Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2008), *22*, 249–273. Printed in the USA.

doi:10.1017/S0890060408000176

Automated *ab initio* synthesis of complete designs of four patented optical lens systems by means of genetic programming

JOHN R. KOZA,[1] SAMEER H. AL-SAKRAN,[2] AND LEE W. JONES[2]

[1]Biomedical Informatics Program, Department of Medicine, Stanford University, Stanford, California, USA
[2]Genetic Programming Inc., Mountain View, California, USA

(RECEIVED February 20, 2005; ACCEPTED July 20, 2007)

Abstract

This paper describes how genetic programming has been used as an invention machine to automatically synthesize complete designs for four optical lens systems that duplicated the functionality of previously patented lens systems. The automatic synthesis of the complete design is done *ab initio*, that is, without starting from a preexisting good design and without prespecifying the number of lenses, the topological arrangement of the lenses, or the numerical or nonnumerical parameters associated with any lens. One of the genetically evolved lens systems infringed a previously issued patent, whereas the others were noninfringing novel designs that duplicated (or improved upon) the performance specifications contained in the patents. One of the patents was issued in the 21st century. The designs were created in a substantially similar and routine way, suggesting that the approach described in the paper can be readily applied to other similar problems in the field of optical design. The genetically evolved designs are instances of human-competitive results produced by genetic programming in the field of optical design.

Keywords: Automated Design; Computer-Aided Design; Developmental Genetic Programming; Genetic Programming; Human-Competitive Results; Optical Lens Systems

1. INTRODUCTION

An optical lens system is an arrangement of refractive or reflective materials that manipulate light (Smith, 1992, 2000; Fischer & Tadic-Galeb, 2000).

This paper describes how genetic programming has been used as an invention machine to automatically synthesize complete designs for optical lens systems that duplicated the functionality of four previously patented lens systems. The automatic synthesis of the complete design is done *ab initio*, that is, without starting from a preexisting good design and without prespecifying the number of lenses, the topological arrangement of the lenses, or the numerical or nonnumerical parameters associated with any lens. One of the genetically evolved lens systems infringed a previously issued patent, whereas the others were noninfringing novel designs that duplicated (or improved upon) the performance specifications contained in the patents. One of the patents was issued in the 21st century.

The design process for optical systems is more of an art than a science. As Warren J. Smith states in *Modern Optical Engineering* (Smith, 2000, p. 393),

> Optical design is in great measure a systematic application of the cut-and-try process.
>
> There is no "direct" method of optical design for original systems; that is, there is no sure procedure that will lead (without foreknowledge) from a set of performance specification to a suitable design.

A complete design for a classical optical lens system encompasses numerous decisions, including the choice of the system's topology (i.e., the number of lens surfaces and their spatial arrangement and placement), choices for numerical parameters, and choices for nonnumerical parameters.

The topological decisions required to define a lens system include the physical placement of lens surfaces between the object (OBJ) and the image (IMS), decisions as to whether consecutive lenses touch or are separated by air, the nature of the mathematical expressions defining the curvature of

Reprint requests to: John R. Koza, Biomedical Informatics Program, Department of Medicine, Stanford University, Stanford, CA 94305 USA. E-mail: koza@stanford.edu

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

each lens surface (traditionally spherical, but today sometimes aspherical), and the locations and sizes of the field and aperture stops that determine the field of view (FOV) and the maximum illumination of the IMS, respectively.

The numerical choices include the thickness of each lens and the separation (if any) between lens surfaces, the numerical coefficients for the mathematical expressions defining the curvature of each surface (which in turn, implies whether the surface is concave, convex, or flat), and the aperture (semidiameter) of each surface.

The nonnumerical choices include the type of material associated with each lens surface. The materials may be glass, air, vacuum, oil, polymer, or a gas. Each type of material has various properties of interest to optical designers, notably including the index of refraction *n* (which varies by wavelength), the Abbe number *V*, and the cost. Choices of glass are typically drawn from a standard glass catalog for ordinary commercial applications, although it is possible to manufacture custom glasses with specified properties at additional cost.

When optical design is done by humans or by humans with the aid of optimization software, an existing human-created design is frequently the starting point for the new lens system (Smith, 1992, 2000).

Smith (2000, p. 394) identifies four stages in the "ordinary design process" of optical systems by human designers:

> The ordinary design process can be broken down into four stages, as follows: first, the selection of the type of design to be executed, i.e., the number and types of elements and their general configuration. Second, the determination of the powers, materials, thicknesses, and spacings of the elements. These are usually selected to control the chromatic aberrations and the Petzval curvature of the system, as well as the focal length (or magnifying power), working distances, field of view, and aperture. (Choices made at this stage may affect the performance of the final system tremendously, and can mean the difference between success and failure in many cases). In the third stage, the shapes of the elements or components are adjusted to correct the basic aberrations to the desired values. The fourth state is the reduction of the residual aberrations to an acceptable level.

Section 2 provides general background on genetic programming, and Section 3 mentions previous work on optical design in the field of genetic and evolutionary computation.

Section 4 provides general background on the design of optical lens systems, and Section 5 discusses the domain-specific operations used to apply genetic programming to problems of optical design.

Section 6 introduces the four patented lens systems discussed in this paper, Section 7 discusses the preparatory steps used to apply genetic programming to the four problems, and Section 8 presents the results.

Section 9 discusses the use of parallel computing systems for work in genetic programming. Section 10 discusses computer time, Moore's law, and the future of genetic programming for automated design.

Section 10 is the conclusion.

