非支配集不断逼近最优解集，最终达到最优。

多目标进化算法的研究主要实现三个目标。一是使算法收敛到 True Pareto 最优边界；二是使解集具有好的分布性或多样性；三是在确保算法收敛到 True Pareto 最优边界，并使解集具有好的分布性的前提下，算法具有较快的收敛速度。从已有的研究成果中，如文献[27]，可以看出，具有良好分布性的算法通常需要较大的时间开销；同样地，运行效率比较高的算法，其解集的分布性和多样性就相对较差一些。接下来，本文将较为详细地介绍多目标优化进化算法的研究成果，同时，也概要性地介绍所存在的不足。概括起来，当前比较典型的多目标进化算法主要有：

（1） 聚集函数法（Aggregation Function）

（2） Schaffer 的向量评估进化算法（VEGA）

（3） Fonseca 和 Fleming 的 MOGA

（4） Horn 和 Nafpliotis 的 NPGA

（5） Deb 等提出的 NSGA-II

（6） Zitzler 等提出的 SPEA2

（7） Corn 等提出的 PESA2

2.2 聚集函数法

聚集函数法将多目标优化问题中的多个目标进行线性组合，并对各个子目标赋予不同的权值，将其转化为单目标问题的优化。

设有 r 个子目标 $f_1(\ )$、$f_2(\ )$、…、$f_r(\ )$，其对应的适应度函数分别为 f_1、f_2、…、f_r，采用权重系数法可综合目标适应度为：

$$\omega_1 f_1+\omega_2 f_2+\ldots+\omega_r f_r \tag{2-7}$$

其中 $\omega_i \geq 0\ \ i=1,2,\ldots r$

且满足：$\sum_{i=1}^{r}\omega_i=1$ （2-8）

优化目标可表示为：$Min\sum_{i=1}^{r}\omega_i f_i(\bar{x})$ (2-9)

这是早期使用的方法，将多目标优化问题转化为单个目标的优化问题，方法比较简单。这种方法的主要优点是效率比较高，比较适合产生初始非支配集（non-dominated solutions），然后再用其他方法进一步优化。但缺点也比较明显，如在群体进化过程中通常要不断地改变权值，而对一个具体的多目标优化问题，要准确地找出权重系数的变化规律也不是一件容易的事情；有时根据需要也可以考虑使用惩罚函数（penalty function）,对参数或惩罚函数稍有改变，算法都表现得非常灵敏。更为严重的是，当搜索空间呈凹状时，无论如何改变权值都难以找到合适的最优解（即 Pareto Optimal Solutions）。这种方法已被证明存在着严重问题。

2.3 Schaffer 的向量评估遗传算法

Schaffer 于 1984 年，在他的博士论文中，提出了向量评估进化算法（VEGA：Vector Evaluated Genetic Algorithm）[10]，并用 VEGA 求解多目标优化问题，但似乎只能发现最优边界（Pareto Optimal Front）上的极端点，因为它不能根据各子目标的属性（attribute）进行折衷和权衡。

Schaffer 对简单进化算法（SGA：Simple Evolutionary Algorithms）进行扩充，使 SGA 对单个目标的处理变成 VEGA，从而可以对目标向量进行处理，即对多个目标进行处理。设有 r 个目标，针对每个目标利用比例选择算法，分别可产生 r 个子群体，每个子群体的大小为 N/r，其中 N 为群体大小。将 r 个子群体合并为一个群体大小为 N 的新群体，进行交叉操作和变异操作。Schaffer 认识到，采用他的 VEGA 所产生的非支配解（non-dominated solutions）不是全局的，因为其非支配解集总是限制在当前群体，局部的被支配个体（locally dominated individual）肯定是全局被支配个体，（globally dominated individual），但局部的非支配个体（locally non-dominated individual）未必就是全局非支配个体（globally non-dominated individual）。一个个体在当前是非支配的，可能在下

一代或后续代变成被支配个体。另一个更为严重的问题就是“物种形成（speciation）”，如在群体中可能存在着某些方面表现突出的物种。究其原因是因为在选择操作时只考虑了一个目标而忽视了其它的目标。“物种形成（speciation）”与寻求折衷解（compromise solution）的目标是不一致的。Schaffer 建议使用启发式方法弥补上述缺陷。

VEGA 的主要优点是简单，但当搜索空间呈凹状时难以找到 Pareto 最优解集（Pareto Optimal Solutions）。

2.4 Fonseca 和 Fleming 的 MOGA

Fonseca 和 Fleming 在 MOGA 中提出了一种方法[13]，对每个个体分别计算其分类序号（rank），所有非支配个体的分类序号定义为 1，其它个体的分类序号为支配它的个体数目加 1。设 n_i 为支配个体 i 的个数，则对任意一个个体 i 有分类序号：

$$rank(i)=1+n_i \tag{2-10}$$

这样有可能存在多个个体具有相同的分类序号的情况，选择操作按分类序号从小到大依次进行，具有相同分类序号者用目标函数共享机制进行选择。

在 MOGA 中，依据个体之间的支配关系来确定个体的分类序号，这是一个非常重要的工作。但这种方法可能产生较大的选择压力从而导致非成熟收敛。此外，当多个不同的 Pareto 最优解（Pareto Optimal points）对应于相同的目标函数值时，MOGA 难以找出多个解。

MOGA 的主要优点是运行效率高，同时比较易于实现。它的不足之处是过于依赖共享参数的选择。

2.5 Horn 和 Nafpliotis 的 NPGA

多目标进化算法的目标是寻找一个满足总目标最优化的解集，这个解集称之为 Pareto 最优解集（Pareto Optimal Set）。Goldberg 建议使用“非支配排序（non-domination ranking）方法[9]，使候选解不断集散到最优解集边界上（Pareto Optimal Front）。Horn 和 Nafpliotis 等基于 Pareto 支配关系，提出了 NPGA（A Niched Pareto Genetic Algorithm for Multiobjective Optimization）[13]。采用小生境技术（niche）实现共享来维持候选解集（non-dominated Set）的多样性。

随机地从进化群体中选择两个个体 i 和 j，再随机地从进化群体中选取一个比较集 CS（其 size 大于 2，一般约为 10），然后用个体 i 和个体 j 分别与 CS 中的个体进行比较，如果其中之一受 CS 支配（称之为 dominated individual），而另一个个体不受 CS 支配（称之为 non-dominated individual），那么这个 non-dominated individual 将被选中参入下一代进化。如果个体 i 和个体 j 都不受

或都受 CS 支配，则只能采用共享机制来选取其中之一参入下一代进化。将选择共享适应度大的（或小生境计数小的）个体参入下一代进化操作。

适应度共享（fitness sharing）是实现群体多样性的有效方法。设个体 i 针对某个子目标的适应度为 f_i，个体 i 的小生境计数（niche count）为 m_i, m_i 的计算方法如下：

$$m_i = \sum_{j \in Pop} sh[d(i,j)] \tag{2-11}$$

其中 Pop 为当前进化群体，$d(i,j)$ 为个体 i 和 j 之间的距离或称相似程度，$sh[d]$为共享函数，$sh[d]$的定义如下：

$$sh[d] = \begin{cases} 0 & d > \delta_{share} \\ 1 - d/\delta_{share} & d < \delta_{share} \end{cases} \tag{2-12}$$

其中 δ_{share} 为小生境半径（niche radius），通常由用户根据最优解集（Pareto Optimal Set）中个体之间的最小期望间距来确定。

定义 f_i/m_i 为共享适应度（sharing fitness），此处 m_i 实质上就是个体 i 的小生境之聚集程度。同一小生境内的个体互相降低对方的共享适应度。个体的聚集程度越高，其相对于适应度的共享适应度就被降低的越多。使用共享适应度的目的在于将进化群体分散到整个搜索空间（search space）的不同峰值区域上。在经典 GA 中引入小生境技术的目的是为了防止早熟收敛。在 MOGA 中，当共享适应度与锦标赛选择（tournament selection）相结合时可能会出现混沌行为（chaotic behavior）[19]，此时小生境子群体内不会出现剧烈的波动。文献[19]建议不断修改共享参数。

NPGA 的主要优点是运行效率比较高，且能获得较好的 Pareto 最优边界（Pareto Optimal Front）。它的不足是共享参数的选择以及比较集大小的选择没有一个通用的法则。

2.6 Deb 等提出的 NSGA-II

Deb 等提出了非支配集排序方法 NSGA-II[28]。算法采用了快速排序的方法来构造非支配集：根据非支配级别对一个规模为 N 的种群进行排序，每一个解需要与任何一个其它的解比较是否被支配，在所有种群个体中，寻找出非劣个体，归为第一层（rank1），找出剩下个体中的非劣个体，归为第二层（rank2），重复上述过程，直到所有个体都被分属某一层次中。

另外它采用聚集比较过程保持解得分布性：计算同一面上相邻的两个个体在每个目标上距离的平均值。在种群中将仅包含个体 i 的最大的立方体的大小 $i_{distance}$ 作为一个评估（称为密集距离）。然后基于密集距离对同一个体面上的点进行排序，选择分布度较好（密集距离大）的个体。

算法 NSGA-II：

输入：N（种群规模）

T（最大遗传代数）

输出：Nds（非支配集）

1）初始化：产生初始化的种群 P_0，t=0。

2）交配选择：使用 2-锦标赛选择 P_t 中的个体进入交配池，其规模等于 N

3）遗传操作：对交配池中个体使用交叉和变异操作，产生新的个体进入 Q_t

4）适应度赋值：计算 P_t 和 Q_t 中个体的适应度。

5）环境选择：将 P_t 和 Q_t 中的所有非支配个体拷入非支配集 Nds。如果 Nds 的规模超过 N，使用聚集算法计算个体的聚集度，选择聚集度低的个体优先进入 P_{t+1}，直到规模等于 N；如果其规模小于或者等于 N，拷贝 Nds 中的个体进入 P_{t+1}，规模不够 N 的部分，选择适应度高的支配个体进入 P_{t+1},直到其规模等于 N。

6）终止：如果 t<T，t=t+1，去第 2 步，否则将 P_{t+1} 中的非支配个体作为 A 输出。

NSGA-II 的优点在于运行效率高；解集有良好的分布性，特别是在低维问题的优化上有很好的表现。缺点在于在高维问题上，因为聚集过程的缺陷，解集多样性不理想。

2.7 Zitzler 等提出的 SPEA2

Zitzler 和 Thiele 于 1999 年提出了 SPEA（Strength Pareto Evolutionary Algorithm）[29]，主要有下列特点：

（1） 另外设置一个非支配集 NDSet（Non-Dominated Set），且其随种群的不断进化而更新；

（2） 用个体强度来对个体进行适应度赋值。

（3） 采用聚类过程来降低 NDSet 的大小并维持群体的多样性。

Zitzler 等对 SPEA 算法进行了改进，于 2001 年提出算法 SPEA2（Improving the Strength Pareto Evolutionary for Multi-objective Optimization）[30]。该算法与 SPEA 的主要区别在于：

（1） 用了改进的适应度评价策略，它考虑了每一个个体的支配与被支配情况

（2） 采用了最近邻居密度估计策略，能对搜索过程作精确指导

（3） 采用了新的外部截断方法，可以保证不丢失边界解

算法 SPEA2：

输入：N（群体大小）

$\bar{N}$（外部集大小）

T（运行代数）

输出：A（非支配集）

1）初始化：初始化群体 P_0 和空的外部集 $\bar{P}_0=\phi$，并设置当前代 t=0。

2）适应度评价：计算当前代群体 P_t 和外部集 $\bar{P}_t$ 中的个体适应度值。

3）环境选择：把当前代群体 P_t 和外部集 $\bar{P}_t$ 中的非支配个体复制到下一代的外部集 $\bar{P}_{t+1}$ 中。如果 $\bar{P}_{t+1}$ 中个体的数目超过 $\bar{N}$，就采用截断算法来缩小 $\bar{P}_{t+1}$；如果 $\bar{P}_{t+1}$ 中个体数目少于 $\bar{N}$，就把当前代群体 P_t 和外部集 $\bar{P}_t$ 中的非支配个体填充到 $\bar{P}_{t+1}$ 中。

4）终止条件：如果 t≥T 或者满足另外的终止条件，就用一个集合 A 来存储 $\bar{P}_{t+1}$ 中的非支配个体。

5）交配选择：在 $\bar{P}_{t+1}$ 中采用二进制锦标赛算法来选择个体添加到交配池中。

6）变异：应用交叉和变异操作到交配池中，并将 $\bar{P}_{t+1}$ 存储为解集，增加代数（t=t+1），然后转到第二步。

SPEA2 的优点在于可以取得一个分布度很好的解集，特别是在高维问题的求解上，这一点很明显。缺点在于其聚类过来保持多样性时间耗费很大，运行的效率不高。

2.8 Corn 等提出的 PESA2

PESA2 采用基于网格的区域选择，其具体算法如下：

输入：N（群体大小）

M(外部集大小）

T（最大遗传代数）

输出：A（非支配集）

1） 初始化：产生初始化的外部种群 P_0 和一个空的外部集 A_0,t=0。

2) 将 P_t 中的非支配个体一个接一个进入 A_t，并更新 A_t，使其保持它的非支配性。如果 A_t 的规模大于 M，使用网格，计算密集度，淘汰一个差的个体。

3） 终止：如果 t<T，继续第 2 步，否则，将 A_t 作为 A 输出。

4） 根据区域密度，使用基于网络的区域选择，按概率选择 A_t 中的个体进行交叉和变异操作产生新的个体，直到填满 P_{t+1}。T=t+1，转到第 2 步。

PESA2 的优点在于其解的收敛性很好，往往能很接近最优面，特别是在高维问题的情况下。但其缺点在于选择操作只能一次选取一个个体，时间耗费太

长，而且解集的多样性不佳。

2.9 理论研究方面

目前国际上对多目标进化算法的研究主要集中在对已有技术的改进和应用上，[31]，在理论研究方面的成果较少，概括起来主要有下列几个方面：

（1）Carlos M.Fonseca and Peter J.Fleming [32]提出了有关 Pareto 排序、非支配关系（集）以及共享参数等重要概念和方法。

