



江西财经大学

JIANGXI UNIVERSITY OF FINANCE AND ECONOMICS



Y3582051

学校代码 \_\_\_\_\_

密 级 \_\_\_\_\_

中图分类号 \_\_\_\_\_

UDC \_\_\_\_\_

# 硕士学位论文

## MASTER DISSERTATION

论文题目 \_\_\_\_\_ 基于遗传算法优化钢铁烧结配料建模  
(中文)

论文题目 \_\_\_\_\_ Optimization of steel sintering batching model  
(英文)  
\_\_\_\_\_ based on genetic algorithm

作 者 \_\_\_\_\_ 杜家楠 导 师 \_\_\_\_\_ 罗世华

申请学位 \_\_\_\_\_ 硕 士 培养单位 \_\_\_\_\_ 统计学院

学科专业 \_\_\_\_\_ 统计学 研究方向 \_\_\_\_\_ 复杂系统建模

二〇一九 年 六 月

# 独创性声明



本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得江西财经大学或其他教育机构的学位或证书所使用过的材料。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了谢意。

签名：杜家林 日期：2019年6月3日

## 关于论文使用授权的说明

本人完全了解江西财经大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后遵守此规定）

签名：杜家林 导师签名：[Signature] 日期：2019年6月3日

# 目 录

第1章 引 言.....	1
1.1 论文研究的背景和意义.....	1
1.2 国内外研究现状.....	2
1.3 基于人工智能的专家系统配料的研究.....	9
1.4 论文的主要内容及成果.....	10
第2章 遗传算法.....	12
2.1 人工智能算法的概述.....	12
2.2 遗传算法的研究现状.....	13
2.3 遗传算法生物学基础.....	15
2.4 遗传算法的发展过程.....	16
2.5 遗传算法的基本理论.....	19
2.6 遗传算法操作步骤.....	22
2.7 本章小结.....	35
第3章 多种群遗传算法.....	37
3.1 遗传算法早熟问题.....	37
3.2 多种群遗传算法概述.....	38
3.3 多种群结构设计.....	39
3.4 变异算子设计.....	42
3.5 本章小结.....	43
第4章 新型遗传算法.....	45
4.1 改进遗传算法.....	45
4.2 自适应遗传算法.....	46
4.3 混合遗传算法.....	47
4.4 并行遗传算法.....	48
4.5 免疫遗传算法.....	48
4.6 精英保留策略新思路.....	49
4.7 本章小结.....	52
第5章 烧结配料模型的建立.....	53
5.1 烧结配料模型的数据及约束条件.....	53
5.2 建立烧结配料的数学模型.....	54
5.3 罚函数.....	55
5.4 结果分析.....	56
5.5 本章小结.....	56

结束语.....57

参考文献.....58

致 谢.....62

# Contents

<b>Chapter 1 Introduction .....</b>	<b>1</b>
1.1 Background and significance of the research.....	1
1.2 Research Status at Home and Abroad.....	2
1.3 Research on the Expert System Based on Artificial Intelligence.....	9
1.4 Main contents and results of the paper.....	10
<b>Chapter 2 Genetic Algorithm .....</b>	<b>12</b>
2.1 Overview of Artificial Intelligence Algorithms.....	12
2.2 Research Status of Genetic Algorithms.....	13
2.3 Biological basis of genetic algorithm.....	15
2.4 Development process of genetic algorithm.....	16
2.5 Basic theory of genetic algorithms.....	19
2.6 Genetic Algorithm Operation Steps.....	22
2.7 Summary of this chapter.....	35
<b>Chapter 3 Multi-population genetic algorithm .....</b>	<b>37</b>
3.1 Genetic algorithm precocious problem.....	37
3.2 Overview of Multi-Group Genetic Algorithms.....	38
3.3 Multi-group structure design.....	39
3.4 Mutation Operator Design.....	42
3.5 Summary of this chapter.....	43
<b>Chapter 4 New Genetic Algorithms.....</b>	<b>45</b>
4.1 Improved genetic algorithm.....	45
4.2 Adaptive Genetic Algorithm.....	46
4.3 Hybrid Genetic Algorithm.....	47
4.4 Parallel Genetic Algorithm.....	48
4.5 immune genetic algorithm.....	48
4.6 Elite retention strategy new ideas.....	49
4.7 Summary of this chapter.....	52
<b>Chapter 5 Establishment of Sintering Formulation Model .....</b>	<b>53</b>
5.1 Sintering batching model data and constraints.....	53
5.2 Establishing a mathematical model of the sintering ingredients .....	54
5.3 penalty function.....	55
5.4 Analysis of results.....	56

5.5 Summary of this chapter.....56

**Conclusion.....57**

**References..... 58**

**Thank you..... 62**

## 摘 要

在 20 世纪 80 年代,一些科学家通过观察生物的进化过程开发了一些智能算法。这些算法为各行各业的优化工作提供了一种全新的视角。发明者都是在自然界一些进化与生活现象的启发下脑洞大开提出这些优化算法。本文写作的目的就是要应用智能算法当中的遗传算法与它的改进算法优化钢铁烧结配料建模,并提出一种新的思路改进遗传算法的方法。

本文着重介绍了遗传算法与多种群遗传算法优化钢铁烧结配料的过程,并且详细的介绍了这两种算法的原理以及操作的步骤流程。先介绍了传统遗传算法的原理以及其应用的领域,再介绍了多种群遗传算法并且把这种改进的遗传算法应用在优化钢铁烧结配料建模中。以往的钢铁烧结配料建模都是应用传统遗传算法进行优化,本文作者查遍了很多网站的论文和相关书籍并没有发现学者应用多种群遗传算法对烧结配料建模进行优化。作者应用多种群遗传算法优化烧结配料建模算是一种应用领域的创新,并应用计算机把这两种算法进行了模拟与实证。

由于烧结配料建模日益复杂,各种限制条件越来越多,应用传统遗传算法进行优化容易陷入局部最优。为了解决传统遗传算法的缺陷作者针对此问题提出了一种新型精英保留策略,能保证算法在优化烧结建模时最后输出的结果达到全局最优解并为以后研究遗传算法精英保留的科研人员提供一种新思路。

**关键词:** 传统遗传算法、多种群遗传算法、新型精英保留策略、烧结配料建模

## Abstract

Since the 1980s, some intelligent optimization algorithms such as genetic algorithm, particle swarm algorithm and BP neural network algorithm have been born. These intelligent algorithms provide a new perspective for the optimization work of all walks of life. The inventors are all inspired by some evolutionary and life phenomena in nature to propose these optimization algorithms. The purpose of this paper is to apply the genetic algorithm in the intelligent algorithm and its improved algorithm to optimize the modeling of steel sintering ingredients, and propose a new way to improve the genetic algorithm.

This paper focuses on the process of genetic algorithm and multi-population genetic algorithm to optimize steel sintering ingredients, and introduces the principle of these two algorithms and the steps of the operation. Firstly, the principle of traditional genetic algorithm and its application field are introduced. Then, a variety of genetic algorithm is introduced and this improved genetic algorithm is applied to optimize the modeling of steel sintering batching. In the past, the modeling of steel sintering batching was optimized by traditional genetic algorithm. The authors have searched many websites and related books and found that scholars applied multi-group genetic algorithm to optimize the modeling of sintering ingredients. The author applied multi-group genetic algorithm to optimize sintering batching modeling calculation is an application field innovation, and applied MATLAB programming to empirically demonstrate traditional genetic algorithm and multi-population genetic algorithm.

Due to the increasingly complex modeling of sintering ingredients and the increasing number of constraints, the application of traditional genetic algorithms for optimization is likely to fall into local optimum. In order to solve the defects of traditional genetic algorithms, the author proposes a new elite retention strategy for this problem, which can ensure that the final output of the algorithm in the optimization of sintering modeling reaches the global optimal solution and provides the researchers who will study the genetic algorithm elites in the future. A new idea.



**Key words:**Traditional genetic algorithm、 Multi-population genetic algorithm、  
New elite retention strategy、 Sintering batching modeling.

# 第1章 引言

## 1.1 论文研究的背景和意义

钢铁行业现在发展越加迅猛, 烧结品质的提高对高炉来说越来越重要, 优化烧结技术的重要环节就是怎样提高烧结配料的品质。在我国铁矿石种类繁多且丰富, 质量差的矿多, 质量好的矿少, 化学性质和粒度都不均匀。烧结配料要满足工艺对它的高标准, 往往烧结配料的配比要严格按照工艺的限制条件去生产<sup>[1]</sup>, 一方面生产出的烧结矿产量要达到既定的要求满足产量的生产指标, 另一方面生产烧结矿的各种原料要按一定的配比来进行添加, 这样才能满足烧结矿的质量要求, 否则生产出的烧结矿质量参差不齐无法使用。很多国内的钢铁厂商使用的烧结配料测量机器测量出的配比都不稳定, 长时间使用后准确率有所下降, 导致生产出的烧结矿质量有所下降, 同时, 烧结生产这个工业过程具有很大的时滞特性, 从一开始在配料室确定配料比下料到生产出烧结矿再到对烧结矿出料的化验检测需要4个小时的时间, 当你发现烧结指针出现偏差在想调整烧结配料的配比时, 此时为时已晚, 会生产出很多不达标准的烧结矿, 浪费大量的原料与时间, 增加烧结矿的成本。以上这些问题都影响了配料系统自动控制目标的能力, 导致配料系统自动控制能力不稳定<sup>[2]</sup>, 为了保证工艺流程高效稳定的运行并且生产出高质量的成品矿, 这样就要求我们要对烧结配料进行不断合理与有效优化, 才能保证生产出的烧结矿符合生产的要求。

目前钢铁工业的发展速度日益增长, 地球上的生产资源有限并且国家大力倡导各行业在发展过程中走可持续发展的道路为后代留下绿水青山就是为他们积攒下金山银山, 因此, 钢铁企业要发展首要考虑的就是不断提高钢铁的质量和效益并且要降低成本不能浪费钢铁的原材料。由于国际市场上的铁矿石价格忽高忽低及其不稳定, 国内钢铁企业在生产钢铁时就要把原料成本作为首要的考虑, 会调整生产钢铁原料的种类和配量比, 当铁矿石的一些性质变得不稳定时, 为了减少钢铁厂商的生产成本, 怎样优化烧结配方案就更加需要关注了。低成本、可操作的配料方案成为诸多钢铁企业的现实追求, 具有重要的理论和实践意义。

目前在生产钢铁的烧结过程中铁矿石是主要的原料, 因为在自然界众多的矿

石中铁矿石含铁量相对较高。钢铁工业的发展日益迅猛,发展的速度有目共睹对铁矿石的需求则越来越多,然而国内的铁矿石资源十分有限,这样铁矿石原料就会由钢铁厂商大量的从外国进口。由于进口的原料质量比较好,所含杂质也比较低且炼钢性能很好,综合的优势比国内的铁矿石要大,所以目前国内的大多数钢铁厂都采用进口的铁矿石,这样一来进口的铁矿石价格就水涨船高导致生产钢铁的成本不断增加<sup>[3]</sup>。在铁矿石资源越来越吃紧供应量不足,进口铁矿石品种丰富,国内铁矿石原料质量好坏差别比较大的情况下,怎样能使炼钢的成本降低并且不浪费铁矿石资源合理利用它,我国钢铁企业必须对此问题加以高度的关注,这个问题将会成为企业未来发展首要考虑的问题。怎样选择铁矿石的种类和合理的配比并且此配比要为钢铁厂带来可观的效益,同时,这样的配料方案要能满足生产过程中的质量、物理及化学的各项指标要求,所以说合理利用和优化铁矿石资源很重要,进而对高炉炼铁也有一定的影响。要考虑烧结矿的质量满足高炉的需求并且根据钢铁企业原料储存情况,先将生产钢铁的原料按照一定的规则混放在一起,在添加一些其他的生产原料,然后再把这些原料进行集体烧结,最后就形成了我们需要的烧结矿<sup>[4]</sup>。其他的原料可能对生产烧结配料起到较大的影响,会导致成本升高,产量降低,质量也有所下降,要想生产出好的烧结矿,那么烧结矿的烧结过程是很重要的。好坏起决定作用。质量好的中和矿和混合料对生产钢铁具有极其重要的作用,二者可以降低生产钢铁的成本和资源,还可以提高烧结矿和钢铁的质量。因此,研究如何优化烧结配料对于钢铁企业具有极其重要的意义。

## 1.2 国内外研究现状

目前,外国和国内主要从两个方面来研究烧结配料模型问题,一是用数学的方法对烧结配料模型进行改进,二是用智能的方法设计一套系统去优化烧结配料模型。

### 1.2.1 构建烧结配料的数学模型进行优化

外国二十世纪六十年代,国内二十世纪八十年代才开始用数学方法改进烧结配料模型,在每个时期都运用了大量的数学方法优化模型,主要运用的数学方法有离线计算分析法,参数优化法,多目标综合法。要依照烧结原料的物化性质、

供给量和烧结矿铁的品质<sup>[5]</sup>、碱度等来确定烧结配料的配比，建立烧结配料的基本模型时要依照物质守恒的定理，再通过一系列的数学计算确定原料的配比。

在二十世纪七十年代，能作为烧结配料使用的原料铁矿石的品种数量有限，实现原料的配比通常会采用建议理论法。简易理论这种计算方法是比较常规的一种方法，计算每种矿原料的配比数时通常依据原料的物理和化学性质。这种方法有优点也有缺点，优点就是计算简单并且十分快捷，缺点是仅可以在矿原料种类少的情况下使用，当矿原料多的时候此种方法效果则不是十分理想。

运筹学的发展十分迅速，进行烧结配料优化时越来越多的采用线性规划法。运筹学应用最为广泛的一种方法就是线性规划法，其应用于化学、国民经济核算、以及军事等领域。线性规划法可以给我们提供清晰明了的路径使我们在利用现有资源时更加充分，从而使企业的效益最大化。依据这种想法，通过分析烧结原料价值，确立优化目标是使生产烧结矿时成本最低，以烧结矿原料供给量和品质指标等为限制条件，进行配料优化时用线性规划的方法，在很大的范围上去寻求烧结生产成本最低，达到降低生产成本的目标。

烧结配料的研究越来越细致深入<sup>[6]</sup>，烧结矿的物理和化学性质还有冶金性能对烧结矿的质量都有影响。在考虑加入烧结矿的物理和化学性质以后，模型的变量和限制条件数目增加，优化模型的计算量会变大。因此，遗传算法、非线性规划法、蚁群算法被大规模的应用在烧结配料模型优化中。

用线性规划法可以得到烧结配料原料的最优组合，但是钢铁厂商所能买到的铁矿石的质量好坏不一，往往买到的铁矿石的物理和化学性质不是很稳定。这就要求设计优化烧结配料方案时要充分的考虑原料的成本，还要充分的使用现有的原料资源。钢铁厂商在这种情况下就要结合钢铁厂的实际情况充分的考虑原料的成本和原料使用情况，遗传算法可以为钢铁厂铁工更优的配料方案，在提供配料方案这方面，遗传算法要比线性规划法提供的配料方案好的多，遗传算法可以综合各个配料的优势来达到整个配料模型最优，也可以使钢铁厂商炼钢成本有所降低。

线性规划的方法在优化烧结配料模型中十分的不智能，会受到很多方面因素的影响，解决的问题也十分的有限<sup>[7]</sup>。在烧结配料模型中碱度的约束问题十分的复杂，首先要引用罚函数将碱度的复杂约束变得简单些方便求解。BP 神经网络、

遗传算法以及差分进化算法可以解决线性和非线性问题。用 BP 神经网络进行优化配料时,用一些数据把烧结模型中的一些系数预测出来,然后在用这些预测出来的系数建立配料模型,最后优化出想要的结果。用遗传算法这种智能算法优化烧结配料模型的,配料建立的模型不必很复杂就可以求出结果,通常为求目标函数的大小值,比通常用的方法便捷,惩罚函数可以有效的调整控制约束条件的先后顺序,确保得到的解是最优的<sup>[8]</sup>。粒子群优化烧结配料的过程比较简单,只需要用罚函数对烧结配料的约束条件改进处理以下即可,这样得出的结果不仅可以满足工艺指标的要求,还可以达到降低成本的目标。运用蚁群算法优化烧结配料模型时易陷入局部最优,但是蚁群算法可以处理一些线性规划法无法处理的复杂问题。果蝇算法、模拟退火算法等先进的智能算法被大量的应用于烧结配料模型的优化中,为优化烧结配料模型提供了更广阔的思路<sup>[9]</sup>。

目前,国内外在优化烧结配料模型时把物理和化学性质的标准引入,这样一来约束的限制条件变多,变量也会随着增加,优化模型变得更加复杂,对烧结矿品质的要求也在不断的提高<sup>[10]</sup>。随着烧结配料的数学模型变得复杂,原来被广泛应用于烧结配料模型中的线性规范法逐渐被一些先进的智能算法所取代。蒙特卡洛法被淘汰就是一个很典型的例子,蒙特卡洛法在处理简单的线性规划问题还比较实用,一旦变量增加模型变得复杂则算法越加显得迟钝,求解的结果不能很好地达到既定的要求。遗传算法和粒子群算法在惩罚函数的帮助下可以很好地处理复杂的烧结配料模型,在烧结配料模型中得到了广泛的应用<sup>[11]</sup>。