2. BACKGROUND ON GENETIC PROGRAMMING

Genetic programming starts from a high-level statement of what needs to be done and automatically creates a computer program to solve the problem. Genetic programming uses the Darwinian principle of natural selection along with analogs of recombination (crossover), mutation, gene duplication, gene deletion, and mechanisms of developmental biology to breed an ever-improving population of programs (Koza 1990, 1992, 1994*a*, 1994*b*; Koza & Rice 1992; Banzhaf, Nordin, Keller, & Francone, 1998; Koza, Bennett, Andre, & Keane, 1999; Koza, Bennett, Andre, Keane, & Brave 1999; Langdon & Poli 2002; Koza, Keane, Streeter, Mydlowec, Yu, & Lanza 2003*a*, 2003*b*). Additional information on genetic programming may be found in Section 2 of another article in this Special Issue (Koza, this issue).

3. PREVIOUS WORK IN OPTICS USING GENETIC ALGORITHMS AND GENETIC PROGRAMMING

Genetic algorithms (Holland, 1975) have been extensively used for several decades for optimizing the choices of parameters of optical systems having a prespecified number of lenses and a prespecified topological arrangement of the lenses. Jarmo Alander's voluminous *An Indexed Bibliography of Genetic Algorithms in Optics and Image Processing* (Alander, 2000) lists many examples of earlier work.

In a noteworthy paper in 2002, Beaulieu, Gagné, and Parizeau used genetic programming to "reengineer" and improve a preexisting design of a four-lens system (itself produced, as it happens, by a run of a genetic algorithm; Beaulieu et al., 2002). Their approach used functions that incrementally adjusted (additively or multiplicatively) the distance between lens surfaces, the radius of curvature of lens surfaces, and stop location values. The result of their reengineering was an improvement over the best design produced by 11 human teams in an international design competition held at the 1990 International Lens Design Conference.

4. BACKGROUND ON THE DESIGN OF OPTICAL LENS SYSTEMS

A classical lens system is conventionally specified by a table called a *prescription* (or, if the system is being analyzed by modern-day optical simulation software, a *lens file*).

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

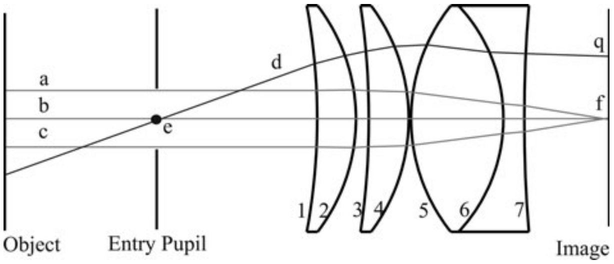


Fig. 1. The Tackaberry–Muller lens system.

4.1. Example of an optical lens system

To aid in explaining optical prescriptions, Figure 1 shows a system (Smith, 2000, p. 508) composed of four spherical lenses intended for use as a telescope eyepiece. Tackaberry and Muller patented this lens system in 1958 (Tackaberry & Muller, 1958), and is an improved design within a subclass of the Konig lens systems patented in 1940 (Konig, 1940). In this optical lens system, the OBJ is shown at the far left of the figure and the IMS is shown at the far right. Curved surfaces 1 and 2 together define the first lens. Surfaces 3 and 4 together define the second lens. Curved surfaces 5, 6, and 7 together define a *doublet* (i.e., two lenses fitting snuggly together). Light rays *a*, *b*, and *c* (so-called *axial rays*) start at the OBJ (at the far left), enter the system in parallel at the entry pupil (EP), and eventually converge (at the far right) to a single point (*focal point* f) on the IMS. Light ray *d* (so-called *chief ray*) starts from the point farthest from the axis on the OBJ plane that is visible through the lens system (*FOV*). This chief ray runs through the center, *e*, of the aperture stop (*EP* here) and to the IMS (at a point *q*, called the *image height*).

4.2. Example of a prescription (lens file)

Table 1 shows a prescription (slightly modified for tutorial purposes) for the Tackaberry–Muller lens system (a telescope eyepiece) of Figure 1. Each row in the table represents a surface. The OBJ appears in the first row, the EP appears in the second row, and the IMS appears in the last row. The other surfaces are consecutively numbered (from 1 to 7). All entries in this table are normalized so that the system has a focal length of 1. As will be seen, a single lens is defined by two surfaces having a material, such as glass, between them.

Table 1. *Lens file for the patented Tackaberry–Muller lens system*

Surface	Distance	Radius	Material	Aperture
OBJ	10^{10}	Flat	Air	
EP	0.88	Flat	Air	0.18
1	0.21900	−3.52361	BK7	0.62
2	0.07280	−1.05274	Air	0.62
3	0.22500	−4.40723	BK7	0.62
4	0.01360	−1.07043	Air	0.62
5	0.52100	1.02491	BK7	0.62
6	0.11800	−0.93493	SF61	0.62
7	0.47485	7.94281	Air	0.62
IMS		Flat		

Some of the information on a given row in a prescription pertains to the surface, whereas other information pertains to the space between the surface and the next surface. For example, in Table 1, the radius of curvature of the surface (column 3) and the aperture (column 5) are characteristics of the surface itself. The distance (column 2) refers to the distance between the surface represented by that row and the surface represented by the next row (i.e., the surface to the right). Similarly, the material (column 4) refers to the material (e.g., air, vacuum, glass, polymer, oil, other material) between the surface represented by that row and the surface represented by the next row.

