

分类号 _____
UDC _____

学 号 _____ GS0504038

工程硕士学位论文

电磁频谱管理中的频率分配技术研究

硕士生姓名	古邦伦
学 科 领 域	电子与通信工程
研 究 方 向	通信信号处理与通信网络
指 导 教 师	魏急波 教授

国防科学技术大学研究生院

二〇〇六年五月

The Research of Frequency Assignment in Frequency Spectrum Management System

Candidate: Gu banglun

Advisor: Prof. Weijibo

A thesis

**Submitted in partial fulfillment of the requirements
for the professional degree of Master of Engineering
in Electron and Communication Engineering
Graduate School of National University of Defense Technology
Changsha, Hunan, P.R.China**

May, 2006

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知,除了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和撰写过的研究成果,也不包含为获得国防科学技术大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目: 电磁频谱管理中的频率分配技术研究

学位论文作者签名: 古邦伦

日期: 2006年5月18日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档,允许论文被查阅和借阅;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。

(保密学位论文在解密后适用本授权书。)

学位论文题目: 电磁频谱管理中的频率分配技术研究

学位论文作者签名: 古邦伦

日期: 2006年5月18日

作者指导教师签名: 魏彦俊

日期: 2006年5月18日

目 录

摘 要.....	i
ABSTRACT	ii
第一章 绪论	1
1.1 课题研究的背景及意义	1
1.2 研究现状	1
1.3 本文研究的内容	2
第二章 频率分配方法及约束条件分析	3
2.1 频率复用	3
2.2 频率分配方法	3
2.2.1 固定信道分配	3
2.2.2 动态信道分配	4
2.2.3 混合信道分配	5
2.3 干扰及约束条件	5
2.3.1 同频干扰	5
2.3.2 邻频约束条件	6
2.3.3 同小区频率间隔	7
2.3.4 互调成分约束条件	7
2.3.5 噪声及其响应	7
2.4 利用干扰信息进行频率分配及数学模型	7
2.4.1 理想情况下的数学模型	7
2.4.2 实际应用干扰系数矩阵的数学模型	10
第三章 遗传算法及其在频率分配中的应用	14
3.1 遗传算法的基本思想和步骤	14
3.1.1 遗传算法的基本思想和步骤	14
3.1.2 遗传算法基本理论	15
3.2 遗传算法关键参数和操作设计	18
3.2.1 编码方法	18
3.2.2 适应度函数	20
3.2.3 选择操作	22
3.2.4 交叉操作	24
3.2.5 变异操作	25

3.2.6 遗传算法控制参数选取	26
3.2.7 约束条件的处理	26
3.3 遗传算法在频率分配中的应用	27
3.4 遗传算法性能	33
第四章 模拟退火算法在频率分配中的应用	35
4.1 模拟退火算法的基本思想和步骤	35
4.2 模拟退火算法关键参数和操作设计	37
4.3 模拟退火算法在频率分配中的应用	39
4.4 模拟退火与遗传算法结合应用	41
4.4.1 GASA算法及步骤	41
4.4.2 性能比较	43
第五章 禁忌搜索算法	46
5.1 禁忌搜索算法的基本思想和步骤	46
5.2 禁忌搜索算法关键参数和操作设计	47
5.3 禁忌搜索算法在频率分配中的应用	50
第六章 结束语	53
致 谢	54
参考文献	55

图 目 录

图 2.1	两个基站间的同频干扰.....	6
图 3.1	基本遗传算法流程图.....	15
图 3.2	适应度函数尺度变换(正常情况).....	21
图 3.3	适 应 度 函 数尺度变换(出现负值).....	22
图 3.4	遗传算法仿真图.....	34
图 3.5	改变参数后遗传算法仿真图.....	34
图 4.1	模拟退火算法流程图.....	36
图 4.2	GASA混合策略流程图.....	43
图 4.3	两种算法对比图.....	44
图 4.4	遗传算法与遗传退火算法仿真比较图.....	45
图 5.1	禁忌搜索算法流程图.....	47

摘 要

随着无线通信技术的快速发展，如何采用优化的频率分配技术提高频谱资源的使用效率是频率管理领域研究的热点问题。在战场电磁环境相当复杂的军用无线通信中，如果缺乏有效的频谱管理，不同的电子设备间轻则相互干扰，重则丧失作战能力。为确保部队正常指挥、打赢现代战争就必须保证无线电指挥系统的正常运行。因此加强频谱管理、合理利用频率资源显得日趋重要。

本文对电磁频谱管理中的频率分配技术进行了研究，论文所做的主要工作有：

- 1、研究了频率分配的限制性因素并建立以干扰信息进行频率分配的数学模型；
- 2、研究了遗传算法的基本理论及其在频率分配中的应用，进行了算法的设计实现与仿真；
- 3、研究了模拟退火算法及其操作设计，并将算法应用于频率分配中；
- 4、研究了遗传算法和模拟退火算法的结合应用，对遗传算法和模拟退火算法以及算法结合的性能进行了比较；
- 5、研究了禁忌搜索算法及其在频率分配中的应用。

本文以GSM为例，对网络中的干扰进行分析研究，建立了网络中以干扰信息进行频率分配的数学模型，并且通过对遗传算法、模拟退火算法和禁忌搜索算法的研究，结合频率分配的特点，用这些优化算法解决频率分配问题，对移动通信网络建设和部队无线电通信频率管理具有一定的指导意义。

关键词：频率分配、遗传算法、模拟退火、禁忌搜索。

ABSTRACT

Along with the rapid development of the wireless communication. How to adopt optimized frequency assignment technology to improve the spectrum efficiency has been a main problem in the battlefield spectrum management. In the military wireless communication, the electromagnetic environment had become very complex on the modern battlefield. The inefficient frequency planning maybe results massive interference or losing capability of operation. To guarantee the command of the army in gear, and win the modern war, we must ensure the wireless chain of command to run in gear. Therefore, spectrum management is critical for modern information war.

In this paper, several frequency assignment schemes are studied and analyzed, the main work of the paper includes:

1.The limitative factors on frequency assignment are studied and the mathematics models for frequency assignment are created. The calculation results of interference analysis are introduced the procedure of frequency assignment.

2.The genetic algorithms is introduced to realize the optimization of frequency allocation. The performance of GA algorithms is also analyzed in this paper.

3.The simulated anneal algorithm and it's manipulation and design are studied, it is applied in frequency assignment.

4. The genetic algorithm and the simulated anneal algorithm are combined to apply in frequency assignment. The performances of GA, SA, GASA are compared.

5.The taboos search algorithm and its application in frequency allocation are also studied.

This paper is to take GSM as an example, the interference in the network is analyzed and studied, the mathematics model for frequency allocating with using the interference information in the network is created, and through studying the genetic algorithms, simulated anneal algorithm research and taboos search algorithm, combining the characteristics of the frequency allocation, and use of these optimization algorithms to resolve the frequency allocation problem. It has guidance meaning to the mobile communication network construction and the

management of radio frequency communications in the army.

Key Words: Frequency Assignment, Genetic Algorithms, Simulated Annealing, Taboo Search.

第一章 绪论

1.1 课题研究的背景及意义

随着科学技术的进步和人们生活水平的提高,移动通信正在逐步走进千家万户,无线用户数正随着移动通信的迅速发展和普及以惊人的速度增长。高速增长无线用户数与有限的频率资源这对矛盾变得更加突出。人们不得不将分配给无线通信的有限频率资源进行有效和合理的复用。另外从降低服务成本的角度看,对无线频率资源的有效利用也是很重要的,它能够使基站数减少,降低系统的投资。因此如何提高频率利用率就成为当前移动通信发展的关键所在,而采用合理的信道分配技术就能够有效解决频率利用问题。

同样,随着现代通信技术的发展,无线电通信设备在军队的应用也日益广泛。在现代战争中,为保证部队能有效获取作战信息、保证部队的合成指挥、保证部队之间的协同作战,大量无线电子设备应用于现代战场,使得现代战场电磁环境相当复杂。在信息化战争中,军事通信、指挥控制、预警探测、情报侦察、导航定位和武器制导等系统的正常运转,无不取决于电磁频谱这一有限的资源。如果缺乏有效的频谱管理,不同的电子设备间轻则相互干扰,重则丧失作战能力。为确保部队正常指挥、打赢现代战争就必须保证无线电指挥系统的正常运行。合理使用频率资源,加强频率管理就显得日趋重要。这就提出了战场频谱管理的课题。因此利用频率分配技术,合理为战场环境下各电子设备分配相应的频率,减少无线通信设备之间的相互干扰,确保无线电通信畅通无阻,是保障现代战争指挥顺畅的重要手段,也是打赢现代战争的重要条件。

无线电频谱作为宝贵的自然资源,是无限和有限的统一。频谱资源的上限没有极限,它随着人类科学的发展而不断延伸,用之不竭,在使用中没有任何消耗;同时,受现有的科学发展和认识水平的局限,它在特定的一个时间和空间内,可供使用的范围又是有限的。无线电通信的重要性和无线电频谱资源有限性的矛盾,已成为当今世界各国都普遍面临的一个尖锐问题。因此,加强对频谱规划的研究,提高频谱利用率,就成为无线电通信管理部门一项十分重要和紧迫的任务。

1.2 研究现状

频率分配是优化频谱利用、提高信道容量、减少干扰的主要手段。

目前用于频率分配的基本算法有:启发式算法、找最大独立团法、染色法、爬山当、神经网络法、模拟退火算法和遗传算法等。在应用这些算法进行研究时有以下一些频率优化分配的研究:

- 采用满足频点间隔结合遗传算法的频率优化分配算法;
- 免疫网络调节算法;

- 免疫遗传算法;
- 遗传退火算法;
- 图的边着色等等。

基于这些算法的研究,国内外各移动通信系统制造商和通信软件公司都开发出自己的频率规划商用软件。这些商用软件都具有自己的频率规划算法,这些算法的目的都是使网络干扰减到最小。

1.3 本文研究的内容

本文研究的主要内容有:

1. 通过对频率分配限制性因素的研究,建立了以干扰计算为基础的频率分配的数学模型,将频率分配问题归结为通信网络的干扰评价函数的优化问题
2. 遗传算法及其在频率分配中的应用,对遗传算法的理论及操作设计进行了研究并将遗传算法应用于通信网络的频率分配的工程实践中。
3. 模拟退火算法及其在频率分配中的应用,对模拟退火算法的基本理论进行了研究,将模拟退火算法与遗传算法相结合应用于频率分配中。
5. 禁忌搜索算法及其在频率分配中的应用,对禁忌搜索算法的基础理论进行研究,并提出将禁忌搜索法应用于频率分配工程中的具体操作。
6. 对算法的性能进行比较,对遗传算法不同参数选取对算法性能影响进行了比较,对遗传算法及模拟退火算法以及遗传算法和模拟退火算法结合的性能进行比较。

第二章 频率分配方法及约束条件分析

随着移动通信飞速发展，移动电话的普及，无线用户数以惊人的速度增长。高速增长的无线用户与有限的频率资源这对矛盾变得更加突出。人们不得不将分配给无线通信的有限频率资源进行有效和合理的复用。

2.1 频率复用

蜂窝是解决频率不足和用户容量问题的一个重大突破。它能在有限的频谱上提供非常大的容量，而不需要做技术上的重大修改。这个概念是一种系统级的概念，其思想是许多小功率的发射机来代替单个的大功率发射机，每一个小覆盖区只提供服务范围内的一小部分覆盖。每个基站分配整个系统可用频率资源的一部分，相邻基站则分配了频率资源的其他部分，这样所有的可用频率就分配给了所有的基站。给相邻的基站分配不同的频率，使得基站之间的干扰最小。通过系统地给整个系统中的基站分配频率，整个频率资源就要以在整个系统的地理区域内得到分配，而且尽可能地复用，只要基站间的同频干扰低于门限值。

蜂窝无线系统依赖于整个覆盖区域内频率的智能分配和复用。每一个基站都分配一组信道，占用一小部分频率资源，用于一个小区。给相邻的小区的基站分配一个信道组，使该组中的信道不能用在相邻的小区中以避免同频干扰。基站天线设计要做到能获得某一特定小区内期望的覆盖。通过将覆盖范围限制在小区边界以内，相同的频率就可以用于不同的小区，只要这些小区两两之间的距离足够远，以使得相互之间的干扰水平在可接受的界限之内即可。

在提高频率资源利用率的诸多方法中，最被广泛研究和利用的是以频率复用为基础的频率规划技术。频率复用就是将同一无线频率用作不同小区（或扇区）的载频，这些使用相同的载频的小区彼此相隔足够的距离来把同频（或邻频）干扰抑制到允许的范围以内的方法。

2.2 频率分配方法

2.2.1 固定信道分配

在固定信道分配（FCA）方案中，服务区域被分为许多小区，信道根据一定的复用模式分配给每个小区，这些信道被称为标称信道。只要小区之间的距离满足复用距离，同一组信道就可以在其他小区使用。对于简单 FCA 策略，每个小区被分配给相同数目的标称信道。当系统的业务也是平均分布时，均匀信道分配是有效的。而且在高话务量的情况下性

能也是非常好的。但移动系统中话务量分布多数情况下是不均匀的,随空间和时间而变化,因此对每个小区也是非均匀分配信道,以便使每个小区所拥有的信道数与所承担的话务量相匹配,这就是非平均信道分配策略。在这种策略中,分配给每个小区的标称信道数取决于该小区的统计话务量。因此高承载小区比低承载小区要分配更多的信道,从而改善信道利用率。虽然在处理突发话务量方面,非均匀分配比均匀分配有相对较好的性能,但由于小区中任何呼叫只能使用该小区中的空闲信道。如果该小区中所有信道都已被占用,则呼叫阻塞,故对信道的利用并没有达到一个最佳的水平。为了解决这个问题,固定分配方案有许多变种形式,其中借用信道分配方案就是一种,即如果该小区的所有信道都已被占用,将允许小区从其相邻小区中借用。这样的借用过程由 MSC(移动交换中心)来管理,并且保证借用信道不会中断借出小区的任何正在进行的呼叫。信道借用的基础是它只对相邻小区产生最小的影响。当某信道被借用时,则该信道在其他小区将禁止使用,称之为“信道锁定”。由于信道借用将导致禁止使用该信道的小区数取决于所采用的信道复用因子、小区结构类型以及信道初始分配类型。信道借用是临时性的,仅仅持续在呼叫通话过程中。一旦呼叫结束,被借用的信道将返回到它原来的小区,并且将锁定信道释放。对中低等话务量,信道借用策略(CBS)的性能比 FCA 要好一些,但对高话务量,FCA 的性能要优于 CBS,因为当话务量很低时,借用的信道数比较少,它们能够处理所承担的话务量波动;但是当承担的话务量很高时,借用的信道数将迅速增大,当激增到一定程度将导致信道利用率急剧下降,呼叫阻塞率上升,这主要是由信道锁定所引起的。

信道借用策略只需要本地或者相邻小区的相关信息,分为简单信道借用策略(Simple CBS)和混合信道借用策略(Hybrid CBS)两类,主要取决于从相邻小区选取空闲信道的方式。在简单信道借用策略中,只要信道空闲且借用不会导致对相邻小区正在进行的呼叫产生干扰,小区中的任何标称信道都可以随时加以借用。简单信道借用策略主要包括:富有小区借用、基本算法、再分配基本算法、优先可用借用。

混合信道借用策略中,每个小区的标称信道被分成两个小组:A 组(包含只能在本小区使用的标称信道)和 B 组(包含能供给相邻小区使用的信道)。对于一个固定的 A/B 比率,复用因子越大,可能借用信道的小区数增加,而每个小区可供借用的信道数将越少。不同混合信道借用方案主要包括:简单混合信道借用策略、具有信道排序的借用、具有方向性信道锁定借用、具有借用与再分配的信道分配、重新调整的信道排序分配策略。

2.2.2 动态信道分配

由于蜂窝系统中话务量在时间和空间上的短期变化,FCA 策略不能获得很高的信道利用率。可采用动态信道分配策略来改善信道利用率。动态信道分配借助动态方式分配信道,来克服固定信道分配对话务量的时间和空间变化缺乏自适应性的缺点。在动态信道分配方

案（DCA）中，话音信道不是固定地分配给某个小区，而是放在一个中央信道库中（MSC），每次产生呼叫请求时，为它服务的基站就向 MSC 请求一个信道。交换机会根据一种特定的算法给发出请求的小区分配一个信道。这种算法会考虑该小区中未来阻塞的可能性，候选信道的使用频次，信道的复用距离和其他代价函数等因素。当呼叫结束后，所使用的信道将返还给信道库。实际中，DCA 根据其拥有的特点被分为以下三种主要类型：

- 干扰自适应的 DCA；
- 话务变化自适应的 DCA；
- 信道复用的 DCA。

这两种分配方案各有自己的优缺点，FCA 对话务量在时间和空间变化缺乏自适应性。当话务量在时间和空间上发生短期变化时，FCA 不能获得很高的信道利用率。DCA 虽然克服了 FCA 的这个缺点，但当系统出现很高的话务量时，FCA 的性能反而要优于 DCA。

2.2.3 混合信道分配

混合信道分配方案（HCA）是将 FCA 和 DCA 混合使用的分配技术。

在 HCA 中，能用来服务的所有有用信道被分为固定和动态的两组。固定信道组包含与 FCA 策略中一样分配给小区的一组标称信道，在任何情况下，该组信道只能在所分配的小区使用。另外一组信道可以供系统中所有用户共享，这样能够提高系统的灵活度。当某小区有呼叫请求服务但此时该小区所有标称信道都忙时，则从动态信道组中选择一个信道分配给该呼叫。从动态信道组中分配信道的过程可以遵循上节所述的任意一种 DCA 策略。固定信道与动态信道之比是一个很重要的参数，它将决定整个蜂窝系统的性能。为了达到最好的性能，我们希望能找到最佳的固定动态信道比。HCA 通过提高信道的占有率使系统性能得到了很大提高，但是在太系统中进行信道重调整需要非常大的计算量。

在本文中我们将只针对固定信道分配（FCA）进行遗传算法、模拟退火算法和禁忌搜索法的研究。

2.3 干扰及约束条件

2.3.1 同频干扰

在一个基站的服务小区内，如果来自其他基站的发射信号足够强且两个基站使用相同的频率，则在该服务小区内就会产生干扰称为同频干扰。在蜂窝系统中，同频干扰是主要的噪声来源。