（2）Horn 和 Nafpliotis 以及 Goldberg[13]提出了选择合适的共享参数的原则。

（3）Rudolph[33,34]和 Veldhuizen 等对 MOGA 的收敛特性进行了理论分析。

总的来说，目前国际上对进化计算的研究非常重视，每年都有多个国际会议（如 Congress on Evolutionary Computation，Genetic and Evolutionary Computation Conference 等等），从事多目标进化计算研究的学者也越来越多，每年都会涌现出不少有价值的研究成果。我国学者也比较重视进化计算的研究，如中国科技大学陈国良院士所带领的研究团队每年都有很多研究成果，武汉大学康立山主持了在我国召开的第一个国际进化计算的 Workshop，2004 年 10 月陈国良主持召开了“自然计算与应用国际研讨会”，中南大学蔡自兴和王勇在进化计算国际杂志 IEEE Transaction on Evolutionary Computation 等上发表论文多篇，等等。近年来，国内学术期刊上也出现了不少有价值的研究成果，如基于聚类的快速多目标遗传算法[35]，解约束多目标优化问题的一种鲁棒的进化算法[36]，一种求解高维优化问题的多目标遗传算法及其收敛性分析[37]，一种用于多目标优化的混合遗传算法[38]，基于多目标进化算法的混合 H-2/H−∞优化控制[39]，自动区域划分的分区域搜索狭义遗传算法[40]；以及各类应用研究成果，如文献[41,42,43,44,45,46,47,48]。

综上所述，目前国内国际上有关 MOEA 的研究非常活跃，进入了快速发展阶段，在多目标优化的方法中，还有很大的研究发展空间，还有很多问题需要进行进一步完善或改进。

第三章 基于 ELECTRE 法的多目标优化进化算法

实际的工程优化问题大多数是多目标优化问题，目标之间一般都是互相冲突的。多目标优化很早就得到了人们的重视，到目前已经发展了较多的求解多目标优化的方法。

一个多目标优化如果存在非劣解，往往存在无穷多个，形成非劣解集。在求解实际问题时，过多的非劣解是无法直接应用的。决策者只能选择令其最满意的一个非劣解作为最终解。求最终解主要有三类方法，一类是求非劣解的生成法，即先求出大量的非劣解，构成非劣解的一个子集，然后按照决策者的意图找出最终解，另一类为交互法，不先求出很多的非劣解，而是通过分析者与决策者对话的方式，逐步求出最终解。最后一类是事先要求决策者提供目标之间的相对重要程度，算法以此为依据，将多目标问题转化为单目标问题进行求解，该类方法也可以被认为是第一类方法的一个子方法，该类方法的难点在于，如何得到决策者真实的权重信息[49]。

相当部分 MOEAs 采取的排序策略是针对 Pareto 最优解实施进化,即进化个体只比较关于所有子目标的 Pareto 优劣关系,在当前种群中仅考虑属于 Pareto 最优解的个体。这种机制的优点在于进化结果与均衡面（trade-off surface）的凹凸性无关且无需任何事先偏好信息，但搜索空间的维数制约了这种方式的优化性能。因为当问题空间的维数高时，则群体中的各个个体很难进行 Pareto 排序比较，有时甚至出现所有个体皆为 Pareto 最优解，从而无法实施正常的进化选优[50,51,52]。在这种情况下则不能采用通常的 Pareto 优劣关系对个体进行优劣排序，必须引入启发式信息。本文将多目标属性决策方法中的 ELECTRE 法引入到算法中，采取一种比 Pareto 优劣关系弱的排序策略对个体进行排序。

3.1 ELECTRE 法介绍

ELECTRE 法(Elimination Et Choice Translation Reality)由 Benoyown 等提出后逐步得以完善[53]。ELECTRE 法的关键是利用了所谓的超序关系(outranking relationship)概念，也即当方案 R_k 和 R_l 没有明显的优劣之分时，决策者以一定的风险程度认为 R_k 优于 R_l,这样通过一系列的超序关系评估，就可以去掉超序关系意义下的劣方案.建立超序关系的方法有多种，这里介绍 Van Delft 和 Nijkamp 提出的用净优势值和净劣势值来建立超序关系。

首先,设 $R=(r_{ij})_{m\times n}$ 为有 m 个方案 n 个目标决策的决策矩阵，r_{ij} 表示第 i 个方案在第 j 个目标下经规范化后的属性值，R_i 为 n 维向量 $(r_{i1},r_{i2},...r_{in})$,即第 i 个方案在各目标下的属性值，R_i 也可表示第 i 个方案。目标权重向量

$W=(w_1,w_2,...w_n)$。

$C_{kl}=\{j\mid r_{kj}\ge r_{lj}\},D_{kl}=\{j\mid r_{kj}<r_{lj}\}$. C_{kl} 称为方案 R_k 对方案 R_l 的优势集，D_{kl} 称为 R_k 对方案 R_l 的劣势集,显然,劣势集为优势集的补集。

定义 3-1 $c_{kl}=\sum_{j\in C_{kl}} w_j$ 称为方案 R_k 相对方案 R_l 的优势指数（其中 $\sum_{j=1}^{n} w_j=1$）。

定义 3-2 $d_{kl}=\dfrac{\max_{j\in D_{kl}}|w_j(r_{kj}-r_{lj})|}{\max_{j\in J}|w_j(r_{kj}-r_{lj})|}$ 称为方案 R_k 相对与方案 R_l 的劣势指数。

上式中分子为劣势集中加权属性值之差的最大值,而分母为各目标中加权属性值之差的最大值。

定义 3-3 $c_k=\sum_{\substack{l=1\\l\ne k}}^{m} c_{kl}-\sum_{\substack{l=1\\l\ne k}}^{m} c_{lk}$ 称为方案 R_k 的净优势值。

定义 3-4 $d_k=\sum_{\substack{l=1\\l\ne k}}^{m} d_{kl}-\sum_{\substack{l=1\\l\ne k}}^{m} d_{lk}$ 称为方案 R_k 的净劣势值。

c_k 为方案 R_k 对其它方案的优势指数之和减去其它方案相对 R_k 的优势指数之和，反映了 R_k 在方案集中所处的优势程度.同理 d_k 反应了 R_k 在方案集中所处的劣势程度.若 c_k 越大且 d_k 越小则方案 R_k 在方案集中就越具有优势。因此定义超序关系“$\prec_n$”:

定义 3-5 若 $R_l\prec_n R_k$ 当且仅当 $c_l\ge c_k$ 且 $d_l\le d_k$，若 $R_l\sim R_k$ (R_l 与 R_k 无差别)，当且仅当 $c_l=c_k$ 且 $d_l=d_k$。若非 $R_l\prec_n R_k$ 且非 $R_k\prec_n R_l$，则称 R_k 与 R_l 不可比较。

性质 1 超序关系 $\prec_n$ (outranking relationship) 是一个偏序关系，即 $\forall a,b,c\in R(R为方案集)$，$\prec_n$ 满足如下性质：

(1) 自反性: $a\prec_n a$；

(2) 反对称性: $a\prec_n b,b\prec_n a$，$a=b$；

(3) 传递性: $a\prec_n b,b\prec_n c\Rightarrow a\prec_n c$；

定理1 设 I 为种群 R 的最小优势集, 即 $I=\{R_i\mid\neg\exists R_j\in R,使得R_j\prec_n R_i\}$；$P$ 为种群 R 非劣个体集, 即 $P=\{R_i\in R,使得R_j\prec R_i\}$, 则 $I\subseteq P$。

证明:假设 $I\,\text{Ø}\,P$, 那么, 至少存在一个个体 R_i, 满足: $R_i\in I且R_i\notin P$。根据 P 的定义，则必存在一个个体 $R_j,满足R_j\prec R_i$。依定义有 $c_{ji}>c_{ij},d_{ji}<d_{ij}$ 且 $\forall R_k\in R(k\ne i,j)$ $c_{jk}\ge c_{ik},d_{jk}\le d_{ik}$, $c_{kj}\ge c_{ki},d_{kj}\le d_{ki}$。

$$
\begin{aligned}
& c_j - c_i \\
&= (\sum_{\substack{k=1 \\ k\neq j}}^{m} c_{jk} - \sum_{\substack{k=1 \\ k\neq j}}^{m} c_{kj}) - (\sum_{\substack{k=1 \\ k\neq i}}^{m} c_{ik} - \sum_{\substack{k=1 \\ k\neq i}}^{m} c_{ki}) \\
&= \sum_{\substack{k=1 \\ k\neq j \\ k\neq i}}^{m} (c_{jk} - c_{ik}) + (c_{ji} - c_{ij}) + \sum_{\substack{k=1 \\ k\neq j \\ k\neq i}}^{m} (c_{ki} - c_{kj}) \\
&\quad + (c_{ki} - c_{kj}) \geq 0
\end{aligned}
$$

同理，有 $d_j - d_i \leq 0$，依定义 3-5，$R_j \prec_n R_i$，与 $R_i \in R_j$ 矛盾。证毕。

定理 1 说明超序关系 $\prec_n$ 比 Pareto 优劣关系弱，这样利用关系 $\prec_n$ 对种群中的个体进行排序，能在进化过程中增大选择压，加快进化过程的收敛。

3.2 群体的多样性和分布性

群体的多样性和分布性是衡量多目标优化进化算法的重要指标，因为多目标进化算法的收敛过程是非支配集不断逼近最优解集。目前，许多研究采用小生境技术（nich）实现来维持候选解集的多样性，如[13]。随机地从进化群体中选择两个个体 i 和 j，再随机地从进化群体中选取一个比较集 CS，然后用个体 i 和个体 j 分别与 CS 中的个体进行比较，如果其中之一受 CS 支配，另一个个体不受 CS 支配，那么非支配个体将被选中参入下一代进化。如果个体 i 和个体 j 都不受 CS 支配，则只能采用共享机制来选取其中之一参入下一代进化。将选择共享适应度大的（或小生境计数小的）个体进入下一代繁殖操作。选择适应度共享（fitness sharing）是实现群体多样性的有效方法。设个体 i 针对某个子目标的适应度为 fi，个体 i 的小生境计数为 mi，定义 fi/mi 为共享适应度。同一小生境内的个体互相降低对方的共享适应度。个体的聚集程度越高，其相对于适应度的共享适应度就被降的越多。使用共享适应度的目的在于将进化群体分散到整个搜索空间（search space）的不同峰值区域上。在经典 GA 中引入小生境技术的目的是为了防止早熟收敛。采用小生境技术的主要优点是算法的运行效率比较高，且能获得较好的 Pareto 最优边界（Pareto Optimal Front）。但当共享适应度与竞标赛选择相结合是可能出现混沌行为（chaotic behaviour），文献[]建议不断修改共享参数。它的最大问题是共享参数的选择没有一个通用的法则。为了避免共享参数的设置和不断调整，我们采用 k 近邻法来维持与增强群体的多样性与分布性。

3.3 算法描述

3.3.1 算法流程

我们将 ELECTRE 方法与基于种群进化思想相结合,对群体中个体依照 ELECTRE 方法进行排序,得到一类新的算法,具体流程描述如下:

Step1. 随机产生一个初始环境进化群体 P_0,同时初始化外部优势群体 P_0',并使之为空,令进化代数 t=0。

Step2. 执行环境选择(Environment Selection)操作,即用 ELECTRE 方法对 $(P_t + P_t')$ 中的个体进行排序,将产生的最小优势集拷贝至 P_{t+1}'。

Step3. 若 $|P_{t+1}'| > N'$,则对 P_{t+1} 采取某种截断策略,使其规模变为 N',同时使外部优势群体中的个体具有良好的分布性。

若 $|P_{t+1}'| < N'$,则从 $(P_t + P_t')$ 中选取 $(N' - |P_{t+1}|)$ 个个体,将其补充到 P_{t+1} 中。

Step4. 若终止条件满足,则结束并将 P_{t+1} 作为返回结果。

Step5. 对 P_{t+1}' 执行繁殖选择(Mating Selection)操作(如 binary tournament selection),选择优势个体,形成配对池,产生的 N 个个体赋给 P_{t+1}。

Step6. 进化繁殖进化群体 P_{t+1},即对群体执行交叉操作和变异操作。

Step7. 令 t=t+1,并转 Step2。

3.3.2 交叉算子

在 RCGA 中,广泛使用的交叉算子有模拟二进制交叉(SBX),单峰正态分布交叉(UNDX),单形交叉[54]等。本算法采用单形交叉作为惟一的重组算子,它基于均匀分布来产生后代个体且不需要个体的适应值信息。在 R^n 中,$n+1$ 个独立的父体向量($\vec{x}_i, i=1,\cdots,n+1$)行成了一个单形, 后代个体的产生按照以下步骤进行:1)按一定的比率将单形向各个方向($\vec{x}_i - \vec{o}$,这里 $\vec{o}$ 是 $n+1$ 个向量的中心即 $\vec{o} = \frac{1}{n+1}\sum_{i=1}^{n+1}\vec{x}_i$)进行扩张得到一个新的单形,2)从新的单形中选择一个点作为

后代个体。例如，我们考虑在两维空间中由三个点$\vec{x}_1, \vec{x}_2$和$\vec{x}_3$构成的一个单形。将这个单形以$(1+\varepsilon)$ $(\varepsilon \geq 0)$的比例向各个方向进行扩张，令$\vec{o}=\frac{1}{3}\sum_{i=1}^{3}\vec{x}_i$，$\vec{y}_i=(1+\varepsilon)(\vec{x}_i-\vec{o})$，则由$\vec{y}_1$，$\vec{y}_2$和$\vec{y}_3$形成了一个新的单形。在新单形中任取一点$\vec{z}$，则$\vec{z}=k_1\vec{y}_1+k_2\vec{y}_2+k_3\vec{y}_3+\vec{o}$，其中$k_1, k_2,$和$k_3$为[0, 1]中均匀分布的随机数且满足$k_1+k_2+k_3=1$。

使用单形交叉的原因是它的实现比较简单，同时它的计算复杂度仅为$O(N)$。图 1 表示由三个父体组成的单形所产生的后代个体分布。

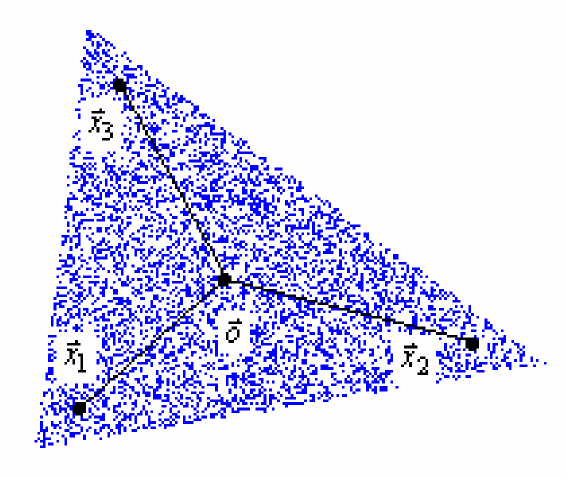

图 2-1 单形交叉产生的后代个体示意图

3.3.3 适应值分配及环境选择：

对于群体中的个体 i，其适应值的计算方法为：

$$F(i)=R(i)+D(i) \tag{3-1}$$

其中：$$R(i)=\sum_{j\in P+P', j\succ i} S(j) \tag{3-2}$$

$$S(i)=|\{j \mid j\in P+P' \wedge i\succ j\}| \tag{3-3}$$

$$D(i)=\frac{1}{\delta_i^k+2} \tag{3-4}$$

$$k=\sqrt{|P|+|P'|} \tag{3-5}$$

σ_i^k 为个体 i 到其 k 个邻近个体之间的距离，这里需要计算个体 i 到进化群体 P_t 和外部优势群体 P_t' 中其它所有个体之间的距离，并按增序排序。$S(i)$ 称为强度值，为群体中优超自己的个体的个数。$D(i)$ 中分母的值加 2 是为了确保 $D(i)$ 的值大于 0 而小于 1。

