

## 摘 要

本文主要研究了共形相控阵天线的波束赋形技术。采用两种新颖的智能算法,编写程序对阵列的关键参数进行优化,得到了满意的赋形结果。

首先,研究了两种优化算法——粒子群优化算法(Particle Swarm Optimization, PSO)和遗传算法(Genetic Algorithms, GAs)。计算了两种常见的天线阵列,通过与文献中的结果进行比较,验证了算法的正确性。比较两种算法,发现PSO更适合于该类问题的计算。

其次,应用PSO算法,对多种情况进行反复计算,确定了一个满足工程项目要求的波束赋形方案。理论上实现了用较少的天线单元,且与小直径圆柱面共形的阵列,激励产生副瓣电平较低、主瓣有一定的宽度且顶部较平坦的波束。

最后,设计了天线阵列的馈电网络,介绍了功分器的原理,分别完成了等功分和功分比为2:1的一分二微带功分器的仿真。其中等功分的功分器已经加工、调试完毕,得到的测量结果,满足输入阻抗频带宽、各端口驻波系数小、插入损耗低、输出端口之间隔离度大的要求。

关键词: 共形相控阵列 波束赋形 粒子群算法 遗传算法 功分器

## ABSTRACT

In this thesis, the research of beam form synthesis of conformal phased array is described. Two novel optimizations are introduced, using MATLAB to program and optimize some key parameters of conformal phased array, and the beam form is satisfying.

First, the principle of Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs) are investigated. Two kinds of familiar arrays are calculated, the results are as good as reference literature, the arithmetic are correct. Comparing with GA, PSO is better.

Second, apply PSO to calculate many arrays and get the required scheme according to a project. Theoretically, a column conformal array with small size and few elements to generate a shaped beam with low side lobe and main beam with fixed width and flat top is carried out.

Finally, the theory of microstrip power divider is discussed. The equal and 2:1 unequal double-way power divider are simulated and the equal one is produced. The test data is good, wide bandwidth, small return loss, small insertion loss and excellent isolation, are obtained.

**Keywords:** Conformal phased array   Shaped beam   Particle swarm optimization  
Genetic algorithms   Power divider

## 创新性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技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做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：刘瑞斌

日期 2006.2.27

## 关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。本人保证毕业后离校后，发表论文或使用论文工作成果时署各单位仍然为西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密的论文在解密后遵守此规定）

本人签名：刘瑞斌

日期 2006.2.27

导师签名：郭卫平

日期 2006.2.27

## 第一章 绪论

### § 1.1 研究背景

相控阵天线是从阵列天线发展起来的,早在 20 世纪 30 年代后期就已经出现。其基本原理是基于阵列天线的分析,阵列由多个天线单元组成,每个天线单元具有近似的无方向性辐射方向图,天线单元增益较低,在按一定的规则排列在一起,形成阵列后,可获得较大的天线增益<sup>[1]</sup>。相控阵天线技术主要包括有源相控阵天线技术、数字波束形成技术、共形相控阵技术、宽带相控阵技术及低和超低副瓣相控阵天线技术等。相控阵天线主要包括辐射单元、馈电网络、移相器等几个部分。

一般的相控阵天线是直线阵或者平面阵,而共形相控阵天线是一种特殊形式的相控阵天线,它是辐射单元排列在曲线上或曲面上的天线阵。通常的共形阵天线有环形阵、圆环阵、圆锥阵、圆柱阵、半球面阵等。由于共形相控阵天线的结构特点,共形相控阵天线除了具有一般相控阵天线的优点以外,还能实现一般相控阵天线不能实现的功能:可以克服线阵和平面阵扫描角度小的缺点,易于扩展天线波束的扫描范围;在天线扫描过程中能维持或基本维持天线波束的形状和天线增益,可大体上保持相同的互耦;能以一部天线实现全空域电扫等。

随着通信系统在容量和质量上的不断升级,人们对通讯天线提出了越来越高的性能指标要求。天线工作在复杂的传播环境下,信道通常受到地形、温度、湿度等环境因素的影响,电波在空中传播时将受到多方面的衰落,这些都会对通信质量产生不利的影响。因此必须采取必要的技术手段和方法来改善通信质量。充分挖掘天线硬件的全部潜力,形成满足系统要求的天线系统。这就对天线阵的设计提出了更高的要求。对于不同的通讯需求,需要不同形状的波束进行覆盖,因此共形相控阵天线波束赋形技术将日益凸显它的魅力。

### § 1.2 研究的历史和现状

根据系统所要求的天线指标和波束形状求解阵列天线单元的激励幅值、相位、单元间距的过程称为阵列综合。阵列天线综合在天线领域一直都是一个比较重要的课题,早期提出的天线阵列方向图综合方法都是针对某一个特定问题而提出的,如实现切比雪夫方向图的方法,泰勒综合方法,伍德福德方法,贝利斯方法等。

随着对任意阵列方向图综合问题的深入,产生了如利用加权最小二乘算法进行阵列天线综合,基于自适应理论的阵列天线综合等。近几年,人们开始利用多种智能随机算法进行阵列天线综合设计,如:遗传算法(Genetic Algorithms,GA)是在上世纪70年代被提出的,它将生物进化原理应用于优化、搜索技术的过程中,是一种模拟生物进化过程和基于统计随机理论的组合算法。遗传算法只需利用目标的取值而无需梯度等高价值信息,程序通用、稳健性强、适于并行处理,尤其适用于处理传统搜索方法难于解决的复杂和非线性问题<sup>[2]</sup>。遗传算法已经被许多学者成功的运用到了天线领域,用它解决了许多阵列天线综合的问题,如:用于简单的直线阵<sup>[3]</sup>,唯幅度控制等间距线阵的零点生成<sup>[4]</sup>,良导体圆柱轴向线阵的波束赋形<sup>[5]</sup>等。粒子群优化算法<sup>[6]</sup>(Particle Swarm Optimization,PSO)是在上世纪末开发出来的另一种随机智能搜索算法,其思想来源于对一个简化社会模型的模拟。具有既适合科学研究,又特别适合工程应用的特点。因此,引起了演化计算等领域的学者们的广泛关注,并在短短的几年时间里出现大量的研究成果,形成了一个研究热点<sup>[7]、[8]</sup>。在阵列天线领域也有了许多成功的应用,如:多波束形成<sup>[9]</sup>,基站天线阵列波束赋形<sup>[10]</sup>,旁瓣零陷的生成<sup>[11]</sup>。

### § 1.3 论文的主要工作和内容安排

本文研究了遗传算法和粒子群优化算法及其在共形相控阵天线波束赋形问题中的应用。下面就本文各章的内容作一个简要介绍:

第一章,概述了论文的选题背景及意义,叙述了阵列天线综合技术的发展过程,以及共形相控阵天线综合的研究意义。

第二章,简要介绍了相控阵天线的特点及工作原理,并简单介绍了共形阵天线方向图的分析方法。

第三章,介绍了遗传算法(GA),讨论了算法的原理,给出了参数的具体设置,通过对文献中实例的计算,验证了该算法的有效性。

第四章,介绍了粒子群优化算法(PSO),详细讨论了算法的原理和参数的设置对优化效率的影响,通过对上一章中两个实例的计算,相互对比、互相验证,更好的掌握了这两种算法的应用,同时也发现PSO更适合于在阵列综合问题中应用。根据项目的要求:设计一个使用较少天线单元、具有较小尺寸且与圆柱面共形的阵列,产生较低副瓣、主波束有一定的宽度和顶部平坦的波束。应用粒子群算法经过对多种方案的多次优化,确定了一种可实现的方案。

第五章,设计了阵列的馈电网络。介绍功分器的原理,完成了等功分一分二微带功分器的设计、加工、调试。得到的测量结果,满足输入阻抗频带宽、各端

口驻波系数小、插入损耗低、输出端口间隔离度大的指标。设计了功分比为 2: 1 的一分二功分器，给出了满足需要的仿真结果。

在结束语中，回顾了本文的工作，并就后续的研究工作以及预期目标谈了一些看法。

## 第二章 共形相控阵天线

### § 2.1 相控阵天线原理及其特点

为了加强天线的方向性,由若干个辐射单元按某种方式排列而成的天线系统,称为阵列天线。按天线单元的排列方式,阵列天线可分为直线阵,平面阵和立体阵。

阵列天线的方向性理论包括两方面内容:一是已知天线单元的排列方式、天线单元的个数、间距和各天线单元的电流(幅度和相位)分布规律,分析天线阵的方向性,称为阵列天线方向性分析;二是根据预定的天线方向图,寻求能形成该方向图的天线阵的参数:天线单元个数、间距和各天线元电流分布规律等,称为阵列天线方向性综合。

相控阵的样式很多,为了减小线性尺寸,实现圆对称波束,提高增益,常使用平面阵;为了减小副瓣电平,实现波扫描时的恒定,常常使用加权相控阵等<sup>[12]</sup>。由于存在单元互耦,天线单元在阵列中的辐射特性与其本身单独工作时的辐射特性不尽相同。这里,讨论忽略单元间互耦影响情况下的天线阵辐射特性。下面先简单介绍均匀线阵的基本原理再对面阵进行简要说明。

均匀线阵方向图函数为:

$$f_1(\theta) = \frac{\sin \frac{n}{2}(kd \sin \theta - \phi)}{\sin \frac{1}{2}(kd \sin \theta - \phi)} \quad (2-1)$$

式中,  $k = 2\pi/\lambda$ ,  $d$  表示单元间距。

最大值出现在  $kd \sin \theta_m - \phi = 0$ , 即:

$$\sin \theta_m = \frac{\phi \lambda}{2\pi d} \quad (2-2)$$

由式(2-2)可知:

(1) 对于固定频率( $\lambda$ 不变),若移相器的移相量 $\phi$ 发生变化,阵的最大值就会发生变化,这就是所谓的波束扫描。

(2) 若想避免栅瓣的出现,必须满足:

$$d \leq \frac{\lambda}{1 + |\sin \theta_s|} \quad (2-3)$$

式中,  $\theta_s$  为相控阵天线最大扫描角度。

相控阵天线与其它天线相比有以下优点:

(1) 天线波束扫描不需要笨重的机械转动装置和旋转空间,改善了波束的

稳定性, 天线系统体积小、重量轻。

(2) 天线扫描无惯性, 波束控制灵活, 波束能在很短时间 (几  $\mu\text{s}$  内) 指向预定空间, 从而大大节约了扫描时间。

(3) 波束控制灵活, 能用同一天线形成多个波束搜索跟踪和干扰多个目标以适应密集信号环境。

(4) 多个天线单元, 使天线波束能量集中, 易于实现高增益。

(5) 故障弱化。由于相控阵是由很多阵元形成的综合效果, 即使个别阵元出现故障, 也不会影响它的正常工作。

典型的平面阵<sup>[13]</sup>是由  $n = n_x \times n_y$  个相同天线单元组成的矩形平面阵, 沿  $x$  轴和  $y$  轴排列的天线单元的间距分别为  $d_x$  和  $d_y$ , 如图 2.1 所示。

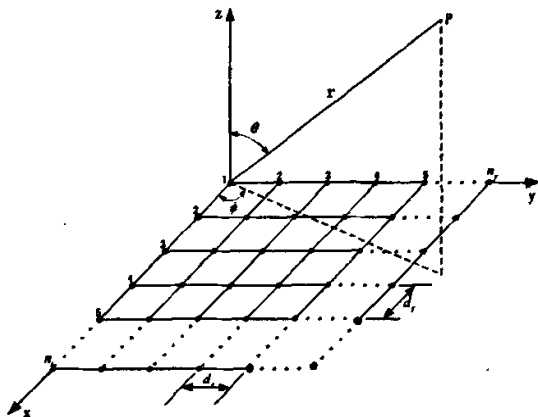


图 2.1 矩形栅格平面阵

由方向图乘积定理可得平面阵列的方向函数

$$|f_s(\theta, \phi)| = |f_1(\theta, \phi)| \times |f_{ax}(\theta, \phi)| \times |f_{ay}(\theta, \phi)| \quad (2-4)$$

式中,  $|f_1(\theta, \phi)|$  表示天线单元方向函数, 即单元因子。

$|f_{ax}(\theta, \phi)|$  ——表示  $x$  阵的阵因子;

$|f_{ay}(\theta, \phi)|$  ——表示  $y$  阵的阵因子。

等幅平面阵的  $x$  阵和  $y$  阵方向函数为

$$|f_{ax}(\theta, \phi)| = \left| \frac{\sin(\frac{n_x}{2} \psi_x)}{\sin(\frac{1}{2} \psi_x)} e^{j\frac{n_x-1}{2} \psi_x} \right| = \left| \frac{\sin(\frac{n_x}{2} \psi_x)}{\sin(\frac{1}{2} \psi_x)} \right| \quad (2-5)$$

$$|f_{ay}(\theta, \phi)| = \left| \frac{\sin(\frac{n_y}{2} \psi_y)}{\sin(\frac{1}{2} \psi_y)} e^{j\frac{n_y-1}{2} \psi_y} \right| = \left| \frac{\sin(\frac{n_y}{2} \psi_y)}{\sin(\frac{1}{2} \psi_y)} \right| \quad (2-6)$$



式中

$$\psi_x = kd_x \sin \theta \cos \phi + \beta_x \quad (2-7)$$

$$\psi_y = kd_y \sin \theta \sin \phi + \beta_y \quad (2-8)$$

$$\xi_x = (n_x - 1)\psi_x / 2 \quad (2-9)$$

$$\xi_y = (n_y - 1)\psi_y / 2 \quad (2-10)$$

理想点源等幅平面阵的归一化方向函数为

$$|F_\infty(\theta, \phi)| = \left| \frac{\sin(\frac{n_x}{2}\psi_x)}{n_x \sin(\frac{1}{2}\psi_x)} \right| \times \left| \frac{\sin(\frac{n_y}{2}\psi_y)}{n_y \sin(\frac{1}{2}\psi_y)} \right| \quad (2-11)$$

## § 2.2 共形阵列天线方向图

在图 2.2 中,  $N$  个阵列天线单元安装在某一曲面上, 每一天线单元的坐标位置可用其位置矢量  $\vec{r}_i$  来表示。每一天线单元的电场强度辐射方向图分别为  $f_i(\theta, \phi)$ 。即使每一天线单元的场强方向图都相同, 但由于它们分布在曲面上, 同样  $(\theta, \phi)$  方向上的  $f_i(\theta, \phi)$  也是不相同的<sup>[14]</sup>。

设第  $i$  个天线单元的幅度与相位加权系数分别为  $a_i$  和  $\Delta\phi_{i0}$ , 即第  $i$  个天线单元的复加权系数  $W_i$  为

$$W_i = a_i e^{-j\Delta\phi_{i0}} \quad (2-12)$$

另外, 假定相位参考点选择坐标原点, 则阵列中所有天线单元在  $\vec{r}$  方向上辐

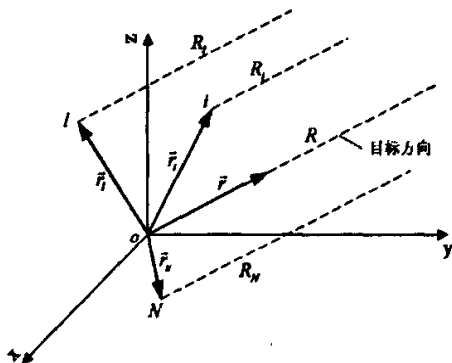


图 2.2 共形阵列天线单元位置矢量

射的合成场强或天线的场强方向图，可以表示为

$$E(\vec{r}) = E(\theta, \phi) = \sum_{i=1}^N f_i(\theta, \phi) a_i e^{-j\Delta\phi_0} \times \frac{e^{-j\frac{2\pi}{\lambda} R_i}}{R_i} \quad (2-13)$$