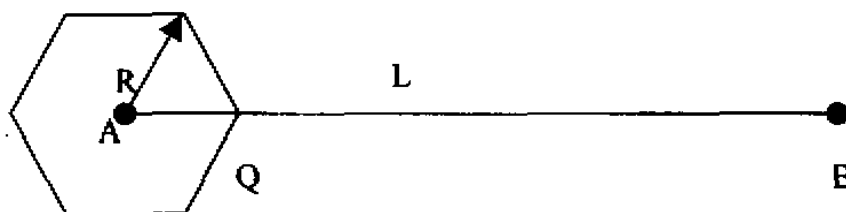


图 2.1 两个基站间的同频干扰

设基站 A 的小区服务半径为 R ，基站 A 和 B 之间的距离为 L ，基站 A 和 B 采用相同的频率。则 A 的服务范围内的移动台可能受到 B 的干扰，当载干比 $C/I \geq 18\text{db}$ 时，认为 B 对 A 产生的干扰不足以对 A 小区内的移动台的服务质量产生影响，即认为 B 对 A 不产生同频干扰。反之，若 $C/I < 18\text{db}$ ，则认为 B 对 A 产生同频干扰。

对移动通信无线信道的传播测量表明，在任一点接收到的平均信号能量随发射机和接收机间的距离的幂指数下降。在距离发射天线 r 处接收到的平均信号功率 $P(r)$ 可以由下式估算：

$$P(r) = P_d - 10n \log(r/d) \quad 2-1$$

其中， r 是基站与被服务点之间的距离， d 是基站附近一个小的距离， P_d 是在该点接收到的信号功率。这样在 Q 点有：

$$C = P_d - 10n \log\left(\frac{R}{d}\right) \quad \text{单位为 db;}$$

$$I = P_d - 10n \log\left(\frac{L-R}{d}\right) \quad \text{单位为 db;}$$

$$\begin{aligned} \text{则有载干比} \quad C/I(\text{db}) &= P_d - 10n \log\left(\frac{R}{d}\right) - [P_d - 10n \log\left(\frac{L-R}{d}\right)] \\ &= 10n \log\left(\frac{L-R}{R}\right) \end{aligned} \quad 2-2$$

若 B 对 A 有干扰，则下式成立：

$$C/I = 10n \log\left(\frac{L-R}{R}\right) < 18\text{db}$$

可以依此判断网络中基站 i 对基站 j 的干扰情况，得到干扰矩阵。

2.3.2 邻频约束条件

当发射机与接收机工作在相近的频率时（这种相近通常指 3 个频点以内），仍然存在干扰的可能性。因此对某 m 值，由下面不等式给出若干个约束条件

$$|f_i - f_j| > m$$

式中 m 为所需的频率间隔数。

2.3.3 同小区频率间隔

对于大部分情况，分配给同一小区发射机的任何一对频率都应有一定的间隔。比如 250kHz 或 5 个频点的间隔。这个间隔通常远大于相邻频点的约束条件。如果这个频率用于大功率发射机则频率间隔要更大，比如 500kHz 或 10 个频点。因此，约束条件可以表示为 $|f_i - f_j| \geq m$ ，其中 m 代表发射机 i 与 j 之间的所需的频率间隔数。

2.3.4 互调成分约束条件

当两个或多个信号在非线性系统中时，会产生新的频率分量。当这个频率分量太靠近有用频率时，就很难滤掉这个频率分量，从而产生干扰。相应的约束条件为

$2f_i - f_j \neq f$	两个信号，三阶互调
$3f_i - 2f_j \neq f$	两个信号，五阶互调
$f_i + f_j - f_k \neq f$	三个信号，三阶互调
$2f_i + f_j - 2f_k \neq f$	三个信号，五阶互调

2.3.5 噪声及其响应

无线收发信机都会有特征噪声与响应，这在同一基站或同小区内也是要极力避免的。相应的频率关系式为

$$f_i \neq mf_j - nf_{LO} + f_{spur} \quad 2-3$$

式中 f_{LO} 是本振频率， m, n 是整数， f_{spur} 是频率偏移常量，上术参量都由接收与发射机的器件性能所决定。

2.4 利用干扰信息进行频率分配及数学模型

频率复用的技术来提高了系统的容量，但不可避免地带来了不同小区间载波频率的同频或邻频干扰。下面以 GSM 网络为例进行分析。

2.4.1 理想情况下的数学模型

设某个 GSM 网络的某个区域 R 中有 n 个小区，在某个时间 t ，在第 i ($i = 1, 2, \dots, n$) 个小区的服务区域 (该区域构成的空间记为 Ω_i) 的某个点 p_i 接收到 (即采样到) 第 i 个小区的信号强度为 $s_{ii}(t, p_i)$ ，在同一时间 t 和同一点 p_i 接收到第 j 个小区的信号强度为 $s_{ij}(t, p_i)$ ，其中

$j=1, 2, \dots, n$, 并且 $j \neq i$ 。GSM规范规定载干比C/I(即某一载波的信号强度和与它同频的干扰信号的信号强度的差值, 这里假设信号强度用dBm表示)必须大于18dB, 因此, 若 $s_{ii}(t, p_i) - s_{ij}(t, p_i)$ 的值小于或等于18dB, 则可认为第i个小区存在受到第j个小区同频干扰的可能性, 这时, 若分配给第i个小区中的某个或某些载波的频率与第j个小区中的某个或某些载波的频率相同时, 则第i个小区将受到第j个小区的同频干扰。

设采样时间从 t_0 开始, 采样时长为 Δt , 变量 c'_{ij} 用来统计在 $[t_0, t_0 + \Delta t]$ 时间范围内以及第i个小区服务区域内 $s_{ii}(t, p_i) - s_{ij}(t, p_i) \leq CI_{th}$ 的次数, 式中 CI_{th} 为载干比的门限值, c'_{ij} 的值越大, 表示若第i个小区分配到与第j个小区中某个或某些载波相同的频率, 则第i小区将受到来自第j个小区越强烈的同频干扰。在 $[t_0, t_0 + \Delta t]$ 时间范围内, 区域R中所有小区之间变量 c'_{ij} 的值可构成下面的矩阵:

$$C' = \begin{bmatrix} c'_{11} & c'_{12} & \cdots & c'_{1n} \\ c'_{21} & c'_{22} & \cdots & c'_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c'_{n1} & c'_{n2} & \cdots & c'_{nn} \end{bmatrix} \quad 2-4$$

式中,

$$c'_{ij} = \begin{cases} 0, & i = j \\ |\{(s_{ii}(t, p_i), s_{ij}(t, p_i)) | (s_{ii}(t, p_i) - s_{ij}(t, p_i)) \leq CI_{th}\}|, & i \neq j \end{cases} \quad 2-5$$

式中 $p_i \in \Omega_i$, $i, j = 1, 2, \dots, n$, $|\{(s_{ii}(t, p_i), s_{ij}(t, p_i)) | (s_{ii}(t, p_i) - s_{ij}(t, p_i)) \leq CI_{th}\}|$ 表示

$\{(s_{ii}(t, p_i), s_{ij}(t, p_i)) | (s_{ii}(t, p_i) - s_{ij}(t, p_i)) \leq CI_{th}\}$ 中元素的个数, $t \in [t_0, t_0 + \Delta t]$ 。

式(2-4)反映了 $t \in [t_0, t_0 + \Delta t]$ 时间范围内区域R中各小区之间产生潜在同频干扰的强弱。由于可以通过式(2-4)计算出 $t \in [t_0, t_0 + \Delta t]$ 时间范围内区域R中各小区实际(对应于已运行小区)受到或将要(对应于等待配频的小区, 包括新建小区、载波扩容小区以及频率优化调整小区)受到的同频干扰, 因此将矩阵 C' 称为区域R的同频干扰系数矩阵, 同时将 c'_{ij} 称为同频干扰系数。然而, 式(2-4)和(2-5)在实际应用中是不可能的, 因为它假设在区域R各小区服务区域中的所有点均进行采样, 因此, 式(2-4)和(2-5)只是理想情况下的数学模型。

采用与前面理想模型中相同的假设, 并设在区域R的第i个小区的覆盖区域内的不同地方放置 e^i 个采样器(用来接收不同小区的信号强度), 这 e^i 个采样器可以在第i个小区的覆盖区域内自由移动, 每个采样器在 $t \in [t_0, t_0 + \Delta t]$ 时间范围内可以自行决定每次采样的开始时间和结束时间, 在 $t \in [t_0, t_0 + \Delta t]$ 时间范围内, 每个采样器可进行一次采样也可进行多次采

样。

在上述模型下，第*i*个小区受到第*j*个小区的同频干扰的系数为：

$$c_{ij} = \begin{cases} 0, & i = j \\ | \{ (s_{ii}(t, q_i), s_{ij}(t, q_i)) | (((s_{ii}(t, q_i) - s_{ij}(t, q_i)) \leq CI_{th}), & i \neq j \end{cases} \quad 2-6$$

式中， $i, j = 1, 2, \dots, n$ ， $t \in [t_0, t_0 + \Delta t]$ ， q_i 为第*i*个小区的第 q_i 个采样器，并有： $q_i = 1, 2, \dots, e_i$ 。对应的区域R的同频干扰系数矩阵为：

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \quad 2-7$$

上式中第*i*行构成的矩阵为：

$$c_i = [c_{i1}, c_{i2}, \dots, c_{in}] \quad 2-8$$

式中 $i = 1, 2, \dots, n$ ，并将 c_i 称为第*i*个小区的同频干扰系数矩阵。

采用同样的模型，可得第*i*个小区受到第*j*个小区的邻频干扰的系数

$$a_{ij} = \begin{cases} 0, & i = j \\ | \{ (s_{ii}(t, q_i), s_{ij}(t, q_i)) | (((s_{ii}(t, q_i) - s_{ij}(t, q_i)) \leq CA_{th}), & i \neq j \end{cases} \quad 2-9$$

式中， CA_{th} 为载邻比CA的门限值。载邻比是指某一载波的信号强度和与它邻频的干扰信号的信号强度的差值(假设信号强度用dBm表示)。

由 a_{ij} 构成的矩阵为：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad 2-10$$

将上式称为区域R的邻频干扰系数矩阵，式中第*i*行构成的矩阵为：

$$a_i = [a_{i1}, a_{i2}, \dots, a_{in}] \quad 2-11$$

其中 $i = 1, 2, \dots, n$ ，并将 a_i 称为第*i*个小区的邻频干扰系数矩阵。

这种模型中的采样器可由 $[t_0, t_0 + \Delta t]$ 时间范围内区域R中各个小区内建立了通话的用户手机来充当，用户手机在通话过程中同时也测量了服务小区及其相邻小区的信号强度，并将这些测量报告传至基站控制器。可以从GSM系统中获取这些测量报告，例如，爱立信GSM900系统中的CTR(Cell Traffic Recording)功能即可收集某个小区在指定的时间内所有建立通话的手机的测量报告并能以文件的形式输出，通过对CTR这类含有手机测量报告的文件进行处理即可生成某个小区的同频干扰系数矩阵和邻频干扰系数矩阵。

上述模型只是理想模型的一个近似，生成的干扰系数矩阵的准确度与小区的话务量、采样时长及通话用户的分布等有关，可在小区话务量比较高的时段选取适当长的统计时长对手机的测量报告进行采样收集。

2.4.2 实际应用干扰系数矩阵的数学模型

在对新基站或新小区进行频率分配时，由于它们开通运行前不可能从系统中采集这些小区的手机测量数据，因此无法从系统文件中生成相应小区的同频干扰系数矩阵和邻频干扰系数矩阵，这时可以手工构造这些新小区的同频干扰系数矩阵和邻频干扰系数矩阵。

在手工构造某新小区的同频干扰系数矩阵和邻频干扰系数矩阵时，可根据该小区与其相邻小区的地理位置关系、这些小区的覆盖方向以及它们所在区域的地形地物来给定该小区受到其各个相邻小区的同频和邻频干扰系数，某相邻小区对该小区可能构成的干扰越大。则分配给此相邻小区的同频和邻频干扰系数也越大。

根据手工构造的某新小区的同频干扰系数矩阵和邻频干扰系数矩阵对该小区进行频率分配后，即可用分配后的频率将该小区开通运行，若规划后的频率使得该小区的性能和质量均达到要求，则可保留这些频率，否则，若认为规划后的频率不够理想，则可以从系统中采集该小区中的手机测量数据，并根据这些数据生成的该小区的同频干扰系数矩阵和邻频干扰系数矩阵再进行一次频率分配。

设第*i*个小区所有载波频率构成的集合为：

$$V_i = \{f_1^{(i)}, f_2^{(i)}, \dots, f_{k_i}^{(i)}\} \quad 2-12$$

式中*i*=1, 2, ..., *n*, *k_i*为第*i*个小区的载波数。

设第*j*个小区所有载波频率构成的集合为：

$$V_j = \{f_1^{(j)}, f_2^{(j)}, \dots, f_{k_j}^{(j)}\} \quad 2-13$$

式中，*j*=1, 2, ..., *n*, *k_j*为第*j*个小区的载波数。

设在 $[t_0, t_0 + \Delta t]$ 时间范围内，第*i*个小区受到第*j*个小区的同频干扰为 c_{ij}^* ，则有：

$$c_{ij}^* = \sum_{l=1}^{k_j} (\gamma_{lj}^{(i)} c_{lj}) \quad 2-14$$

式中， c_{lj} 满足(2-6)式，并有：

$$\gamma_{lj}^{(i)} = \begin{cases} 0, & f_l^{(i)} \notin V_j \\ 1, & f_l^{(i)} \in V_j \end{cases} \quad 2-15$$

其中， $l = 1, 2, \dots, k_i$ 。由 c_{ij}^* 构成的矩阵为：

$$C^* = \begin{bmatrix} c_{11}^* & c_{12}^* & \cdots & c_{1n}^* \\ c_{21}^* & c_{22}^* & \cdots & c_{2n}^* \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1}^* & c_{n2}^* & \cdots & c_{nn}^* \end{bmatrix} \quad 2-16$$

上式称为区域R的同频干扰矩阵，其中第i行构成的矩阵为：

$$c_i^* = [c_{i1}^*, c_{i2}^*, \dots, c_{in}^*] \quad 2-17$$

式中， $i = 1, 2, \dots, n$ ，并将式 c_i^* 为第i个小区的同频干扰矩阵。

设在 $[t_0, t_0 + \Delta t]$ 时间范围内，第i个小区受到第j个小区的邻频干扰为 a_{ij}^* ，则有：

$$a_{ij}^* = \sum_{l=1}^{k_i} (\theta_{lj}^{(i)} a_{ij}) \quad 2-18$$

式中， a_{ij} 满足 (2-9) 式，并有：

$$\theta_{lj}^{(i)} = \begin{cases} 0, & (f_l^{(i)} + 1) \notin V_j \text{ or } (f_l^{(i)} - 1) \notin V_j \\ 1, & (f_l^{(i)} + 1) \in V_j \text{ or } (f_l^{(i)} - 1) \in V_j \end{cases} \quad 2-19$$

其中， $l = 1, 2, \dots, k_i$ 。由 a_{ij}^* 成构成的矩阵为：

$$A^* = \begin{bmatrix} a_{11}^* & a_{12}^* & \cdots & a_{1n}^* \\ a_{21}^* & a_{22}^* & \cdots & a_{2n}^* \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1}^* & a_{n2}^* & \cdots & a_{nn}^* \end{bmatrix} \quad 2-20$$

上式称为区域R的邻频干扰矩阵，其中第i行构成的矩阵为：

$$a_i^* = [a_{i1}^*, a_{i2}^*, \dots, a_{in}^*] \quad 2-21$$

式中 $i=1, 2, \dots, n$ ，并将 a_i^* 称为第i个小区的邻频干扰矩阵。

设第i个小区的所有载波频率构成的向量为：

$$f_i = (f_1^{(i)}, f_2^{(i)}, \dots, f_{k_i}^{(i)}) \quad 2-22$$

式中， $i=1, 2, \dots, n$ 。

由 (2-13) 和 (2-17) 可得在 $[t_0, t_0 + \Delta t]$ 时间范围内区域R中第i个小区受到来自R中其他小区的总的干扰(含同频和邻频)为：

$$g_i'(f_i) = \sum_{j=1}^n c_{ij}^* + \sum_{j=1}^n a_{ij}^* \quad 2-23$$

式中 $i=1, 2, \dots, n$ ， c_{ij}^* 和 a_{ij}^* 分别满足 (2-14) 和 (2-18) 式。

在实际中对小区所受总干扰进行评价时，对小区所受的同频干扰和邻频干扰考虑的侧重点程度会不同，因此，对 (2-23) 中的总同频干扰和总邻频干扰分别分配不同的权重，从而得到对第i个小区在 $[t_0, t_0 + \Delta t]$ 时间范围内所受干扰的评价值为：

$$g_i(f_i) = \alpha_i \sum_{j=1}^n c_{ij}^* + \beta_i \sum_{j=1}^n a_{ij}^* \quad 2-24$$

式中 $i = 1, 2, \dots, n$, α_i 和 β_i 分别为R中第i个小区的同频干扰权重和邻频干扰权重, 并有:

$$\alpha_i + \beta_i = 1 \quad 2-25$$

令:

$$r_i = \frac{\beta_i}{\alpha_i} \quad 2-26$$

称 r_i 为第i个小区的邻同干扰权重比, 由(2-25)和(2-26)得:

$$\alpha_i = \frac{1}{1+r_i} \quad 2-27$$

$$\beta_i = 1 - \frac{1}{1+r_i} \quad 2-28$$

将(2-27)和(2-28)式代入(2-24)式中得:

$$g_i(f_i) = \frac{1}{1+r_i} \sum_{j=1}^n c_{ij}^* + \left(1 - \frac{1}{1+r_i}\right) \sum_{j=1}^n a_{ij}^* \quad 2-29$$