## 一、粒子群算法

### 1. 粒子群简介

上个世纪八九十年代,很多科学家开始通过模拟自然界生物的日常行为来解决一些复杂的计算问题,这是一种计算创新,形成了以生物活动智能的一种新型优化算法。粒子群算法是模拟生物智能算法的典型代表,通过模拟鸟类的生活习性以及运动规律来求解复杂的计算问题,这种算法得到了学术界越来越多的关注。Eberhart 和 Kennedy 在上个世纪 90 年代中期开发的一种算法就是粒子群算法<sup>[12]</sup>,假设单个一个粒子没有质量、密度和体积,在为粒子规定一些限制条件,这样当大量粒子聚集在一起时就可以解决复杂的问题。粒子群算法被提出之后,被很多国际会议列为主要讨论的问题<sup>[13]</sup>。

## 2. 粒子群算法的基本思想

粒子群算法是基于生物群体的优化智能算法。Reynolds 在观察鸟类飞行的过程中,一个鸟只会遵循它有限邻居的飞行轨迹,但是当大规模鸟类在一起飞行时好像处在同一个控制中心之下,单个鸟类简单的运动的相互作用会使整个群体运动变得复杂<sup>[14]</sup>。通过对大量鸟类捕食的规律发现,如果在一个地域有食物,那么在这个地域最快的捕食方法就是搜寻离食物最近鸟类附近的区域。粒子群算法就是模拟鸟群这种捕食方法来优化复杂的数学问题。另外,一些基于社会经验的决策也可以看做是粒子群算法的基本思想。群体会聚集单个个体的简单特征,最终呈现出比较复杂的特征,科学家应用一定的规则将这些特征在计算机中进行模拟。。例如,Reynolds 使用了以下的三个标准作为简单特征的规则<sup>[15]</sup>:

- (1) 向远离最近同伴的方向移动;
- (2) 向他们寻找食物的方向移动;
- (3) 向整个同伴的中央运动。

这就是学术界比较有名的鸟类模型。在一个群体中,这个群体中的任何一个个体都必须遵循以上三条标准,通过这三条标准就可以用计算机模拟整个动物种群的运动规律。粒子群优化的基本思想就是这样的。粒子群中的每个个体运动都可以用以上几条规则简单的约束并且很容易实现,所以粒子群这种算法越来越得到学术界的重视。粒子群是这样模拟生物界场景的:很多饥饿的鸟在寻找食物,而在这个地域里只有一个地方有它们想要的食物,但是鸟群却不知道这个地方在哪里,但是有食物的地方离他们鸟群有多远它们是知道的<sup>[16]</sup>。那么怎样才能快速的找到食物呢?其实只要在离食物最近的鸟的附近周围寻找食物即可,这种寻找的方式也是最快速便捷的。粒子群算法就是按照这样一种思路模拟生物界物种特征来解决复杂的数学难题的。这种算法可以放在一个四维的空间里,放在四维的空间里我们想象着把粒子定义为任意一个问题的最好的方案。它以一种速度在向目标值前进,但是速度会变,这个速度的变动规律就要靠经验了,靠自己的经验和其他伙伴的经验都可以。粒子都有相应的值,把解带到目标函数中就可以计算出来,并且知道自己从寻找食物开始到现在离食物的最近距离以及自己所处的当前位置。这个是粒子本身自己的经验。除此之外,每个粒子还知道整个群体中每一个个体从开始寻找食物到目前为止距离食物最近的位置。这是和他一起寻找食

物个体的经验。则每个粒子要按照如下规则改变自己的当前位置坐标。

- (1) 当前坐标;
- (2) 当前飞行速度是多少;
- (3) 当前自己的位置与离食物最好的位置之间的距离;
- (4) 当前位置与群体最好位置之间的距离。

初始化随机生成一堆粒子组成粒子群，然后每个粒子依据以上规则开始搜索，依据以上规则不断的改变自己的坐标，不断的迭代往复，这就是粒子群智能算法的基本原理。

## 二、模拟退火优化算法简介

### 1. 算法概述

在军事，经济以及医疗的许多实际问题中，这些领域的目标函数和优化条件是及其复杂的，易存在局部最优解，当约束条件和变量增加时需要优化问题的规模也同时增大<sup>[17]</sup>。因此，有效的求解出复杂目标函数的最优解一直是一个难题。一般有两种方法可以用来求解全局最优，一种是确定性求解法，另一种是随机求解法。确定性求解法可以求解一些简单的问题，并且这些问题又具有一定的特征性，而随机性求解法易陷入局部最优，求出的最优值是局部的最优值而非全局最优。

模拟退火算法是一种应用概率进行搜索的智能算法，在一个很大的搜索解空间中来搜索全局的最优解<sup>[18]</sup>。在上个世纪中期，一位外国的教授第一次提出模拟退火这种智能算法。上个世纪 80 年代初 Kirkpatrick 将 SA 这种算法引入了数学中的优化问题。这种方法对于解决 NP 问题效果十分的明显，搜索的最优值可以达到全局最优，没有很依赖初始产生的值。已经在军事、医疗、航海及优化领域得到了广泛的认可。

### 2. 算法基本步骤

- (1) 令  $T = T_0$ ，用计算机产生一个初始解，这个解运用随机发产生即可。 $T_0$  就是模拟退火算法的初始温度，然后计算相应的适应度值  $E_0$ 。
- (2) 冷却进度中的下一个值是  $T_i$ ，令  $T = T_i$ 。
- (3) 依据目前解  $x_i$  进行扰动情况，产生一个全新的解  $x_j$ ，，计算目标函数

值  $E(x_j)$ ，得到  $\Delta E = E(x_j) - E(x_i)$ 。

(4) 若  $\Delta E < 0$ ，则接受这个新解  $x_j$ ，作为目前的解；若  $\Delta E > 0$ ，则新解  $x_j$  按概率  $\exp(-\Delta E / T_i)$  接受， $T_i$  为目前的温度。

(5) 在温度下，不断的重复扰动和接受新解的过程。即执行步骤 (3) 与 (4)。

(6) 判断是否达到最优值标准，是，则结束算法；否，则转到步骤 (2) 继续执行。

模拟退火算法实际上分两层运行，在任何一种温度下可以通过扰动产生新的解，然后在计算目标函数值，看目标函数值是否符合条件再决定接受或者不接受。通常情况下模拟退火算法的温度初始值较高，这样在扰动过程中产生的新解被接受的概率增加<sup>[19]</sup>。在低温时模拟退火算法接受解的概率已经很小了，但还是存在一定的概率去接受更差的解。因此，通常都会把模拟退火整个运算过程中的最优解保存下来与达到终止条件时输出的最优解一并输出。

### 三、差分进化算法的简介

#### 1. 差分进化算法

上个世纪 90 年代中期，有两位外国的学者提出了差分进化算法。这个算法的应用的领域比较广泛，尤其是在求解一些实数的复杂难题上<sup>[20]</sup>。这个算法是在一些群体之上去搜索全局最好的解的一种算法，它的一些优点是构建的结构不是很复杂，收敛到最优解的速度比较快，不易受其他因素影响，因而这种算法被广泛的应用在一些复杂的领域，例如：军事、经济和医疗卫生等领域。再一次国际的赛事中证明了这种算法的计算效率是比较快的<sup>[21]</sup>。

像其他拥有智慧的算法一样，这个算法同样也拥有智慧，也是根据自然界生物进化过程的一些特点来发明的。一个群体当中的任意一个个体之间通过合作及竞争产生优化过程中的搜索方向<sup>[22]</sup>。这种算法的原理是：要产生一个种群并且这个种群是用随机法产生的，在这个种群中任一两个个体要做向量差，两个个体得出的向量差再加上一个个体，就得到了一个崭新的个体，这个崭新的个体分别与群体中的每一个个体相比，如果现在的个体值没有好于崭新那个个体的值，就用崭新的那个个体代替现在这个个体，否则用当前的个体继续计算<sup>[23]</sup>。按照这种规律不断的淘汰不符合要求的个体，保留更加优良的个体，将解向最优的方向引进。



在上个世纪 90 年代后期 Price 和 Store 建立了一个关于差分进化算法的网站,一个目的是为了更多的学者了解这个算法,另一个目的则是为了获得学者的关注与支持,这个网站建立以后为研究差分进化的相关学者提供了极大的方便,他们可以自由地在网站上进行交流学习<sup>[21]</sup>。另外,Price 和 Store 具有无私奉献的精神,并没有把差分进化算法据为己有,也没有申请专利,这也为促进差分进化算法的发展起到了关键的作用。

## 2. 差分进化算法步骤

其具体进化流程如下。

- (1) 首先要把控制差分进化算法的参数定下来,在确定目标函数值。
- (2) 产生一个用随机法生成的初始种群。
- (3) 计算初始种群中任意一个个体的值。
- (4) 设计出符合问题终止的条件,这个条件可以是相应的进化代数。若达到了终止条件,算法停止进行,否则,算法继续进行运算直到达到终止条件时为止。
- (5) 进行交叉和突变操作,得到算法的过程种群。
- (6) 在旧种群和过程种群中选择个体,产生新一代种群。
- (7) 进化代数  $g=g+1$ , 转步骤(4)。

## 四、蒙特卡洛法的简介

在上个世纪 40 年代初期,有两位外国科学家提出了一种新的智能算法,这个算法的名字叫做蒙特卡洛法,这两位科学家当时都在研究核武器<sup>[25]</sup>。这个算法的名字是这两位科学家用美国的一个城市的名字来进行命名的,就这样这种算法被披上了一层神秘的面纱。其实很早以前就有人提出这种算法,只不过是推广而已。

### 1. 蒙特卡洛法步骤

#### (1) 构造或描述概率过程

像鸟类运输问题本身就是具有很大随机性的复杂问题,主要是模拟和描述整个运输的过程,但是对于本来不是随机性的问题如莱布尼茨定理,就必须认为的构造一个概率过程,所求得问题的最优值往往就是他其中的某些参量。我们可以把不确定的问题变成确定的问题<sup>[26]</sup>。

## (2) 实现从已知概率分布抽样

每一种概率分布的构成可以作为概率模型的基础,构造的概率模型要遵循概率分布,粒子群算法实验的基础就是已知概率分布的随机向量,这也是粒子群算法为什么被称为随机抽样的原因。一种方法是用数学计算公式产生。这样产生的随机序列,与真正的随机数序列不同,所以称为假随机数,或假随机数序列。这种假的随机数可把它当做真的随机数来看待使用,科学家们用了多种方法对它进行测试,得到的结果与随机数列比较像<sup>[26]</sup>。

## (3) 建立各种估计量

一般的情况下,科学家会事先建立一个概率模型,然后从这个概率模型中进行抽样<sup>[27]</sup>。我们事先要给定一个问题的解,这个解就要我们确定一个随机变量。建立无偏估计量,这相当于把模拟实验结果登记造册记录下来,从中选取最优解。

# 1.3 基于人工智能的专家系统配料的研究

在烧结配料的生产过程中,人工智能系统对这个生产过程十分的重要,运用BP神经网络对烧结矿配料的物理化学性质进行预测,为配料室的工作人员提供相应的指导<sup>[28]</sup>。构造一个数学模型要依据烧结配矿的理论和钢铁厂生产过程中所产生的数据,采用定量和定性的综合方法,将规则模型和数学理论模型相结合,提出计算烧结配料配比的优化算法。专家系统及模糊技术是模糊专家系统的基本原理,运用二者指导烧结配料的配比,与其他的专家系统相同,都是结合数学模型来进行使用。当模型得出的最优值无法达到要求时就需要引入修正模型,依据品质指标的设定值和实际测得数值的误差,重新确定一个比较合理的配料方案,品质指标的设定值和实际测得数值的误差可以为重新构建方案提供信息。

专家系统与BP神经网络相结合作为神经专家策略的基础,这两种方法指导烧结配料原料进行配比。依据钢铁厂生产的实际数据和烧结的基本思想,建立了由BP神经网络与数学模型相结合和计算模型,运用专家经验得出模型,这种模型是根据专家得出的定量的方法确立的,得到了优化烧结配料模型的一种新的方法。这种方法已经被很多的钢铁厂商应用到碱度的配比当中,结合钢铁厂产生的以往数据和专家的经验<sup>[29]</sup>,建立了一种稳定、可靠及高效的数学模型,B.P神经网络这种优化算法可以描述出一些非线性的关系,比如炼焦的非线性和碱度的非

线性，构成炼焦和碱度的预测模型，提出了一种新型的优化烧结配料的方法。

把数学的模型和专家的经验相结合在一起就是研究炼焦配料的关键内容，然后在运用专家系统不断的改变烧结所需的配比。这要求按照非常严格的标准进行配料，如果工艺要求和生产过程发生变化，将会增添更多的标准或者增添的标准与原来的标准产生冲突，白白的浪费人力、物理，这样会增加钢铁厂的经济负担[30]。

## 1.4 论文的主要内容及成果

论文研究的重点主要集中于遗传算法、多种群遗传算法优化烧结配料建模并且比较这两种算法得出的烧结成本高低，其中重中之重就是论文提出了一种精英保留策略的新思路来改进遗传算法<sup>[31]</sup>。遗传算法是本世纪十大智能算法之一，对于解决非线性函数有着很好的寻优性，但是再好的优化算法依然存在着缺点就像人一样不可能有十全十美的人，这也从侧面衬托出遗传算法是依照生物进化的角度而衍生出来的。论文第一章写了引言说明了论文的写作背景及意义。第二章介绍了遗传算法，这种优化算法虽然简单易操作效果比新型智能算法差一些但它作为解决非线性规划的鼻祖本文作者觉得很有必要在文中介绍一下并用它实际解决一下烧结配料建模问题。第三章着重介绍了一下多种群遗传算法，介绍了多种群遗传算法的产生及其怎样弥补传统遗传算法的缺陷，解决传统遗传算法过早收敛问题并且清楚地解释了多种群遗传算法的构造原理最。第四章介绍了学术界改进遗传算法的一些方法并提出一种新型精英保留策略改进遗传算法的方法用于解决传统遗传算法易陷入局部最优情况<sup>[32]</sup>。第五章给出烧结配料数据并建立数学模型，应用传统遗传算法和多种群遗传算法分别对模型进行优化比较分析优化后的结果。

论文的主要成果有两方面：一方面是把多种群遗传算法成功的应用在优化钢铁烧结配矿领域，另一方面是成功提出了一种新型精英保留策略用于解决传统遗传算法局部最优问题。在遗传算法中算法的初始种群选取非常重要，它可以影响算法的计算时间、局部与全局的搜索能力以及，以及最后的计算效果。烧结配料建模的目标函数异常复杂，使得遗传算法心有余而力不足，运行变得缓慢，极易陷入局部最优，寻找全局最优解的能力下降。本文针对此问题提出了一种新型精

英保留策略，它是把遗传算法每一代种群的最优解保存下来，保存在一个精英储存库当中，然后当满足终止条件时再从精英储存库当中选取最优解输出，这样输出的最终解为整个搜索过程的最优解。另一个主要成果就是本文的视角创新多种群遗传算法在工程信息类领域应用较多，在优化烧结配料建模中应用较少，本文鉴于此把多种群遗传算法应用到求烧结配料成本中并和传统遗传算法进行对比，突出了新型智能算法的优越性。

## 第2章 遗传算法

### 2.1 人工智能算法的概述

现代社会的三大支柱是大数据、信息的快速传播以及知识无国界的交流，交流、传播和大数据是现代社会的关键词。知识与信息和数据的概念并不是相同的，尽管数据和信息一直支撑着知识<sup>[33]</sup>。美国一位著名的科学家在其一本经典的著作中解释道：数据本身包含的使用价值有限，尽管数据简单、清晰及明了。例如，某个企业可能有数据表明其销售的产品中有 30% 销往上海，要想使这个数据具有使用价值，就必须把这个数和其他数据综合联系起来，如“这些产品被 30~40 岁的中年男女买走”，将其转换成可以使用的信息。信息的含义就是与其他的数据相联系后进行综合的分析，使其具有极其重要的使用价值。知识就要比信息更高级一些，它是指将一些信息与另一些信息相联系后通过分析得到的一种结论。知识与数据和信息进行对比，知识总是包含着数据和信息中的共同的特点，同时也比数据及信息更加智能化。书是信息的载体是知识的源泉，但是书必须被人类汲取之后才可以将书上的信息变为知识。书上的一些知识是在社会已有的信息上成立的。书上的某些知识是建立在人类已有的经验、理解和分析的基础之上的，它包含着人类认知和行动的基本信息。

智能领域中一个重要的分支就是智能算法。人们生活在知识和信息高速发展的新世界李，信息的快速传播和知识无国界的相互流通是智能算法产生的关键原因，用数据分析与数据挖掘从人类的各种生活习性以及消费习惯中提取出有价值的信息和知识，为公司提供生产和销售产品方向，以及从以往公司销售的大量数据中运用数据分析与数据挖掘过滤出有价值的知识和信息以便能提高公司的效益，或者用电脑代替人脑建立复杂的数学模型。怎样从大量的数据中获取到你想要的有价值的信息是一项艰难、复杂和费时的的工作。为了让从事数据分析的人员不那么劳累同时也为了减轻他们的劳动量，出现了一些具有人类分析能力的算法，这些算法可以代替人类进行工作与分析，方便、快捷和实用，这些算法被统称为机器学习。不论是果蝇算法还是 BP 神经算法，也不论是差分进化算法还是粒子群算法，更不论是禁忌搜索算法还是贪心算法，本质上都是机器学习。机器

