The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/28609

Please be advised that this information was generated on 2019-02-26 and may be subject to change.
Optical design with the aid of a genetic algorithm

Derek C. van Leijenhorst**, Carlos B. Lucasius**, Jos M. Thijssen**

**Department of Theoretical Computer Science, University of Nijmegen, Toernooiveld 1, NL-6525 ED Nijmegen, The Netherlands
**Department of Chemistry, Dalhousie University, Halifax, Nova Scotia, Canada B3H 4J3
**Department of Theoretical Physics, University of Nijmegen, Toernooiveld 1, NL-6525 ED Nijmegen, The Netherlands

Received 29 January 1995; accepted 2 May 1995

Abstract

Natural evolution is widely accepted as being the process underlying the design and optimization of the sensory functions of biological organisms. Using a genetic algorithm, this process is extended to the automatic optimization and design of optical systems, e.g. as used in astronomical telescopes. The results of this feasibility study indicate that various types of aberrations can be corrected quickly and simultaneously, even on small computers.

Keywords: Genetic algorithms; Optimization; Geometrical optics; Optical design; Ray tracing

1. Introduction

The preliminary design of an optical system consisting of lenses and mirrors involves only elementary geometry with idealized components. However, modifications are always necessary in order to correct fuzziness and distortion. These are caused by the remaining defects (aberrations) of which there are six main types (see 'Background' below). The classic approach is to treat each of these defects separately. Such methods often demand intermediate input of optical knowledge. For example, Rutten and Van Venrooij (1988) provide PC software to optimize astronomical optics interactively by the well-known method of ray tracing, but refrain from automatic methods. Modern optimization algorithms in optics are mainly based on mathematical analysis (Frieden, 1980). All these approaches are suitable for local optima and tend to be complex.

This paper describes a novel application of the genetic algorithm. In addition to being conceptually simple and easy to implement, this application also demonstrates that fast automatic optical optimization and design with simultaneous correction of all aberrations is feasible — even on a small personal computer.

From the mathematical framework derived by Holland (Holland, 1975; Goldberg, 1989; Davis, 1991), it follows that a genetic algorithm (GA) samples the search space in a near-exponential way, i.e. extremely efficiently. Being widely appreciated for their efficiency, robustness, and versatility, GAs are normally applied to complex,
large-scale optimization problems. The optical design problem is exactly such a problem: it features many parameters, and a complex fitness landscape with many steep and isolated optima (‘peak’ designs, (Rutten and Van Venrooij, 1988)).

The premise of this study is that a fitness value (merit function (Frieden, 1980)) which measures the quality of an optical system, can be defined using classic ray tracing. In broad terms, a number of light rays from various point sources are used to test the lenses. Due to aberrations, the rays from a particular point do not focus into one image point, but instead this image is smeared out into a so-called spot diagram. Thus, from a measure of the sizes of these diagrams, a fitness value for the optical system can be derived.

It is important to realize that any set of parameters defining an optical system can be used in the GA optimization: refraction indices, the numerical constants defining the shape of the refracting surfaces (spherical, other conic sections, or higher order), the geometry of so-called ‘Schiefspiegler’s’, and so on. The GA is oblivious to their meaning. Also, the GA can, in principle, be adapted so as to make the image distortion-free. We have only implemented some of these options, though, since it was not our intention to develop a complete software environment.

2. Background

Throughout this paper, it will be assumed that the reader is knowledgeable about GAs. As for geometrical optics, only the bare essentials that are necessary to understand our specific implementation of the algorithm will be outlined here. The standard theory and equations can be found in many textbooks (e.g. Van Heel, 1958; Meyer-Arendt, 1984; Pedrotti and Pedrotti, 1987).

Fig. 1 depicts a typical lens system. It consists of \( k \) refracting lens surfaces \( S_1, S_2, \ldots, S_k \), usually parts of spheres, each of which is rotation-symmetric about the optical (\( x \)-) axis. These surfaces separate the media (glass or air) \( M_0 \) (air), \( M_1, \ldots, M_k \), with refractive indices \( n_0 \) (=1), \( n_1, \ldots, n_k \), respectively. These indices depend on the light frequency (color). It is straightforward to include mirrors, since these are formally equivalent to lenses with a negative refractive index.