对于整个区域R的干扰评价为:

$$g = \sum_{i=1}^n g_i(f_i) \quad 2-30$$

当 $r_1 = r_2 = \dots = r_n = r$ 时, 有:

$$g = \frac{1}{1+r} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^* + \left(1 - \frac{1}{1+r}\right) \sum_{i=1}^n \sum_{j=1}^n a_{ij}^* \quad 2-31$$

令:

$$g_{i1}(f_i) = \sum_{j=1}^n c_{ij}^* \quad 2-32$$

$$g_{i2}(f_i) = \sum_{j=1}^n a_{ij}^* \quad 2-33$$

将(2-32)和(2-33)式代入(2-29)式得:

$$g_i(f_i) = \frac{1}{1+r_i} g_{i1}(f_i) + \left(1 - \frac{1}{1+r_i}\right) g_{i2}(f_i) \quad 2-34$$

将(2-14)式代入(2-32)式得:

$$g_{i1}(f_i) = \sum_{j=1}^n \sum_{i=1}^{k_j} (\gamma_{ij}^{(i)} c_{ij}) \quad 2-35$$

式中 c_{ij} 满足 (2-6) 式, $\gamma_{ij}^{(i)}$ 满足 (2-15) 式。

将 (2-18) 式代入 (2-33) 式得:

$$g_{i2}(f_i) = \sum_{j=1}^n \sum_{l=1}^{k_j} (\theta_{ij}^{(i)} a_{ij}) \quad 2-36$$

式中 a_{ij} 满足 (2-9) 式, $\theta_{ij}^{(i)}$ 满足 (2-19) 式。

假设对区域R的第i个小区的所有载波的频率进行重新分配, 并将第i个小区的所有载波频率看作变量, 同时将 (2-29) 式和 (2-34) 看作第i个小区的干扰评价函数, 则对第i个小区的所有载波进行频率规划的目标即是使 (2-29) 式或 (2-34) 式的值达到最小, 从而将频率分配问题转化为干扰评价函数(在频率分配中, 只需考虑由等待分配的载波频率所构成的干扰评价函数)的优化问题。若不考虑频率分配中的约束条件, 则上述频率分配的数学模型为:

$$\min \frac{1}{1+r_i} \sum_{j=1}^n c_{ij}^* + (1 - \frac{1}{1+r_i}) \sum_{j=1}^n a_{ij}^* \quad 2-37$$

或者为:

$$\min \frac{1}{1+r_i} g_{i1}(f_i) + (1 - \frac{1}{1+r_i}) g_{i2}(f_i) \quad 2-38$$

若对整个区域R中所有小区中的所有载波的频率进行重新分配, 且不考虑频率分配中的约束条件, 则该频率分配的数学模型为:

$$\min \sum_{i=1}^n g_i(f_i) \quad 2-39$$

由于利用 GSM 网络中的干扰信息进行频率分配时, 可以将频率分配问题归结为某个小区或某个区域的干扰评价函数的优化问题, 而模拟退火算法、禁忌搜索算法、遗传算法是一种重要的智能优化算法, 它已被广泛应用于函数的优化问题以及工程中的各种优化问题中, 因此, 将这些算法应用于 GSM 网络的频率分配是可行的。依据同样的原理, 这个模型, 同样也可应用于其它无线通信网络。

第三章 遗传算法及其在频率分配中的应用

随着人工智能应用领域的不断扩大,传统的基于符号处理机制的人工智能方法在知识表示、信息处理和解决组合爆炸等方面所遇到的困难越来越明显,从而使得寻求一种适合于大规模问题并具有自组织、自适应、自学习能力的算法成为有关学科的一个研究目标。遗传算法(genetic algorithms,简称 GA)就是 J.Holland 于 1975 年受生物进化论的启发而提出的。

3.1 遗传算法的基本思想和步骤

3.1.1 遗传算法的基本思想和步骤

遗传算法是基于“适者生存”的一种高度并行、随机和自适应的优化算法,它将问题的求解表示成“染色体”的适者生存过程,通过“染色体”群的一代代不断进化,包括复制、交叉和变异等操作,最终收敛到“最适应环境”的个体,从而求得问题的最优解或满意解。

遗传算法是一类随机优化算法,但它不是简单的随机比较搜索,而是通过对染色体的评价和对染色体中基因的作用,有效地利用已有信息来指导搜索有希望改善优化质量的状态。标准遗传算法的主要步骤描述如下:

- (1) 随机产生一组初始个体构成初始种群,并评价每一个体的适配值(fitness value)。
- (2) 判断算法收敛准则是否满足。若满足则输出搜索结果;否则执行以下步骤。
- (3) 根据适配值大小以一定方式执行复制操作。
- (4) 按交叉概率 p_c 执行交叉操作。
- (5) 按变异概率 p_m 执行变异操作。
- (6) 返回步骤(2)。

上述算法中,适配值是对染色体(个体)进行评价的一种指标,是 GA 进行优化所用的主要信息,它与个体的目标值存在一种对应关系;复制操作通常采用比例复制,即复制概率正比于个体的适配值,如此意味着适配值高的个体在下一代中复制自身的概率大,从而提高了种群的平均适配值;交叉操作通过交换两父代个体的部分信息构成后代个体,使得后代继承父代的有效模式,从而有助于产生优良个体;变异操作通过随机改变个体中某些基因而产生新个体,有助于增加种群的多样性,避免早熟收敛。

标准遗传算法的流程图如下:

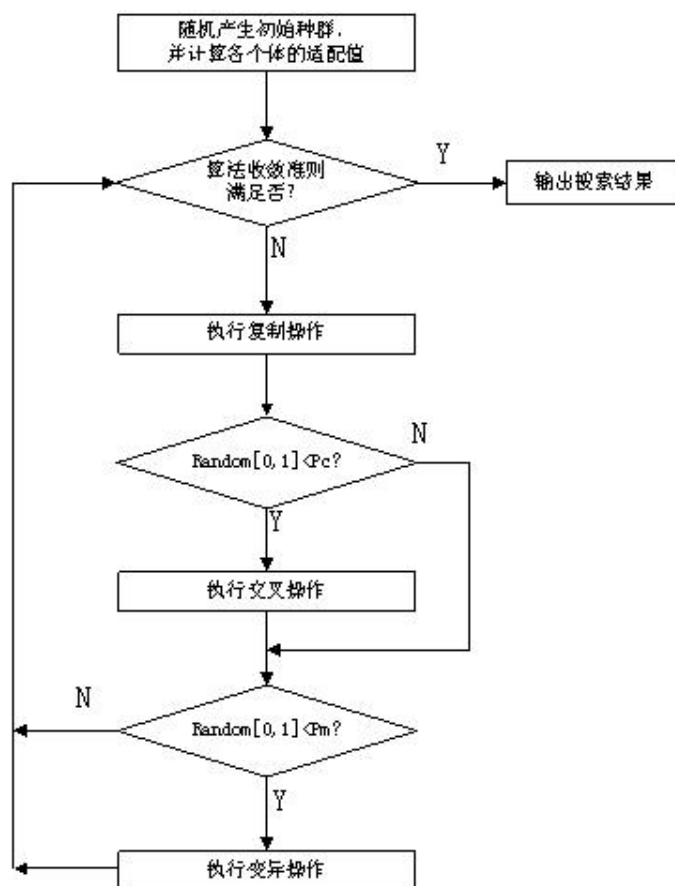


图 3.1 基本遗传算法流程图

遗传算法利用生物进化和遗传的思想实现优化过程，区别于传统优化算法，它具有以下特点：

(1) GA 对问题参数编码成“染色体”后进行进化操作，而不是针对参数本身，这使得 GA 不受函数约束条件的限制，如连续性、可导性等。

(2) GA 的搜索过程是从问题解的一个集合开始的，而不是从单个个体开始的，具有隐含并行搜索特性从而大大减小了陷入局部极小的可能。

(3) GA 使用的遗传操作均是随机操作，同时 GA 根据个体的适配值信息进行搜索，无需其他信息，如导数信息等。

遗传算法的优越性主要表现在：

(1) 算法进行全空间并行搜索，并将搜索重点集中于性能高的部分，从而能够提高效率且不易陷入局部极小。

(2) 算法具有固有的并行性，通过对种群的遗传处理可处理大量的模式，并且容易并行实现。

3.1.2 遗传算法基本理论

1. 模式定理

在采用二进制编码的遗传算法中,模式指的是基于三元字符集 $\{0, 1, *\}$ 的字符串,其中*为通配符,模式中某一位出现*表示该位可取 0 或 1。模式描述了种群中具有相似结构特征的个体的集合,即与模式相匹配的个体所构成的集合。模式与某一个体相匹配是指模式中确定位(取值为 0 或 1 的位置)的值与该个体对应位置的值相同(同为 0 或同为 1),模式中的*可匹配个体中对应位置的 0 或 1。

模式与某一个体相匹配也称该个体包含此模式。由于不同的模式可以匹配相同的个体,因此某一个体可以包含多个不同的模式。设个体染色体编码长度为 l , 则该个体所包含模式数为 2^l , 从而一个规模为 N 的种群可包含的最大模式数为 $2^l \times N$ 。模式将不同的个体联系起来,遗传算法中对个体进行的选择、交叉和变异等操作本质上是对模式的操作。

在遗传算法的模式分析中,用模式的阶来描述模式的确定性以及模式在变异操作中被破坏的概率,用模式的定义距来描述模式在交叉操作中被破坏的概率。

模式 H 的阶定义为模式 H 中确定字符(0 和 1)的个数,记为 $O(H)$ 。模式的阶越高,其确定性也越高,它所匹配的个体也就越少。在遗传算法的变异操作中,模式的阶越高,它被破坏(即变异后的下一代中无包含该模式的个体)的概率也越大。

模式 H 的定义距定义为模式 H 中首尾两个确定字符位置之间的距离(用对应位数的差表示),记为 $\delta(H)$ 。在遗传算法的交叉操作中,模式的定义距越长,它被破坏(即交叉后的下一代中无包含该模式的个体)的概率也越大。

设种群规模为 N , 第 t 代种群为 $P(t) = \{x_1(t), x_2(t), \dots, x_N(t)\}$, $x_i(t) (i=1, 2, \dots, N)$ 为种群 $P(t)$ 中的第 i 个个体, H 为任一模式, $S(H, t)$ 为种群 $P(t)$ 中与模式 H 相匹配的个体的集合, $m(H, t)$ 为种群 $P(t)$ 中与模式 H 相匹配的个体的数目, $m(H, t+1)$ 为第 $t+1$ 代种群 $P(t+1)$ 中与模式 H 相匹配的个体的数目, 模式 H 的平均适应度 $f(H, t)$ 为模式 H 相匹配的所有个体的平均适应度有:

$$f(H, t) = \frac{1}{m(H, t)} \sum_{x \in S(H, t)} f(x) \quad 3-1$$

$\bar{f}(t)$ 为 $P(t)$ 中个体的平均适应度, 即有: $\bar{f}(t) = \frac{1}{N} \sum_{x \in P(t)} f(x)$

SG A 的模式定理为:

设遗传算法的交叉率和变异率分别为 P_c 和 P_m , 个体染色体编码长度为 l , 则有:

$$m(H, t+1) \geq m(H, t) \frac{f(H, t)}{\bar{f}(t)} [1 - P_c \frac{\delta(H)}{l-1} - O(H)P_m] \quad 3-2$$

设存在常数 $C > 1$, 使得对于任意的 t , 均有 $\frac{f(H, t)}{\bar{f}(t)} = C$, 令

$K = C \cdot \left[1 - P_c \frac{\delta(H)}{l-1} - O(H)P_m \right]$, 当 $\delta(H)$ 和 $O(H)$ 很小时, 若 $K > 1$, 并保持为一常数, 则有:

$$m(H, t+1) \geq m(H, t)K$$

3-3

按上式可递推出: $m(H, t) \geq m(H, 0)K^t$

故有模式定理的如下推论: 在遗传算法中, 具有低阶、短定义距和平均适应度高于种群平均适应度的模式在选择、交叉和变异的作用下其所匹配的个体数目将以指数级增长。由于该推论是在 $K > 1$ 并保持为一常数的假设条件下推出的, 但此条件不一定总是满足, 因此上面的推论有时不一定成立。当个体染色体编码长度较长, 种群规模很大时, 上述结论在很大程度上是有效的。

2. 积木块假设

在遗传算法中, 具有低阶、短定义距和平均适应度高于种群平均适应度的模式被称为积木块。

遗传算法在执行过程中, 不是通过逐一或随机搜索式地测试每个可行解的组合来寻找更优的可行解, 而是通过对算法中积木块的遗传操作产生出适应度更高的个体以找到更优的可行解。

积木块假设的内容为: 低阶、短定义距、高平均适应度的模式(即积木块)在遗传操作的作用下, 能够相互组合, 产生适应度更高的个体, 最终生成全局最优解。

由模式定理可推导出在一定条件下积木块相匹配的个体数目以指数级增长, 故由模式定理可知遗传算法存在找到全局最优解的可能性。积木块假设则指出了遗传算法具有找到全局最优解的能力。

上述假设未得到证明, 但遗传算法的大量实践和诸多的成功应用均支持了这个假设。

3. 欺骗问题

在遗传算法中, 将所有妨碍适应度高的个体生成从而影响遗传算法正常工作的问题统称为欺骗问题。

个体适应度函数设计不恰当, 染色体编码不合适均可能产生欺骗问题, 使得遗传算法的执行陷入局部最优解。

在实际应用中, 存在着一些遗传算法难以求解的问题(称为 GA 难题), 在这些问题的求解过程中, 遗传算法往往被迷惑和欺骗, 使得算法的执行偏离全局最优解。

上述 GA 难题存在欺骗问题的原因是: 这些 GA 难题中往往含有孤立的最优点, 即最优点的周围是一些较差的点。现实中遇到的问题中很少具有这种孤立最优点的性质。

4. 隐含并行性

在SGA中, 一个染色体编码长度为 l , 规模为 N 的种群包含了 2^l 到 $2^l \times N$ 个模式。由模式定理知, 在算法的执行过程中, 一些定义距较长的模式被交叉操作破坏。遗传算法的隐含并行性对算法中有效处理的模式(即以指数级增长的模式)的数目进行了估计, 它指出, 遗传算法有效处理的模式数与种群规模的立方成正比, 即, 设种群规模为 N , 则遗传算法

有效处理的模式数为 $O(N^3)$ 。

隐含并行性表明, 虽然从表面上看遗传算法执行过程中的每一代只处理了 N 个个体, 但实际上算法同时处理了 $O(N^3)$ 个有效模式。

5. 收敛性理论

在遗传算法的执行过程中, 种群逐代进化, 若将每一代种群看作一种状态, 则算法的整个执行过程可视为一个随机过程, 并可用 Markov 链对该过程和算法的收敛性进行分析。

设遗传算法所求解的优化问题为使目标函数 $f(x)$ 达到最大值,

$P(t) = \{x_1(t), x_2(t), \dots, x_N(t)\}$ 为算法中的第 t 代种群, N 为种群规模, $x_k(t)$ 是 $P(t)$ 中的第 $k(k = 1, 2, \dots, N)$ 个个体。个体的适应度函数用目标函数 $f(x)$ 来表示,

$Z_t = \max\{f(x_k(t)) \mid k = 1, 2, \dots, N\}$ 为 $P(t)$ 中个体的最大适应度 $f^* = \max\{f(x) \mid x \in B^l\}$, 其中, l 为个体染色体编码长度, $B^l = \{0, 1\}^l$ 为个体所构成的空间。遗传算法收敛到全局最优解, 当且仅当下式成立: $\lim P\{Z_t = f^*\} = 1$, 式中, P 表示概率。

在SGA中, 设个体染色体编码长度为 l , 种群的规模为 N , 由于算法中的每一代种群为一个状态, 则算法的状态空间为: $S = B^{l \cdot N} = \{0, 1\}^{l \cdot N}$, 状态空间 S 所含的状态数为 $2^{l \cdot N}$, 当给定 l 和 N 且为有限数时, 则 S 是一个有限的状态空间。SGA的当前种群只与其前一代种群有关, 在算法的执行过程中, 每一代种群(状态)通过选择、交叉和变异操作变换为另一代种群(状态), 这种状态的转移与进化代数(即时间)无关, 因此SGA可看成是一个有限的齐次Markov链($P(t), t > 0$), 其中 $P(t)$ 为第 t 代种群。

关于SGA的收敛性, 主要有以下定理性的内容:

1. SGA 不能以概率 1 收敛到全局最优解。
2. 对 SGA 进行改进, 在选择操作之后保留当前最优解, 则 SGA 能够以概率 1 收敛到全局最优解。
3. 对 SGA 进行改进, 在选择操作之前保留当前最优解, 则 SGA 能够以概率 1 收敛到全局最优解。

3.2 遗传算法关键参数和操作设计

遗传算法利用生物进化和遗传的思想实现优化过程, 区别于传统优化算法, 其关键参数与操作的设计如下:

3.2.1 编码方法

遗传算法通过编码将所求问题的可行解变换为基因空间中的染色体。编码方法的设计与所求问题的性质有关, 另一方面, 编码方法的设计也将影响甚至决定着交叉、变异等遗传操作的设计。

下面介绍几种主要的编码方法。

1. 二进制编码 采用二进制编码时, 所求问题的可行解被变换为由 0 和 1 构成的个体。设所求问题中的变量为 x , 其定义域为 $[x_{\min}, x_{\max}]$, 二进制编码后的个体染色体长度为 l , 问题所要求的求解精度为 δ , 则 δ 和 l 存在如下关系式:

$$\delta = \frac{x_{\max} - x_{\min}}{2^l - 1} \quad 3-4$$

设有一个体, 其染色体编码为: $b_l b_{l-1} \dots b_1$, 其中 $b_i \in \{0, 1\}$ 该个体对应的问题中的变量值为: $x = x_{\min} + (\sum_{i=1}^l b_i \cdot 2^{i-l}) \cdot \frac{x_{\max} - x_{\min}}{2^l - 1}$, 对应着编码的反过程, 即解码。

2. 格雷码编码 格雷码是在二进制编码基础上进行的一种编码。

设某个体的二进制编码为 $b_l b_{l-1} \dots b_1$, 其相应的格雷码为 $g_l g_{l-1} \dots g_1$ 则从二进制编码到格雷码的变换公式为:

$$g_i = \begin{cases} b_l, & i = l \\ b_{i+1} \oplus b_i, & i = l-1, l-2, \dots, 1 \end{cases} \quad 3-5$$