The Tackaberry lens system has an EP. The distance between the OBJ and the EP is infinite (represented here by the value of 10^{10}). The EP is a flat opaque surface with a small circular hole of semidiameter 0.18 mm, centered at the axis line.

Surface 1 in Table 1 is located at a distance (column 2) of 0.88 from the previous surface (i.e., EP). The value of −3.52361 (column 3) for the radius of curvature of surface 1 indicates that surface 1 is a sector of a sphere with a radius 3.52361. The negative sign indicates that the sphere's center is located to the surface's left. The material (column 4) located to the right of surface 1 is BK7, a commercially available crown glass. The aperture semidiameters for surfaces 1 through 7 (found in column 5) have a uniform value of 0.62 for this particular lens system (a telescope eyepiece).

Surface 2 is located at distance 0.21900 from surface 1. The negative sign of the radius of curvature of surface 2 (−1.05274) indicates that the sphere's center is located to the surface's left. The material to the right of surface 2 is air. Together, surfaces 1 and 2 define a lens of thickness 0.21900 composed of BK7 glass with a concave left surface and a convex right surface. As can be seen in Figure 1, the leftmost surface of this lens is concave and its rightmost surface is convex. Note that surface 2's radius of curvature (1.05274) is smaller than that of surface 1 (3.52361), giving surface 2 a greater curvature.

Surfaces 3 and 4 together specify a lens (composed of BK7 glass) at distance 0.07280 from surface 2 with a concave left surface and a convex right surface.

Surfaces 5, 6, and 7 together define a *doublet* lens. The material to the right of surface 5 is BK7 glass and the material to the right of surface 6 is SF61, a commercially available flint glass. The positive radius of surface 5 (1.02491) and the negative radius of surface 6 (−0.93493) indicate that the left lens of the doublet is biconvex. The negative radius of surface 6 (−0.93493) and the positive radius of surface 7 (7.94281) indicate that the right lens of the doublet is biconcave. The relatively large (absolute) value of the radius of surface 7 (7.94281) means that surface 7 is almost flat.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

The material between surface 7 and the IMS is air and its distance is 0.47485 (the *back focal length*).

The distance of 0.88 between the EP and surface 1 defines the *eye relief* of the eyepiece. Eye relief describes the distance between the eye and the eyepiece lens and is an important design criterion of eyepieces. Eye relief (from the perspective of design constraints) is determined by the FOV and the aperture of the EP. In general, the design tradeoff is that the smaller the aperture of the EP and/or the larger the FOV, the shorter the eye relief. This system here has a half FOV (HFOV) of 19.8°.

The format of lens files used in modern-day optical simulation software can accommodate many complex features not illustrated by the above prescription. For example, the surfaces need not be spherical. Instead, an aspherical surface can be specified by a higher order polynomial or even an arbitrary mathematical function. Such aspherical lenses are becoming increasingly common today because it is now possible to manufacture them economically. Another example of a feature not illustrated by the above prescription is that surfaces can be reflective as well as refractive, thereby permitting light rays to travel in nonsequential paths (not just left to right). In addition, it is possible for a surface to consist of multiple subsurfaces containing various geometric shapes (e.g., circular, polygonal) residing at various physical locations on the surface.

4.3. Analysis of an optical system

Once a classical optical system is specified by means of its prescription (lens file), many of its optical properties can be calculated by tracing the paths of light rays of various wavelengths through the system. Ray-tracing analysis is extremely tedious when performed by hand. Today it is typically performed by optical simulation software. Numerous commercial software packages (e.g., OSLO, Zemax, Code V) and public domain software packages (e.g., KOJAC) are available in the field of optical engineering to aid in this process.

Modern optical simulation software yields a multiplicity of curves describing the system's optical characteristics. For example, Figure 2 shows conventional curves for distortion (Fig. 2a), astigmatism (Fig. 2b), chromatic aberration