环境选择时，首先选择适应度小于 1 的个体进入外部优势群体 P_{t+1}' 中，即：

$$P_{t+1}'=\{i \mid i\in P_t+P_t', \wedge F(i)<1\} \tag{3-6}$$

此时，若 P_{t+1}' 中个体数少于约定值 N'，即 $|P_t'|<N'$，则在上一代群体 P_t 和 P_t' 中选择 $(N'-P_{t+1}')$ 个适应值小的优秀个体进入 P_{t+1}' 中。

若 $|P_t'|>N'$，则按下列方法(archive truncation procedure)依次选取个体 i 从 P_{t+1}' 中删除：

$$i \le_d j \Leftrightarrow \forall 0<k<|P_{t+1}'|:\sigma_i^k=\sigma_j^k \vee \tag{3-7}$$

$$0<k<|P_{t+1}'|:(\forall 0<l<k:\sigma_i^k=\sigma_j^k)\wedge \sigma_i^k<\sigma_j^k \tag{3-8}$$

直至 $|P_{t+1}'|=N'$。

3.4 实验分析

3.4.1 测试函数

为了测试算法的性能，使用两个典型的函数来对其检验。

测试函数 T1:

$$\min\ f_1(x)=1-\exp(-\sum_{i=1}^{3}(x_i-\frac{1}{\sqrt{3}})^2);$$

$$\min\ f_2(x)=1-\exp(-\sum_{i=1}^{3}(x_i+\frac{1}{\sqrt{3}})^2);$$

$$\text{s.t}\quad x_i\in[-4,4]\ \ i=1,2,3$$

这一函数由 Fonseca 和 Fleming[55]提出，T1 的 Pareto 最优解集为 $x_1=x_2=x_3\in[-1/\sqrt{3},1/\sqrt{3}]$。

测试函数 T2：

$$\min\ f_1(x)=0.5\times(x_1^{\ 2}+x_2^{\ 2})+\sin(x_1^{\ 2}+x_2^{\ 2});$$

$$\min\ f_2(x)=\frac{(3x_1-2x_2+4)^2}{8}+\frac{(x_1-x_2+1)^2}{27}+15;$$

$$\min\ f_3(x)=\frac{1}{x_1^{\ 2}+x_2^{\ 2}+1}-1.1\times e^{(-x_1^{\ 2}-x_2^{\ 2})};$$

$$\text{s.t}\quad x_1,x_2\in[-3,3]。$$

这个函数来自文献[56]有三个 Pareto 均衡面，它们在目标空间呈现为一条非线性非对称的三维螺旋曲线。

3.4.2 测试结果

新算法的实验在 Matlab 7.0 中完成。实验中，群体的规模为 100。交叉算子采用模拟二进制交叉，交叉概率取为 0.9，分布指数取为 20；变异算子采用非一致高斯变异算子，变异概率取为 0.1，方差为 0.5。对测试函数 A 和 B，目标权重均为 $W=(0.333,0.333,0.333)$。图 3-2 和图 3-3 分别为测试函数 A 和 B 的目标函数均衡面的演化结果，从图中可看出，非劣解较均匀地分布 Pareto 前沿上，说明算法具有较强的搜索能力，是可行的、有效的。

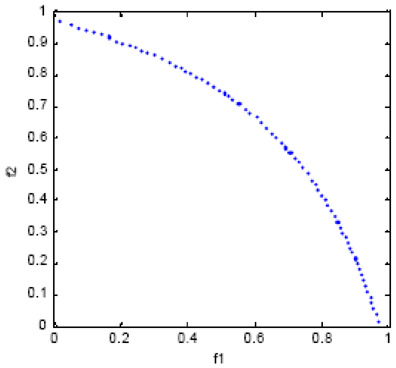

图 3-2 函数 T1 的 Pareto 曲线

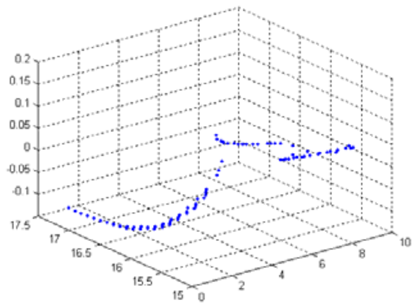

图 3-3 函数的 Pareto 曲线

3.5 小结

进化算法特别适合求解多目标优化问题，现有的多目标优化进化算法大多是基于 Pareto 优劣关系进行排序的，当目标空间维数增大时，会出现种群中多数个体都是非劣的情况，导致算法收敛速度下降。本章将 ELECTRE 法引入到进化算法中，构造出一种比 Pareto 优劣关系弱的超序关系对种群中的个体进行排序，实验结果证明了算法的可用性和有效性。

第四章 差异进化多目标优化算法

差异进化算法由 Storn 和 Price[4]提出，它是一种基于种群的随机搜索算法，对实数值、多模式目标函数优化具有很好的寻优效果。差异进化算法的特征是：简单，高效，采用浮点数编码而非二进制编码，对初始值无要求，收敛速度快，适合求解非线性问题。同其他进化算法类似，差异算法也是先随机产生初始种群，然后采用选择，变异交叉等操作，对种群进行不断的优化。

4.1 差异进化算法介绍

差异进化是一种并行直接搜索方法，在每一代 G 中，利用 NP 个 D 维参数向量作为一个种群

$x_{i,G} = 1, 2, \ldots, NP$

NP 在整个进化过程中一直保持不变。初始向量种群随机产生并且要覆盖到整个搜索空间，如无特别指出，通常假定所有的随机变量服从均匀概率分布，如果初始解是可获得的，那么产生初始种群时可在该解上加上服从标准正态分布产生的方差。差异进化中新参数的产生是将两个种群向量之差乘上一定的权重系数，加在第三个向量上，这个操作称之为变异。变异向量的参数同另一个预先确定的参数相混合（目标向量），来产生所谓的试验向量。参数混合同进化界中的交叉类似。在稍后的章节中，我们将进行详细的阐述。如果生成的试验向量的函数值小于目标向量，那么试验向量将取代目标向量进入下一代群体，最后的这步操作称为选择。种群中的每一个向量都必须作为目标向量，且仅有一次，因而，一次迭代中将有 N 次竞争。下面，我们将进行差异进化的具体阐述。

➢ 变异

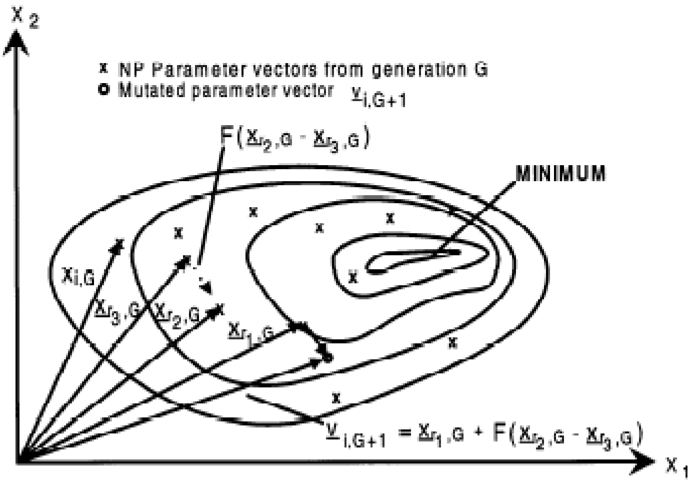

图 4-1 变异向量的产生[4]

对于每一个目标向量$x_{i,G}=1,2,...,NP$，变异向量依照如下公式产生：

$$v_{i,G+1}=x_{r_1,G}+F\cdot(x_{r_2,G}-x_{r_3,G}) \tag{4-1}$$

其中，r_1，r_2，$r_3\in\{1,2,...,NP\}$，为互不相同的随机整数；$F>0$。要满足随机产生的整数r_1，r_2，r_3互不相同这个条件，NP的值至少应为 4。F是一个实值常数，取值范围为[0，2]，用来控制差异的变化，即$x_{r_2,G}-x_{r_3,G}$的幅度变化。图 4-1 中描述了不同的参数向量在变异向量$v_{i,G+1}$产生的过程中所起的作用。

➢ 交叉

为了增加扰动参数向量的多样性，引入交叉操作，为此，生成试验向量（trial vector）；

$$u_{i,G+1}=(u_{1i,G+1},u_{2i,G+1},...,u_{Di,G+1}) \tag{4-2}$$

这里

$$u_{ji,G+1}=\begin{cases}v_{ji,G+1} & if\,(randb(j)\le CR)\,or\; j=rnbr(i)\\ x_{ji,G} & if\,(randb(j)>CR)\,or\; j\ne rnbr(i)\end{cases} \tag{4-3}$$

$$j=1,2,...,D$$

在公式 4-2 中，$randb(j)$是第 j 个服从均匀分布的随机数，取值在[0，1]之间。CR 为交叉常数，取值范围为[0，1]。$rnbr(i)$是在 1，2，…D 中随机选取的一个标志位，用以确保向量$u_{ji,G+1}$中至少有一个参数来自向量$v_{i,G+1}$。图 4-2 给出了一个 7 维向量交叉机制的示例。

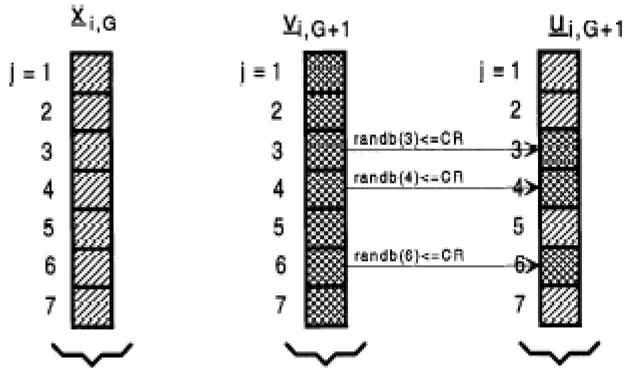

图 4-2 个体交叉示意图

➢ 选择

为了选择向量进入下一代群体（G+1 代），将试验向量$u_{i,G+1}$同目标向量$x_{i,G}$依照贪婪法则进行比较：如果向量$u_{i,G+1}$产生的函数值小于$x_{i,G}$所产生的函数值，则$u_{i,G+1}$进入下一代种群，即$x_{i,G+1}=u_{i,G+1}$，否则，$x_{i,G}$进入下一代种群，即$x_{i,G+1}=x_{i,G}$。

➢ 伪代码

图 4-3 所示的 C 伪代码充分说明了 DE 的简洁性。

➢ 其它形式的差异进化算法

上面所讲述的只是差异进化算法的一种形式，还有其它变种，也被证明是高效的。为了便于对这些算法进行分类，我们引入如下标识：

$$DE/\ x\ /\ y\ /\ z \tag{4-4}$$

x 指明当前变异的向量是在种群中随机选择（rand）还是选择种群中函数花费最小的向量（best）。

y 为差异向量的数目，一般为 1 或 2

z 为交叉策略的类型。

采用这种标识，上述的差异进化算法可表示为 DE/ rand / 1 / bin

其它形式的差异进化算法有：

DE/ best / 2 / bin ：

$$v_{i,G+1}=x_{best,G}+F\cdot(x_{r_1,G}+x_{r_2,G}-x_{r_3,G}-x_{r_4,G}) \tag{4-5}$$

DE/ rand / 2 / bin：

$$v_{i,G+1}=x_{r_1,G}+F\cdot(x_{r_2,G}+x_{r_3,G}-x_{r_4,G}-x_{r_5,G}) \tag{4-6}$$

DE/ current to best / 2 / bin：

$$v_{i,G+1} = x_{i,G} + F \cdot (x_{best,G} - x_{i,G}) + F(x_{r_1,G} - x_{r_2,G}) \tag{4-7}$$

```
/*----------------------------------------Main   loop-----------------------------------*/
While (count < gen_max)                    /*-Halt after gen_max generations. */
{
    For (i=0; i<NP; i++)      /*------------Start loop through population.---------*/
                              /***********Mutate/recombine***************/
        do a=rnd_uni( )*NP; while (a==i) ; /*Randomly pick 3 vectors,          */
        do b=rnd_uni( )*NP; while (b==i || b==a) ;/* all different form i        -*/
        do c=rnd_uni( )*NP; while (c==i || c==a || c==b);
        j=rnd_uni( )*D;          /*--- Randomly pick 3 vectors ---------------------*/
        for (k=1; k<=D; k++)
        {
           If (rand_uni( ) < CR || k==D)
           {
               trial[j]=x1[c][j]+F*(x1[a][j]-x1[b][j]);
           }
           else trial[j]=x1[i][j];
           j=(j+1)%D;
                                  /************Evaluate/select *************/

       score=evaluate(trial);
       if (score<=cost[i])
       {
          for (j=0; j<D; j++) x2[i][j]=trial[j];
          cost[i]=score;
       }
       else for (j=0; j<D; j++)    x2[i][j]=x1[i][j];
    }
                    /*--------------------End of population lop; swap arrays ------*

    for (i=0; i<NP; i++)
    {
       for (j=0; j<D; j++)    x1[i][j]=x2[i][j];
    }
    count++;
}                          /*            End of generation…increment counter. */
/*-------------------------------------End of main loop--------------------------------*/
```

图 4-3 DE 的 C 伪代码

4.2 相关研究

将差异进化扩展到多目标优化算法中，国内外已有学者做了相关方面的工作，这些算法大都采用非劣排序准则来进行后代的繁殖，利用距离尺度（distance metric）来避免拥挤，下面将对其中的一些重点算法进行简要介绍：

Abbass 于 2002 年提出 PDE 算法[58]，它采用单一种群策略，繁殖操作只在非劣解中进行，后代个体只有优于父代个体才能进入下一代种群。采用距离比较准则来维持多样性。

Madavan 于 2002 年提出 PDEA 算法[59]，它将稳健高效的差异进化算法同 NSGA-II 算法相融合。

Parsopoulus 于 2004 年提出 VEDE 算法[60]，它是一种并行的多种群差异进化算法，该算法受向量评估遗传算法（VEGA）启发而来。

Kukkonnen 在[61]中描述了一种通用的求解约束多目标优化的差异进化算法。

Iorio 将 NSGA-II 算法做了小的变动[62]，将其中的实数编码交叉算子和变异概率用差异进化机制代替。

Robic 于 2005 年提出 DEMO[63]，将 DE 的优势与基于 Pareto 分级机制和拥挤程度排序向结合起来，在 DEMO 中，新产生的候选个体，将马上被随后产生的候选个体替代。

