中图分类号

密　　级　公开

学校代码　10224

学　　号　140710607

東北農業大學

硕士学位论文

基于实数遗传算法的有约束优化问题初始内点的求解方法研究

作　　者　方堃

导　　师　王福林　教授

学位类别　工学硕士

所在学院　工程学院

一级学科　农业工程

二级学科　农业系统工程与管理工程

二〇一七年六月

Classified Index: Code: 10224

Confidential (yes/no): no No. 140710607

Dissertation for the Master Degree

Research on the Method of Solving the Initial Interior Point of Constrained Optimization Problem Based on Real-coded Genetic Algorithm

Candidate: Fang Kun

Supervisor: Prof. Wang Fulin

Degree Category: Master of Engineering

College: College of Engineering

First level discipline: Agricultural Engineering

Second level discipline: Agricultural Systems Engineering and Management Engineering

Harbin China

June 2017

目 录

CONTENES

独 创 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，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过的研究成果，也不包含未获得______________（注：如没有其他需要特别声明的，本栏可空）或其他教育机构的学位或证书使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：方堃　　　　日期：2017年 6 月 8 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权学校可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。（保密的学位论文在解密后适用本授权书）

学位论文作者签名：方堃　　　　日期：2017 年 6 月 8 日

导 师 签 名：王福林　　　　日期：2017 年 6 月 8 日

摘 要

约束优化问题所涉及的领域非常广泛，为系统的优化和管理提供了有力的支持。约束极值问题的理论和方法来源于最优化理论和优化设计，主要研究内容是在众多的方案中如何找到那个最优的方案。内点法和复合形法等优化算法是求解这一类问题较为常用的方法。但在应用内点法和复合形法等优化算法时，首先要找到一个初始内点。对于有些问题，由于缺少先验知识，很难人为给出一个可行的初始内点，所以，对有约束优化问题初始内点的研究就有了重大的理论意义和应用价值。为此，本文在这一方面进行了深入的分析与研究，给出了基于实数遗传算法来求解约束优化问题初始内点的一种新方法。

本文通过对国内外有关文献的深入研究，采用数学、计算机科学等多学科的综合分析法，以制约函数法的相关理论为基础，提出了基于改进实数遗传算法求解有约束优化问题初始内点的模型，并进行了示例计算。取得的主要研究成果如下：

（1）本文经研究分析给出了求解约束优化问题初始内点的一种方法——实数遗传算法。相对传统方法需要约束条件是可导的，本文提出的新方法则没有这些要求，因此更具有普适性。

（2）提出了基于改进实数遗传算法求解有约束优化问题初始内点的数学模型。

（3）给出了用实数遗传算法求解约束优化问题初始内点时适应度函数的构造方法与模型。

（4）本文给出的进化策略具有如下特点：1）在交叉操作时将交叉概率取 1，以保证所有的父代配对个体都需要进行交叉，即：选择这样的交叉概率能够增加产生子代个体的数量，能够增大产生更加优秀个体的可能性，使算法的运算速度能够得到提高。2）将交叉产生的个体与父代中精英保留的 m 个个体放在一起排序，重新选出精英保留的 m 个个体。这样可以避免交叉操作产生的优秀个体在变异操作过程中遭到破坏的不足，使得交叉操作产生的优秀个体能够得到生存。3）对变异操作后的个体排序，然后用精英保留的 m 个优秀个体对变异操作后最差的 m 个个体进行替换，形成新的种群。这样既有效的保证了种群的多样性，同时又使得求得的最优结果不会比前一次求得的结果差。

（5）进行了实例计算，结果表明该方法是一种行之有效的方法。由于该方法不用对约束条件进行求导，所以较传统方法具有更广泛的实用性。

关键词：约束优化问题；初始内点；实数遗传算法；进化策略

Research on the Method of Solving the Initial Interior Point of Constrained Optimization Problem Based on Real-coded Genetic Algorithm

Abstract

The constraints optimization problem involves a wide range of fields, providing strong support for system optimization and management. The theory and algorithm of the constraint extreme value problem are derived from the optimization theory and method, the main research content is how to find the optimal scheme in many schemes. Optimization methods,such as interior point method and compound method are the most commonly used methods to solve this kind of problem. However, in the application of internal point method and composite method optimization algorithm, we must first find an initial point. For some problems, because of the lack of prior knowledge, it is difficult to give a feasible initial point, so the study of the initial point of the constrained optimization problem has significant theoretical significance and application value. Therefore, this paper makes a deep analysis and research in this field, and gives a new method based on real-coded genetic algorithm to solve the initial interior point of constraint optimization problem.

Based on the comprehensive research of domestic and foreign literatures, use of mathematics, computer science and other multi-disciplinary comprehensive analysis, Based on the theory of constrained function method, a model based on improved real-coded genetic algorithm for solving the initial interior point problem of constrained optimization problem is proposed, And the example is calculated. The main findings are as follows:

(1) In this paper, a method of solving the initial interior point of the constraint optimization problem is presented. The traditional method of solving needs to ensure that the constraint is have derivative, and the new method proposed in this paper does not have these requirements, so it is more universal.

(2) The mathematical model of the initial interior point of the constrained optimization problem based on the improved real-coded genetic algorithm is proposed.

(3) The construction method and model of the fitness function based on the real number genetic algorithm are given to solve the initial interior point of the constrained optimization problem.

(4) The evolutionary strategy presented in this paper has the following characteristics: 1) The crossover probability is taken at 1 to ensure that all parent pairs need to be crossed, choosing such

a crossover probability can increase the number of offspring individuals, can increase the possibility of generating more outstanding individuals, so that the algorithm can be improved speed of operation.2) The individual generated by the cross operation and the parent to retain the m elite individuals together to sort, re-select m elite individuals to retain, this avoids the fact that the excellent individuals produced by the crossover operation are destroyed during the mutation operation, so that the excellent individuals produced by the cross operation can survive.3) Sort the individuals generated by the mutation operation, then use the m individuals who were retained by the elite replaced the worst m individuals after the mutation operation, form a new population. This effectively ensures the diversity of the population, but also can make the optimal results obtained will not be worse than the previous results obtained.

(5) Carried out example calculation, the results show that the method is an effective method. The method does not need to seek derivative of constraints, so more practical than the traditional method.

Key Words: Constrained optimization problem; Initial interior point; Real-coded genetic algorithm; Evolutionary strategy

Candidate: Fang Kun
Speciality: Agricultural Systems Engineering and Management Engineering
Supervisor: Prof. Wang Fulin

1 引言

1.1 研究的目的与意义

各行各业中，会遇到大量关于控制与决策方面的问题，这些问题都需要用优化的方法来求取最终的结果。根据其约束条件的有无，便可以将优化问题划分为两种类型，即：第一种是有约束优化问题，第二种是无约束优化问题。其中，有约束优化问题是最优化理论和优化设计的重要内容，也是经济管理、农业生产以及其他许多研究方向最为常见的内容之一。其主要是研究从众多方案中选取最优方案的问题，其不仅可以有效的避免资源的浪费同时还可以产生巨大的经济效益和社会效益。有约束优化问题的全局最优解求解过程特别复杂，寻找可以大规模并行且具有智能特性的最优化方法也就有了重大的理论意义和应用价值。

目前较为常用的如遗传算法、模拟退火算法以及其他的一些智能优化算法都会对这些比较复杂的约束优化问题有比较好的解决方法[1-3]。内点法和复合形法是求解这一类问题最为常见的方法之一[4]。但在应用内点法和复合形法时，首先要找到一个初始内点，对于有些问题，由于缺少先验知识，很难人为的给出一个可行的初始内点。同时，确定初始种群的方法之一也是由一个满足约束函数的内点为基础，产生其余可行初始个体。为此，研究者对已有文献进行了深入的分析研究，在此基础之上给出了一种求解初始内点的新方法，基于实数遗传算法来求解有约束优化问题的初始内点。

遗传算法[5]具有其他基础优化方法所没有的优点：

（1）遗传算法与许多传统搜索方法的不同是：传统算法是以单点进行搜索，而该算法则是以许多可能解所构成的初始种群进行搜索，这种搜索遍布整个区域，所以大大增加了找到全局最优解的可能性。

（2）遗传算法一般通过适应度函数来评价个体是否优劣，对问题本身并没有过多的依赖，有较好的鲁棒性。

（3）遗传算法只需要待优化问题的目标函数和约束条件是可计算的即可，对其数学模型并没有严格的要求，因而该算法具有非常广泛的应用，普适性非常好。

（4）遗传算法不同于穷举法那样对可行域内的所有点进行搜索，也不同于随机法那样对可行域内的点进行随机的搜索，而是一种启发式的搜索方法。其主体思想是不断进行种群引导，最终可以使其收敛于最优解，只要能够合适的选择初始种群和遗传操作方式便就可以快速地找到最优解。

（5）遗传算法是并行计算方法的一种，可以更加快速的解决较为复杂的优化问题。

实数遗传算法采用十进制编码，不同于二进制编码的实数遗传算法，省去了编码和解码的过程，可以直接采用实数对个体进行表达，因此具有如下优点：

（1）通过二进制编码的遗传算法只能通过加长染色体的长度来增加数域的范围，但编码长度过长则会使得求解问题的效率极大的降低，而实数编码的个体与解向量是一一对应的，因此实数遗传算法便可以极大限度的使数域范围的表达得到增加。

（2）遗传算法的编码如果采用二进制编码形式，需要很长的编码才能使其能够获得较高的求解精度，编码解码的过程在编码加长时一定会浪费很多时间，相比而言，实数遗传算法与此截然不同，更容易使问题的求解变得方便。,

（3）如今我们在工程中所遇到的问题大多数是目标函数庞大、约束条件众多的约束优化问题，在较短的时间内求得最优解是问题的关键所在。二进制编码的遗传算法由于需要时间编码解码所以运算速度不尽人意，然而实数遗传算法没有此过程所以更高效，更适合求解复杂的大规模约束优化问题。

1.2 国内外研究动态

1.2.1 国内研究动态

詹士昌[6]在通过借鉴退火的自然过程，在不可行度的条件下，给出了约束优化问题的遗传算法，其与传统的约束优化问题处理方法相比使得满足约束条件与目标函数值达到最优之间的关系存在的缺点得以弥补了。首先，该算法在生成初始可行解时，通过利用退火不可行选择策略，然后再通过利用该遗传算法将解的搜索空间通过不断收敛，使其至可行域内部。其中，该遗传算法的策略是通过进行比较个体的不可行度和递减的阀值对个体是否被选择加以判断，并且使得被拒绝的个体能够让该子代群体中不可行度小的个体取代。

孙有发[7]等通过把混沌映射以及后天强化学习策略加入到传统的遗传算法中，提出了具有反馈机制的混沌遗传算法。该算法对混沌理论的随机性和遍历性的利用十分充分，因此便可以最大程度地使得种群多样性得以维持，这样就使得种群在演化过程中出现的提前收缩现象得以避免，纯随机的演化过程也在后天强化学习情况下得以克服。在算法中，通过嵌入防僵化算子，就保护了在迭代中有效的模式，同时，使得进化过程中的临时最优解一直处于更新之中。最后，以具体的例子验证该算法的有效性和鲁棒性。

梁昔明[8]等通过改进选择操作来改进遗传算法对问题的求解，该算法通过对可行解和不可行解进行选择主要是透过操作适应度函数值和约束违反适应度值的大小来进行的，将可行解与不可行解混合交叉。为了使变异后的个体避免进入偏离约束区域，算法必须通过设置非均匀变异和边界变异的方法来实现。另外，还通过把维变异的思想融入算法，这种方法可以维持种群多样性，并通过数值实验证明了其良好的鲁棒性。

徐斌[9]等提出了一种新的混合算法，该算法主要基于遗传算法和蚁群算法的理论基础，以解决 QOS（服务质量）约束的组播路由优化问题。其主要思想是通过遗传算法生成若干个优化解，并将这些优化解转变为蚁群算法的信息素初始值。作者为了更好的提高算法的性能采取如下措施：第一，通过设置了预处理机制；第二，建立备选路径集；第三，采用基于路径的树结构编码方式。

刘大莲[10]等提出了内外交叉遗传算法，该方法可以解决最优解出现在可行域的边界上或边界附近的情况。首要做法是把优化问题通过转化使其变为双目标优化问题，然后是利用该遗传算法对转化后的问题进行寻优。主要步骤是：第一，每进行一次的迭代，都需要保留某一比例的不可行解，并且使其与可行解进行交叉；第二，每一次迭代中的变异算子都采用粒

子群变异法则作为基础进行变异。算法在改进后，其迭代次的数少和种群规模变小的优势情况更为明显。同年，作者[11]又提出了一种新的改进遗传算法，主要按照种群中个体通过分类排队后以等级交叉的交叉算子和按照线性搜索的变异算子为改进，改进的算法提高搜索质量及收敛性。

鲁延京[12]等通过对处理约束优化问题所用智能算法的缺点和长处的深刻讨论分析，创造出了粒子进化变异遗传算法抹去已有算法的不足。已被改进的算法是对以前算法的三个方面进行了改进，即：对约束处理机制的改进、对变异算子的改进、同时维持种群多样性。通过改进机制使其避免算法过早陷入局限。其中，根据个体经过变异操作之后所产生的新的子代个体是否越过边界为条件，创造了三种变异算子，即：边界、区域和全局搜索粒子进化的变异算子。并且对其例证了算法的有效性、可靠性、稳定性，结果表明其稳定性较差，仍有待对其加以改善，但该算法的有效性和可靠性较好。

王福林[13]等提出了快速产生初始种群及新的子代个体的实数遗传算法，子代个体可以在合理的搜索方向上产生，并且经过交叉操作之后可以使得新的子代个体能够在不同的方向上产生，这样就可以保证多样性的子代种群。

金芬[14]等提出了的新实数遗传算法可以解有效的决解有边界的复杂函数优化问题。由目标函数值排序得出个体适应度值，采用线性逼近交叉策略以及精英保留等方法使子代个体高速的向高适应度区移动，加快算法的收敛速度。

1.2.2 国外研究动态

Deb[15]提出了联赛选择算子法。这种方法对两个个体选择的标准如下：第一是，对于二者中的一个为可行解，但是另一个却是非可行解的情况，则需要选择可行解；第二是，对于二者均是可行解的情况，选取目标函数值较小的个体；第三是，二者若均是非可行解，我们就要以较小违反约束选取个体。

Farmani[16]等提出的一种自适应表示法，该方法是基于典型自适应惩罚函数法而实施的改进。其惩罚过程首先是让种群中最差的不可行解的适应值能够与种群中最好解的适应度值相等或比其更高，其次是让种群中最差的不可行解的适应值与种群中最好解的适应度值相等。

Takahama[17]等提出了以α约束法定义：个体满足约束条件的程度用约束满足水平来表达，判断每个个体的优劣程度的依据是最终水平取值，两个个体的参数只取决其约束条件的违反程度或者目标函数值。

Piya Chootinan[18]为了更好地处理遗传算法中的约束条件而引入梯度法的理论知识，即为了修补不可行的解将梯度作为特殊的算子嵌入遗传算法结构中。其中，由约束函数的形式是否复杂，决定是从约束函数的显式表达式推导梯度信息还是通过有限差分法近似取得。每代种群中不可行解被修复的比例决定了这种算法的效率。只需要对不可行解集的一个子集进行修复是为了减少计算量；虽然修复率越高，收敛到最优解的概率越高，但计算量会变大，这就是只需对其中一个子集修复的原因。