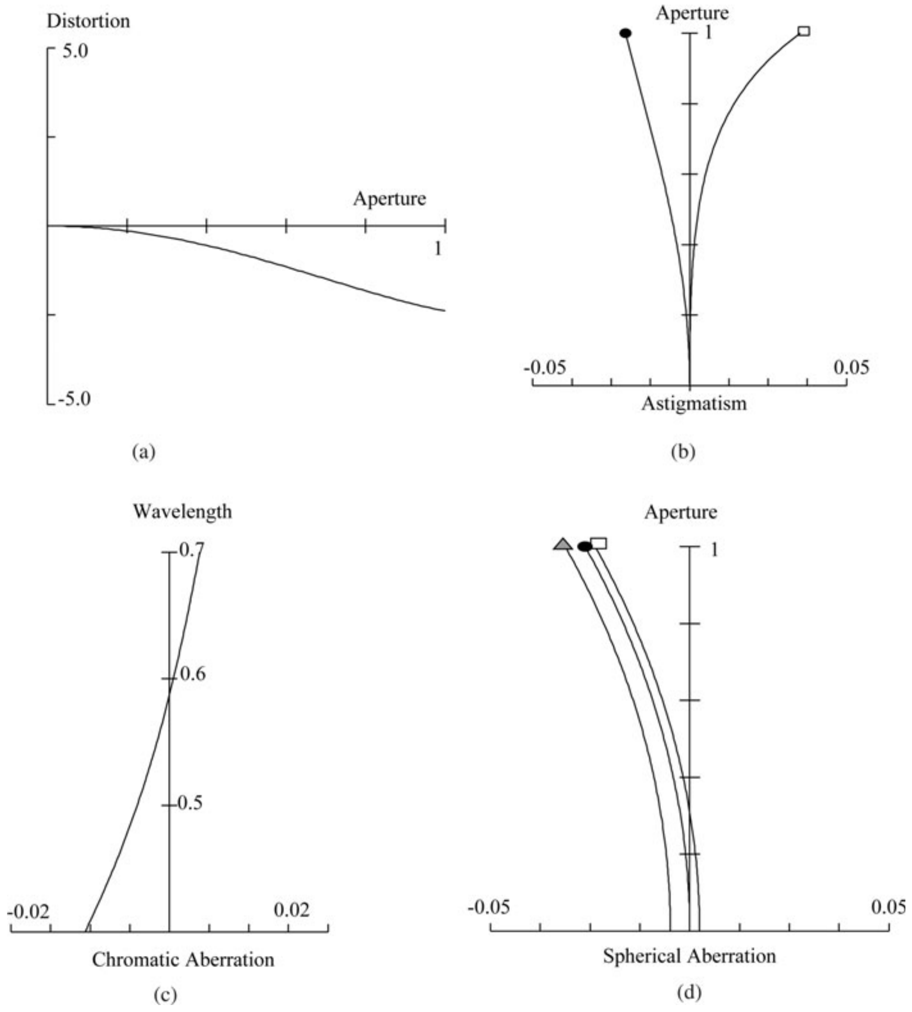


Fig. 2. Characteristics of the Tackaberry–Muller system.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

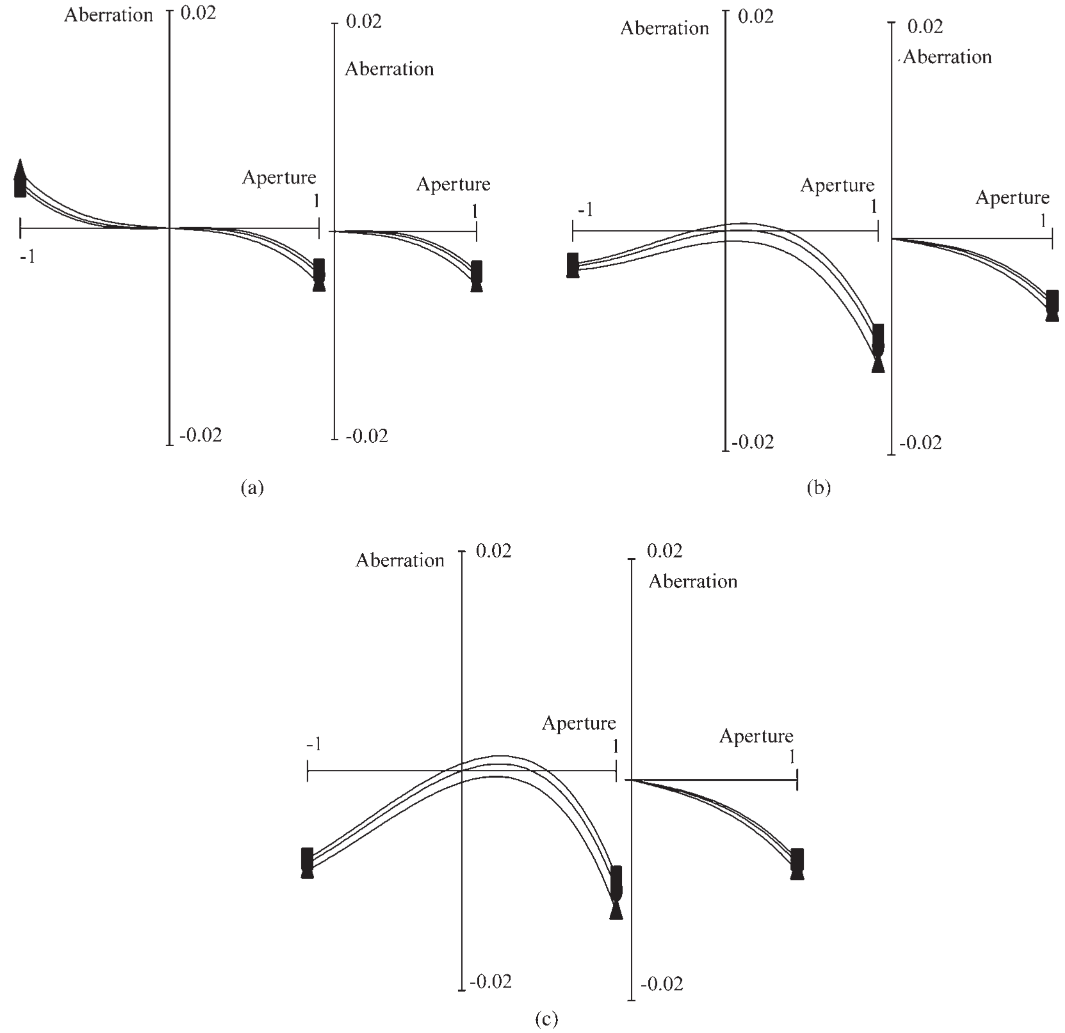


Fig. 3. Additional characteristics of the Tackaberry–Muller system.

(Fig. 2c), and spherical aberration (Fig. 2d) for the Tackaberry–Muller system (Fig. 1).

Figure 3 shows the on-axis ray intercept diagram (Fig. 3a), a partial field ray intercept diagram (Fig. 3b), and the full field ray intercept diagram (Fig. 3c). Within each subfigure, the diagram for the meridional plane (i.e., planes that contain the optical axis and axial rays) is on the left, with the sagittal plane (i.e., skew planes that contain off-axis rays such as the chief ray) on the right.