A point \( P_0 \) in object space sends out light rays from the left, e.g. light ray \( L_0 \). \( L_0 \) is refracted, resulting in rays \( L_1, \ldots, L_k \). (Thus, the optical path is decomposed into sub-paths between the surfaces.) Their initial points \( P_i \) and direction vectors \( v_i \) can be computed successively, using Snell’s Laws and the equations defining the surfaces. For example, \( L_0 \) hits \( S_1 \) in the point \( P_1 \). Snell’s Laws

\[
\begin{align*}
\phi_0 & = \phi_1 \\
\phi_1 & = \phi_k \\
\frac{n_0}{\sin \phi_0} & = \frac{n_k}{\sin \phi_k} \\
\frac{L_0}{P_0} & \quad \text{and} \\
L_0 & \quad \text{is refracted,} \\
L_1 & \quad \text{resulting in rays} \\
& \quad \text{Thus, the optical path} \\
L_k & \quad \text{is decomposed into sub-paths between the surfaces.} \\
& \quad \text{Their initial points} \\
& \quad \text{using Snell’s Laws} \\
& \quad \text{defining the surfaces. For} \\
& \quad \text{example,} \\
& \quad \text{Snell’s Laws}
\end{align*}
\]
state that the refracted ray $L_1$ is in one plane with $L_0$ and the normal of $S_1$ in $P_1$, whilst the angles $\phi$ and $\phi'$ satisfy

$$n_0 \sin \phi_0 = n_1 \sin \phi_1.$$ 

The computation of $L_1$ is called an exact ray trace. This is done for several rays from $P_0$. $P_0$ may lie at infinity, in which case only the direction (incidence angle) of these rays plays a role.

There exists an 'ideal,' shape preserving, first order approximation of the exact ray trace, known as the paraxial mapping; it forms the basis of the theory of geometrical optics as it is often taught in high school. This approximation is valid near the optical axis. The corresponding ray trace — either a translation or a refraction at a surface — can be described by a product of two $2 \times 2$ matrices. Paraxially, all rays coming from any point $P_0$ in object space intersect in one image point in image space, and the plane through $P_0$ perpendicular to the optical axis is mapped to a similar, flat image plane. In exact ray tracing, the best approximation of this image is curved.

The design of telescopes, photographic lenses, etc., is often based on the paraxial approximation followed by analytical calculations or exact ray tracing using spot diagrams. A spot is the point where $L_k$ hits a surface (e.g. the paraxial image plane) in the exact ray trace. In practice, due to optical aberrations (lens errors) the rays from a point in object space do not focus into one image point in any plane. The classical aberrations are generally known as: chromatic aberration, astigmatism, coma and spherical aberration. Chromatic aberration is the phenomenon that different light colors are refracted in different ways by the media, resulting in colored fringes in the image. The standard way to correct this is by combining different glasses so that for two or three fixed wavelengths the chromatic aberrations cancel. The other three aberrations are geometric by nature. In addition, field curvature (a curved image plane) and distortion are present. An image is distortion free in a region if the magnification in this region is independent of the distance to the axis. A more detailed, theoretical discussion on this subject need not be given here, since any set of mentioned aberrations can be corrected simultaneously by the GA.

In the exact ray trace, different rays from $P_0$ hitting a plane (which may be curved) yield a collection of spots — a spot diagram. It is our aim to optimize the optical system in such a way that these spot diagrams, in some plane, are as small as possible. From diffraction theory it follows that the smallest useful size is given by the so-called Airy disk (Rutten and Van Venrooij, 1988). For astronomical objects (to which we shall restrict ourselves) this is $2.44 \lambda f/D$ nm, with $\lambda$ the wavelength in nm (e.g. 555 nm for green light) and $D$ the diameter of the front lens (i.e. stops and diaphragms are not considered).

### 3. Method and materials

#### 3.1. The genome

The lens parameters, i.e. the parameters describing the lens system illustrated in Fig. 1, are all binary encoded. That is to say, each parameter is represented as a finite segment of bits, or bitfield for short. The bitfields ('genes') are placed contiguously, in a predefined order, to form a bitstring in the population. For reasons given below, the placement or order will not affect the search performance of our GA implementation. The following symbols will be used to denote the lens parameters:

$$x_1, R_1, a_1, x_2, R_2, a_2, ..., x_n, R_n, a_n,$$

where $|R_i|$ is the radius of the $i$th (spherical) surface $S_i$ with center $x_i + R_i$; the sign of $R_i$ determines whether $S_i$ is convex or concave to the left. The $i$th medium $M_i$ is described by its refraction indices $n_i, n'_i, n''_i$ at three different wavelengths. $M_0$ is air, and therefore omitted. In practice, only a few different glass types are used, i.e. only a few (discrete) values for $n_i, n'_i, n''_i$ need to be taken into account. For this reason, these parameters can be encoded as a small integer, $a_i$ — the address of the $i$th glass in a table. This 'data compression;' implemented as an option in our GA, compresses the bitstrings and thus reduces the search space for the GA in a meaningful way.