学习最显著的特性就是可以模拟人脑进行机械的运算。中国的一名院士在其出版的一本著名著作中曾经提到,数学是一门非常机械的学科,数学的机械性体现在其演算和证明的过程中,再得到结果之前每一步都是确定的,一步一步的向结论走去,整个过程都是有规律可循的<sup>[34]</sup>。20世纪70年代吴院士首先创立了用电脑将解析几何转化成向量进行证明。这个例子应用成功之后很好的体现了数学是一门机械的学科。由此可见,数学机械化、机器学习以及智能算法在本质上都有共同的特点,可以被看成一个问题的多个方面。

智能算法实际上就是生物科学,是一门多学科交叉的科学。各个学科很好的融合在一起。每一个学科的交集往往将来会有很大的发展。马克思曾经在《自然辩证法》里就说过:“化学家以及物理学家有人为自己在原子和分子科学的交叉点中无能为力,然而他们对这个点抱有很高的期待,期待能收获理想的成果”在这个交叉点上有时会形成一门全新的学科,是新学科的培基,是最具有发展前景的地方。比如,燃气灶上的火焰外焰的温度始终要比内焰高并带有淡蓝色,焰心温度比较低;台风的中央风力级别为零,但是外边周围的风力最大;靠近水壶边的水先沸腾;电荷会在铁塔的尖端集聚。随着社会的快速发展,智能算法的发展速度也十分的迅速,不一样的智能算法之间,相互借鉴各自的长处来弥补各自的短处,这样不同算法之间的差别就会变小。

## 2.2 遗传算法的研究现状

20世纪90年代,遗传算法发展速度大幅加快,进入了蓬勃的发展时期,无论是实际的应用还是理论的发展都是学者愿意研究的课题<sup>[35]</sup>。最为显著的就是遗传算法实际应用的行业,很多行业都应用遗传算法,这些行业促进其发展极为迅速。遗传算法的优化效果也显著的提高了,同时也在不断地摸索着去适用更多的产业。另外,一些新的方法和思想在学术界得到了快速的发展及应用。遗传算法应用的领域从最初的线性和非线性规划领域扩展到军事、经济和医疗等领域。随着遗传算法应用行业更加多样性,遗传算法有几个新的发展方向引起了科学家的关注,一是这种算法被看做是及其学习的一种,这一新的方向把遗传算法从历来求解非线性规划中扩展到具有独特规则的机器学习中。这一新的发展为智能领域中知识的获取和过滤等难题带来了一丝的火种。二是这种算法在和别的智能算法

混合在一起进行发展，不断的吸收别的算法的长处补充自己算法的短处，这对智能算法将来的发展具有极其重要的意义。三是科学家在大力的发展遗传算法，使其成为更好的优化工具，但是这种算法的劣势就是容易在局部的最值附近徘徊，不能保证得到的最优解是全局最优，改进之后是遗传算法的结果更加精确，更具有使用价值<sup>[36]</sup>。四是遗传算法不断汲取人工生命领域中的精华，快速发展壮大。人工生命科学就是用电脑去模拟大自然各种各样的生物活动现象，其中生物的适应性、不断进化和免疫机制是人工生命研究的主要对象，而遗传算法在这些方面将会发挥很强的作用。五是遗传算法与进化策略和规划等先进理论加大融合。这两种算法几乎是与遗传算法在一个时期发展起来的，他们的基本思想也是生物的进化思想，遵循着“物竞天择，适者生存”的道理，但是三者之间不完全一样，有相同点也有不同之处。当下，越来越多的学者对这三者之间的结合与比较感兴趣。

20 世纪 90 年代初，D. Whitey 再一次学术会议上提出交叉算子的概念，这个概念的提出释义领域交叉为基础的，交叉效果很好，并将它用于解决其他的复杂难题上，通过大量试验检验其准确性。D. Hackley 等著名教授提出了爬山算法，此算法也是模拟自然界生物的习性来除了复杂的问题的，爬山算法模拟了 N 个人来进行投票，共同投出新的个体值（N 表示种群大小）。大量的实验表明爬山算法与传统遗传算法、粒子群算法相比在计算速度方面具有更好的性能，得出的解更加优化，用五个函数来进行验证都表现出更优的效果，总的来说爬山算法也比学术界其他的算法在优化问题是更加节省时间。有两位国外的科学家把遗传算法和其他的简单方法放在一起。与此同时，有的学者还将三个解的交叉算子与其他交叉算子进行了比较，实验结果显示，三个解的交叉算子比其他的交叉算子性能更好一些。

在 20 世纪 90 年代初，英国一个大学的教授在指导博士生做实验时，将遗传算法的编码形式进行了扩展，这样可以拓展算法的应用行业<sup>[37]</sup>。在二十世纪八十年代后期，有的科学家开发了传统遗传算法计算机程序，这个计算机程序一经面世收到了学者的一致好评，此程序也是世界上开发最早的遗传算法计算机程序，第二届遗传算法编程国际会议在格拉斯哥进行召开，会议上讨论了遗传算法编程的基本原理与性能。遗传算法的编程是由斯坦福的两位教授发明的，充分的探索

了传统遗传算法与 BP 神经网络结构相结合寻优的优点，超越了日常生活中的一些限制。

中国也有很多的科学家对遗传算法的变异算子进行改进<sup>[38]</sup>。到了 21 世纪，戴小明教授引入了多种群遗传算法的思想，对于不同的种群采用不同的遗传方案，例如交叉概率，采用不同的交叉算子来进行优化，多种群遗传算法的种群之间可以不断的进行信息之间的交流，直到满足输出条件为止，并利用信息交流跳出局部收敛的包围圈，解决了遗传算法容易在局部最值附近徘徊的问题。

在 21 世纪初，赵教授针对传统遗传算法在运算较大规模组合的问题上计算效率并不高的现象，提出了一种新型的并型遗传算法，这种新方法采用了在基因上编码的手段。以短染色体作为基础，在对一些短染色体进行重新编码作为初始群体。

在 21 世纪初，江教授对并行遗传算法求解背包问题时所出现的问题提出了解决的方法，为了保持遗传种群的多样性要采用不同的策略，这样做的好处是可以防止遗传算法过早的徘徊在局部最值镀金，可以更好的寻找到全局最好的解。

## 2.3 遗传算法生物学基础

在非洲的大草原上存在一群斑马还有一群狮子，狮子在饥饿的时候就会去寻找斑马来填饱自己的肚子，狮子在不断的追足斑马，斑马在不断的奔跑，在斑马群体中，总会出现一些体弱多病的个体，这些个体就称为了狮子的盘中餐供给狮子填饱肚子，种群中剩下的其他斑马，即没有被狮子吃掉的斑马，大都身体健康十分强壮。他们之所以幸存下来就是因为他们的奔跑能力十分的强悍，这样剩下基因优良的斑马不断的产生基因更好的斑马，也就是说会产生奔跑能力更强的斑马。这样斑马的整个种群都向着更优的方向进行发展，整个群体达到了进化。再来看狮子种群，在狮子种群中有一些奔跑能力较好的个体，还有一些奔跑能力差的个体，奔跑能力差的个体会抓不到斑马就饿死了，那些抓到斑马的狮子就继续存活下去，这些存活下来的狮子就继续产生基因更加优良的新狮子。这些新的狮子都是奔跑能力很好的狮子，这样狮子种群就达到了进化目的，狮子种群和斑马种群都在不停地进化，狮子和斑马的故事是生物进化很好的实例。

斑马和狮子不断进化的过程就是大自然不断发展的过程，其基本的思想就



是：大自然生物不断发展的原因就是生物基因不断向好的方面进行改进，因为大自然中的所有种群构成大自然中的生物。物种基因的变化有几种情况包括重组、突变和相互隔离还有选择，生物经过这样的一系列的变化不断的向更好的方向发展，最终产生更新的种群。在生物的基因向更好的方向发展过程中，基因重组和突变是极其关键的，它们决定着生物向哪个方向进行发展。不一样的动物之间是不可以进行生物遗传的<sup>[39]</sup>。

形成一个全新的物种的方式有两种：一种是逐渐生成式，另一种是突然发展式。逐渐生成式主要经过重组、突变生成第二物种，再由第二物种形成很多的新物种；突然发展式不需要经过第二物种这个阶段，它的发展有三种，杂交染色体生成新的物种和多倍体新物种记忆这两种物体的杂交的个体。传统的遗传算法体现了逐渐生成式和突然发展式两种思想。

## 2.4 遗传算法的发展过程

遗传算法起源于对大自然生物的观察，后来很多科学家都用计算机编程技术模拟实现了遗传算法。在 1940 年左右，就有科学家利用计算机科学技术模拟自然界生物的进化，他们用电脑模拟着生物进化的过程。

早期遗传算法的研究目的是为了解决一些复杂的数学问题。J. H. Holland 教授是最早发现遗传算法的，他发现遗传算法可以代替人脑工作并且具有像人类一样的智慧。在 20 世纪 60 年代中期，一位外国的教授提出了智能代替人脑的重要性，并将智能应用在计算系统中。在 20 世纪 60 年代的后期，H 教授的一位博士生在其毕业论文中“遗传算法”这个名词在这篇文章中出现了，这是遗传算法首次的为世人所知。这位博士还发表过关于遗传算法在实际领域中应用的论文，这是学术界首次有人发表关于遗传算法应用的相关论文，从而使得自适应遗传算法应运而生。在 1970 年左右，有一种关于遗传算法的定理形成了，这个定理就是模式定理，它是由外国的一位教授发明的。种群中好坏的样本个数将以指数的形式进行增长。同一时期 K 博士在其毕业论文《遗传自适应系统的行为分析》中做了很多的优化函数值的计算并且结合了模式定理思想。在这篇博士论文中构建了遗传算法的体系与框架，为遗传算法的将来发展打下了十分牢固的基础，这篇论文中得出了很多具有实际价值的结论，至今为止还有很多的学者在使用这篇论文

产生的理论。分类器系统在 1980 年左右被提出,这种新的概念是以遗传算法为基础的。在 20 世纪 80 年代后期,《搜索、优化和机器学习中的遗传算法》一书问世。该书把学术界研究遗传算法的成果梳理了一遍,构建成一个体系,论述了遗传算法的应用领域与基本的思想。可以说这本书为现今遗传算法奠定了基石,也吸引了很多研究遗传算法的科学家的目光。在 20 世纪初,《遗传算法手册》一书问世了,它是由 L. Davis 所撰写,这本书中包括了遗传算法在很多领域中应用的大量样本的数据,为遗传算法的发展和推广起到了很大的关键应用。遗传规划在上个世纪后期被提出,它是由, J. R. Koza 提出的,在优化计算机程序的过程中引入了遗传算法并<sup>[40]</sup>。

### 2.4.1 遗传算法的应用

遗传算法是一种具有强鲁棒性的算法,可以为求解复杂模型提供一种通用的规则,它可以应用各个问题的领域,不会受到领域的限制。所以很多的学科都可以应用遗传算法来进行优化。遗传算法主要应用在以下这些领域。

(一) 遗传算法主要应用在优化函数的领域,学术界常常应用一些函数来测试遗传算法的优化性能。许多学者用各种各样的复杂函数作为测试遗传算法的测试函数。有发散型函数也有约束型函数,有凹凸函数也有 N 维函数,有定性函数也有随机函数。用这些复杂的函数测试遗传算法的性能更能凸显出遗传算法的性能。遗传算法在求解线性规划和非线性规划方面可以得到很好并且有效的结果。

(二) 在现如今的生产生活中复杂的问题越来越多,限制问题的条件也越来越多,这样在优化问题时问题的解空间也在不断的扩张。问题变得复杂求解不易,以前的简单问题用枚举法就可以解决,但是现在复杂的问题用枚举法效果甚微。对于这类极其难解的问题很难精确地求解,但是我们可以求一个令大家都满意的解,而遗传算法这种工具为之提供了可能。经过很多的实践证明,遗传算法已经在很多问题上得到了成功的运用。

(三) 生产调度问题是一种比较复杂的数学问题,需要考虑到很多的方面,比如原料的成本还有人工成本,这种问题建立起来的数学模型会有很多的限制条件,即使用罚函数把非线性约束条件变成线性约束条件后,所求到的解也会与实际的结果相差很大,目前,一些调度工作主要依靠一些有经验的工作人员来完成。

自从遗传算法兴起以后在解决调度问题上就多了一种方法，同时也省时省力。

（四）自动控制领域是一个非常复杂的领域，在这个领域中有很多复杂的数学模型需要求解与优化。自从遗传算法发展与兴盛之后，有些学者就把遗传算法引入到自动控制领域中进行优化模型，优化之后的效果比较好。例如用遗传算法优化导弹的控制系统，使用遗传算法优化潜艇的控制系统、基于遗传算法的航天控制系统、基于遗传算法的轮船控制系统、基于遗传算法的火车控制系统、基于遗传算法的军事指挥系统、基于遗传算法的医疗卫生系统、基于遗传算法的 BP 神经学习算法。都显示出了遗传算法有很好的优化性能。

（五）机器人的发展极为迅速，现在很多的工厂都应用机器人来代替人进行工作，而遗传算法最开始就是模拟人工自适应系统发展起来的。因此，机器人顺理成章的成为算法应用的一个行业。例如，遗传算法已经在机器人的很多方面提供了优化的可能，遗传算法优化机器人如何行走，遗传算法优化机器人如何协调各个方面的工作在工厂里，遗传算法如何优化机器人的大脑。

（六）在学术界研究的领域中图像处理是一个备受关注的领域。在处理图像的过程中，例如激光扫描，提取图像特性。图像分割法是一种很有效的方法，但是再好的方法也会存在偏差，从而影响生成图像的效果。怎样可以把这些偏差尽量的缩小并且降到最低，是产生的图像能达到计算机要求的效果。遗传算法在优化以及处理图片中体现出了很强的实用性。

（七）电脑模拟自然界生物的生活习性以及行为规则构建了一个具有生命力的自然系统，依据行为规则和习性构建人工生命系统。人工生命具有两大特点，第一是自学习能力比较强，另一个就是自组织能力比较强。遗传算法是依靠人工生命发展起来的，因此他两有着紧密的联系。遗传算法理论模型是人工生命的重要的理论部分，支撑着人工生命的发展。当人工生命还处在发展的初始期时，遗传算法已经在，理论模型、数学模型、学习模型以及优化模型等模型方面显示出了一些实用的能力，遗传算法将来的发展前景必然很可观。遗传算法是人工生命的基础，人工生命又促进遗传算法进行发展，二者互相取长补短，遗传算法的发展会更加理想。

（八）在 20 世纪 90 年代后期，国外斯坦福大学的一位科学家提出一查 UN 撒 UN 发编译程序的概念。编程中心思想是：在计算机编译程序中用大树的形状

来表示程序,运用物种不断进化的思想,让电脑自己产生程序去解决复杂的难题。尽管遗传算法编程的基本思想和理论现在还不太成熟,应用的领域也不是十分的广泛,但是人工智能和机器学习等领域应用遗传算法的次数在逐渐增多。现在遗传编程有很多,但是大多数都不公开,公开的系统也就是 Koza 发明的那么几个

(九) 学习能力是很高级的能力,学习是一个很智能的过程,是人类所特有的一种功能。遗传算法中很多的行业都得到了相应的运用,尤其是机器学习中的分类器系统。例如,利用遗传算法学习函数怎样被优化;怎样才可以更好地使模糊系统变得更加智能;把遗传算法和 BP 神经网络相结合调整阈值等。

(十) 数据挖掘是一种比数据分析工作难度更大的一项工作,是最近几年才兴起的技术,随着社会的不断的发展信息量也随之增加,引入数据挖掘这种工具,我们可以再海量的信息中提取出我们想要的信息。数据挖掘这种新型挖掘数据的工具可以从海量的信息中挖掘出公司或者个人想要的有应用价值的信息,挖掘出隐藏在数据背后的潜在价值。其实数据挖掘问题实质上就是一种寻找问题,在寻找的空间中,挖掘出解决问题的策略。在遗传算法中应用随机数产生初始种群,并用数据挖掘去挖掘数据库中潜在的价值。挖掘数据库中的知识与规则,直到挖掘出数据库中隐藏的价值为止。一种新型的数据挖掘工具已被开发出来,利用这种新型的挖掘工具对三辆出事的汽车进行了分析,结论表明 GA 这种算法对于数据挖掘这种工具的利用价值还是很大的。

## 2.5 遗传算法的基本理论

### 2.5.1 模式定理与隐含并行性

自从科学家模拟自然界万物的进化获得一定的效果以来,学术界就要解释遗传算法的内涵,想方设法的寻找一定的思想和理论来完美的解释遗传算法机理。模式分析是从模式定理中演变发展而来的。隐含并行性定理是解释 SGA 的时候由 H 提出来的,模式定理也是那个时候被提出来的。