5. DOMAIN-SPECIFIC OPERATIONS FOR OPTICAL SYSTEMS

Genetic programming is typically applied to design problems using the domain-independent operations such as crossover, reproduction, tree mutation, numerical parameter mutation, and architecture-altering operations. However, when applying genetic programming to a particular field, it may be advantageous (or necessary) to modify these standard domain-independent operations to the special requirements of the field or to augment the standard operations with domain-specific operations. Four such adjustments are useful for the field of optical design.

5.1. Practical limitations on numerical values

When applying genetic programming to a design problem in a particular field, practical considerations usually dictate certain limitations on the range of numeric values that are allowed. In the field of optical design, for example, there is a practical limitation on the radius of curvature of a lens surface and the distance between surfaces. For this work, where the focal lengths of the target systems were normalized to 1 mm, the minimum radius of curvature is -15 and the maximum is $+15$. The maximum thickness (for glass or air) is 1.0 mm. The minimum thickness for air is 0.01 mm and the minimum thickness for glass is 0.1 mm. The minimum aperture is 0.1 mm (where an opaque mounting is added to cradle a lens that would otherwise be hovering in air). The aspheric

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

coefficient terms have a range from −10 to +10 (for optical systems normalized to a focal length of 1).

5.2. Glass mutation operation

The numerical parameter values for distance and the radius of curvature of a lens can each be established using the standard technique of perturbable numerical values. These perturbable numerical values are perturbed during the run using the standard method in genetic programming for numerical parameter mutation. In this standard method, each perturbable numerical value is set, individually and separately, to a random value in a chosen range in the initial generation of a run (generation 0). After generation 0, the perturbable numerical value may be perturbed. The perturbation is usually relatively small. The value to be perturbed is considered to be the mean of a Gaussian distribution. A relatively small preset parameter establishes the standard deviation of the Gaussian distribution. The value to be perturbed is then perturbed by an amount determined by the Gaussian distribution.

However, the standard method for perturbing numerical values cannot be readily used when mutating materials (e.g., types of glass). In optical design, each type of material is characterized by a set of wavelength-dependent indices of refraction, its Abbe number, and its cost. The Abbe numbers and indices of refraction are interrelated, and therefore cannot be independently varied with total freedom. The 199 commercially available glasses in the Schott catalog reside in a relatively small and compact crescent-shaped area occupying only about 27% of the area of the rectangle bounded by the extreme values of n and V for the catalog, namely, $1.46 < n < 2.02$ and $20 < V < 77$. The optical designer usually cannot freely choose numerical values for the materials to be used, but must, instead, limit the choices to the relatively small number of commercially available types of materials. This limitation is not just a matter of the higher cost of manufacturing a custom type of glass because it may not be possible to manufacture a type of glass with an arbitrarily chosen Abbe number and a set of arbitrarily chosen indices of refraction for particular wavelengths. Accordingly, our mutation operation for materials changes one type of material in the chosen catalog to another type of material in the catalog whose properties are nearby in the multidimensional space of properties.

5.3. Toroidal mutation operation for the radius of curvature

A flat surface can be viewed as a spherical surface with a very large positive or negative radius of curvature. That is, a very large positive or negative radius represents the same thing. Thus, it is advantageous to slightly modify the standard method used in genetic programming for numerical parameter mutation when perturbing the radius of curvature. Accordingly, our numerical parameter mutation operation for curvatures operates in a toroidal way (wrapping +15 to −15) when it is applied to a terminal representing the radius of curvature.

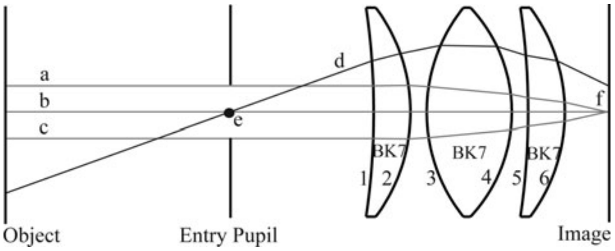


Fig. 4. Lens system before lens-splitting operation.

5.4. Lens splitting operation

A lens-splitting operation appears to be useful for the field of optical design. The lens-splitting operation is performed on a single parent selected probabilistically from the population based on its fitness. The lens-splitting operation replaces one randomly picked lens with two new lenses. Figure 4 shows an illustrative lens system, and Figure 5 shows the result of applying the lens-splitting operation. The thickness of each of the two new lenses is half of the thickness of the original lens. The radius of curvature of the first surface of the first new lens is set equal to the radius of curvature of the first surface of the original lens. The second surface of the first new lens (which is the same as the first surface of the second new lens) is flat. The radius of curvature of the second surface of the second new lens is set equal to the radius of curvature of the second surface of the original lens. When the original lens is a single lens (as is the case in Fig. 5), the result of the lens splitting operation is a doublet lens (as shown in Fig. 5). The lens-splitting operation is intended to be optically neutral (and, hence, does not change the fitness of the lens system involved). The one exception occurs if half of the thickness is less than the minimum permissible lens thickness. In that event, the thickness of each new lens is set to the minimum (and the overall length of the lens system is slightly increased). Note that because of the toroidal behavior of the numerical parameter mutation operation, the newly created flat surface has an equal probability of being perturbed to a negative or positive radius of curvature when it is first mutated.