对于远区的目标，由于阵列相位参考点到目标距离  $R$  远大于相位参考点到各天线单元的距离  $(r_1, r_2, \dots, r_N)$ ，即  $R \gg r_i$ ，故式 (2-13) 分母中的  $R_i$  可用  $R$  代替。各天线单元到目标的距离  $R_i$  可表示为

$$R_i = R - \Delta R_i \quad (2-14)$$

去除公共相位因子，不考虑幅度的常数项，则式 (2-13) 可以改写成

$$E(\theta, \phi) = \sum_{i=1}^N f_i(\theta, \phi) a_i e^{j(\frac{2\pi}{\lambda} \Delta R_i - \Delta\phi_0)} \quad (2-15)$$

$\Delta R_i$  是第  $i$  个天线单元到目标的距离与参考点 (坐标原点) 到目标距离之差值，取决于

$$\Delta R_i = \vec{r}_i \cdot \hat{r} \quad (2-16)$$

即  $\Delta R_i$  是  $i$  个天线单元的位置矢量  $\vec{r}_i$  与参考点到目标点单位矢量  $\hat{r}$  的点积。可以用其方向余弦来表示

$$\vec{r} = \hat{i} \cos \alpha_x + \hat{j} \cos \alpha_y + \hat{k} \cos \alpha_z \quad (2-17)$$

由不同坐标系之间的变换关系，不难得到

$$\begin{cases} \cos \alpha_x = \cos \theta \cos \phi \\ \cos \alpha_y = \cos \theta \sin \phi \\ \cos \alpha_z = \sin \theta \end{cases} \quad (2-18)$$

第  $i$  个天线单元在阵面上的位置，如果以它在  $(x, y, z)$  坐标系中的投影来表示，即

$$\vec{r}_i = \hat{i}x_i + \hat{j}y_i + \hat{k}z_i \quad (2-19)$$

则根据式 (2-16)， $\Delta R_i$  可表示为

$$\Delta R_i = x_i \cos \alpha_x + y_i \cos \alpha_y + z_i \cos \alpha_z \quad (2-20)$$

$\Delta R_i$  对应的相位，即第  $i$  个单元相对于参考点 (坐标原点) 的信号相位为

$$\Delta\phi_i = \frac{2\pi}{\lambda} \Delta R_i = \frac{2\pi}{\lambda} (x_i \cos \alpha_x + y_i \cos \alpha_y + z_i \cos \alpha_z) \quad (2-21)$$

为了使天线波束的最大值在  $(\theta_0, \phi_0)$  方向上，由式 (2-15) 和式 (2-21) 可得每一个天线单元的移相器所提供的相移  $\Delta\phi_0$  必须为

$$\Delta\phi_0 = \frac{2\pi}{\lambda} (x_i \cos \alpha_{x_0} + y_i \cos \alpha_{y_0} + z_i \cos \alpha_{z_0}) \quad (2-22)$$

式中

$$\begin{cases} \cos a_{x_0} = \cos \theta_0 \cos \phi_0 \\ \cos a_{y_0} = \cos \theta_0 \sin \phi_0 \\ \cos a_{z_0} = \sin \theta_0 \end{cases} \quad (2-23)$$

显然,按式(2-22)改变每一天线单元的相移,即改变每一天线单元复加权系数中的相位项,就能使天线波束实现扫描。

这里需要指出以下两点:

(1) 由式(2-22)可见,每一天线单元移相器提供的移相值,除了与波束最大值指向位置 $(\theta_0, \phi_0)$ 或波束指向的方向余弦 $(\cos a_{x_0}, \cos a_{y_0}, \cos a_{z_0})$ 有关外,还与每一天线单元的位置 $(x_i, y_i, z_i)$ 有关。由于在共形阵上,各单元位置之间没有像前面叙述的线阵或平面阵那样简单的线性关系,故对每一天线单元,其移相器的控制信号需要单独运算,即共形阵中移相器控制信号的产生比线阵或平面阵情况要复杂一些,运算量也要大一点。

(2) 由于天线单元安装在某一个曲面上,其单元方向图最大值指向不同方向,因此,各天线单元的方向图 $f_i(\theta, \phi)$ 在某一方向上的增益是不同的。这使得共形阵天线方向图计算公式(即式(2-15))中的单元方向图因子 $f_i(\theta, \phi)$ 不能作为公因子从求和符号 $\Sigma$ 里提取出来。为了满足一定的天线波束副瓣要求,在式(2-15)中, $f_i(\theta, \phi)a_i$ 应满足一定的天线口径电流分布要求。对所有的天线单元来说,由于 $f_i(\theta, \phi)$ 是不相等的,故各天线单元的幅度加权系数 $a_i$ 应随 $(\theta, \phi)$ 的变化而变化,这样才能获得必需的单元方向图与幅度加权系数的乘积 $f_i(\theta, \phi)a_i$ 。

## 第三章 遗传算法

### § 3.1 引言

遗传算法<sup>[15]</sup> (Genetic Algorithms, 简称 GA) 是一种基于生物自然选择和基因遗传学原理的优化搜索方法。在遗传算法的创立过程中有两个研究目的: 一是抽象和严谨地解释自然界的适应过程; 二是为了将自然生物系统的重要机理运用到工程系统、计算机系统或商业系统等人工系统的设计中。遗传算法在计算机上模拟生物的进化过程和基因的操作, 并不需要对象的特定知识, 也不需要对象的搜索空间是连续可微的, 它具有全局寻优的能力。遗传算法常用于许多领域的实际问题, 如函数优化、自动控制、图像识别、机器学习等。同传统优化算法相比, 遗传算法有以下特点:

- (1) 遗传算法是对参数的编码进行操作, 而非对参数本身;
- (2) 遗传算法是从许多点开始并行操作, 而非局限于一点, 因而可以有效地防止搜索过程收敛于局部最优;
- (3) 遗传算法通过目标函数来计算适值, 而不需要其它推导和附加信息, 从而对问题的依赖性小;
- (4) 遗传算法的寻优规则是由概率决定的, 而非确定性;
- (5) 遗传算法在解空间进行高效启发式搜索, 而非盲目地穷举或完全随机搜索;
- (6) 遗传算法对需要优化的函数基本上无限制, 因而应用范围较广;
- (7) 遗传算法更适合大规模复杂问题的优化求解;
- (8) 遗传算法具有并行计算的特点, 因而可以通过大规模并行计算来提高计算速度;

### § 3.2 遗传算法

与传统的搜索法不同, 遗传算法是从一组随机产生的初始解 (称为“种群/population”) 开始搜索。种群中的每个个体是问题的一个解的编译串 (称为“个体位串/individual string”或“染色体/chromosome”), 染色体是一串符号。这些染色体在后续迭代中不断进化, 称为遗传。在每一代中用“适值 (fitness)”来测量

染色体的好坏。生成的下一代染色体，称为后代 (offspring)。后代是由前一代染色体通过遗传算法 (即交叉 (crossover) 和变异 (mutation) 运算) 形成的。经过若干代之后，算法收敛于最好的染色体，它很可能就是问题的最优解或次优解。它的工作流程如图 3.1 所示：

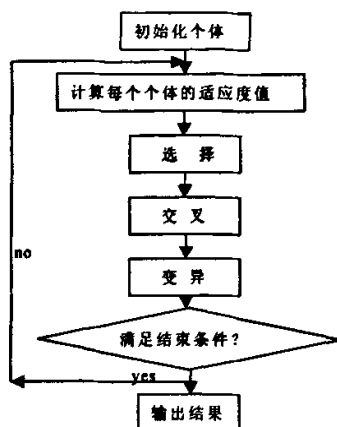


图 3.1 遗传算法工作流程示意图

复制 (又称繁殖) 是一个从旧种群 (old population) 中选择生命力强的个体位串产生新种群的过程。或者说，复制是个体位串根据其适值函数拷贝自己的过程。直观的讲，可以把适值 (或目标) 函数看作是我们期望的最大效益或好处的某种度量。根据位串的适值拷贝位串意味着，具有较高适值的位串更有可能在下一代中产生一个或多个子孙。在我们的复制操作过程中，适值函数是该位串被复制或被淘汰的决定因素。

复制操作的初始种群 (旧种群) 的生成往往是随机产生的。例如我们可以通过掷硬币 20 次，可以产生维数  $n=4$  的初始种群如下 (正面=“1”，背面=“0”)：

01101, 11000, 01000, 10011

将其编号为四个位串：

位串 1: 01101

位串 2: 11000

位串 3: 01000

位串 4: 10011

位串 1~4 可分别解码为如下十进制的数：

位串 1 = 13, 位串 2 = 24, 位串 3 = 8, 位串 4 = 19

通过一个五位无符号二进制数,我们可以得到一个从零到 31 的数值  $x$ , 它可以是系统的某个参数。取目标函数或适值  $f(x)=x^2$ , 可得计算结果如表 3.1

表 3.1 种群的初始位串及对应的适配值

编号	位串 ( $x$ )	适配值	占总数的百分比 (%)
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
总和 (初始种群整体)		1170	100.0

表 3.2 复制操作之后的各项数据

位串号	随机生成的 初始种群	$x$ 值 (无符号 数)	$f(x)=x^2$	选择复制 的概率 $f_i/\sum f_i$	期望的复 制数 $f_i/\bar{f}_i$	实际得到的 复制数
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
总计			1170	1.00	4.00	4
平均值			293	0.25	1.00	1
最大值			576	0.49	1.97	2

简单的交叉操作分两步实现。在由等待配对的位串构成的匹配池中, 第一步是将新复制产生的位串个体随机两两配对, 第二步是随机地选择交叉点, 对匹配的位串进行交叉繁殖, 产生一个新的位串。具体过程如下:

设位串的字符长度为  $l$ , 在  $[1, l-1]$  的范围内, 随机的选取一个整数值  $k$  作为交叉点。将两个配对位串从位置  $k$  后的所有字符进行交换, 从而生成两个新的位串。列在表 3.3 中。

表 3.3 交叉操作之后的各项数据

新串号	复制后的匹 配池 (“:” 为交叉点)	配对对象 (随机选 择)	交叉点 (随机 选择)	新群体	$x$ 值	$f(x)=x^2$
1	0110 : 1	2	4	01100	12	144
2	1100 : 0	1	4	11001	25	625
3	11 : 000	4	2	11011	27	729
4	10 : 011	3	2	10000	16	256
总计						1754
平均值						439
最大值						729

表 3.3 中, 位串 1 和位串 2 配对, 位串 3 和位串 4 配对, 然后随机选取交叉点, 位串 1 和位串 2 交叉点为  $k=4$ , 二者只交换最后一位, 从而生成两个新位串, 位串 3 和位串 4 交叉点为  $k=2$ , 二者交换后三位, 结果生成两个新位串。

尽管复制和交叉操作很重要, 在遗传算法中是第一位, 但不能保证不会遗漏一些重要的信息。在人工遗传系统中, 变异用来防止这种不可弥补的遗漏。在简单遗传算法中, 变异就是某个字符串某一位的值偶然的 (概率很小) 随机的改变, 即在某些特定的位置上简单地把 1 变成 0, 或反之。变异是沿着位串字符空间的随机移动。当它有节制地和交叉一起使用时, 它就是一种防止过度成熟而丢失重要概念的保险策略。例表 3.4 所示群种中, 无论怎样交叉, 在位置 4 上不可能得到有 1 的位串。

表 3.4 随机群种

编号	位串	适配值
1	01101	169
2	11001	625
3	00101	25
4	11100	784

变异在遗传算法中的作用是第二位的, 但却是必不可少的。变异操作可以起到恢复位串字符多样性的作用, 并能适当地提高遗传算法的搜索效率。通常情况下, 变异的概率为每一个千位的传送中, 只变异一位, 即变异的概率为 0.001。在上面介绍的种群中, 共有 20 个串位符号, 期望的变异串位数为  $20 \times 0.001 = 0.02(\text{位})$ 。

例: 求使函数  $f(x) = x^2$  在  $[0, 31]$  上取得最大值的点。

求解的步骤为:

- (1) 在区间  $[0, 31]$  上的整参数  $x$  可用一个五位的二进制位串进行编码;
- (2) 按抛硬币的方法随机产生一个由 4 个位串组成的初始种群 (参见表 3.1);
- (3) 计算适值和选择率, 计算结果见表 3.1;
- (4) 复制。操作结果见表 3.2;
- (5) 交叉。计算结果见表 3.3;
- (6) 变异。位串变化为  $20 \times 0.001 = 0.2(\text{位})$ , 因此未有变异发生;
- (7) 对比。平均值:  $293 \rightarrow 437$ ; 最大值:  $576 \rightarrow 729$

基本遗传算法使用固定长度的二进制符号串来表示群体中的个体, 其等位基

因是由二值符号集 $\{0, 1\}$ 所组成的。初始群体中各个个体的基因值可用均匀分布的随机数来生成。如： $X=100111001000101101$  就可表示一个个体，该个体的染色体长度是 $n=18$ 。按与个体适应度成正比的概率来决定当前群体中每个个体遗传到下一代群体中的机会多少。为正确计算这个概率，这里要求所有个体的适应度必须为正数或零。这样，根据不同种类的问题，必须预先确定好由目标函数值到个体适应度之间的转换规则，特别是要预先确定好当目标函数值为负数时的处理方法。复制运算使用比例复制算子、交叉运算使用单点交叉算子、变异运算使用基本位变异算子或均匀变异算子。群体大小，即群体中所含个体的数量，一般取为20~100。遗传运算的终止进化代数，一般取为100~1000。交叉概率，一般取为0.40~0.99。变异概率，一般取为0.001~0.100。

### § 3.3 改进的遗传算法

一，交叉运算，它同时对两个染色体操作，组合二者的特性产生新的后代。交叉的最简单方法是在双亲的染色体上随机地选一个断点，将断点的右端互相交换，从而形成两个新的后代。遗传算法的性能在很大程度上取决于采用的交叉运算的性能。

交叉率：定义为各代中交叉产生后代数与种群中的个体数之比。较高的交叉率可达到更大的解空间，从而降低停在非最优解的机会；通过交叉，能够产生遗传后代，同时也会产生非法染色体。以下介绍部分交叉算子：

#### (1) 部分映射交叉 (PMX/partially matched crossover operator)

PMX 可看作二进制串的两点交叉在换位表达中的扩展，它用特别的修复程序来解决简单两点交叉引起的非法性，它的运算步骤如下：

- (a) 在双亲字串上随机地选择两点，由这两点定义的子串称为映射段；
- (b) 交换双亲的映射段，产生原始后代；
- (c) 确定两映射段之间的映射关系；
- (d) 根据映射关系将后代合法化。

#### (2) 顺序交叉 (OX/order crossover operator)

OX 可将其看作是一种带有不同修复程序的 PMX 的变型。步骤如下：

- (a) 从第一双亲中随机选一个子串；
- (b) 将子串复制到一个空字符串的相应位置，产生一个原始后代；



(c) 删去第二代双亲中子串已有的基因,得到原始后代需要的其它基因的顺序;

(d) 按照这个基因顺序,从左到右将这些基因定位到后代的空间位置上。

### (3) 循环交叉 (CX/cycle crossover)

该方法从一个双亲中取一些基因,而其它基因则取自另一个双亲。不同的是来自第一个双亲的基因不是随机产生的,而是根据两个双亲相应位置基因构成的环确定的。步骤如下;