其中, \oplus 为模 2 加运算。从二进制编码到格雷码的变换公式为:

$$b_i = \begin{cases} g_l, & i = l \\ b_{i+1} \oplus g_i, & i = l-1, l-2, \dots, 1 \end{cases} \quad 3-6$$

在格雷码中, 任何两个相邻整数对应的格雷码之间仅有一位取值不同, 其余位取值完全相同, 即任何两个相邻整数对应的格雷码之间的 Hamming 距离为 1。

3. 浮点数编码 浮点数编码将个体中每个基因的值用对应变量定义域内的一个浮点数来表示, 个体染色体编码的长度等于问题中变量的个数。

采用浮点数编码时, 往往需要针对编码特性, 设计一些专门的遗传操作。

4. 符号编码 在符号编码中, 个体染色体由一串无数值含义的字符组成。符号编码可用于解决组合优化中的有序问题(即优化问题中的目标函数与个体染色体中各字符在染色体中的位置有关), 如, 旅行商问题。

采用符号编码方法时, 若在遗传算法中使用一般的遗传操作, 则在后代中将产生无效个体, 因此对该编码方法需设计专门的遗传操作。

5. 多参数级联编码 多参数级联编码用于含多个变量的优化问题求解中, 该编码方法将所求问题中的每个变量分别以某种编码方法编码, 然后将所有变量的编码按一定顺序级联起来构成一个完整的个体。

在多参数级联编码中, 对每个变量进行编码时可采用前面介绍的二进制编码、格雷码、浮点数编码或符号编码中的任何一种编码方法, 并且每个变量可以具有不同的定义域和编码长度。

6. 可变长编码 可变长编码中, 个体染色体编码长度在遗传算法的执行过程中是动态变化的。

Goldberg 等提出的被称为 MessyGA(简称 MGA)的遗传算法即采用了可变长编码方法。

在 MGA 的可变长编码方法中, 将常规遗传算法(采用染色体长度固定的编码方法)中染色体的基因位置编号和对应的基因值组成一个二元组, 再将这些二元组按一定顺序排列起来便构成了长度可变的染色体。可变长染色体表示为: $X^m : (i_1, v_1)(i_2, v_2) \cdots (i_k, v_k) \cdots (i_n, v_n)$, 式中 $i_k (k=1, 2, \dots, n)$ 为 X^m 中基因在原常规染色体中的基因位置编号, $v_k (k=1, 2, \dots, n)$ 为第 i_k 号基因位置对应的基因值。

3.2.2 适应度函数

遗传算法用适应度函数指导算法的搜索, 因此适应度函数的设计将直接影响遗传算法的性能。

1. 目标函数与适应度函数 适应度函数通常由优化问题中的目标函数转换而来。由于遗传算法搜索的是适应度最大的个体, 并且适应度比例法的选择操作要求适应度函数必须非负, 因此, 在由目标函数构造适应度函数时需考虑原优化问题的种类(最大值问题或最小值问题)以及适应度函数的非负性。

设优化问题中的目标函数为 $g(x)$, x 为决策向量, 当目标函数含有 n 个变量时, x 即为一个 n 维向量, 与 $g(x)$ 对应的适应度函数为 $f(x)$, 则可采用的适应度函数构造方法有:

(1) 当求 $g(x)$ 的最小值时, 变换公式可用:

$$f(x) = \begin{cases} C_{\max} - g(x), & g(x) < C_{\max} \\ 0, & g(x) \geq C_{\max} \end{cases} \quad 3-7$$

式中 C_{\max} 可以是一个适当的输入值(如 $g(x)$ 的最大值估计), 也可以是当前种群中 $g(x)$ 的最大值或到目前为止各进化代中 $g(x)$ 的最大值。

(2) 当求 $g(x)$ 的最大值时, 变换公式可用:

$$f(x) = \begin{cases} g(x) - C_{\min}, & g(x) > C_{\min} \\ 0, & g(x) \leq C_{\min} \end{cases} \quad 3-8$$

式中 C_{\min} 可以是一个适当的输入值(如 $g(x)$ 的最小值估计), 也可以是当前种群中 $g(x)$ 的最小值或到目前为止各进化代中 $g(x)$ 的最小值。

2. 适应度函数的尺度变换 在遗传算法进化的初期, 可能存在适应度远远大于种群平均适应度的个体, 若采用适应度比例法的选择操作, 则这些个体将控制算法的选择过程从而迅速地使它们在种群中占据绝对的比例, 与此同时, 那些含有优良基因但适应度不高的个体也被迅速地种群中淘汰掉, 导致算法过早地收敛到局部最优解。

在遗传算法进化的后期, 算法逐渐收敛, 种群的平均适应度已接近种群中个体的最大

适应度。种群中具有平均适应度的个体和具有最大适应度的个体被选中的概率几乎相等从而种群中已不存在生存上的竞争，算法继续优化搜索的能力降低。此时算法收敛速度降低并可能收敛到局部最优解。

适应度函数的尺度变换用于在遗传算法的不同执行阶段对适应度函数进行缩小或放大。在算法执行的初期，对适应度函数进行适当的缩小变换减小种群中个体之间的差异，避免出现过早收敛。在算法执行的后期，对适应度函数进行适当的放大变换增大最优个体和种群中其他个体适应度之间的差异，从而提高个体的竞争力。

尺度变换方法有:线性尺度变换、幂尺度变换和指数尺度变换。

(1)线性尺度变换 变换公式为:

$$f' = af + b \quad 3-9$$

式中, f 为原适应度, f' 为变换后的适应度, a 和 b 为系数。对于变换前后, 希望满足:

$$f'_{avg} = f_{avg} \quad 3-10$$

$$f'_{max} = Cf_{avg} \quad 3-11$$

上面两式中, f_{avg} 为原适应度平均值, f'_{avg} 为变换后的适应度平均值, f'_{max} 为变换后的适应度函数的最大值, C 作为系数用来控制适应度最大的个体在下一代中的期望复制数。当种群规模为 50~100 时, 常取 $C=1.2\sim 2$ 。可保证适应度为平均适应度的个体下一代中的期望复制数为 1。

将图 3-2 中 $P(f_{avg}, f_{avg})$ 、 $Q(f_{max}, Cf_{avg})$ 两点的坐标值代入(3-9)式中可解得:

$$a = \frac{(C-1)f_{avg}}{f_{max} - f_{avg}} \quad 3-12$$

$$b = \frac{(f_{max} - Cf_{avg})f_{avg}}{f_{max} - f_{avg}} \quad 3-13$$

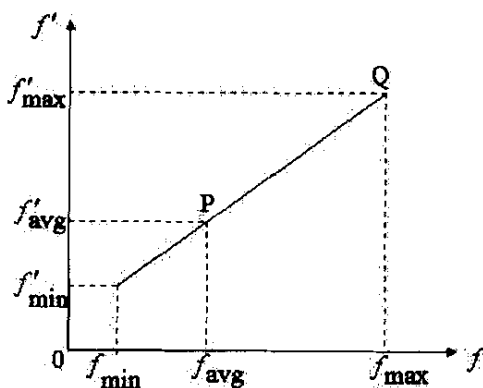


图 3.2 适应度函数尺度变换(正常情况)

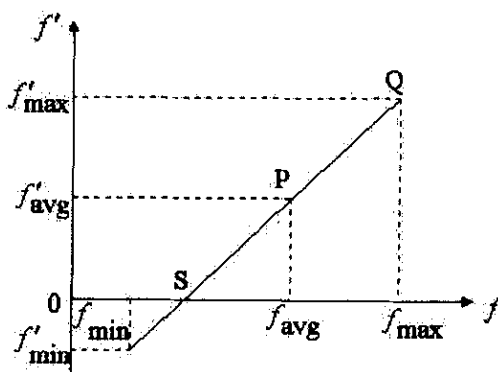


图 3.3 适应度函数尺度变换(出现负值)

在遗传算法执行的后期,种群中个体的平均适应度接近种群中个体的最大适应度,若种群中存在某些个体的适应度远远低于种群中个体的平均适应度和种群中个体的最大适应度,则为了满足(3-11)式,尺度变换后的适应度函数可能会出现负值,如图 3-3 所示。为保证适应度函数的非负性和进行尺度变换,可令尺度变换后的适应度函数的最小值 $f'_{\min} = 0$ (f'_{\min} 与原适应度函数的最小值 f_{\min} 对应),并保证(3-10)式成立。

在尺度变换后的适应度函数出现负值的情况下,将点 $(f_{\min}, 0)$ 和点 $(f_{\text{avg}}, f_{\text{avg}})$ 的坐标值代入(3-9)式中可解得:

$$a = \frac{f_{\text{avg}}}{f_{\text{avg}} - f_{\min}} \quad 3-14$$

$$b = \frac{-f_{\min} f_{\text{avg}}}{f_{\text{avg}} - f_{\max}} \quad 3-15$$

在(3-9)式中令 $f' = 0$ 得直线 $f' = af + b$ 的过零点 S 对应的 f 值(f_s)为:

$$f_s = \frac{Cf_{\text{avg}} - f_{\max}}{C-1} \quad 3-16$$

因此,当原适应度函数的最小值 $f_{\min} \geq f$ 时,(3-9)式的系数 a 和 b 分别采用(3-12)式和(3-13)式;当原适应度函数的最小值 $f_{\min} < f$ 时,(3-9)式的系数 a 和 b 分别采用(3-14)式和(3-15)式。

(2)幂尺度变换变换公式为:

$$f' = f^k \quad 3-17$$

式中, f 为原适应度, f' 为变换后的适应度, k 为幂指数, k 的取值与求解的问题有关。

(3)指数尺度变换变换公式为:

$$f' = e^{-\beta f} \quad 3-18$$

式中, f 为原适应度, f' 为变换后的适应度, β 为系数。

3.2.3 选择操作

选择操作根据个体的适应度对个体进行选择从而引导遗传算法进行有目的优化搜索。下面介绍常用的选择操作方法。

1. 适应度比例 选择适应度比例选择也称轮盘赌选择,在该方法中,每个个体的选择概率(被选中的概率)与其适应度成正比。

设种群规模为 N , 种群中第 $i(i=1,2,\dots,N)$ 个个体的适应度为 f_i , 则第 i 个个体的选择概率 p_i 为:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (i=1,2,\dots,N) \quad 3-19$$

适应度比例选择可用轮盘赌的方法决定哪些个体被选中,即将整个轮盘划分为 N 个扇形,第 $i(i=1,2,\dots,N)$ 个扇形的中心角为 $2\pi p_i$, 设置一个指针使其能够指向上述轮盘中的任意一个扇形并且指针固定不动。转动轮盘,当轮盘停止不动时,指针指向的扇形所对应的个体即被选中。

适应度比例选择的算法实现可以是:在 $[0,1]$ 区间产生一个随机数 r , 对 p_i 从零开始进行累加,当满足条件 $\sum_{j=1}^{i-1} p_j < r \leq \sum_{j=1}^i p_j$ 时则停止累加并选择第 i 个个体作为选择操作的结果。

2. 最优个体保存策略 在这种选择策略中,当前种群中适应度最高的个体不进行交叉及变异操作,而是用它替换当前种群中经过交叉和变异等操作后产生的适应度最低的个体。

该选择策略的操作过程如下:

(1)找出当前种群中适应度最高和最低的个体。

(2)若当前种群中最优个体的适应度比到当前代为止的最优个体的适应度高,则当前种群中的最优个体作为到当前代为止的最优个体。

(3)以到当前代为止的最优个体替换当前种群中的最差个体。

3. 期望值方法 在适应度比例选择方法中,由于选择是随机的,因此可能会出现适应度高的个体未被选择而适应度低的个体被选择的情况,即存在选择误差。期望值选择方法可降低选择误差,其操作过程如下:

(1)计算种群中每个个体在下一代中生存的期望数目。设种群规模为 N , 种群中第 $i(i=1,2,\dots,N)$ 个个体的适应度为 f_i , 第 i 个个体的选择概率为 p_i , 则第 i 个个体在下一代中生存的期望数目 $N_i = N \cdot p_i$, p_i 的计算见(3-19)式。

(2)若某一个体被选中进行交叉运算,则它在下一代中的生存期望数目减 0.5; 若某一个体未被选中进行交叉运算,则它在下一代中的生存期望数目减 1。

(3)在选择操作的过程中,若某个个体的生存期望数目小于 0,则该个体不再参与选择。

4. 排序选择 该方法首先按种群中个体适应度由大到小的顺序对个体进行排序, 再根据排序后的顺序为每个个体分配其选择概率, 最后采用适应度比例选择方法对个体进行选择操作。个体的选择概率与个体在排序后的顺序号有关, 根据个体选择概率和排序后的顺序号之间关系的不同, 排序选择又分为线性排序选择和非线性排序选择。由于排序依据的是个体适应度的大小, 因此排序选择对适应度函数无非负性的要求。

5. 锦标赛选择 该方法每次从种群中随机地选择一定数目(称为竞赛规模)的个体, 然后从这些个体中选择适应度最大的个体进入下一代, 反复执行此操作直到产生规定数目的个体为止。竞赛规模一般取为 2。由于锦标赛选择方法根据个体适应度的大小进行选择, 因此它对适应度函数也无非负性的要求。

3.2.4 交叉操作

遗传算法主要通过交叉操作产生新的个体, 使得算法可以在整个基因空间范围内进行搜索。下面介绍常用的交叉操作方法。

1. 单点交叉 在进行单点交叉时, 首先在进行交叉操作的两个个体中随机地选取某个位置作为交叉点, 然后将两个个体在交叉点之后或者之前的染色体段进行互换从而产生两个新的个体。当个体染色体长度为 l 时, 则可能的交叉点位置有 $l-1$ 个。

2. 两点交叉 在进行两点交叉时, 首先在进行交叉操作的两个个体中随机地选取两个位置作为交叉点, 然后将两个个体在交叉点之间的染色体段进行互换从而产生两个新的个体。

3. 多点交叉 在进行多点交叉时, 在进行交叉操作的两个染色体个体中随机地选取 k 个位置作为交叉点, 假设对这 k 个交叉点从进行交叉操作的个体的最左端开始按顺序进行编号(从 1 开始), 则从第一个交叉点开始, 两个个体间断地交换相邻两个交叉点之间的染色体段(对于最后一个交叉点, 则是最后一个交叉点之后的染色体段)从而产生两个新的个体。当交叉点数为 k 时, 进行交叉操作的个体将有 k 个上面所指的染色体段, 同样, 对这 k 个染色体段从进行交叉操作的个体的最左端开始按顺序进行编号(从 1 开始), 则间断地交换染色体段指的是只交换奇数号的染色体段。

单点交叉和两点交叉均为多点交叉的特例, 分别对应 $k=1$ 和 $k=2$ 。

4. 均匀交叉 在均匀交叉方法中, 进行交叉操作的两个个体的每一个基因均可能参加交换。该方法首先产生一个与进行交叉操作的个体的染色体等长的二进制字符串(称为掩码或交叉模板), 此掩码用于指示进行交叉操作的两个个体的某个基因是否交换。掩码的某一位取值为 1 表示进行交叉操作的两个个体的对应位置的基因进行交换, 若取值为 0 则表示不交换。在产生掩码后, 进行交叉操作的两个个体根据掩码的指示进行交换从而生成两个新的个体。

5. 算术交叉 算术交叉针对浮点数编码, 它又分部分算术交叉和整体算术交叉。

设进行交叉操作的两个个体分别为 X_1 和 X_2 , $x_i^{(1)}$ 和 $x_i^{(2)}$ 分别为 X_1 和 X_2 中从最左端开始排列的第 $i(i=1,2,\dots,l)$ 个分量, 其中 l 为个体染色体长度, X_1 和 X_2 算术交叉后产生的两个个体记为 X_1' 和 X_2' , $x_i^{(1)'}$ 和 $x_i^{(2)'}$ 分别为 X_1' 和 X_2' 中从最左端开始排列的第 i 个分量。

设从 X_1 和 X_2 , 中的第 $k(k=1,2,\dots,l-1)$ 分量进行部分算术交叉, 则有:

$$x_i^{(1)} = \begin{cases} x_i^{(1)}, & i=1,2,\dots,k \\ a_i x_i^{(1)} + (1-a_i)x_i^{(2)}, & i=k+1,k+2,\dots,l \end{cases} \quad 3-20$$

$$x_i^{(2)} = \begin{cases} x_i^{(2)}, & i=1,2,\dots,k \\ a_i x_i^{(2)} + (1-a_i)x_i^{(1)}, & i=k+1,k+2,\dots,l \end{cases} \quad 3-21$$

式中 $a_i(i=k+1,k+2,\dots,l)$ 为区间 $(0, 1)$ 内的随机数。

整体算术交叉对进行交叉操作的两个个体的每个分量均进行交叉并有:

$$x_i^{(1)} = a_i x_i^{(1)} + (1-a_i)x_i^{(2)}, i=k+1,k+2,\dots,l \quad 3-22$$

$$x_i^{(2)} = a_i x_i^{(2)} + (1-a_i)x_i^{(1)}, i=k+1,k+2,\dots,l \quad 3-23$$

式中 $a_i(i=k+1,k+2,\dots,l)$ 为区间 $(0, 1)$ 内的随机数。

3.2.5 变异操作

变异操作可增加种群中个体的多样性以及提高遗传算法的局部搜索能力。下面介绍常用的变异操作方法。

1. 基本位变异 该方法针对二进制编码, 它随机地选取进行变异操作的个体中的某些基因位置, 并对这些位置上的基因值进行取反操作, 从而生成新的个体。

2. 均匀变异 该方法用于浮点数编码下的变异操作。设进行变异操作的个体为 $X=(x_1, x_2, \dots, x_l)$, 式中 $x_i \in [a_i, b_i]$, $i=1,2,\dots,l$, 其中 l 为个体染色体长度。在进行均匀变异时, 首先从 X 中随机选取某个分量, 假设选择了 x_i , 然后在该分量的定义域内(对于本例, 为 $[a_i, b_i]$)随机地选取一个数并用它取代该分量原先的值(对于本例为 x_i)从而生成新的个体。