4.3 具体算法描述

```
Begin
   G=0
   Create a random initial population P
   Evaluate each vector
   For G=1 to MAX _GEN Do
      For i = 1 to NP Do
         Select three distinct individuals randomly
         Perform recombination using DE scheme
         Perform mutation
         Perform selection
      End For
      Identify non-dominated solutions in population
      Add non-dominatd solutions into the archive population
   End For
   G=G+1
End
```

图 4-4 ZCDE 的算法流程

算法的伪代码在图 4-4 种给出，我们将算法简称为（ZCDE）。在算法中，我们采用两个种群，进化种群 $P(t)$ 和外部种群 $E(t)$（这里 t 为迭代次数）在外部种群中，我们采用 ε-支配原则来存储进化过程中的非劣个体以及维持非劣个体分布的均匀性。

同其它进化算法类似，ZCDE 算法采用实数编码，此外，将约束处理机制融入到算法中，该约束处理机制允许可行解介入到重组之中，这对于求解高度约束多目标优化问题是一个很好的途径。

在差异进化中没有明确的变异操作算子，因为在某种程度上来说，该操作被包含在重组操作中。然而，在多目标优化问题中，我们通过试验研究发现，为了更好的对搜索区域进行勘探，增加变异操作相当之必要，为此，我们采用均匀变异操作算子。

4.3.1 个体比较

子代个体产生后，要同相应的父代个体进行比较，以此来选择个体进入下一代的遗传进化操作。算法采用如下的准则来进行父代个体与子代个体之间的比较：

对于无约束的多目标优化问题，采用如下的三条准则：

1. 父代个体优于子代个体，父代个体被选择。
2. 子代个体优于父代个体，子代个体被选择。
3. 父代个体与子代个体非劣，选择的概率各为 0.5。

对于带约束条件的多目标优化问题，由于要考虑到个体的可行性，个体之间的比较变得比较复杂。Deb 等人在 NSGA-II 中定义了一个约束主导准则：一个解 xi 约束主导另一个解，当且仅当下列条件满足，

1. 解 x_i 是可行解而解 x_j 是不可行解。
2. 解 x_i 与解 x_j 都不是可行解，但 x_i 违反约束条件的程度小于解。
3. 解 x_i 与解 x_j 都是可行解且 x_i 主导 x_j。

采用 Deb 的比较准则，当约束条件不是很强时，能很快地收敛到可行解，因为在此准则中，可行解永远优于不可行解，使得在搜索过程中，种群能快速的向可行区域靠近。但对于约束程度较高时，.该准则的缺陷是难以发挥不可行解的作用,在群体中不能维持好的可行解与不可行解比例 。为此，将 ε-水平比较准则引入到算法当中。

➢ **ε-水平比较准则**

我们引入 $\phi(x)$ 来描述搜索点 x 违反约束条件的程度， $\phi(x)$ 应满足如下条件：

$$\begin{cases}\phi(x)=0\ (x\in F)\\ \phi(x)>0\ (x\notin F)\end{cases} \tag{4-9}$$

在罚函数法中，约束违反程度一般被当作为一个惩罚量，：$\phi(x)$ 可被定义如下：

$$\phi(x)=\max\{\max_j\{0,g_j(x)\},\max_j\left|h_j(x)\right|\} \tag{4-10}$$

$$\phi(x)=\sum_j\max\{0,g_j(x)\}^P+\sum_j\left|h_j(x)\right|^P \tag{4-11}$$

这里 P 为正的实数。

ε 水平比较[64]是定义在集合 $(f(x),\phi(x))$ 上的次序关系。采用 $\phi(x)$ 优先于 $f(x)$ 的字典排序方法，因为对于带约束的问题，可行性相较目标函数值而言，更加重要。

假设 $f_1(f_2)$ 和 $\phi_1(\phi_2)$ 分别为点 $x_1(x_2)$ 函数值和违反约束条件违反值，那么，对于 $\varepsilon\geq 0$，ε 水平比较 $<_\varepsilon$ 和 $\leq_\varepsilon$ 可被定义如下：

$$(f_1,\phi_1)<_\varepsilon(f_2,\phi_2)\Leftrightarrow\begin{cases}f_1<f_2, if\ \phi_1,\phi_2\leq\varepsilon\\ f_1<f_2, if\ \phi_1=\phi_2\\ \phi_1<\phi_2, otherwise\end{cases} \tag{4-12}$$

$$(f_1,\phi_1)\leq_\varepsilon(f_2,\phi_2)\Leftrightarrow\begin{cases}f_1\leq f_2, if\ \phi_1,\phi_2\leq\varepsilon\\ f_1\leq f_2, if\ \phi_1=\phi_2\\ \phi_1\leq\phi_2, otherwise\end{cases} \tag{4-13}$$

当 ε=0 时，$<_0$ 和 $\leq_0$ 等价于在字典排序时，$\phi(x)$ 优于 $f(x)$；同理，当 $\varepsilon=\infty$ 时，等价于一般意义下所定义的函数值之间的比较，$<$ 和 $\leq$

将单目标下的 ε 水平比较引入到多目标优化问题中，需作小的修改，因为对目标函数进行比较不是单个值的比较，而是向量之间的比较：

$$(f_1,\phi_1)\prec_\varepsilon(f_2,\phi_2)\Leftrightarrow\begin{cases}f_1\prec f_2, if\ \phi_1,\phi_2\leq\varepsilon\\ f_1\prec f_2, if\ \phi_1=\phi_2\\ \phi_1<\phi_2, otherwise\end{cases} \tag{4-14}$$

这里，f_1，f_2 为目标函数向量。

➢ **ε 约束方法的性质：**

求解优化问题时，将普通的比较准则用 ε 水平比较替换，即为 ε 约束方法，可

定义如下：

$$(\mathrm{P}\leq_{\varepsilon})\ \min imize \leq_{\varepsilon}\ f(x) \tag{4-15}$$

$\leq_{\varepsilon}$表明最小化问题是基于$\leq_{\varepsilon}$比较准则进行的，$\mathrm{P}\leq_{\varepsilon}$表明问题 P 的约束条件违反程度由$\phi(x)=0$松弛为$\phi(x)\leq\varepsilon$，约束条件降低。

$$\begin{aligned}(\mathrm{P}\leq_{\varepsilon})\ &\min imize \leq_{\varepsilon}\ f(x)\\ &\text{subject to }\ \phi(x)\leq\varepsilon\end{aligned} \tag{4-16}$$

从上式可以看出，采用ε水平比较，一个约束优化问题可以转换为相应的无约束优化问题。将ε水平比较准则同现有的无约束优化方法结合起来，可为求解约束优化问题提供新的途径。在求解无约束优化问题的相关算法中，将通常的比较准则用ε水平比较替换，即可利用算法来求解约束优化问题。需要注意的是，当ε变为 0 时，就可得到原问题的最优解。类似于在罚函数方法中，将惩罚系数增加到无穷大。

ε的控制

一般情况下，ε的值不需要特别的控制，很多约束优化问题可直接按照当ε等于 0 时字典排序求解。但当问题中存在等式约束条件时，为了获取高质量的解，对ε的值必须进行控制。关于ε的控制有多种方法，我们采用如下一种简单的方法来对其进行控制：

$$\varepsilon(0)=\phi(x_{\theta}) \tag{4-17}$$

$$\varepsilon(t)=\begin{cases}\varepsilon(0)(1-\dfrac{t}{T_c})^{cp}, 0<t<T_c\\ 0, \qquad\qquad t\geq T_c\end{cases} \tag{4-18}$$

x_{θ}为第θ个个体违反约束条件的程度（排序后），$\theta=0.2N$。ε值不断更新，直到迭代次数达到所需要控制的代数T_c。当迭代次数超过T_c时，ε得值被置为0，从而来获得违反约束条件程度最小的解。

4.3.2 档案群体的更新

这里我们引入一种基于ε-dominance 原理的多目标优化进化算法。搜索空间被划分为许多网格，每个网格只能被一个个体所占用，以此来维持解的多样性。虽然与 PAES 及其后继的改进算法中多样性的方法类似，但ε-dominance 更具通用性，在算法中，采用两个种群，一个进化种群$P(t)$，一个档案种群

$E(t)$.$P(t)$中产生的非劣个体同档案群体中的每一个个体进行比较，档案中的每一个个体都有一个坐标向量$B=(B_1,B_2,...B_m)^T$

$$B_j(f)=\begin{cases}\lfloor (f_j-f_j^{\min})/\varepsilon_j \rfloor, & for\ \text{minimizing}\ f_j \\ \lceil (f_j-f_j^{\min})/\varepsilon_j \rceil, & for\ \text{maximizing}\ f_j \end{cases} \tag{4-19}$$

m为目标个数。$f_j^{\min}$是第j个目标向量的最小可能值，ε_j表示用户对第j个目标所能容忍的最小阈值，低于此值的两个目标对用户来说没有显著意义。ε_j的值同ε-dominance 中定义的值一样。如果在档案中存在个体 a，其B向量优超子代个体c_i的B向量，即子代个体被档案中的个体优超，将此个体淘汰，反之，如果子代个体c_i的B向量优超档案种群中个体 a 的B向量，将 a 从档案中删除，将子代个体添加到档案种群中。如果上述两种情况都不发生，则意味着子代个体同档案中的个体ε-非劣。我们将此情况又分为两种：如果c_i同档案中的个体有相同的B向量值，比较两个个体到B向量之间的距离，如c_i离B近，则保留c_i而将另一个体踢出；如果c_i同档案中的个体不共用B向量，则将c_i加入到档案群体中。值得注意的是，在ε-MOEA 中，当c_i同档案中的个体有相同的B向量值时，先检查两个体是否非劣，然后再计算比较两个个体到B向量之间的距离。我们可以证明，在本算法中所采取的方法和ε-MOEA 中的方法是等价的，但显然我们所采取的方法更加简洁 。

4.4 实验结果及其比较

为了验证算法的有效性，我们将它与经典的多目标进化算法 NSGA2 和ε-MOEA 进行实验比较，我们将在相同的实验环境下，选用标准的测试函数对这三个算法进行评价、比较。

4.4.1 评价方法：

评价一个多目标进化算法的性能可以从两个方面去考虑：

1、收敛性：评价所求解与非劣最优解趋近程度。

2、分布性：评价所求解在目标空间分布是否均匀。

针对上述两个标准，我们分别采取一种评价方法去比较三个算法的相关性能。

Generational distance(GD): 该方法介绍于[65]中，用来估计算法的最终解集与全局非劣最优区域的趋近程度，计算如下：

$$GD=\sqrt{\sum_{i=1}^{n} d_i^2}\Big/ n \tag{4-20}$$

n 是解集中个体的数目，d_i 是每个个体到全局非劣最优解的最小欧几里德距离。GD 的值越小就说明解集越靠近全局非劣最优区域，如果 GD =0 就说明算法的解都在全局非劣最优区域上，这是最理想的情况。

Spacing （SP）：Schott 提出该方法通过计算解集中的每个个体与邻居个体的距离变化来评价解集在目标空间的分布情况[66]，其函数定义如下：

$$SP = \sqrt{1/(n-1) * \sum_{i=1}^{n} (\bar{d} - d_i)^2} \tag{4-21}$$

$d_i = \min_j \left(\left| f_1^i(\vec{x}) - f_1^j(\vec{x}) \right| + \left| f_2^i(\vec{x}) - f_2^j(\vec{x}) \right| \right)$，$i, j = 1, \ldots n, n$ 是解集中个体的数目，$\bar{d}$ 是所有 d_i 的平均值。如果 SP =0 说明解集中所有个体之间的距离都相等，分布均匀，SP 的值越小说明解集分布越均匀。

4.4.2 测试函数及实验结果

测试函数 ZDT1

$$\min f_1(x) = x_1$$

$$\min f_2(x) = g(x)(1 - \sqrt{x_1 / g(x)})$$

$$g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$$

subject to $0 \le x_i \le 1, i = 1, 2, \ldots 30$

ZDT1 具有一个连续的凸的 Pareto 前沿，图 4-5 是三个算法的求解结果。

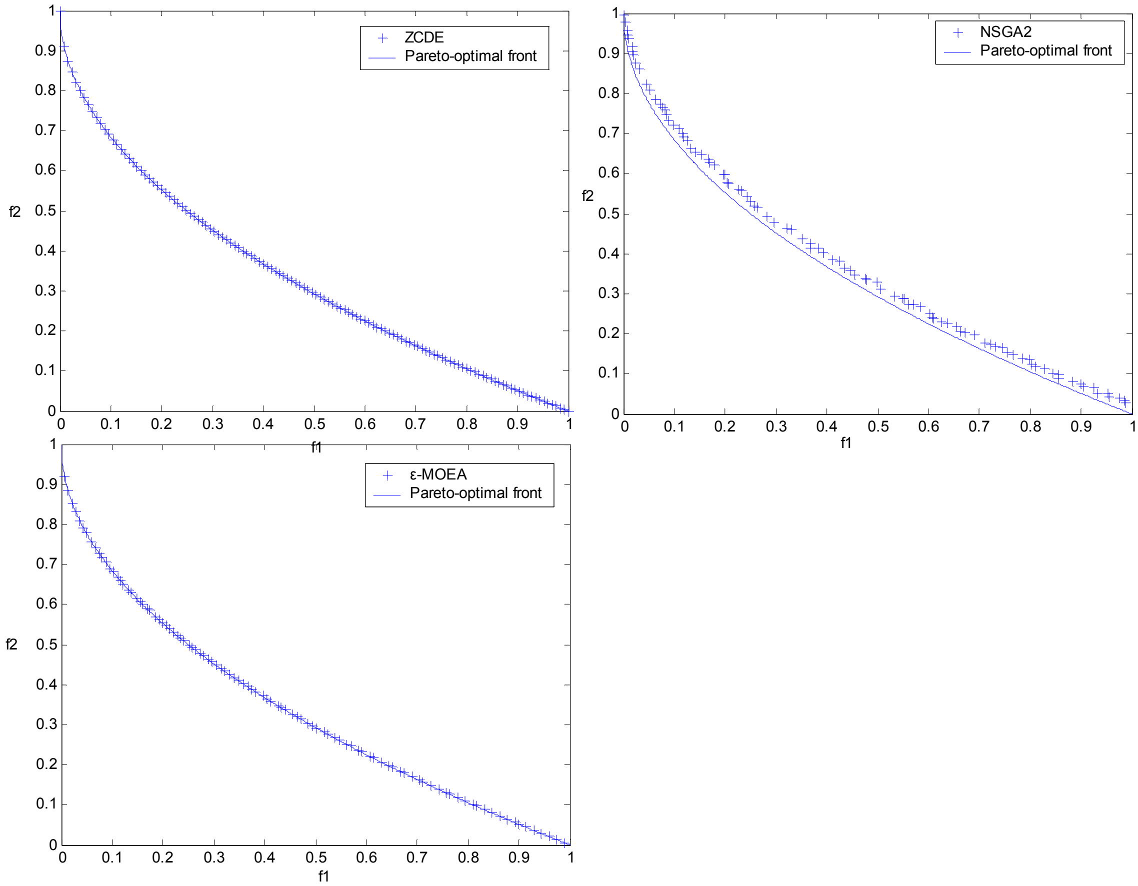

图 4-5 测试函数 ZDT1 解的分布

表 4-1 GD 的比较结果（ZDT1）

GD	Best	Average	Std.Dev
NSGA-II	0.00725	0.00468	0.01452
ε-MOEA	0.00245	0.00393	0.00124
ZCDE	0.00208	**0.00316**	0.00108