定义 1 模式:在一个模板上会有一连串的数字,这些数字在某些方面都具有相似性。

定义 2 模式阶:模式的阶的含义就是:能确定整个模式 H 中有多少个位置可

以被准确的确定，记作  $O(H)$ 。

定义 3 模式的定义距：在整个模式  $H$  之间，确定位置是有先后顺序的，模式距就是最先确定的位置到最后确定位置的距离，记作  $\delta(H)$ 。

定理 1 模式定理：在遗传的各种算子的作用下，阶数比较低，矩的定义不长及平均函数值高于群体函数值的模式在下一代中，将以指数形式不断的增长。在考虑遗传算法各个算子的作用下，一个确定的模式，下一代中出现的个体数目可以表示为：

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[ 1 - \frac{p_c \cdot \delta(H)}{l-1} - O(H) \cdot P_m \right]$$

传统遗传算法在一定程度上可以用模式定理来解释其意义，但是模式定理仍然存在不足之处：

(1) 模式定理对其他的编码方式并不适用，只有采用二进制编码时模式定理才会起作用。

(2) 模式定理给出的知识期望值中的一个最小值，无法用它来确定一个算法是否已经达到了收敛的效果。

(3) 算法的参数设计用模式定理来确定其实并没有什么作用，也不会有什么很好的效果。

一位国外的教授非常有道理的分析遗传算法处理并行性的效果，遗传算法处理模式的效率是  $O(n^3)$ ，这一结论揭示了遗传算法处理并行的一种能力，但是现在大部分学者都认为其证明不够严谨与准确。

## 2.5.2 收敛性分析

遗传算法的局部收敛效果好，全局收敛效果就不是很令人满意，但是全局收敛效果又很重要意义重大。在这样一种情况下学者们还是要继续研究遗传算法在全局收敛的时间上的问题，这对把遗传算法应用在实际的领域中意义同样的重要。

一方面是以模式效率分析为基础。从模式的方面看，在模式的空间上寻找最优解的过程就是遗传算法的运行过程，寻找最优解的时间要依据空间的范围和每次遗传寻找的效率。空间范围的大小是由具体的情况所决定的，所以遗传

算法搜索的快慢取决于遗传的快慢，被两个方面所决定，一是每一次遗传产量的新的模式，每次遗传的有效保留地模式，这两个方面是及其矛盾的。一个好的遗传算法应该在这两者之间取中，尤其要把遗传算法运动过程中的动态撇清，这样才会拥有比较高的寻找效率。另一方面运用马氏链来对效率进行分析，遗传算法的寻找效率可以用马氏链来进行数学模拟与分析。每个马氏链就是一个遗传算法的迭代过程，运用马氏链对遗传算法分析得出以下几个定理结论。

定理 2：传统的遗传算法搜寻全局最优解的时候不能以概率 1 进行收敛。

定理 3：如果在每一代的演变过程中，传统的遗传算法都会保存最优的解，如果在一代接着一代的发展过程中，传统的遗传算法都会保存最优的解，并且该算法以另一种变异作为随机算子，对于求解全局最优解的问题中，随着发展的代数没有穷尽，传统的遗传算法收敛到全局最好的解的时候是以概率 1 进行收敛的。

遗传算法中算法过早的收敛和搜索停滞的情况是遗传算法讨论的一个重要的问题。过早的收敛就是经过遗传算法的各种变化使得大于群体平均函数值的模式在下面一代中获取更多的样本，迭代在不断的进行重复，一些模式在种群中有了相应的优势，而经典的遗传算法都会使这种优势变强，此时搜索的空间变窄，表现成向某些一样的数据串收敛的情况。有时在寻找的后段时间，虽然群体是丰富的、多样的，但是种群的最优函数值有可能会接近种群的平均函数值，在这样的现象下，一个群体中已经没有个体在开展竞争活动了，从而寻找最优解的进程也很难在进行了，这种现象就是遗传算法的一种很大的缺点，叫停滞。针对以上的一些问题，解决的方法一般为：

(1) 要加强遗传算法在发展过程阶段的随机寻找能力，就要把突变的概率提高；

(2) 不断的变化，调整，改变选择的概率，优化选择概率时就要像优化参数一样；

(3) 要使群体中的每一个个体尽可能的丰富与多样，如扩大群体的数量，进行优化局部的工作。

遗传算法的优点具体如下：

(1) 在求解比较难的问题时遗传算法的求解稳定性是非常高的，不易受到

其他的因素影响，能很快的适应当前问题并进行稳定求解，比一些确定性算法好处有很多，例如节省计算时间，其作用得到了广泛的肯定。

(2) 能同时优化多个目标。可以很好的掌握搜索发展情况，在应用到实际情况中可以很好的搜索接的空间。

遗传算法虽然有很多优点，但是仍存在着不足：

(1) GA 在搜索局部的过程中能力是有限的，遗传算法可以很好的掌握全局最优解，但是对于局部最优解的问题，通常是不能进行更加深入的发掘。在算法的后段时期，群体中类似的个体增加，由于其调整的幅度不够大，很难再产生更优的解，导致整个算法处在局部最优附近停止。

(2) 容易导致在局部最优解附近徘徊，由于遗传算法操作有一定的规则，使得优秀的得个体急速的发展，在遗传算法寻找最优解的前期中对整个种群的影响是非常大的，对群体中其他的个体影响是非常大的，使得进化速度停止。

(3) 搜索到的最优解不够准确，随机性强。遗传算法就像人类的大脑一样学习能力是很强的，种群通常是用计算机采用俄罗斯轮盘赌的方法产生的，求得所有寻到的解的平均数得出的，音粗，搜索的结果不够准确

## 2.6 遗传算法操作步骤

在 20 世纪 70 年代中期，美国的一位教授首次提出遗传算法这一概念，并出版了一本著作《在自然和人工系统中的应用》。美国的这位教授提出遗传算法是一种并不复杂的遗传算法，还处于萌芽期，还可以继续发展，未来的前景十分可观，这对于美国这位教授来说是一小步，但是对于智能算法领域却是一大步。

传统的遗传算法需要经过很多复杂的过程操作。改进的优种遗传算法即和别的算法惊醒杂交都是传统遗传算法的另一种形式。在遗传算法中，种群就是被经过计算机编码以后的染色体的一个集合，群体中的任何一个个体在生物界中对应的就是生物的染色体。

### 2.6.1 编码

对遗传算法编码有很多形式，但是由于遗传算法是在计算机上进行模拟生物进化行为，计算机采用二进制编码，所以我们在对遗传算法编码时也采用二进制

编码的形式。这样做也方便对遗传算法的过程进行操作，具有极其实用的价值。设一个参数的适应范围是  $(L, U)$ ，使用二进制对染色体进行编码，这个二进制的长度为  $k$ ，则编码的种类有很多。编码和实数的对应关系如下所示：

$$\begin{aligned}
 000000000000000000 &= 0 \rightarrow L \\
 000000000000000001 &= 1 \rightarrow L + \delta \\
 000000000000000010 &= 2 \rightarrow L + 2\delta \\
 000000000000000011 &= 3 \rightarrow L + 3\delta \\
 &\dots\dots \\
 111111111111111111 &= 2^k - 1 \rightarrow U
 \end{aligned}$$

易知：

$$\delta = \frac{U - L}{2^k - 1}$$

## 2.6.2 解 码

解码就是为了将编码中编成的一连串数据编程生活中通用的十进制的数。把算法中的每一个染色体都看成是生物中的每一个染色体，然后再用计算机对他们进行编码，编码为  $b_k b_{k-1} b_{k-2} \dots b_3 b_2 b_1$ ，相应的解码公式如下所示。

$$x = L + \left( \sum_{i=1}^k b_i 2^{i-1} \right) \frac{U - L}{2^k - 1}$$

例如，设有参数  $x \in [2, 4]$ ，现用 5 位二进制数对  $x$  进行编码，可得  $2^5 = 32$  条染色体：

```

00000,00001,00010,00011,00100,00101,00110,00111
01000,01001,01010,01011,01100,01101,01110,01111
10000,10001,10010,10011,10100,10101,10110,10111
11000,11001,11010,11011,11100,11101,11110,11111

```

对染色体中的每一个编码，只要用解码公式，都可以得到我们现实生活中使用的数字，具体解码方式如下



$$\sum_{i=1}^5 b_i 2^{i-1} = 1 * 2^0 + 0 * 2^1 + 1 * 2^2 + 0 * 2^3 + 1 * 2^4 = 21$$

对应参数的  $x$  值为

$$x = 2 + 21 * \frac{4-2}{2^5-1} = 3.3548$$

大自然中的生物具有基因，这个基因可以相对于遗传算法中的编码，而表现性则对应解码。

2.6.3 交 配

交配运算可以一个基因进行交换，也可多个基因进行交换。事先要用计算机随便产生一个数，然后在两个染色体的相应位置互换基因，可以单个基因互换，也可以多个基因互换，最后产生两个新的个体。例如，有两个染色体，交换其后三位基因，如下图所示。

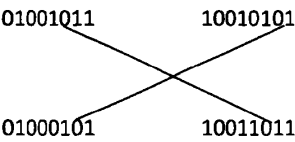


图 1

2.6.4 突 变

突变是整个遗传算法的核心，正是有了突变遗传算法才会不断的进化，遗传算法引入突变的目的就是为了防止算法收敛过早。算法中的突变就好比生物钟基因的突变，在生物的突变种，突变的方向是不确定的，可以向有利于种群的方向突变，也可以向不利于种群的方向进行突变。那么当把生物中的突变引入算法中的时候同样的也存在这两个方面的问题，染色体的基因可以向有利于最优解的方向进行突变，也可以向不利于最优解的方向进行突变。

2.6.5 倒 位

在遗传算法的操作中除了交配，还有突变，除了这两种方法外还有一种鲜为人知的方法就是倒位法。倒位的方法是要在整个染色体中选择一段基因，然后把

这段基因进行 180 度的旋转，这样就可已构成一个新的染色体，因为这个操作改变了基因的顺序。在自然界的生物中，每三个基因组成一个密码子，每一个密码子会控制一种表现型，倒位正好改变了密码子的排列顺序，所以才产生新的个体，也就是新的解。

到位运算看齐汉字的表意就会明白是什么意思，真正操作起来的过程其实和倒位二字的含义是相同、相近的。例如，染色体  $S'$  现在具有的基因和原来所具有的基因是完全一样的，只不过是基因的排列组合发生了一些变化。

$$S = 1001011011101110011010101001 \Rightarrow S' = 100101100101100111011101001$$

### 2.6.6 个体适应度评估

达尔文的进化论表明了一个很实用的观点，在现实生活中人类也是一样的，能适应环境的人往往生活的都很好，不能适应环境的人都会生活的很差。我们用一个一个小的六边形建立起一个三角形，这个三角形具有一定的筛选作用，它的筛选原理就类似于达尔文的进化论。从三角形的最上方投掷一些铁块让他们下落，这些铁块经六边形之间的空格坠落到三角形的底部，如果么有其他的影响，落在三角形中央的铁块是最多，因为能落到三角形中央的空格比较多，所以铁块下落这种方式就像生物的遗传信息遗传到他们的子代一样，目标函数值大的个体有很大的几率被遗传到下一代，目标函数值小的那个个体被下一代选取选取的几率就会小，就不容易被遗传到下一代。

### 2.6.7 复制

复制是遗传算法中一个比较简单的操作，就是依据目标函数值的大小来决定这个个体是否可以被遗传到下一代，如果目标函数值大则遗传到下一代的几率变大，如果目标函数值小则遗传到下一代的几率会减小。若设一个种群中个体的数目是  $N$ ，每个个体  $i$  的适应度是  $f_i$ ，则选取个体  $i$  的概率是

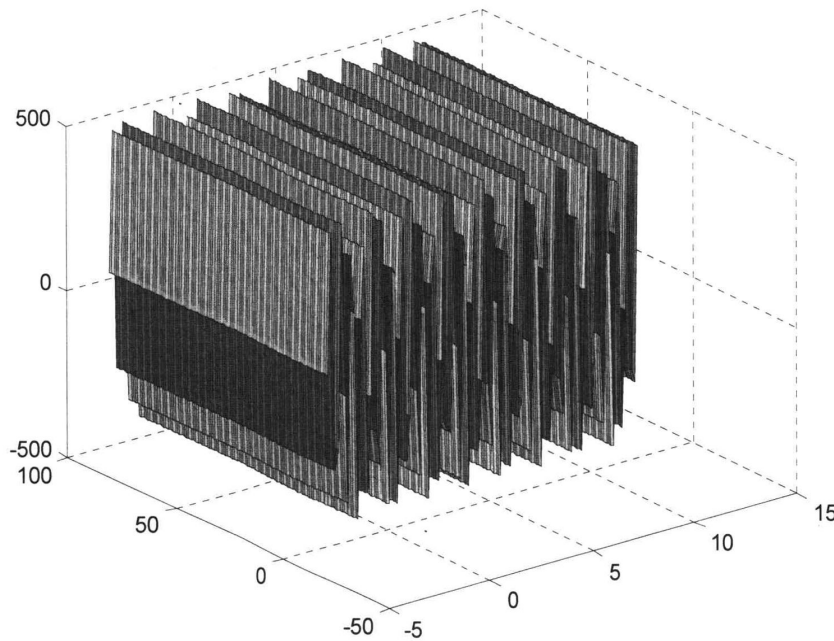
$$P_i = \frac{f_i}{\sum_{k=1}^N f_k}$$

当我们确定复制的概率是多少以后，用计算机模拟的方法产生一些数，这些数字使用随机的方法产生的并且这些数字都是在[0, 1]区间产生的。目标函数值大的个体，被选择的概率就会大一些，可能好几次选择都会选择他，目标函数值小的个体被选取的概率就要小很多，可能被选到一次，也有可能一次都不被选到。就这样目标函数值小的个体逐渐被淘汰，留下来的都是目标函数值好的个体。这步操作完全符合了达尔文进化的思想，同时也是遗传算法的核心部分。

遗传算法的计算过程比较复杂令人难懂，下面介绍一个案例来具体说明遗传算法的计算过程。

$$\text{求 } \max f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(4\pi x_2)$$

$$s.t \begin{cases} -3.0 \leq x_1 \leq 12.1 \\ 4.1 \leq x_2 \leq 5.8 \end{cases}$$



1. 编码

首先要做的工作就是对我们熟知的十进制的数进行编码，要按二进制把它编成计算机能识别的数字。编成二进制的数时，这个数具体有多长还要取决于我们要求精确到小数点后几位。例如，一个变量 y 的取值范围是(L, U)，要求精确到小

数点后四位,也就意味着任意一个变量,应该被分成很多部分,至少是 $(L,U)*10^4$ 个部分。对于一个十进制的数具体要如何的把它转化成二进制的数,要应用下列公式进行转换。

$$2^{m_j-1} < (U-L)*10^4 \leq 2^{m_j} - 1$$

本题要求保留到小数点后四位,则目标函数的两个自变量 $x_1$ 及 $x_2$ 所构成的染色体数串可以表示如下:

$$\left\{ \begin{array}{l} -3.0 \leq x_1 \leq 12.1 \\ [12.1 - (-3.0)] * 10^4 = 151000 \\ 2^{m_1-1} < 151000 \leq 2^{m_1} - 1 \\ m_1 = 18 \\ 4.1 \leq x_2 \leq 5.8 \\ (5.8 - 4.1) * 10^4 = 17000 \\ 2^{m_2-1} < 17000 \leq 2^{m_2} - 1 \\ m_2 = 15 \\ m = m_1 + m_2 = 18 + 15 = 33 \end{array} \right.$$

因为 $18+15=33$ ,所以在本题中每一个染色体当中就会有 33 个基因,如果把 $m_2$ 的位数放在一段染色体之前那么前 15 位代表的就是 $x_2$ 如表 2.6.1 所列

表 2.6.1 染色体编码

自变量	二进制	十进制	实际值
$x_1$	000001010100101001	5417	-2.687969
$x_2$	101111011111110	24318	5.361653

