

文章编号 :0253-2239(2002)01-0074-05

基因算法在光学系统优化中的应用*

张丽琴¹⁾ 王涌天¹⁾ 李 林¹⁾ 程雪岷¹⁾ 威克特·科特里尔²⁾

(1),北京理工大学光电工程系,北京 100081

(2),俄国科学院图像处理系统研究室,萨马拉 4333001,俄罗斯)

摘要: 将基因算法移植于光学系统的优化中,构造了相应的数学模型,并编程实现了算法。实验证明基因算法可以克服以往优化算法不能自动增减光学面(元件)的缺点,从而提高了光学系统设计的智能化程度。

关键词: 光学设计;优化;基因算法;选择;杂交;突变

中图分类号: TN202 文献标识码: A

1 引 言

当前大多数的光学计算机辅助设计(CAD)软件采用阻尼最小二乘法或适应法等局部优化算法对光学系统进行优化设计,这些算法在找到解空间中某一部局极小值后就停滞不前,结束优化过程,得到局部最优而非全局最优的设计结果。为了解决局部极小值问题,提出了各种全局优化算法。全局搜索^[1]和模拟退火^[2]等全局优化算法解决了优化过程中的局部极小值问题。但是这些算法具有耗用优化时间随变量数量的增加而呈指数性增长以及由于随机抽样而容易得到一些无法实现的系统(如出现负边、负中心厚度等)等缺点,只能用于简单系统的分析。Isshiki^[3]提出的逃逸函数算法通过对评价函数加一个逃逸函数的方法,实现了自动跳出局部极小值继续寻找其它更好的光学结构,这种算法克服了全局搜索和模拟退火的缺点。以上提到的光学设计优化算法都是在选定了初始光学结构的基础上,只是对光学结构参数进行优化选择,在优化过程中不能够自动改变光学面数,其光学系统设计的智能化程度还不是很高。

基于自然界中“适者生存”原理的基因算法是近年来迅速发展起来的一种新的优化技术,目前该算法已经成功地应用于音乐创作、基因合成、大规模集成电路技术、战略计划和机器学习等多个领域^[4]。从理论上讲将该算法应用于光学设计当中就可以实现光学面的自动增减^[5,6]。但一段时间内该算法在智能化光学系统设计中并未达到实用化。本文阐述

了将基因算法移植于光学设计的优化过程,包括构造数学模型和算法的编程实现等,并通过实验证实该算法是提高光学系统设计智能化程度的有效工具。

2 基因算法用于光学设计的基本原理

在基因算法中,将被优化的每个光学系统结构称为一条染色体,每条染色体被编码成为一个二进制字符串,若干条染色体组成一代,第一代是根据某些条件随机初始化生成的。下一代通过使用基因算子选择、杂交和突变当前代的字符串产生。这个过程重复进行,直到满足基因算法的结束条件。标准的基因算法如图 1 所示。下面给出基因算法实现过程中涉及到的几个细节问题的简单描述。

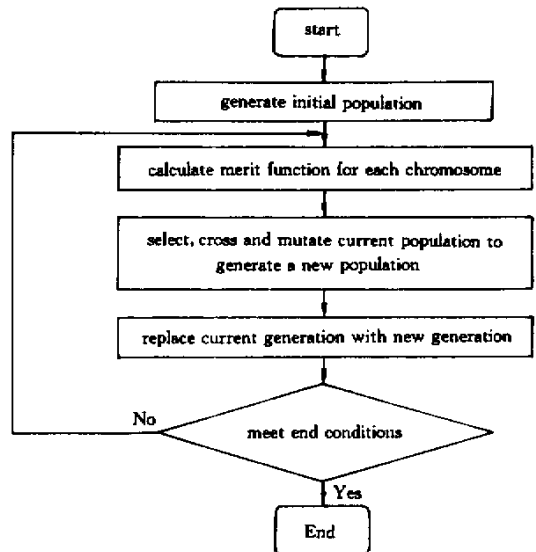


Fig.1 Flow chart of genetic algorithm

* 国家自然科学基金(69978001)资助课题。

1) 编码机制

在基因算法中,定义一组零光焦度的算子。选择了透镜结构以后,从定义的算子中选择一组适用于该透镜结构的算子,选择的算子包含在二进制字符串 $A = a_1 a_2 \dots a_n$ 中,其中 A 称为染色体, n 代表选择的算子个数, $a_i (i = 1, 2, \dots, n)$ 称为算子,取值为 1 或 0,若值取为 1,则代表第 i 个算子起作用;反之,则第 i 个算子不起作用。