Sushil Kumar[19]等通过研究提出了一种消除惩罚因子的实数遗传算法，其可以让不可行解在种群中得以存在以保障种群的多样性，模仿二进制遗传算法的交叉操作和变异操作，使

得单点交叉和比特翻转变异的不足得以避免，由测试结果知该算法的搜索性是优于传统遗传算法的。

Nedim Tutkun[20]等提出了能够将此算法应用到连续多峰类函数最优化问题的有效实编码交叉操作。为了限制个体在给定的搜索空间产生不同的个体，便采用线性和二次映射算法去进行，为了更好的寻优，采用牛顿局部搜索法去进行。

Omar Arif Abdul-Rahman[21]等提出了自适应混合遗传算法，给它使用最优控制参数估计且进行性能测试，结果显示该算法能够求解约束优化类问题并且具有良好性能。

目前比较流行的智能优化算法在求解比较复杂的问题时常常采用内点法。在应用内点法时，一般是人为的给出一个可行的初始内点。然而有些问题却很难人为给出一个可行的初始内点。现有文献对初始内点的求解方法并无太多详尽的阐述。因此本文在现有文献研究的基础上，对初始内点的求解方法作了进一步研究。

1.3 研究的主要内容、方法和技术路线

1.3.1 研究内容

本文通过对国内外相关文献的深入分析，结合数学、生物学、计算机学等多个学科的理论知识与技术方法，对实数遗传算法进行了改进研究，内容如下：

（1）介绍了实数遗传算法和约束优化问题的相关概念，分析研究了约束优化问题的基本思想、理论基础以及求解约束优化问题初始内点的相关成熟算法，为本文研究提供了有效的理论依据。

（2）介绍了算法的基本操作，包括：进化策略、遗传操作（选择、交叉和变异）、种群的初始化、进化终止条件、适应度函数的构造、以及算法的基本流程。同时，也深入的分析了传统的子代产生方法和算法的进化策略中存在的不足，为实数遗传算法的进一步改进研究打好基础。

（3）研究和改进子代产生方法中存在的不足，新的子代产生方法可以使子代个体更加快速的向更加优秀的搜索区域靠近，使算法的性能有所提高。

（4）将交叉操作的概率设定为 1，增大种群的交叉概率，使得每一个个体都能参与到交叉操作，能够增大优秀个体产生的可能性。

（5）针对实数遗传算法的传统进化策略的两方面缺陷：第一，在搜索方面表现出效率较低、第二、在计算方面比较容易出现早熟状况。本文对其进行了改进研究，提出了一种新的改进的进化策略。

（6）对改进后的实数遗传算法进行了测试分析，发现其性能优于传统的实数遗传算法，从而表明了改进方法的有效性。

（7）通过示例验证可知，虽然采用实数遗传算法来求解不同问题的初始内点所用的时间有所不同，但每次都能求得初始内点，从而证明了本文所提出的算法对求解约束优化问题初始内是行之有效的。

1.3.2 研究方法和技术路线

本文通过了解国内外的最新研究动态以及有关文献，从中发现目前国内外现有文献中应用遗传算法求解实际问题的优缺点，奠定了本文新方法提出的基础。采用数学、计算机科学等多学科的综合分析法，提出了基于改进的实数遗传算法来求解有约束优化问题初始内点的新方法。

技术路线能够清晰有效地反应文章的总体框架与布局，可以以明了的形式指导文章整体推进工作。具体路线如图 1-1 所示：

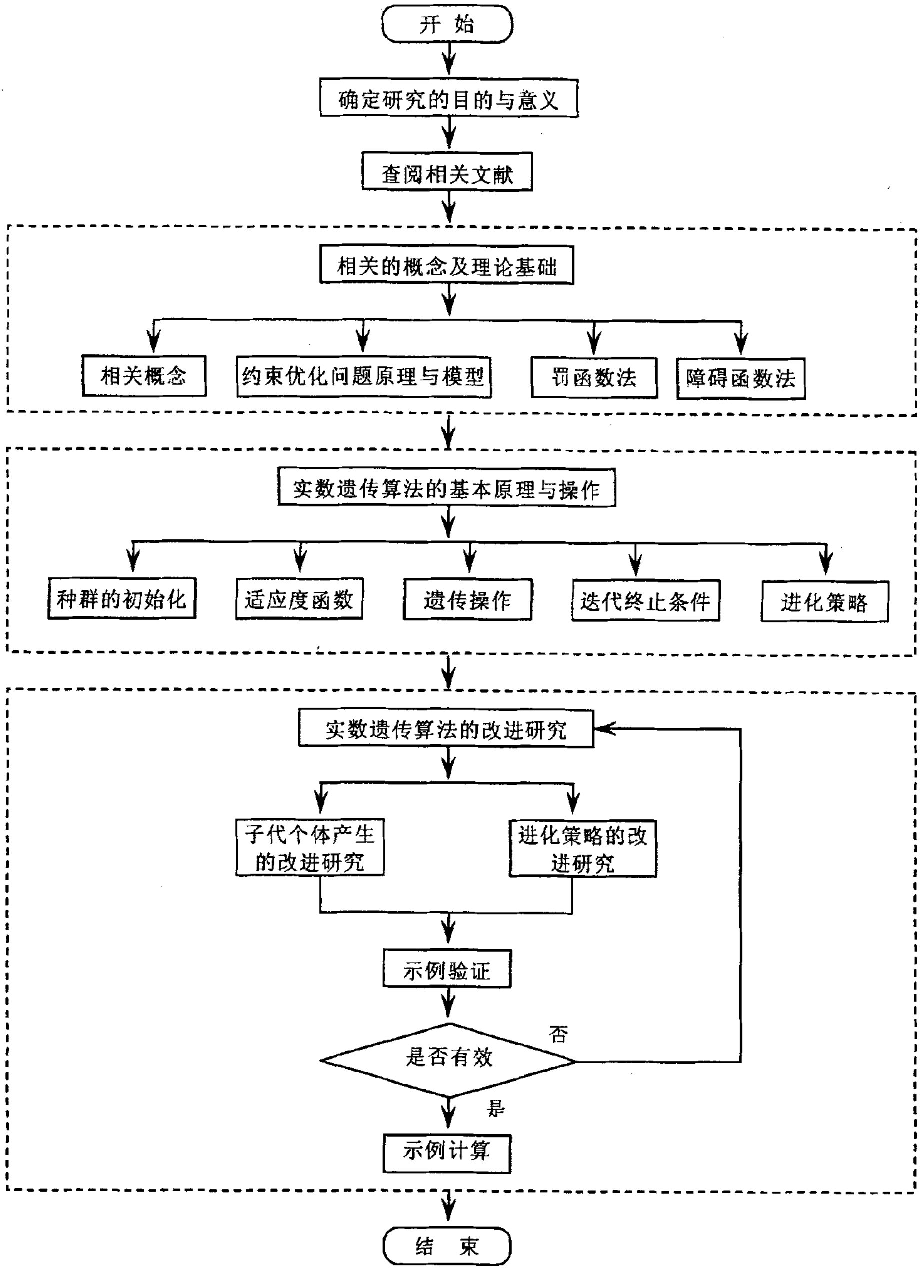

图 1-1 课题研究的技术路线图

Fig.1-1 The technical roadmap for research

2 概念界定与理论基础

2.1 相关概念

集合：具有某种确定属性的对象的全体就叫做集合。组成集合的每一个对象称为该集合的元素。

开区间：直线上介于两个固定点间的所有点的集合称为开区间（不包含这两个给定的固定点）。

邻域：某点附近的一切点构成的集合称为邻域，通常可以取以该点为中心的一个开区间作为邻域，以P为中心的任何开区间称为点P的邻域，记作$U(P)$。

内点：有集合E，若存在点P的某邻域$U(P)$，使得$U(P)$包含于集合E，那么P就称为集合E的内点。

外点：有集合E，若存在点P的某邻域$U(P)$，使得$U(P)$与集合E的交集为空集，那么P就称为集合E的外点。

界点：若P的任何邻域内既有属于E的点，又有不属于E的点，则称点P是E的界点。

可行域：可行解组成的集合叫做可行域。可行解是可以满足线性约束条件的解。可行域既可以有界，也可以无界。

个体：指染色体带有特征的实体，在优化问题中，个体指代问题的解。

种群：带有相似特征的染色体个体构成的集合就是种群，初始种群是由最初的可行个体构成的。

种群规模：种群中个体的个数称为群体的大小，即种群规模。

进化：生物为了在其延续生存的过程中使其品质不断得到改良，需要逐渐适应生存环境的变化，进化就是表示这种生命现象的一个代名词。一般情况下，生物进化都是以种群为单元进行的。

适应度：为了研究自然界中生物的遗传和进化现象，适应度的概念应用而生，它表示某个物种对生存环境的适应程度状况。低适应度生物在生存环境中将失去更多的繁殖机会，可能灭绝，反之，其繁殖机会就会相对增大，种群便会繁荣。

交叉概率：以一定概率来在解集中选取两个解进行交叉使其产生新的解，那么此处的概率就是交叉概率。

变异概率：从解集中以一定概率来选取一个解进行变异以便产生新的解，这个概率就是变异概率。

2.2 约束优化问题简介

2.2.1 无约束优化问题的模型

无约束优化问题的求解是数值计算领域的重要研究方向。一些约束优化问题也可以通过

转换为无约束问题来进行求解，也就是因为这样，无约束优化算法的求解效率会相应的影响到约束优化问题的求解。因此，研究无约束优化算法具有重要的价值。对于无约束优化问题，其数学模型可描述为：

$$\min f(x)=f(x_1,x_2,\cdots,x_n) \quad (2\text{-}1)$$

式中：x——设计变量向量，$x=(x_1,x_2\cdots x_n)^T$；

$x_i(i=1,2,\cdots,n)$——第 i 个设计变量；

$f(x)$——目标函数。

求解上述无约束优化问题的方法有很多，其中最为常用的方法就是迭代法，迭代法大体上可以分为解析法和数值法两类。解析法通过一阶、二阶导数等函数的解析性质，或者用一些 Hesse 矩阵构造出的优化方法等来对问题进行求解，如牛顿法、变尺度法、梯度法等方法都属于这一类方法。数值法在无约束优化问题的求解迭代过程中仅仅需要利用到函数的值而不需要利用到函数的解析性质，如单纯形搜索法、模式搜索法、坐标轮换法等方法都属于这一种方法。两种方法各有特点和自己的适用条件，大体上说解析法一般需要给出明确的解析式，而数值法在目标函数的解析式具有十分复杂的表达式，无法给出具体的解析式，或者无法求出解析式的导数时比较适合使用数值法，数值法的优点是计算时迭代步骤简单、变量较少，缺点是收敛速度较慢。数值法的基本原理和操作思想是，首先选择一个初始估计值，沿着一条对结果有利的方向进行搜索，找出一个新的估计值，然后继续循着一个有利的方向去搜索，找到下一个近似点，按照这种方法不断的进行迭代，一直到结果满足设定的精度要求。在这个过程中不要设置迭代次数，找出的是一系列与问题的解有关的近似值。

2.2.2 有约束优化问题的模型

约束极值问题的理论和算法是约束优化问题主要研究的内容，它来源于最优化理论和方法。近年来约束优化问题得到了迅速的发展，结合数学理论中其他分支的成果逐步形成了具备自身特色的学科。约束优化问题涉及的领域非常广泛，为系统的优化和管理提供了有力支持。常用的搜索方法一般要求所求目标函数必须满足如下条件：

（1）目标函数以及约束条件都连续可微。

（2）待优化问题的可行域是有界的。

（3）待优化问题存在最优解。

求解这类问题的方法就是所谓的约束优化方法。对于约束优化问题，其数学模型可描述为：

$$\begin{gathered}\min f(x)=f(x_1,x_2,\cdots,x_n)\\ s.t.\begin{cases}g_j(x)=g_j(x_1,x_2,\cdots,x_n)\geq 0, j=1,2,\cdots,p\\ h_i(x)=h_i(x_1,x_2,\cdots,x_n)=0, i=1,2,\cdots,q\end{cases}\end{gathered} \quad (2\text{-}2)$$

式中：x——设计变量向量，$x=(x_1,x_2\cdots x_n)^T$；

$x_i(i=1,2,\cdots,n)$——第 i 个设计变量；

$f(x)$——目标函数；

$g_j(x)$——第 j 个不等式约束条件；

$h_i(x)$——第 i 个等式约束条件；

n——设计的变量个数；

p——不等式约束的个数；

q——等式约束的个数。

令 D 是满足所有约束关系的解的集合，也就是待求解的约束优化问题（2-2）的可行解的集合，则（2-2）式的数学模型也写成如下形式：

$$\min_{x\in D} f(x)$$
$$s.t. D=\left\{x \middle| g_j(x)\geq 0, j=1,2,\cdots,p; h_i(x)=0, i=1,2,\cdots,q\right\} \tag{2-3}$$

虽然对约束优化问题求解理论的研究还不如无约束优化问题理论那么深入，但只要优化方法选取得当就能求得具有较高精度的解。在许多实际问题中，目标函数可能具有多峰、多极值的性质，会让以梯度法来求解这类问题受到相当大的限制，于是人们通过研究提出了智能优化算法，其中有一类便具备可求解不可导、多峰等数学特征问题，遗传算法恰恰是这类智能优化算法中的一种。

对于约束优化问题，可以利用罚函数法，对其进行一系列的操作使之变为无约束优化问题后再进行求解。将待求解问题的约束条件以恰当的函数方式使其能够与目标函数的表达式合并是罚函数法最为主体的思想，这样就可以使约束优化问题成功转化为无约束类型的，通过这些步骤后便以无约束优化的搜索方法对原始问题求解。罚函数法以搜索过程中解和可行域的关系可以分为两种类型：一是外部罚函数法，二是内部罚函数法两类。障碍函数法是内部罚函数法的另一个名称。对于罚函数法的最新研究几乎都集中在对其乘子法的研究上，拉格朗日乘子法就是其中的一种。

目标函数解的性质和极值问题是解决约束优化问题之前需要了解的两个问题，他们是待求解的约束优化问题中目标函数的基本性质，该内容主要包括两方面：一是目标函数解的性质，二是极值问题。

定义 1　称满足约束条件的点为可行点，称所有可行点的集合为可行集合（或可行域），记为 D。即：

$$D=\left\{x \middle| g_j(x)\geq 0, j=1,2,\cdots,p; h_i(x)=0, i=1,2,\cdots,q\right\} \tag{2-4}$$

定义 2　对式子（2-2），设 n 维空间中某点 $x^*\in D$，若存在 $\delta>0$，使得对一切 $x\in D$，且 $\|x-x^*\|<\delta$ 都有 $f(x^*)\leq f(x)$，则称 x^* 是 $f(x)$ 在 D 上的局部极小值点（局部最优解）。当 $x\neq x^*$ 时，若 $f(x^*)<f(x)$，则称 x^* 是 $f(x)$ 在 D 上的严格局部极小值点（严格局部最优解）。