将二进制已经编好码的实数转化为我们生活中所用的数字为

$$x_1 = -3.0 + 5417 * \frac{12.1 - (-3.0)}{2^{18} - 1} = -2.687969$$

$$x_2 = 4.1 + 24318 * \frac{5.8 - 4.1}{2^{15} - 1} = 5.361653$$

假设一开始群体中有 10 个生物,每个生物的染色体可以生成如下所示:

$$U_1 = [000001010100101001 \ 101111011111110]$$

$$U_2 = [001110101110011000 \ 000010101001000]$$

$$U_3 = [111000111000001000 \ 010101001000110]$$

$$U_4 = [100110110100101101 \ 000000010111001]$$

$$U_5 = [000010111101100010 \ 001110001101000]$$

$$U_6 = [111110101011011000 \ 000010110011001]$$

$$U_7 = [11010001011110001 \ 001100111011101]$$

$$U_8 = [00101101010000110 \ 010110011001100]$$

$$U_9 = [111110001011101100 \ 011101000111101]$$

$$U_{10} = [111101001110101010 \ 0000110110100]$$

相对应的实际数值是  $[x_1, x_2]$  为

$$U_1 = [-2.687969, 5.361653], U_2 = [0.474101, 4.170144]$$

$$U_3 = [10.419457, 4.661461], U_4 = [6.159951, 4.109598]$$

$$U_5 = [10.419457, 4.661461], U_6 = [11.788084, 4.174346]$$

$$U_7 = [9.342067, 5.121702], U_8 = [-0.330256, 4.694977]$$

$$U_9 = [11.671267, 4.873501], U_{10} = [11.446273, 4.171908]$$

## 2 评价个体适应度

在完成一系列的遗传操作之后，要对这些操作结果进行评价，应适应度来表示，具体有三个步骤。

① 将染色体转成我们生活中熟知的十进制的数，也就是说要把二进制的数转换成十进制的数。

$$x^k = (x_1^k, x_2^k), \quad k = 1, 2, 3, \dots$$

② 对目标函数求函数值也叫评价  $f(x^k)$ 。

③ 将目标函数值变成相应的适应度

$$eval(U_k) = f(x^k), \quad k = 1, 2, 3, \dots$$

在传统的遗传算法中，评价函数其实就是大自然中的环境，不断的对最优解进行选择，淘汰：

$$eval(U_1) = f(-2.687969, 5.361653) = 19.805119$$

$$eval(U_2) = f(0.474101, 4.170144) = 17.370896$$

$$eval(U_3) = f(10.419457, 4.661461) = 9.590546$$

$$eval(U_4) = f(6.159951, 4.109598) = 29.406122$$

$$eval(U_5) = f(-2.301286, 4.477282) = 15.686091$$

$$eval(U_6) = f(11.788084, 4.174346) = 11.900541$$

$$eval(U_7) = f(9.342067, 5.121702) = 17.958717$$

$$eval(U_8) = f(-0.330256, 4.694977) = 19.763190$$

$$eval(U_9) = f(11.671267, 4.873501) = 26.401669$$

$$eval(U_{10}) = f(11.446273, 4.171908) = 10.252480$$

要根据染色体的适应函数，来求出染色体的适应值，再依据适应值进行种群的复制，具体做法如下：

- ① 算出染色体  $U_k$  的函数值，也就是适应度的值

$$eval(U_k) = f(x^k), \quad k = 1, 2, 3, \dots$$

- ② 把种群中所有染色体的函数值加在一起求和，也就是求出适应度的总和

$$F = \sum_{k=1}^{pop-size} eval(U_k)$$

- ③ 算出每一个染色体有多大的概率被复制

$$P_k = \frac{eval(U_k)}{F}$$

- ④ 算出每个染色体被复制过的次数总和

$$Q_k = \sum_{j=1}^k P_k$$

### 3. 新种群的复制

根据俄罗斯轮盘的方法，来对染色体进行选择，拨动轮盘 10 次，每一次选

到的就是一个染色体，假设拨动十次产生的数字如下：

0.301434,0.322062,0.766503,0.881893,0.350871  
0.588392,0.177618,0.343242,0.032685,0.197577

根据以上的做法，可以优先算出每个染色体的函数值，也就是适应度和概率。  
如表 2. 6. 2 所列。然后用计算机会产生一些随机数字，具体产生的随机数字如下  
所示：

表 2. 6. 2 种群每条染色体的适应度、被复制概率和被复制的积累率

染色体	适应度值	$P_k$	$Q_k$
$U_1$	19.805119	0.111180	0.111180
$U_2$	17.370896	0.097515	0.208695
$U_3$	9.590546	0.053839	0.262534
$U_4$	29.406122	0.165077	0.427611
$U_5$	15.686091	0.088057	0.515668
$U_6$	11.900541	0.0668606	0.582475
$U_7$	17.958717	0.100815	0.683290
$U_8$	19.763190	0.110945	0.794234
$U_9$	26.401669	0.148211	0.942446
$U_{10}$	10.252480	0.057554	1.000000

第一个，产生的随机数是：0.301431，比  $Q_3$  大比  $Q_4$  小，所以  $U_4$  被选中；

第二个，产生的随机数是：0.322062，比  $Q_3$  大比  $Q_4$  小，所以  $U_4$  又一次被选中；

第三个，产生的随机数是：0.766503，比  $Q_7$  大比  $Q_8$  小，所以  $U_8$  被选中；

.....

第十个，产生的随机数是：0.197577，比  $Q_1$  大比  $Q_7$  小，所以  $U_2$  被选中。

根据轮盘赌选择的方法，产生的新的染色体如下图所示：

$$U_1 = [100110110100101101 \ 000000010111001](U_4)$$

$$U_2 = [100110110100101101 \ 000000010111001](U_4)$$

$$U_3 = [001011010100001100 \ 010110011001100](U_8)$$

$$U_4 = [111110001011101100 \ 011101000111101](U_9)$$

$$U_5 = [100110110100101101 \ 000000010111001](U_4)$$

$$U_6 = [110100010111110001 \ 001100111011101](U_7)$$

$$U_7 = [001110101110011000 \ 000000010111001](U_2)$$

$$U_8 = [100110110100101101 \ 000010101001000](U_4)$$

$$U_9 = [000001010100101001 \ 101111011111110](U_1)$$

$$U_{10} = [001110101110011000 \ 000010101001000](U_2)$$

这种俄罗斯轮盘选择的中心思想就是：如果染色体的函数值大，也就是适应度大，那意味着 $[Q_k, Q_{k+1}]$ 之间的跨度比较大，计算机随意产生的随机数就会有更大的概率落在较长的 $[Q_k, Q_{k+1}]$ 的区间里，这样较大 $P_k$ 值的染色体会会有更多的机会被选择复制到下一代

#### 4. 新种群交配

##### (1) 选择交配的染色体的数量

要把这个种群中所有的染色体又要乘以计算出的需要交配的概率。这里假设交配概率 $P_c$ 为0.25，设染色体总量数目为10条，所以 $10 \times 0.25 = 2.5$ ，需要交配的条数就是 $[2.5]$ 条。符号 $[]$ 表示取整数的意思，这里取的整数就是2，就是要重染色体中选出2个染色体再进行配对。

##### (2) 确定交配的染色体数量

用电脑模拟产生 $[0, 1]$ 区间的10个随意的数也叫随机数如下：

0.625721   0.266823   0.288644   0.295114   0.163274

0.567461   0.085940   0.392865   0.770714   0.548656

假定其分别对应的是 $U_1 \sim U_{10}$ 这10条染色体，则其中小于交配概率0.25的



$U_5$  和  $U_7$  参加染色体交配。这样操作的原因就是：交配概率越低，在交配概率下面您的随机数就会不断的减少，所以，需要交配的染色体概率越小则可用来交配的染色体数量就会越少。

（3）交配要在一个固定的场所发生，这个场所就叫交配池

染色体  $U_5$  和  $U_7$  被选中作为第一对开始交配的染色体，具体在染色体的什么位置交配要看计算机产生的随机数是怎样的。交配的方式有两种，一种就是在一个基因上进行交配另一种就是在多个基因上进行交配。用电脑随机生成一个在  $0 \sim 32$  之间的整数，因为整个染色体用二进制编码后的位数是 33 位。假设所产生的数是 2，那么就在染色体的第三位基因上开始进行变换。即

$$\begin{aligned} U_5 &= [10011011010010110100000010111001] \\ U_7 &= [001110101110011000000010101001000] \\ &\quad \downarrow \\ U_5^* &= [101110101110011000000010101001000] \\ U_7^* &= [00011011010010110100000010111001] \end{aligned}$$

5. 基因突变

在遗传算法中突变是一个比较重要的环节，它可以引入新的基因，设突变率为  $P_m$  0.01，也就是种群内的多有基因都是以 0.01 的概率进行突变。在本例中共有  $33 \times 10 = 330$  个基因，即也就是下一代中会有 3.3 个基因进行突变，每一个基因的变化概率是相同的。因此，要将这 330 个介于  $0 \sim 1$  之间的数字编上号码，以后经小于 0.01 的数选出来，把选出来的数的基因值加以翻转，产生新的个体。假设 330 次中产生  $0 \sim 1$  之间的随机数，其值小于 0.01，如表 2.6.3

表 2.6.3 基因突变位置

基因位置	染色体位置	基因位数	随机数
105/33=4...6	4	6	0.009857
164/33=5...32	5	32	0.003113
199/33=7...1	7	1	0.000946
329/33=10...32	10	32	0.001282

在表中的第一列中显示出了有哪些染色体被选择了突变，又在这些染色体的哪一些具体的位置上进行突变。例如第二行  $164/33=5 \dots 32$ ，就是说要在第 5 条

染色体上的第 32 位基因上发生突变, 因为第 5 条染色体对应的第 32 位基因编号是  $33 * (5-1) + 32 = 164$ 。

在突变完成以后, 最终生成的新的染色体组是:

$$\begin{aligned}
 U_1 &= [100110110100101101 & 000000010111001] \\
 U_2 &= [100110110100101101 & 000000010111001] \\
 U_3 &= [001011010100001100 & 010110011001100] \\
 U_4 &= [111111001011101100 & 011101000111101] \\
 U_5 &= [101110101110011000 & 000010101001010] \\
 U_6 &= [110100010111110001 & 001100111011101] \\
 U_7 &= [100110110100101101 & 000000010111001] \\
 U_8 &= [100110110100101101 & 000000010111001] \\
 U_9 &= [000001010100101001 & 101111011111110] \\
 U_{10} &= [001110101110011000 & 000010101001010]
 \end{aligned}$$

产生的新一代个体的数值  $[x_1, x_2]$  和相应的函数值也就是适应度如下:

$$\begin{aligned}
 eval(U_1) &= f(6.159951, 4.109598) = 29.406122 \\
 eval(U_2) &= f(6.159951, 4.109598) = 29.406122 \\
 eval(U_3) &= f(-0.330256, 4.694977) = 19.763190 \\
 eval(U_4) &= f(11.907206, 4.873501) = 5.702781 \\
 eval(U_5) &= f(8.024130, 4.170248) = 19.910251 \\
 eval(U_6) &= f(9.342067, 5.117020) = 17.958717 \\
 eval(U_7) &= f(6.159951, 4.109598) = 29.406122 \\
 eval(U_8) &= f(6.159951, 4.109598) = 29.406122 \\
 eval(U_9) &= f(-2.687969, 5.361653) = 19.805119 \\
 eval(U_{10}) &= f(0.474101, 4.170248) = 17.370896
 \end{aligned}$$

到这一步为止, 已经完成了该算法的第一代循环流程。重复以上操作迭代, 在第 451 代中我们得到了目标函数最大的那个染色体, 具体如下

$$U_{best} = [111110000000111000 \quad 1111010010101110]$$

相应实际值  $[x_1, x_2] = [11.631407, 5.724824]$ , 适应度值  $eval(U) = f(11.631407, 5.724824) = 38.818208$ 。

## 6. GA 操作过程

Step1: 最开始要确立遗传算法中的参数, 确定好我们所求得的目标函数和描述问题的形式以及我们建立的数学模型。

Step2:用计算机的方法随机产生一个种群作为初始中;

Step3:我们要测算出适应的值,测算结果作为下一次操作的根据,也为下一次操作进行了指南;

Step4:依据上一步操作的结论,使用一些其他的方式继续选择后续的操作个体;

Step5:将上一步操作作后的个体,设定怎样去实行交叉和概率的确定;

Step6:将上一步操作后的个体,设定一种变异和一种方法;

Step7:看看结果是否已近达到了终止条件的要求,如果达到了就不需要在进行循环了,跳出操作,然后继续下一步的运作,如果没有达到,那么就要进行第三步继续执行;

Step8:当到达条件终止时,循环结束了,最后得到问题的最优的解。遗传算法的具体流程图如下所示:

## 2.6.8 遗传算法的特点

遗传算法是学术界新发明的一种算法,这种算法的应用性十分的强悍,很多领域在遗传算法被提出来的时候,就主动的把遗传算法引入自己的领域中进行优化操作。遗传算法和以往的那些不智能的方法相比是及其智能了,相比于他们有了很多的优势,具体表现在以下几点中:

1. 这种算法一开始寻找解的时候就是在整个解的范围内寻找的,并不是在解的范围内一个一个的去寻找的。这就是为什么遗传算法要比传统的那些优化算法效果好的真正的原因。以前那些非智能的优化的算法就是从一个值一个值来开始求解的,这样做的不好的地方就是容易达到局部的最优解就停止了算法的进程,一直徘徊在局部极值的附近。遗传算法从整个解的空间一起开始搜索,好处就是搜索的范围变大,可以容易的达到全局的最优解。

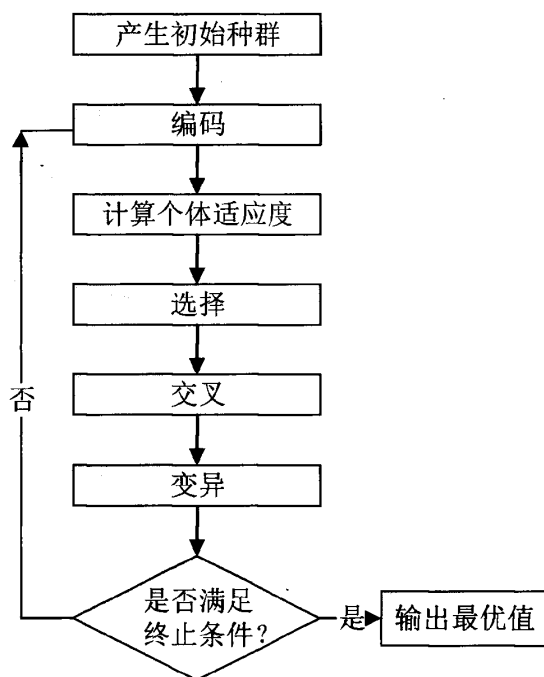
2. 遗传算法可以在一个时间内一起处理群体内的很多的个体,要对搜索的解的空间进行计算与评估,尽量的减少遗传算法徘徊在局部极值的附近,与此同时这个算法的本身很容易的就实现了并行化。

3. 遗传算法在空间中寻找的时候,一般情况下不会用其他的知识和信息对他进强化,都是依靠自己来进行寻找解的。而仅仅就用目标函数值来对所求出的解

进行考察，考察他们是否符合要求。目标函数值不受其他的因素影响，而且他的变量的取值范围也可以认为的规定，这些优势都会推动遗传算法不断的向前进行发展。

4. 遗传算法解出来的解并不是一定的，因为他每一次的搜索前进方向都是用概率的方法完成的。

5. 遗传算法是模拟生物的一种算法，所以它也具备了生物所应有的能力，比如高效的学习性，强大的适应性等等。遗传算法在寻找解的过程中不断的获取相应有效的知识，并且利用这个知识来指导算法搜索前进的方向，函数值大的生存下来的机会就会更大一些，就获得了更多的基因型，获得的这些新的基因型可以更好的适应环境。最后这些基因型都被保存了下来。



## 2.7 本章小结

本章介绍了最基本的遗传算法传统遗传算法，现在学术界对遗传算法的改进都是基于传统遗传算法，在此基础上对其出现的缺点进行改进。本章详细的介绍了遗传算法的起源，发展过程、研究背景以及其算法步骤，为广大读者全方位的

理解遗传算法提供了方便。

## 第3章 多种群遗传算法

### 3.1 遗传算法早熟问题

遗传算法是一种模拟生物界自然进化的一种优化能力很强的智能算法,这种算法不易受到其他外界的因素影响。在优化的时候不依赖一些其他的方面,具有很强的独立性和寻找全局最优解的能力,因此被很多行业引入进行优化目标函数,例如,军事、医疗和经济的领域。遗传算法从开始兴起的时候,就吸引到了很多学者的目光,很多学者也愿意去研究这个算法,随着研究者的不断增加,算法不断的发展,遗传算法的不足之处也渐渐的显露出来,例如,遗传算法为大多数人所熟知的一个问题就是过早的收敛的问题。

遗传算法中没找到最优解时,算法就会停滞不前,这是遗传算法的一个很大的不足,主要表现在在一个种群中每一个个体的特征都是相同的,他们的状态也是相同的,具有很高的相似性,这样算法就会停滞,徘徊在局部极值的附近。当算法停滞的时候,就不能继续的向前搜索,也就不能给出大家的期望的最优解。

(1) 选择操作很简单,选择个体时要依据概率进行选择,这个概率是由目标函数值所决定的,值大的个体被选取的几率就大,这样在几次重复的过程中这个个体的性能会影响其他个体,整个群体中的个体就会失去竞争性,最后算法停滞。

(2) 遗传算法中的一些操作是受他们相应的概率来进行控制的,比如突变就会收到突变率的控制,这些操作的概率对于遗传算法的整个寻优过程都是影响很大的,这些概率不能设置的太大也不能设置的太小,否则会影响搜索的结果,导致最后的求解不稳定,也不够准确。

(3) 初始种群的大小也会影响整个算法的优越性,也会导致搜索的结果不够理想。当一个种群中的个体数较小的时候,每个个体之间的争夺性会变小,在搜索的时候,整个搜索的过程会慢慢下降,整个过程比较单一化;但是当一个种群的个体非常的大时候也会出现一些问题,那就是计算过程变慢,因为当种群变大的时候数目增多必然会导致计算效率的下降,搜索速度变慢。