2) 适度函数(fitness function)

适度函数 f ,即被优化的函数,提供了评价每个字符串的机制。适度函数有多种取法,通常取 $f = 1/\phi$, ϕ 为评价函数(MF:merit function)。

3) 选择机制

对一些性能较好的染色体(光学结构)进行选择 and 复制。通过选择,保留了较好的染色体,一些不是很好的染色体被消除。

4) 杂交机制

继选择之后,进行杂交操作,它是基因算法的关键操作。从要进行杂交的染色体中随机选择数对,当随机产生的 $r_i \in (0, 1)$ 大于 p_c [p_c 称作杂交率,

$p_c \in (0, 1)$] 基因算法对选择的染色体进行杂交。

5) 突变机制

突变会引起染色体字符串位的偶发性变化。位的突变即位位的反转:0 变成 1 或 1 变成 0。如前所述,杂交率 p_c 控制杂交的机率,类似地,突变率 p_m 控制一个位进行突变的机率。突变的作用是恢复丢失的基因信息。

6) 结束条件

可用以下条件之一结束操作:优化进行了指定的时间;找到了满足给定条件的光学结构;种群中的所有染色体有一定程度的同种性,即大量字符串对应的光学结构有相同的适度函数值。

3 基因算法的数学建模

根据基因算法用于光学系统优化的基本原理,对基因算法建立数学模型如下:

1) 编码机制

首先定义一个正单透镜作为基准透镜,然后定义 8 个零光焦度的算子(如表 1 所示),即每个字符串的长度为 8。

Table 1. Zero power operators

| No. | name | description |
|-----|----------------------------|--|
| 1 | thick corrector 1 | inserting a thick meniscus into an internal air-space, convex to object |
| 2 | thick corrector 2 | inserting a thick meniscus into an internal air-space, convex to image |
| 3 | thick corrector external 1 | adding a thick meniscus in the object space, convex to object |
| 4 | thick corrector external 2 | adding a thick meniscus in the object space, convex to image |
| 5 | flip group | reversing a group of elements about its center |
| 6 | split | splitting a thick element into two thin elements |
| 7 | compound color aberration | replacing a single element by a positive-negative combination having different Abbe values |
| 8 | stop position | changing the stop surface number |

2) 选择和杂交机制

在基因算法中,我们取适度函数 $f = 1/\phi$ (ϕ 为评价函数),在 t 代中,设 f_i 代表第 i 条染色体的适度函数值, \bar{f} 代表适度函数的平均值,则第 i 条染色体被选择用来产生后代的次数为 f_i/\bar{f} 。使用一点杂交、两点杂交和多点杂交的方法对选择的字符串进行杂交。

(i) 一点杂交

在一点杂交方法 c_1 中,假定 l 是字符串的长度, t 代中的一对染色体 $\{x_i, x_j\}$ 从 $(1, l)$ 内随机地选择一个杂交点,将两个染色体杂交点的前后两部分进行交换,形成两个新的字符串,从而产生一对后代。其过程如下所示。

$$\begin{cases} x_p^{i+1} \\ x_{p+1}^{i+1} \end{cases} = c_1 \left(\begin{bmatrix} x_i^t \\ x_j^t \end{bmatrix} \right) = \begin{cases} \left[\begin{array}{cc} \underbrace{x_{i,1} \dots x_{i,k}}_k & \underbrace{x_{j,k+1} \dots x_{j,l}}_{l-k} \\ \underbrace{x_{j,1} \dots x_{j,k}}_k & \underbrace{x_{i,k+1} \dots x_{i,l}}_{l-k} \end{array} \right]^T \\ \left[\begin{array}{cc} \underbrace{x_{j,1} \dots x_{j,k}}_k & \underbrace{x_{i,k+1} \dots x_{i,l}}_{l-k} \\ \underbrace{x_{i,1} \dots x_{i,k}}_k & \underbrace{x_{j,k+1} \dots x_{j,l}}_{l-k} \end{array} \right]^T \end{cases} \quad (1)$$