(a) 根据双亲相应的基因位置找出一个循环;

(b) 把一个双亲的循环上的基因复制到一个后代上;

(c) 删去另一个双亲的已在循环上的基因,剩下的基因既可用于来确定剩余基因的位置。

二,变异是一种基本运算,它在染色体上自发地产生随机的变化。一种简单的变异方法是替换一个或多个基因。在遗传算法中,变异可以提供初试群种中不含有的基因或选择过程中丢失的基因,为群种提供新的内容。

变异率定义为群种中变异基因数在总基因数中的百分比。对于换位表达的变异运算相对比较容易。过去十多年来,曾提出集中用于换位表达的变异运算。如下所述;

(1) 反转变异:在染色体上随机地选择两点,将两点间的子串反转;

(2) 插入变异:随机地选择一个基因,并将它插入到一个随机的位置中;

(3) 移位变异:随机地选择一个子巡回,并将它插入到一个随机的位置中;

(4) 互换变异:随机的选择两个位置,并将这两个位置上的基因相互交换。

三,遗传算法的基本原理就是达尔文的自然选择原理,选择是遗传算法的推动力。选择压力是一个内含的准则,压力过大搜索会过早终止,压力过小搜索又会不必要的缓慢。选择包括三个基本方面:

#### (1) 采样空间

采样空间由大小和成分(双亲或后代)两个因素来确定。规则的采样空间的大小等于群种规模,含有所有后代和部分双亲。规则的采样空间原则是,新的种群是由孩子替代掉双亲形成的。而扩大的采样空间的大小等于种群规模与每代产生

的后代数，包括所有后代和双亲。这种扩大的采样空间显著优点是可以增加交叉率和变异率来改进遗传算法的性能。

## (2) 采样机理

采样机理是关于如何从采样空间中选择染色体的理论，选择染色体的方法有随机采样、确定采样和混合采样。

随机采样的特点是在选择阶段根据生存率来确定每个染色体的实际繁殖数。而选择阶段由确定染色体和期望值转换为后代两部分构成。染色体的期望值是表明该染色体应产生的后代的平均数的实数，采样过程则将这实数转换为后代数。随机采样方法的基本思想是保持每个染色体下一代的期望繁殖数。

确定采样是从采样空间中选择 pop-size（种群规模）个体最好的染色体。

混合采样同时具有随机采样和确定采样的特征，典型的例子是竞赛选择。该方法是随机地选出一组染色体，从中选出最好的一个进行繁殖，每组的染色体个数称为竞赛人数，通常竞赛人数为 2。

## (3) 选择概率

选择概率的方法有三种，即正比选择法、标定法和排序法。

正比选择法是选择概率正比于染色体的适值；

标定法是将原目标函数映射为某个实数值，然后用这些实数值来确定每个染色体的生存概率；

排序法则忽略实际目标函数值，而用染色体的顺序来替代生存概率。

适值的标定具有的作用是：使染色体之间的适值比保持合理的差距，阻止某些超级染色体太快的把持遗传过程，以满足早期限制竞争、晚期鼓励竞争的需要。

§ 3.4 应用遗传算法的波束赋形实例

为了验证遗传算法在阵列天线波束赋形问题中的有效性，下面分别对两种比较常见的阵列的波束进行赋形计算。

例一 综合直线阵扇形波束方向图

用十个各向同性的天线单元组成直线阵，单元间距d均为半个波长，不计单元间的互耦。要综合出的扇形波束为： $F_{desired}(\theta)=\begin{cases} 1, -\pi/4 < \theta < \pi/4 \\ 0, \text{其余值} \end{cases}$ 。对于这种阵列

形式，当激励幅度为 $I_n$ ，激励相位为 $\beta_n$ 时，其辐射方向图为 $F(\theta)=\sum_{n=1}^{10} I_n e^{j(nkd \sin(\theta)+\beta_n)}$ ，

(在本例中 $\beta_n=0$ 或 $\pi$ ， $n=1\sim 10$ )。

考虑到阵列的对称性，只优化1到5号天线单元的激励幅度和相位。种群个数设为100，采用二进制编码，基因设定为45位的二进制数，进化200代。

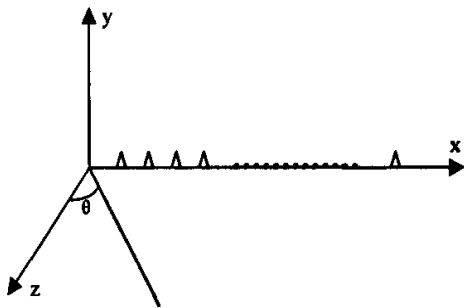


图3.2 阵列示意图

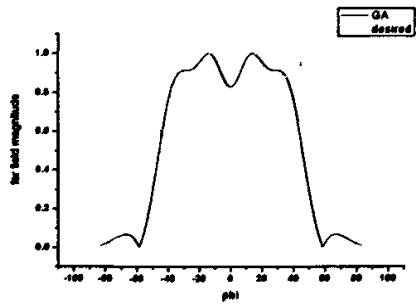


图3.3 扇形方向图

表3.5 GA综合扇形方向图的天线单元幅度和相位

阵元 序数	1	2	3	4	5	6	7	8	9	10
激 励 幅 度	0.0784	0.0784	0.1216	0.0784	0.9176	0.9176	0.0784	0.1216	0.0784	0.0784
激 励 相 位	$\pi$	0	$\pi$	$\pi$	0	0	$\pi$	$\pi$	0	$\pi$

由图 3.3 可以看出，优化得到了符合要求的扇行方向图。表 3.5 给出了相应的激励幅度和相位的优化结果。

## 例二 综合一个圆环阵

阵列由 31 个各向同性的单元组成，单元分布在半径为 6 个波长的一段圆环上，间距为半个波长。要求：主瓣波束宽度为  $30^\circ$ ，副瓣电平小于  $-35\text{dB}$ 。

由于阵列较大，需优化的变量较多，为了减少优化变量的个数，引入了修正的伯恩斯坦多项式<sup>[16]</sup>，如式 (3-1)

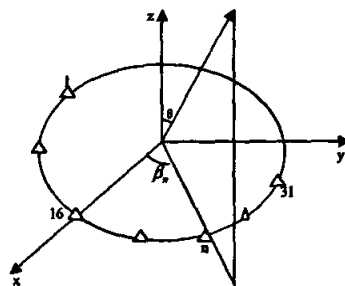


图 3.4 31 元阵列示意图

$$F(U) = \begin{cases} B1 + \frac{1-B1}{A^{MA}(1-A)^{M(1-A)}} U^{MA}(1-U)^{M(1-A)}, & 0 \leq U \leq A \\ B2 + \frac{1-B2}{A^{MA}(1-A)^{M(1-A)}} U^{MA}(1-U)^{M(1-A)}, & A \leq U \leq 1 \end{cases} \quad (3-1)$$

其中， $B1, B2, M, A$  是多项式中的四个参数。多项式的单峰值出现在  $U=A$  处， $B1, B2$  分别确定多项式两个端点的值  $F(0)$ 、 $F(1)$ 。增大  $M$  的值可以使多项式的形状变得陡峭，减小  $M$  的值多项式的形状变平坦。

图 3.5 为  $B1=0.1, B2=0.2, N$  分别取 5 和 20 时修正波恩斯坦多项式的形状

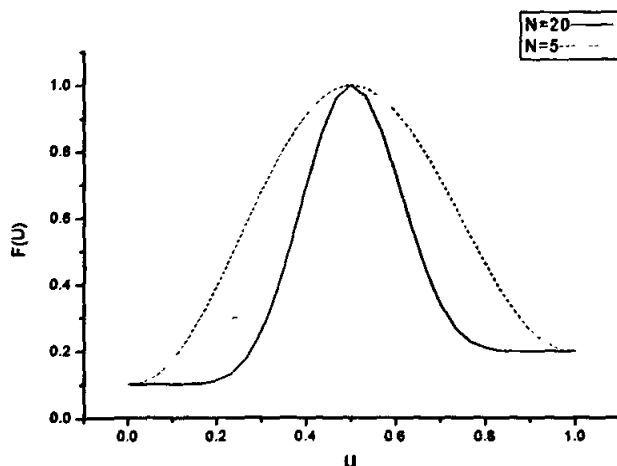


图 3.5 波恩斯坦多项式形状

$\theta = 90^\circ$  时的方向图函数为：

$$E(\theta, \phi) = \sum_{n=1}^{31} F(U)_n * \exp(j * k * r * (\cos(\phi - \beta_n) - \cos(\beta_n))) \quad (3-2)$$

其中,  $F(U)_n$  是第  $n$  个单元的激励幅度, 是  $F(U)$  的第  $n$  个等分采样点的值,  $\beta_n$  是第  $n$  个单元的相位补偿 (以阵列中心为相位参考点),  $r$  是阵列的半径,  $k$  是波数。

通过对  $M, A, B1, B2$  四个参数的适当选择, 可以得到合适的  $F(U)$  并在其上采样 31 个点, 采样点的值作为对应天线单元的激励幅度值, 进而得到阵列的低副瓣、窄波束的方向图。选取适当的适应度函数, 运用 GA 算法只需对  $M, A, B1, B2$  四个变量进行优化 ( $A, B1, B2$  变化范围是  $0 \sim 1$ ,  $M$  的变化范围是  $1 \sim 20$ )。使主瓣波束宽度小于  $30^\circ$ , 副瓣电平低于  $-35\text{dB}$ 。对于 GA, 种群个数设为 100, 基因设为 40 位的二进制数 ( $A, B1, B2, M$  分别用八位二进制数表示), 进化代数设定为 100 代。图 3.6 给出了运算结果, 方向图达到了优化要求。与参考文献中的类似阵列的优化结果相比较, 可以证明计算结果的正确性。

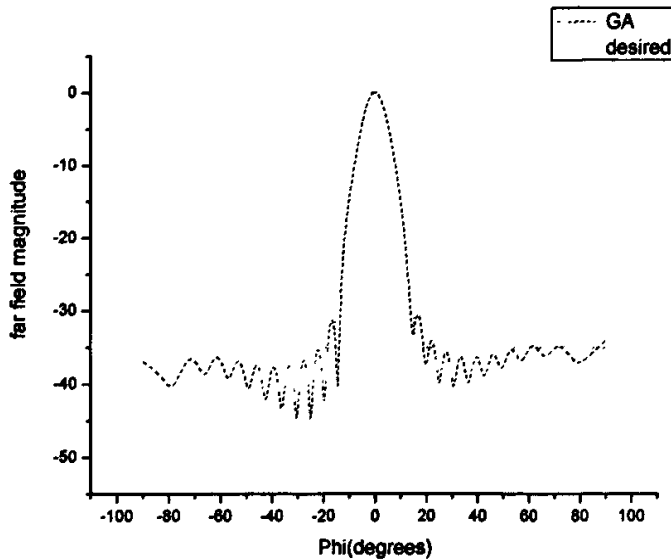


图 3.6 方向图优化结果

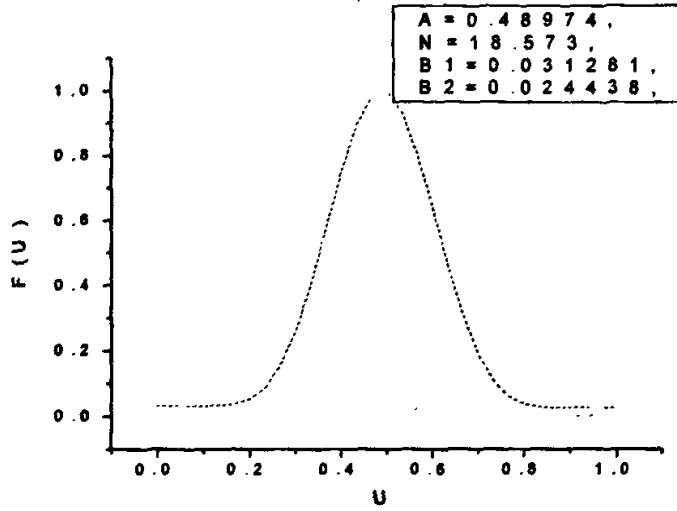


图 3.7 单元激励幅度分布

由于采用了波恩斯坦多项式,大大减少了优化参数(只有4个),提高了优化速度,但是由于它是单峰值函数,在优化中只得到了锐波束方向图。

## 第四章 粒子群优化算法

### § 4.1 引言

当前,通过模拟生物群体的行为来解决计算问题已经成为新的研究热点,形成了以群体智能(Swarm Intelligence)为核心的理论体系,并已在一些实际应用领域取得突破性进展。通过对生物群体的观察和研究发现,生物群体内个体间的合作与竞争等复杂行为产生的群体智能往往能对某些特定的问题提供高效的解决方法。粒子群优化算法<sup>[17]</sup> (PSO, Particle Swarm Optimization)作为一种新的随机智能搜索算法,其思想来源于对一个简化社会模型的模拟。其中“群(swarm)”来源于微粒群符合Millonas在开发应用于人工生命(artificial life)的模型时所提出的群体智能的5个基本原则。而“微粒(particle)”则是一个折衷的选择,因为既需要将群体中的成员描述为没有质量、没有体积的,同时也需要描述它的速度和加速状态。由于PSO算法概念简单,实现容易,短短几年时间,PSO算法便获得了很大的发展,并在一些领域得到应用<sup>[18]~[20]</sup>。目前已被“国际演化计算会议”(CEC)列为讨论专题之一,许多学者仍在对粒子群算法做着种种改进,使它的理论更完善,功能更强大。

本章首先介绍了基本的PSO算法,并应用该算法优化上一章中的两个阵列实例,得到了正确的结果。通过比较发现PSO比GA的优化结果更好且程序移植性好、寻优速度更快,更适合于在阵列综合问题中的应用。在本章的后半部分,选择用PSO算法解决一个项目中的共形相控阵天线的波束赋形问题,编写程序反复计算多种阵列形式,最终得到了满意的结果,确定了解决方案。

### § 4.2 基本粒子群优化算法

PSO算法与其它演化算法相似,也是一种基于群体的优化算法。设想这样一个场景:有一群蜜蜂,它们的任务是在一个区域里寻找花蜜最多的花丛,所有的蜜蜂都不知花丛的具体位置。每只蜜蜂都只是从一个随机的位置,以一个随机的速度开始寻找花丛,而且每一只蜜蜂都有记忆它自己和整个蜂群所经历最好花群地点的能力。那么,找到花丛的最优策略就是每只蜜蜂都根据某种原则不断的改变飞行方向,直到找到花蜜最多的花丛。PSO算法从这种模型中得到启示并用于解决优化问题。PSO算法中,每个优化问题的潜在解都是搜索空间中的一只蜜蜂,称之为“粒子”。所有的粒子都有一个由被优化的函数决定的适应度值,并且每

个粒子还有一个速度决定它们飞翔的方向和距离。PSO 算法首先初始化一群随机粒子（随机解），在搜索空间中以一定的速度飞行，然后通过迭代找到最优解。在每一次迭代中，粒子通过跟踪两个极值来更新自己，第一个就是粒子本身所找到的最优解，另一个极值是整个种群目前找到的最优解。