The motivation for the lens-splitting operation is that the insertion of additional surfaces or lenses by means of

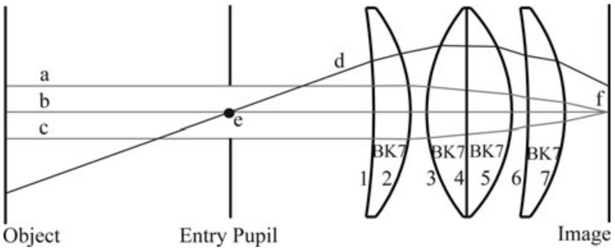


Fig. 5. Lens system after lens-splitting operation.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

crossover rarely yields an improved individual. The lens-splitting operation creates an offspring that almost always has the same (reasonably good) fitness as its parent. It thus introduces topological diversity without changing fitness. Subsequent glass mutations or numerical mutations of the distance or radius of curvature can then be done gradually.

6. FOUR PATENTED LENS SYSTEMS

This paper is concerned with four previously patented eyepieces:

- the 1985 Nagler patent (Nagler, 1985),
- the 1968 Scidmore patent (Scidmore, 1968),
- the 2000 Koizumi–Watanabe patent (Koizumi & Watanabe, 2000), and
- the 1940 Konig patent (Konig, 1940) and the related 1958 Tackaberry–Muller patent (Tackaberry & Muller, 1958).

The Tackaberry–Muller patent (Tackaberry & Muller, 1958) cites the 1940 Konig patent (Konig, 1940) and the former is a special case of the latter.

Each patent document contains the inventor's high-level design goals, a detailed specification of the invention, one or more independent claims (usually with an associated prescription), and a diagram of the optical lens system.

7. PREPARATORY STEPS FOR APPLYING GENETIC PROGRAMMING TO OPTICAL DESIGN

Genetic programming starts from a high-level statement of the requirements of a problem and attempts to produce a computer program that solves the problem.

The human user communicates the high-level statement of the problem to the genetic programming system by performing certain well-defined preparatory steps.

The five major preparatory steps for genetic programming require the human user to specify

1. the set of primitive functions for the program being evolved,
2. the set of terminals (e.g., the independent variables of the problem, zero-argument functions, perturbable numerical values, and random constants) for the program being evolved,
3. the fitness measure (for explicitly or implicitly measuring the fitness of individuals in the population),
4. certain parameters for controlling the run, and
5. the termination criterion and method for designating the result of the run.

The first two preparatory steps specify the programmatic ingredients (i.e., the functions and terminals) that are available to create programs that represent optical lens systems. The representation scheme for optical lens systems is defined by the functions, terminals, and the constrained syntactic structure specifying the allowable combinations of the functions and terminals. The run of genetic programming is a competitive search through a space of programs composed of the available functions and terminals.

The third preparatory step concerns the fitness measure for the problem. The fitness measure is the primary mechanism for communicating the high-level statement of the problem's requirements to the genetic programming system. The fitness measure specifies what needs to be done. The first two preparatory steps define the search space, whereas the fitness measure specifies the direction of the search.

7.1. Function Set

The first preparatory step in applying genetic programming involves identifying the function set for the programs to be evolved. The same function set is used for all the problems presented in this paper.

The widely used and well-established format for optical prescriptions (and lens files for optical analysis software) suggests a developmental process suitable for representing optical lens systems. This developmental representation employs a "turtle" similar to that used in Lindenmayer systems (Lindenmayer, 1968; Prusinkiewicz & Lindenmayer, 1990). A turtle was used in our previous work in synthesizing geometric patterns using developmental genetic programming (Koza, 1993) and in synthesizing antennas using developmental genetic programming (Comisky, Yu, & Koza, 2000; Koza, Keane, Streeter, Mydlowec, Yu, & Lanza, 2003*a*).

In our developmental representation for optical lens systems, the turtle starts at the point where the system's main axis intersects with the EP surface (point *g* in Fig. 6) and inserts optical surfaces and materials as it moves. The distance between point *g* and point *e* (where the EP surface intersects the system's main axis *b*) defines the eye relief of the system. Figure 6 shows the beginnings of the Tackaberry–Muller lens system shown of Figure 1 (where the eye relief is 0.88 mm).

The function set `F` contains two functions

$$\texttt{F} = \{\texttt{SS}, \texttt{PROGN2}\}$$

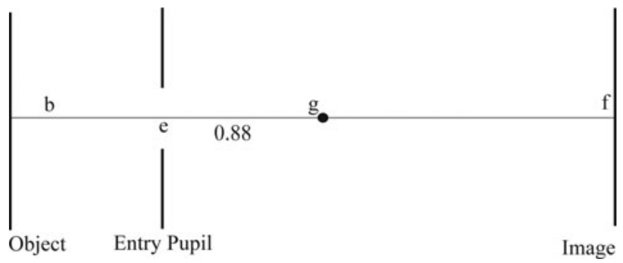


Fig. 6. Turtle starts at point g along main axis b.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

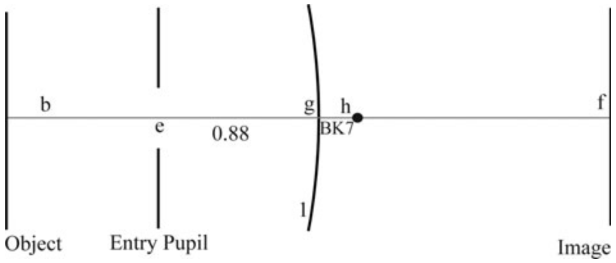


Fig. 7. Turtle inserts surface 1.