3. 正态变异 正态变异用于浮点数编码下的变异操作。设个体的表示同 2, 在进行正态变异时, 首先从 X 中随机选取一个分量 x_i , 然后用服从正态分布 $N(\mu_i, \delta_i^2)$ 的一个随机数 x_i' 代替 x_i , 从而生成新的个体。正态分布中的 μ_i 和 δ_i^2 与 a_i 和 b_i 有关。

4. 非均匀变异 非均匀变异用于浮点数编码下的变异操作。它将变异操作与进化代数联系起来, 在进化初期, 变异范围较大, 而在进化后期, 变异范围较小, 从而增强了进化后期遗传算法的局部搜索能力。

设个体的表示同 2, 在进行非均匀变异时, 首先从 X 中随机选取一个分量 x_i , 然后用一个随机数 x_i' 代替 x_i 从而生成新的个体, 其中 x_i' 的计算式为:

$$x_i' = \begin{cases} x_i + \Delta(t, b_i - x_i), & \text{random}(0,1) = 0 \\ x_i + \Delta(t, x_i - a_i), & \text{random}(0,1) = 1 \end{cases} \quad 3-24$$

式中 t 为当前进化代数, $\Delta(t,y)$ 为 $[0,y]$ 区间内服从非均匀分布的一个随机数。当 t 增加时, $\Delta(t,y)$ 接近 0 的概率随之增加。 $\Delta(t,y)$ 的计算式可用:

$$\Delta(t,y) = y \cdot (1 - r^{(1-t/T)^\lambda}) \quad 3-25$$

式中 r 为 $[0,1]$ 区间内的一个随机数, T 为最大进化代数, 参数 λ 用于调整局部搜索区域。

3.2.6 遗传算法控制参数选取

遗传算法中的控制参数主要有:个体染色体编码长度、种群规模、交叉率 p_c 、变异率 p_m 以及最大进化代数。下面对 SGA 中种群规模、交叉率 p_c 、变异率 p_m 、以及最大进化代数, 这些控制参数的选取作些说明。

1. 个体染色体编码长度 在采用二进制编码时, 个体染色体编码长度与所求问题的精度要求和问题中变量的个数有关。在采用浮点数编码时, 个体染色体编码长度等于所求问题中变量的个数。

2. 种群规模 由遗传算法的隐含并行性可知, 种群规模越大, 算法同时处理的模式越多, 则算法通过积木块的组合求得问题最优解的概率越大, 亦即, 种群规模越大, 种群中个体的多样性越高, 遗传算法陷入局部最优解的危险越小。另一方面, 种群规模太大将增加遗传算法的计算量从而影响算法的执行效率。种群规模太小将使得种群中个体多样性降低并容易使遗传算法过早收敛。

3. 交叉率 交叉率越大, 每代进行交叉操作的次数越多, 从而加快了遗传算法向最优解区域收敛的速度, 但交叉率太大将破坏种群中适应度高的个体并可能造成算法过早收敛。若交叉率太小, 则每代进行交叉操作的次数少, 导致算法搜索缓慢。

4. 变异率 大的变异率可增加种群中个体的多样性, 使算法不断搜索解空间中的新的区域, 但变异率太大将破坏种群中适应度高的个体, 使得遗传算法成为随机搜索。交叉率太小将降低种群中个体的多样性以及遗传算法的局部搜索能力。

5. 最大进化代数 最大进化代数用来控制遗传算法终止运行的条件, 对其大小的设定依问题而定。

3.2.7 约束条件的处理

在应用遗传算法求解带约束条件的优化问题时, 根据约束条件和问题的不同, 对约束条件的处理可采用不同方法。

设优化问题的决策向量为 $x = (x_1, x_2, \dots, x_n)$, 问题的约束条件为: $a_i \leq x_i \leq b_i$, 式中 $i = 1, 2, \dots, n$, 则对这种约束条件的处理可通过编码的方法来实现。根据问题计算精度的要求,

设每个变量的编码位数为 l_i ，则个体染色体的长度为 $l = \sum_{i=1}^n l_i$ 。

对于优化问题中等式形式的约束条件，可通过删除多余变量来进行处理。设优化问题中决策向量的表示与前面相同，问题的约束条件为： $h_j(x) = 0$ ，式中 $j = 1, 2, \dots, m$ ，其中 $m < n$ ，则可以通过求解这 m 个约束条件组成的方程组，将方程组中的 m 个变量用剩余的 $n-m$ 个变量来表示，从而消除了问题中的约束条件。该方法将约束问题转化为无约束问题。

在遗传算法求解带约束条件的优化问题的过程中，可能会由于交叉操作和变异操作产生无效个体(即个体对应的解不满足问题中的约束条件)。为了不让这些无效个体进入下一代，可以在交叉操作和变异操作中加入循环控制语句，使得交叉操作和变异操作在未产生有效个体(即个体解码之后的值是问题解空间中的一个可行解)之前一直进行循环操作。

将有约束问题转化为无约束问题的另一种方法是罚函数法。该方法对无效个体的适应度进行惩罚，惩罚值由罚函数给出。设约束优化问题中的原适应度函数为 $f(x)$ ，其中 x 为问题中的决策向量， X 为优化问题的可行解域，则采用了罚函数后的适应度函数 $f'(x)$ 为：

$$f'(x) = \begin{cases} f(x), & x \in X \\ f(x) - \sigma p(x), & x \notin X \end{cases} \quad 3-26$$

式中， $\sigma > 0$ ，称为罚因子，罚函数 $P(x)$ 为连续函数，并满足：

$$p(x) = \begin{cases} 0, & x \in X \\ > 0, & x \notin X \end{cases} \quad 3-27$$

3.3 遗传算法在频率分配中的应用

1. 个体编码的方法

对于频率分配问题，每个染色体长度为 N ，相当于要分配频率的发射机数量。染色体里每个元素值都为整数，对应一个频率，一种频率分配为 $f = (f_1, f_2, \dots, f_N)$ 。我们采用符号编码的方法，即对个体每一个基因座用可用频率的编号进行编码。如果有 12 个频率可用，那么第一个个体解就是由 1 到 12 这 12 正整数组成的。例如，对于长度为 6 的个体，其解的编码可能是 3-12-11-5-9-3，表示编号为 1 到 6 的基站分别采用编号为 3、2、11、5、9 和 3 的频率。这样有以下好处：直观，便于理解并研究算法的运算过程；易于解码，因为在基站编号（即频率分配的解）和个体基因之间有简单的一一对应关系。

2. 初始化种群

染色体初始群体可以随机选取，即对每个染色体，在染色体内每个发射机从可用频率集合中随机分配一个频率。这也是最常用的办法，但提高效率可借助启发式方式快速产生解，并由此产生 GA 的初始种群。下面程序采用二进制编码随机初始化种群：

```
void generateinitialpopulation( ) //种群初始化
```

```

{
    int i, j;
    for (i=0; i<popsize; i++)
    {
        for(j=0; j<chromlength; j++)
        {
            population[i].chrom[j]=(rand()%10<5)?'0':'1';
        }
        population[i].chrom[chromlength]='\0';
    }
}

```

3. 适配函数

在频率分配中，目标是求干扰最小值，即目标值小的个体的适应度就大，而且随着运算的深入，目标值会逐渐减小。如何把在算法后期运行阶段的个体之间已经差别不大的适应度放大呢？我们受指数尺度变换方法的启发，并将适应度置换和适应度尺度变换这两步结合起来，得到如下变换方法：

$$F(x)' = C_{\max} - \ln[f(x)]$$

这里， x 是解空间中的个体， $f(x)$ 是目标函数值， $F(x)'$ 是变换后的适应度， C_{\max} 是一个较大的数。

这样，在算法搜索初期阶段，虽然各个个体之间的目标函数值相差较大，但适应度差别并不大，可以在解空间进行比较均匀的搜索。在后期阶段，对解空间搜索的范围减小到最优解附近，群体中的解之间的适应度差别被放大，加速了搜索速度。

```

void calculatefitnessvalue()//计算适应度
{
    int i;
    double temp;
    for(i=0; i<popsize; i++)
    {
        if(functionmode==maximization)
        {
            if((population[i].value+cmin)>0.0)
            {
                temp=cmin+population[i].value;
            }
        }
    }
}

```

```

        else
        {
            temp=0.0;
        }
    }
    else if (functionmode==minimization)
    {
        if(population[i].value<cmax)
        {
            temp=cmax-population[i].value;}
        else
        {
            temp=0.0;
        }
    }
    population[i].fitness=temp;
}
}

```

4. 选择算子的设计

针对频率分配问题，我们使用最优保存策略来设计选择算子。其步骤如下：

1. 从父辈群体中依适应度大小的概率选择健壮的个体进行交配，产生与父辈群体规模相同的子辈群体。
2. 从父辈群体中选择适应度最大的 m 个个体，取代子辈群体中适应度最小的 m 个个体。
3. 将子辈群体取代父辈群体。

这里较为关键问题是保留个数 m 的选取，如果 m 过大，则容易陷入局部最优；相反，如果 m 过小，则不能保证优良个体的保护，会影响到算法的收敛速度。而且我们对这个参数的选取又缺乏一个直观的概念。

在下面程序中函数selectoperator()中首先用rand()函数产生 0~1 间的选择算子，当适度累计值不为零时，比较各个体所占总的适应度百分比的累计和与选择算子，直到达到选择算子的值那个个体就被选出，即适应度为 f_i 的个体以 $f_i/\sum f_k$ 的概率继续存在；显然，个体适应度愈高，被选中的概率愈大。但是，适应度小的个体也有可能被选中，以便增加下一代群体的多样性。

```
void selectoperator() //选择算子
{
    int i, index;
    double p, sum=0.0;
    double cfitness[POPSIZE];
    struct individual newpopulation[POPSIZE];
    for(i=0;i<popsiz; i++)
    {
        sum+=population[i].fitness;
    }
    for(i=0;i<popsiz; i++)
    {
        cfitness[i]=population[i].fitness/sum;
    }
    for(i=1;i<popsiz; i++)
    {
        cfitness[i]=cfitness[i-1]+cfitness[i];
    }
    for (i=0;i<popsiz;i++)
    {
        p=rand()%1000/1000.0;
        index=0;
        while (p>cfitness[index])
        {
            index++;
        }
        newpopulation[i]=population[index];
    }
    for(i=0;i<popsiz; i++)
    {
        population[i]=newpopulation[i];
    }
}
```

5. 交配算子的设计

交配算子的目的对优良模式进行搜索，如何在交配过程中避免破坏好的模式而又产生新的模式以增加群体多样性从而尽可能避开局部最优是我们设计交配算子的目标。

最佳模式应该规定了关键基因的优良组合，即要找最优模式的话，首先应该在这些关键基因的这些组合中搜索。因此，我们要通过交配运算，使群体中所有的个体涵盖尽可能多的这些组合模式，从而尽量增加多样性。基于此，我们结合了点交叉和均匀的优点，依概率把交配点设定为关键点所在的基因座。这样，通过交配运算，就主生尽可能多的关键点组合。

但是交配点的数量如何选取呢？如果交配点的数量较多，则容易破坏一些比较好的模式，降低收敛速度；若交配点的数量较少，则不利于在整个解空间内搜索好的模式，从而导致陷入局部最优。因此，需要进一步研究参数选取的方法。

针对频率分配问题，交配算子描述如下：

1. 依概率取 s 个最关键点所处位置为交配位置，其中 s 为交配点数量。
2. 依概率取健壮的两个个体做为父母，以 1 中确定的交配位置进行多点交配，产生两个子代个体，加入子辈群体。
3. 检查子辈群体的规模是否等于父辈群体，若是，则跳转到 1；若非，则跳转到 2。

下面程序首先用 `rand()` 函数产生随机概率，若小于交叉概率，则进行染色体交叉，同时交叉次数加 1。这时又要用 `rand()` 函数随机产生一位交叉位，把染色体的交叉位的后面部分交叉即可；若大于交叉概率，则进行简单的染色体复制即可。

```
void crossoveroperator() //交叉算子
{
    int i, j;
    int index[POPSIZE];
    int point, temp;
    double p;
    char ch;
    for (i=0; i<popsize; i++)
    {
        index[i]=i;
    }
    for (i=0; i<popsize; i++)
    {
        point=rand()%(popsize-i);
```



```

        temp=index[i];
        index[i]=index[point+i];
        index[point+i]=temp;
    }
    for (i=0;i<popsi-1;i+=2)
    {
        p=rand()%1000/1000.0;
        if (p<pc)
        {
            point=rand()%(chromlength-1)+1;
            for (j=point; j<chromlength;j++)
            {
                ch=population[index[i]].chrom[j];
                population[index[i]].chrom[j]=population[index[i+1]].chrom[j];
                population[index[i+1]].chrom[j]=ch;
            }
        }
    }
}

```

6. 变异算子的设计

变异有均匀变异和非均匀变异之分，“均匀”是针对变异点处出现的新基因而言的，而非针对变异位置。我们正是要针对这一点对其进行适合于频率分配的改进。

变异算子的目的和交配算子一样，都是在解空间中寻找最优解。我们希望变异算子在开始运行阶段，变异操作使群体多样性增加，出现更多交配算子所不能产生的模式（变异算子作为交配算子的补充）。随着算法运行的深入，最佳模式逐步地被确定下来，这时候变异算子能够在最佳模式的附近搜索最优解。这样，我们使用一种新的变异算子，它不但可以针对频率分配问题，对于其他类似的问题也会有好的效果。

变异是针对染色体字符变异的，而不是对个体而言，即个体变异的概率是一样。随机产生比较概率，若小于变异概率，则 1 变为 0，0 变为 1，同时变异次数加 1。

```

void mutationoperator() //变异算子
{
    int i,j;
    double p;

```

```

for (i=0;i<popsi; i++)
{
    for(j=0;j<chromlength; j++)
    {
        p=rand()/1000.0;
        if (p<pm)
        {
            population[i].chrom[j]=(population[i].chrom[j]=='0')?'1':'0';
        }
    }
}
}

```

7. 算法终止准则

由于最大进化代数很难合适设置,为适应算法性能的动态变化,较好地兼顾算法的优化性能和时间性能,可采用阈值法设计算法终止准则,即若最佳优化值连续若干代进化仍保持不变,则终止搜索过程。

3.4 遗传算法性能

假设无线通信网络中总共有 N 个发射机(基站)等待分配频率,切彼此之间限制关系已知;能使用的频率范围为 F_{\min} --- F_{\max} ;

则我们可以用如下方法来处理问题:

首先,为了解决同小区频率间隔的限制.我们将 $F=F_{\max}-F_{\min}$ 分成若干个小的频段,假定分为 45 个频段,分别为 f_1, f_2, \dots, f_{45} , 然后取 $F_1=f_1, f_{16}, f_{31}; F_2=f_2, f_{17}, f_{32}; \dots$ $F_{15}=f_{15}, f_{30}, f_{45}$. 这样将 15 组频率段分配给不同的小区,则小区内部就不存在频率间隔的限制了。而邻频与同频的约束就要靠算法使干扰约束数减到最小了。

遗传算法的性能参数如下:

最大代数 80, 群体大小 60, 交叉次数 20, 变异次数 5, 基因范围 1--15。所得到的仿真结果如图 3-4:

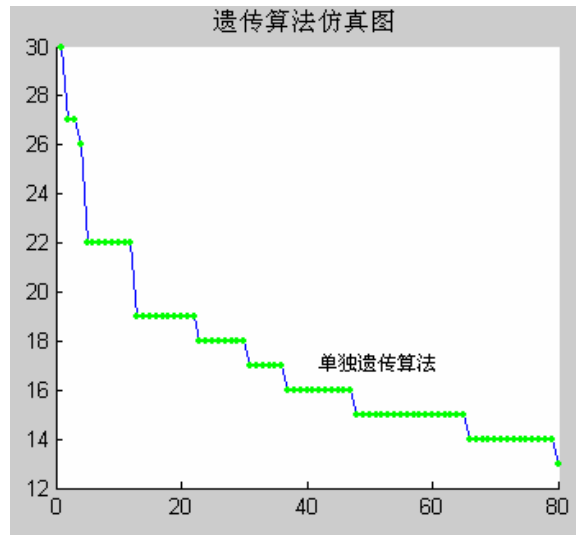


图 3.4 遗传算法仿真图

图中,横轴为代数,纵轴为干扰约束数。可见,运行到第80代的时候,对上面的参数,最小违背约束数仍达不到0。这主要受到频点个数,限制矩阵,最大代数,群体大小,交叉次数,变异次数这些参数的影响,其中前两项的影响最大,但由于它们是由客观条件决定的,故我们无法依靠在程序中改变它们的值来优化结果,它们只能通过改变客观条件来优化。我们所能改变的是后四个参数,达到更好的效果。

下面我们改变一下前两个参数,取最大代数为800代,群体大小为80进行仿真,得到结果如图3-5。

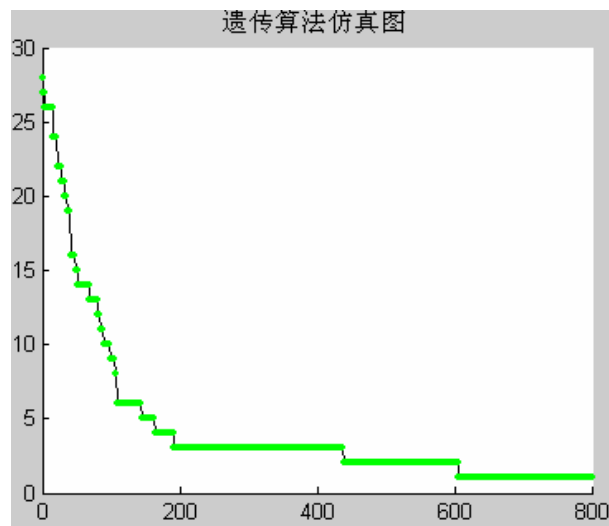


图 3.5 改变参数后遗传算法仿真图

当程序运行到800代时,干扰约束数仍然为1而达不到0。

第四章 模拟退火算法在频率分配中的应用

模拟退火算法 (Simulated Annealing, 简称 SA) 的思想最早由 Metropolis 等提出的, 1983 年 Kirkpatrick 等将其用于组合优化。SA 算法是基于 Monte Carlo 迭代求解策略的一种随机寻优算法, 其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。模拟退火算法在某一初温下, 伴随温度参数的不断下降, 结合概率突跳特性在解空间中随机寻找目标函数的全局最优解, 即在局部最优解能概率性地跳出并最终趋于全局最优。模拟退火算法是一种通用的优化算法, 目前已在工程中得到了广泛应用, 诸如生产调度、控制工程、机器学习、神经网络、图像处理等领域。