设第 $i$ 个粒子用 $X_i=(x_{i1}, x_{i2}, \dots, x_{id})$ 来表示，它经历过的最好位置用 $P_{best}=(p_{i1}, p_{i2}, \dots, p_{id})$ 表示。而群体所有粒子经历过的最好位置用 $g_{best}=(g_1, g_2, \dots, g_d)$ 表示。粒子 $i$ 的速度用 $V_i=(v_{i1}, v_{i2}, \dots, v_{id})$ 表示。对每一代个体，其第 $d$ 维( $1 \leq d \leq D$ )的速度和位置根据如下方程变化：

$$v_{id} = \omega * v_{id} + c1 * \text{rand}() * (p_{id} - x_{id}) + c2 * \text{rand}() * (g_d - x_{id}) \quad (4-1)$$

$$x_{id} = x_{id} + v_{id} \quad (4-2)$$

其中： $\omega$ 为惯性权重(inertia weight),  $c1$ 和 $c2$ 为加速常数(acceleration constants),  $\text{rand}()$ 和 $\text{rand}()$ 为两个在 $[0, 1]$ 范围内变化的随机函数。此外，粒子的速度 $v_i$ 被一个最大速度 $v_{\max}$ 所限制。如果当前对粒子的加速导致它在某维的速度 $v_{id}$ 超过该维的最大速度 $v_{\max}$ ，则该维的速度被限制为该最大速度 $v_{\max}$ 。

式(4.1)的第1部分为微粒先前的速度；第2部分为“认知(cognition)”部分，表示微粒本身的思考；第3部分为“社会(social)”部分，表示微粒间的信息共享与相互合作，“认知”部分可由Thorndike的“影响法则(law of effect)”解释，即一个得到加强的随机行为在将来更有可能出现。这里的行为即“认知”，并假设获得正确的知识是得到加强的，这样一个模型假定微粒被激励着去减小误差。“社会”部分可由Bandura的代理(vicarious)加强概念的解释。根据该理论的预期，当观察者观察到一个模型在加强某一行为时，将增加它实行该行为的几率，即微粒本身的认知将被其它微粒所模仿。PSO算法的这些心理学假设是无争议的。在寻求一致的认知过程中，个体往往记住它们的信念，同时考虑伙伴的信念。当个体察觉伙伴的信念较好时，它将进行适应性地调整。标准PSO算法流程如图4.1

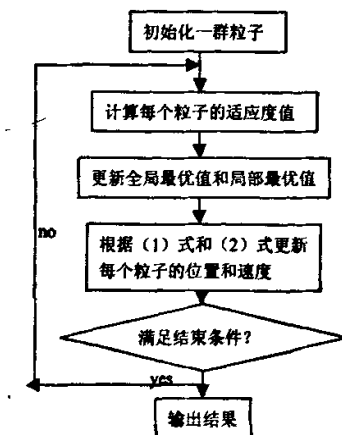


图 4.1 PSO算法框图



标准PSO算法的工作流程如下:

- 第一步: 初始化一群粒子(群体规模为swarm size)包括起始位置和速度;
- 第二步: 计算每个粒子的适应度值;
- 第三步: 对每个粒子, 将其适应度值与其经历过的最好位置 $p_{best}$ 作比较, 如果好于后者, 则将此时的适应度值作为当前的最好位置 $p_{best}$ ;
- 第四步: 对每个粒子, 将其适应度值与全局所经历的最好位置 $g_{best}$ 作比较, 如果好于后者, 则重新记录 $g_{best}$ 的大小;
- 第五步: 先根据方程(4.1)重新计算粒子的速度, 然后根据方程(4.2)重新计算粒子位置;
- 第六步: 如果满足结束条件(通常为产生足够好的适应度值或达到一个预设最大代数 $\max G$ ), 程序终止, 否则跳转到第二步。

以上部分就是标准PSO算法的流程, 随着人们对算法的不断完善, 在基于标准的PSO算法模型的基础上又提出了几种改进的模型<sup>[23], [24], [25]</sup>, 例如: 混合PSO(HPSO)模型, 带有邻域操作的PSO模型, 采用拉伸(Stretching technique)的PSO模型, 并且分别在一些特殊的优化问题中得到了成功的运用, 但在天线阵综合的运用中我们选择了标准的PSO模型, 对其它模型有兴趣的读者可以参考相关文献。

算法中参数的设置对优化效果起到至关重要的作用, PSO参数包括: 群体规模 $m$ , 惯性权重 $w$ , 加速常数 $c_1$ 和 $c_2$ , 最大速度 $V_{\max}$ , 最大代数次数 $G_{\max}$ 。 $V_{\max}$ 决定当前位置与最好位置之间的区域的分辨率(或精度)。如果 $V_{\max}$ 太高, 微粒可能会飞过好解; 如果 $V_{\max}$ 太小, 微粒不能在局部好区间之外进行足够的探索, 导致陷入局部最优。该限制有3个目的:

- (1) 计算溢出; 它将保持静止;
- (2) 实现人工学习和态度转变;
- (3) 决定问题空间搜索的粒度。

惯性权重 $w$ 使微粒保持运动惯性, 使其有扩展搜索空间的趋势, 有能力探索新的区域。加速常数 $c_1$ 和 $c_2$ 代表将每个微粒推向 $p_{best}$ 和 $g_{best}$ 位置的统计加速项的权重。低的值允许微粒在被拉回之前可以在目标区域外徘徊, 而高的值则导致微粒突然的冲向或越过目标区域。如果没有后两部分, 即 $C_1 = C_2 = 0$ , 微粒将一直以当前的速度飞行, 直到到达边界。由于它只能搜索有限的区域, 所以很难找到好解。如果没有第1部分, 即 $\omega = 0$ , 则速度只取决于微粒当前位置和其历史最好位置 $p_{best}$ 和 $g_{best}$ , 速度本身没有记忆性。假设一个微粒位于全局最好位置, 它将保持静止。而其它微粒则飞向它本身最好位置 $p_{best}$ 和全局最好位置 $g_{best}$ 的加权中心。在这种条件下, 微粒群将收缩到当前的全局最好位置, 更像一个局部算法。在加上第1部分后, 微粒有扩展搜索空间的趋势, 即第1部分有全局搜索能力。这也使得 $w$ 的作用为针对不同的搜索问题, 调整算法全局和局部搜索能力的平衡。如果没有第2部分, 即 $C_1 = 0$ , 则微粒没

有认知能力,也就是“只有社会(social only)”的模型。在微粒的相互作用下,有能力到达新的搜索空间。它的收敛速度比标准版本更快,但对复杂问题,则比标准版本更容易陷入局部最优值点。如果没有第3部分,即 $C2=0$ ,则微粒之间没有社会信息共享,也就是“只有认知(cognition only)”的模型。因为个体间没有交互,一个规模为 $m$ 的群体等价于运行了 $m$ 个单个微粒的运行,因而得到解的几率非常小。对全局搜索,通常的好方法是在前期有较高的探索能力以得到合适的种子,而在后期有较高的开发能力以加快收敛速度。

### § 4.3 粒子群算法优化实例

用PSO算法综合方向图的目的就是根据对波束形状的要求来求解天线阵列每个单元的激励幅值、相位和间距。以下我们来讨论PSO算法的各个参数在天线阵列综合问题中的具体设置,以及其适应度函数的具体设计:

一,种群的规模一般是由待优化的参数的个数来决定的,一般原则是种群个数远多于优化参数的个数;惯性权重 $\omega$ 可以是定值,也可是随叠代的次数呈线性变化,本文使用的是前者;加速常数 $c1$ 和 $c2$ 通常均等于2.0;最大速度 $v_{max}$ 为步长的十分之一左右;这些参数的设置都是根据大量的数值实验得出的,而程序的终止条件是程序运行达到了设置的最大叠代次数。种群中个体的维数就是所要优化的参数的个数,它由所要解决的实际问题来决定,在天线阵综合中,它通常可以由天线阵列单元间距,单元的激励幅度和相位这些参数构成。

二,适应度函数的设计是算法的关键,它的好坏直接决定了优化效果,它必须根据所要优化的问题具体设计。在天线阵列综合中,适应度函数一般表示为天线实际所产生的方向图与目标方向图的差异大小。首先计算出每个粒子的方向图与规定的理想方向图的误差,再对这个误差作变换得到适应度。误差越大,适应度越低;误差越小,适应度越高。在计算误差时,采用了最大误差的形式,即计算实际天线阵方向图与规定方向图在各个取样点(一般为副瓣位置和零深的位置)的误差之和,优化的目的就是使最大误差减小到最小,这样实际的方向图就最接近规定方向图。比较常用的适应度函数是:

$$fitness(x) = \min(|f_a(x, \theta_i) - f_d(\theta_i)|^2) \quad (4-3)$$

其中 $x = (m_1, \dots, m_u, \beta_1, \dots, \beta_u, d_1, \dots, d_u)^T$ 即由激励幅度,激励相位和单元间距组成,  $\theta_i$ 是辐射角,它们是离散的取样点(通常为方向图的副瓣位置和零深位置)。  $f_a(x, \theta_i)$ 是根据每个粒子计算出的方向图,  $f_d(\theta_i)$ 是所要逼近的目标方向图。

为了验证PSO在阵列天线波束赋形问题中的有效性,下面对上一章中的两个实例进行计算。

例一，综合直线阵扇形波束方向图

十个各向同性天线单元组成的直线阵，单元间距d均为半个波长，激励幅度为  $I_n$ ，激励相位为  $\beta_n$ 。其辐射方向图为： $F(\theta)=\sum_{n=1}^{10} I_n e^{j(nkd \sin(\theta)+\beta_n)}$ ，（其中  $\beta_n=0$ 或 $\pi$ ， $n=1\sim 10$ ）。要综合出的扇形波束为： $F_{desired}(\theta)=\begin{cases} 1, \pi/4 < \theta < 3\pi/4 \\ 0, \text{其余值} \end{cases}$

对于PSO，粒子个数设定为100，每个粒子的维数定为10（利用单元的激励幅度和相位分布的对称性），迭代200代。

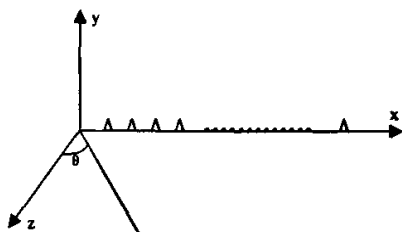


图4.2 阵列示意图

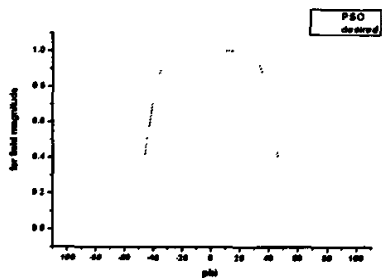


图4.3 方向图优化结果

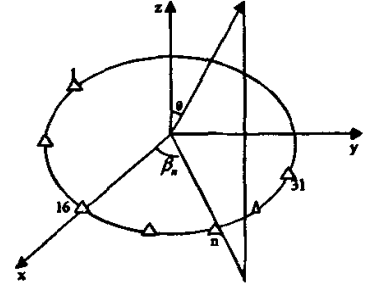
表4.1 PSO综合十元扇形方向图的单元激励幅度和相位

阵元 序号	1	2	3	4	5	6	7	8	9	10
激 励 幅 度	0.2743	0.9763	0.3645	0.1019	0.3085	0.3085	0.1019	0.3645	0.9763	0.2743
激 励 相 位	0	$\pi$	0	0	0	0	0	0	$\pi$	0

由图 4.3 可以看出，在此例中，PSO 的优化方向图达到了要求。表 4.1 给出了激励幅度和相位的优化结果。

## 例二 综合一个圆环阵

波束要求：主瓣波束宽度为  $30^\circ$ ，副瓣电平小于  $-35\text{dB}$ 。阵列由 31 个各向同性的单元组成，单元间距为半个波长，分布在半径为 6 个波长的一段圆环上。为了减少优化变量的个数，同样引入了修正的伯恩斯坦多项式，如式 (4-4)



$$F(U) = \begin{cases} B1 + \frac{1-B1}{A^{MA}(1-A)^{M(1-A)}} U^{MA}(1-U)^{M(1-A)}, & 0 \leq U \leq A \\ B2 + \frac{1-B2}{A^{MA}(1-A)^{M(1-A)}} U^{MA}(1-U)^{M(1-A)}, & A \leq U \leq 1 \end{cases} \quad (4-4)$$

其中， $B1, B2, M, A$  是多项式中的四个参数。多项式的单峰值出现在  $U=A$  处， $B1, B2$  分别确定多项式两个端点的值  $F(0)$ 、 $F(1)$ 。增大  $M$  的值可以使多项式的形状变得陡峭，减小则平坦。 $\theta=90^\circ$  时的方向图函数为：

$$E(\theta, \phi) = \sum_{n=1}^{31} F(U)_n \cdot \exp(j \cdot k \cdot r \cdot (\cos(\phi - \beta_n) - \cos(\beta_n))) \quad (4-5)$$

其中， $F(U)_n$  是第  $n$  个单元的激励幅度，是  $F(U)$  的第  $n$  个等分采样点的值， $\beta_n$  是第  $n$  个单元的相位补偿（以阵列中心为相位参考点）， $r$  是阵列的半径， $k$  是波数。

通过对  $M, A, B1, B2$  四个参数的适当选择，可以得到合适的  $F(U)$  并在其上采样 31 个点，采样点的值作为对应天线单元的激励幅度值，进而得到阵列的低副瓣、窄波束的方向图。选取适当的适应度函数，运用 PSO 对  $M, A, B1, B2$  四个变量进行优化（ $A, B1, B2$  变化范围是  $0 \sim 1$ ， $M$  的变化范围是  $1 \sim 20$ ）。使主瓣波束宽度小于  $30$  度，副瓣电平低于  $-35\text{dB}$ 。对于 PSO，粒子个数设置为 20，每个粒子的维数设为 4（ $A, B1, B2, M$ ），迭代 300 次。图 4.5 给出了运算结果：

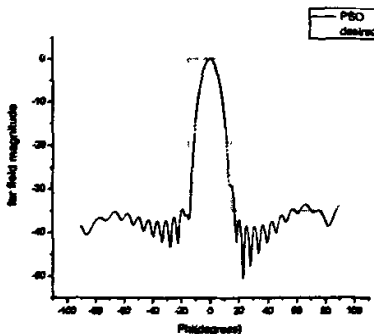


图 4.5 方向图优化结果

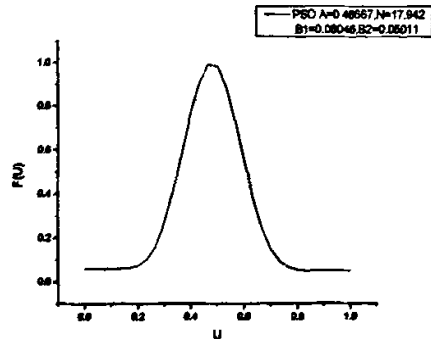


图 4.6 单元激励幅度分布

由图 4.5 可以看出, PSO 也达到了优化要求。为了便于比较遗传算法和粒子群优化算法结果的优异, 分别将两个实例的结果画在同一个图中, 见图 4.7 和图 4.8。

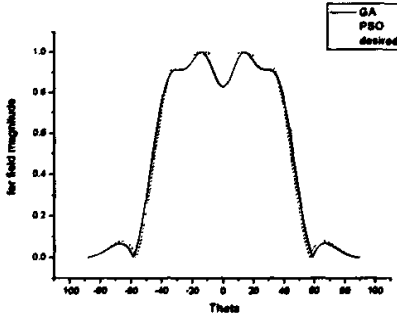


图 4.7 GA 和 PSO 十元阵

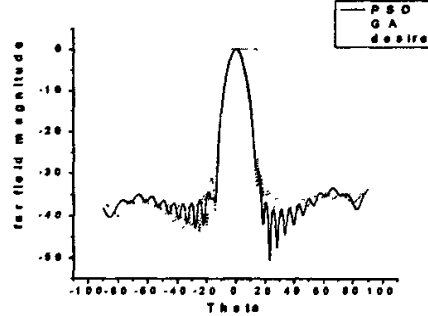


图 4.8 GA 和 PSO 三十一元阵

由图 4.7 可以看出对于十元阵, PSO 优化的结果中波束主瓣顶部更平坦, 在图 4.8 中, 结果较为接近, 但 PSO 优化的结果旁瓣更低一点。同时, 在编程过程中发现 PSO 的程序比 GA 的程序更短, 移植性更好, 需要调整的参数更少, 非常适于复杂阵列的综合。鉴于此, 以下的共形相控阵波束赋形问题都用 PSO 来优化。