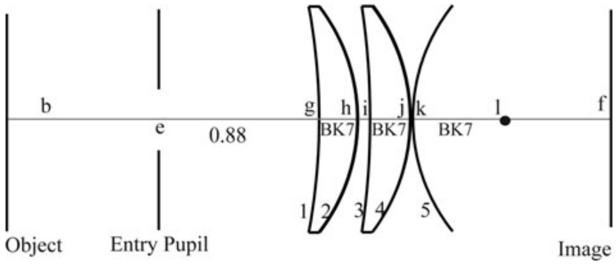


Fig. 9. Turtle inserts surface 5.

The three-argument spherical surface (SS) function causes the turtle to do three things at its starting point (and each subsequent point to which the turtle moves). First, it inserts a spherical surface with a specified radius of curvature at the turtle's present location. Second, the SS function moves the turtle to the right by a specified distance along the system's main axis. Third, the SS function fills the space to the right of the just-added surface with a specified type of material (e.g., air, vacuum, glass, polymer, oil, other material).

Radius and distance values are each established by a value-setting subtree of the SS function consisting of a single perturbable numerical value.

The material is established by a value-setting subtree of the SS function consisting of a single terminal identifying the type of material.

The two-argument PROGN2 function is a connective (glue) function that first executes its first argument and then executes its second argument. It does not itself add anything to the developing structure.

Figure 7 shows the result of the insertion of spherical surface 1 (with a radius of curvature of −3.52361). After inserting this surface, the turtle moves the specified distance of 0.219 from its starting point *g* to point *h* along axis line *b*. BK7 indicates that glass type BK7 (a commercially available crown glass) will be used to fill the space between *g* and whatever surface is subsequently inserted at point *h* (by the turtle's next step). These actions by the turtle reflect the information contained on the row labeled surface 1 in the prescription of Table 1.

Figure 8 shows the result of the insertion by the turtle of spherical surface 2 (with a radius of curvature of −1.05274). After inserting this surface, the turtle moves the specified distance of 0.07280 from point *h* to *i*. Surfaces 1 and 2 together define a lens of thickness 0.219 of BK7 glass. Because the centers of the two spheres defining surfaces 1 and 2 are both positioned in the negative direction (i.e., to the left) along the axis *b*, this first lens has a concave left surface and a convex right surface. Note that the material inserted to the right of surface 2 is air. That is, air will be used to fill the space between *h* and whatever surface is subsequently inserted at point *i* (by the turtle's next step). These actions by the turtle reflect the row labeled surface 2 in the prescription in Table 1.

In a similar manner, the rows labeled surfaces 3 and 4 in the prescription of Table 1 correspond to a second lens of BK7 glass (as shown in Fig. 9).

Figure 9 also shows the result of the insertion by the turtle of spherical surface 5 (with a radius of curvature of 1.02491). The turtle moves a distance of 0.52100 from point *k* to *l*. BK7 glass will be used to fill the space between *k* and whatever surface is subsequently inserted at point *l* (by the turtle's next step). These actions by the turtle reflect the row labeled surface 5 in the prescription of Table 1.

Figure 10 shows the result of the insertion by the turtle of spherical surface 6 (with a radius of curvature of −0.93493). The turtle moves a distance of 0.11800 from point *l* to *m*. SF4 glass is used to fill the space between *l* and whatever surface is subsequently inserted at point *m* (by the turtle's next step). Surfaces 5 and 6 together define a lens of thickness 0.52100 of BK7 glass. The positive radius of surface 5 and the negative radius of surface 6 together indicate that the third lens is biconvex. These actions by the turtle reflect the row labeled surface 6 in the prescription of Table 1. Note that a pair of snuggly fitting lens is being formed by this step

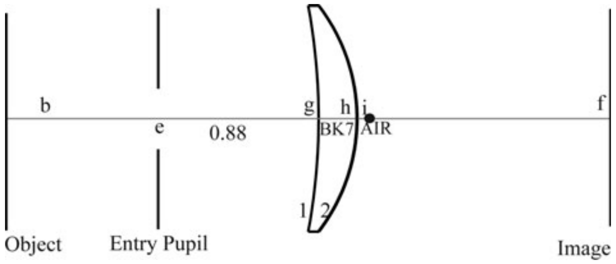


Fig. 8. Turtle inserts surface 2, thereby completing the first lens.

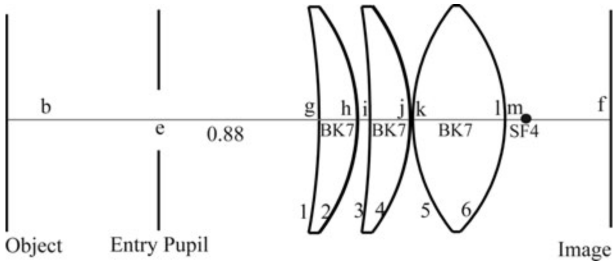


Fig. 10. Turtle inserts surface 6, thereby completing third lens.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

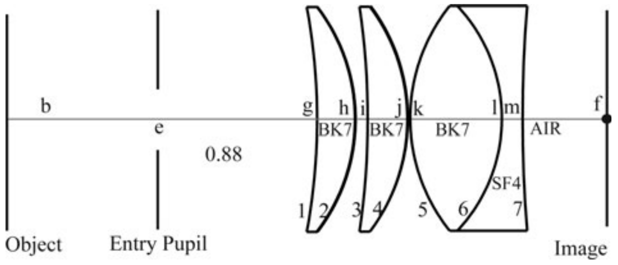


Fig. 11. Turtle inserts surface 7, thereby completing the forth lens (and creating a doublet).

because glass, as opposed to air, is filling the space between points *l* and *m*. These actions by the turtle reflect the row labeled surface 6 in the prescription of Table 1.