4.1 模拟退火算法的基本思想和步骤

模拟退火算法最早是针对组合优化问题提出的, 其目的在于: (1) 为具有 NP 复杂性问题提供有效的提供求解算法; (2) 克服优化过程陷入局部极小; (3) 克服初值依赖性。

SA 由某一较高温度开始, 利用具有概率突跳性的 Metropolis 抽样策略在邈空间中进行随机搜索, 伴随温度的不断下降重复抽样过程, 最终得到问题的全局最优解。

模拟退火算法可以分解为解空间、目标函数和初始解三部分。

模拟退火的基本思想:

(1) 初始化: 初始温度 T (充分大), 初始解状态 S (是算法迭代的起点), 每个 T 值的迭代次数 L 。

(2) 对 $k=1, \dots, L$ 做第(3)至第 6 步。

(3) 产生新解 S'

(4) 计算增量 $\Delta t' = C(S') - C(S)$, 其中 $C(S)$ 为评价函数。

(5) 若 $\Delta t' < 0$ 则接受 S' 作为新的当前解, 否则以概率 $\exp(-\Delta t'/T)$ 接受 S' 作为新的当前解。

(6) 如果满足终止条件则输出当前解作为最优解, 结束程序。

终止条件通常取为连续若干个新解都没有被接受时终止算法。

(7) T 逐渐减少, 且 $T \rightarrow 0$, 然后转第 2 步。

算法对应流程图 4.1:

模拟退火算法新解的产生和接受可分为如下四个步骤:

第一步是由一个产生函数从当前解产生一个位于解空间的新解; 为便于后续的计算和接受, 减少算法耗时, 通常选择由当前新解经过简单地变换即可产生新解的方法, 如对构成新解的全部或部分元素进行置换、互换等, 注意到产生新解的变换方法决定了当前新解的邻域结构, 因而对冷却进度表的选取有一定的影响。

第二步是计算与新解所对应的目标函数差。因为目标函数差仅由变换部分产生，所以目标函数差的计算最好按增量计算。事实表明，对大多数应用而言，这是计算目标函数差的最快方法。

第三步是判断新解是否被接受,判断的依据是一个接受准则，最常用的接受准则是Metropolis 准则：若 $\Delta t' < 0$ 则接受 S' 作为新的当前解 S ，否则以概率 $\exp(-\Delta t'/T)$ 接受 S' 作为新的当前解 S 。

第四步是当新解被确定接受时，用新解代替当前解，这只需将当前解中对应于产生新解时的变换部分予以实现，同时修正目标函数值即可。此时，当前解实现了一次迭代。可在此基础上开始下一轮试验。而当新解被判定为舍弃时，则在原当前解的基础上继续下一轮试验。

模拟退火算法与初始值无关，算法求得的解与初始解状态 S (是算法迭代的起点) 无关；模拟退火算法具有渐近收敛性，已在理论上被证明是一种以概率 1 收敛于全局最优解的全局优化算法；模拟退火算法具有并行性。

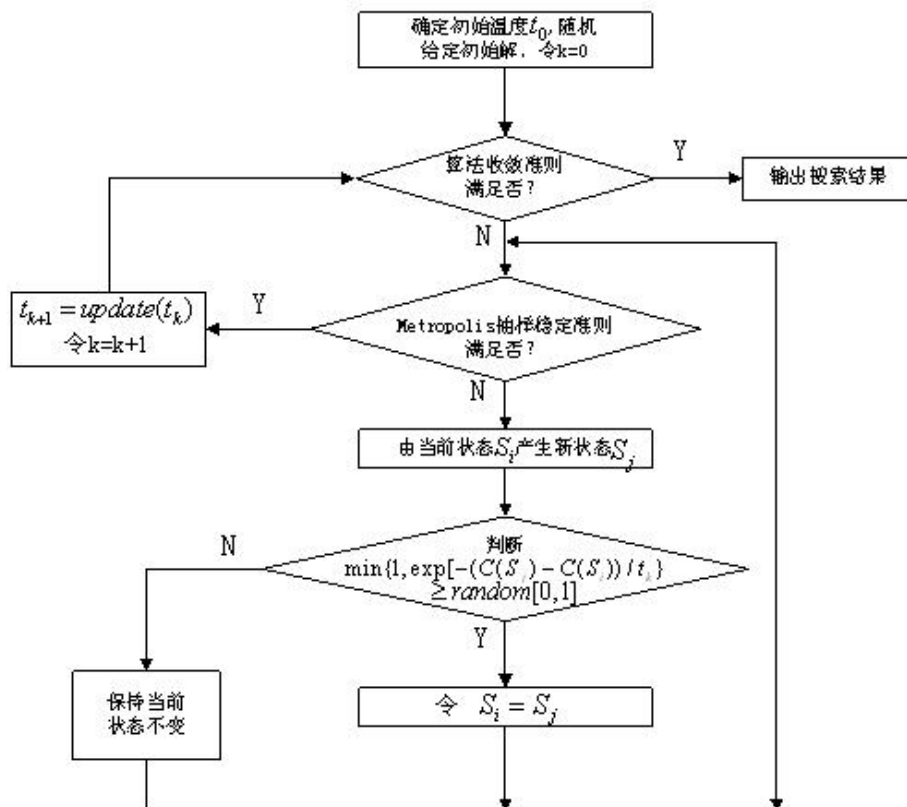


图 4.1 模拟退火算法流程图

从算法的结构知，新状态产生函数、新状态接受函数、退温函数、抽样稳定准则和退火结束准则（简称三函数两准则）以及初始温度是直接影响算法优化结果的主要环节。模拟退火算法的实验性能具有质量高、初值鲁棒性强、通用易实现的优点。但是，为寻求

到最优解，算法通常要求较高的初温、较慢的降温速率、较低的终止温度以及各温度下足够多次的抽样，因而模拟退火算法往往优化过程较长，这也是 SA 算法最大的缺点。因此，在保证一定优化质量的前提下提高算法的搜索效率，是对 SA 进行改进的主要内容。

4.2 模拟退火算法关键参数和操作设计

从算法流程看，模拟退火算法包括三函数两准则，即状态产生函数、状态接受函数、温度更新函数、内循环终止准则和外循环终止准则，这些环节的设计将决定 SA 算法的优化性能。此外，初温的选择对 SA 算法性能也有很大影响。

理论上，SA 算法的参数只有满足算法的收敛条件，才能保证实现的算法依概率 1 收敛到全局最优解。然而，由 SA 算法的收敛性理论知，某些收敛条件无法严格实现，如时齐马氏链的内循环终止准则，即使某些条件可以实现，如非时齐马氏链的更新函数，但也常常会因为实际应用的效果不理想而不被采用。因此，至今 SA 算法的参数选择依然是一个难题，通常只能依据一定的启发式准则或大量的实验加以选取。

1. 状态产生函数

设计状态产生函数（邻域函数）的出发点应该是尽可能保证产生的候选解遍布全部解空间。通常，状态产生函数由两部分组成，即产生候选解的方式和候选解产生的概率分布。前者决定由当前解产生候选解的方式，后者决定在当前解产生的候选解中选择不同的状态的概率。候选解的产生方式由问题的性质决定，通常在当前状态的邻域结构内以一定概率方式产生，而邻域函数和概率方式可以多样化设计，其中概率分布可以是均匀分布、正态分布、指数分布、柯西分布等。

2. 状态接受函数

状态接受函数一般以概率的方式给出，不同接受函数的差别主要在于接受概率的形式不同。设计状态接受概率，应该遵循以下原则：

（1）在固定温度下，接受使目标函数值下降的候选解的概率要大于使目标函数值上升的候选解的概率；

（2）随温度的下降，接受使目标函数值上升的解的概率要逐渐减小；

（3）当温度趋于零时，只能接受目标函数值下降的解。

状态接受函数的引入是 SA 算法实现全局搜索的最关键的因素，但实验表明，状态接受函数的具体形式对算法的性能的影响不显著。因此，SA 算法中通常采用 $\min[1, \exp(-\Delta C/t)]$ 作为状态接受函数。

3. 初温

初始温度 t_0 、温度更新函数、内循环终止准则和外循环终止准则通常被称为退火历程 (annealing schedule)。

在非时齐 SA 算法收敛性理论中, 初温由退温函数 $t_k = \alpha / \log(k + k_0)$ 中的 α 决定, 但求解实际问题时很难精确得到 α 值或其下界。

在时齐 SA 算法收敛性理论中, 虽没有对初温给出限制, 但根据与物理退火过程的类比关系, 初温应选择充分大以使几乎所有产生的候选解都能被接受, 如此保证最终优良的收敛性。

实验表明, 初温越大, 获得高质量解的几率越大, 但花费的计算时间将增加。因此, 初温的确定应折衷考虑优化质量和优化效率, 常用的方法包括:

(1) 均匀抽样一组状态, 以各状态目标值的方差为初温。

(2) 随机产生一组状态, 确定两两状态间的最大目标值差 $|\Delta_{\max}|$, 然后依据差值, 利用一定的函数确定初温。譬如, $t_0 = -\Delta_{\max} / \ln p_r$, 其中 p_r 为初始接受概率。若取 p_r 接近 1, 且初始随机产生的状态能够一定程度上表征整个状态空间时, 算法将以几乎等同的概率接受任意状态, 完全不受极小解的限制。

(3) 利用经验公式给出。

4. 温度更新函数

温度更新函数, 即温度的下降方式, 用于在外循环中修改温度值。

在非时齐 SA 算法收敛性理论中, 更新函数可采用函数 $t_k = \alpha / \log(k + k_0)$ 。由于温度与退温时间的对数函数成反比, 所以温度下降的速度很慢。当 α 取值较大时, 温度下降到比较小的值需要很长的计算时间。快速 SA 算法采用更新函数 $t_k = \beta / \log(1 + k)$, 与前式相比, 温度下降速度加快了。但需要强调的是, 单纯温度下降速度加快并不能保证算法以较快的速度收敛到全局最优, 温度下降的速率必须与状态产生函数相匹配。

在时齐 SA 算法收敛性理论中, 要求温度最终趋于零, 但对温度的下降速度没有任何限制, 但这并不意味着可以使温度下降得很快, 因为在收敛条件中要求各温度下产生的候选解数目无穷大, 显然这在实际应用时是无法实现的。通常, 各温度下产生候选解越多, 温度下降的速度可以越快。

目前, 最常用的温度更新函数为指数退温, 即 $t_{k+1} = \lambda t_k$, 其中 $0 < \lambda < 1$ 且大小可以不断变化。

5. 内循环终止准则

内循环终止准则, 或称 Metropolis 抽样稳定准则, 用于决定在各温度下产生候选解的数目。在非时齐 SA 算法理论中, 由于在每个温度下只产生一个或少量候选解, 所以不存在选择内循环终止准则的问题。而在时齐 SA 算法理论中, 收敛性条件要求在每个温度下产生候选解数目趋于无穷大, 以使相应的马氏链达到平稳概率分布, 显然在实际应用算法时这是无法实现的。常用的抽样稳定准则包括:

(1) 检验目标函数的均值是否稳定;

(2) 连续若干步的目标值变化较小;

(3) 按一定的步数抽样。

6. 外循环终止准则

外循环终止准则, 即算法终止准则, 用于决定算法何时结束。设置温度终值 t_e 是一种简单的方法。SA 算法的收敛性理论中要求 t_e 趋于零, 这显然是不实际的。通常的做法包括:

(1) 设置终止温度的阈值;

(2) 设置外循环迭代次数;

(3) 算法搜索到的最优值连续若干步保持不变;

(4) 检验系统熵是否稳定。

由于算法的一些环节无法在实际设计算法时实现, 因此 SA 算法往往得不到全局最优解, 或算法结果存在波动性。许多学者试图给出选择“最佳”SA 算法参数的理论依据, 但所得绪论与实际应用还有一定距离, 特别是对连续变量函数的优化问题。目前, SA 算法参数的选择仍依赖于一些启发式准则和待求问题的性质。SA 算法的通用性很强, 算法易于实现, 但要真正取得质量和可靠性高、初值鲁棒性好的效果, 克服计算时间较长、效率较低的缺点, 并适用于规模较大的问题, 尚需进行大量的研究工作。

4.3 模拟退火算法在频率分配中的应用

一种频率分配 $f = (f_1, f_2, \dots, f_N)$ 可以利用指数阵 $[x_1, x_2, \dots, x_N]$ 来表示, 其中对 $1 \leq j \leq N$, $f_j = d_{x_j}$ 。

单迁移邻域是指 f 邻域为指数阵恰好在某分量上不同的那些分配。因此, 如果 f' 用 $[x'_1, x'_2, \dots, x'_N]$ 描述, 那么 $f' = (f'_1, f'_2, \dots, f'_N)$ 为 f 的邻域。如果存在 $j (1 \leq j \leq N)$, 且 $x_j \neq x'_j$, 则对所有 $i = 1, 2, \dots, N$ 及 $i \neq j$, 有 $x_i = x'_i$ 。

因此任何一种分配方案都会有 $N(M-1)$ 个邻域解, 其中 M 为频率数。每一个邻域解对应一对 (j, x'_j) , 其中 $1 \leq j \leq N$, $1 \leq x'_j \leq M$, $(x'_j \neq x_j)$ 。

双迁移邻域是指 f 邻域为指数阵恰好在两个分量上不同的那些分配。如果 f' 用 $[x'_1, x'_2, \dots, x'_N]$ 描述, 那么 $f' = (f'_1, f'_2, \dots, f'_N)$ 为 f 的邻域。如果存在 j_1 和 j_2 ($1 \leq j_1, j_2 \leq N$), 结果有 $x_{j_1} \neq x'_{j_1}$ 和 $x_{j_2} \neq x'_{j_2}$, 且对所有 $i = 1, 2, \dots, N$ 及 $i \neq j_1$ 或 j_2 , 有 $x_i = x'_i$ 。

任何一种分配方案 f 都会有 $\frac{N!(M-1)^2}{2!(N-2)!}$ 个邻域。每个邻域对应两对 (j_1, x'_{j_1}) 和 (j_2, x'_{j_2}) 及

$1 \leq j_1, j_2 \leq N$, $1 \leq x'_{j_1}, x'_{j_2} \leq M$, $x'_{j_1} \neq x_{j_1}$, $x'_{j_2} \neq x_{j_2}$ 。

用于频率分配的模拟退火算法如下:

(1) 输入参数:

发射机数 N ;

约束图 G ;

频率集合 D ;

最低温度 t_{min} 。

(2) 初始条件:

给各个发射机生成随机的频率分配方案, 用 f_{old} 表示;

计算分配的代价 $E(f_{old})$;

确定初始温度 t_0 。

(3) 主循环:

$k=0$

WHILE $t_k > t_{min}$ DO

FOR $u = 1$ to N DO

从 f_{old} 的邻域生成新的分配 f_{new} ;

计算代价函数 $E(f_{new})$;

计算 $\Delta E = E(f_{new}) - E(f_{old})$;

IF $\Delta E < 0$ or $random < prob = e^{-\Delta E / t_k}$ THEN

$f_{old} \leftarrow f_{new}$

$E(f_{old}) \leftarrow E(f_{new})$

END IF

END FOR

减小 t_k 给出 t_{k+1}

$k = k + 1$

END WHILE

输出: 发射机频率分配 f 方案及相关代价函数 $E(f)$ 。

首先设 $t_0=1$ 确定初始温度, 并从收述算法的主循环中 FOR 循环进行 100 次迭代。如果容忍率 $x < 0.9$, 那么加倍当前值 t_0 。容忍率 x 定义为可接受的试验分配数除以 100。继续上述过程, 直至所观察的容忍率 x 超过 0.9 (在开始 FOR 循环之前 x 重新初始化为 0)。

当温度 t_k 低于给定的 t_{min} 或凝固温度超过用户给定值 (通常为 10), 算法结束。对给定温度 t_k (即在 FOR 循环 N 次迭代后), 当无新分配 f_{new} 冰点可接受, 凝固温度就产生了。

1. 编码的选择

用模拟退火算法求解频率分配问题, 采用符号编码的方法非常直观, 即用可用频率的编号进行个体编码。与遗传算法编码类似, 如果有 12 个频率可用, 那么每一个个体解就是由 1 到 12 这 12 个正整数组成的。例如, 对于长度为 6 的个体, 解的编码可能是 3-2-11-5-9-3, 表示编号为 1 到 6 的基站分别采用编号为 3、2、11、5、9 和 3 的频率。

2. SA 状态产生函数的设计

对于 SA 状态产生函数操作, 可将其设计为:

- (1) 互换操作 (SWAP), 即随机交换染色体中两不同基因的位置;
- (2) 逆序操作 (INV), 即将染色体中两不同随机位置间的基因串逆序;
- (3) 插入操作 (INS), 即随机选择某个点插入到串中的不同随机位置。

例如在频率分配中, 初始解的状态为 (5-4-1-7-9-8-6-2-3), 两随机位置为 2,6, 则 SWAP 的结果为 (5-8-1-3-9-4-6-2-3), INV 的结果为 (5-8-9-7-1-4-6-2-3), INS 的结果为 (5-8-4-1-7-9-6-2-3)。

3. SA 状态接受函数的设计

状态接受函数是算法产生概率突跳能力的关键, 能够在分布机制指导下避免局部极小。结合基于 SWAP 算子的状态产生函数, 为了使搜索过程具有克服陷入局部极小的能力, 并满足 SA 算法的对称条件, $\min\{1, \exp(-\Delta/t)\} > \text{random}[0,1]$ 准则是作为接受新状态的条件最常用方案, 其中 Δ 为新旧状态的目标值差, t 为“温度”。