定义 3　对式子（2-2），设 n 维空间中某点 $x^*\in D$，若对任意 $x\in D$，都有 $f(x^*)\leq f(x)$，则称 x^* 是 $f(x)$ 在 D 上的全局极小值点（全局最优解）。当 $x\neq x^*$ 时，若 $f(x^*)<f(x)$，则称 x^* 是 $f(x)$ 在 D 上的严格局部全局极小值点（严格全局最优解）。

2.3 制约函数法

制约函数法是对目标函数的约束条件加上某种制约函数，这就可以使得原有约束优化问题能够转为无约束优化问题来求解[22-24]。本文主要介绍罚函数法（即：外点法）和障碍罚函数法（即：内点法）两种方法。

2.3.1 罚函数法

以求解式（2-2）的非线性规划问题为例，可以将原有的线性规划转换成以下形式：

$$\min f(x)=f(x_1,x_2,\cdots,x_n)$$
$$s.t.\begin{cases}x\in R\subset E_n\\ R=\left\{x\middle|g_j(x)\geq 0,\left(j=1,2,\cdots,p\right)\right\}\end{cases} \tag{2-5}$$

式中：R——表示优化问题的可行域；

E_n——表示n维欧式空间。

构造函数$\psi(t)$：

$$\psi(t)=\begin{cases}0 & 当t\geq 0\\ \infty & 当t<0\end{cases} \tag{2-6}$$

将约束函数$g_j(x)$整体看作t，当x满足约束条件时，$g_j(x)\geq 0$，则$\psi(g_j(x))=0$；当x不满足该约束条件时，则$\psi(g_j(x))=\infty$。之后把每个约束条件下的上述函数形式并入（2-5）式的目标函数上，便能将其转化成式（2-7）：

$$\varphi(x)=f(x)+\sum_{j=1}^{p}\psi\left(g_j(x)\right) \tag{2-7}$$

将式（2-7）作为新的目标函数，化为求解无约束优化问题：

$$\min\varphi(x)=f(x)+\sum_{j=1}^{p}\psi\left(g_j(x)\right) \tag{2-8}$$

提前假设式（2-8）的极小点是x^*，由（2-6）式知，对于全部的j一定有$g_j(x^*)\geq 0$，即$x^*\in R$。则可以肯定x^*除了是式（2-8）的极小点以外还是原来的非线性规划问题（2-5）的极小值点。对（2-5）式所描述的非线性规划问题的求解，通过上述方法即可被转变成对式（2-8）所描述的无约束极值的问题的求解。

但是有许多时候，通过以上方法所构造的函数$\psi(t)$在$t=0$处是不连续的，换句话说该函数在此没有导数，这将会把许多行之有效的无约束极小化方法的应用拒之门外。所以为了能够有效的解决该问题，可将式（2-6）进行如下修改：

$$\psi(t)=\begin{cases}0 & 当t\ge 0\\ t^2 & 当t<0\end{cases} \tag{2-9}$$

以式（2-9）将函数$\psi(t)$做修改后，即：当$t\ge 0$时导数为0，当$t<0$时，导数为$2t$，则$\psi(t)$和$\psi'(t)$对任意t都是连续的。当$x\in R$时还满足：

$$\sum_{j=1}^{p}\psi\left(g_j(x)\right)=0 \tag{2-10}$$

当$x\notin R$时

$$0<\sum_{j=1}^{p}\psi\left(g_j(x)\right)<\infty \tag{2-11}$$

此时式（2-8）的极小点就不一定再是原问题（2-5）的极小点。但是，如果通过选取一个特别大的实数$M>0$，将式（2-7）转化为：

$$P(x,M)=f(x)+M\sum_{j=1}^{p}\psi\left(g_j(x)\right) \tag{2-12}$$

则当$x\in R$时，$P(x,M)=f(x)$；当$x\notin R$时，由于事先给定的M的值非常大，所以这将使得$M\sum_{j=1}^{p}\psi\left(g_j(x)\right)$也会很大，最终使得$P(x,M)$的值会非常大，其实就是对非可行点的一种“惩罚”。而且如果点x离可行域越远那么$M\sum_{j=1}^{p}\psi\left(g_j(x)\right)$也就会越来越大，即惩罚函数值会越来越大，但这对可行点没有任何的惩罚作用。当M的值足够大时，式(2-12)的无约束极小点$x(M)$便足够接近原始约束问题的极小点。当$x(M)\in R$时其就变为原约束问题的极小点。

这种将原问题通过转化成求取无约束条件问题的方法就是罚函数法，M被称做（惩）罚因子，$M\sum_{j=1}^{p}\psi\left(g_j(x)\right)$被称为惩罚项，则$P(x,M)$被为惩罚函数。

公式（2-12）可以简化成为如下形式：

$$P(x,M)=f(x)+M\sum_{j=1}^{p}\left[\min\left(0,g_j(x)\right)\right]^2 \tag{2-13}$$

从数学式上其与（2-12）式是一样的，当$x\in R$时，$P(x,M)=f(x)$；当$x\notin R$时，$P(x,M)=f(x)+M\sum_{j=1}^{p}\left(g_j(x)\right)^2$。需要加以重视的是：此处的$g_j(x)$为$x$点不满足的约束的集合。

通过引进阶跃函数：

$$u_j\left(g_j(x)\right)=\begin{cases}0 & 当g_j(x)\ge 0\\ 1 & 当g_j(x)<0\end{cases} \tag{2-14}$$

便可将（2-12）式改写成下式：

$$P(x,M)=f(x)+M\sum_{j=1}^{p}\left(g_j(x)\right)^2 u_j\left(g_j(x)\right) \tag{2-15}$$

假定对其中一个罚因子 M_1 有 $x(M_1)\notin R$，需要进一步加大罚因子的值。伴随罚因子的进一步增大，则罚函数 $P(x,M)$ 中的惩罚项 $M\sum_{j=1}^{p}\psi\left(g_j(x)\right)$ 的作用就会随之增大而增大，则 $\min P(x,M)$ 的解 $x(M)$ 对可行域会更接近一些，可行域 R 与它之间的“距离”会越来越小，当 $0<M_1<M_2<\cdots<M_k<\cdots$ 不断趋向无穷大时，假设点列 $\{x(M_k)\}$ 收敛，它便可从可行域 R 的外部逐渐趋于原问题的极小点。在没有找到最优点之前，那些迭代过程中产生的近似点一般都不在可行域的范围内，所以也成这种方法为“外点法”。

根据等式和不等式约束问题之间的相关性，对于一个等式约束问题：

$$\begin{gathered}\min f(x)=f(x_1,x_2,\cdots,x_m)\\ h_i(x)=h_i(x_1,x_2,\cdots,x_m)=0,(i=1,2,\cdots,q)\end{gathered} \tag{2-16}$$

罚函数采用如下形式：

$$P(x,M)=f(x)+M\sum_{i=1}^{q}\left(h_i(x)\right)^2 \tag{2-17}$$

当然对于那些包含等式约束和不等式约束的双约束一般非线性规划问题：如式（2-2），其罚函数一般表示如下为：

$$P(x,M)=f(x)+M\sum_{i=1}^{q}\left(h_i(x)\right)^2+M\sum_{j=1}^{p}\left(\min\left(0,g_j(x)\right)\right)^2 \tag{2-18}$$

或

$$P(x,M)=f(x)+M\sum_{i=1}^{q}\left(h_i(x)\right)^2+M\sum_{j=1}^{p}\left(g_j(x)\right)^2 u_j\left(g_j(x)\right) \tag{2-19}$$

当罚因子 M_k 取的足够大时，上述无约束问题的最优解便可以满足所有约束条件，从而使其成为原约束问题的最优解。

下面是罚函数法的迭代步骤：

（1）取第一个罚因子 $M_1>0$（例如 $M_1=1$），设其误差 $\varepsilon>0$，并令 $k:=1$。

（2）求下面无约束极值问题的最优解：

$$\min P(x,M_k) \tag{2-20}$$

其中 $P(x,M_k)$ 可取（2-18）式或（2-19）式的形式，设其极小点为 $x^{(k)}$。

（3）若存在某一个 j $(1\le j\le p)$，有

$$-g_j\left(x^{(k)}\right)>\varepsilon \tag{2-21}$$

或（和）存在某一个 $i\ (1\le i\le q)$，有

$$\left|h_i\left(x^{(k)}\right)\right|>\varepsilon \tag{2-22}$$

则取 $M_{k+1}>M_k$（例如 $M_{k+1}=cM_k$， $c=5$或 10），并且令 $k:=k+1$。然后再转回第（2）步；否则便需停止迭代，得到要求的点 $x^{(k)}$。

2.3.2 障碍函数法

得益于罚函数法特别而重要优点，就是 $P(x,M)$ 可在整个 E_n 空间内进行优化，至此变为可以非常方便的随意选取初始点。我们知道罚函数法一般都是在可行域外进行迭代的，这样的话就无法用中间结果直接作为解的近似解。因此，有两种情况便无法使用上述的罚函数法了：其一是求解的目标是为了能够方便观察目标函数值的改善情况，此时则会要求迭代的每一次近似解都是可行解，其二是目标函数在可行域内比较简单或者求取解比较方便但在可行域外的函数性质比较复杂、甚至是无定义的。

障碍函数法（内点法）的求解过程可以说与罚函数法完全相反。罚函数法求解过程是在可行域外部进行的，而相反的是障碍函数法的求解过程始终是在可行域内进行迭代的。参考罚函数法的方法，对原有问题的目标函数用函数叠加的方法进行改造，最后要使得改造后的目标函数尽可能的满足：改造后的目标函数与原目标函数的值在可行域 R 离边界比较远的域内要尽可能的接近；而在可行域 R 离边界比较近的域内改造后的目标函数值可以趋向于任意大的值。如果在求解时在可行域的内部取得初始迭代点，即以不是在可行域边界上的严格内点作为初始迭代点，那么在求解无约束极小值问题的迭代过程中，函数就会像一面围墙一样把迭代点挡在可行域 R 的边界之内，从而能够让迭代自始至终一直在可行域的内部运行。就此改造，新目标函数就被称为所谓的障碍函数。当然，当障碍函数满足该要求时，将一定不会在 R 的边界上取得极小值。这样的话经过改造的问题本质上成为一种求解无约束性质的极值问题，因为极小化过程中肯定不会包含可行域边界上的点，可以利用无约束极小化的求解方法进行求解。

以式（2-2）描述的非线性规划为例，在点 x 不断从可行域 R 的内部趋于边界时，则其中至少存在一个约束函数 $g_j(x)(1\le j\le p)$ 趋于 0。可以得到倒数函数：

$$\sum_{j=1}^{p}\frac{1}{g_j(x)} \tag{2-23}$$

和对数函数

$$-\sum_{j=1}^{p}\lg\left(g_j(x)\right) \tag{2-24}$$

都将无限制的被增大。可以通过把（2-23）式和（2-24）引入到式（2-2）所描述的非线性规划问题的目标函数 $f(x)$ 上构成一个新的目标函数来解决问题。为了能够尽可能的逼近式（2-2）所描述的极小值点，取实数 $r_k>0$，使之构成一系列无约束性质的极小化问题：

$$\min_{x\in R_0}\overline{P}(x,r_k) \tag{2-25}$$

其中

$$\overline{P}(x,r_k)=f(x)+r_k\sum_{j=1}^{p}\frac{1}{g_j(x)} \tag{2-26}$$

或

$$\overline{P}(x,r_k)=f(x)-r_k\sum_{j=1}^{p}\lg\left(g_j(x)\right) \tag{2-27}$$

此处，R_0 为所有严格内点的集合，即

$$R_0=\left\{x\middle|g_j(x)>0,j=1,2,\cdots,p\right\} \tag{2-28}$$

公式（2-26）和（2-27）右端的第二项就是障碍项，r_k 为障碍因子，函数 $\overline{P}(x,r_k)$ 为障碍函数。如果从其中一点 $x^{(0)}\in R_0$ 出发，通过无约束极小化方法对问题（2-25）迭代，随着障碍因子 r_k 的慢慢减小（$r_1>r_2>\cdots>r_k>\cdots>0$），障碍项的作用便也会慢慢变小，式（2-25）的解 $x(r_k)$ 就会不停的向原约束（2-2）的极小解点进行逼近。则会随着 r_k 的不断减小，式（2-2）的极小点出现于可行域 R 的边界上，其所起“障碍”作用慢慢降低，需要求取的障碍函数的极小点会不断地往边界上靠近，直到满足精度要求。

障碍函数法的具体操作过程如下：

（1）选取取第一个障碍因子 r_1，并且有 $r_1>0$（例如 $r_1=1$），规定一个允许误差 $\varepsilon>0$，并令 $k:=1$。

（2）构造出障碍函数，障碍项可见（2-23）式或（2-24）式的形式。

（3）对障碍函数在 R_0 内进行无约束极小化过程，设所得极小解为 $x^{(k)}\in R_0$。

（4）检查有没有满足下面的收敛准则：

$$r_k\sum_{j=1}^{p}\frac{1}{g_j\left(x^{(k)}\right)}\le\varepsilon \tag{2-27}$$

或

$$\left|r_k\sum_{j=1}^{p}\lg\left(g_j\left(x^{(k)}\right)\right)\right|\le\varepsilon \tag{2-28}$$

如果满足，则停止迭代，$x^{(k)}$ 就为原约束的近似极小解；否则取 $r_{k+1}<r_k$（例如 $r_{k+1}=\frac{r_k}{10}$ 或

$\frac{r_k}{5}$)，令 $k:=k+1$，转回第（3）步迭代过程。

2.4 本章小结

本章首先对相关的基本概念进行了详尽的阐释，其次对约束优化问题和制约函数法的相关理论基础以及数学模型进行了深入的分析。

3 实数遗传算法的基本操作

上个世纪 60 年代，美国密切根大学的 John Holland 教授等人创立了遗传算法这种优化算法，其基本思想源自于生物进化论和群体遗传学，这种群体寻优的搜索算法具有自然选择法则和基因遗传学原理的特点[25]。对于复杂和非线性问题，传统的搜索算法一般难于解决，但它却特别适用于处理这种问题，其被广泛地用在人工智能、人工生命、机器学习、智能系统、自适应控制、规划设计、组合优化、系统工程等领域，被认为是智能计算发展的关键技术之一。为了对种群中的个体实施各项遗传操作，可以在用实数遗传算法求解优化问题时，只依据适应度函数在不断的进化中来判定种群个体是否优劣[26-28]。

3.1 初始化种群

实数遗传算法的搜索开始于许多可能解所构成的初始种群，有可以搜索多个峰值的特点，并且因其并行性的特点而具有较高的搜索效率。初始种群是在整个搜索域中随机选取的点集，这种点集式的搜索方式比点到点的搜索方式有更大的可能找到最优解。种群规模的设定对遗传算法的性能具有重大的影响，种群大小要根据具体问题而设定。在实际应用案例中，种群的规模常取几百，一般建议在 20～500 范围内取值较为适宜。种群规模确定后就要对其进行初始化，在搜索空间内，选取初始种群时，对于有约束优化问题，可以在满足一定约束条件下随机选；对于无约束优化问题，可以以均匀分布的方式随机生成。

3.2 适应度函数

在自然选择法则中，对环境适应性高的个体繁衍下去的可能性高，对环境适应性低的个体逐渐被淘汰，适应度函数构成了整个生存空间。因此用适应度函数值来评价个体的优劣程度，它对算法的收敛速度和性能有很大影响。为了对个体的优劣情况有一个更加直观的评价，一般在应用中，适应度函数的设计一般需要满足以下四条条件：