表 4-2 SP 的比较结果（ZDT1）

SP	Best	Average	Std.Dev
NSGA-II	0.00627	0.00835	2.6674E-4
ε-MOEA	0.00246	0.00471	1.5586E-4
ZCDE	0.00192	**0.00342**	1.3365E-4

从 ZDT1 上的试验结果可以发现，ZCDE 在收敛性和分布性方面，均好于其它两个算法。

测试问题 ZDT3

$$\min\ f_1(x) = x_1$$

$$\min\ f_2(x) = g(x)(1 - \sqrt{x_1 / g(x)} - \sin(10\pi x_1) x_1 / g(x))$$

$$g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$$

subject to $0 \le x_i \le 1, i = 1, 2, \ldots 30$

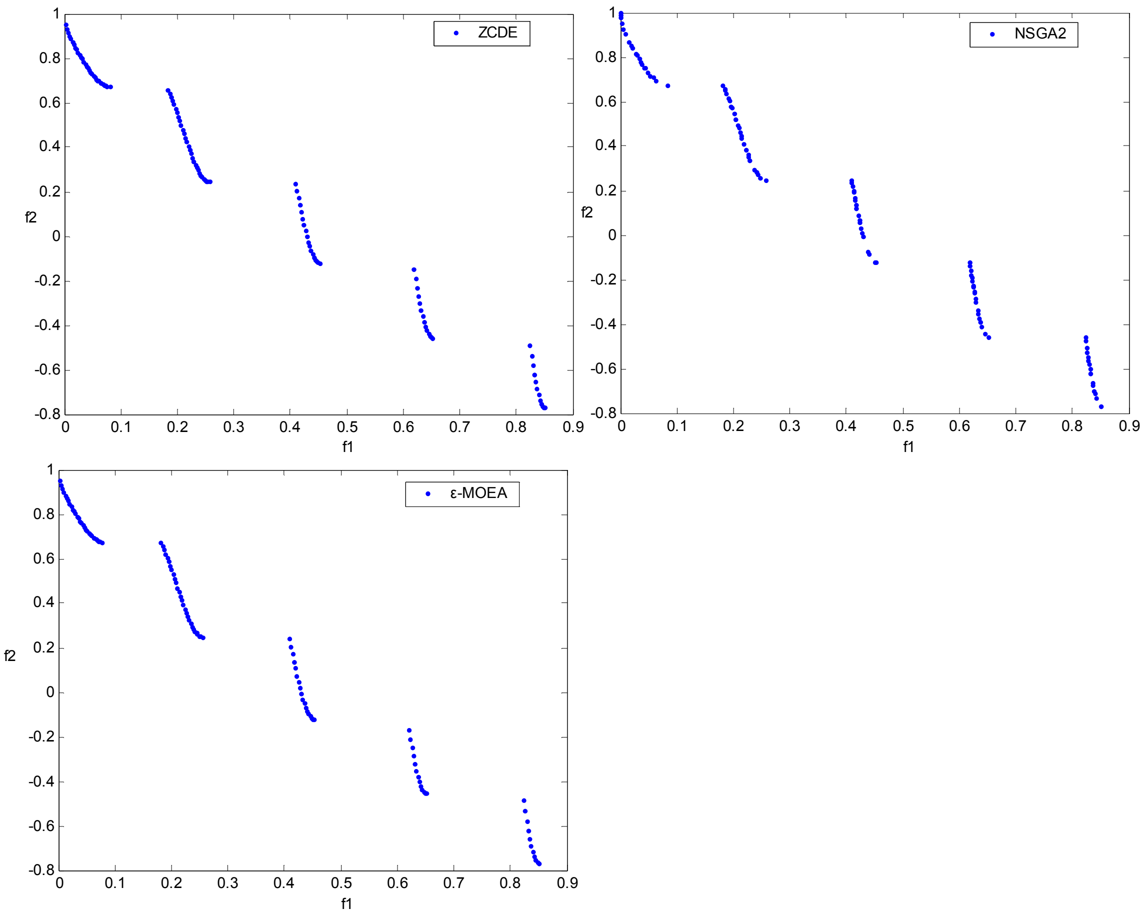

图 4-6 测试函数 ZDT3 解的分布

表 4-3 GD 的比较结果（ZDT3）

GD	Best	Average	Std.Dev
NSGA-II	0.00340	0.00369	1.5437E-4
ε-MOEA	0.00427	0.00487	1.8573E-4
ZCDE	0.00358	**0.00373**	1.1225E-4

表 4-4　SP 的比较结果（ZDT3）

SP	Best	Average	Std.Dev
NSGA-II	0.00405	0.00478	1.2136E-4
ε-MOEA	0.00157	0.00241	1.1879E-4
ZCDE	0.00138	**0.00159**	1.0458E-4

从 ZDT3 的实验结果可以看出，和 ZDT1 类似，ZCDE 在收敛性及分布性方面，均优于其它两个算法。

测试问题 ZDT6

$$\min f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$$

$$\min f_2(x) = g(x)(1 - (f_1(x)/g(x))^2)$$

$$g(x) = 1 + 9\left[\sum_{i=2}^{n} x_i/(n-1)\right]^{0.25}$$

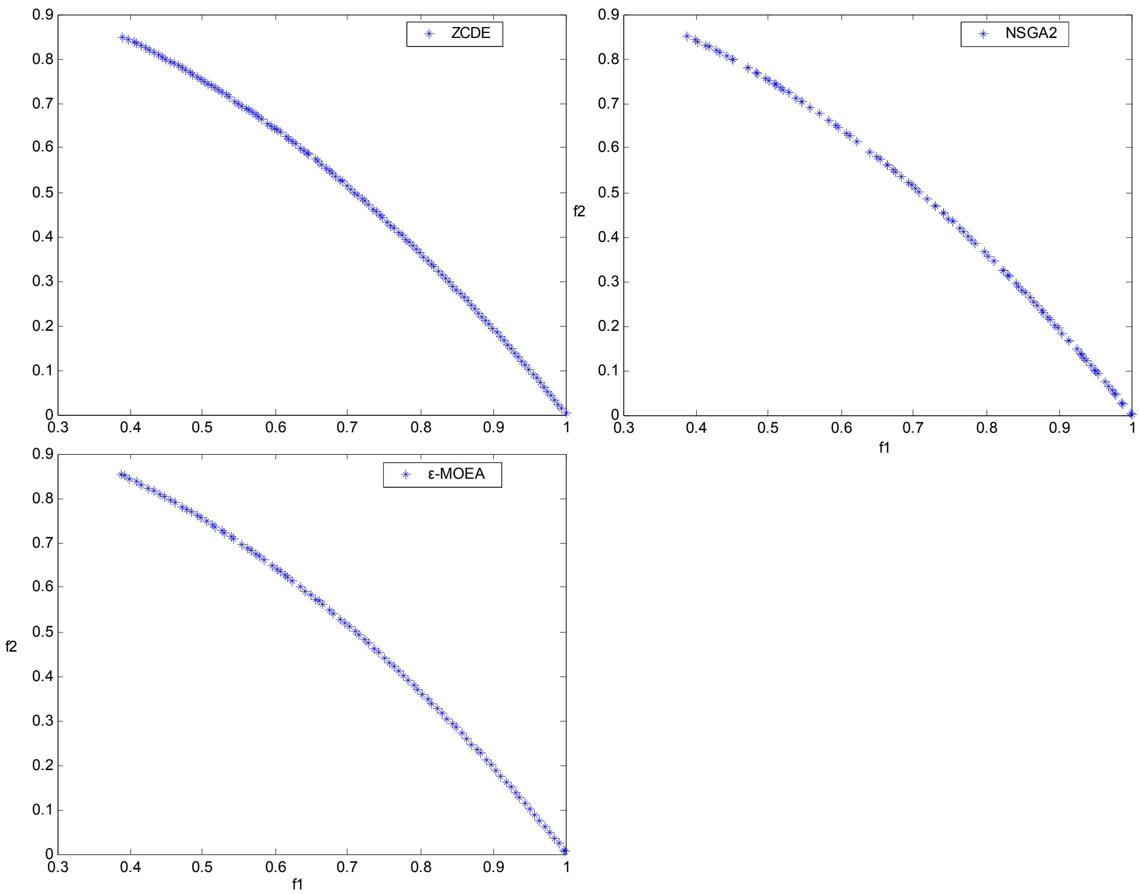

图 4-6　测试函数 ZDT6 解的分布

表 4-5　GD 的比较结果（ZDT6）

GD	Best	Average	Std.Dev
NSGA-II	0.00346	0.00373	9.8571E-5
ε-MOEA	0.00362	0.00395	8.3765E-5
ZCDE	0.00324	**0.00348**	9.1642E-5

表 4-6　SP 的比较结果（ZDT6）

SP	Best	Average	Std.Dev
NSGA-II	0.00213	0.01423	1.7382E-4
ε-MOEA	0.00152	0.00265	0.1009
ZCDE	0.00139	**0.00341**	0.0946

从实验结果可以看出，ZCDE 在收敛性和分布性上要优于 NSGA2 和 ε-MOEA。

测试问题 OSY

$$\min f_1(x) = -(25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 + (x_5-1)^2)$$

$$\min f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$$

subject to:

$$c_1(x) = x_1 + x_2 - 2 \geq 0$$
$$c_2(x) = 6 - x_1 - x_2 \geq 0$$
$$c_3(x) = 2 - x_2 + x_1 \geq 0$$
$$c_4(x) = 2 - x_1 + 3x_2 \geq 0$$
$$c_5(x) = 4 - (x_3-3)^2 - x_4 \geq 0$$
$$c_6(x) = (x_5-3)^2 + x_6 - 4 \geq 0$$
$$0 \leq x_1, x_2, x_6 \leq 10, 1 \leq x_3, x_5 \leq 5, 0 \leq x_4 \leq 6$$

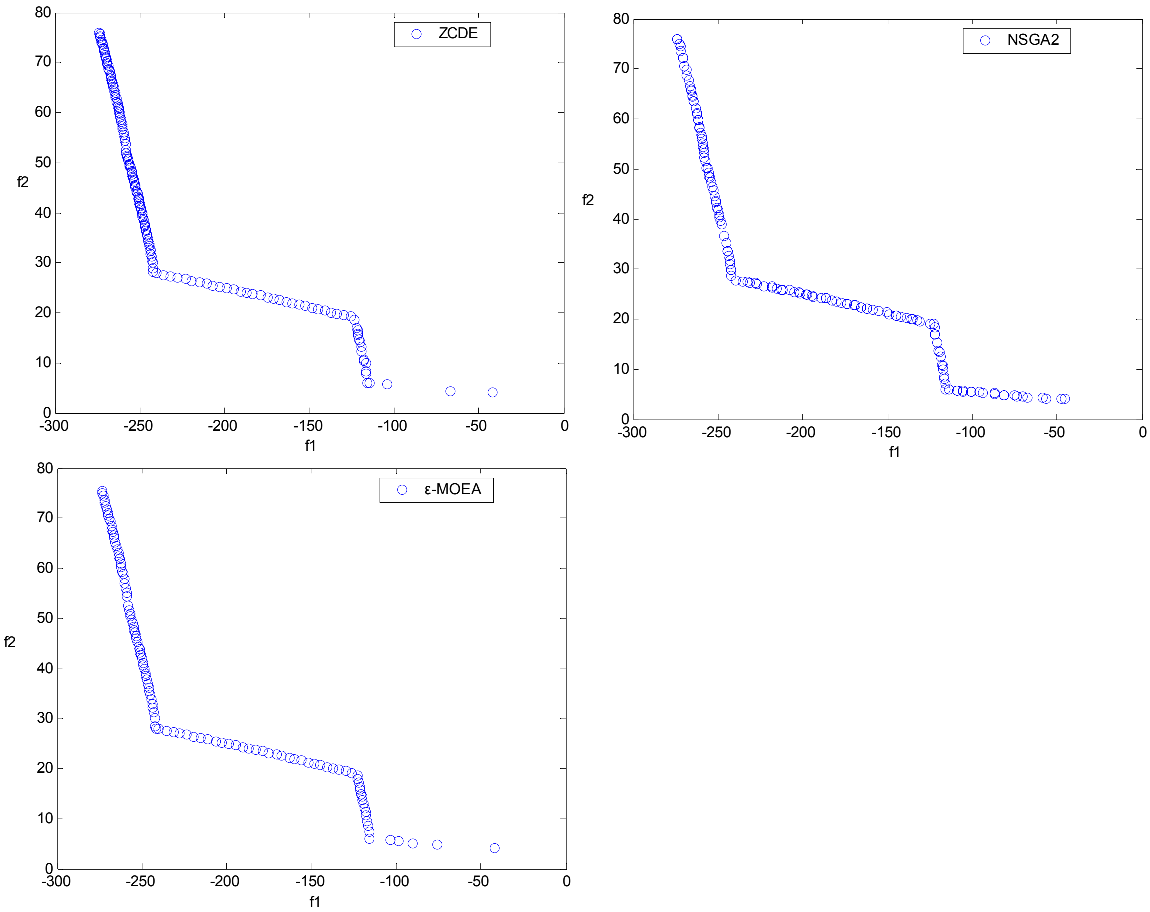

图 4-7 测试函数 OSY 解的分布

表 4-7　GD 的比较结果（OSY）

GD	Best	Average	Std.Dev
NSGA-II	0.00196	0.00283	4.2983E-4
ε-MOEA	0.00187	0.00275	1.4237E-4
ZCDE	0.00175	**0.00249**	1.1975E-4

表 4-8　SP 的比较结果（OSY）

SP	Best	Average	Std.Dev
NSGA-II	0.00289	**0.00359**	1.0343E-4
ε-MOEA	0.00312	0.00382	1.7588E-4
ZCDE	0.00317	0.00386	1.8739E-4

在这个测试函数中，ZCDE 的收敛性，要优于 NSGA2 和ε-MOEA，但其分布性较 NSGA2 要差些，主要原因是采用ε-domiance 比较时，容易丢失边界上的一些解。