4. 初温和初始状态

最常用且可理解的初温确定方案是, 首先随机产生一组状态, 确定两两状态间的最大目标值差 $|\Delta_{\max}|$, 然后由式 $t_0 = -\Delta_{\max} / \ln p_r$, 其中 p_r 为初始接受概率 (理论上应接近于 1, 实际设计时也可取 0.1)。

初始状态可以随机取, 但为了提高优化效率, 可采用启发式算法快速得到一个解, 并以此为 SA 的初始状态。

5. 退温函数的设计

理论上, 温度应以很慢的速度下降, 如对数的倒数方式, 但为了避免过于冗长的搜索过程, 较好地折衷兼顾优化质量和时间性能, 指数退温函数是最常用的退温策略, 即 $t_k = \lambda t_{k-1}$, λ 为退温速率。

6. 温度修改准则和算法终止准则的设计

为适应算法性能的动态变化, 较好地兼顾算法的优化性能和时间性能, 可采用阈值法设计的“温度修改”和“算法终止”两准则。也即, 若优化过程中得到的最佳优化值边 20 代进化保持不变, 则进行退温; 若最佳优化值 20 次退温仍保持不变, 则终止搜索过程, 以此优化值为算法的优化结果。

4.4 模拟退火与遗传算法结合应用

4.4.1 GASA 算法及步骤

GA 和 SA 两种算法均属于基于概率分布机制的优化算法。不同的是, SA 通过赋予搜索过程一种时变且最终趋于零的概率突跳性, 从而可有效避免陷入局部极小并最终趋于全局最优; GA 则通过概率意义下的基于“优胜劣汰”思想的群体遗传操作不断完善再做化。对选择优化机制上如此差异的两种算法进行混合, 有利于丰富优化过程中的搜索行为, 增强全局和局部意义下的搜索能力和效率。

SA 算法采用串行优化结构, 而 GA 采用群体并行搜索。两者结合, 能够使 SA 成为并行 SA 算法, 提高其优化性能; 同时 SA 作为一种自适应变概率的变异操作, 增强和补充了 GA 的进化能力。

SA 算法的状态产生和接受操作每一时刻仅保留一个解, 缺乏冗余和历史搜索信息; 而 GA 的复制操作能够在下一代中保留种群中的优良个体, 交叉操作能够使后代在一定程度上继承父代的优良模式, 变异操作能够加强种群中个体的多样性。这些不同作用的优化操作相结合, 丰富了优化过程中的邻域搜索结构, 增强了全空间的搜索能力。

由于复制操作对当前种群外的解空间无探索能力。种群中各个体分布“畸形”时交叉操作的能力有限, 小概率变异操作很难增加种群的多样性。所以, 若算法收敛准则设计不好, 则 GA 经常会出现进化缓慢或“早熟”收敛的现象。另一方面, SA 的优化行为对退温历程具有很强的依赖性, 而理论上的全局收敛对退温历程的限制条件很苛刻, 因此 SA 优化时间性能较差。两种算法结合, SA 的两准则可控制算法收敛性以避免出现“早熟”收敛现象, 并行化的抽样过程可提高算法的优化时间性能。

SA 和 GA 对算法参数具有很强的依赖性, 参数选择不合适将严重影响优化性能。SA 的收敛条件导致算法参数选择较为苛刻, 甚至不实用; 而 GA 的参数又没有明确的选择指导, 设计算法时均要通过大量的试验和经验来确定。GA 和 SA 的相混合, 使算法各方面的搜索能力均有提高, 因此对算法参数的选择不必过分严格。研究表明, 混合算法在采用单一算法参数时, 优化性能和鲁棒性均有大幅度提高, 尤其对较大规模复杂问题。

在优化机制、结构和行为上, GASA 混合优化策略均结合了 GA 和 SA 的特点, 使两种算法的搜索能力得到相互补充, 弥补了各处各自的弱点, 它是一种优化能力、效率和可靠性较高的优化方法。对于存在多极小的复杂问题, 混合策略的优化性能将体现出更明显的优越性。

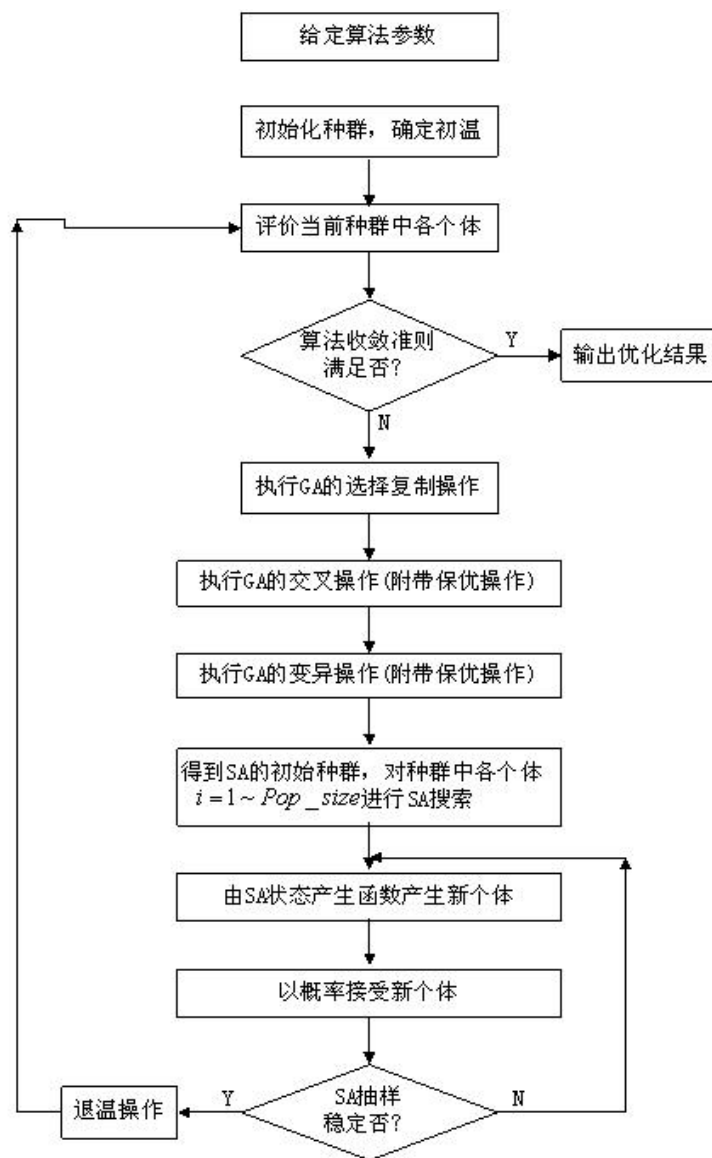


图 4.2 GASA 混合策略流程图

4.4.2 性能比较

在传统遗传算法中, 竞争是在子代中进行, 而子代和父代之间没有竞争, 这样父代中优良个体有可能丢失。有些算法通过直接将群体中最优解放入下一代群体中来保存最优解, 但这有可能引起早熟收敛问题。此外, 由于遗传算法采用随机交叉和变异算子, 交叉和变异后个体并不一定都是优良个体, 这会破坏原有优良个体, 影响算法性能。因此, 通过退火来减轻遗传算法的选择压力, 对所选交叉个体均实施交叉操作, 并且把交叉和变异后的子代同父代竞争, 不但有利于优良个体保留, 同时防止早熟收敛问题。随着讲化过程的讲行, 温度逐步下降, 接收劣质解的概率也逐渐减小。同时, 又利用退火法的爬山性能, 改善了遗传算法性能, 提高了算法的收敛速度。

同样假设网络中有 N 发射机（基站）等待分配频率, 切彼此之间限制关系已知; 能使用的频率范围为 F_{\min} --- F_{\max} 。

起始温度 $T=10$, 冷却参数 $K=0.975$, 终止温度 $T=1$, 最大代数 80, 群体大小 60, 交叉次数 20, 变异次数 5, 基因范围 1---15。仿真结果如下:

遗传与退火联合算法:

```
gen 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 num
    7 11 13 8 7 5 7 11 15 9 4 3 11 1 13 5 3 9 1 7 10 14 5 1 3 6
```

遗传算法:

```
gen 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 num
    76 12 5 14 9 3 11 2 5 11 9 2 9 6 10 8 12 13 14 1 6 6 3 13 15 2 15
```

仿真图如图 4-3

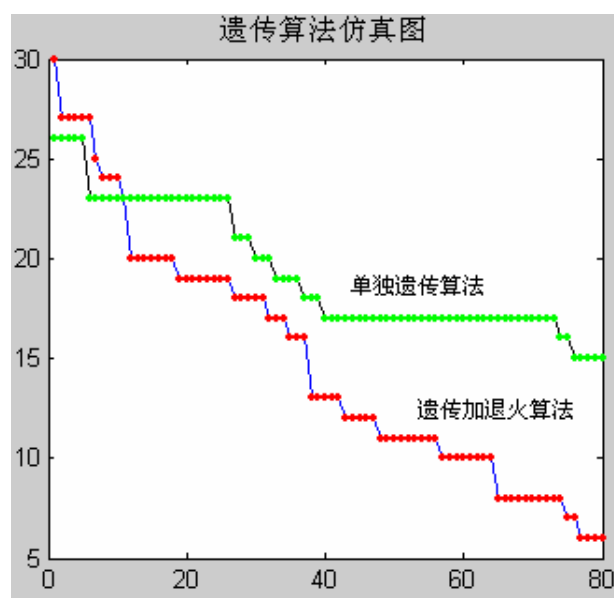


图 4.3 两种算法对比图

遗传与退火算法相结合的收敛速度比单用遗传算法要快很多。

改变参数如下:

最大代数 1000, 群体大小 80, 交叉次数 20, 变异次数 5, 基因范围 1---15。

所得结果:

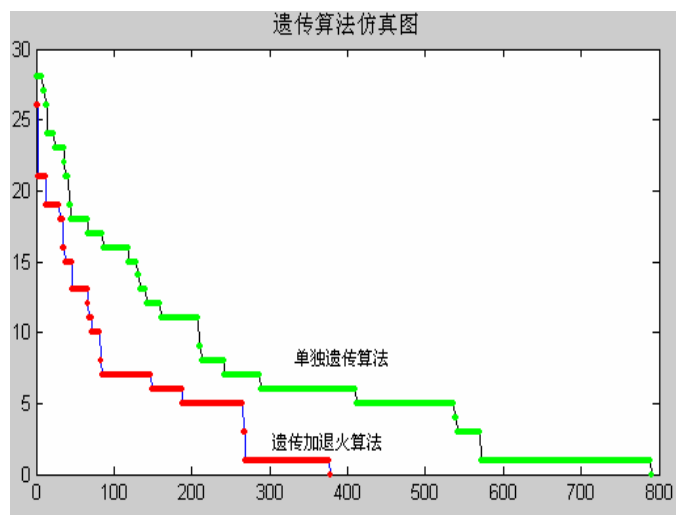


图 4.4 遗传算法与遗传退火算法仿真比较图

遗传算法要到第 800 代才能取得最优解，而遗传与退火相结合的算法在第 380 代就得到了最优解，可见遗传与模拟退火相结合的算法的效果确实比较理想。

第五章 禁忌搜索算法

禁忌搜索 (Tabu Search 或 Taboo Search, 简称 TS) 的思想最早由 Glover(1986)提出, 它是对局部邻域搜索的一种扩展, 是一种全局逐步寻优算法, 是对人类智力过程的一种模拟。TS 算法通过引入一个灵活的存储结构和相应的禁忌准则来避免迂回搜索, 并通过藐视准则来赦免一些被禁忌的优良状态, 进而保证多样化的有效探索以最终实现全局优化。相对于模拟退火和遗传算法, TS 是又一种搜索特点不同的 meta-heuristic 算法。

5.1 禁忌搜索算法的基本思想和步骤

局部邻域搜索是基于贪婪思想持续地在当前解的邻域中进行搜索, 虽然算法通用易实现, 且容易理解, 但其搜索性能完全依赖于邻域结构和初始解, 尤其容易陷入局部极小而无法保证全局优化性。针对局部邻域搜索, 为了实现全局优化, 可尝试的途径有: 以可控性概率接受劣解来逃逸局部极小, 如模拟退火算法; 扩大邻域搜索结构, 如 TSP 的 2opt 扩展到 k-opt; 多点并行搜索, 如进化计算; 变结构邻域搜索; 另外, 就是采用 TS 的禁忌策略尽量避免迂回搜索, 它是一种确定性的局部极小突跳策略。

禁忌搜索是人工智能的一种体现, 是局部邻域搜索的一种扩展。禁忌搜索最重要的思想是标记对应已搜索到的局部最优解的一些对象, 并在进一步的迭代搜索中尽量避开这些对象 (而不是绝对禁止循环), 从而保证对不同的有效搜索途径的探索。禁忌搜索涉及到邻域 (neighborhood)、禁忌表 (tabu list)、禁忌长度 (tabu length)、候选解 (candidate)、藐视准则 (aspiration criterion)、等概念。

简单的 TS 算法的基本思想是: 给定一个当前解 (初始解) 和一种邻域, 然后在当前解的邻域中确定若干候选解; 若最佳候选解对应的目标值优于 “best so far” 状态, 则忽视其禁忌特性, 用其替代当前解和 “best so far” 状态, 并将相应的对象加入禁忌表, 同时修改禁忌表中各对象的任期; 若不存在上述候选解, 则选择在候选解中选择非禁忌的最佳状态为新的当前解, 而无视它与当前的优劣, 同时将相应的对象加入禁忌表, 并修改禁忌各对象的任期; 如此重复上述迭代搜索过程, 直至满足停止准则。

算法流程图如下:

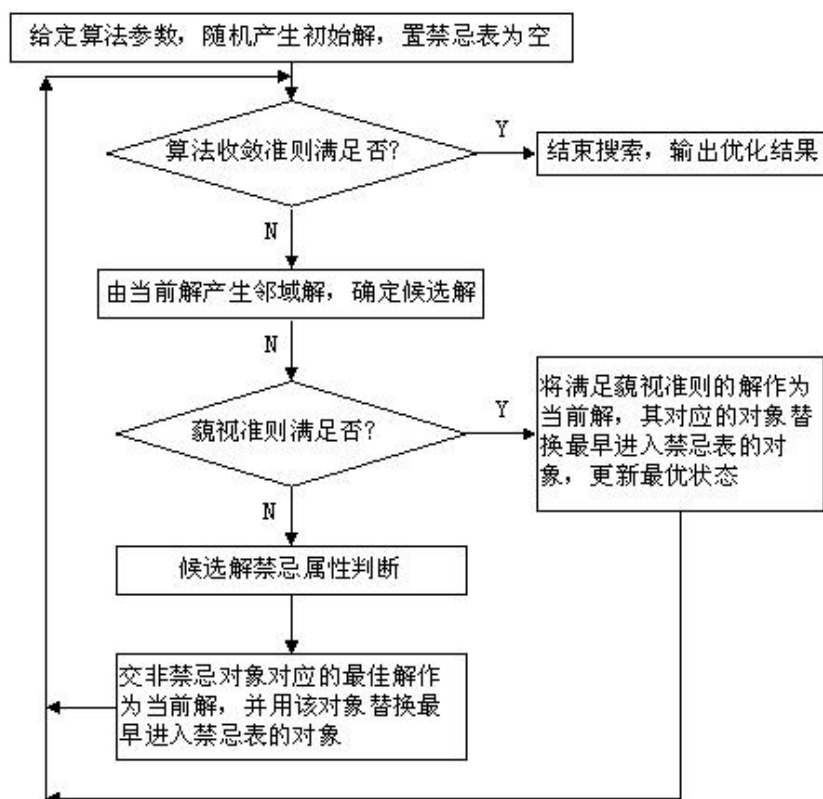


图 5.1 禁忌搜索算法流程图

5.2 禁忌搜索算法关键参数和操作设计

禁忌搜索算法最基本的操作和参数的常用设计原则和方法，包括适配值函数、禁忌对象、禁忌长度、候选解、藐视准则、禁忌频率和终止准则等。

1. 适配值函数

禁忌搜索的适配值函数也是用于对搜索状态的评介，进而结合禁忌准则和藐视准则来选取新的当前状态。显然，目标函数直接作为适配值函数是比较容易理解的做法。当然，目标函数的任何变形都可作为适配值函数，譬如对极小化问题可将状态 x 的适配值 $f(x)$ 定义为 $M - c(x)$ 或 $e^{ac(x)}$ ，其中 M 为一足够大正数， $c(x)$ 为目标值， $a > 0$ 。

若目标函数的计算比较困难或耗时较多，如一些复杂工业过程的目标函数值需要一次仿真才能获得，此时可采用反映问题目标的某些特征值来作为适配值，进而改善算法的时间性能。当然，选取何种特征值要视具体问题而定，但必须保证特征值的最佳与目标函数的最优性一致。

2. 禁忌对象

所谓禁忌对象就是被置入禁忌表中的那些变化元素，而禁忌的目的则是为了尽量避免迂回搜索而多探索一些有效的搜索途径。归纳而言，禁忌对象通常可选取状态本身或状态分量或适配值的变化等。

(1) 以状态本身或其变化作为禁忌对象是最为简单、最容易理解的途径。具体而言,当状态由 x 变化到状态 y 时,将状态 y (或 $x \rightarrow y$ 的变化) 视为禁忌对象,从而在一定条件下禁止了 y (或 $x \rightarrow y$ 的变化) 的再度出现。

(2) 状态的变化包含了多个状态分量的变化,因此以状态分量的变化为禁忌对象将扩大禁忌的范围,并可减少相应的计算量。譬如,对置换问题,SWAP 操作引起的两点互换意味着状态分量的变化,这就可作为禁忌对象;对高维函数优化问题,则可将某一维分量本身或其变化作为禁忌对象。

(3) 类似等高线的原理,以适配值或其变化为禁忌对象则将处于同一适配值的状态视为相同状态,这在函数优化中经常采用。由于一个值的变化隐含着多个状态的变化,因此这种情况下的禁忌范围相对于状态的变化将有所扩大。

以状态本身为禁忌对象比以状态分量或适配值为禁忌对象的禁忌范围要小,从而给予的搜索范围要大,容易造成计算时间的增加。然而,在禁忌长度和候选解集大小相同且较小的情况下,后两者也会因禁忌范围过大而使搜索陷入局部极小。