（1）适应度函数要单值、连续、非负、最大化，这个条件是比较容易实现的。

（2）适应适应度函数要合理、具有一致性，要求适应度值可以真实的反应种群的优劣程度。

（3）适应度函数的计算量必须小，以简单的方式设计适应度函数，这样可以使计算的时长降低同时也可以使空间的复杂性变小，从而能够降低计算成本。

（4）适应度函数必须具有良好的通用性，适应度函数对某类具体的优化问题应尽可能具有通用性，最好不需要应用该适应度函数的人来改变其中的参数。

选择函数变换 T： $g \to f$ ，使得对于最优解 x^*， $\max f\left(x^*\right)=\operatorname{opt} g\left(x^*\right)\left(x^* \in[u,v]\right)$。

（1）以目标函数 $g(x)$ 直接转化为适应度函数 $f(x)$，即：

$$f(x)=\begin{cases}g(x) & \text{求最大值的优化问题}\\ -g(x) & \text{求最小值的优化问题}\end{cases} \tag{3-1}$$

该转换操作的方法虽比较简单直观，但是适应度函数的取值可能为负值，而且对于有些结果分布不均匀的优化问题如果采用此方法求解平均适应度值，则不能较好地体现出种群的优劣情况，从而对算法的性能产生影响。

（2）若待优化问题是最小值问题，对目标函数 $g(x)$ 和适应度函数 $f(x)$ 建立如下映射关系：

$$f(x)=\begin{cases}c_{\max}-g(x) & g(x)<c_{\max}\\ 0 & \text{其他}\end{cases} \tag{3-2}$$

式中：$c_{\max}$——目标函数 $g(x)$ 的一个预先设定的最大估计值。

若待优化问题是最大值问题，则对目标函数 $g(x)$ 和适应度函数 $f(x)$ 建立如下映射关系：

$$f(x)=\begin{cases}g(x)-c_{\min} & g(x)>c_{\min}\\ 0 & \text{其他}\end{cases} \tag{3-3}$$

式中：$c_{\min}$——目标函数 $g(x)$ 的一个预先设定的最小估计值。

该方法虽然是对方法（1）的改进，但仍然存在不足之处，当适应度函数设计不得当时，可能造成在算法刚开始时，产生一些较优的个体，因为竞争力优势，进而影响接下来的选择阶段，“优胜劣汰”的效果过于明显，降低了遗传算法的全局优化特性。

（3）若待优化问题是最小值问题，对目标函数 $g(x)$ 和适应度函数 $f(x)$ 建立如下映射关系：

$$f(x)=\frac{1}{1+c+g(x)} \qquad c\ge 0, c+g(x)\ge 0 \tag{3-4}$$

式中：c——目标函数 $g(x)$ 的界限保守估计值。

若待优化问题是最大值问题，则对目标函数 $g(x)$ 和适应度函数 $f(x)$ 建立如下映射关系：

$$f(x)=\frac{1}{1+c-g(x)} \qquad c\ge 0, c-g(x)\ge 0 \tag{3-5}$$

式中：c——目标函数$g(x)$的界限保守估计值。

此方法与方法（2）的思想基本类似，因此也存在着类似的问题。

（4）除了上述常用的三种构造适应度函数的方法外，实际应用中也经常采用基于序的适应度函数构造方法。首先将计算得到的个体的目标函数值以降序的顺序排列，然后按照某种关系进行映射，最后计算适应度值即可。适应度函数的这种计算方式，使得个体被选择只与被排列的顺序有关，概率与目标函数值并不相关。因为不是实际的目标函数值，所以不需要在意目标函数值的正负，同时也不需要加入其他参数到目标函数中。有基于顺序的线性适应度函数和基于顺序的非线性适应度函数两种基于顺序的适应度函数。

1）基于顺序的线性适应度函数构造方法：

计算个体的目标函数值，将目标函数排好序后，按公式（3-6）进行映射：

$$f_i = \frac{pop - i + 1}{pop} \qquad (i = 1, 2, \cdots, pop) \tag{3-6}$$

式中：i——排序后的第i个个体；

f_i——第i个个体的适应度值；

pop——种群的规模。

相邻的两个个体间的适应度值之差公式为：

$$f_{i+1} - f_i = -\frac{1}{pop} \qquad (i = 1, 2, \cdots, pop) \tag{3-7}$$

可见，通过这样设计适应度函数可以使个体之间差异变小，也可以使群体的平均性能只与种群数目有关，适应度函数值不会受影响，由此算法的效果可以被增强。

2）基于序的非线性适应度函数构造方法：

与基于顺序的线性适应度函数的构造方法相似，先对目标函数排序，然后按公式（3-8）进行映射：

$$f_i = \beta(1-\beta)^{i-1} \qquad (i = 1, 2, \cdots, pop) \tag{3-8}$$

式中：β——$\beta \in (0,1)$，一般取值范围在$0.01 \sim 0.3$之间。

令$\alpha = 1-\beta$，则有$\alpha \in (0,1)$，从而（3-8）式可简化为：

$$f_i = \alpha^{i-1} \qquad (i = 1, 2, \cdots, pop) \tag{3-9}$$

在遗传算法适应度函数的设计时有一定概率出现以下问题：

（1）在算法的进化前期，可能就会产生适应度值非常巨大的超常个体，如果按比例选择法进行选择，遗传算法的选择操作可能受到生存概率较大的超常个体影响，从而导致子代种群多样性较差，同时也会使算法陷入局部最优。

（2）在算法的进化后期，种群个体的适应度值将会接近最佳适应度值，从而在进行选择操作时其他的个体与最佳个体被选择的概率大致相同，种群有较小的进化可能性，容易陷入局部最优解。

为了有效解决上述问题，在算法的不同过程需要对适应度函数进行尺度变换，从而在实际差异不是很明显的前提下扩大种群中个体适应度之间的差异程度，使算法在选择操作时可以更好的区别不同个体，从而提高提高算法的性能使其收敛于全局最优解。常用的尺度变换方法如下：

（1）线性调整

令原来的适应度函数为 f，经调整之后的适应度函数为 f'，则线性适应度调整公式如下：

$$f' = af + b \tag{3-10}$$

a,b 为调整系数，在确定系数 a,b 时一般要满足以下两个要求：一是，以不改变种群个体适应度的均值为在进行选择操作时扩大种群个体适应度的差异，要求调整后的要等于原来适应度的平均值；二是，为了在选择操作中控制最佳个体被选择的次数，要求调整后的最佳个体适应度要相等于原来适应度平均值的指定倍数。其中指定的倍数为最佳个体在选择操作中希望被选择的次数，试验指出，当种群规模在 50~100 之间时，指定的倍数在 1.2~2.0 之间取值最为适宜。在适应度较低的个体中，如果 $f' < 0$ 时，则取 $f' = 0$。

（2）乘幂调整

令原来的适应度函数为 f，调整之后的适应度函数为 f'，则乘幂适应度调整公式如下：

$$f' = f^k \tag{3-11}$$

指数 k 的选择与具体实际问题有关，需要进行反复不断地修正试验来确定效果理想的 k 值。该方法由 Gillies 在机器视觉试验中提出，经过反复不断地修正试验确定较为理想的 k 值为 1.005。

（3）指数调整

令原来的适应度函数为 f，经调整之后的适应度函数为 f'，指数适应度调整公式如下：

$$f' = e^{-af} \tag{3-12}$$

模拟退火过程是这种变换的思想的来源，当 a 较大时，选择种群中最佳个体的概率会随之增大；当 a 较小时，选择种群中最佳个体的概率会随之减小。

3.3 遗传操作

遗传算法通过模拟生物界的繁殖、杂交和突变现象进行遗传操作，遗传操作的基本操作算子包括选择（selection）、交叉（crossover）和变异（mutation）。这几种操作算子是遗传算法强大的搜索能力的核心组成部分，决定了遗传算法搜索性能的优劣，是遗传算法的进化搜索过程得以实现的基础。遗传算法是在选择、交叉和变异的三种基本算子的共同作用下从初始种群开始搜索的，逐步地向目标函数的最优解收敛、逼近。三个算子在整个搜索的过程中作用各有不同、缺一个不可，只有将不同操作算子的特征进行合理有效利用，才是提高遗传算法的性能的必备方法[29-33]。

3.3.1 选择

选择即是从当前的种群中选出优秀的个体进行交叉操作，这样既可以使优良基因得以保护，又可以使父代的优良特性可以更好地被子代去继承。选择可以使种群实现优者胜出劣者淘汰，在种群的进化过程中有着重要作用。通过评价个体适应度来确定如何进行选择，如果个体的适应度越高，则个体越优秀，被选中的概率越大；相反的，如果个体的适应度越低，则其被选中的概率越低，淘汰率越大[34-38]。遗传算法的选择操作，在进化论中，是适者生存的原则完美得的体现了。然而对于复杂的优化问题，传统的选择操作方法后期常常会出现子代多样性差、收敛到局部最优解等问题，针对此问题，本文主要介绍适应度比例选择法，也就是通常所说的轮盘赌选择法，该方法基本思想如下：

假设群体为 $P=\left\{x_1,x_2,\cdots,x_{pop}\right\}$ 的种群规模为 pop，群体中第 j 个个体 x_j 的适应度值为 $f\left(x_j\right)$，其中 $x_j\in P$，则个体 x_j 的选择概率就为 p_j 为：

$$p_j=\frac{\mathrm{Fit}\,f\left(x_j\right)}{\sum_{i=1}^{pop}f\left(x_i\right)} \tag{3-13}$$

种群中第 j 个个体 x_j 的累积概率为 q_j 为：

$$q_j=\sum_{i=1}^{j}p_i, j=1,2,\cdots,pop \tag{3-14}$$

当 $j=0$ 时，$q_0=0$；当 $j=pop$ 时，$q_{pop}=\sum_{i=1}^{pop}p_i=p_1+p_2+\cdots+p_{pop}=1$。同时规定，个体 $x_1,x_2,\cdots,x_{pop-1},x_{pop}$ 的选择区间分别为 $[0,q_1),[q_1,q_2),\cdots,[q_{pop-2},q_{pop-1}),[q_{pop-1},q_{pop}]$，则个体 $x_j\left(j=1,2,\cdots,pop\right)$，选择概率 p_i、累积概率 q_i、选择区间 $[q_{i-1},q_i]$ 之间的关系见表 3-1。

表 3-1 轮盘的选择概率 p_i、累积概率 q_i、选择区间 $\left[q_{i-1},q_i\right)$ 之间的关系

Tab.3-1 The relationship between the roulette selection probability p_i、cumulative probability q_i、selection interval $\left[q_{i-1},q_i\right)$

个体	x_1	x_2	$\cdots$	x_{pop-1}	$x_{pop}^{(k)}$
选择概率	p_1	p_2	$\cdots$	p_{pop-1}	p_{pop}
累积概率	q_1	q_2	$\cdots$	q_{pop-1}	q_{pop}
概率区间	$[0,q_1)$	$[q_1,q_2)$	$\cdots$	$[q_{pop-2},q_{pop-1})$	$[q_{pop-1},1]$

当产生[0,1]区间上均匀分布的随机数r，则r必落在概率区间的某个范围，则对应该区间的个体就被选中，称这种选择方法为轮盘赌选择方法。轮盘赌选择方法也可以用图形来表示，见图 3-1。

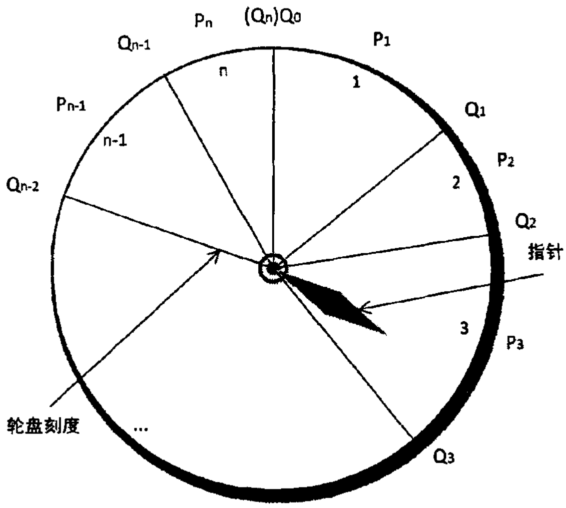

图 3-1 赌盘赌示意图

Fig.3-1 The sketch of betting

在轮盘赌示意图中，轮盘赌的刻度相当于累积概率，轮盘上任意两个相邻刻度的距离相当于对应个体的选择区间。当旋转指针，指针停留的位置相当于随机数的值，指针停留的区间对应的个体，就是被选中个体。轮盘的图形表示形象直观，易于理解。

例如：假设种群规模为$pop=6$的，则$P=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，个体的适应度值分别为：$f(x_1)$=16.3709、$f(x_2)$=8.2094、$f(x_3)$=36.1992、$f(x_4)$=15.5438、$f(x_5)$=21.3787、$f(x_6)$=24.3834，对应的选择概率分别为：$p_1$、$p_2$、$p_3$、$p_4$、$p_5$、$p_6$，对应的累加概率分别为：$q_1$、$q_2$、$q_3$、$q_4$、$q_5$、$q_6$，其计算结果如下：

p_1=0.1375、p_2=0.0689、p_3=0.3040、p_4=0.1053、p_5=0.1795、p_6=0.2048，q_1=0.1375、q_2=0.2064、q_3=0.5104、q_4=0.6157、q_5=0.7952、q_6=1。

每次通过转动轮盘选择一个个体，共转动两次轮盘选择两个个体，设在$[0,1]$区间产生的两个随机数r_1和r_2分别为：r_1=0.3716、r_2=0.8024。对于r_1=0.3716，在q_2和q_3之间，则第三个个体被选中；对于r_2=0.8024，在q_5和q_6之间，则第六个个体被选中。这两个被选中的个体将进行交叉操作来产生子代个体。

3.3.2 交叉

把两个父本个体的部分结构通过替换重组产生子代就是交叉操作[39-42]。交叉算子的具体操作方式是通过对自然界中基因重组方式的模仿而形成的，在交叉过程中起到核心作用。经交叉操作产生的个体继承了父代的优良基因，由此组成了更复杂基因的新个体，增加了子代适应不同环境的可能，确保了种群的多样性和进化，同时也提高了算法的搜索能力。

（1）离散交叉

实数遗传算法的离散交叉操作是通过交换两个个体的基因数组来生成新个体，离散交叉操作不会产生新的基因只是将父代的基因进行重组。下面就离散交叉做简单介绍：

设两个被选择参与交叉的父代个体如下所示：

父代个体 1：14　37　19

父代个体 2：43　26　25

经过离散交叉后的子代个体的每个变量可按同一概率随机的选择其父代个体所对应的变量，其子代个体可能为：

子代个体 1：43　26　19

子代个体 2：14　26　19

（2）中间交叉

实数遗传算法的中间交叉操作可以由父代产生具有新基因型的子代个体，该操作方法可以产生遍历甚至超出父代所能到达的搜索空间，加大了种群多样性的可能。下面就中间交叉做简单介绍：

按如下公式产生子代个体：

$$\text{子代个体}=\text{父代个体 }1+\alpha\text{（父代个体 }2-\text{父代个体 }1\text{）} \tag{3-15}$$