#### 3.2. The fitness function

Two basic fitness criteria will be considered. The
first criterion quantifies the global spot size, which must be minimized. The second criterion accounts for the flatness of the image plane, which must be maximized. Both criteria can be combined into a more sophisticated criterion, which can then be used for bi-objective optimization.

An exact ray trace is performed for bundles of \( N \) parallel rays, the incidence angle of each bundle defining a test point \( P_0 \) (Fig. 1) at infinity. The rays from each bundle hit the front surface \( S_1 \) in equidistant circles around the center of this surface. This procedure is carried out for three wavelengths in order to correct chromatic aberration; so in fact \( 3N \) rays are used.

In a plane, the size \( D \) of a \( 3N \)-spot diagram from one test point is defined as the root mean square of the distances \( d_{ij} \) between spots \( i \) and \( j \). As there are \( 3N(3N - 1)/2 \) pairs \((i,j)\), we obtain:

\[
D = \left[ \frac{2}{3N(3N - 1)} \sum_{i<j} (d_{ij})^2 \right]^{1/2}
\]

For each angle of incidence, the program determines the \( x \)-coordinate (Fig. 1) where \( D \) is minimal. This is done by differentiating \( D \) w.r.t. \( x \); note that \( d_{ij} \) is a linear function of \( x \). The optimal position depends on the angle of incidence, if we allow a curved image plane. In the case of a flat image plane, \( \textit{all} \) the rays leaving the last optical surface are used to calculate the axial position of the plane where the sum of the different spot diagram sizes becomes minimal.

The fitness function is defined as the reciprocal of the quantity \( D \) in the image plane, summed over the various angles of incidence, and multiplied by the number of rays passing through the design (in order to preserve the light intensity at the edge of the image). The ranges of the fitness function's parameters \( R_i \) and \( x_i \) must be specified by the user of the GA. The number of optical surfaces and their diameters are kept fixed during calculations, as are the axial position of the paraxial image, the number and maximum of the different incidence angles, the number of rays in the circles of incoming rays, and the table of refraction indices.

3.3. General provisions

It can not be ruled out that among the designs that will come up during the optimization, there will be some for which the order of the surfaces is not preserved, surfaces intersect within the prescribed diameter, or one of their radii is less than half that diameter. When this happens, the fitness is set to a small constant (0.15). Also, the program will tend to generate rather flat surfaces at a great distance from the focal position, since these produce small aberrations. To keep the design within bounds during the optimization, the front surface is fixed at the origin \( x = 0 \) and the paraxial image (focal plane) is kept at \( x = I > 0 \). Each design in a new GA generation is linearly rescaled to these bounds.

3.4. Configuration

The GA was built with the aid of the software library GATES — Genetic Algorithm Toolbox for Evolutionary Search (Lucasius and Kateman, 1994c). GATES provides genetic shells (GA prototypes) for three classes of problems, and the one for numerical parameter estimation problems was used.

The configuration adheres closely to the traditional 'simple genetic algorithm' (Goldberg, 1989) based on binary encoding and fitness proportional reproduction. In order to stabilize competition between the bitstrings, GATES' sigmoidal fitness scaling provision was used each time before the reproduction procedure. Furthermore, the bitfield sizes for the respective lens parameters were allowed to be different, thus enabling different resolutions for each lens parameter separately; this is reasonable because the lens parameters are of different natures (categories) and it is reasonable to expect that the nature of a particular lens parameter will determine how sensitive the fitness will be to changes in the value of that parameter.

The following modifications are well-grounded in the literature, and have indeed proven to be useful in our application: the binary encoding is interpreted as \textit{Gray coding} (Caruana and Schaffer, 1988) in order to avoid Hamming gaps; thus, the complexity of the fitness landscape is reduced. Also, \textit{elitism} — guaranteed survival of a small fraction of the best individuals — is used; this improves the survival of promising schemata (building blocks), which is essential to good overall performance. Finally, bit-level \textit{uniform...}
Table 1
Control parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutation probability:</td>
<td>1%</td>
</tr>
<tr>
<td>B-UX probability:</td>
<td>90%</td>
</tr>
<tr>
<td>B-UX bit swap:</td>
<td>30%</td>
</tr>
<tr>
<td>Resolution for lens radii:</td>
<td>10 bits</td>
</tr>
<tr>
<td>Resolution for lens position:</td>
<td>10 bits</td>
</tr>
<tr>
<td>Resolution for lens type:</td>
<td>3 bits</td>
</tr>
</tbody>
</table>