4.5 小结

本章首先对差异进化算法进行了详细介绍，接着，将差异进化算法引入到多目标优化中，并对相关的研究工作进行了回顾。在个体比较时，将约束优化中的ε-水平比较准则扩展到多目标优化中，采用ε-MOEA 中的方法来保持群体的多样性，并对其进行了改进。最后通过对一些标准测试函数的实验，证明算法既能很好地收敛到 Pareto 前沿又能保持较好的分布性。

第五章 多目标优化进化算法求解约束优化问题

5.1 约束优化问题介绍

约束优化问题(Constrained Optimization Problems， COPs)是工程应用领域经常会遇到的一类数学规划问题.不失一般性，一个非线性规划问题(nonlinear programming， NLP)可描述为：

minimize：$f(\vec{x}) \qquad \vec{x}=(x_1,x_2,\cdots,x_n)\in\Re^n$ (5-1)

subject to：$g_j(\vec{x})\le 0,\quad j=1,\cdots,l$ (5-2)

$h_j(\vec{x})=0,\quad j=l+1,\cdots,m$ (5-3)

这里，$\vec{x}\in\Omega\subseteq S$ 为决策向量，Ω 为可行域，S 为决策空间.一般地，S 为 $\Re^n$ 中的 n 维长方体：$l(i)\le x_i\le u(i)$，$l(i)$，$u(i)$ 为常数， $i=1,\cdots,n$. $f(\vec{x})$，$g_j(\vec{x})$，$h_j(\vec{x})$ 均为 $\Re^n$ 上的 n 元函数， $f(\vec{x})$ 为目标函数，$g_j(\vec{x})\le 0$ 为第 j 个不等式约束条件，$h_j(\vec{x})=0$ 为第 j 个等式约束条件.l 表示不等式约束条件的个数，$(m-l)$ 表示等式约束条件的个数。

定义 4-1. Ω 为问题(1)的可行域(feasible region)当且仅当 $\Omega=\{\vec{x}\,|\,g_j(\vec{x})\le 0, j=1,\cdots,l; h_j(\vec{x})=0, j=l+1,\cdots m, \vec{x}\in S\}$。

Ω 在 S 上的补集即为问题(1)的不可行域，可行域内的点称为可行解，不可行域内的点称为不可行解.如果任一不等式约束条件满足 $g_j(\vec{x})=0$，则称 $g_j(\vec{x})$ 在 $\vec{x}$ 处活跃(active).显然，所有的等式约束条件 $h_j(\vec{x})$ ($j=l+1,\cdots m$)对于可行域 Ω 中的任意点均活跃。

传统的优化方法求解这类问题时都是基于梯度的信息，只适用于目标函数和约束条件可微的情形，而且求得的解多为局部最优解.进化算法(Evolutionary Algorithms， EAs)是一种模拟自然进化过程的全局优化方法，它通过选择、交叉、变异等机制来提高个体的适应性.同传统优化方法相比，进化算法更适合于求解约束优化问题.近十年来，利用进化算法求解约束优化问题已有许多学者进行了广泛的研究，并且提出了大量的约束优化进化算法(Constrained Optimization Evolutionary Algo- rithms， COEAs).文献[22]公布了 13 个相关的标准测试函数，数值实验结果表明，现有的 COEAs 都不能很好的处理各种类型的约束优化问题，如等式约束、不等式约束、混合约束等。

本章首先将约束优化问题转换为多目标优化问题并利用多目标优化技术来处理，接着提出了一个基于群体的算法发生器模型与一个不可行解存档和替换机制.实验结果表明，新算法不仅能搜索到所有测试函数的最优解，而且其收敛稳定性显著优于其他算法。

5.2 约束处理技术

Michalewicz 等和 Coello 对当前较为流行的约束处理技术进行了广泛的调查，并分别将它们划分为四类和五类。

惩罚函数法是进化算法处理约束优化问题的常用方法，其本质是容许群体中的个体在一定程度上违反约束条件，但必须对该个体依其违反约束条件的程度进行惩罚以减小它被选择的概率.个体违反约束条件的程度由惩罚函数来确定，大多数方法都采用如下途径构造惩罚函数。

令

$$G_j(\vec{x})=\begin{cases}\max\{0,g_j(\vec{x})\} & 1\le j\le l\\ |h_j(\vec{x})| & l+1\le j\le m\end{cases} \tag{5-4}$$

表示个体 $\vec{x}$ 与第 j 个约束条件的距离，则

$$G(\vec{x})=\sum_{j=1}^{m}G_j(\vec{x}) \tag{5-5}$$

表示个体 $\vec{x}$ 到可行域的距离，也反映了个体 $\vec{x}$ 违法约束条件的程度。

惩罚函数法分为：静态惩罚法、动态惩罚法、退火惩罚法、自适应惩罚法、协同惩罚法、死惩罚法等.一般来说，自适应惩罚函数法[23]具有较好的优化效果，因为它能利用搜索过程中的反馈信息动态的调节参数.最近，Farmani 等[24]提出了一种自适应适应值表示法，该方法将惩罚分为两个阶段进行，但缺点是算法的计算成本很高.惩罚函数法的主要缺点是罚参数的选取比较困难，而且算法的性能强烈地依赖于参数的选取.同时文献[25]指出即使最具动态性的惩罚函数法，对于无约束全局最优解与具有约束条件时的全局最优解相距很远的问题，其优化效果不可能很好。

同态映射法通过在立方体$[-1,1]^n$与可行域Ω之间建立映射关系，将约束优化问题转化为无约束优化问题[26]。其缺点是当可行域非凸时算法的实现较为复杂，而且算法需要初始可行解。

近年来，将约束优化问题转换为多目标优化问题来处理受到了极大的重视.这类算法的主要特点是将约束条件作为一个或多个目标看待。基于上述思想的算法可分为两类：区分可行解与不可行解法和多目标优化法。

区分可行解与不可行解法通常将约束优化问题重新定义为具有两个目标的多目标优化问题，一般来说，其中一个目标为原问题的目标函数 $f(\vec{x})$，另一个目标为个体违反约束条件的程度$G(\vec{x})$。几种典型的算法有：Deb 提出了一种不需要任何罚参数的联赛选择算子，并采用三个准则来比较成对的个体。该选择算子的缺陷是难以发挥不可行解的作用，在群体中不能维持好的可行解与不可

行解比例。在提出的算法基础上，文献[26]认为在群体中保持一定比例的接近可行域边界的不可行解，对于找到全局最优解很有帮助，并提出了一种新的个体比较准则。实验表明，改进后的算法非常适合于求解全局最优解位于可行域边界的一类约束优化问题，但参数ε的合理选取通常比较困难，而且是依赖于问题的.文献[22]通过定义概率p_f来协调上述两个目标，并采用随机排序法选择个体.但实验表明该算法对参数p_f的选取比较敏感。

多目标优化法的主要思想是将约束优化问题转换为多目标优化问题后，利用多目标优化技术处理。在这类算法中，约束处理技术常常与多目标优化中的以下几种机制相融合来选择个体：1)将 Pareto 优超关系作为个体选择准则；2)通过 Pareto 排序定义个体的等级并使可行解具有比不可行解更高的等级值；3)将主群体划分为若干个子群体，其中一个子群体将目标函数作为适应值函数来评价个体，其余的子群体将相应的约束条件作为适应值函数来评价个体。

其中几种典型的算法有：文献[67]提出了一种类似于向量评估遗传算法的群体多目标技术，文献[68]提出了一种基于 Pareto 排序过程的方法，文献[69]提出了一种基于优超关系的联赛选择方法，文献[70]提出了一种基于 Pareto 存档进化策略的方法。为了评估基于多目标优化技术的约束处理方法的有效性，Mezura 等[71]对其中四个较好的算法进行了实验比较.结果表明，作为比较个体的方法，Pareto 优超关系具有比 Pareto 排序和群体法更好的优化性能.此外，文献[71]得到的一个重要的结论是必须引进额外的机制来改进这类算法的有效性.同时值得注意的是，上述方法大都是对多目标进化算法(Multi-objective Evolutionary Algorithms， MOEAs)的简单模仿和借鉴，尽管已经将约束优化问题转换为多目标优化问题，但其在与一般的多目标优化问题的求解上仍然具有本质的区别.文献[72]利用向量之间的 Pareto 优于关系，在群体中通过定义个体 Pareto 强度值指标对个体进行排序.虽然通过 Pareto 强度值可以很方便地比较个体优劣，但 Pareto 强度值具有很强的“欺骗性”.假设群体仅包含 5 个个体，它们所对应的适应值向量分别为(2，0)，(1，0.1)，(1.1，0.2)，(1.2，0.3)和(1.3，0.4).根据文献[72]，第二个个体具有最大的 Pareto 强度值.若第一个个体即为全局最优解，则采用上述准则选择个体时便出现了失误。

5.3 基于多目标优化的进化算法

类似于 MOEAs，COEAs 也具有两个明确的目标：1)群体快速地靠近或进入可行域;2)收敛于全局最优解.从上一节的分析中可以看出，现有的约束处理技术存在的主要问题是如何设计合理的个体比较和选择准则，这直接影响着第二个目标的实现.此外，约束优化问题的搜索空间由可行域和不可行域两部分组成.

因而如何有效地利用不可行解显得尤为重要，特别是当全局最优解位于可行域边界或可行域占整个搜索空间的比例相对较小时，而且它对第一个目标的实现起着决定性的作用。

基于以上考虑，本文提出了一类新的基于多目标优化的进化算法求解约束优化问题.事实上，新算法属于第 2 节中定义的多目标优化法，它将约束优化问题转换为具有两个目标的多目标优化问题，其中一个目标为 $f(\vec{x})$，另一个目标为 $G(\vec{x})$.为方便叙述，令 $\mathbf{f}(\vec{x})=(f(\vec{x}),G(\vec{x}))$。

5.3.1 非劣个体替换策略

本文中，我们仅考虑目标函数极小化的情况.对于极大化的情况，可将其转化为极小化的情况来处理，值得注意的是，由于将约束优化问题转换为多目标优化问题，定义 Pareto 最优解时应针对于原问题的定义域 S 而不是可行域 Ω.

在 MOEAs 中，群体中的所有个体均被赋予等级(rank)，而且群体中的非劣个体具有相同的等级值。例如，在文献[73]中非劣个体的等级值为 1，在文献[72]中非劣个体的等级值为 0。与 MOEAs 不同的是，因为群体中的主要信息包含在非劣个体中，所以我们感兴趣的仅仅是非劣个体，因而避免了对群体中其它个体定义等级所带来的计算花销，这使得新算法更加的高效.辨识所有非劣个体的时间复杂度为 $O(MN^2)$，其中，M 为目标个数，N 为群体规模。

当子代群体中的非劣个体被选择后，它们将用来替换父代群体中的劣于个体.根据非劣个体所具有的特征，可以得到以下三个重要的结论：1)非劣个体可能由可行解、不可行解或部分可行解与不可行解组成；2)非劣个体中最多包含一个可行解；2)子代群体非劣个体中的可行解可以 Pareto 优超父代群体中的可行解或不可行解，但子代群体非劣个体中的不可行解只能 Pareto 优超父代群体中的不可行解。

5.3.2 基于群体的算法发生器模型

作为进化算法的一个重要分支，近年来进化策略在处理约束优化问题时受到了极大的重视. 对此有以下两个原因：1)具有理论背景支持进化策略收敛；2)进化策略的自适应机制对其处理约束搜索空间有一定的帮助。实验结果表明，在相同的约束处理技术下采用进化策略时算法的整体性能明显优于遗传算法。通过融合三个简单的个体比较准则，Mezura 等对几种不同的进化策略进行了实验比较，结果表明对于具有高维搜索空间、非线性等式约束条件的测试函数，利用进化策略不能得到很好的结果。同时还应注意到进化策略本身存在着一个缺陷，即在进化策略中交叉操作往往被忽略。但就约束优化问题而言，可行解与不可行解在一些重要的区域进行交叉对于找到全局最优解非常有帮助，特别

是当最优解位于可行域边界时。尽管在进化策略中也可以使用交叉操作(如中间交叉、离散交叉等)，但其搜索能力十分有限。

此外，意识到二进制编码的缺陷，研究者提出了实数编码遗传算法(Real-coded Genetic Algorith- m， RCGA).在RCGA中，交叉作为主要的搜索操作能根据父代个体中的信息自适应地产生后代个体.Hajime 对采用自适应变异的进化策略和采用UNDX杂交算子的RCGA进行了实验比较，结果显示，RCGA在求解高维和多极值函数优化问题时性能占优。最近，Deb 提出了一个专门用于实数编码的基于群体的算法发生器，它将寻优过程分为四个主要策略(plan)：1)选择策略(SP)，2)产生策略(GP)，3)替换策略(RP)，4)更新策略(UP)。具体描述如下：

Setp1.采用选择策略（selection plan）从B中选出u个个体。

Step2.利用繁殖策略（generation plan），从Q中生成λ个子代个体（集合C）。

Step3.采用一定的替换策略，从B中选出γ个个体（集合R）。

Setp4.采用更新策略，从集合R，Q和C中选出γ个个体，更新集合R中原来的γ个个体。

基于群体的算法发生器的主要优点是上述四个重要计划的功能分解。在上述算法发生器的基础上，通过融合非劣个体替换策略，我们得到了以下计算模型：

该算法发生器的主要优点是将寻优的过程进行了功能分解，使得寻优过程更加清晰。在此算法发生器的基础上，我们将非劣个体替换策略融入到其中，得到以下的计算模型：

Step1.随机从B中选出u个个体 。

Step2 $B=B-Q$。

Step3 利用UNDX交叉算子，从群体Q中产生λ个子代个体（群体C）。

Setp4 从群体C的非劣个体中随机的选取一个个体，记为$\vec{x}_1$。

Step5 假定群体Q中有n'个个体被$\vec{x}_1$优超

```
if n'=0， then
   替换不发生；
else if n'=1， then
   被优超的个体用 x1 替代；
else  /*即 n'>1*/
   随机选择一个被优超的个体，用 x1 替代。
end if
```

Step6 $B = B \cup Q$。

在群体进化过程中，n'是动态变化的. 为了更合理地描述上述模型，我们对 Deb 的算法发生器做了适当的修改，但这并不影响原算法发生器的正确性和有效性。

上述模型具有协调勘探和开采的能力：1)群体的方差反映了区域潜在的搜索程度。在搜索早期，随机选择的若干个个体间具有较大的方差，因而利用这些个体进行交叉产生的后代个体可以搜索到更广泛的区域;在搜索后期,群体中的个体具有较小的方差，因而交叉生成的后代个体更加集中地分布在最优解附近。2)子代群体中的非劣个体被选择并用来替换父代群体中的劣于个体(这里，子代群体对应于集合C，父代群体对应于集合Q)，这保证了群体在搜索早期对全局信息的开采和在搜索后期对局部信息的开采。此外，上述模型具有较小的选择压，能有效地保持群体的多样性。