我们先进一步分析一下第二个例子, 虽然已满足前面的要求也和文献中的结果相近, 但是对于所设计的适应度函数, 它只满足了副瓣低于 $-35\text{dB}$  和主瓣宽度小于 $30^\circ$ 这两点, 主瓣顶部不够平坦, 跌落很快。这可能是因为波恩斯坦多项式是单峰值函数的缘故。于是, 直接将三十一个天线单元的激励电流幅度作为变量进行优化, 这样虽然增加了变量个数, 加大了计算量, 但是得到了主瓣顶部平坦度很好的结果, 此时激励幅度是双峰值分布。图 4.9 和图 4.10 是用 PSO 得到的不同主瓣宽度的平顶方向图。由于增加了需要优化变量的个数, 相应的在优化中将粒子个数增大到 1200 个, 适当调整适应度函数, 经过 100 步迭代, 得到了预期的结果。

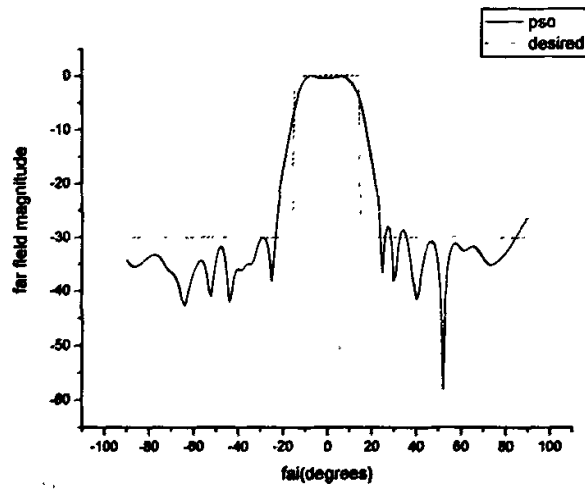


图 4.9 宽波束方向图

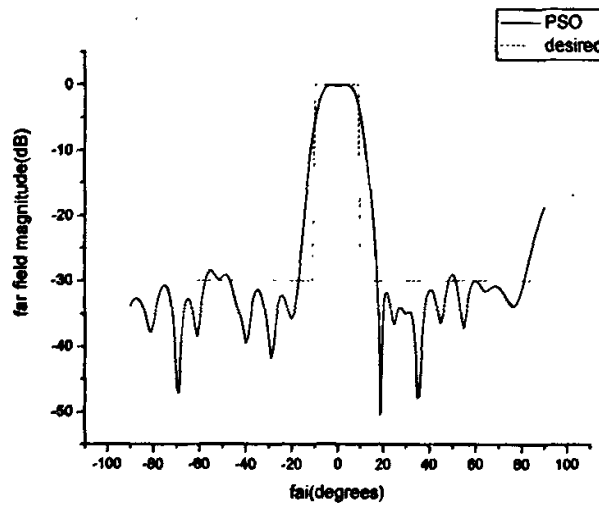


图 4.10 窄波束方向图

§ 4.4 在共形相控阵天线波束赋形中的应用

通过上一节中对 PSO 和 GA 的比较,发现 PSO 的一些优点使其更适合在共形相控阵波束赋形问题中应用。下面选用 PSO 算法对一个工程项目中波束赋形问题进行优化。项目中要求用小尺寸、少单元的圆柱共形阵列得到副瓣电平较低并且主瓣顶端较平坦的波束方向图。考虑到大功分比的功分器不易实现,所以在用粒子群算法进行优化时,程序中将单元间的幅度锥削按照功分比为 2:1 的功分器所能提供的幅度值进行预先设定。为了用尽量少的天线单元得到符合要求的辐射方向图,对多种可能的阵列形式进行了多次计算。在计算时忽略了互耦的因素,由于单元间距较大,也应该不会对实际的方向图有很大的影响。以下是所尝试的一些阵列形式和优化结果。阵列中使用的天线单元都是全向的,并且忽略了圆柱对阵列的影响。

先尝试用四个单元组成阵列:将四个单元按间距  $d$  均匀排列在半径为 6 个波长的圆环上,单元间距  $d$  和激励幅度按照表 4.2 所示的八种情况进行设定。

表 4.2 尝试的 8 种四单元阵列形式

单元间距 $d$	$2*\lambda$	$1.5*\lambda$	$\lambda$	$0.75*\lambda$
等幅激励(归一)	1: 1: 1: 1			
锥削幅度(归一)	0.707: 1: 1: 0.707			

由于天线单元个数太少,这几种阵列形式都没有能优化出符合要求的波束形状,最接近的波束如图 4.11 所示,远不能满足对副瓣电平和主瓣宽度的要求。

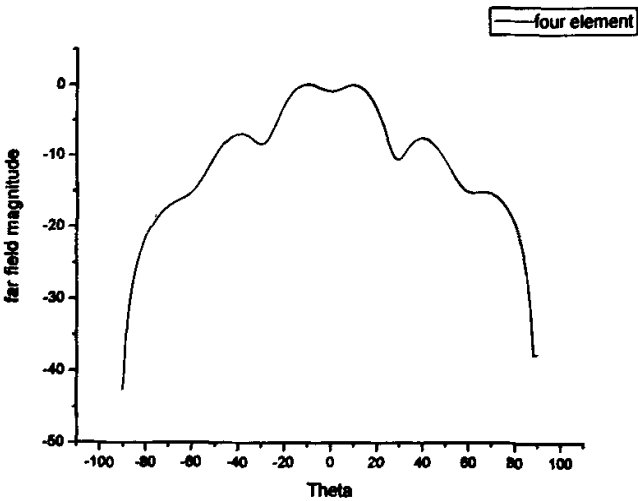


图 4.11 四元阵中最佳结果

从中可以看出：阵元太少使得优化的自由度太小，并且功分比不够大无法提供足够的馈电锥削，所以很难使波束形状达到要求。

于是我们增加了两个天线单元，下面是对六元阵的尝试：将六个单元以间距  $d$  均匀排列在半径为 6 个波长的圆环上。分别按照表 4.3 所示的单元间距和激励幅度的十二种组合情况进行计算。

表 4.3 尝试的 12 种六单元阵列形式

单元间距 $d$	$2*\lambda$	$1.5*\lambda$	$\lambda$	$0.75*\lambda$
等幅激励（归一）	1: 1: 1: 1: 1: 1			
锥削激励 1（归一）	0.5: 0.5: 1: 1: 0.5: 0.5			
锥削激励 2（归一）	0.408: 0.577: 1: 1: 0.577: 0.408			

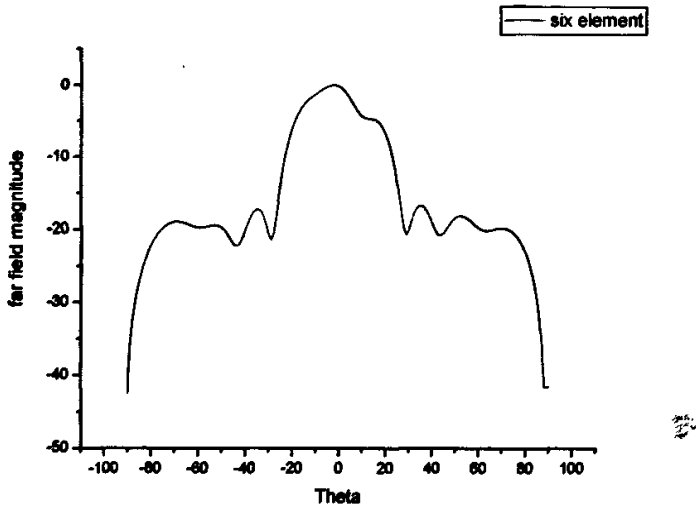


图 4.12 六元阵中最佳结果

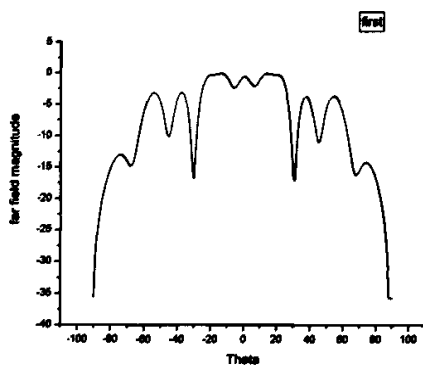
图 4.12 给出了最好的一个结果，是按照第二种锥削情况优化出来的，方向图的副瓣电平已经比较低了，但是主瓣顶端的波纹起伏太大，仍旧无法满足要求，有待于进一步的改善。

还得再增加天线单元，加大自由度，才能优化出理想的方向图。下面，对等幅馈电八元阵列进行优化。采用同样的阵列排布方式，按照表 4.4 的九种情况进行计算。

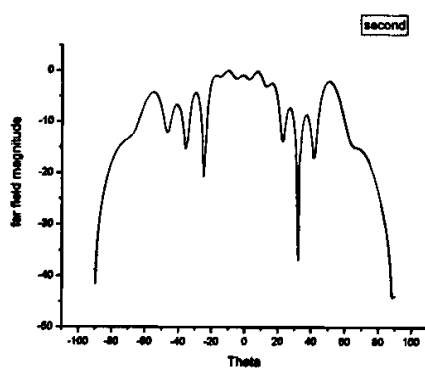


表 4.4 尝试的 9 种等幅八元环阵形式

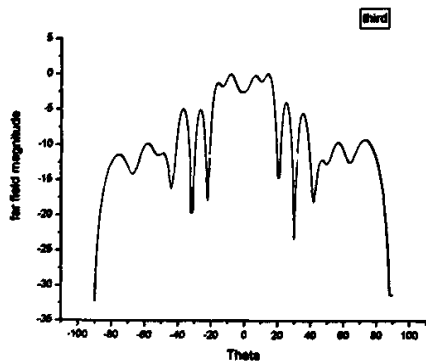
圆 环 半径	$7\lambda/\pi$	$14\lambda/\pi$	$28\lambda/\pi$	$21\lambda/4\pi$	$21\lambda/2\pi$	$21\lambda/\pi$	$7\lambda/2\pi$	$7\lambda/\pi$	$14\lambda/\pi$
单 元 间距	$\lambda$			$3*\lambda/4$			$\lambda/2$		
计 算 结果	图 4.13 a	图 4.13 b	图 4.13 c	图 4.13 d	图 4.13 e	图 4.13 f	图 4.13 g	图 4.13 h	图 4.13 i



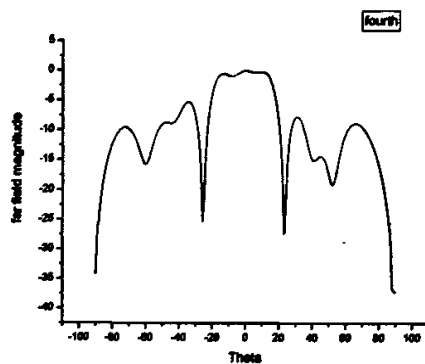
a



b



c



d

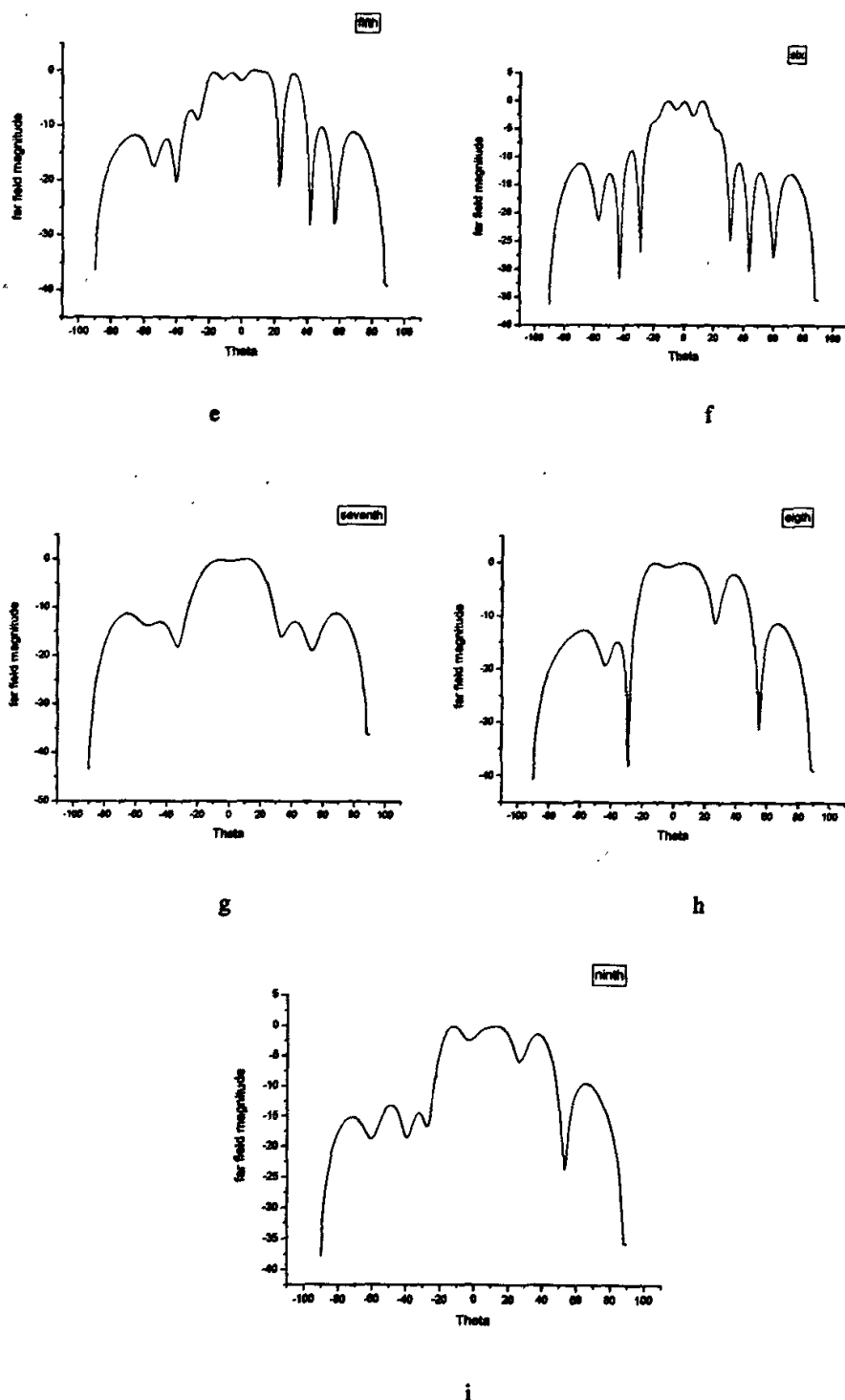


图 4.13 9 种八元环阵等幅激励的方向图

由图 4.13 可见,虽然增加了两个天线单元但是由于采用等幅激励使优化的自由度更少了,很难优化出令人满意的结果。只有第 g 种情况,阵元间距是半个波

长，圆弧半径是  $7\lambda/2\pi$  时优化的波束方向图还基本能达到要求，但是阵列半径太小无法排列单元，所以这种阵列形式也不能应用到这个项目中。

考虑到单元间距较小时，天线单元之间的互耦会比较强烈，对方向图的影响相对较大。天线单元的间距较大时，又会使天线阵列体积过大，不够紧凑，影响使用。于是综合考虑后确定将天线单元间距定为  $3/4$  个波长，阵列圆弧半径为  $21/\pi$  个波长，采用锥削幅度激励。图 4.14 给出了对八元阵列提供锥削馈电的用四阶一分二功分器级联出的一分八功分器的示意图。表 4.5 列出了八元阵列可采用的四种馈电方式。