其中 $i, j, p, p+1 \in [1, 2, \dots, n]; i \neq j, k < l$ 。

(ii) 两点杂交

对两点杂交 c_2 ,一对染色体在 $(1, l)$ 的两个随机的位置分成三个字符串片段,然后通过交换字符串片段来进行。其过程如下所示:

$$\begin{Bmatrix} x_p^{i+1} \\ x_{p+1}^{i+1} \end{Bmatrix} = c_2 \left(\begin{Bmatrix} x_i^t \\ x_j^t \end{Bmatrix} \right) = \begin{Bmatrix} \left[\begin{array}{ccc} x_{i,1} & x_{i,2} & \dots & x_{i,k_1} \\ x_{j,k_1+1} & \dots & x_{j,k_2} \\ x_{i,k_1+1} & \dots & x_{i,l} \end{array} \right]^T \\ \left[\begin{array}{ccc} x_{j,1} & x_{j,2} & \dots & x_{j,k_1} \\ x_{i,k_1+1} & \dots & x_{i,k_2} \\ x_{j,k_1+1} & \dots & x_{j,l} \end{array} \right]^T \end{Bmatrix} \quad (2)$$

其中 $i, j, p, p+1 \in [1, 2, \dots, n]; i \neq j; k_1, k_2 < l; k_1 < k_2$ 。

(iii) 多点杂交

对于多点杂交 c_m , 在一对染色体中随机地选择杂交点的数目和位置, 然后通过交换相应的字符串片段来实现。其过程如下所示。

$$\begin{Bmatrix} x_p^{i+1} \\ x_{p+1}^{i+1} \end{Bmatrix} = c_m \left(\begin{Bmatrix} x_i^t \\ x_j^t \end{Bmatrix} \right) = \bigcup_{k=1}^l \left\{ \left(\begin{Bmatrix} x_{i,k}^t \\ x_{j,k}^t \end{Bmatrix}, \text{if } r = 0 \right) \vee \left(\begin{Bmatrix} x_{j,k}^t \\ x_{i,k}^t \end{Bmatrix}, \text{if } r = 1 \right) \right\} \quad (3)$$

其中 $i, j, p, p+1 \in [1, 2, \dots, n]; i \neq j; r \in [0, 1]$, 是一个随机产生的数。

3) 突变机制

当随机产生的 $r_i \in (0, 1)$ 小于突变率 p_m 时 [$p_m \in (0, 1)$] 则对字符串进行突变操作。

4 设计实例

采用上述数学模型, 我们编制了基因算法软件。使用该算法, 设计了一个光学特性参数如下的照相物镜:

焦距 $f' = 100 \text{ mm}$; 相对孔径 $D/f' = 1/3$; 视场角 $2\omega = 50^\circ$ 。

选定的基准透镜如图 2 所示。表 2 是初始化种群和经过对其选择、杂交与突变后产生的第一代种群对应的字符串和其适应函数值。

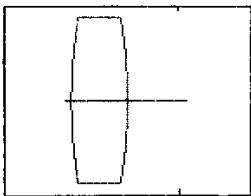


Fig.2 Starting lens

图 3 是初始化种群中第 20 个字符串所对应的光学结构。

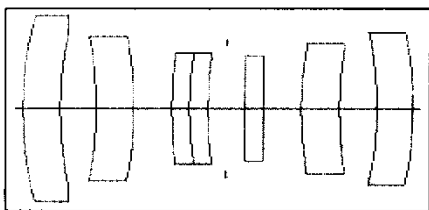


Fig.3 Optical system corresponding to the 20th string in initial population

Table 2. Strings and corresponding fitness function values of initial and first generation populations

| initial population | | | |
|--------------------|-----------------|------------------------|----------------|
| No. | strings | fitness function value | selected times |
| 1 | 0 0 0 0 0 0 0 | 0.000450 | 0 |
| 2 | 0 0 0 1 0 1 0 1 | 0.010774 | 0 |
| 3 | 1 0 0 1 1 0 0 1 | 0.002261 | 0 |
| 4 | 1 1 1 0 0 1 0 0 | 0.952381 | 1 |
| 5 | 1 1 1 1 0 1 1 1 | 6.329114 | 6 |
| 6 | 0 0 1 1 0 0 0 0 | 0.023930 | 0 |
| 7 | 1 0 0 0 0 1 1 | 0.000017 | 0 |
| 8 | 0 1 1 0 0 1 0 1 | 0.292997 | 0 |
| 9 | 0 1 0 0 0 0 1 1 | 0.016805 | 0 |
| 10 | 1 0 1 1 1 1 1 0 | 2.283105 | 2 |
| 11 | 0 0 0 1 0 1 1 0 | 0.038794 | 0 |
| 12 | 1 0 0 0 1 1 1 1 | 0.066542 | 0 |
| 13 | 1 0 0 1 0 1 0 1 | 0.648929 | 1 |
| 14 | 1 1 0 0 1 1 1 1 | 0.369413 | 0 |
| 15 | 1 0 0 1 0 1 1 1 | 0.176087 | 0 |
| 16 | 1 0 0 0 0 0 1 0 | 0.010757 | 0 |
| 17 | 1 1 1 0 0 0 0 0 | 0.432339 | 0 |
| 18 | 1 1 1 1 1 1 1 0 | 3.496503 | 3 |
| 19 | 1 0 1 1 1 0 0 1 | 0.516796 | 1 |
| 20 | 1 1 1 1 0 1 1 1 | 6.329114 | 6 |