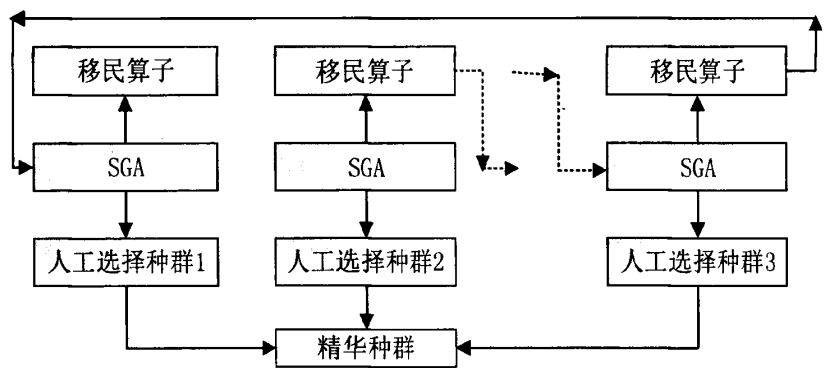
(4) 遗传算法的终止条件有很多,但是最常用判断终止的条件就是要人为

的去规定遗传的代数，遗传的代数不能规定的过少，应为如果规定的过少算法的进化不会很充分，也可能导致算法的过早收敛，代数规定的过大也是不可以的，如果人为的规定这个代数过大那么势必不造成进化过程中浪费时间，费时费力。在提出遗传算法没多久之后，有的科学家就发现了遗传算法会出现过早收敛的这种情况，为了解决这个问题，学术界做了很多关于这方面的探索。有的学者在遗传参数上做文章，并已经取得了一定的成果。但是遗传算法的过早收敛问题与许多因素有关，遗传参数应该怎样设定与控制才能更好的解决过早收敛并且在实际过程中怎样应用到各个领域，这些问题都值得学术界进行思考。

3.2 多种群遗传算法概述

针对遗传算法的多种问题，科学家们辛辛苦苦。呕心沥血的发明了一种新型的遗传算法，多种群遗传算法，多种群遗传是以标准遗传算法为基础，并对标准遗传算法进行了一些改进而形成的一些算法。下面引入几个概念：

- (1) 标准遗传算法采用的是一个初始种群来进行进化的，但是一个种群未免太单一，这时候引入多个初始种群，让这些多个种群同时的开始进行搜索。
- (2) 每一个种群都用一种叫移民算子的算子进行信息间的交流，实现多各种群一起进化发展；最后这种方法得出的最优结果是多个种群共同进化的结果。
- (3) 每一个操作的算子都要人为的设定，然后保存进化中最优的个体，作为判断是否收敛的凭证。



每一个种群算子都会影响算法的局部结果还有全局结果，决定着最优解的效果。变异率，交叉率等算子不能取得太大也不能取得太小，不然都会影响最优解。在标准的遗传算法中，变异算子，还有交叉概率都是整个算法过程中的主要算子，

这些算子共同决定寻有结果的好坏。变异算子是一种能产生新个体的算子，基因经过变异产出新的个体，他决定整个算法的局部搜索能力。但一些科学家给出了建议选择比较大的  $p_c(0.7 \sim 0.9)$  和比较小的  $p_m(0.001 \sim 0.05)$ 。 $p_c$  和  $p_m$  的选取方式有很多种，不同的选取方式会产生不同的效果，最后得到的最优解之间的差距也比较大。遗传算法的这种缺点，新发明的多种群遗传算法起到很好的效果来弥补这种不足，就是多个种群相互配合的来完成整个进化过程。

每一个种群都是相互独立的一个群体，不受其他群体操作算子的影响，但是他们也不是完全群独立的，而是通过一个移民算子来进行每一个种群之间的信息交换。移民算子交流信息是这样操作的，把我们要求的那个目标函数中的种群中最不好的个体，来代替原来那个种群中最好的个体。多种群遗传算法这种新型的算法很好的弥补了标准遗传法的缺点，移民算子在这个过程中起到了至关重要的作用。

精英保存种群和初始种群是不同的，他们有很多方面不尽相同，在每产生新一代的个体中，把每一代的最好的那个个体保存下来。精英种群不进行遗传的各种操作，直接被保存下来，目的就是为了防止这些性能好的个体遭到破坏。

### 3.3 多种群结构设计

多种群遗传算法有很多种用途，可以用到静态模型中，也可以把它应用到动态的模型中，在应用到动态模型中的时候就要求多种群算法必须要有很好的适应性，这样才能得到满意的效果。为了及时的了解求解状态必须要时时关注环境的变化，必须要适时地观测所求到的解，直到求解出最优解为止。这就是把这种算法应用到动态中和应用到静态中最根本的区别。以前那些近乎算法的进化过程是在搜索最优解的过程中使算法慢慢的收敛，直到获取一个满意的解为止。这种设计种群的方法使种群过于单一不丰富，搜索效果不好，而多种群的群体设计恰恰弥补了传统遗传算法在这方面的不足，为在整个解空间中搜索解提供了必要的可能。因此，以前的那些进化的算法在搜索最优解的后期失去了继续前进的动力，这就是为什么有些算法在局部最优解附近徘徊的真正原因。这些年，学者们提出了一些解决遗传问题的具体可行的办法，并取得了很理想的效果，如下所示。

(1) 要改变遗传算法中的一些进化算子，使遗传算法能够很快的适应当前



变化好的环境。

(2) 要提防种群过早的收敛，尽量是种群丰富，这是因为当种群多的时候这个算法能更好的融入变化的环境中，得到更好的解。

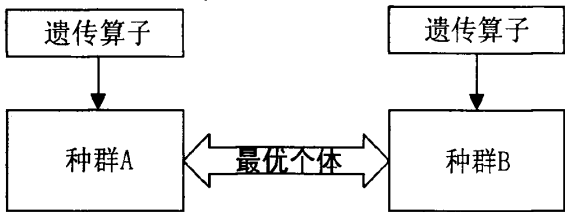
(3) 遗传算法中要引入一种记忆法，假设种群中的每一个个体都能够记起他们以前的进化过程，在环境有规律的进行变化的时候这种方法十分的适用。

(4) 采用种群分割的办法，将一个大的种群分成大大小小的其他种群，一些种群继续进化寻找最优解，另一部分在当前最优值附近徘徊搜索。

与以前的那些遗传算法的算子相比较，动态遗传算法中的算子更能保证种群的丰富，要使算法能够在变化的环境中取得良好的搜索效果，就要使这个种群的初始种群变的丰富。对现有其他学者所研究得出结果进行分析，提出一种新型的遗传算法，这个算法直接使用了两个群体，群体A以及群体B。如下所示：

(1) 群体A在解空间上进行全局搜索最优解，目的就是为了快点找到最优解。

(2) 群体B在局部上进行解的搜索，这样可以保持群体丰富。



MPGA 的结构

下面对 MPGA 算法的主要算子进行描述。

(1) 编码方式

群体A和群体B均运用我们日常生活中熟悉的实数来进行编码，实数这种编码稍微有点麻烦即复杂，但是这种编码方式可以提高计算效果。

(2) 选择算子

选择操作的第一步就是运算目标函数值也叫适应度，根据运算的目标函数值来确定每个个体能被选择到的几率。

一、轮盘赌选择法

俄罗斯轮盘赌选择法又叫一定比例选择法。每个个体被选中的概率和这个个体的函数值大小的有关。设群体大小为  $n$ ，个体  $i$  的适应度为  $F_i$ ，则个体  $i$  被选

中遗传到下一代群体的概率为:

$$P_i = \frac{F_i}{\sum_{i=1}^n F_i}$$

种群中每一个个体的适应性分数, 这个适应性分数用一个圆形的图来表示, 圆形的图中的每一小块都表示一个染色体。适应性分数大的在圆形的图中表示他的那块就大, 适应性小的在圆形的图中表示他的那块就小。当选取染色体的时候, 要使这个圆形的图旋转起来, 当图停止转动的时候, 看看圆形的图上的指针指在哪个小块上, 指在哪个小块上哪个小块代表的染色体就被选中。

群体 A 和群体 B 用俄罗斯轮盘赌的方式进行选择算子, 这种方法, 简单、便捷。具体步骤如下。

(a) 计算每个个体的函数值也叫适应度  $f_i$ ;

(b) 计算每个个体的权重:  $w_i = 1 - f_i / \text{sum}(f_i)$ ;

(c) 计算每个个体被选中的比例:  $p_i = w_i / \text{sum}(w_i)$ ;

(d) 产生一个[0-1]之间的随机数, 将其作为依据来确定那个染色体可以被选定。

### (3) 交叉算子

用我们日常生活中的十进制来进行编码, 采用以下交叉算子。设  $\alpha$  为一随机数,  $p_1$  和  $p_2$  为依据选择方法所选择的父代个体, 则子代个体为

$$c_1 = \alpha * p_1 + (1 - \alpha) * p_2$$

$$c_2 = (1 - \alpha) * p_1 + \alpha * p_2$$

### (4) 个体的迁移机制

群体 A 和群体 B 在个体迁移的时候存在两种迁移方式:

1. 群体 B 中最好的个体迁移到群体 A 中。由于群体 A 的目标是在空间上以最快的速度寻找到最优解, 所以直接把群体 B 中的那个最好的解替换掉群体 A 中最不好的那个解。

2. 群体 A 中最好的解迁移到群体 B 中。因为群体 B 中的的是在局部上进行寻找最好的解, 同时还要保证种群丰富, 因此不应该随意的去替换群体 B 中最不好的

那个个体。为了让群体 B 更加丰富,要在群体 B 中找一个个体与群体 A 中的个体相似,然后在进行替换,相似性定义为,与群体 A 距离最近的那个个体,计算公式如下。

$$d_i = \sum_{m=1}^D \sqrt{(p_{(A)}^m - p_{(B)i}^m)^n}$$

首先要找出种群 A 中准备迁移的个体,然后计算这个个体与种群 B 中的每一个个体的距离计算出来,然后找一个最相近的进行替换。

### 3.4 变异算子设计

一般情况下认为突变的概率在算法的进化过程中是十分重要的,一是因为它可以是群体变的更加丰富,不易让算法在局部极值附近徘徊,二是因为算子也可以加快计算速度。把算子进行变异,在群体 A 中用一种变异算子,在群体 B 中用另外一种算子,这样为了达到不一样的结果。

运用随机的变异方法在群体 A 中,随机数来自下面这个函数:

$$f(x) = \frac{x-a}{b-a}$$

为了实现染色体的变异,用计算机产生新的随机数来代替变异个体的染色体上的基因。突变的结果是很难把握的及其的不确定,突变的结果可能比现在的好一些,也可能比现在的不好一些。但是一旦比现在的结果好就可以推进整个算法的过程,如,加快算法的计算效率。

突变可以加快群体的计算效率,但是算法还是会在局部的极值附近徘徊。由于群体 A 的计算效率高,因此,群体 A 采用随机的突变方案。

在群体 B 中引用高斯方案,高斯分布的密度函数可描述为:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty$$

其中,  $\sigma$  为高斯分布方差,  $\mu$  为高斯分布的均值或期望值。设  $p$  是概率分布,

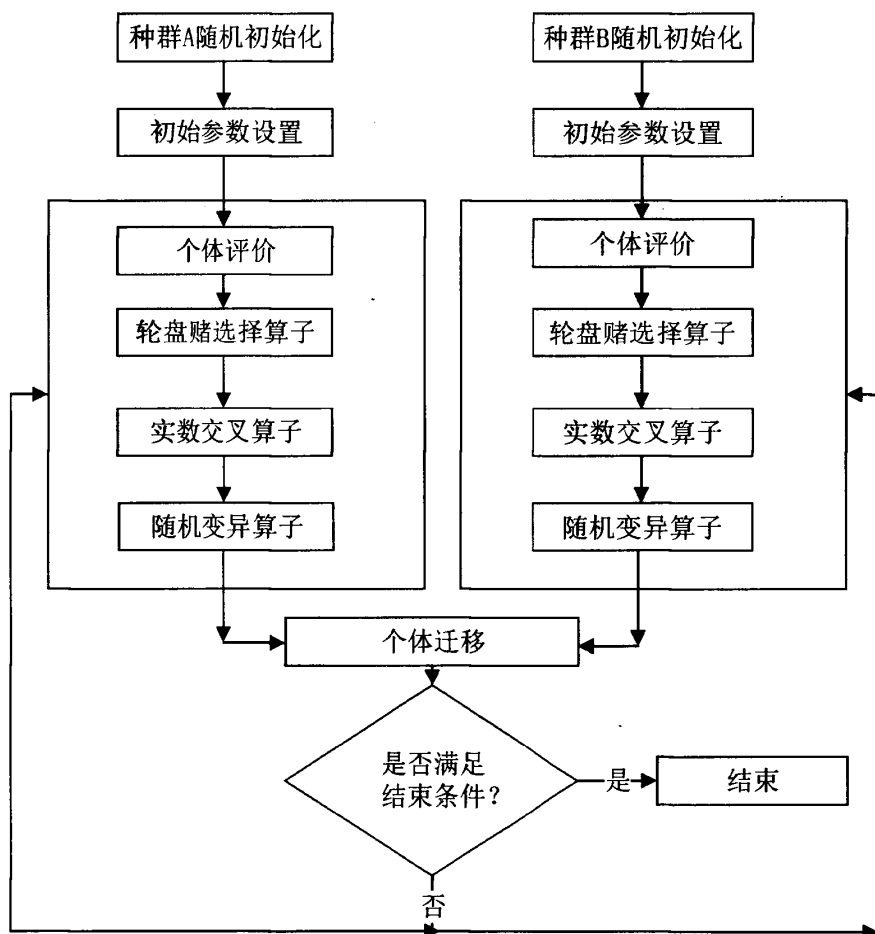
$\xi(w)$  为随机变量,则高斯分布具有如下性质:

$$p(-\delta \leq \xi(w) \leq \delta) = \frac{1}{\sqrt{2\pi}\delta} \int_{-\delta}^{\delta} e^{-\frac{x^2}{2\delta^2}} dx \approx 0.688$$

$$p(-2\delta \leq \xi(w) \leq 2\delta) = \frac{1}{\sqrt{2\pi}\delta} \int_{-2\delta}^{2\delta} e^{-\frac{x^2}{2\delta^2}} dx \approx 0.955$$

$$p(-3\delta \leq \xi(w) \leq 3\delta) = \frac{1}{\sqrt{2\pi}\delta} \int_{-3\delta}^{3\delta} e^{-\frac{x^2}{2\delta^2}} dx \approx 0.997$$

高斯突变的效果非常的好,可以让染色体在个体附近发生很小的突变,在一定的程度上是可以在局部寻找最优解的,还可以把算法的收敛效率提高,同时也可以提高准确率,但是在那个范围变化,这就要由高斯分布的方差决定了。虽然高斯每一次变化的幅度有点小,但是也可以产生很大幅度的变化。算法流程图



### 3.5 本章小结

本章介绍了遗传算法的一种改进方法,多种群遗传算法,该方法对于克服传

统遗传算法过早收敛的问题有着较好的实验效果, 作者将把多种群遗传算法应用在优化钢铁烧结配矿中, 为钢铁厂提供一种全新的优化方法。

## 第4章 新型遗传算法

### 4.1 改进遗传算法

传统的遗传算法自从被提出来以后,就受到了学术界的普遍关注,近些年的发展也十分的迅速,在学术界已经得到了学者们的广泛认可。他在处理我们生活中的写实际的问题的时候是一个非常有效的工具,但是算法仍然有一些缺点,因此一些科学家对遗传算法进行了大量的改进以便它可以更好的解决生产生活中约到的问题。

遗传算法在寻找全局最好的解的过程是比较有效的,可以很快的找到全局最好的解,他依据自然界生物进化的基本思想进行寻找最好的解。在每产生一代新的个体的时候,他都会把这一代中最好的个体保存下来以便以后进行挑选。然后按照一定的选取规则从这些保存的解中挑出最优的解。利用一些遗传过程中的算子进行一些操作在产生新的解,然后再一次重复进行上述过程,直到算法满足终止条件为止。

要引入一个函数,这个函数的名称叫做适应度函数,这个函数可以用来测试一个个体是好还是坏,实际上适应度函数就是目标函数,目标函数值越大,解出来的解越好。这个目标函数推动着遗传算法不断的向前发展,也是遗传算法进化的一种标准并且这种标准是唯一的。这个函数是要根据实际问题来进行构建的,遗传算法是通过选择计算来达到进化的。目标函数值大的个体在下一代运算中被选择的几率就会变大,这种操作就上从上一代群体中选择一些符合某种规则的个体。遗传算法用俄罗斯轮盘赌的方法来确定操作中的算子。首先要确定一个规则就是一个种群经过有限次的进化可以寻找到全局的最优个体,其次是要保证选出的个体是最优的,引入保优策略,遗传算法的一些操作算子对遗传算法有很大的影响。

遗传算法是一种智能算法,本身具有很大的优势,在全局搜索中具有很强的能力,所以常常被一些复杂的领域引入作为一种优化的工具。但是这个算法也有不足之处。科学家们做了很多关于遗传算法改进的工作,并取得了一些实际客观的效果。很多学者把遗传算法和其他的算法混合使用来改进算法的不足,比如,