5.3.3 不可行解存档和替换机制

约束优化问题与无约束优化问题的显著差别是，前者的搜索空间由可行域和不可行域两部分组成. 因此，两种区域占搜索空间的比例，全局最优解所处的位置(位于可行域边界或可行域内部)和约束条件的类型(线性，非线性，等式，不等式)均给搜索约束优化问题的全局最优解带来了困难。目前，研究者已经逐渐意识到了不可行解对于找到全局最优解的重要性。文献[63]使得违反约束条件很小且具有较小目标函数值的不可行解依然可行。文献[67]和[69]通过增加一个多样性机制来发挥不可行解的作用。与上述方法不同的是，本文提出了一种不可行解存档和替换机制来促使群体从不同的方向向可行域逼近，其主要思想如下：当前子代群体的非劣个体中违反约束条件最小的不可行解(如果存在)被保存到一个预先定义好的档案A中，经过若干次迭代后，群体P(群体P对应于 5.3.2 节中的集合B)中的部分个体将被存储在档案A中的部分不可行解随机替换. 伪程序描述如下：

R： 子代群体C中的非劣个体集。

$\hat{x}$：最好的不可行解，它是指集合R中违反约束条件最小的不可行个体(如果存在)。

A： 档案，它用来存储最好的不可行解。

```
1. if 种群 P 中存在可行解 then
       if 群体 R 中不存在可行解
       then
       A := A ∪ x̂        //将最好的不可行解保存
       end if
```

```
    else      /*群体中没有可行解 */
        if  群体R中存在不可行解 then
        A:= A ∪ x̂        //将最好的不可行解保存
        end if
    end if
2. if  mod (gen, m")=0 then
       从档案A中随机选取一些个体，至多n"，替换从种群P中随机选择
       的同等数量的个体
       A:= φ;
   end if
```

这里，$m^{"}$和$n^{"}$均为用户设置的参数。上述程序具有很强的自适应性，因为进行存档和替换的不可行解的个数随着进化代数的变化而变化。

对于可行域占搜索空间比例较大的约束优化问题，上述机制对于找到全局最优解的作用并不明显。因为在这种情况下，初始群体中包含着大量的可行解而且在搜索早期可行解便可能被接收，因而不可行解存档和替换不会经常发生除非集合R全部由不可行解组成，但这种情况发生的概率很小。

然而，对于可行域占搜索空间比例较小的约束优化问题，上述机制对于找到全局最优解非常有效。在这种情况下，因为在进化早期集合R经常由不可行解组成，这样不可行解存档和替换将频繁地发生，因此上述机制会不断地引导群体向可行域靠近.随后在一般情况下，1)如果最优解位于可行域边界，则根据非劣个体替换策略，全局最优解的搜索将从可行域边界两边同时进行；2)如果最优解位于可行域内部，则根据非劣个体替换策略，大约在搜索中期以后，群体中的可行解会不断增多，最后群体收敛于全局最优解。此外，该机制还可以作为一个多样性操作来维持群体中可行解与不可行解之间的比例。

5.3.4 重组算子

在实数编码遗传算法中，广泛使用的交叉算子有单形交叉，模拟二进制交叉，单峰正态分布交叉(UNDX)等。本文采用单峰正态分布交叉(UNDX)作为惟一的重组算子，它基于单峰的正态分布来产生后代个体且不需要个体的适应值信息.在R^n中，随机选取$\mu-1$个独立的父体向量($\vec{x}_i, i=1,\cdots,n+1$)，计算这些个体的中心点$g$。以此中心点，形成$\mu-1$个方向向量$d^{(i)}=x^{(i)}-g^{(i)}$，方向余眩为$e^{(i)}=d^{(i)}/\left|d^{(i)}\right|$。然后，计算出另一个随机选出的父体向量与中心点形成的方向向量$x^{(u)}-g$到所有方向余弦$e^{(i)}$的最短距离（正交）D。设

$e^{(j)}$ $(j=\mu,\ \ldots n, n$为变量x的维数）是所有余弦向量所形成的子空间的扩展正交基。子代个体依如下公式产生：

$$y = g + \sum_{i=1}^{\mu-1} \omega_i \left| d^{(i)} \right| e^{(i)} + \sum_{i=u}^{n} \upsilon^i D e^{(i)} \tag{5-6}$$

5.3.5 整体流程

在上述几节对所提出的算法详细描述的基础上，新算法的整体流程如下：

1 选择合适的参数，随机产生初始种群 P。（操作 1）

2 采用算法发生器模型，对种群进行进化。

3 实施不可行解的存档与替换策略。（操作 2）

4 终止：满足停机准则，停止；否则，去第 2 步。

其中，停机准则可以是用户能够接受的一个解或者是一个预先设定好的适应值函数评价次数.值得注意的是，在群体进化过程中，子代群体中的非劣个体数，从档案 A 中选出的用于替换群体 P 中个体的个体数都是动态变化的。

正如已经讨论的那样，COEAs 具有两个明确的目标.为了实现第一个目标，上述整体流程中的操作 2 可以不断地促使群体向可行域靠近；为了实现第二个目标，上述整体流程提供了另外一个操作(操作 1)使得群体能够收敛于全局最优解。同时，随着群体逐步进入可行域，操作 2 的影响将变得越来越小，这有利于群体后期的收敛。

为了评估新算法的全局优化性能，我们选取 6 个常用的标准测试函数进行试验研究。它们都是高维的带约束条件的函数优化 问题，约束条件有等式约束，不等式约束以及两者的混合。

Minimize:

$$G1(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$

subject to:

$g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0$,

$g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0$,

$g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0$,

$g_4(\vec{x}) = -8x_1 + x_{10} \le 0$, $g_5(\vec{x}) = -8x_2 + x_{11} \le 0$,

$g_6(\vec{x}) = -8x_3 + x_{12} \le 0$,

$g_7(\vec{x}) = -2x_4 - x_5 + x_{10} \le 0$,

$g_8(\vec{x}) = -2x_6 - x_7 + x_{11} \le 0$,

$g_9(\vec{x}) = -2x_8 - x_9 + x_{12} \le 0$,

$0 \le x_i \le 1 \, (i = 1,\cdots,9)$，$0 \le x_i \le 100 \, (i = 10,11,12)$，$0 \le x_{13} \le 1$.

已知最优解为 $\vec{x}^*$=(1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)，$f(\vec{x}^*)$=15. 约束条件 g_1，g_2，g_3 g_7，g_8 和 g_9 活跃。

Maximize:

$$G2(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

subject to:

$g_1(\vec{x}) = 0.75 - \prod_{i=1}^{n} x_i \le 0$，

$g_2(\vec{x}) = \sum_{i=1}^{n} x_i - 7.5n \le 0$，

$0 \le x_i \le 10 \, (i = 1,\cdots,n)$，$n = 20$.

全局最优解未知，目前公布的最好结果为 $f(\vec{x}^*) = 0.803619$. 约束条件 g_1 几乎活跃 $(g_1 = -10^{-8})$。

Minimiz:

$$G3(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

subject to:

$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 + 0.0022053x_3x_6 - 92 \le 0$

$g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_6 \le 0,$

$g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0,$

$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0,$

$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0,$

$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0,$

$78 \le x_1 \le 102$，$33 \le x_2 \le 45$，$27 \le x_i \le 45 \, (i = 3,4,5)$.

已知最优解为 $\vec{x}^*$ =(78, 33, 29.995256025682, 45, 36.775812905788)，$f(\vec{x}^*)$=-30665.539. 约束条件 g_1 和 g_6 活跃。

Minimize:

$$\begin{aligned} G4(\vec{x}) = {} & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 \\ & + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\ & + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \end{aligned}$$

subject to:

$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$，

$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$，

$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$，

$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$，

$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0,$

$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0,$

$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0,$

$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,$

$-10 \le x_i \le 10\ (i = 1, \cdots, 10)$.

已知最优解为 $\vec{x}^* =$ (2.171996，2.363683，8.773926，5.095984， 0.9906548，1.430574，1.321644，9.828726，8.280092，8.375927)， $f(\vec{x}^*) =$ 24.3062091. 约束条件 g_1， g_2， g_3， g_4， g_5 和 g_6 活跃。

Minimize:

$$G5(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to:

$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$，

$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$，

$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$，

$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$，

$-10 \le x_i \le 10\ (i = 1, \cdots, 7)$.

已知最优解为 $\vec{x}^* =$ (2.330499，1.951372，-0.4775414， 4.365726，-0.6244870，1.1038131，1.594227)， $f(\vec{x}^*) =$ 680.600573. 约束条件 g_1 和 g_4 活跃。

Minimize:

$$G6(\vec{x}) = x_1 + x_2 + x_3$$

subject to:

$g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \le 0$，

$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0$，

$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \le 0$，

$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0$，

$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0$，

$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0$，

$100 \le x_1 \le 10000$，$1000 \le x_i \le 10000\ (i=2,3)$，$10 \le x_i \le 1000\ (i=4,\cdots,8)$.
已知最优解为 $\vec{x}^* =$ (579.3167，1359.943，5110.071，182.0174， 295.5985，217.9799，286.4162，395.5979)，$f(\vec{x}^*)$=7049.3307. 约束条件 g_1，g_2 和 g_3 活跃。

数值实验在 MATLAB 中完成，对每个测试函数，我们进行了 30 次独立的实验，在实验中，群体规模设置为 50，UNDX 中父体个数 μ 取为 $n+1$：

此外，$\mu = n+1$，$\lambda = 10$，$m'' = 10$，$n'' = 2$。

5.3.6 实验结果分析

表 5-1 实验结果比较

fun		G1	G2	G3	G4	G5	G6
optimal		-15.000000	-0.803619	-30665.539	24.3062091	680.6300573	7049.3307
Best	MOCO	-15.000000	-0.803619	-30665.539	24.3062091	680.6300573	7049.248021
	RY	-15.000000	-0.803515	-30665.539	24.307	680.63	7054.316
	SAFF	-15.000000	-0.80297	-30665.5	24.48	680.64	7061.34
Mean	MOCO	-15.000000	-0.80322	-30665.539	24.3062091	680.6300573	7049.248021
	RY	-15.000000	-0.781975	-30665.539	24.374	680.656	7559.192
	SAFF	-15.000000	-0.7901	-30665.2	26.58	680.72	7627.89
Worst	MOCO	-15.000000	-0.792608	-30665.539	24.3062091	680.6300573	7049.248023
	RY	-15.000000	-0.726288	-30665.539	24.642	680.763	8835.655
	SAFF	-15.000000	-0.76043	-30663.3	28.4	680.87	8288.79

为了对比，我们将新算法的运行结果(记为 MOCO)与另外两个较好的约束优化进化算法进行了比较，它们分别为文献[22]中的随机排序法(记为 RY)和文献[24]中的自适应适应值表示法(记为 SAFF).比较结果见表 4。从表 4 中可以看出，MOCO 在最优结果，平均结果，最差结果等性能指标方面显著占优.尽管其它两个算法提供了非常好的实验结果，但绝大部分结果都与 MOCO 得到的结果相符或比它差。另一方面，因为等式约束条件转换为成对的不等式约束条件，RY 中的有实验些结果要优于已知的最好结果(用黑体表示)，但这并不意味着 RY 找到了更好的最优解。

在本文的实验中，MOCO 对 6 个测试问题(除对 G2 有两次陷入局部最优外)一致地找到了全局最优解。然而 RY 和 SAFF 却极易陷入局部最优，特别是对其中几个复杂的测试函数，如 G2，G4，G5 和 G6 等.对于高维多峰测试函数 G2 ，RY 和 SAFF 不能找到精确的最优解，对于具有很高约束的测试函数 G6，RY 和 SAFF 所得到的结果仍远离最优解。而且对于测试函数 G6 ，这两个算法在找到可行解方面都存在着困难，例如，RY 在 30 次实验中仅 7 次找到了可行解，SAFF

在 20 次实验中找到可行解的次数为 16 次。

从以上比较研究中可以看出，无论在解的质量上还是在计算代价上，MOCO 都优于其它两个算法。

5.4 结论

基于多目标优化技术，本章提出了一类新的进化算法求解约束优化问题.在新算法中，父代群体中的个体可能被替换如果它们劣于子代群体中的非劣个体。此外，新算法还包括两个重要的操作，其一是一个基于群体的算法发生器模型，其二是一个不可行解存档和替换机制.约束优化进化算法具有两个明确的目标，通过上述两个操作可以在一定程度上实现这两个目标。

从实验结果可以看出，新算法具有很强的处理各种类型约束优化问题的能力.同其它算法相比，新算法的实验结果在诸多方面都显著占优.更为重要的是新算法能搜索到所有测试函数的最优解。

今后的工作将从两方面展开，一方面采用或设计更具广泛性和典型性的测试函数来验证新算法，另一方面将算法推广到多目标约束优化问题。

第六章 总结与展望

6.1 论文工作总结

总结全文，在以下几个方面提出了一些有一定的创新性的方法：

1、将多目标属性决策方法中的 ELECTRE 法引入到多目标优化进化算法中，提出了一种新的多目标优化算法。构造出一种新的超序关系对个体进行排序，并证明了该超序关系比 Pareto 优劣关系弱，利用此超序关系能增强进化过程中的选择压，加快收敛速度。

2、提出了一种基于差异进化的多目标优化算法，利用ε优超关系来保存进化中的非劣个体 ，并对ε-MOEA 算法中个体存档准则进行了改进。将ε-水平比较准则引入到算法中，通过对一系列标准的测试函数进行实验，实验结果表明该算法在保持解集分布性和收敛性方面非常有效。

3、将多目标优化思想引入到约束优化中，提出了非劣个体替换准则，为了有效利用不可行解，提出了一种不可行解保存和替换机制，对 6 个测试函数进行了测试，测试结果表明该算法优于其它算法。

6.2 下一步的研究方向

进一步的主要研究有：

1、多目标的进化机理，以及进化个体之间的内在联系和进化的一般规律。

2、多目标进化算法收敛性的研究。

3、将聚类，主成分分析法，回归分析等引入到多目标优化进化算法中，用以降低目标的维数。

4、构造 Pareto 最优解集（或支配集）的方法。

5、用多目标进化算法解决实际问题。

参考文献

[1] 郑金华．基于 Pareto 最优的多目标进化算法及其应用研究．北京：中国科学院计算技术研究所博士后工作报告, 2005。

[2] 蔡自兴，徐光佑．人工智能及其应用． 北京：清华大学出版社，2004

[3] Back T, Hammel U, and Schwefel H P. Evolutionary computation: Comments on the history and current state. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 3-17.