| first generation population | | | |
|-----------------------------|-----------------|------------------------|----------------|
| No. | strings | fitness function value | selected times |
| 1 | 1 1 1 1 0 1 0 1 | 1.420455 | 0 |
| 2 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 3 | 1 1 1 1 1 1 1 1 | 2.320186 | 1 |
| 4 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 5 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 6 | 1 1 1 0 0 1 1 1 | 0.140924 | 0 |
| 7 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 8 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 9 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 10 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |
| 11 | 1 1 1 1 0 0 1 1 | 3.861004 | 1 |
| 12 | 1 1 1 0 0 1 1 1 | 0.140924 | 0 |
| 13 | 1 1 1 1 1 1 1 0 | 3.496503 | 1 |
| 14 | 1 0 1 1 1 1 1 0 | 2.283105 | 0 |
| 15 | 1 1 1 1 1 1 1 0 | 3.496503 | 1 |
| 16 | 1 0 1 1 1 1 1 0 | 2.283105 | 0 |
| 17 | 1 0 1 1 1 1 1 0 | 2.283105 | 0 |
| 18 | 1 1 1 0 0 1 0 0 | 0.952381 | 0 |
| 19 | 1 0 0 1 0 1 0 1 | 0.648929 | 0 |
| 20 | 1 1 1 1 0 1 1 1 | 6.329114 | 2 |

初始化种群经过 500 次迭代,可以得到 3 个性能相对较好的系统。其光学结构图和对应的调制传递函数(MTF)曲线图如图 4 所示。由设计实例可以看出:使用基因算法对光学系统进行优化,能够实