其中 α 为比例因子，可在 $[-d,1+d]$ 上均匀分布随机数产生，当 $d=0$ 时为绝对中间交叉操作，经验指出一般选择 d=0.25。

设两个被选择参与交叉的两个父代个体如下所示：

父代个体 1：10　30　15

父代个体 2：45　20　25

假设随机生成的比例因子 α 值如下所示：

子代个体 1 的 α：0.2　1.2　0.4

子代个体 2 的 α：0.8　0.5　0.6

生成的子代个体为：

子代个体 1：17　18　19

子代个体 2：38　25　21

（3）线性交叉

线性交叉操作在实数遗传算法中与中间交叉操作相类似，区别之处在于每个子代个体的所有变量只有一个 α 值，下面就线性交叉做简单介绍：

设两个被选择参与交叉的两个父代个体如下所示：

父代个体 1：10　30　15

父代个体 2：45　20　25

假设随机生成的比例因子 α 值如下所示：

子代个体 1 的 α：0.6

子代个体 2 的 α：0.8

生成的子代个体为：

子代个体 1：38　24　21

子代个体 2：31　22　23

现有的实数遗传算法在通过选择操作来确定交互信息的个体时所采用的方法是随机法，从而在选择个体进行交叉操作时可能使带有更多有价值信息的优秀个体被忽略；同时，对父本做交叉操作以后，如果直接使得子代个体进入变异操作，而不是在对子代个体评估后再进行的话，可能会使交叉操作的优化信息没能被有效的利用导致经过交叉操作后的优秀个体进行变异操作，可能会破坏交叉操作后所产生的优秀个体。算术交叉算子与启发式交叉算子是实数遗传算法的两种基本交叉算子，它们的搜索方式可以理解为以其中一个父代个体为起点沿着不同的方向进行以均匀概率搜索。使其能够产生更好的效果，一般把二者结合起来使用。

由于种群中适应度值高的个体在进化过程中被选择进行交叉操作的概率偏大，因此种群漂移的规律在很大程度上是由实数遗传算法的优势搜索反映的。与此同时，如果种群通过进化达到一个确定程度时，产生优势交叉和有效交叉的概率便会随着种群平均适应度值的逐渐增大而随之减小。种群的进化程度一般由交叉操作的有效性去直接影响。如果种群通过进化达到一个确定程度时，且新个体产生于原来父本邻域附近，则其在下一代中更容易被保留下来，即交叉操作便会以较大限度转化为优势交叉和有效交叉。

单次交叉操作类与线段上的随机搜索是相似的。在进化过程的不断进行中，种群集中的趋势便不断趋向于各优势群体。在算法进化初期，种群以分散形式分布，标准交叉算子处于随机搜索状态，这种搜索可以认为是局部范围的随机搜索，这种随机状态和两个父本个体在其连线进行搜索是一样的。优势交叉和有效交叉的可能性在此期间是相当大的，原因是原个体的适应度值以及种群的平均适应度值在该期间都是相当小的。由于进化的慢慢推进，种群就会不断向小生境集聚。由相关理论可知交叉操作有两部分构成：一是邻域搜索、而是局部搜索。在这种情况下邻域搜索有较大概率能够实现优势交叉和有效交叉操作。因为随着进化的不断进行，局部搜索的优势交叉和有效交叉的可能性随着种群的平均适应度值与原个体的适应度值逐渐增大而减小。

基本交叉算子是线性交叉算子的一种，父代个体间连线及其延长线上是其有效的搜索范围。它以随机搜索的方式在确定线段进行搜索，并且它在计算过程中不会在乎父代种子的优劣状态。这种搜索是在优秀个体的附近能够以更高的概率搜索到比其更好的个体。但是算数交叉算子对此却有不同的表现，其不管对于优秀种子还是劣势种子附近都会以等概率产生新个体；而启发式交叉算子对此又有不同表现：其不管在优秀种子一侧还是在劣势个体一侧的延长线都有可能产生新个体，这说明其还需对做进一步改进。基本交叉算子还有一个特点是不需要将考虑父代种子到产生新个体间函数值的变化。

3.3.3 变异

实数遗传算法中的变异操作是通过对生物进化论中的遗传变异现象的模拟而得到，进而把其引入到了算法中，就是等位基因以随机的形式代换个体上的一个或者多个基因，不仅可以使子代个体产生新的特性而且还可以将选择和交叉过程中破坏的重要的基因找回[43-45]。使用变异操作在实数遗传算法中的作用不仅能够使算法的局部搜索能力增强，也能够增加种群的多样性的，避免了算法陷入局部收敛，也避免了种群出现非成熟的收敛。

（1）方法一：

步长的选择一般视具体的情况而定，较小的步长与较大步长相比，其成功的可能性较大，但较大的步长在一定条件下会有较快的效率。通过研究，一般采用变异算子如下：

$$x' = x \pm 0.5L\Delta \tag{3-16}$$

式中：Δ——$\Delta = \sum_{i=0}^{m=1} \frac{a(i)}{2^i}$，通常 $m = 20$，$a(i)$ 以概率 $1/m$ 取值 1，以概率 $1-1/m$ 取值 0；

L——变量的取值范围；

x——变异前变量取值；

x'——变异后变量取值。

该变异操作方法步长的选择应该根据实际具体的问题所要达到的优化效果来确定，在优化过程中步长既可以固定不变有时又可以改变。

（2）方法二：

设需要进行变异操作的个体为 $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})^{\mathrm{T}}, i \in \{1,2,\cdots,m\}, j \in \{1,2,\cdots,n\}$，$x_{ij}$ 为第 i 个个体的第 j 个分量的值，其变异操作的基本公式如下：

$$x_{ij} = \begin{cases} x_{ij} + r_{ij}(b_j - x_{ij}) & r_{ij} > 0.5 \\ x_{ij} - r_{ij}(x_{ij} - a_j) & r_{ij} < 0.5 \ , j = 1,2,\cdots,n \\ 0.5x_{ij} + 0.25(b_j + a_j) & r_{ij} = 0.5 \end{cases} \tag{3-17}$$

式中：a_i——设计变量 x_i 的取值上限；

b_i——设计变量 x_i 的取值下限。

为了有效的防止变异后操作后所产生的个体会和父体个体之间的差异较大，保证子代个体对父代个体更好的继承性，一般变异概率的取值都很小。

3.4 进化终止条件

一般的进化终止条件是指，对每一次进化后所得到的新种群中每个个体的目标函数值进行计算，如果相邻两次（本次和上次）进化后种群中个体的平均目标函数值相差小于或者等于一个已经给定的精度 ε，即，可以满足下列条件：

$$\left| f\left(x^{(t+1)}\right) - f\left(x^{(t)}\right) \right| \le \varepsilon \tag{3-18}$$

式中：$f\left(x^{(t+1)}\right)$——表示第 $t+1$ 次进化后种群的平均目标函数值；

$f\left(x^{(t)}\right)$——表示第 t 次进化后种群的平均目标函数值。

便可停止迭代，终止进化。

3.5 进化策略

遗传算法能否收敛于全局最优解以及其寻优效果的优劣与否都会受到进化策略的影响，好的进化策略就如同求解数学题时所使用的简便算法一样，求解效率比其它的方法更快并且会有更高的准确率[46-49]。实数遗传算法尽管选择的参数相同但如果所选择的进化策略不同，算法的求解效果也会是截然不同的。能够更有效地利用群体中的个体得益于实数遗传算法所采用的进化策略的高效性，同时高效的进化策略能够最快的寻找出其中更为优秀的个体，也能够避免算法陷入局部最优。

De Jong KA 于 1975 年在其博士论文中提出了精英保留方法，通过更深层次的研究，提出了新的进化策略，这种策略具有多精英保留和复制转化为选择的特性。Rudolph 于 1994 年以马尔可夫链为理论基础证明了遗传算法的精英保留策略的收敛概率为 1，同时也由此说明了种群的精英保留策略可以使得优化问题有效的收敛于最优解。因此，当前的遗传算法几乎都是运用这种进化策略。

通过计算初始种群中产生的个体的目标函数值，并以目标函数的优劣为基本原则对其进行排序，再通过计算每个个体的基于序的适应度函数值，并将优秀的个体保留下来。接着对父代个体做选择操作、交叉操作和变异操作，并对变异后个体的目标函数值进行计算，并根据计算结果进行排序。最后用前述保留的优秀个体替换变异后相应个数的较差的个体，使其形成新的种群。通过计算新产生的种群中个体的目标函数值，检验其是否符合终止条件，不符合时，做下一次迭代，符合则终止计算。其中需要特别注意的是，传统的进化策略在进行交叉后没有保留较为优秀的个体，这些没有被保留的优秀个体可能在其后的变异操作过程中被破坏掉，从而导致种群整体的质量下降，因此容易使算法的收敛速度过慢，并且可能使其陷入局部最优解，得不到令人满意的结果。

实数遗传算法的传统进化策略具体步骤如下所示：

（1）确定种群规模、交叉概率、变异概率以及进化次数上限等遗传参数，根据待优化问题的约束条件以及变量的取值范围产生符合要求的初始种群。

（2）设置迭代次数$(t=0)$。

（3）形成父代种群，并对种群中父代个体的目标函数值进行计算，根据计算所得目标函数值的优劣情况对父代种群进行排列并对计算其适应度，排序后，保留父代种群中的 m 个优秀个体。

（4）采用轮盘赌选择法对父代个体进行选择操作，利用线性交叉的方法对选择操作所选定的个体进行交叉操作，然后再对交叉后所生成的新个体进行变异操作。

（5）对变异操作后所产生的种群个体进行目标函数值的计算，并且根据目标函数值的优劣对其进行排序，用父代个体中所保留的 m 个优秀个体对变异后所产生的个体中 m 个最差个体进行替换，形成新的子代并对该其的目标函数值进行计算。

（6）令 $t=t+1$，检验经过遗传操作所新产生种群中个体的目标函数值有没有符合迭代终止条件的规定，符合迭代终止条件时终止计算；做下一次迭代，若不符合迭代终止条件，则令该子代种群代替原始父代为新的父代种群，返回（3）。

在进行交叉操作之后，实数遗传算法的传统进化策略不能够对所产生的优秀个体进行保

留，在随后的变异操作过程中有可能致使这些优秀的个体很遭到破坏，从而影响进化的效率，影响算法的性能[50-53]。

3.6 实数遗传算法的基本流程

（1）对问题建立优化模型，以确定以下遗传参数：种群规模大小、变异概率、交叉概率以及进化次数上限等。

（2）根据参数产生初始种群。

（3）定义和计算适应度函数。

（4）通过选择操作来选定进行交叉操作的父代个体，一般可采用轮盘赌选择法来确定进行交叉操作的个体。

（5）对种群进行交叉操作，为了确定子代个体各交叉的位置，采用产生随机数的方法进行。在交叉过程中，算数交叉算子是在父本连线之间直接产生两个新个体，并使其替换父代个体。

（6）对种群进行变异操作，一般所用的变异算子分为两种：一是均匀变异，二是非均匀变异算子。均匀变异算子的特点在于其对种群进行变异是在问题取值的上下限之间以随机的方式进行的。非均匀变异与此不同的是需要把进化个体函数值和进化代数考虑在其中，变异产生的个体以更大的概率分布在较优解附近。

（7）通过检查终止条件，得到其符合条件输出的最优解，不符合则返回（4）。

下面是传统实数遗传算法的流程图 3-2：

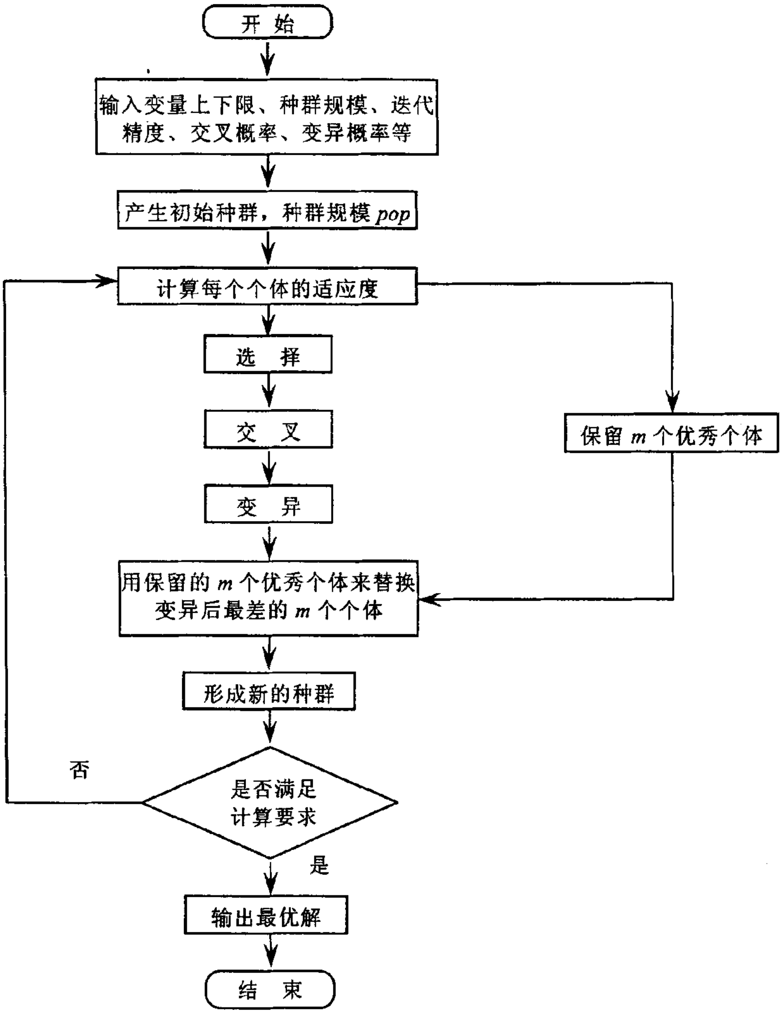

图 3-2 传统实数遗传算法的流程图

Fig.3-2 The flow chart of traditional RCGA

3.7 本章小结

本章主要通过系统性的概述，介绍了实数遗传算法的一些基本操作，其中包括：遗传操作（即：选择、交叉和变异）、种群的初始化、进化策略以及算法、进化终止条件、适应度函数的构造的基本流程。实数遗传算法的改进研究就是在这些比较成熟的基本操作的基础上进行的。同时，本章也对子代的产生方法以及算法的进化策略存在的不足进行了深入分析，为算法的改进研究做了基础。

4 实数遗传算法的改进研究

本章在现有文献的研究基础之上，主要是针对于实数遗传算法的交叉算子和进化策略进行了改进研究[54-57]，以提高算法的性能。

4.1 交叉算子的改进研究

4.1.1 改进后的交叉算子

假设通过选择操作确定选择父代个体 $x_j^{(0)}$ 与父代个体 $x_i^{(0)}$ 进行配对，同时又有式子 $\mathrm{Fit}\,f\left(x_i^{(0)}\right)>\mathrm{Fit}\,f\left(x_j^{(0)}\right)$ 成立，则可按下式产生一个子代个体 $x_j^{(1)}$

$$x_j^{(1)}=\gamma x_i^{(0)}+(1-\gamma)x_j^{(0)} \tag{4-1}$$

式中：$\gamma=\dfrac{\mathrm{Fit}\,f\left(x_i^{(0)}\right)}{\mathrm{Fit}\,f\left(x_i^{(0)}\right)+\mathrm{Fit}\,f\left(x_j^{(0)}\right)}$；