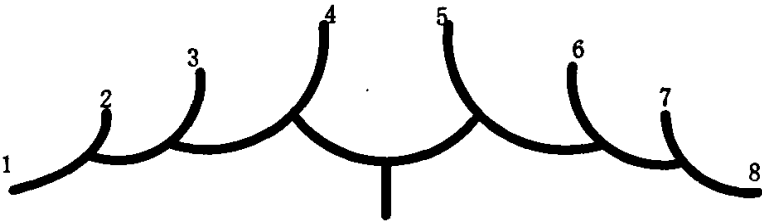


图 4.14 四阶一分八功分器示意图

表 4.5 4 种锥削馈电八元阵列

	单元馈电幅度分布	计算结果
锥削 1	电压 0.354: 0.354: 0.5: 1: 1: 0.5: 0.354: 0.354	图 4.15
	功率 0.125: 0.125: 0.25: 1: 1: 0.25: 0.125: 0.125	表 4.6
锥削 2	电压 0.289: 0.289: 0.577: 1: 1: 0.577: 0.289: 0.289	图 4.16
	功率 1/12: 1/12: 1/3: 1: 1: 1/3: 1/12: 1/12	表 4.7
锥削 3	电压 0.236: 0.333: 0.577: 1: 1: 0.577: 0.333: 0.236	图 4.17
	功率 1/18: 1/9: 1/3: 1: 1: 1/3: 1/9: 1/18	表 4.8
锥削 4	电压 0.5: 0.5: 0.707: 1: 1: 0.707: 0.5: 0.5	图 4.18
	功率 0.25: 0.25: 0.5: 1: 1: 0.5: 0.25: 0.25	表 4.9

下面是分别采用这四种锥削激励的优化结果：

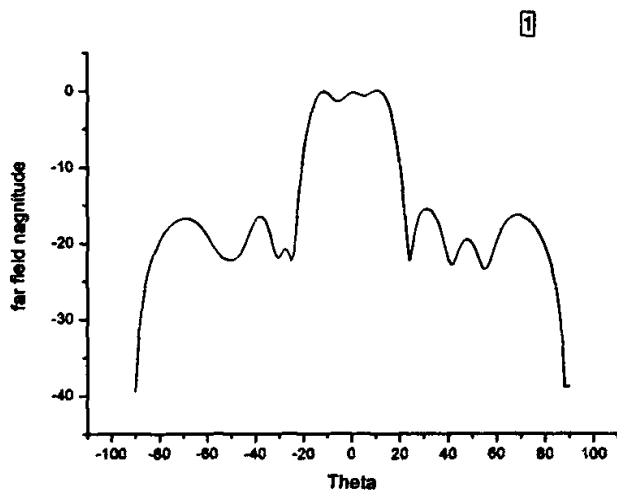


图 4.15 采用锥削 1 激励的方向图

表 4.6 采用锥削 1 激励的幅度和相位值

单元序数	1	2	3	4	5	6	7	8
激励幅度(归一)	0.354	0.354	0.5	1	1	0.5	0.354	0.354
激励相位(度)	93.742	107.172	35.541	-27.256	-20.243	9.219	101.219	105.470

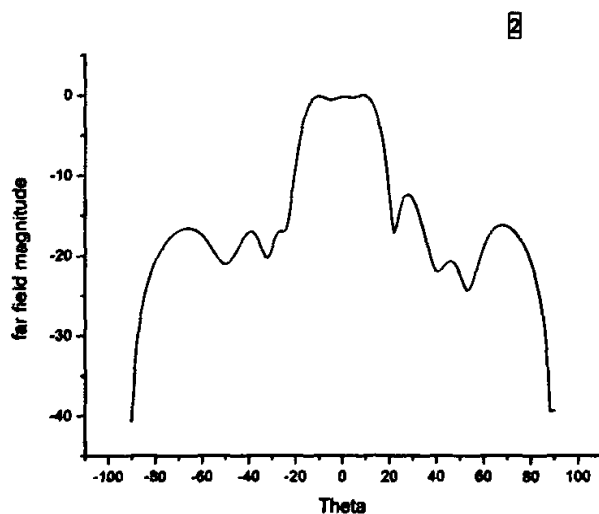


图 4.16 采用锥削 2 激励的相位

表 4.7 采用锥削 2 激励的幅度和相位值

单元序数	1	2	3	4	5	6	7	8
激励幅度(归一)	0.289	0.289	0.577	1	1	0.577	0.289	0.289
激励相位(度)	100.188	97.769	13.476	-37.202	-21.486	0.372	100.234	111.956

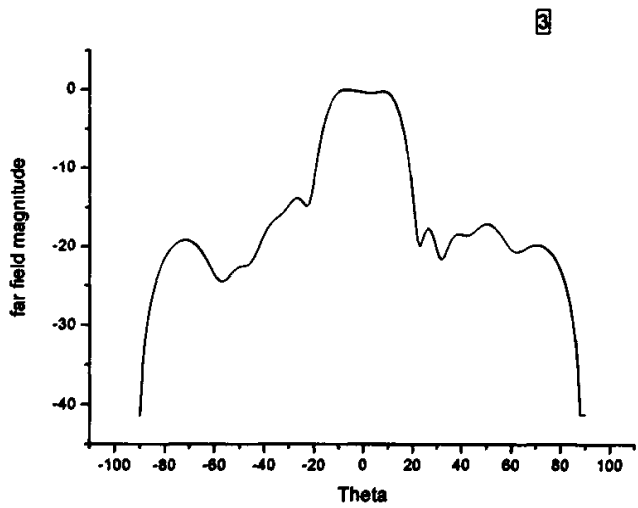


图 4.17 采用锥削 3 激励的相位  
表 4.8 采用锥削 3 激励的幅度和相位值

单元序数	1	2	3	4	5	6	7	8
激励幅度(归一)	0.236	0.333	0.577	1	1	0.577	0.333	0.236
激励相位(度)	146.815	80.908	-4.079	-7.179	-18.037	11.299	109.292	145.543

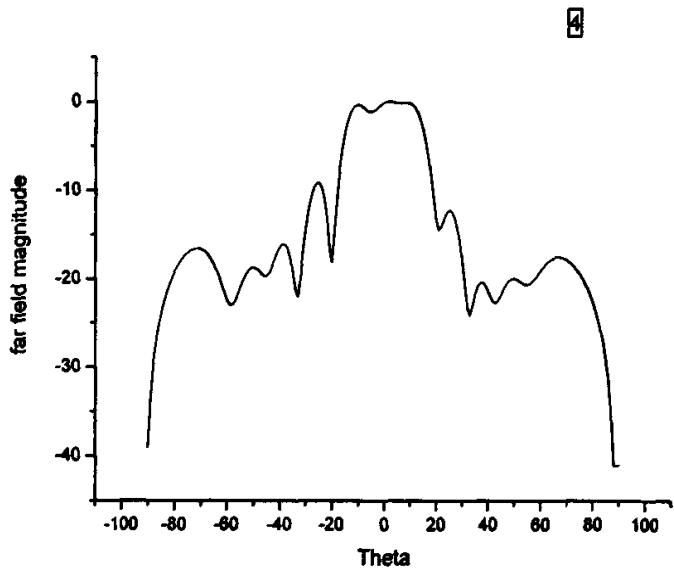


图 4.18 采用锥削 4 激励的相位  
表 4.9 采用锥削 4 激励的幅的和相位值

单元序数	1	2	3	4	5	6	7	8
激励幅度(归一)	0.5	0.5	0.707	1	1	0.707	0.5	0.5
激励相位(度)	116.403	101.225	7.735	-16.513	-16.662	-6.979	60.866	125.192

以上八元阵列的计算结果中,图 4.16 和图 4.17 所示的方向图不仅副瓣较低而且主瓣顶端较为平坦,满足了设计要求。通过对表 4.6 到表 4.9 的比较不难看出,可以通过加大馈电锥削程度、减小阵列边缘单元的激励幅度来降低副瓣电平。

已经得到了较好的优化结果,但是其馈电网络是用一分二的功分器经过四阶级联得到的。考虑到功分器的级联阶数越多,其馈电网络的插入损耗也就越大,在相同的信号输入情况下天线阵列的增益会降低,于是又想尝试利用一分二的功分器进行三阶级联得到一分八的功分器作为馈电网络。其示意图见图 4.19。

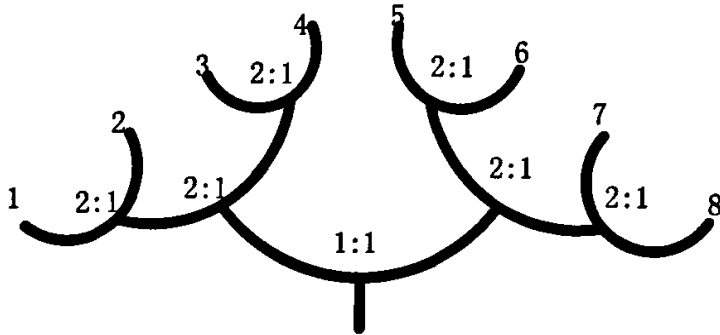


图 4.19 三阶级联一分八功分器

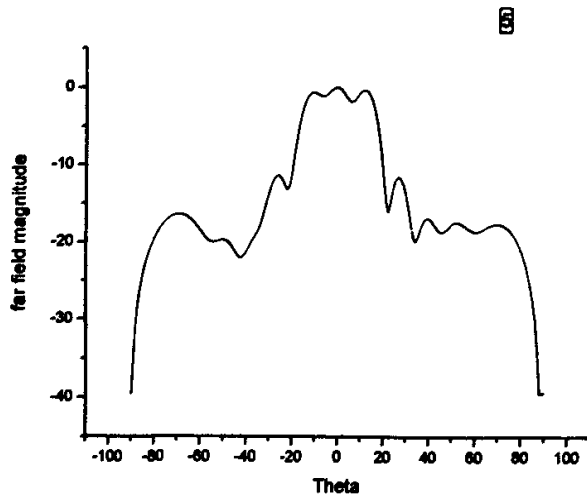


图 4.20 采用三阶级联时得到的方向图

表 4.10 采用三阶级联时得到的幅度和相位值

单元序号	1	2	3	4	5	6	7	8
激励幅度（归一）	0.5	0.707	0.707	1	1	0.707	0.707	0.5
激励相位（度）	121.708	78.564	2.223	-14.095	-24.746	15.138	94.601	110.753

虽然功分器的级联少了一阶可以提高增益,但是这种功分器所能给阵列提供

的馈电幅度锥削形式很有限。从图 4.20 可以看出,方向图的主瓣平坦度并不是很好且副瓣电平也有所升高。所以,只能采用表 4.7 和表 4.8 中提供的四阶级联的方式构建馈电网络。到此,通过优化计算已经基本上完成了对阵列的设计。在下一章中将介绍功分器的仿真和实测结果。

## 第五章 馈电网络

### § 5.1 引言

对于实现天线阵列的波束赋形, 一个重要的问题就是如何将发射机输出的信号, 按一定的幅度和相位分布馈送给阵面上的每一个天线单元。相控阵天线的馈电网络, 就是为解决这一问题而使阵面上众多的天线单元与发射机或接收机相连接的传输系统。当然, 馈电网络插入损耗的大小, 也将直接影响到相控阵天线增益的高低。

功分器, 是相控阵天线馈电网络的重要组成部分。在微波设备中, 常需要将某一输入功率按一定的比例分配到各分支电路中, 例如对相控阵, 要将发射机功率分配到各个发射单元中去; 多路中继通信机中要将本地振荡源功率分到收发混频电路中等。在数目较少的分功率电路中也可以将定向耦合器作为功率分配器, 但是定向耦合器的结构较为复杂, 其功率分配的比值又往往与频率有关; 而在较复杂的功率分支电路中 (特别是微带电路) 需要元件较多, 就要采用结构比较简单的功分器。功分器的基本要求是: 输出功率按一定的比例分配, 各输出口之间要互相隔离以及各输入输出口必须匹配。一般功分器也可以应用其逆过程, 即作为功率混合器。

### § 5.2 功分器原理

功分器可以分为二进制和累进制, 即一分二和一分多等类型。可以是等功率分配的, 也可以是不等功率分配的。二进制功分器结构和分析都较简单, 用的也较多。下面主要介绍二进制不等功分功率分配器<sup>[20]</sup>, 当 $k$ 取1时就是等功分的情况。

不等分功分器如图5.1, 其中输入端口特性阻抗为 $Z_0$ , 分成的两段输出分支的电长度为 $\lambda_g/4$ , 特性阻抗分别是 $Z_{02}$ 和 $Z_{03}$ , 终端分别接有电阻 $R_2$ 和 $R_3$ 。对于理想功分器其基本要求是: 输入端口(1)无反射; 输出分支在(2)、(3)处电压相等且同相; 端口(2)、(3)输出功率比值可为任意指定值, 设为 $1/k^2$ 。



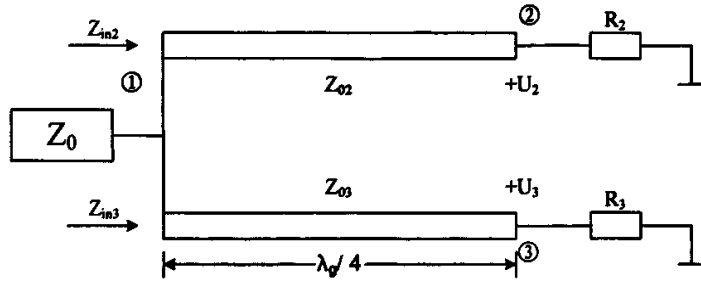


图 5.1 功率分配器的平面结构

根据以上三条有：

$$\frac{1}{Z_{in2}} + \frac{1}{Z_{in3}} = \frac{1}{Z_0} \quad (5-1)$$

$$\left( \frac{1}{2} \frac{U_2^2}{R_2} \right) / \left( \frac{1}{2} \frac{U_3^2}{R_3} \right) = \frac{1}{k^2}$$

$$U_2 = U_3$$

有传输线理论有

$$Z_{in2} = \frac{Z_{02}^2}{R_2}$$

$$Z_{in3} = \frac{Z_{03}^2}{R_3} \quad (5-2)$$

这样共有  $R_2$ 、 $R_3$ 、 $Z_{02}$ 、 $Z_{03}$  四个参数而只有三个约束条件，故可任意指定其中的一个参数，现设  $R_2 = kZ_0$ ，于是有上两式可得其它参数：

$$Z_{02} = Z_0 \sqrt{k(1+k^2)}$$

$$Z_{03} = Z_0 \sqrt{(1+k^2)/k^3}$$

$$R_3 = \frac{Z_0}{k} \quad (5-3)$$

实际的功率分配器终端负载往往是特性阻抗为  $Z_0$  的传输线，而不是纯阻抗，此时可用  $\lambda_g/4$  的阻抗变换器将其变为所需电阻，另一方面  $U_2$ 、 $U_3$  等幅同相，在 (2)、(3) 端跨接电阻  $R_j$ ，既不影响功率分配器性能，又可增加两输出端口之间的隔离度。于是实际功分器平面结构如图 5.2 所示，其中  $Z_{04}$ 、 $Z_{05}$  及  $R_j$  有以下公式确定：

$$Z_{04} = \sqrt{R_2 Z_0} = Z_0 \sqrt{k}$$

$$Z_{05} = \sqrt{R_3 Z_0} = Z_0 / \sqrt{k}$$

$$R_j = Z_0 \frac{1+k^2}{k} \quad (5-4)$$

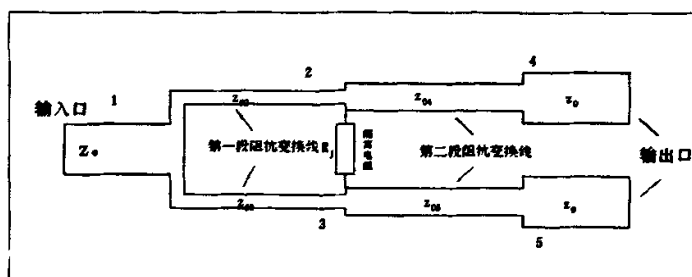


图 5.2 不等功分比功率分配器平面结构

和它的微带电路元件一样，功分器也有一定的频带响应特性。以上讨论的单节功分器的适用频带是有限的，若要提高功分器的频带宽度，可以采用多节变换形式的功分器来实现宽带特性。