现从一个基准透镜开始,通过对代表光学结构的染色体字符串进行选择、杂交和突变来得到不同光学面数的结构,即可以根据需要实现光学面的自动增减。

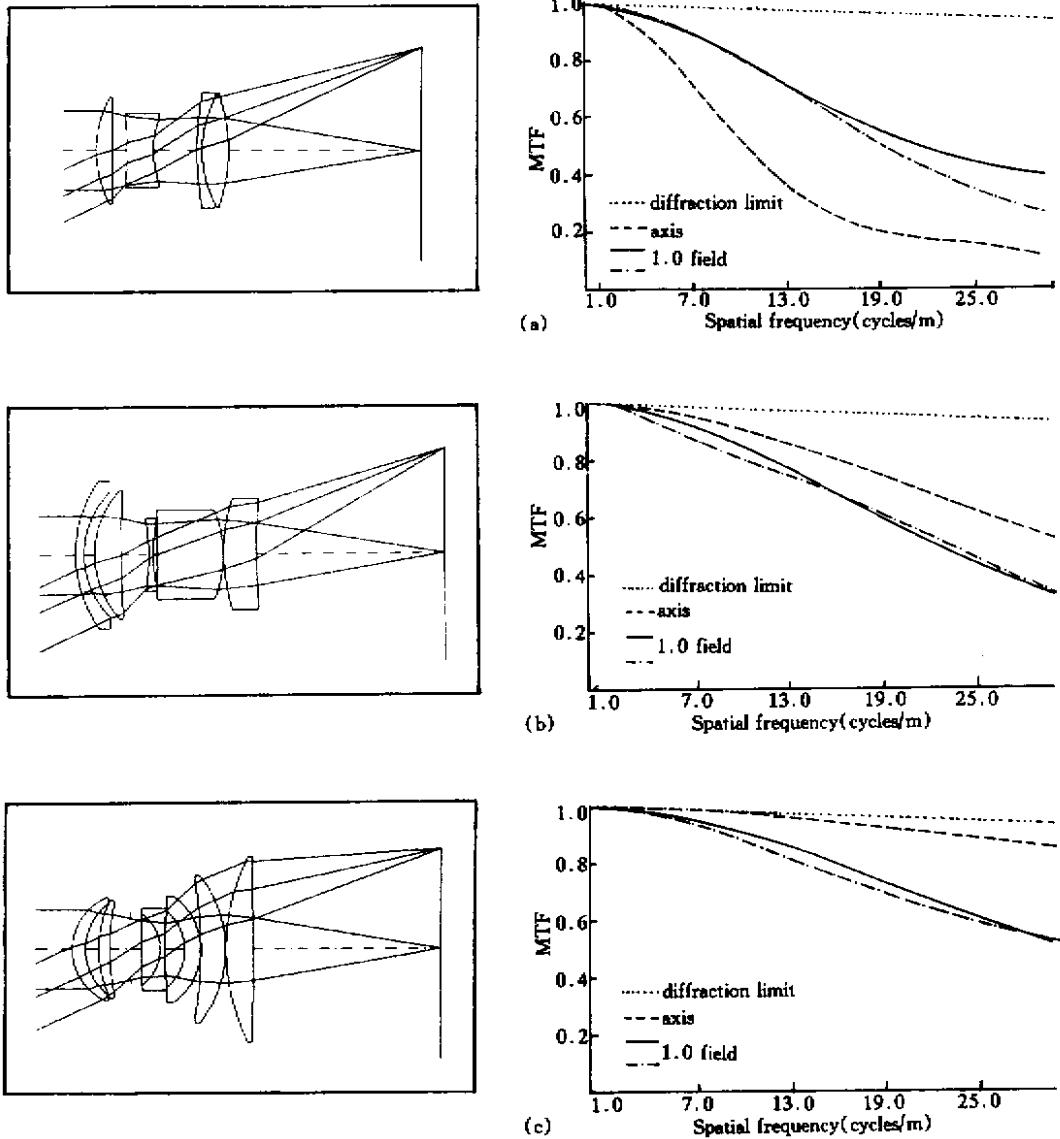


Fig. 4 Different optimization results achieved after 500 iterations by genetic algorithm

结论 基因算法将需要优化的光学结构编码成为二进制字符串组成一代,然后对当前代进行选择、杂交和突变来产生下一代,并重复进行直至筛选到最优结构。这种算法克服了以往优化算法不能实现光学面自动增减的缺点,提高了光学系统设计的智能化程度。实验证明基因算法是一种有效的光学设计优化算法。但在基因算法中,运算量比较大,杂交率和突变率不易控制,下一步的工作是设法解决这些问题,提高基因算法的优化效率。

参 考 文 献

- [1] Brixner B. Lens design and local minima. *Appl. Opt.*, 1981, 20(3) 384~387
- [2] Forbes G, Jones A. Towards global optimization with adaptive simulated annealing. *Proc. SPIE*, 1990, 1354: 144~153
- [3] Isshiki M, Ono H, Hiraga K *et al.*. Lens design: Global optimization with escape function. *Opt. Rev.*, 1995, 2(6) 463~470
- [4] Goldberg D E. *Genetic Algorithms in Search Optimizing and Machine Learning*. Reading, MA: Addison-Wesley, 1989
- [5] Betensky E. Postmodern lens design. *Opt. Engng.*,

1993, **32**(8):1750~1756

optimization in lens design. *J. Modern Optics*, 1997, **44**

[6] Chen Xiaogang, Yamamoto K. An experiment in genetic

(9):1693~1702

Genetic Algorithm Applied to Automatic Lens Design

Zhang Liqin Wang Yongtian Li Lin Cheng Xuemin

(*Department of Opto-Electronic Engineering, Beijing Institute of Technology, Beijing 100081*)

Victor Kotlyar

(*Institute of Image Processing System, Russian Academy of Sciences, Samara 443001, Russia*)

(Received 31 July 2000; revised 20 February 2000)

Abstract: An attempt is made to use the genetic algorithm (GA) in the field of lens design. Mathematical model is constructed, and program code developed. Experimental results show that GA can realize the automatic increase or decrease of optical surfaces in the lens system during the optimization process, which is an important step towards automatic design with artificial intelligence.

Key words: lens design; optimization; genetic algorithm; selection; crossover; mutation