遗传算法和 B.P 神经网络相结合, 形成混合算法。这些操作极大的改进了遗传算法, 增加了遗传算法的功效使其成为一种更好的优化工具。

## 4.2 自适应遗传算法

外国的一个教授提出了一种新型的遗传算法, 并把这种新型的遗传算法命名为自适应遗传算法, 从命名上我们就可以看出这种算法有很强的适应性, 这种算法可以根据具体的问题的环境来自动调整操作过程中算子的大小, 以便于是整个算法更加好用, 更加智能化, 也可以避免算法过早的收敛。设  $f_{\max}$  为某一代种群中最好的染色体的目标函数值,  $f_{\text{avg}}$  为这一代群体中每一个个体函数值的平均, 又设  $\Delta = f_{\max} - f_{\text{avg}}$  则认为  $\Delta$  越小, 群体中的个体之间相差的就越小, 群体乡收敛的方向发展。  $\Delta$  越大, 则群体的这些性能与  $\Delta$  小的时候都相反, 因此为预防群体早熟, 当  $\Delta$  比较小时, 应加大  $P_c$  和  $P_m$ ; 当  $\Delta$  比较大时, 应缩小  $P_c$  和  $P_m$ 。

$$\begin{cases} P_c = K_1 / (f_{\max} - f_{\text{avg}}) \\ P_m = K_2 / (f_{\max} - f_{\text{avg}}) \end{cases}$$

上面的式子可以给我们一些重要的信息就是这种算法可以根据实际问题的环境来自动的调整自己寻找最优值的能力。但是上面的公式可以表达出另一个观点, 当个体可以寻找到全局最好的解的时候, 最好的解被破坏的可能性也会变大, 为了弥补这一缺点, 上面的公式改写如下:

$$\begin{cases} P_c = K_1 / (f_{\max} - F)(f_{\max} - f_{\text{avg}}), F \geq f_{\text{avg}} \\ P_m = K_3 & F < f_{\text{avg}} \end{cases}$$

或者取:

$$\begin{cases} P_c = K_2 / (f_{\max} - f)(f_{\max} - f_{\text{avg}}), f \geq f_{\text{avg}} \\ P_m = K_4 & f < f_{\text{avg}} \end{cases}$$

上面的两个公式中,  $f_{\max}$  是两个个体产生新个体中目标函数值最大的一个,  $f_{\text{avg}}$  是要变异个体的函数值, 自适应这种算法在保证种群丰富性的时候, 也可以避免算法过早的收敛。

### 4.3 混合遗传算法

遗传算法在搜索全局最好的解的时候有非常好的效果。但是，局部的搜索效果并不是很理想，但是有一些算法局部搜索效果很好，这类算法的局部寻找最好的解的能力强，所以把这些算法与遗传算法相结合可以很好地弥补遗传算法的不足。如下图所示：

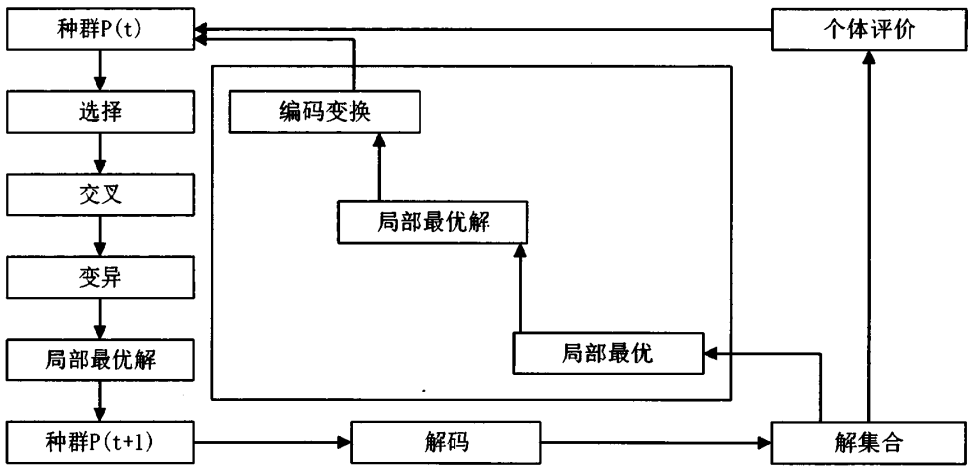


图 4.3.1 混合遗传算法示意图

图 4.3.1 是遗传算法和别的算法混合在一起的图。首先要对将要做优化的问题进行计算机的编码，其中的  $P(t)$  可以看作是上一代的问题经过了计算机的编码， $P(t)$  经过一系列的遗传操作会产生下一代染色体  $P(t+1)$ ，对新产生的个体进行解码得到新的染色体的解。求出每个个体的目标函数值，就是要看看这个个体是否满足题目的要求，是否达到了终止的条件，达到了要求则算法终止。在种群内部选取最优解避免遗传算法陷入局部的最优。

自从遗传算法被提出以后发展到现在，已有许多的学者把遗传算法和别的算法结合在一起进行研究计算，努力为各行业各领域提供更好的优化工具。比如：一位外国的教授就把退火算法和遗传算法进行混合，经过混合之后的遗传算法对于解决复杂的问题具有很好的效果。还有一些混合的例子，比如，把遗传算法和 BP 神经算法相结合，把粒子群算法镶嵌到遗传算法中，还有就是蚁群算法和遗传算法相结合，这些融合都得到了很好的效果。



## 4.4 并行遗传算法

遗传算法无论从哪个角度来看都具有并行性。并行性有三种模型，分别是出的粒度的计算模型，细的粒度的计算模型，还有主的从式等数学模型。这三种模型的根本区别就是他们往复的次数不尽相同。有的是种群数目比较丰富，有的是种群比较大。和其他两种模型相比，粗的并行模型比较好。具体原因如下

(1) 操作简单，只需要对传统的遗传算法，稍微的改进一下就可以得到我们想要的模型。

(2) 算法可以再很多的环境下运用，适应问题的能力是比较强的。

## 4.5 免疫遗传算法

为了解决一些遗传算法在优化现实问题中的一些不足<sup>[41]</sup>。将自然界生物能产生免疫的基本原理，把这种原理再加入到遗传算法中，就可以提高遗传算法的寻找全局最优解的能力。把生物体的免疫原理引入到遗传算法中就可以建立起一种新的算法叫做免疫遗传算法。免疫遗传算法就是将大自然中的个体自己发生免疫的过程的原理引入到遗传算法中，免疫的很多优点都可以被遗传算法所采用，采用之后效果就会非常的好，可以增强遗传算法的搜索能力。这个算法是根据大自然中生物免疫的基本原理而设置的。它是一种新型的遗传算法，在这个算法中可以体现很多生物免疫功能上的优点，这种算法的具体流程如下。

(1) 对具体的问题我们要具体的进行分析，分析问题的特征，要重问题中得到一些我们想要的有用信息，然后构建一个合理表达解的方法。

(2) 要确定免疫遗传算法中的一些操作参数。

(3) 产生初始种群：要对初始的种群产生记忆性，就像细胞中经过一次免疫后就会产生一种特殊的记忆，当同一种病毒再次入侵时会产生更多的抗体来消灭这种病毒。初始种群就是要从这样的抗体中选出；否则就用计算机随机的去产生一个种群。对这些抗体进行计算机的编码，一些任务就要从这些抗体开始进行不断的进化直达到终止的条件为止。

(4) 抗体的产生和遗传算法产生新的个体的方法是一样的，都要经过交叉和突变的运算之后，才会产生新的抗体体。

(5) 求解上述各个抗体的目标函数值也就是适应度；

(6) 算法的进化。把新产生的抗体的目标函数值求出来与以往的抗体函数值作比较，哪一个更好则保留哪一个，另一个抗体则被淘汰掉。

(7) 怎样决定抗体：把群体中相近个体的数目找出来与整个群体中的数目做除法，这就叫做抗体的浓度，如果浓度比较高就要减少个体的数目，浓度低则要增加个体的数目。对产生的新的种群重新进行一系列的遗传操作，不断的重复这种操作直到满足终止的条件为止，对新的一代个体要进行一些重复的遗传操作，最后经过不断的进化的到理想的最优个体。

免疫遗传算法与传统的遗传算法相比具有如下优势：

(1) 具有生物细胞的中抗原的记忆性，可以使算法寻找最优解的速度变快，提高在全局中寻找最优解的能力，并且可以保证算法已最快的速度收敛。

(2) 可以使抗体十分的丰富，这样就可以避免算法过早的收敛，也可以提高找寻最优解的能力。

(3) 自我适应的能力、自我调节的能力，可以提高局部的寻找极值的效果。

## 4.6 精英保留策略新思路

遗传算法一般用计算机来进行实现，对一些保存出来的较好的解进行计算机操作，使其向更好的方向发展，尽最大的努力去搜索得到全局最好的解，遗传算法的初始种群用计算机编码，二进制的编码一般较为简单和便捷。进化从计算机产生初始种群的个体开始，然后一代接着一代的不断进行进化搜索寻找最优解，当达到终止条件是输出最优解。在每一代生成染色体的时候，都要对这一代的染色体进行评价，也就是计算出目标函数值，目标函数值是很重要的，它决定着染色体被复制到下一代的几率，这个值越大被复制到下一代的几率就大，函数值小被复制到下一代的几率就会变小，即被复制到下一代的几率与函数值的大小成正比。从初始种群开始，经过一系列的进化操作产生下一代新的种群。

遗传算法这种先进的智能算法包含着智慧，整个算法完全体现了物竞天择，适者生存的思想。尽管这个算法的功能十分的强大，在很多行业都得到了应用，但是再好的算法也有他的不足之处。

### 4.6.1 遗传算法的缺陷

遗传算法在达到终止条件之前都会进行不断的重复，不断的进化与更新染色体群。但是在整个操作过程中有两个步骤是可以破坏算法最优解的，这两个步骤就是交叉和变异，当遗传算法向着背离我们优化目标进行优化时就会破坏已经产生的最优解。比如：

种群 1：

$$U_1 = [100001110010101110 \ 01110111010101] = 28.562$$

$$U_2 = [101101101110000101 \ 01011101110111] = 26.159$$

对种群 1 进行交叉和变异操作种群 1 变成种群 2

种群 2：

$$U'_1 = [000001110010101110 \ 0101011011101] = 25.786$$

$$U'_2 = [101101101110000101 \ 01011101110111] = 26.159$$

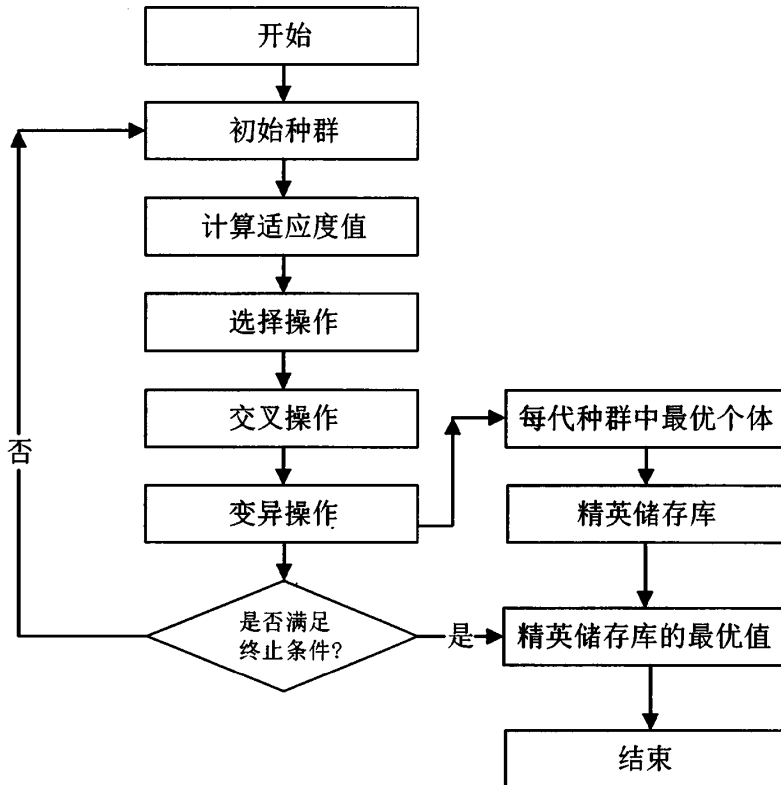
种群 1 中最好的解是  $U_1=28.562$ ，但是经过一系列的遗传操作之后种群 1 中  $U_1$  变化成了种群 2 中的  $U'_1=25.786$ ，最好的解遭到了破坏。在整个算法中最好的解有可能被破坏也有可能不被破坏，所我们要尽量弥补算法的这种缺陷。在遗传操作过程中如果最好的解被选择进行操作，那么最好的解就有可能遭到破坏，因遗传算法进化不一定向好的方向进化，也有可能向不好的方向进化。

遗传算法求解的结果是问题的最大值，遗传操作的过程局是为了让解不断的变大，慢慢的接近最优解，这样的操作就符合自然界进化的理论，也符合科学家提出生物进化的思想。但是遗传操作的结果是不定向的，换句话说就是遗传操作是一种随机的操作不定项的，这就会产生破坏最优解的情况，当操作把使最优解变小的时候就是无用的操作同时与与开发者最开始开发的目标相悖。这样看来遗传算法在设计还是存在不足的<sup>[42]</sup>。

### 4.6.3 新型精英保留策略

受传统的精英保留策略的启发本文作者想到的一种新方式保留遗传算法中的精英个体，为研究保留“精英”的学者提供一种全新的思路。遗传算法每迭代一次会产生一个种群，直到满足条件时输出最优解。在满足输出条件输出最优解的过程中可以产生  $n$  个种群。我们在每次产生种群时把每个种群中的最

优解保存下来在满足输出条件将要输出时，比较这  $n$  个种群中的最优值，然后将最优解输出。这样就可以保证输出的是这  $n$  个种群中的最优解，保护了最优解，避免最优解因为交叉和变异两个操作而遭到破坏。具体流程图如下：



为了使读者更加方便的阅读流程图，论文作者自定义精英储存库的概念。精英储存库是每一代种群中最优个体的集合。这种改进的新型遗传算法可以很好弥补传统遗传算法输出最优解的缺陷，传统的遗传算法输出的最优解为最后一代也就是最后第  $n$  代种群中的最优解，最后一代种群中的最优解并不能保证一定比前  $n+1$  代种群中最优解更优，因此作者设立了精英储存库的概念，设每一代种群中最优个体为  $a_i$ ，设  $A$  为精英储存库的集合即  $A = \{a_1, a_2, a_3, \dots, a_n\}$ 。新型精英策略和传统精英策略最终的目的和结果都是相同的，但是二者在保留精英个体上所采用的方法完全不同，这也为以后研究精英策略的科研人员提供一种保留精英个体的新思路。

## 4.7 本章小结

本章介绍了学术界现有的几种弥补遗传算法不足的方法,最后一种改进的遗传算法新型精英保留策略法是本文作者提出来的一种改进遗传算法的新方法。每个种群的最优解相当于局部最优解,  $n$  个种群的最优解相当于  $n$  个局部最优,传统遗传算法只输出了最后一代种群的最优解极易陷入局部最优。新型精英保留策略算法则避免了传统遗传算法陷入局部最优的困境,保证了遗传算法输出的结果为全局最优解,由于本文作者计算机编程方面的能力不足并没有给出新型遗传算法的实证部分,但是新型精英保留策略法在理论层面是比较完善并且具有可行性。

第 5 章 烧结配料模型的建立

5.1 烧结配料模型的数据及约束条件

表 5.1.1、表 5.1.2、表 5.1.3 的数据均来自于 2017 年 esteel 点钢网配矿大赛数据, 原料物理和化学性质都是炼钢首要考虑的问题, 但是本次配矿大赛的数据要求只考虑化学性质就可以了。

表 5.1.1 含铁原料化学成分及单价

原料名称	Fe/%	SiO <sub>2</sub> /%	CaO/%	Al <sub>2</sub> O <sub>3</sub> /%	MgO/%	P/%	S/%	单价/元 t <sup>-1</sup>
塞拉利昂粉	57.59	2.67	0.01	6.22	0.02	0.080	0.050	558.27
印度粉	57.65	5.62	0.01	3.62	0.02	0.063	0.016	549.01
PB 粉	61.40	3.89	0.02	2.48	0.21	0.095	0.020	762.68
南非粉	63.44	6.27	0.12	1.65	0.21	0.047	0.011	751.42
巴西精粉	64.26	2.73	0.03	1.78	0.06	0.048	0.000	890.41
巴西卡粉	65.78	1.38	0.01	1.23	0.01	0.059	0.000	863.19
乌克兰精粉	65.00	5.80	0.07	0.16	0.09	0.003	0.039	907.10
智利精粉	68.80	1.89	0.26	0.55	0.84	0.010	0.016	965.33

表 5.1.2 烧结用熔剂化学成分表

熔剂	Fe/%	SiO <sub>2</sub> /%	CaO/%	Al <sub>2</sub> O <sub>3</sub> /%	MgO/%	P/%	S/%	单价/元 t <sup>-1</sup>
生石灰	0.00	1.60	85.00	0.00	0.00	0.01	0.12	300.00