$l=\{1,2,\cdots,n\}$。

然后再求 $x_j^{(1)}$ 关于 $x_i^{(0)}$ 的映射点 $x_i^{(1)}$，即：

$$x_i^{(1)}=x_i^{(0)}+\lambda\left(x_i^{(0)}-x_j^{(1)}\right) \tag{4-2}$$

式中：λ——映射系数，通常取值大于 1 的数。

映射点 $x_i^{(1)}$ 的位置见图 4-1 所示。

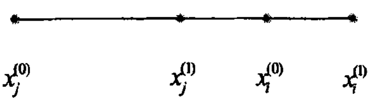

图 4-1 $x_j^{(0)}$、$x_j^{(1)}$ 与 $x_i^{(0)}$、$x_i^{(1)}$ 的相对位置图

Fig.4-1 The relative location of $x_j^{(0)}$、$x_j^{(1)}$ and $x_i^{(0)}$、$x_i^{(1)}$

对于有约束优化问题，按照式子（4-1）和式子（4-2）来产生的个体 $x_i^{(1)}$ 和 $x_j^{(1)}$ 不一定满足约束条件，若 $x_i^{(1)}$ 和 $x_j^{(1)}$ 全不满足约束条件，则可以使其不断地向 $x_i^{(0)}$ 靠拢，按照式子（4-3）和式子（4-4）来进行迭代：

$$x_i^{(1)}=x_i^{(0)}+\alpha(x_i^{(1)}-x_i^{(0)}),\alpha=\alpha/2 \tag{4-3}$$

$$x_j^{(1)} = x_i^{(0)} + \alpha(x_j^{(1)} - x_i^{(0)}), \alpha = \alpha / 2 \tag{4-4}$$

通过不断地迭代，最终能够使 $x_i^{(1)}$ 和 $x_j^{(1)}$ 迭代成为可行个体。

如果经过检验后发现 $x_i^{(0)}$ 为界点，并且通过式子（4-1）和式子（4-2）共同产生的两个子代个体 $x_i^{(1)}$ 和 $x_j^{(1)}$ 不都满足约束条件时，则需令：

$$x_i^{(1)} = x_i^{(0)} \tag{4-5}$$

$$x_j^{(1)} = x_j^{(0)} \tag{4-6}$$

为了提高运算速度，当产生的子代个体 $x_i^{(1)}$ 可以满足约束条件，并且还比 $x_i^{(0)}$ 优秀，就可以令 $\lambda = 2\lambda$，再通过（4-2）式从新求取 $x_i^{(1)}$，通过这样不断进行，一直到求得的 $x_i^{(1)}$ 不能够满足约束条件或者所求的已不如前一次的 $x_i^{(1)}$，这时保留前一个 $x_i^{(1)}$。

通过上述方法就能够产生所有子代个体。

4.1.2 改进后的交叉算子测试分析

基本测试思路如下：在种群规模均为 100，二者都采用基于序的适应度函数构造法、轮盘赌选择法、改进后的进化策略，其他参数也均一致的前提下对改进交叉算子的算法和采用前文所述线性交叉算子的算法分别测试。进行测试时，要保证改进交叉算子后的算法与未被改进的算法除交叉策略不同之外其他的操作步骤均一致。

测试函数 $f_1(x,y)$ 如下：

$$f_1(x,y) = 0.5 - \frac{\sin^2\sqrt{x^2+y^2} - 0.5}{\left(1+0.001\left(x^2+y^2\right)\right)^2}, -10 < x, y < 10 \tag{4-7}$$

测试函数 $f_1(x,y)$ 的图像如 4-2：

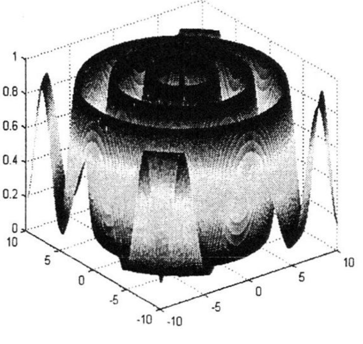

图 4-2 函数 $f_1(x,y)$ 的示意图

Fig.4-2 Schematic Diagram of Function $f_1(x,y)$

测试函数 $f_2(x,y)$ 如下：

$$f_2(x,y)=\left(4-2.1x^2+\frac{x^4}{3}\right)x^2+xy+\left(-4+4y^2\right)y^2,-10<x,y<10 \tag{4-8}$$

测试函数 $f_2(x,y)$ 的图像如 4-3：

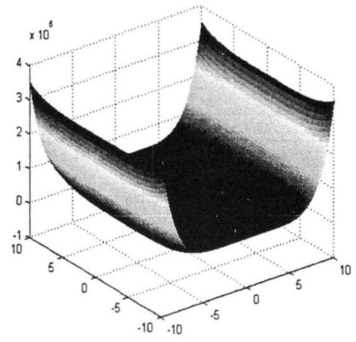

图 4-3 函数 $f_2(x,y)$ 的示意图

Fig.4-3 Schematic Diagram of Function $f_2(x,y)$

测试函数 $f_1(x,y)$ 在 $-10<x,y<10$ 范围内虽然只有一个全局最大值点（0，0），在点（0，0）处取得全局最大值 1，但在最大峰周围有两个分别取值为 0.990284 和 0.962776 的圈脊，不断优化的过程中容易在这些局部极大点陷入到局部最优解中。通过原始数值计算知测试函数 $f_2(x,y)$ 有两个全局最小点，即：（-0.0898,0.7126）和（0.0898,-0.7126），函数的最小值为 -1.031628，但函数 $f_2(x,y)$ 总共是有六个局部极小点。因此，常通过它们来评价算法的性能。

运行结果如下：

表 4-1 两种方法的测试结果

Tab.4-1 The test results of two methods

函数	方法	变量 x	变量 y	函数 f 的值	误差	运行代数
f_1	传统交叉策略	-3.560712E-0.2	-9.009123E-0.3	0.998650	0.001350	78
	改进交叉策略	1.317752E-0.4	1.012450E-0.5	1	0	21
f_2	传统交叉策略	-1.0061990	0.6947317	-1.028417	-0.003211	53
	改进交叉策略	-0.0897771	0.7126523	-1.031628	0	14

测试表明，改进后的子代产生方式可以使算法可以通过较少的迭代次数取得最优解，同时具有较高的精度。改进后的子代产生方法使得新子代个体的移动方向趋向于优秀的区域，从而有效的提高了寻优效率[58-60]。

4.2 进化策略的改进研究

4.2.1 改进后的进化策略

通过深入的分析和测试实数遗传算法传统的进化策略，从中可以看出传统的进化策略虽

然能够对父代的优秀个体进行保留却还是不能对交叉后所产生的优秀个体进行有效地保护，这可能致使之后的变异操作将这些优秀的个体破坏掉，使这些优秀个体不能得到生存，致使算法的性能被影响。

实数遗传算法传统的进化策略中，无论交叉概率选取什么值，对算法收敛性会产生不利的影响。一方面，在其传统的进化策略中，对父代的优秀个体进行保留之后，按一定的概率进行父代个体之间的选择交叉操作，随后接着进行变异操作，然而该操作后的个体很有可能会含有保留的父代中的优秀个体，如果仍然用保留的优秀父代个体对变异后所产生的较次的个体进行替换，这种做法将会使种群多样性变差，对种群的进一步进化产生不利影响，导致算法的早熟，容易陷入局部最优解。另一方面，如果交叉的概率选为 1（每个父代个体都进行交叉操作），交叉操作后就进行变异操作，此时通过在交叉操作中得到的优秀个体有很大概率被随后的变异操作破坏掉，从而对实数遗传算法优秀个体的产生有不利影响，最终导致算法的收敛性能受到不利的影响[61-64]。

针对实数遗传算法传统进化策略的上述问题，本文提出如下改进措施：

（1）取消交叉概率（交叉概率为 1），以增加较大的可能性产生优秀个体并对所有的个体都进行交叉操作。

（2）对父代种群的优秀个体和交叉后产生的种群中的优秀个体进行保留，这样的进化策略可以使父代中的优秀个体不会在选择操作中遭到破坏，同时在变异操作中，也可以不破坏交叉后所产生的优秀个体，从而使交叉操作后所生成的优秀个体和父代种群中的优秀个体不会丢失。

实数遗传算法的传统进化策略具体步骤如下所示：

（1）确定交叉概率、变异概率、种群规模以及进化次数上限等遗传参数，根据待优化问题的约束条件以及变量的取值范围产生符合要求的初始种群。

（2）设置迭代次数 $(t=0)$。

（3）形成父代种群，并对种群中父代个体的目标函数值进行计算，通过对父代种群以计算所得目标函数值的优劣情况排列并对计算其适应度，最后对排序后父代种群中的 m 个优秀个体进行保留。

（4）采用轮盘赌选择法对父代个体做选择操作，利用线性交叉的方法对选择操作所选定的个体进行交叉操作，将交叉后所产生的与父代所保留的 m 个优秀个体放在一个集合中，即临时种群中，对其进行排序并将 m 个优秀个体进行保留，然后再对交叉后所生成的新个体进行变异操作。

（5）对变异操作后所产生的种群个体进行目标函数值的计算，并且根据目标函数值的优劣排序，以之前步骤保留的 m 个优秀个体对变异后所产生的个体中 m 个最差个体进行替换，形成新的子代并对该子代个体的目标函数值进行计算。

（6）令 $t=t+1$，检验通过（5）所产生的子代种群中个体的目标函数值与符合迭代终止条件相比，看其是否已满足，符合迭代终止条件时终止计算；做下一次迭代，如果还是不符合迭代的终止条件，则令，此时该子代种群成为新的父代种群，返回（3）中继续循环。

通过对改进进化策略后实数遗传算法的分析，我们发现以此改进后的进化策略有三点好处：一是其能使父代中优秀的个体进行保留，二是能够使交叉所产生的子代中的优秀个体得

以保留，三是下一代个体中较差的个体可以被代替。这样可以使进化后的那代种群个体优于上代种群，避免变异操作破坏交叉操作所产生的优秀个体。更为关键的是：将父代种群中优秀的个体与交叉操作后所产生的优秀个体均得以保留，使其能够进入到下一代循环之中，这样使得后代优秀个体得以增加，从而使种群向最优方向上的收敛速度加快。

交叉操作的重要作用在于可以使好的模式聚集，研究者为此想要通过交叉操作所产生的个体要比父代更为优秀。因为交叉算子应该使用多少的频率是由交叉概率决定的，则可以知道经过交叉操作有 $p_c \cdot pop$（pop 表示种群的规模）个新个体产生在新产生的种群中，所以种群子代更新速度的快慢速度由 p_c 来决定，如果能把 p_c 降低，则种群的更新速度会被抑制，反之如果能把 p_c 升高，则会越强新区域的搜索能力。但交叉操作常常破坏了父代中性能良好的优秀基因，在实数遗传算法的进化策略被改进后，由于把交叉操作后所产生的 pop 个新个体与父代种群中所保留的 m 个优秀个体混合在一起再保留出 m 个优秀个体，所以即使优秀的个体会在交叉操作中被破坏，但在新的子代种群中不可能致使原父代种群及其通过交叉操作产生新个体中的优秀个体丢失。因此，在改进进化策略的实数遗传算法中，我们把交叉概率 p_c 的值取 1.0，使父代种群中所有的个体都能够加入交叉操作中，可以增加产生优秀个体的可能性，让交叉操作所产生的个体数量达到最大。由此可知，通过改进的进化策略，便很好地改善了实数遗传算法的性能。

通过对实数遗传算法传统的进化策略和改进后的进化策略的分析可知，改进后的进化策略将交叉概率设为 1.0，所以通过交叉操作后产生的新个体数目能够得到增加，从而时算法找到优秀个体的可能性会增加；同时，改进后的进化策略将父代中优秀的个体和交叉操作后产生的优秀个体都进行了保留，可以避免父代中的优秀个体不会在进行交叉操作时被破坏，同时也能够避免交叉操作后产生的优秀个体在变异操作时被破坏，能够增加子代中优秀个体的数量，提高迭代和寻优的效率。

4.2.2 改进后的进化策略测试分析

基本测试思路如下：在种群规模均为 100、计算精度均为10^{-8}、最大运行次数均为 500，其他参数也均一致的前提下对改进后的算法和未改进的算法分别测试 100 次。进行测试时，要保证改进之后的算法与未被改进的算法除进化策略不同之外其他的操作步骤一致，都采用相同的基于序的适应度函数的构造方法、轮盘赌选择方法、线性交叉方法、变异方法二，只测试不同进化策略对算法的影响。

测试函数 $f_1(x,y)$ 如下：

$$f_1(x,y)=\min f(x,y)=100*\left(y-x^2\right)^2+\left(x-1\right)^2,-10\le x,y\le 10 \tag{4-9}$$

测试函数 $f_1(x,y)$ 的图像如 4-4：

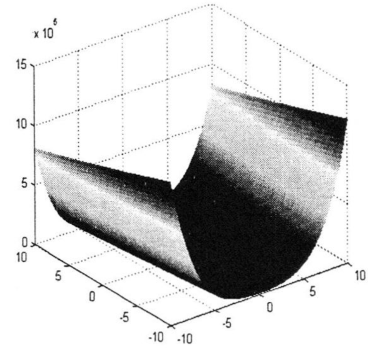

图 4-4 函数 $f_1(x,y)$ 的示意图

Fig.4-4 Schematic Diagram of Function $f_1(x,y)$

测试函数 $f_2(x,y)$ 如下：

$$f_2(x,y)=\max f(x,y)=-\frac{3600\left(\sin^2\sqrt{x^2+y^2}-0.5\right)}{\left[1+0.001\left(x^2+y^2\right)\right]^2}+\left[\frac{3}{0.1+x^2+y^2}\right]^2+\left(x^2+y^2\right)^2 \tag{4-10}$$

$$-4\le x,y\le 4$$

测试函数 $f_2(x,y)$ 的图像如 4-5：

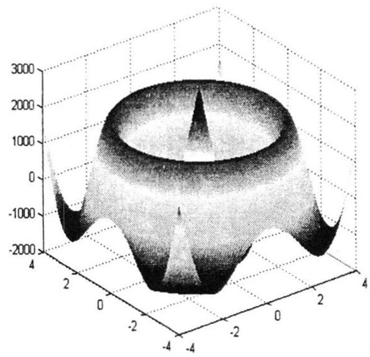

图 4-5 函数 $f_2(x,y)$ 的示意图

Fig.4-5 Schematic Diagram of Function $f_2(x,y)$

测试函数 $f_1(x,y)$ 是 Rosenbrock 函数，在点（1,1）处取得全局最小值 0。虽然该函数在 $-10\le x,y\le 10$ 内只有一个全局极小点，但是由于它的函数值在最小值点邻近的狭长区域内变化很缓慢，是典型的病态函数，很难极小化。测试函数 $f_2(x,y)$ 虽然在 $-4\le x,y\le 4$ 范围内只有一个全局最大值点但却有无限多个局部最大值点，并且存在一个均值为 2698.6 的圈脊包围在最大值 2700 的附近，有 4 个局部极值点分布在该圈脊的周围，导致算法很容易在该圈脊周围收敛。因此，常用它们来评价算法的性能。