功率分配器两功分臂之间的距离不易过大，这样可使跨接在两臂之间的隔离电阻寄生效应尽量减小。由于电阻寄生引线的电感效应，将使匹配性能和隔离性能变坏，此时可变动电阻焊接的位置，使其稍偏离原来的位置（一般来说，应移出 $\lambda_g/4$ 第一、二段阻抗变换线的交界点，偏向第二段一边），这样可得到较好的匹配和隔离特性。

### § 5.3 微带功分器的设计

设计一个一分二等功分器，现有介质为 $\epsilon_r = 2.65$ 的聚四氟乙烯双面敷铜板，分析可知，增加介质板的厚度，可以使功分器的工作带宽加宽，但介质损耗也会增加，而且会增大功分器的设计面积；减小介质板厚度，将使工作带宽变窄，减小介质损耗，同时可以减小功分器的设计面积<sup>[27], [28], [29]</sup>。综合考虑微带线的工作性能及加工时可能带来的误差后，选择基片厚度 $h=1\text{mm}$ 的聚四氟乙烯双面敷铜板为功分器微带电路的介质基片。

为了便于馈电网络中各器件之间的连接，输入输出端口微带线特性阻抗设为 $Z_0 = 50\Omega$ ，一分二支路微带线的特性阻抗

$$Z_1 = \sqrt{2}Z_0 = 70.7\Omega \quad (5-5)$$

微带电路带线特性阻抗的近似计算公式

$$Z_0 = 60 \ln \left( \frac{8h}{W} + \frac{W}{4h} \right) \Omega \quad (W/h \leq 1) \quad (5-6)$$

$$Z_0 = \frac{120\pi}{W/h + 2.42 - 0.44 \frac{h}{W} + (1 - \frac{h}{W})^6} \Omega \quad (W/h \geq 1) \quad (5-7)$$

对于  $\varepsilon_r = 2.65$ ,  $h=1\text{mm}$  的介质材料, 特性阻抗  $Z_0 = 50\Omega$  对应的带线宽度  $W_1 = 2.85\text{mm}$ , 特性阻抗  $Z_1 = 70.7\Omega$  对应的带线宽度  $W_2 = 1.61\text{mm}$ 。

中心频率  $f_0$  对应的介质工作波长为

$$\lambda_g = \frac{\lambda_0}{\sqrt{\varepsilon_e}} \quad (5-8)$$

式中,  $\varepsilon_e$ ——介质有效介电常数

$$\varepsilon_e = \frac{1+\varepsilon_r}{2} + \frac{\varepsilon_r-1}{2} \left(1 + \frac{10h}{W}\right)^{-\frac{1}{2}} \quad (5-9)$$

结合一分二功分器的设计尺寸, 由式 (5-8) 可得到

$$\frac{1}{4}\lambda_g \approx 0.167\lambda_0 \quad (5-10)$$

采用 Ansoft9.0-Designer 对功分器进行仿真计算, 仿真过程中隔离电阻用等效双端口网络代替, 介质的损耗角正切设为  $\tan \delta = 0.001$ , 通过对功分器两支路中阻抗变换段长度  $l$  的不断调整, 最终得到了一个较为理想的结果, 仿真中一分二等功分器的结构如图 5.3 中所示。

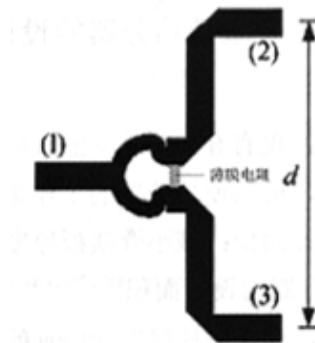


图 5.3 一分二等功分器的结构图

测量图 5.3 功分器的 S 参数, 并与仿真比较, 结果如图 5.4、图 5.5、图 5.6 所示:

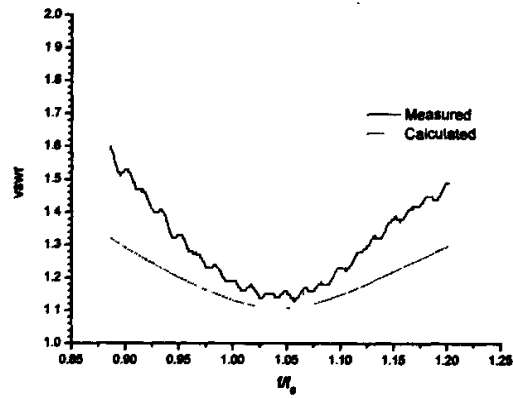


图 5.4 一分二等分功分器 1 口驻波测量和仿真结果

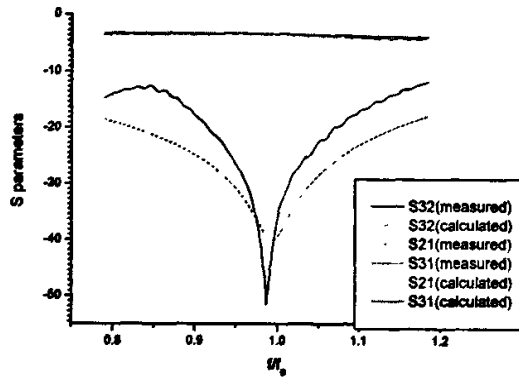


图 5.5 一分二等分功分器 S 参数仿真和实测结果

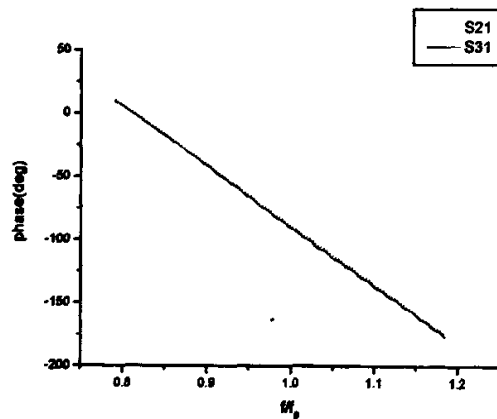


图 5.6 输出端口的相位测量结果

$f_0$  为中心频率, 由图 5.4, 图 5.5, 图 5.6 可见, 实测的各项参数与仿真结果基本吻合。测量的驻波比  $\rho < 1.5$  带宽为 35%。仿真和实测的 S 参数值见表 5.1。

表 5.1 S 参数的仿真和实测值

		S21 (dB)	S31 (dB)	S23 (dB)
仿真	最大值	-3.2	-3.29	-22.2
	最小值	-3.7	-3.9	-41
实测	最大值	-3.5	-3.2	-16.38
	最小值	-3.8	-3.7	-51

表 5.1 中数据均在中心频率  $f_0$  附近得到。 $S_{21}$  和  $S_{31}$  的相位差的最大测量值为  $1.85^\circ$ 。实测的驻波值比仿真的结果略大一点，这可能是由于焊接上 SMA 接头带来的影响。

对于功分比为 2: 1 的微带功分器，同样选用介质厚度  $h=1\text{mm}$ ， $\epsilon_r=2.65$  的聚四氟乙烯双面敷铜板，工作在  $f_0$  时各段微带线的宽度如表 5.2 所示，功分器的结构如图 5.7 所示。

表 5.2  $k^2$  为 2 时各段微带线的宽度

	$Z_0$	$Z_{02}$	$Z_{03}$	$Z_{04}$	$Z_{05}$	$R_j$
阻值(欧姆)	50	103	51.5	59.5	42	106
导带宽度 $/\lambda_0$	0.0709	0.019	0.0659	0.0522	0.0899	薄膜电阻

采用表 5.2 中的数据进行建模，用 Designer 仿真，通过对多种结构的尝试和对阻抗变换段长度在  $\lambda_0/4$  附近反复微调最终确定了图 5.7 所示的功分器结构，计算结果见图 5.8 和图 5.9

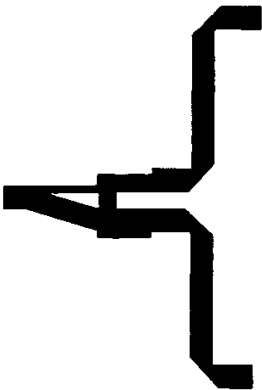


图 5.7 不等分功分器结构图

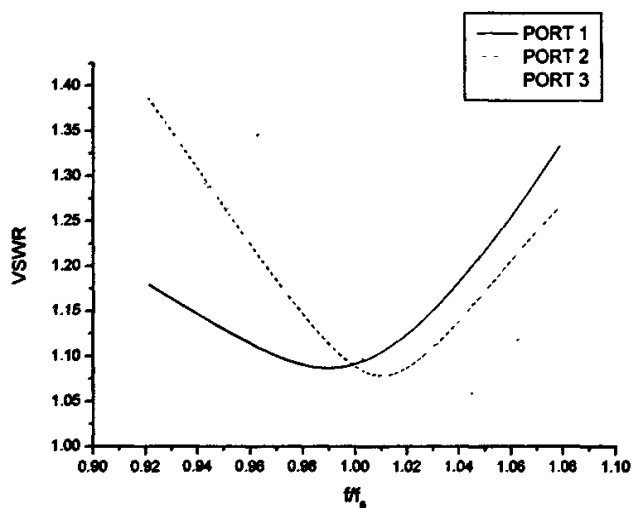


图 5.8 三个端口的驻波

图 5.8 中给出了三个端口的驻波值, 在中心频率  $f_0$  处端口 1、2、3 的驻波分别为 1.090, 1.088, 1.106。较小的驻波值保证了能量的顺畅传输。

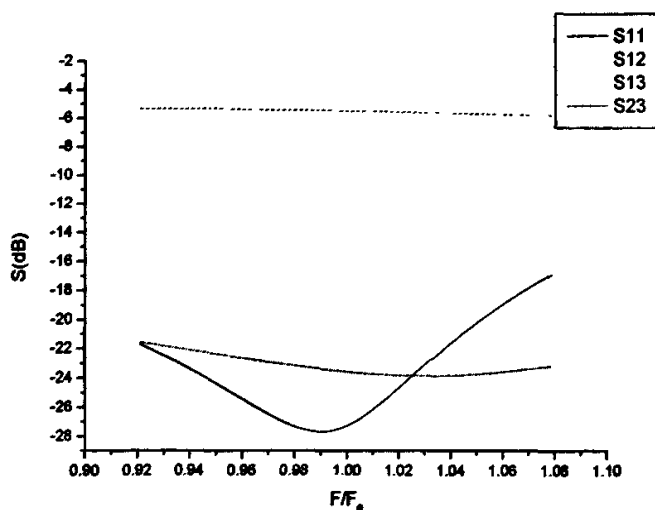


图 5.9 S 参数

图 5.9 给出了几个关键的 S 参数。在中心频率  $f_0$  处隔离度  $S_{23}$  达到了 -23.57dB, 使得端口 2、3 之间的影响很小;  $S_{12}$  是 -5.49dB,  $S_{13}$  是 -2.58dB,  $S_{13} - S_{12} = 2.91$ dB (接近 3dB) 保证了功分比为 2:1。  $S_{12}$  比理论值 (-4.8dB) 小了 0.69dB,  $S_{13}$  比理论值 (-1.8dB) 小了 0.78dB, 这可能是因为介质的损耗和阻抗变换段匹配的还不够理想等原因造成的, 会有一些能量损失掉。这个功分器还可以进一步完善, 不

过对于这个项目的要求它已经基本满足了。

## 结束语

文中对两种新的智能随机算法:粒子群优化算法(PSO)和遗传算法(GA)做了详细介绍,它们只需利用目标的取值,而无需梯度等高价值信息,适于并行处理,尤其适用于处理传统搜索方法难于解决的复杂和非线性问题。文中把这两种算法应用到了共形相控天线的波束赋形问题中,应用MATLAB语言编写程序进行计算,取得了很好的效果。论文中所做的主要工作如下:

1. 介绍了两种智能随机算法——粒子群优化算法(PSO)和遗传算法(GA),详细讨论了算法的原理和具体参数的设置,然后把它们应用于天线阵列的波束综合设计中,编写程序对参考文献中的实例进行计算,得到了正确的计算结果,更好的掌握了这两种算法。并且通过相互对比,发现PSO更适合于在波束赋形问题中的应用。
2. 根据工程的要求:设计一个使用较少数天线单元、较小的阵列尺寸、在圆柱面上共形,副瓣要低的圆形阵,主波束要有一定的宽度且顶部的平坦度较好。编写程序对多种方案进行多次优化,最后得到了一种理想的可实现方案。
3. 设计了天线阵列的馈电网络,介绍了功分器的原理,分别完成了等功分和功分比为2:1的一分二微带功分器的仿真,其中等功分的功分器已经加工、调试完毕。得到的测量结果,满足输入阻抗频带宽、各端口驻波系数小、插入损耗低、输出端口间隔高度大的要求。

由于作者的水平有限加之时间仓促,本文仅对两种优化算法做了较为粗浅的研究,在共形相控阵列天线波束赋形问题上的应用也不够深入,毫无疑问这是一个很广阔的领域,还有许多工作有待我们今后去做。例如:在共形阵列综合中充分考虑到天线所附着的物体对输入阻抗的影响和各个天线单元之间互耦的影响,以及由此而来的匹配网络的设计,以便更好的与实际情况相吻合;对粒子群算法和遗传算法做适当的改进,设计更好的适应度函数,使它能更好的应用于天线阵综合问题中;另外,大功分比的功分器还是一个有待解决的问题。

通过对本学位论文的研究工作,使我对天线阵列和优化算法有了较为深入的学习,进一步夯实了专业理论基础,开拓了视野,扩大了知识面。通过阅读大量的英文文献,从而提高了英文科技文阅读水平,为以后从事研究工作奠定了坚实的基础。