crossover (B-UX (Lucasius and Kateman, 1994a and 1994c)) is used instead of the more traditional n-point crossover; this will eliminate positional bias (Syswerda, 1989; Eshelman, Caruana and Schaffer, 1989; Spears and DeJong, 1991), i.e. the search performance will not be affected by the order in which the bitfields are placed in each bitstring. It is a well-established fact that positional bias will, as a rule, seriously degrade performance when the problem parameters are strongly correlated, such as is the case here. Disruptive effects of B-UX are counterbalanced by using a comparatively high selection pressure.

The control parameters of the genetic routines were set as shown in Table 1.

Although poor precision in the end result is widely considered as a drawback of GAs, a simple, easy-to-implement remedy for this problem exists. It is accomplished by sequencing a number runs, as follows (Whitley et al., 1991; Lucasius and Kateman, 1993 and 1994a). For the first run, an educated guess at the solution and at a hypercube symmetrically around it is made. The hypercube represents the search volume containing the search grid, which is defined by the respective sizes of the bitfields. After a number of generations, a new estimate of the solution is obtained. This is then used as an initial guess in a second run, with

Fig. 2. Telescope objectives used as test cases.
the sizes of the bitfields unaffected but with a smaller hypercube, thus yielding a higher resolution in the decoded real space; the shrink factor can be set on a per-bitfield basis.

This precision refinement strategy (incremental focusing schedule) is repeated as often as one deems necessary. Although automating it should in principle not be difficult, in this pilot study we opted for manually updating the input for each run.

The rescaling strategy mentioned in the 'General provisions' section may also be expected to improve performance. Heuristically, if one fixes the image plane \( J \), the designs of high fitness (small spot diagrams) lie in a narrow region of the multidimensional parameter space — namely, close to the hyperplane of designs with paraxial focus in \( J \). Crossing over two of these might then result in offspring far from this hyperplane. By our strategy, such descendants are automatically moved to lower ranks in the fitness domain.

3.5. Possible extensions

In order to control distortion, we suggest the following procedure. Conceptually, a sufficiently fine rectangular grid \( G \) of test points in the object space will be imaged by an exact ray trace in a plane \( P \) near the paraxial image plane of minimal average spot size. By rotational symmetry, \( G \) can actually be replaced by a few axial points. Let \( G' \) be the linear scaling in \( P \) that best matches the image of \( G \) in the exact ray trace. A new fitness function can be defined as the minimum (or a weighted sum) of the previously defined minimal spot size fitness and the reciprocal of the maximal distance of the test point images from their counterparts in \( G' \). A refinement which might increase the speed of the program is: dynamic adaptation of the number of rays from each test point during the genetic search.

4. Results

The GA was tested on two astronomical telescope objectives (Fig. 2). The typical size of the population in the examples given below was 100 individuals, and the number of generations was equal to 50. The three light colors used in all ray traces were e-, C- and F-light with wavelengths of 486.13 nm, 546.07 nm, and 656.27 nm, respectively. Five test points were used (Figs. 4, 5, 6).

4.1. The doublet

The first design is the classic achromatic crown
Spot Diagram Curved Plane Design

WaveL. = 486.13  WaveL. = 546.07  WaveL. = 656.27

Fig. 4. Spot diagrams for the optimized doublet, curved image plane. Spot diagrams obtained for the doublet (third run).

glass/flint glass doublet (left part of Fig. 2), the so-called ‘apoclaas’ design (Rutten and Van Vernoij, 1988). The glass types used are commercially available as: FK 51 (front lens, left) and KzFS N2 (back lens). These were fixed during the GA runs. The image field of this simple design is curved, so that the performance as an eye telescope is substantially better than as a camera. We restrict ourselves to the curved image plane case. There are four optical surfaces and eight parameters in the fitness function. The focal distance is fixed at 2000 mm.
Table 2 shows the ranges of the lens parameters as used in the first GA run. In this table, $t_i$ is the $x$-coordinate of the center of the surface $S_i$ and $R_i$ its radius, all in mm. Although the initial ranges are very large, the GA was capable of optimizing the design.

In two subsequent runs, the ranges were narrowed. The maximal fitnesses obtained in the three runs were 1179, 4859, and 5135, respectively. The evolution of the fitness for the three runs is shown in Fig. 3.