运行结果如下：

表 4-2 两种方法的测试结果

Tab.4-2 The test results of two methods

函数	方法	平均计算时间/s	最长计算时间/s	最短计算时间/s	平均迭代次数/次	平均迭代次数/次	最小迭代次数/次
f_1	传统进化策略	2.2	5	0	327.6	500	11
	改进进化策略	0.1	1	0	12.8	21	5
f_2	传统进化策略	1.7	4	0	416.9	500	22
	改进进化策略	0.1	1	0	15.3	18	12

测试表明，与传统进化策略的实数遗传算法相比，改进后进化策略的算法在计算时间和迭代次数上明显更优。因而，本文给出的进化策略要优于传统的进化策略，具有更快的运算速度[65-66]。

4.3 改进后实数遗传算法的计算步骤

（1）对问题建立优化模型，确定种群规模、进化次数上限、交叉概率以及变异概率等遗传参数。

（2）根据相应的参数产生初始种群。

（3）定义适应度函数并对适应度值进行计算。

（4）通过选择操作来选定进行交叉操作的父代个体，一般可采用轮盘赌选择法来确定进行交叉操作的个体。

（5）对选择操作中选中的父代个体行交叉操作，将交叉概率设为 1。通过产生随机数的方法来确定父代个体进行体交叉操作的位置。在交叉操作过程中，先对父本个体的好坏进行比较，然后在适应性较差的父本到适应性较优的父本延长线上来生成新的子代个体，并用新生成的子代个体对父代个体进行替换。

（6）对交叉操作后所产生的个体进行变异操作时，一般选择均匀变异算子和非均匀变异算子来进行。其中，在问题的取值上下限范围内采用随机法对种群中的个体进行变异的方法为均匀变异算子变异法；而非均匀变异不同于均匀变异的随机性，而是将个体的函数值以及进化代数都进行考虑，这样会增大变异操作后所产生的新个体落在最优解附近的概率。

（7）检查终止条件，符合条件输出最优解，否则返回（4）。

改进后的实数遗传算法的流程如图 4-6 所示：

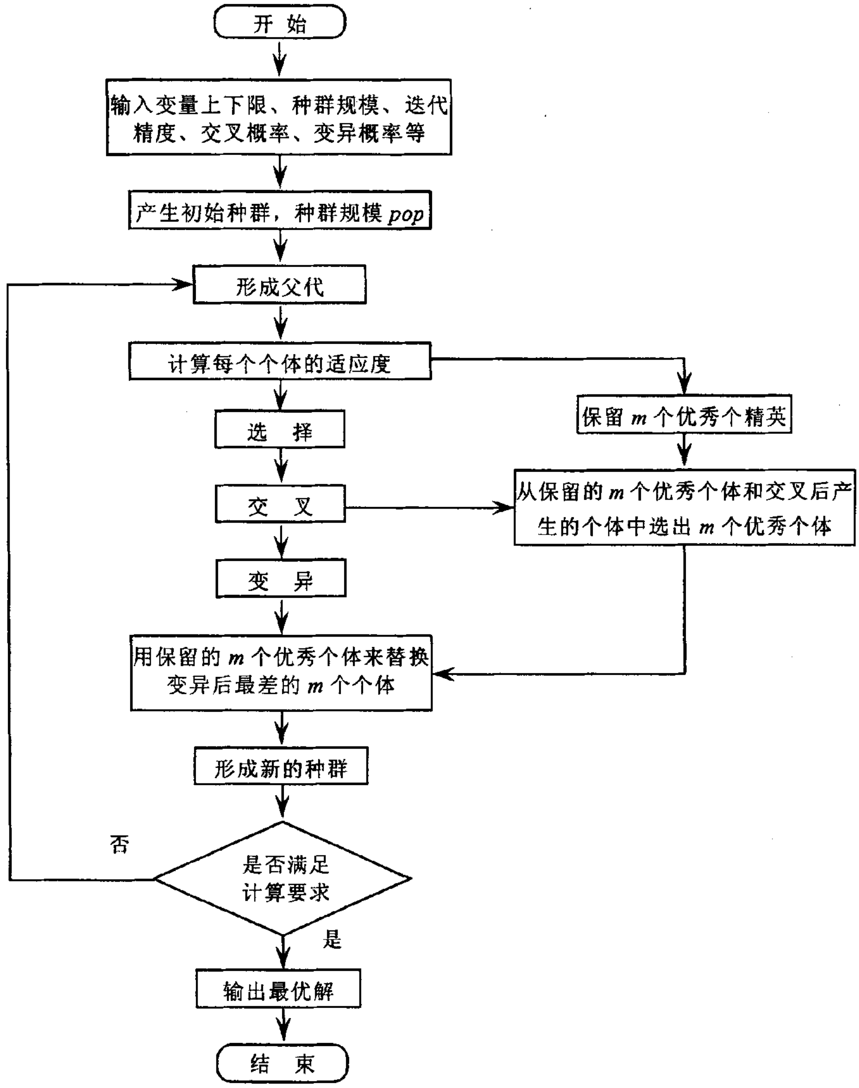

图 4-6 改进后的实数遗传算法流程图

Fig.4-6 The flow chart of improved RCGA

4.4 本章小结

本章主要对实数遗传算法进行了以下改进研究：

（1）对交叉操作过程中的子代的产生方法进行了系统的研究与深入分析，提出了新的子代产生方法，该方法能够使得产生的子代个体更为优秀的搜索区域进行。

（2）考虑传统进化策略的不足之处，提出了新的优秀个体保留策略，不但对父代种群中优秀的个体进行了保留，同时还对交叉操作后所产生的优秀个体也进行了保留，使其避免受

到随后变异操作的破坏。

（3）本文同时对选择概率进行了深刻的思考，把交叉概率设为 1，增大种群的交叉概率，使得每个个体都参与到交叉操作，能够增大优秀个体产生的可能性。

5 基于实数遗传算法的约束优化问题初始内点的求解方法

5.1 求解约束优化问题初始内点的方法与模型

首先任取一点 $x^{(1)} \in E_n$，若该点能够严格满足所有的不等式约束条件，则可以把此点作为待求解约束优化问题的初始内点。但如果该点不能严格满足所有的不等式条件，则以该个体不能严格满足的不等式约束来充当假拟目标函数，能够严格满足的那些不等式约束作为障碍项来构造函数，把原来的有约束优化问题转换成一无约束性质的问题。对这一无约束问题进行求解，可以得到一个新点 $x^{(2)}$，若 $x^{(2)}$ 仍然不为原问题的内点，就减小障碍因子按照上述方法继续进行计算，直到求出初始内点为止。

其求解方法与模型为:

任取一点 $x^{(1)} \in E_n$，$r_1 > 0$（例如取 $r_1 = 1$），令 $k := 1$。确定指标集 T_k 和 $\overline{T}_k$：

$$T_k = \{j \mid g_j(x^{(k)}) > 0, 1 \le j \le p\} \tag{5-1}$$

$$\overline{T}_k = \{j \mid g_j(x^{(k)}) \le 0, 1 \le j \le p\} \tag{5-2}$$

检查（5-2）式是否为空集，若 $\overline{T}_k$ 为空集，则 $x^{(k)}$ 就是初始内点；若 $\overline{T}_k$ 不是空集，则构造函数：

$$P(x, r_k) = -M \sum_{j \in \overline{T}_k} g_j(x) + r_k \times \sum_{j \in T_k} \frac{1}{g_j(x)} \quad (r_k > 0) \tag{5-3}$$

求函数 $P(x, r_k)$ 的最小值，即：

$$\min P(x, r_k) = -M \sum_{j \in \overline{T}_k} g_j(x) + r_k \times \sum_{j \in T_k} \frac{1}{g_j(x)} \quad (r_k > 0) \tag{5-4}$$

上述优化模型是一个无约束优化模型，传统的求解方法一般要求约束条件具有导数而采用实数遗传算法来求解则没有此要求。由于 M 值很大，式（5-4）中第一项的值也很大，这就相当于对非内点个体的“惩罚”，而且 X 点离可行域越远，惩罚越严厉。因此，迫使非内点的个体向可行域内快速移动，从而有利于快速求得初始内点。

5.2 基于实数遗传算法求解有约束优化问题初始内点的方法与模型

先随机产生1个个体，检查它是否严格满足所有不等式约束，如果满足，则此个体可以作为初始内点，输出该点即可；如果不满足，则把该个体保存到集合 X 中并继续生成第2个个体，检查其是否严格满足所有不等式约束，如果满足，则此个体可作为初始内点，输出该点即可；如果不满足，则把该个体继续保存到集合 X 中并继续生成下一个个体。在随机生成的个体总个数小于种群的规模 n 时反复进行上述步骤，要么会找到可行的初始内点要么会产生 n 个都不满足要求的随机个体（已把不满足要求的所以个体都保存到了集合 X 中），若找到了可行的初始内点输出即可；若产生了都不满足要求的个体则对其进行如下处理：以该个体不能严格满足的不等式约束来充当假拟目标函数，能够严格满足的那些不等式约束作为障碍项来构造函数，并把该函数作为目标函数，把集合中的个体分别带入目标函数，即可得到所有个体的目标函数值。接下来，根据所有个体对应的目标函数值，得到所有个体的适应度，进而进行选择、交叉、变异等操作产生新的种群，再依次判断新种群中的每个个体是否严格满足所有不等式约束，如果满足计算停止，该个体即可作为初始内点进行输出；如果新种群中的个体仍都不满足所有不等式约束，则重新对新种群的目标函数进行构造、计算适应度值、进行相关遗传操作（选择、交叉、变异），经过多次的迭代总能得到严格满足所有不等式约束的点，则该点就是要求的初始内点。

采用实数遗传算法来求解约束优化问题初始内点的具体步骤如下所示：

（1）由于目标函数 $P(x_i^{(k)},r_k)>0$，则可根据目标函数构造如下适应度函数：

$$\mathrm{Fit}\,f(x_i^{(k)},r_k)=\frac{1}{P(x_i^{(k)},r_k)} \tag{5-5}$$

（2）按照实数遗传算法的基本原理，对其进行选择、交叉、变异等相关操作，得到新的种群。

（3）对新得到的种群中的个体依次检查 $\overline{T}_k$ 是否为空集，若有点满足 $\overline{T}_k$ 为空集，则该点为可行内点输出即可；若所有的点都不满足 $\overline{T}_k$ 为空集，则用新得到的种群替换原来的种群重复进行上述步骤，直到找到满足要求的初始内点。

基于实数遗传算法求解约束优化问题初始内点的具体流程如图5-1所示：

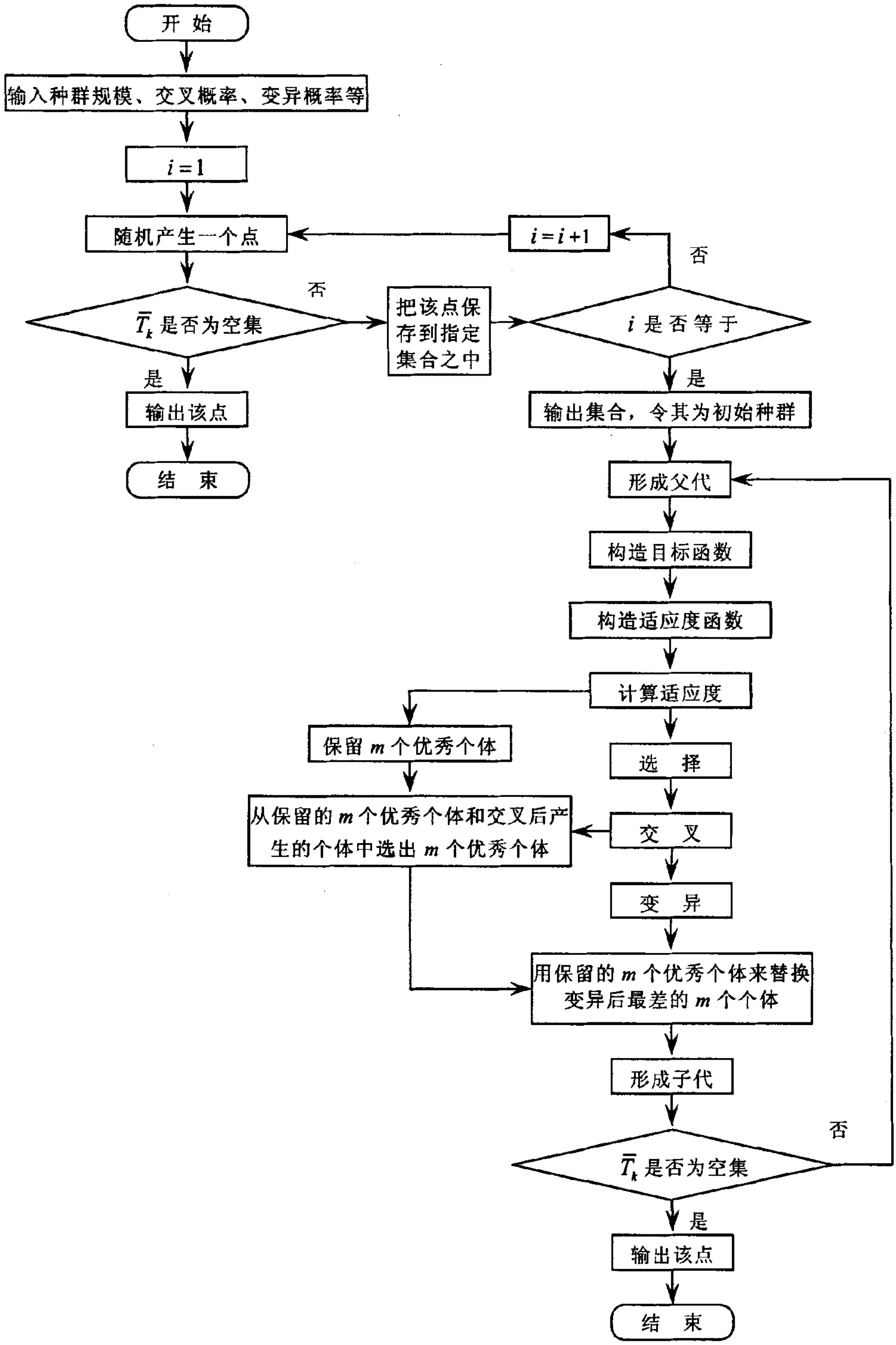

图 5-1 基于实数遗传算法求解初始内点的流程图

Fig.5-1 A flowchart based on RCGA for solving initial interior point

5.3 实例验证

本文选取了如下 4 个计算实例：

实例 1：

$$
s.t.\begin{cases}0.0025(x_4+x_6)\le 1\\0.0025(x_5+x_7-x_4)\le 1\\0.01(x_8-x_5)\le 1\\-x_1x_6+833.33252x_4+100x_1\le 83333.333\\-x_2x_7+1250x_5+x_2x_4-1250x_4\le 0\\-x_3x_8+1250x_5+x_3x_5-2500x_5\le 0\\100\le x_1\le 20000\\1000\le x_{2,3}\le 20000\\10\le x_i\le 20000, i=4,5,6,7,8;\end{cases} \tag{5-6}
$$