[4] R. Storn and K. Price, “Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces,” International Computer Science International Computer Science Institute, Berkeley, Tech. Rep. TR-95-012, 1995.

[5] Fonseca C M, Fleming P J. An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation. 1995, 3(1):1-16.

[6] Valenzuela-Rendon and Uresti-Charre. A non-generational genetic algorithm for multiobjectiv optimization. In: Proceedings of the Seventh International Conference on Genetic Algorithms, Back T (Ed.), San Francisco, California, Morgan Kaufmann, 1997: 658-665.

[7] Wolpert D H and Macready W G. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1):67-82.

[8] Horn Jeffrey. Handbook of Evolutionary Computation, Chap. Multicriterion Decision Making. Vol.1 of Back et al, 1997:F1.9:1-F1.9:15.

[9] J.D.Schaffer. Some experiments in machine learning using vector evaluated genetic algorithms. PhD thesis, Vanderbilt University, 1984

[10] D.E.Goldberg. Genetic algorithms for search, optimization, and machine learning, Reading, MA: Addison-Wesley, 1989

[11] Brian J. Ritzel, J. Wayland Eheart, and S. Ranjithan . Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. Water Resources Research, 30(5): 1589-1603, 1994

[12] Srinivas N and Deb Kalyanmoy. Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 1994,2(3):221-248.

[13] Jeffrey Horn and Nicholas Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm, Technical Report IlliGAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.

[14] Cunha A G, Oliviera P, and Covas J. Use of genetic algorithms in multicriteria optimization to solve industrial problems, In Back T (Ed.), Proceedings of the Seventh International Conference on Genetic Algorithms, San Francisco, California, pp 682-688. Morgan Kaufmann, 1997.

[15] Parks G T and Miller I. Selective breeding in a multiobjective genetic algorithm. In Eiben A E, Back T, and Schoenauer M (Eds.), Fifth International Conference on Parallel Problem Solving from Nature, Berlin, Germany, pp. 250-259. Springer, 1998.

[16] Gunter Rudolph, On a multi-objective evolutionary algorithm and its convergence to the Pareto set, In Proceedings of the 5^{th} EEEE Conference on Evolutionary Computation, pages 511-516, Piscataway, New Jersey, IEEE Press, 1998: 511-516..

[17] Obayashi S, Takahashi S, and Takeguchi Y. Niching and elitist models for mogas, In Eiben A E, Back T, and Schoenauer M (EDs.), Fifth International Conference on Parallel Problem Solving from Nature, Berlin, Germany, pp. 260-269. Springer, 1998.

[18] Parks G T and Miller I. Selective breeding in a multiobjective genetic algorithm, In Eiben A E, Back T, and Schoenauer M (Eds.), Fifth International Conference on Parallel Problem Solving from Nature, Berlin, Germany, pp. 250-259, Springer, 1998.

[19] Horn J., Nafpliotis N., & Goldberg D.E. (1994). A Niched Pareto genetic Algorithm for Multiobjective Optimization. *Proceeding of the first IEEE Conference on Evolutionary Computation*, 82-87.

[20] Van Veldhuizen D A and Lamont G B. Multiobjective evolutionary algorithm research: A history and analysis, Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[21] Carlos A. Coello Coello. An Updated Survey of GA-Based Multiobjective Optimiation Techniques: State of the Art and Future Trends, In 1999 Congress on Evolutionary Computation, pages 3-43, Vol. 1,Washington, D,C., July 1999.IEEE Service Center.

[22] Runarsson T. P., Yao X.. Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, 2000,4(3): 284–294.

[23] Hamida S. Ben., Schoenauer M.. ASCHEA: New results using adaptive segregational constraint handling. In Proc of the Congress on Evolutionary Computation 2002 (CEC’2002), Piscataway, NJ: IEEE Press, 2002. 82–87.

[24] Farmani R., Wright J. A.. Self-adaptive fitness formulation for constrained optimization. IEEE Transactions on Evolutionary Computation, 2003 7(5): 445–455.

[25] Yu J. X., Yao X., Choi C., Gou G... Materialized view selection as constrained evolutionary optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 2003,33(4): 458–467.

[26] Koziel S., Michalewicz Z.. Evolutionary algorithm, homomorphous mappings, and constrained parameter optimization. Evolutionary Computation, 1999, 7(1): 19–44..

[27] Deb, K, Mohan, M. and Mishra, S. (February, 2003) A Fast Multi-Objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions, KanGAL Report No.2003002.

[28] Kalyanmoy Deb, Amrit Pratap, Sameer Agrawal and T.Meyrivan. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182-197, April 2002.

[29] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. EUROGEN 2001-Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, K.C. Giannakoglou et al.,Eds., 2001, pp.95-100. .

[30] D. W. Corne, N. R. Jerran, J. D. Knowles and M. J. Oates, PESA-II: Region-based selection in evolutionary multi-objective optimization, In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann Publishers, 2001,pp. 283-290.

[31] Carlos Artemio Coello Coello. An Updated Survey of GA-Based Multiobjective Optimization Techniques. ACM Computing Surveys, 32(2):109-143, June 2000.

[32] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, In Stephanie Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[33] David A. Van Veldhuizen and Gary B. Lamont. Evolutionary Computation and Convergence to a Pareto Front, In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221-228, Stanford University, California, July 1998. Stanford University Bookstore,

[34] Gunter Rudolph, On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set, In Proceedings of the 5th EEEE Conference on Evolutionary Computation, pages 511-516, Piscataway, New Jersey, 1998, IEEE Press, 1994.

[35] 郑金华， 史忠植。 基于聚类的快速多目标遗传算法。计算机研究与发展，2004，41（7），1081-1087

[36] 周秀芬，刘敏忠，吴志健，康立山。解约束多目标优化问题的一种鲁棒的进化算法。计算机研究与发展，2004，41（06）：985-990。

[37] 崔逊学。一种求解高维优化问题的多目标遗传算法及其收敛性分析。计算机研究与发展，2003，40（7）：901-906。

[38] 马清亮，胡昌华，杨青。一种用于多目标优化的混合遗传算法。 系统仿真学报，2004，16（5）：1038-1040。

[39] 马清亮，胡昌华。基于多目标进化算法的混合 H-2/H-∞优化控制。控制与决策，2004，19（6）：699-701。

[40] 郑金华，蔡自兴。自动区域划分的分区域搜索狭义遗传算法，计算机研究与发展，2000，37（4）：397-400。

[41] 杨青，汪亮，叶定友。基于多目标遗传算法的固定火箭发动机面向成本优化设计。固体火箭技术，2002，25（4）：16-20。

[42] 张潜，高立群。物流配送路径多目标优化的聚类-改进遗传算法[J]。控制与决策，2003，18（4）：418-422。

[43] 李满林，杜雷。多目标优化遗传算法在移动网络规划中的应用。控制与决策，2003，18（4）：441-444。

[44] 赵曙光，王宇平。基于多目标自适应遗传算法的逻辑电路门级进化方法。计算机辅助设计与图形学学报，2004，16（4）：402-406。

[45] 朱力立，张焕春，经亚枝。一种基于模糊遗传算法的多传感器多目标跟踪数据关联算法（英文）.Chinese Journal of Aeronautics，2003，16（3）：176-481。

[46] 杨云，徐永红，李千亩，刘凤玉。一种 QoS 路由多目标遗传算法。通信学报，2004，25（1）：43-51。

[47] 姜山，程君实，陈佳品，包志军，马培荪。基于多目标遗传算法的仿人机器人中枢神经控制器的设计。机器人，2001，23（1）：58-62。

[48] 林丹。遗传算法在证券组合投资中的应用研究。中国科学院数学与系统科学研究院博士后论文，20020501。

[49] 谢涛，陈火旺，康立山。多目标优化的演化算法。软件学报，2003，26（8）：997~1003。

[50] Deb, K. and Saxena, D. (December, 2005). On Finding Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. *KanGAL Report No. 2005011*

[51] Deb, K., Sundar, J. and Uday, B. R. N. (December, 2005). Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. *KanGAL Report No. 2005012*.

[52] 崔逊学，李淼，方廷健。多目标协调进化算法研究。计算机学报，2001（9）：979~984。

[53] Roy B. How outranking relation help multiple criteria decision making.Cochrane J L, Zeleny M eds. Multiple Criteria Decision Making. South Carolina: University of South Carolina Press,1973: 179~201.

[54] S. Tsutsui, M. Yamamure, and T. Higuchi. “Multi-parent recombination with simplex crossover in real coded genetic algorithms,” in *Proc Genetic and Evolutionary Computation Conf (GECCO’99)*, 1999, pp. 657–664.

[55] Fonseca C M, Fleming P J. Multiobjective optimization and multiple constraint handling with evolutionary algorithms, part I: A unified formulation. IEEE Transactions on Systems, Man and Cybernetics, 1998, 28: 26~37.

[56] Velduizen D, Lamont G. Multiobjective evolutionary algorithm test suites. *Proc Symp Applied Computing*. San Antonio, TX, 1999:351.

[57] Rainer Storn and Kenneth V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Globlal Optimization, vol, 11, no. 4, pp. 341-359, Dec 1997.

[58] Hussein A. Abbass and Ruhul Sarker, “ The Pareto differential evolution algorithm,” International Journal on Artificial Intelligence Tools, vol. 11, no. 4, pp. 531-552, 2002.

[59] Nateri K. Madavan, “ Multiobjective optimization using a Pareto differential evolution approach,” in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC’02*,Honolulu, Hawaii, May 2002, pp. 1145-1150.

[60] K.E. Parsopoulos, D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis, Vector Evaluated Differential Evolution for Multiobjective Optimization, *Proceedings of the IEEE 2004 Congress on Evolutionary Computation, 2004*.

[61] Saku Kukkonen and Jouni Lampinen, “ A differential evolution algorithm for constrained multi-objective optimization: Initial assessment,” in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, InnsBruck, Austria, Feb 2004, pp.96-102.

[62] A. Iorio and X.LI, Solving Rotated Multi-objective Optimization Problems Using Differential Evolution, Proceeding of the 17th Joint Australian Conference on Artificial Intelligence, 2004

[63] T. Robic and B.Filipic, DEMO: Differential Evolution for Multiobjective Optimization, *Proceedings of the Conference on Evolutionary Multiobjective Optimization*, 2005

[64] T.Takahama and S. Sakai, “Constrained optimization by ε constrained particle swarm optimizer with ε -level control,” in Proc, of the 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST’05), Muroran, Japan, May 2005, pp: 1019~1029.

[65] D.A.Van Veldhuizen and G.B.Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Dept.Elec.Comput.Eng., Graduate School of Eng., Air Force Inst. Technol, Wright-Patterson AFB, OH, Tech.Rep.TR-98-03, 1998.

[66] Schoot, J. R., Fault tolerant design using single and multicriteria genetic algorithms optimization. Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts

Institute of Technology, Cambridge, MA, 1995.

[67] 林丹,李敏强,寇纪凇. 基于遗传算法求解约束优化问题的一种算法.软件学报,2001,12(4): 628-632.

[68] Coello Coello. A. C.. Treating constraints as objectives for single-objective evolutionary optimization. Engineering Optimization, 2000, 32(3): 275–308.

[69] Coello Coello. A. C.. Constraint handling using an evolutionary multiobjective optimization technique. Civil Engineering and Environmental Systems, 2000, 17:31–346.

[70] Coello Coello CA., Mezura-Montes E.. Constraint-handling in genetic algorithms through the use of dominance-based tournament Selection. Advanced Engineering Informatics, 2002, 16(3): 193–203.

[71] Aguirre A. H., Rionda S. B., Coello Coello. A. C., Lizáraga G. L., Mezura-Montes E.. Handling Constraints using Multiobjective Optimization Concepts. International Journal for Numerical Methods in Engineering, 2004,59(15): 1989–2017.

[72] Mezura-Montes E., Coello Coello. A. C.. On the usefulness of the evolution strategies self-adaptation mechanism to handle constraints in global optimization. Tech. Rep. EVOCINV-01-2003, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAVIPN, México D.F., México.

[73] 周育人,李元香,王勇,康立山. Pareto 强度值进化算法求解约束优化问题.软件学报,2003,14(7): 1243-1249.

致　谢

在中南大学攻读硕士学位的三年时间里，得到了导师、家人、师兄弟、老师、朋友在工作上和生活上的热心指导、关心和帮助。在这里，我要向他们表示最真诚的感谢。

首先要衷心感谢我的导师蔡自兴教授。整个研究生阶段，无论是学习、科研，还是生活上，蔡老师都给了我很多关怀和帮助。蔡老师治学严谨，知识渊博，成果丰硕，具有很高的学术地位，但他依然为人谦和，甘为人梯。蔡老师对学生要求严格，同时又能让我们处处感受到他的慈爱。蔡老师为我提供了良好的学习科研环境，在他的精心指导下，使我论文研究得以顺利完成。蔡老师超出常人的毅力与勤勉敬业作风，是我一生学习的典范。

在这里我还要特别感谢师兄王勇。正是在他的指导与帮助下，我才得以进行进化计算方面的研究。他的勤奋，以及受过无数次挫折依然奋发进取的精神，一直是我学习的榜样。跟随其研究三年，无所成果，辜负厚望，深感愧疚。

感谢进化计算小组的师兄弟们。他们是：肖赤心，刘慧，郭铖，封鹏成，和他们一起学习讨论，使我获益菲浅。

此外还要感谢王璐博士、段琢华博士、高平安博士、陈爱斌博士、屈太国博士、陈白帆博士、潘薇博士给予我大量的指导和帮助。

感谢智能所里的肖晓明老师、唐琎老师、魏世勇老师、刘丽珏老师的帮助。

此外还要感谢和我朝夕相处的亲密同学，他们是：谢峰、申丽曼、黎陟、彭梦、肖正、黄明登、宁火明、胡华梅、夏洁、曾维彪等等，和他们相处的日子给我留下了非常美好的回忆！

最后我要深深的感谢我的父母、弟弟、女友。他们的关怀和鼓励是我顺利的完成学业的直接精神动力！本文也是献给他们的最好礼物！

特别感谢国家自然科学基金重点项目“未知环境中移动机器人导航的理论与方法研究”（基金编号：60234030）的资助！

签名：

日期：

攻读学位期间主要的研究成果

[1] 参加国家自然科学基金重点项目——未知环境中移动机器人导航控制的理论与方法研究．项目号：（60234030）。.

[2] 曾威，蔡自兴，郭铖．一种基于 ELECTRE 法的多目标优化进化算法．东南大学学报．2006，36(sup)：99-103.(EI Compendex Index)

[3] 王勇，蔡自兴，曾威，刘慧．求解约束优化问题的一种新的进化算法。中南大学学报（自然科学版）2006，37(1)：119-123

[4] Yong Wang, Zixing Cai，Yuren Zhou and Wei Zeng. A self-adaptive trade-off model for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*. (accepted for publication)