The parameter values vary considerably during
Spot Diagram Flat Plane Design

Waveλ = 486.13  Waveλ = 546.07  Waveλ = 656.27

Angle = 0.0300
Angle = 0.0225
Angle = 0.0150
Angle = 0.0075
Angle = 0.0000

Fig. 6. Spot diagrams for the optimized doublet with field flattener, flat image plane.

Table 2
Initial parameters and ranges for the doublet

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>1000 ± 700</td>
<td>$t_1$</td>
</tr>
<tr>
<td>$R_2$</td>
<td>-1000 ± 700</td>
<td>$t_2$</td>
</tr>
<tr>
<td>$R_3$</td>
<td>-1000 ± 700</td>
<td>$t_3$</td>
</tr>
<tr>
<td>$R_4$</td>
<td>-2000 ± 1000</td>
<td>$t_4$</td>
</tr>
</tbody>
</table>

Table 3
Parameters of the optimized doublet

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>1204.9085</td>
<td>$t_1$</td>
</tr>
<tr>
<td>$R_2$</td>
<td>-487.3933</td>
<td>$t_2$</td>
</tr>
<tr>
<td>$R_3$</td>
<td>-496.7340</td>
<td>$t_3$</td>
</tr>
<tr>
<td>$R_4$</td>
<td>-2435.0087</td>
<td>$t_4$</td>
</tr>
</tbody>
</table>
the GA run and the fitness fluctuates around a value of about 4700. Changing the radius of the second or the third surface by about 1% causes the fitness to drop to about 1400, consistent with our earlier remark about a narrow region of optimal fitness in parameter space. However, the fitness is less sensitive to other parameters, such as the curvature of the front and back surface.

The parameters of the optimized design resulting from the third run, are given in Table 3 (with notations as in Table 2); the spot diagrams are shown in Fig. 4. Close to the optical axis, the spots are of the order of the Airy disk size. For incidence angles of about 0.02, they become much larger. However, for a 2000 mm lens these spots lie at a distance of 40 mm from the optical axis, so the region where we get a sharp image is quite large.

### 4.2. The doublet with field flattener

The second example consists of four lenses: conceptually a doublet together with a field flattener, a so-called cemented doublet (Rutten and Van Venrooij, 1988). The glass types used in this case are, from left to right: FK 51, KzFS N2, BK7, SF 55. Initially, these were kept fixed during the GA run.

The field flattener and front doublet were optimized simultaneously. There are seven optical surfaces and fourteen fitness parameters. The optimal result for the curved image plane features a fitness of 10160 (Fig. 5) — much better than in the case of a doublet alone, of course.

Using the appropriate fitness function, it is possible to obtain a flat image plane suitable for photography. The best results are obtained when the (flat) back surface coincides with the image plane, i.e. there are only six optical surfaces (Table 4, with notations as in Table 2).

The maximal fitness obtained is 7625, which is to be compared with 1935 for the single doublet. It appears to be much harder to obtain high performance for a flat image plane than for a curved one (Fig. 6). This is related to the fact that relatively large incidence angles are taken into account.

Finally, when the program was allowed to choose the glasses from a table (i.e. when the integers denoted as $a_i$ are now used) the resulting fitness becomes poorer, typically by a factor two or three. The reason is that only two glass combinations ($A$ and $B$, say) turn out to be useful. Apparently, combination $B$, though worse than $A$, has a better fitness in a large region of the parameter space. This problem can always occur in any GA when there are several attraction basins and a small population is used. All calculations were performed on a Sun Sparc 2 workstation with a clock speed of 70 MHz. (A PC with a lock speed of 50 MHz is in practice about three times slower.) A typical running time for four surfaces, using 100 individuals evolving over 50 generations, is 850 s; and for seven surfaces, only 1000 s. In all cases examined, the memory used was about 400 KB.

### 5. Conclusions

The aim of this pilot study was to establish the feasibility of the genetic algorithm as a tool in the automated design of optical instruments. An in-depth study should involve an extension of the GA method to handle distortion, and a comparison between current design methods and the GA, also for many-lens systems.

The results reveal that, at least for comparatively small systems, the GA method is perfectly feasible on a PC. A salient merit of the GA method is its extreme simplicity. A useful implementation could be a shell on top of commercial software packages already available for semi-professional optical designers, e.g. as provided by Rutten and Van Venrooij (1988). Similarly, the GA method can be incorporated as part of more advanced algorithms in use by the large optical industries, targeted to larger, more complex optical systems.
Acknowledgments

We wish to thank the referees for their kind advice leading to a revision of this paper.

References