实例 2：

$$
s.t.\begin{cases}x_{10}+x_{11}+x_{12}\le 300000\\x_{10}x_1+x_{11}x_4+x_{12}x_7\le 1600000\\x_{10}x_2+x_{11}x_5+x_{12}x_8\le 1200000\\x_{10}x_3+x_{11}x_6+x_{12}x_9\le 900000\\x_{10}\le 300000\\x_{11}\le 300000\\x_{12}\le 300000\\x_i\ge 0, i=1,2,\cdots,12;\end{cases} \tag{5-7}
$$

实例 3：

$$
s.t.\begin{cases}x_1+x_2-2\le 0\\x_1^2-x_2+2\le 0\\-x_1+x_2-2.1\le 0\\-100\le x_{1,2}\le 100\end{cases} \tag{5-8}
$$

实例 4：

$$s.t.\begin{cases} 2x_1+2x_2^3+x_{10}+x_{11}-10\le 0 \\ 2x_1+2x_3^3+x_{10}+x_{12}-10\le 0 \\ 2x_2+2x_3^3+x_{11}+x_{12}-10\le 0 \\ -8x_1^3+x_{10}\le 0 \\ -8x_2^3+x_{11}\le 0 \\ -8x_3^3+x_{12}\le 0 \\ -2x_4^3-x_5+x_{10}\le 0 \\ -2x_6^3-x_7+x_{11}\le 0 \\ -2x_8^3-x_9+x_{12}\le 0 \\ x_i\ge 0, i=1,2,\cdots,12; \\ x_i\le 1, i=1,2,\cdots,12; \end{cases} \tag{5-9}$$

对上述 4 个实例分别运行 1000 次，其结果见表 5-1。

表 5-1 实例计算的结果

Tab.5-1 Example of the calculation results

示例	运行时间/s			迭代次数		
	最短时间/s	最长时间/s	平均时间/s	最小次数	最大次数	平均次数
1	0.199042	1805.893	148.9989	3	37374	3045.408
2	0.123441	17.97541	1.313125	2	337	24.161
3	0.052421	299.4415	7.695412	2	9376	270.475
4	0.05334	10124.21	42.66289	1	215024	899.0973

由表 5-1 可以看出，由于 4 个实例可行域的大小不同，其求解所用时间是不一样的。但通过对 4 个实例共计 4000 次的计算，每次都能求得初始内点，因此验证了本文给出的求解约束优化问题初始内点的方法是行之有效的。

5.4 本章小结

通过本章的示例验证可知，采用实数遗传算法来求解不同问题的初始内点虽然所用时间是不一样的，但每次都能求得初始内点，因此表明本文所提出的采用实数遗传算法来求解有约束优化问题的初始内点是行之有效的。

6 结论

（1）本文经研究分析给出了求解约束优化问题初始内点的一种方法——实数遗传算法。

（2）研究给出了基于实数遗传算法来求解有约束优化问题初始内点时，适应度函数的构造方法与模型。

（3）对实数遗传算法的各种操作方法进行了系统的阐述。

（4）本文给出的进化策略具有如下特点：1)在交叉操作时将交叉概率取 1，以保证所有的父代配对个体都需要进行交叉，即：选择这样的交叉概率能够增加产生子代个体的数量，能够增大产生更加优秀个体的可能性，使算法的运算速度能够得到提高。2)将交叉产生的个体与父代中精英保留的 m 个个体放在一起排序，重新选出精英保留的 m 个个体。这样可以避免交叉操作产生的优秀个体在变异操作过程中遭到破坏的不足，使得交叉操作产生的优秀个体能够得到生存。3)对变异操作后的个体排序，然后用精英保留的 m 个优秀个体对变异操作后最差的 m 个个体进行替换，形成新的种群。这样既有效的保证了种群的多样性，同时又使得求得的最优结果不会比前一次求得的结果差。

（5）经实例计算表明，该方法是一种行之有效的方法。由于该方法不用对约束条件进行求导，因此对于约束条件不可导的问题也能适用，较传统方法具有更广泛的实用性。

致 谢

一晃间，三年已去，将要离去这个熟悉的地方，想一想喜忧参半，高兴的是通过不断地学习，我即将毕业，担忧的是又要开始新的工作和征程，对未来的一切还不确定，害怕离开这个祥和而快乐的地方。

这三年间，认识了不少新同学、新朋友，尤其幸运的是遇到了我最好的老师——王福林教授，有他作为我的指导老师，真的受益匪浅。虽然老师初始时看起来不苟言笑，但是，通过不断的了解老师的学识和为人，老师内心的善良和外在的可爱给我留下了深刻的影响，能够跟着老师学习，觉得已无憾于校园生活了，谢谢老师三年的操心与照顾。当然这三年来还有其他有趣而快乐的事，就不一一赘述了。

三年来，父母对我的学习给予了无条件的支持，想一想父母逐年变老，心里略是一震，我已成年已久，父母还把我当成是没有长大的孩子一样疼爱，我却没有半点回报给父母，我想在这里说声谢谢，谢谢亲爱的老爸老妈。

当然在读研的期间我自己通过的生活过程，掌握了很多生活技巧和常识，我也能够更好地照顾自己了，感谢生活给了我许多，感谢生活让我更深的认识了这个世界。最后谢谢那些帮助过我但我却不知道的人和事，谢谢你们。

参考文献

[1] 陈皓，崔社武，严太山，等. 基于竞争指数的模拟退火排序选择算子[J]. 电子学报，2009，37(3)：586-591.

[2] 梁昔明，朱灿，颜东煌. 基于物种选择的遗传算法求解约束非线性规划问题[J]. 中南大学学报自然科学版，2009，40(1)：185-189.

[3] 赵改善. 求解非线性最优化问题的遗传算法[J]. 地球物理学进展，1992，7(1)：92-99.

[4] 蒋平，梁乐. 基于内点法和遗传算法相结合的交直流系统无功优化[J]. 高电压技术，2015，41(3)：724-729.

[5] 马永杰，云文霞. 遗传算法研究进展[J]. 计算机应用研究，2012，29(4)：1201-1206.

[6] 詹士昌. 基于退火不可行度的约束优化问题遗传算法[J]. 应用基础与工程科学学报，2004，12(3)：299-304.

[7] 孙有发，高京广，张成科，等. 反馈混沌遗传算法及其在约束优化中的应用[J]. 华南理工大学学报，2007，35(1)：19-23.

[8] 梁昔明，秦浩宇，龙文. 一种求解约束优化问题的遗传算法[J]. 计算机工程，2010，36(14)：147-149.

[9] 徐斌，李乃乾. 基于遗传-蚁群算法的多 QoS 约束组播路由优化算法[J]. 电子设计工程，2011，19(6)：23-26.

[10] 刘大莲，徐尚文. 求解约束优化问题的内外交叉遗传算法[J]. 系统工程理论与实践，2012，32(1)：189-195.

[11] 刘大莲，张春花，杜金玲. 基于种群分类排序的约束优化遗传算法[J]. 数学的实践与认识，2012，42(8)：190-196.

[12] 鲁延京，陈英武，杨志伟. 求解约束优化问题的粒子进化变异遗传算法[J]. 控制与决策，2012，27(10)：1441-1446.

[13] 王福林，王吉权，吴昌友，等. 一种新的改进实数遗传算法[C]. 中国农业工程学会学术年会，2005.

[14] 金芬. 遗传算法在函数优化中的应用研究[D]. 苏州大学，2008.

[15] Deb K. An Efficient Constraint Handling Method for Genetic Algorithm[J]. Computer Methods in Applied Mechanics and Engieering，2000，186(2-4)：311-338.

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

[17] Takahama T，Sakai S. Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations[J]. IEEE Transactions on Evolutionary Computation，2005，9(5)：437-451.

[18] Chootinan P，Chen A. Constraint handling in genetic algorithms using a gradient-based repair method[J]. Computers & Operations Research，2006，33(8)：2263-2281.

[19] Kumar S，Naresh R. Efficient real coded genetic algorithm to solve the non-convex hydrothermal scheduling problem[J]. International Journal of Electrical Power & Energy

Systems，2007，29(10)：738-747.

[20] Tutkun N. Optimization of multimodal continuous functions using a new crossover for the real-coded genetic algorithms[J]. Expert Systems with Applications，2009，36(4)：8172-8177.

[21] Abdul-Rahman O A，Munetomo M，Akama K. An adaptive parameter binary-real coded genetic algorithm for constraint optimization problems：Performance analysis and estimation of optimal control parameters[J]. Information Sciences，2013，233(2)：54-86.

[22] 胡运权. 运筹学教程[M]. 清华大学出版社，2012.

[23] 郭鹏. 运筹学教程[M]. 兵器工业出版社，2005.

[24] 莫尔斯.P.M，金博尔.G.E，莫尔斯.P.M，金博尔.G.E，等. 运筹学方法[M]. 北京科学出版社，1988.

[25] Holland J H. Adaptation in natural and artificial systems[M]. MIT Press，1975.

[26] 张文修，梁怡. 遗传算法的数学基础[J]. 西安交通大学学报，2000，34(12)：6.

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

[28] 张文修，梁怡. 遗传算法的数学基础[J]. 西安交通大学学报，2000，34(12)：6.

[29] 夏传甲，徐辉，万晓云，等. 基于实数编码遗传算法的 GPS 短基线整周模糊度搜索[J]. 大地测量与地球动力学，2012，32(1)：136-140.

[30] 谭永基，曾毅. 用混合遗传算法求解约束 NLP 问题[J]. 复旦学报，2000，39(5)：467-471.

[31] 朱会霞，王福林，张勇，等. 改进遗传算法优化非线性规划问题[J]. 数学的实践与认识，2013，43(7). 117-125.

[32] 马卫，朱庆保. 求解函数优化问题的快速连续蚁群算法[J]. 电子学报，2008，36(11)：2120-2124.

[33] 申晓宁，郭毓，陈庆伟，等. 一种保持群体多样性的多目标遗传算法[J]. 控制与决策，2008，23(12)：1435-1440.

[34] 袁满，刘耀林. 基于多智能体遗传算法的土地利用优化配置[J]. 农业工程学报，2014，30(1)：191-199.

[35] 管小艳. 实数编码下遗传算法的改进及其应用[D]. 重庆大学，2012.

[36] 吴晓涛，孙增圻. 用遗传算法进行路径规划[J]. 清华大学学报，1995(5)：14-19.

[37] 夏传甲，徐辉，万晓云，等. 基于实数编码遗传算法的 GPS 短基线整周模糊度搜索[J]. 大地测量与地球动力学，2012，32(1)：136-140.

[38] 庄健，杨清宇，杜海峰，等. 一种高效的复杂系统遗传算法[J]. 软件学报，2010，21(11)：2790-2801.

[39] 张瑜，娄卉芳，文良浩，等. 一种改进的遗传算法交叉策略[J]. 湖南科技大学学报自然科学版，2012，27(1)：94-97.

[40] 江中央，蔡自兴，王勇. 求解全局优化问题的混合自适应正交遗传算法[J]. 软件学报，2010，21(6)：1296-1307.

[41] 范青武，王普，张会清，等. 遗传算法交叉算子的实质分析[J]. 北京工业大学学报，2010，36(10)：1328-1336.

[42] 李继哲. 使用约束均匀交叉操作的遗传算法及其应用[D]. 哈尔滨工业大学，2002.

[43] 谢燕丽，许青林，姜文超. 一种基于交叉和变异算子改进的遗传算法研究[J]. 计算机技术与发展，2014(4)：80-83.

[44] 贺永兴，杨瑞，唐伟，等. 基于重构变异算子遗传算法的研究[J]. 计算机技术与发展，2015，25(12)：101-104.

[45] 王康泰，王宁. 信息熵动态变异概率 RNA 遗传算法[J]. 控制理论与应用，2012，29(8)：49-55.

[46] 喻寿益，邝溯琼. 保留精英遗传算法收敛性和收敛速度的鞅方法分析[J]. 控制理论与应用，2010，27(7)：843-848.

[47] 袁煜明，范文慧，杨雨田，等. 一种基于多样化成长策略的遗传算法[J]. 控制与决策，2009，24(12)：1801-1804.

[48] 慕彩红，焦李成，刘逸. 求解约束优化问题 M-精英协同进化算法[J]. 西安电子科技大学学报自然科学版，2010，37(5)：852-861.

[49] 祁荣宾，钱锋，杜文莉，等. 基于精英选择和个体迁移的多目标遗传算法[J]. 控制与决策，2007，22(2)：164-168.

[50] 王利琴，董永峰，顾军华. 改进的精英遗传算法及其在特征选择中的应用[J]. 计算机工程与设计，2014，35(5)：1792-1796.

[51] 谭冠政，刘良敏. 基于复数编码遗传算法的竞争性协进化策略[J]. 中南大学学报，2005，36(3)：475-480.

[52] 洪玮，崔杜武. 函数优化的遗传算法策略优选[J]. 计算机工程与设计，2010，31(13)：3043-3046.

[53] 朱灿. 实数编码遗传算法机理分析及算法改进研究[D]. 机电一体化，2014(6)：31-35.

[54] 吴天羿，许继恒，刘建永，等. 求解有硬时间窗车辆路径问题的改进遗传算法[J]. 系统工程与电子技术，2014，36(4)：708-713.

[55] 巩敦卫，郝国生，严玉若. 交互式遗传算法基于用户认知不确定性的定向变异[J]. 控制与决策，2010，25(1)：74-78.

[56] 明亮，王宇平. n 进制编码遗传算法的收敛速度[J]. 系统工程理论与实践，2006，26(3)：88-93.

[57] 常红伟，马华，张介秋，等. 基于加权实数编码遗传算法的超材料优化设计[J]. 物理学报，2014，63(8)：436-442.

[58] 蒋平，梁乐. 基于内点法和遗传算法相结合的交直流系统无功优化[J]. 高电压技术，2015，41(3)：724-729.

[59] 曹顺锋，焦永昌，张铮. 多种群首领决策遗传算法优化阵列天线[J]. 电波科学学报，2015，30(6) ：1137-1143.

[60] 刘振，彭军，刘勇. 小生境分布估计量子遗传算法及其仿真分析[J]. 计算机工程与科学，2016，38(1)：89-94.

[61] 张玲，刘勇，何伟. 自适应遗传算法在车牌定位中的应用[J]. 计算机应用，2008，28(1)：184-186.

[62] 王兴元，侯宪锋，董文昭，等. 基于遗传算法的城市环境监测点布局分析[J]. 科技创新

与应用，2014(33)：16-16.
[63] 彭东海. 一种锐化解空间的基于模拟退火的混合遗传算法[J]. 电力科学与技术学报，2005，20(2)：58-61.
[64] 周育人，李元香，王勇. 一种有效的实数编码遗传算法[J]. 武汉大学学报，2003，49(1)：39-43.
[65] 李炯城，王阳洋，李桂愉，等. 快速收敛的混合遗传算法[J]. 计算机工程与设计，2014，35(2)：686-689.

攻读硕士学位期间发表的学术论文

[1] 方堃，王福林，张力，李飞. 基于实数遗传算法的有约束优化问题初始内点求解方法的研究[J]. 东北农业大学学报，已录取.

[2] 董志贵，王福林，吴志辉，方堃. 基于 BP 神经网络的玉米种植密度和施肥量优化[J]. 农业工程学报，2017，33(6)：92-99.