文中还存在的疏漏和不足之处,请各位专家、同学批评指正。



## 致 谢

在毕业论文即将完成之际，我衷心感谢我的导师鄢泽洪教授，感谢两年多来鄢老师对我的悉心教育、耐心培养和无微不至的关怀。鄢老师严谨的治学态度，认真的敬业精神，时时刻刻激励着我在工作和学习中不怕困难，刻苦钻研。感谢他用自己的辛勤耕耘和敬业精神指导并感染着我完成了人生中这一重要阶段的学习和工作。同时要感谢鄢老师在生活上给予我的关怀和无私的帮助，使我能以饱满的热情和良好的精神状态投入到繁忙而充满乐趣学习和工作中去。

感谢张小苗教授给予我的帮助，在他的指导和帮助下，我分析问题、解决问题的能力有了明显提高，在多次的讨论中，张老师的指导使我少走了许多的弯路，感谢他在论文研究的整个过程中给予我的无私教诲。张老师是位知识渊博的导师，在我两年多的研究生学习期间，他充实了我的理论基础，开拓了我的眼界，并使我了解和掌握课题研究领域的一些前沿科技。

感谢魏文元教授时常关心我论文研究的进展，给予我天线设计方面许多有益的工程经验，使我少走了很多弯路。魏老师对待科学的严谨的态度同样给我的论文工作以很大的帮助。同时，也感谢魏老师在生活上对我的关心。

感谢李延平老师，感谢她在我两年多的学习、科研生活中对我的照顾，帮我解决了一些生活上、学习中的困难。

感谢李彪老师，是他帮助我解决了许多天线加工、测试中遇到的问题，并且感谢他在生活中对我的关心和鼓励。

感谢我的师兄史永康、孙慧峰、庄建楼、王果宏、魏永华、李强和师姐金颖、王泽美，他（她）们给了我很多关心和帮助。感谢课题组同学李书杰、郑治泰、李毓、吕源、张纪明、高明泉、张天龄、魏宏泼、贾秀娟等人一直来给予我的合作和帮助。

最后，我要特别感谢我的父亲和母亲，他们在我成长的过程中，给予我莫大的支持和鼓励，同时付出了艰辛的劳动，对他们为我付出的心血再次表示最衷心的感谢！

最后，感谢所有关心、支持和帮助过我的人。

## 参考文献

- [1] 汪茂光、吕善伟、刘瑞祥,《阵列天线分析与综合》,电子科技大学出版社,1989年10月
- [2] 王小平,曹立明,《遗传算法——理论、应用》。西安,西安交通大学出版社,2002
- [3] 任盛海,吴志忠 遗传算法在阵列天线方向图综合设计中的应用[J],电波科学学报,1996,11(4):62-67.
- [4] 刘昊,郑明 遗传算法在阵列天线赋形波束综合中的应用[J],电波科学学报,2002,17(5):539-548.
- [5] Rene J. Allard, Douglas H. Werner and Pingjuan L. Werner, "Radiation pattern synthesis for arrays of conformal antennas mounted on arbitrarily-shaped three-dimensional platforms using genetic algorithms", IEEE Trans. Antennas Propagation, Vol.51, No.5, May.2003, pp.1054-1062
- [6] 谢晓峰,张文俊,杨之廉. 粒子群算法综述[J]. 控制与决策,2003,18(3):129-132.
- [7] Carlos A.Handling Multiple Objectives With Particle Swarm Optimization IEEE Transactions on Evolutionary Computation, Vol.8.No.3.June 2004.
- [8] J.Robinson, Y.Rahmat-Samii Particle swarm optimization in electromagnetics[J]. IEEE Trans on AP2004, 52(2): 397-407.
- [9] Dennis Gies ,Yahya Rahmat-Samii, Reconfigurable Array Design Using Parallel Particle Swarm Optimization,IEEE,2003
- [10] 杨科,陈胜兵,焦永昌,张福顺,粒子群优化算法用于阵列天线方向图综合设计,电波科学学报,2005.
- [11] Daniel W. Boeringer and Douglas H. Werner, Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis, IEEE Transaction on Antennas and Propagation, Vol. 52, NO. 3, March, 2004
- [12] 吕善伟,《天线阵综合》,北京航空学院出版社,1988年1月
- [13] 魏文元、宫德明、陈必森,《天线原理》,国防工业出版社,1995年6月
- [14] 张光义,《相控阵雷达系统》,国防工业出版社,1994年8月
- [15] 丁永生,《计算智能——理论、技术与应用》,科学出版社,2004年8月
- [16] D.W Boeringer,D.H.Werner Particle Swarm Optimization of a Modified

- Bernstein Polynomial for Conformal Array Excitation Synthesis.IEEE.2004
- [17] 杨维、李歧强, 粒子群优化算法综述, 中国工程科学, 2004 年 5 月, 第 6 卷第 5 期
- [18] Konstantinos E. Parsopoulos and Michael N. Vrahatis On the Computation of All Global Minimizers Through Particle Swarm Optimization [A], IEEE Transactions on evolutionary computation, 2004,8(3): 211---224
- [19] Carlos A.Coello Coello,Gregorio Toscano Pulido,Maximino Salazar Lechuga, Handling Multiple Objectives With Particle Swarm Optimization[A],IEEE Transactions on evolutionary computation, 2004,8(3): 256---279
- [20] J. Kennedy , R. Eberhart. Particle swarm optimization [A]. In Proc IEEE Int Conf.Neural Networks [C].Perth, Australia ,1995, 1942-1948.
- [21] Millonas.M M. Swarm sphase transition and collective intelligence[M]. MA: Addison wesley,1994.
- [22] Parsopoulos K E ,etal. Stretching Technique for Obtaining Global Minimizers through Particle Swarm Optimization [ R] . Proceedings of the Workshop on PSO , Indianapolis :Purdue school of Engineering and Technology , INPUI ,2001.
- [23] Daniel W. Boeringer and Douglas H. Werner Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis IEEE Transactions on Antennas and Propagation,Vol.52,No.3,March 2004
- [24] Angeline P J . Using selection to improve particle swarm optimization[ R ] . IEEE International Conference on Evolutionary Computation,Anchorage,Alaska, 1998.
- [25] Yuhui Shi,Russel C.Eberhart. Empirical study of particle swarm optimization. [A] Proc of the Congress on Evolutionary Computation [ C ]. Washington DC, 1999: 1945-1950.
- [26] 清华大学,《微带电路》,人民邮电出版社,1976 年 9 月
- [27] Young-Joon Ko, Jae-Yeong Park and Jong-Uk Bu, "Fully integrated unequal wilkinson power divider with EBG CPW", IEEE Microwave and Wireless Components Letters, Vol.13, No.7, Jul.2003, pp.276-278

- 
- [28] Kun-Hui Yi and Bongkoo Kang, "Modified Wilkinson power divider for  $n$ th harmonic suppression", IEEE Microwave and Wireless Components Letters, Vol.13, No.5, May.2003, pp.178-180
- [29] Yung-Jinn Chen and Ruey-Beei Wu, "A wide-band multiport planar power-divider design using matched sectorial components in radial arrangement", IEEE Trans. Microwave Theory and Techniques, Vol.46, Aug.1998, pp.1072-1078

## 研究成果

1. 刘瑞斌、孙从武、鄢泽洪、张小苗、魏文元，“PSO 和 GA 在阵列天线波束赋形中的应用”，《西安电子科技大学学报》，已录待刊。
2. 刘瑞斌、鄢泽洪、张小苗、魏文元，“一种应用于阵列天线波束赋形的新方法”，西安电子科技大学电院学术年会，2005 年。

## 附录 A 遗传算法的模式理论

### A.1 模式概念

定义 1 模式 (schemata): 描述种群中在位串的某些确定位置上具有相似性的位串子集的相似模板 (similarity template)。

看以下事例:

位串	适值
01101	169
11000	576
01000	64
10011	361

从以上事例中可以看出, 凡是以“1”开始的位串, 其适值就高, 而凡是以“0”开始的位串, 其适值就低。

为了描述一个模式, 在用来表示位串的两个字符的字母表  $\{0, 1\}$  中加入一个通配符 “\*”, 就构成了一个表示模式用的三个字符的字母表  $\{0, 1, *\}$ 。因此用三元素字母表  $\{0, 1, *\}$  可以构造出任意一种模式。模式中的 1 与位串中的 1 相匹配, 模式中的 0 与位串中的 0 相匹配, 模式中的 “\*” 可以匹配位串中的 0 或 1。例如模式:  $0*1**$  可以匹配长度为 5、第一位为 0、第三位为 1 的 8 个位串。

定义 2: 模式位数  $O(H)$ :  $H$  中有定义的非 “\*” 位的个数。如  $H=00*1*0$ , 则  $O(H)=4$ 。

定义 3: 模式的定义长度  $\delta(H)$ :  $H$  中最两端的有定义之间的距离:

$O(H)$  和  $\delta(H)$  这两个量为分析位串的相似性及分析遗传操作对重要模式的影响提供了基本的手段。

### A.2 模式定理

#### 一、复制对模式的影响

设在给定时间 (代)  $t$ , 种群  $A(t)$  包含  $m$  个特定模式  $H$ , 记为  $m = m(H, t)$

则可以期望在复制完后, 在  $t+1$  时刻, 特定模式  $H$  的数量为

$$m(H, t+1) = m(H, t)nf(H) / \sum f_i = m(H, t) \frac{f(H)}{\bar{f}} \quad (\text{A-1})$$

若  $f(H) > \bar{f}$ ,  $H$  的数量将增加,  $f(H) < \bar{f}$ ,  $H$  的数量将减少。以上方程 (A-1) 是复制操作对模式  $H$  数量影响的定量描述。

假设  $f(H) - \bar{f} = c \bar{f}$  则

$$m(H, t+1) = (1+c)m(H, t) \quad (\text{A-2})$$

从原始种群开始 ( $t=0$ ), 并假定增长是一个稳定的值, 则有

$$m(H, t+1) = m(H, 0)(1+c)^t \quad (\text{A-3})$$

## 二, 交叉对模式的影响

交叉过程是位串之间的有组织的而又是随机的信息交换。交叉操作对一个模式  $H$  的影响与模式定义长度  $\delta(H)$  有关,  $\delta(H)$  越大, 模式  $H$  被分裂的可能性就越大, 因为交叉操作要随机选择出进行匹配的一对位串上的某一随机位置进行交叉。显然  $\delta(H)$  越大,  $H$  的跨度就越大, 随机落入其中的可能性就越大, 从而  $H$  的存活率就降低。

假设模式  $H$  被破坏的概率为  $Pd$ , 它的存活概率为  $Ps$ , 则有如下关系式

$$Pd = \delta(H)/(l-1) \quad (\text{A-4})$$

$$Ps = 1 - Pd \quad (\text{A-5})$$

在复制、交叉操作之后, 模式  $H$  的数量为

$$m(H, t+1) = m(H, t) \frac{f(H)}{\bar{f}} Ps \quad (\text{A-6})$$

设交叉的概率为  $Pc$ , 则

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - Pc \frac{\delta(H)}{l-1}] \quad (\text{A-7})$$

## 三, 变异对模式的影响

变异是对位串中的单个位置以概率  $Pm$  进行随机替换, 因而它可能破坏特定的模式。一个模式  $H$  要存活, 意味着它所有的确定位置都存活, 因此, 由于单个位置的基因值存活的概率为  $1-Pm$ , 特定模式的存活率为

$$(1-Pm)^{O(H)} \quad \text{一般情况下, } Pm \leq 1,$$

综合考虑复制, 交叉和变异操作的共同作用, 则模式  $H$  在经历了复制、交

叉、变异操作之后,在下一代的数量表示为

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - Pc \frac{\delta(H)}{l-1}] [1 - O(H)Pm] \quad (A-8)$$

上式可以近似表示为:

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - Pc \frac{\delta(H)}{l-1} - O(H)Pm] \quad (A-9)$$

有以上分析可以得到如下定理

定理 1: 定义长度短的、确定位数少的、平均适值高的模式数量将随着世代数的增加呈指数增长。这个结论称为模式理论或遗传算法的基本定理。

根据模式理论,随着遗传算法的一代一代地进行,那些定义长度短的、位数少的、高适值的模式将越来越多,因而可期望最后得到的位串的性能越来越得到改善,并最终趋向全局的最优点。

遗传算法有效处理的模式数量

当位串长度为  $l$  时,一个包含  $n$  个位串的群种中含有的模式个数在  $2^l \sim n2^l$  之间,总模式数的下限值  $n_s$  为:

$$n_s = cn^3 \quad (A-10)$$

### A.3 遗传算法实现中的基本问题

在遗传算法中,除了复制、交叉、变异基本操作外,还必须考虑目标函数到个体适值的映射、适值调整、编码原则和多个参数编码映射方法等基本问题。

一, 目标函数值到适值的映射

适值是非负的,任何情况下总希望越大越好;而目标值有正有负,甚至可能是复数。因此,如求最大值对应点时,目标函数和适值变化方向相同,求最小值对应点时,变化方向恰好相反,目标函数值越小的点,适值越大。因此,存在目标函数值向适值映射的问题。

首先保证映射后的适值是非负的,其次目标函数的优化方向对应于适值的增大方向。

对最小化问题,一般采用如下适值函数  $f(x)$  和目标函数  $g(x)$  的映射关系:

$$f(x) = \begin{cases} g(x) - c_{\max}, & g(x) - c > 0_{\max} \\ 0, & \text{其他} \end{cases} \quad (A-11)$$

其中  $c_{\max}$  可以是一个输入参数,或是理论上的最大值,或是到目前所有代(或最近



的  $k$  代) 之中见到的  $g(x)$  的最大值, 此时  $c_{\max}$  随着代数会有变化。而对于最大化问题, 一般采用下述方法:

$$f(x) = \begin{cases} g(x) - c_{\min}, & g(x) - c_{\min} > 0 \\ 0, & \text{其他} \end{cases} \quad (\text{A-12})$$

其中  $c_{\min}$  既可以是输入值, 也可以是当前最小或最近  $k$  代中的最小值。指数函数方法:

$$\begin{aligned} f(x) &= c^y \\ y &= g(x) \end{aligned} \quad (\text{A-13})$$

其中  $c$  一般取 1.618 (最小化) 或 2 (最大化)。这样既保证了  $f(x) \geq 0$ , 又使  $f(x)$  的增大方向和优化方向一致。

## 二、适值调整

适值线性调整方法如下:

设  $f$  是原适值,  $f'$  是调整后的适值:

$$f' = af + b \quad (\text{A-14})$$

系数  $a, b$  可以通过多种方法选取, 不过, 我们要求无论在何种情况下,  $f'_{\text{avg}}$  应与  $f_{\text{avg}}$  相等, 即应满足的条件为

$$\begin{aligned} f'_{\text{avg}} &= f_{\text{avg}} \\ f' &= c_{\text{mult}} \cdot f \end{aligned} \quad (\text{A-15})$$

其中,  $c_{\text{mult}}$  是最佳种群所要求的期望拷贝数, 这是一个经验值, 对于一个不大的种群 ( $n=50 \sim 100$ ) 来说, 可在 1.2~2 的范围内取值。

线性调整法的遗传算法的后期可能产生一个问题是: 一些个体的适值远远小于平均适值与最大值, 而往往平均适值与最大适值又十分接近,  $c_{\text{mult}}$  的这种选取方法将与原始适值函数伸展成负值。解决的方法是: 当无法找到一个合适的  $c_{\text{mult}}$  时, 我们仍保持  $f'_{\text{avg}} = f_{\text{avg}}$ , 而将  $f_{\min}$  映射到  $f'_{\min} = 0$ 。

## 三、编码原则

遗传算法参数编码原则有两个: 深层意义上的建筑块原则和最小字母表原则。后者是一种应用广泛的实用原则。最小字母表原则要求选择一个使问题得以自然表达的最小字母表进行编译。