表 5.1.3 烧结成分化学性能要求

名称	Fe	R(CaO/SiO <sub>2</sub> )	Al <sub>2</sub> O <sub>3</sub>
指标	≥56.0%	1.85~1.95	≤2.3%

x/%, 表示 x 在相应的矿种中所占的百分比, 例如: 生石灰对应的 SiO<sub>2</sub> 含量百

分比就是 1.6%。

## 5.2 建立烧结配料的数学模型

烧结配料的数学模型建立是依据表 5.1.1、表 5.1.2、表 5.1.3 中的数据进行构建,构建的目的是为了在满足表 5.1.3 中的约束条件下使得表 5.1.2 中各种原料使用成本最低。炼钢厂一般用多种原料放在一起配出能生产出烧结矿的原料。但是按照怎样一种规则来进行配料是非常难得,合理的配料可以生产出质量较好的钢铁,也可以减少钢铁厂的使用成本增加效益。所以按照怎样一种配料方法才能生产出更好的钢铁,是每个钢铁厂首要考虑的问题。对于钢铁厂在建立模型的时候要考虑各个方面的限制条件,在满足限制条件下,使炼钢的成本最低。

$$\text{即: } \min C = \sum_{i=1}^n c_i * x_i \quad (5.2.1)$$

其中  $\min C$  为烧结矿的成本,  $c_i$  为每种原料的成本,  $x_i$  为每种原料的用量

烧结配料要满足烧结对各种不同化学成分的要求,如  $TFe$ 、 $MgO$ 、 $Al_2O_3$ 、 $CaO$ 、 $SiO_2$ 、 $S$ 、 $P$  等

$$\text{即: } d_i \leq \sum_{i=1}^n x_i * l_i^j \leq b_i \quad (5.2.2)$$

式中  $l_i^j$  为物质  $i$  中成分  $j$  的含量;  $b_i$  和  $d_i$  分别为烧结矿各种原料的上限和下限。对烧结碱度  $R(CaO / SiO_2)$  的约束可以表示如下:

$$\text{即: } m < R(CaO / SiO_2) < M \quad (5.2.3)$$

式中  $m$  和  $M$  分别为烧结矿碱度的上限和下限。

综合以上公式列出烧结配料模型如下:

$$\min f(x) = \sum_{i=1}^9 c_i * x_i$$

$$\begin{cases} \sum_{i=1}^9 l_i^{Fe} * x_i \geq 56.00 \\ 1.85 < R(CaO / SiO_2) < 1.95 \\ \sum_{i=1}^9 l_i^{Al_2O_3} * x_i \leq 2.30 \\ \sum_{i=1}^9 \dot{x}_i = 1.00 \\ 0 < x_i \leq 0.30 (i=1, 2, \dots, 9) \end{cases} \quad (5.2.4)$$

其中  $\min f(x)$  为所求最小成本,  $c_i$  为第  $i$  种原料的单价,  $x_i$  为第  $i$  种原料的用量,  $l_i^{Fe}$  为  $i$  第种原料中的含量,  $R(CaO / SiO_2)$  为碱度,  $l_i^{Al_2O_3}$  为第  $i$  种物质中  $Al_2O_3$  的含量。

### 5.3 罚函数

罚函数可以处理一些非常复杂的约束条件, 将他们转化成比较简单的形式以便于求解, 随着烧结配料的限制条件增多, 约束条件也变得比较复杂不利于求解, 而罚函数为简化约束条件提供了可能, 它可以把有约束条件的目标函数变成无约束条件的目标函数方便求解, 在配料建模的时候往往会考虑原料的物理、化学性质, 约束条件变得比较复杂, 因此论文作者引入罚函数的思想化繁为简便于求解。罚函数一般分为内罚函数和外罚函数论文采用内罚函数来对目标函数的约束条件进行优化。

考虑如下问题:

$$\begin{aligned} & \min f(x) \\ & s.t. \begin{cases} g_i(x) \leq 0 & i=1, 2, \dots, r \\ h_i(x) \geq 0 & i=1, 2, \dots, s \\ k_i(x) = 0 & i=1, 2, \dots, t \end{cases} \end{aligned} \quad (5.3.1)$$

取一个数  $M > 0$ , 构造函数如下:

$$P(x, M) = f(x) + M \sum_{i=1}^r \max(g_i(x), 0) - M \sum_{i=1}^s \min(h_i(x), 0) + M \sum_{i=1}^t |k_i(x)| \quad (5.3.2)$$

根据 (5.3.1)、(5.3.2) 式, (5.2.4) 变为如下函数



$$P(x, M) = \sum_{i=1}^9 c_i * x_i + M \max((\sum_{i=1}^9 R_i) - 1.95, 0) + M \max((\sum_{i=1}^9 I_i^{M_i/L_i} * x_i) - 2.30, 0) + M \sum_{i=1}^9 \max(x_i - 0.30, 0) -$$
$$M \min((\sum_{i=1}^9 I_i^{L_i} * x_i) - 56.00, 0) - M \min((\sum_{i=1}^9 R_i) - 1.85, 0) - M \sum_{i=1}^9 \min(x_i, 0) + M \left| (\sum_{i=1}^9 x_i) - 1 \right| \tag{5.3.3}$$

(5.3.3) 式即为整篇论文要优化的目标函数，运用罚函数可以把一些复杂的约束条件变成简单的约束条件，这样做的目的也是为了在计算过程中更加方便。

5.4 结果分析

表 5.4.1 遗传算法与多种群遗传算法结果分析

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	单价/元. $t^{-1}$
遗传算法	0.245	0.195	0.098	0.233	0.07	0.018	0.039	0.02	0.091	691.2743
多种群遗传算法	0.0190	0.1195	0.113	0.210	0.10	0.175	0.029	0.025	0.085	674.2569

在算法中初始种群设定为 50，突变的概率取 0.6，变异的概率取 0.01，, 终止迭代次数 200，单价 691.2743 元，多种群遗传算法种群 A 取 50，突变率 0.5，变异的概率取 0.05，最后输出结果的条件是迭代 200 代为止，种群 B 的规模设定为 50，突变的概率取 0.4，变异率取 0.06，终止迭代次数 200，单价 674.2569 元。MSGA 每吨比 SGA 少用成本 17.0174 元。对于钢铁厂来说减少了成本，现在的大环境下，钢铁库存严重堆积，国家让钢铁厂尽量去库存减少炼钢成本。多种群遗传算法为钢铁厂提供了一种新型的优化方法，能帮助钢铁厂减少成本，节约开支。

5.5 本章小结

本章对前两章介绍的遗传算法和多种群遗传算法应用数学软件 MATLAB 进行了实证，结果表明多种群的优化效果要好于传统遗传算法，把多种群算法应用到钢铁领域是一种应用的创新，为钢铁厂提供一种新的优化思路与方法

## 结束语

对于群集智能优化算法的结合研究,其发展历史尚短,国际上研究成果亦不是很多。遗传算法算是比较经典的智能算法,但是传统的遗传算法本身仍然存在不足。遗传算法在寻找全局最好的解的时候是有很大的优势的,可以快速的去寻找全局的最好的解,在一些复杂的领域中遗传算法也是遗传算法也是一个非常好用的工具。随着社会日益进步不断发展,问题也会变得越来越复杂,遗传算法正好为有效的解决这些问题提供了可行的路径。本文作者在精英保留策略的启发下针对遗传算法局部搜索问题提供了一种新型的精英保留策略,这种新型的精英保留策略和传统精英保留策略最终目的是一样的都是要输出全局最优解,但它们实现的过程却截然不同。

新型精英保留策略可以解决传统遗传算法陷入局部最优的问题,能保证输出的结果为全局的最优解。传统遗传算法输出的结果为在满足输出条件下最后一代进化种群的最优结果,但是在满足输出条件的过程中会产生  $n$  代种群,每一代种群都有一个最优值,种群都要经过交叉和变异两个操作才能进化成更优的下一代,但是在操作过程中有可能破坏种群的最优解最后导致输出的结果并不是全局最优的。本文作者提出了一个精英储存库的概念,精英储存库就是把第一代一直到满足输出条件的最后一代中每一代种群的最优解都保存下来,等到满足输出条件时,然后从精英种群库中选取最优解输出即可,这样就能保证输出的最后解为全局最优解而不是局部最优解。

把新型精英策略应用到钢铁烧结配矿中,从理论上讲可以很好的保证输出最优解为全局最优解,弥补了传统遗传算法在这方面的不足。

## 参考文献

- [1] Brian J, Reardon, Sherri R. Bingert, Inversion of tantalum micromechanical powder consolidation and sintering models using bayesian inference and genetic algorithms(J).Acta Materialia.2000,6(15):15~19.
- [2]C. Ayyappadas, A. Muthuchamy, A. Raja Annamalai et al.An investigation on the effect of sintering mode on various properties of copper-graphene metal matrix composite (J).Advanced Powder Technology.2017,7(15):15~17.
- [3] Deng Hongzhong, Chi Yan, Wan junmin, Tan Yuejin.Systems Science and Systems Engineering--Proceedings of the Fourth International Conference on Systems Science and Systems Engineering(D).2003,11(1): 5~6.
- [4]F Wakai, F Aldinger.Sintering forces in equilibrium and non-equilibrium states during sintering of two particles(J).Science and Technology of Advanced Materials.2014,6(15):521~525.
- [5]Jeremy Croquesel,Didier Bouvard,Jean-Marc Chaix et al.Direct microwave sintering of pure alumina in a single mode cavity(J). Grain size and phase transformation effects, Acta Materialia.2016,6(15):29~31.
- [6]Ke-jiang Li, Jian-liang Zhang, Zheng-jian Liu et al.Optimization model coupling both chemical compositions and high-temperature characteristics of sintering materials for sintering burden(J).International Journal of Minerals Metallurgy and Materials.2014,3(15):11~15.
- [7]Kishalay Mitra, Niloy K, Nath].ANALYSIS OF TWO-LAYER SINTERING PROCESS FOR DIFFERENT BED HEIGHTS BY GENETIC ALGORITHM(J).IFAC Proceedings Volumes.2005,6(15):16~26.
- [8]María-José Sánchez-Rivera, Ana Gozalbo, Valentín Pérez-Herranz et al.Experimental design applied to improving the effect of bismuth oxide as a sintering aid for tin oxide(J).Boletín de la Sociedad Española de Cerámica y Vidrio.2017.12(21):3~4.

- [9] Mróz Jan, Skowronek Ryszard, Francik Przemysław. Investigations on the Influence of Return Sinter Fines on the Iron Ores Sintering Process and on the Properties of Iron Ore Sinter(J). Proceedings of the 5th International Congress on the Science and Technology of Ironmaking. 2009, 10(20): 30~35.
- [10] M. Reihanian, S.R. Asadollahpour, S. Hajarpour et al. Application of neural network and genetic algorithm to powder metallurgy of pure iron(J). Materials and Design. 2011, 6(15): 45~59.
- [11] N Chakraborti, P Mishra, A Banerjee. Optimization of aluminum oxynitride compaction process using a Gray-coded genetic algorithm(J). Materials Letters. 2004, 6(15): 18~36.
- [12] Sung-Min Hur, Kyung-Hyun Choi, Seok-Hee Lee et al. Determination of fabricating orientation and packing in SLS process(J). Journal of Materials Processing Tech. 2001, 6(15): 18~22.
- [13] Wang Xiaokan. Fuzzy Self-Learning Control of Glass Tempering and Annealing Temperature Based on the Optimized Genetic Algorithm(J). 2017, 7(30): 5~6.
- [14] Xin Chen, Xiaoxia Chen, Min Wu et al. Modeling and optimization method featuring multiple operating modes for improving carbon efficiency of iron ore sintering process(J). Control Engineering Practice. 2016, 6(15): 26~28.
- [15] Yuan-dong Pei, Sheng-li Wu, Shao-guo Chen et al. Sintering of solid waste generated in iron and steel manufacturing process in Shougang Jingtang(J). Journal of Iron and Steel Research. 2017, 6(30): 12~14.
- [16] Y. Shi, W. Zhang, Y. Cheng et al. Compound scan mode developed from subarea and contour scan mode for selective laser sintering(J). International Journal of Machine Tools and Manufacture. 2006, 6(15): 21~24.
- [17] Yeon-Tae Yu, Gautam Kumar Naik, Young-Bin Lim et al. Sintering Behavior of Spark Plasma Sintered SiC with Si-SiC Composite Nanoparticles Prepared by Thermal DC Plasma Process(J). Nanoscale Research Letters. 2017, 12(21): 25~29.

- [18] 唐庆利, 张建良, 李克江等. 烧结配料优化方法及计算对比分析[J]. 中国冶金. 2017, 5(1):15~19.
- [19] 吴敏, 王春生, 曹卫华. 基于预测模型与调整规则的烧结配料优化综合集成方法[J]. 系统仿真学报. 2008, 9(1):20~35.
- [20] 杨从锐, 钱谦, 王锋等. 改进的自适应遗传算法在函数优化中的应用[J]. 计算机应用研究. 2018, 35(4):1~10.
- [21] 卓金武, 李必文, 魏永生等. MATLAB 在数学建模中的应用[M]. 北京: 北京航空航天大学出版社, 2014:83~85.
- [22] 白晨光, 梁栋. 烧结配料、高炉生产及调度过程优化模型研究[J]. 知网. 2008, 10:8~21.
- [23] 杨东进, 陈继国, 于忠念等. 烧结配料优化分析[J]. 烧结球团. 2000, 25(1):1~2.
- [24] 王嵘冰, 徐红艳, 郭军. 自适应的非支配排序遗传算法研究[J]. 控制与决策. 2017, 12(19):2~3.
- [25] 任浩然. 基于自适应遗传算法优化的 BP 神经网络股价预测模型[J]. 延安大学. 2017, 6(1):12~15.
- [26] 胡建强. 冶金烧结配料过程优化控制系统的设计及开发[J]. 辽宁工业大学. 2016, 3(1):16~20.
- [27] 周常立. 烧结配料优化方法及工业应用[J]. 中南大学. 2011, 6(1):26~28.
- [28] 李勇, 吴敏, 曹卫华等. 基于线性规划和遗传-粒子群算法的烧结配料多目标综合优化方法[J]. 控制理论与应用. 2012, 1(14):26~29.
- [29] 吕学伟, 白晨光, 邱贵宝等. 基于遗传算法的烧结配料综合优化研究[J]. 钢铁. 2007, 4(15):12~16.
- [30] 牛德良. 水钢烧结配料经济优化方法及模型研究[D]. 重庆: 重庆大学, 2015.
- [31] 陈晓霞, 吴敏, 曹卫华等. 烧结配料多目标模糊满意优化的 BC-PSO 算法[C]. 上海: 上海译文出版社, 2012:16~20.
- [32] 王明. 基于群智能优化算法的烧结配料优化设计与应用[D]. 天津: 天津理工大学, 2012.
- [33] 向齐良, 吴敏, 侯奔等. 基于成分预测模型的矿石烧结配料专家优化方法[J].

山东大学学报. 2005, 8(3):9~13.

[34] 梁栋. 烧结配料、高炉生产及调度过程优化模型研究[D]. 重庆: 重庆大学, 2008.

[35] 黄玉明. 重钢烧结配矿优化研究[D]. 重庆: 重庆大学, 2005.

[36] 刘振. 翼钢烧结矿配矿优化研究[D]. 西安: 西安建筑科技大学, 2007.

[37] 侯奔. 基于成分预测的烧结配料专家优化方法研究[D]. 长沙: 中南大学, 2005.

[38] 贾文君, 张艳允, 魏琼花等. 邯钢烧结优化配料方案的研究及评价[C]. 邯郸: 邯郸出版社, 2014:23~26.

[39] 白晨光, 邹德余, 赵光甫. 烧结矿配料的优化研究[C]. 北京: 钢铁出版社, 1999, 12~16.

[40] 吕学伟, 白晨光, 邱贵宝等. 烧结配料优化研究[C]. 北京: 2006年全国炼铁生产技术会议暨炼铁年会文集. 2006, 36~40.

[41] 何大阔, 王福利, 贾明兴. 遗传算法初始种群与操作参数的均匀设计[J]. 东北大学学报. 2005, 9(1): 829~830.

[42] 唐世浩, 朱启疆. 遗传算法中初始种群与交叉、变异率对解的影响及其解决方案[J]. 科技通报. 2001, 5(5):1~5.

## 致 谢

在这里我要对在我学习生涯有过帮助的人表示谢意,是你们的帮助让我完成了三年的学习过程。

在此首先要非常感谢我的导师罗老师。在本论文工作中,从选题、研究,实现到论文最后成稿,都给予我非常大的帮助,三年的校园生涯即将结束在江财有很多的不舍,舍不得江财,舍不得师父,舍不得同学也舍不得曾经轰轰烈烈爱过的姑娘。是你们让我成长,是你们增加了我的智慧谢谢你们,在此,我向罗世华老师表示深深的谢意。

同样也要感谢师兄戴子安同学对我的帮助与支持。

最后特别要感谢的就是我的亲生父母,在学习生涯中是你们一直给予我鼓励,我才能坚定信念继续前进,是你们给了我生命又给了我学历,在这里谢谢爸爸和妈妈。