3. 禁忌长度和候选解

禁忌长度和候选解集的大小是影响 TS 算法性能的两个关键参数。所谓禁忌长度,即禁忌对象在不考虑藐视准则情况下不允许被选取的最大次数(通俗些,可视为对象在禁忌表中的任期),对象只有当其任期为 0 时才被解禁。候选解集则通常是当前状态的邻域解集的一个子集。在算法的构造和计算过程中,一方面要求计算量和存储量尽量少,这就要求禁忌长度和候选解集的尽量小;但是,禁忌长度过短将造成搜索的循环,候选解集过小将容易造成早熟收敛,即陷入局部极小。

禁忌长度的选取与问题特性、研究者的经验有关,它决定了算法的计算复杂性。

一方面,禁忌长度 t 可以是定常不变的,如将禁忌长度固定为某个数(譬如 $t=3$ 等),或者固定为与问题规模相关的一个量(譬如 $t=\sqrt{n}$, n 为问题维数或规模),如此实现很方便、简单。

另一方面,禁忌长度也可以是动态变化的,如根据搜索性能和问题特性设定禁忌长度的变化区间 $[t_{\min}, t_{\max}]$ (譬如 $[3, 10]$ 、 $[0.9\sqrt{n}, 1.1\sqrt{n}]$ 等),而禁忌长度则可按某种原则或公式在其区间内变化。当然,禁忌长度的区间大小也可随搜索性能的变化而动态变化。

一般而言,当算法的性能动态下降较大时,说明算法当前的搜索能力比较强,也可能当前解附近极小解形成的“波谷”较深,从而可设置较大的禁忌长度来延续当前的搜索行为,并避免陷入局部极小。大量研究表明,禁忌长度的动态设置方式比静态方式具有更好的性能和鲁棒性,而更为合理高效的设置方式还有待进一步研究。

候选解通常在当前状态的邻域中择优选取,但选取过多将造成较大的计算量,而选取过少则容易造成早熟收敛。然而,要做到整个邻域的择优往往需要大量的计算,因此可以

确定性或随机性地在部分邻域解中选取候选解，具体数据大小则可视问题特性和对算法的要求而定。

4. 藐视准则

在禁忌搜索算法中，可能会出现候选解全部被禁忌，或者存在一个优于“best so far”状态的禁忌候选解，此时藐视准则将使某些状态解禁，以实现更高效的优化性能。在此给出藐视准则的几种常用方式。

(1) 基于适配值的准则。全局形式（最常用的方式）：若某个禁忌候选解的适配值优于“best so far”状态，则解禁此候选解为当前状态和新的“best so far”状态；区域形式：将搜索空间分成若干个子区域，若某个禁忌候选解的适配值优于它所在区域的“best so far”状态，则解禁此候选解为当前状态和相应区域的新“best so far”状态。该准则可直观理解为算法搜索到了一个更好的解。

(2) 基于搜索方向的准则。若禁忌对象上次被禁时使得适配值有所改善，并且目前该禁忌对象对应的候选解的适配值优于当前解，则对该禁忌对象解禁。该准则可直观理解为算法正有效的搜索途径进行。

(3) 基于影响力的准则。若候选解均被禁忌，且不存在优于“best so far”状态的候选解，则对候选解中最佳的候选解进行解禁，以继续搜索。该准则可直观理解为对算法死锁的简单处理。

(4) 基于最广泛力的准则。在搜索过程中不同对象的适配值的影响有所不同，有的千真万确，有的较小，而这种最广泛力可作为一种属性与禁忌长度和适配值来共同构造藐视准则。直观的理解是，解禁一个影射力大的禁忌对象，有助于在以后的搜索中得到更好的解。需要指出的是，影响力仅是一个标量和一个与禁忌任期相关的阈值，无疑增加了算法操作的复杂性。同时，这些指标最好是动态变化的，以适应搜索进程和性能的变化。

5. 禁忌频率

记忆禁忌频率（或次数）是对禁忌属性的一种补充，可放宽选择决策对象的范围。譬如，如果某个适配值频繁出现，则可以推测算法陷入某种结构或参数作修改。在实际求解时可以根据问题和算法的需要，记忆某个状态的出现频率，也可以是某些对换对象或适配值等信息，而这些信息又可以是静态的，或者是动态的。

静态的频率信息主要包括状态、适配值或对的等对象在优化过程中出现的频率，其计算相对比较简单，如对象在计算中出现的次数，出现次数与总迭代步数的比，某两个状态间循环的次数等。显然，这些信息有助于了解某些对象的特性，以及相应循环出现的次数等。

动态的频率信息主要记录从某些状态、适配值或对换等对象转移到另一些状态、适配值或对的等对象的变化趋势，如记录某个状态序列的变化。显然，对动态频率信息的记录

比较复杂，而这些所提供的信息量也较多。常用的方法如下：

(1) 记录某个序列的长度，即序列中的元素个数，而在记录某些关键点的序列中，可以按这些关键点的序列长度的变化来进行计算。

(2) 记录由序列中的某个元素出发后再回到该元素的迭代次数。

(3) 记录某个序列的平均适配值，或者是相应各元素的适配值的变化。

(4) 记录某个序列出现的频率等。

上述频率信息有助于加强禁忌搜索的能力和效率，并且有助于对禁忌搜索算法参数的控制，或者可据此对相应的对象实施惩罚。譬如，若某个对象频繁，则可以增加禁忌长度来避免循环；若某个序列的适配值变化较小，则可以增加对该序列所在对象的禁忌长度，反之则缩小禁忌长度；若最佳适配值长时间维持下去，则可以终止搜索进程而认为该适配值已是最估值。

此外，许多改进的禁忌搜索算法还根据频率等信息在算法中增加集中搜索（intensification）和分散搜索（diversification）机制，以增强算法的搜索质量和效率。其中，集中搜索机制强调算法对优良区域的重点搜索，如基于最优或次优状态进行重新初始化或进行多步搜索，加强对取得最优状态的算法参数的被选取概率等；分散收缩机制则强调拓宽搜索范围，尤其是那些未探索的区域，这与增强遗传算法种群多样性有些类似。显然，集中搜索和分散搜索在某些层面上是矛盾的，而两者对算法性能都有很大影响，因此作为一个较好的禁忌搜索算法，应当具有合理平衡集中搜索与分散搜索的能力。

6. 终止准则

与模拟退火、遗传算法一样，禁忌搜索也需要一个终止准则来结束算法的搜索进程，而严格实现理论上的收敛条件，即在禁忌长度充分大的条件下实现状态空间的遍历，这显然是不切合实际的，因此实际设计算法时通常采用近似的收敛准则。常用方法如下：

(1) 设定最大迭代步数。此方法简单易操作，但难以保证优化质量。

(2) 设定某个对象的最大禁忌频率。即，若某个状态、适配值或对换等对象的禁忌频率超过某一阈值，则终止算法，其中也包括最佳适配值连续若干步保持不变的情况。

(3) 设定适配值的偏离幅度。即，首先有估计语法估计问题的下界，一旦算法中最佳适配值与下界的偏离值小于某规定幅度时，则终止搜索。

5.3 禁忌搜索算法在频率分配中的应用

下面描述禁忌搜索过程对频率分配问题的实现。我们将解释分配表示、邻域和移动的概念及怎样测量新频率，并描述分配所有邻域怎样有效地估算，使之能在合理时间内有效迭代。禁忌搜索过程全代价函数最小与模拟退火相同。

1. 分配与邻域结构

一种频率分配方案 $f = (f_1, f_2, \dots, f_N)$ 可用指数阵 $[x_1, x_2, \dots, x_N]$ 来表示, 其中对 $i \leq j \leq N$, 有 $f_j = d_{x_j}$ 。

f 的邻域为指数阵恰好在某分量上不同的那些分配。因此, 如果用 $[x'_1, x'_2, \dots, x'_N]$ 来表示 f' , 则 $f' = (f'_1, f'_2, \dots, f'_N)$ 为 f 的邻域。如果存在 $j (1 \leq j \leq N)$, 则 $x'_j \neq x_j$, 且对所有 $i = 1, 2, \dots, N$ 和 $i \neq j$, 有 $x'_i = x_i$ 。

任何分配 f 才有 $N(M-1)$ 个邻域。每个邻域对应一对 (j, x'_j) 及 $1 \leq j \leq N, 1 \leq x'_j \leq N, x'_j \neq x_j$ 。

对于实际情况, 全部邻域通常过大以致难以搜索下一个可能解。为减小邻域的范围, 通常采用下面的方法。在每次迭代中, 计算每个发射机违背的约束数, 只有具有最大违背数的发射机才开始考虑邻域。由于这种限制邻域趋于变小, 所以可以加快迭代速度。然而, 如果限制邻域由禁忌的移动组成且不满足期望标准, 那么就要为该迭代选择邻域。

2. 短期记忆和长期记忆条件

如果对应改变发射机 i 的分配 x'_i 的邻域移动 (i, x'_i) 不满足短期记忆和长期记忆条件, 则认为是禁忌的。这些条件由两个数来确定: 正整数 r 和实数 β , 其中 $0 < \beta < 1$ 。该短期记忆条件说明, 发射机数 i 不是前面 r 移动任何发射机数。长期条件说明, 发射机 i 在所有迭代上改变的次数之比不超过 β 。因此, 如果迭代 j 次, 频率 $d_{x'_i}$ 分配给发射机 i 已经改变, 则要么短期记忆 $d_{x'_i}^{k+1} = d_{x'_i}$, 其中 $k-1 < t \leq k$, 要么长期记忆 $\frac{1}{k} \sum_{j=i, t_j=i_{k+1}}^k 1 > \beta$, 在迭代 $k+1$ 的移动是禁忌的。

在每次迭代中, 从非禁忌邻域中选取 $f^{(k)}$ 的邻域 $f^{(k+1)}$, 使得 $E(f^{(k+1)})$ 最小。注意, $E(f^{(k+1)}) > E(f^{(k)})$ 是可能的, 这样可以避免局部最小。

当某期望标准满足了, 禁忌搜索允许选择一个禁忌移动。在实现过程中, 如果对 $f^{(k)}$ 所有邻域 f' , 邻域 $f^{(k+1)}$ 满足 $E(f^{(k+1)}) \leq E(f')$ 和 $E(f^{(k+1)}) \leq E(f^{(j)})$, $\forall j, 1 \leq j \leq k$, 则选择 tabu 移动。

总之, 如果 f_{tabu} 是 $f^{(k)}$ 的最佳邻域, 而 f_{non} 是最佳非 tabu 邻域, 那么确定 $f^{(k+1)}$ 的法则为

if $E(f_{tabu}) \leq E(f_{non})$ and
 $E(f_{tabu}) \leq E(f^{(j)})$ for $1 \leq j \leq k$
 then $f^{(k+1)} < f_{tabu}$
 else $f^{(k+1)} < f_{non}$

注意到, 参数 r 与 β 之间必须满足一些隐含关系。如果 $r > N$ 或 $\beta < 1/N$, 那么经过 N 次

非禁忌移动后，每次移动都是禁忌的。在经过 k 次非禁忌迭代移动后，从短期记忆条件可以得到一个发射机已经改变至多 k/r 倍。因此，频率值应该满足 $k\beta < k/r$ ，并由此得到 r 和 β 之间的为 $1/N \leq \beta \leq 1/r$ 。实际上，我们对某个 $\lambda (0 \leq \lambda \leq 1)$ 值选择 $\beta = \lambda/r + (1-\lambda)/N$ 。

3. 算法

$k=1$

生成初始解

WHILE 没完成

识别 $N_{set}(s) \subset S$ 。（邻域集合）

识别 $T_{set}(s) \subseteq N_{set}(s)$ 。（禁忌集合）

识别 $A_{set}(s) \subseteq T_{set}(s)$ 。（期望集合）

选择 $s' \in (N_{set}(s) - T_{set}(s)) \cup A_{set}(s)$ ，此时 $h(s')$ 最大

$s = s'$

$k=k+1$

END WHILE

第六章 结束语

本文对无线通信网络中干扰及约束条件进行了研究，提出应用干扰信息进行频率分配。对禁忌搜索算法、模拟退火算法、遗传算法等现代优化算法进行了研究，并将它们应用频谱管理和频率分配中。

本文的主要工作体现在以下几个方面：

1. 研究了提高频率资源利用率的频率复用问题，以及无线通信网络中影响通信质量的干扰及约束条件，并以 GSM 网络为例，建立了网络中干扰（主要是同频干扰和邻频干扰）信息进行频率分配的数学模型。将频率分配问题归结为某个小区或某个区域的干扰评价函数的优化问题，以便使用智能优化算法对频率分配进行优化。

2. 研究了遗传算法、模拟退火算法、禁忌搜索算法的基本理论及操作设计，并将这些算法应用于通信网的频率分配的优化，以降低通信网络中台站之间的相互干扰，以提高频率的利用率，保障无线通信网络的正常运行。

3. 研究了遗传算法和模拟退火算法结合应用，并对其性能进行比较。单独的遗传算法和模拟退火算法的收敛速度都比不上遗传算法与模拟退火算法相结合应用的收敛速度。

本文在研究过程中，算法只考虑了通信网络中的同频干扰和邻频干扰的台站数目，而在实际应用中，还要考虑发射机之间的互调干扰及其它噪声。而在同频干扰和邻频干扰中干扰的强度对性能也有一定的影响。

致 谢

在此论文即将完成之际，我谨向在我攻读硕士学位期间所关心和帮助我的老师和同学们表示深深的谢意！

衷心感谢导师魏急波教授的悉心指导，从论文的选题、取材、结构、写作到问题的分析、论文的修改与完善过程，导师都给予了具体指导和极大的关心。老师渊博的知识、严谨务实的学风、诲人不倦的态度，将使我终生受益。

感谢熊辉副教授和马东堂副教授经常性的亲切关心和指导。

感谢同学们，特别是祝贵凡、张俊同学，经常一起讨论和交流，是我学习和提高的重要途径。

参考文献

- [1] D.H.Smith,S.Hurley,S.U.Thiel,Improving Heuristics for the Frequency Assignment Problem,European Journal of Operational Research, 107(1998)76-86
- [2] D.H.Smith,S.Hurley, Bounds for the Frequency Assignment Problem, Discrete Mathematics, 167/168(1997)571-582
- [3] D.J.Castelino, S.Hurley, N.M.Stephens, A Tabu Search Algorithm for Frequency Assignment, Annals of Operations Research 63(1996)301-319
- [4] IEEE 802.11-1997 Standard for Wireless Lan Medium Access Control(MAC) and Physical Layer (PHY) Specification
- [5] 许良凤, 蜂窝移动通信中基于遗传退火的固定频率分配, 安徽农业大学学报, 2004
- [6] 林巍, 曹若云, 遗传算法在 GSM 移动通信网络频率规划中的应用, 现代电子技术, 2003 年第 20 期
- [7] 童庆, 曹先彬, 钱海, 王煦法, 用于频率分配的免疫应答求解策略, 小型微型计算机系统, 2003, 24
- [8] 罗文坚, 曹先彬, 王煦法, 免疫网络调节算法及其在固定频率分配问题中的应用, 自然科学进展, 2002, 12
- [9] 李旭, 宋俊德, 宋梅, 智能频率规划系统设计与开发, 控制与决策, 2000, 15
- [10] 吴静, 宋俊德, 一种新的用于蜂窝网的频率分配优化算法, 北京邮电大学学报, 1999, 22(4)
- [11] 李旭, 宋俊德, 一种智能频率规划方法, 北京邮电大学学报, 1999, 1 (1)
- [12] 李旭, 宋俊德, 宋梅, 智能多层计算机辅助频率规划系统设计与开发, 电子学报 2000, 28(2)
- [13] 李旭, 宋俊德, 宋梅, 智能方法在频率规划中的应用, 电子科学学刊, 1999, 21
- [14] 贺志超, 王荣, 机械化部队频谱资源管理与分配系统的设计, 装甲兵工程学院学报, 2000, 14(3)
- [15] 张霖斌, 姚振兴, 纪晨, 张中杰, 快速模拟退火算法及应用, 石油地球物理勘探,

1997,32(5)

- [16] 孙能利, 侯学群, 利用自动频率分配算法规划 GSM 网络, 山东师大学报 (自然科学版), 2001,16(4)
- [17] 彭颢, 移动通信网络中频率规划与优化的研究, 通信世界
- [18] 刘维国, 张大禹, 电磁兼容环境下舰艇通信频率指配研究, 舰船电子对抗, 2002,25(6)
- [19] 郎健敏, 蒋铃鸽, 诸鸿文, 周建成, 慢跳频 GSM 网络中一种基于专用频段的动态频率分配方案, 上海交通大学学报, 2001,35(11)
- [20] 戴美泰, 毛叶琴, 吴志忠, 小区制移动通信频率分配的优化调整, 邮电设计技术, 1996,12
- [21] 杜青, 图的边着色与频率分配, 南京工程学院学报, 2002,2(2)
- [22] 高建文, 李丹镒, 通信网频率分配算法设计, 无线电通信技术, 1999,25(5)
- [23] 王跃宣, 吴澄, 胡昔祥, 牟盛静, 刘连臣, 求解约束优化问题的退火遗传算法, 高技术通讯, 2004.7
- [24] 郭梯云, 邬国扬, 李建东, 移动通信, 西安电子科技大学出版社
- [25] 张业荣, 竺南直, 程勇, 蜂窝移动通信网络规划与优化, 电子工业出版社
- [26] 王凌, 智能优化算法及应用, 清华大学出版社
- [27] 王小平, 曹立明, 遗传算法理论、应用及软件实现, 西安交通大学出版社
- [28] 美]William C.Y.Lee 著, 尹浩, 李卫东, 王诚国, 郑岚, 郑浩译, 移动蜂窝通信模拟和数字系统 (第二版), 电子工业出版社
- [29] 刑文训, 谢金星, 现代优化计算方法, 清华大学出版社
- [30] 康立山, 谢云, 罗祖华, 非数值并行算法 (第一册) 模拟退火算法, 科学出版社, 2000,6
- [31] 刘勇, 康立山, 陈毓屏, 非数值并行算法 (第二册) ——遗传算法, 科学出版社, 1998