Figure 11 shows the result of the insertion by the turtle of spherical surface 7 (with a radius of curvature of 7.94281). Air is used to fill the space between *m* and whatever surface may be subsequently inserted. The turtle moves a distance of 0.47485 starting from point *m*. Surfaces 6 and 7 together define a lens of thickness 0.11800 of SF4 glass. Because there was no air between the third lens and this new lens, a doublet is formed. The negative radius of surface 6 and the positive radius of surface 7 together indicate that the right lens of the doublet is biconcave. These actions by the turtle reflect the row labeled surface 7 in the prescription of Table 1. Because there are no additional surfaces in the prescription, the developmental process ends at this stage. The IMS is automatically inserted where the turtle is now located (i.e., at distance 0.47485 from point *m*).

7.2. Terminal Set

The second preparatory step in applying genetic programming involves identifying the terminal set for the programs to be evolved. The same terminal set is used for all the problems presented in this paper.

The terminal set for each value-setting subtree that establishes thickness or radius values ($T_{\text{perturbable}}$) is

$$T_{\text{perturbable}} = \{\Re\},$$

where $\Re$ denotes perturbable numerical values between 0.0 and 1.0. The value produced by the value-setting subtree is interpreted using a scale that is appropriate for whether the terminal is being used for a surface's thickness or a radius of curvature.

The terminal set for each value-setting subtree that establishes the type of material (T_{material}) consists of the names of all of the permissible materials.

A constrained syntactic structure specifies how the functions and terminals may be combined in a program tree. A constrained syntactic structure consists of a set of rules of construction (Koza, 1992). The constrained syntactic structure here enforces the occurrence of

- a `PROGN2` function at the top of the program tree (because an optical system consisting of just one surface is of no interest),
- either an `SS` or `PROGN2` function as the arguments of the `PROGN2` function,
- a single perturbable numerical value for the value-setting subtree that establishes the radius of curvature (i.e., first argument of the `SS` function),
- a single perturbable numerical value for the value-setting subtree that establishes the numerical value for distance (i.e., second argument of the `SS` function), and
- a single type of material for the value-setting subtree that establishes material (i.e., third argument of the `SS` function).

A given optical lens system can be represented in four equivalent ways. Each representation is useful in some aspect of this paper's discussion of how to apply genetic programming to the design of optical lens systems.

First, there is the schematic diagram of the physical lens system (such as shown in Fig. 1). This representation provides a visual picture of the physical lens system.

Second, there is the prescription or lens file (such as shown in Table 1). This representation corresponds to the traditional way by which a lens system is described as a table of numerical and nonnumerical parameter values.

Third, there is the program tree consisting of a composition of functions and terminals. Figure 12 shows a program tree that represents the first two lenses (four surfaces) of the Tackaberry–Muller lens system of Figure 1 and Table 1. Genetic programming operates on a population of program trees.

Fourth, using the LISP programming language, there is the symbolic expression (S-expression) consisting of a composition of functions and terminals. Of the four equivalent representations, the S-expression is closest to the representation used internally by the computer code that implements genetic programming. The S-expression corresponding to the program tree of Figure 12 is shown in the following:

```
(PROGN2 (PROGN2 (SS -3.523 0.219 BK7)
                SS -1.053 0.073 AIR))
(PROGN2 (SS - 4.407 0.225 BK7)
                (SS -1.070 0.014 AIR)))
```

7.3. Fitness measure

The third preparatory step in applying genetic programming to a problem involves constructing the fitness measure. The purpose of the fitness measure is to specify what the human wants. Specifically, the fitness measure must establish a partial order among any two candidate lens systems that may be encountered

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

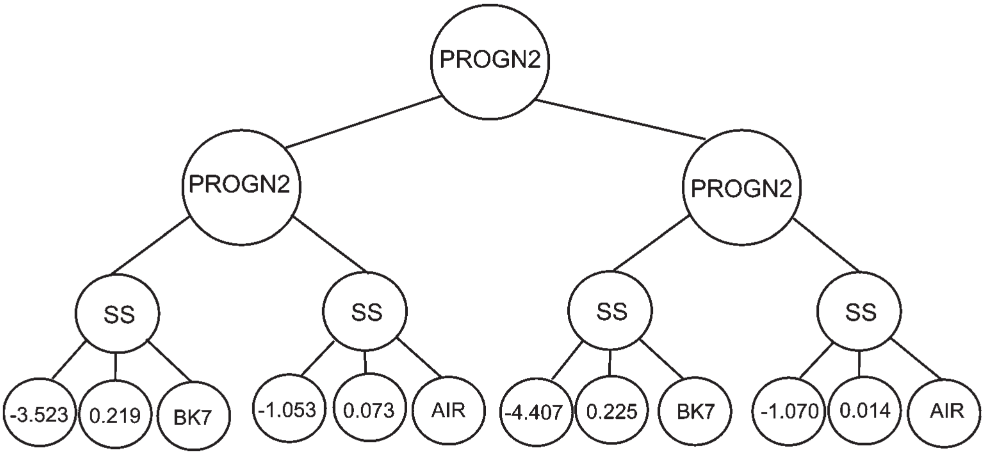
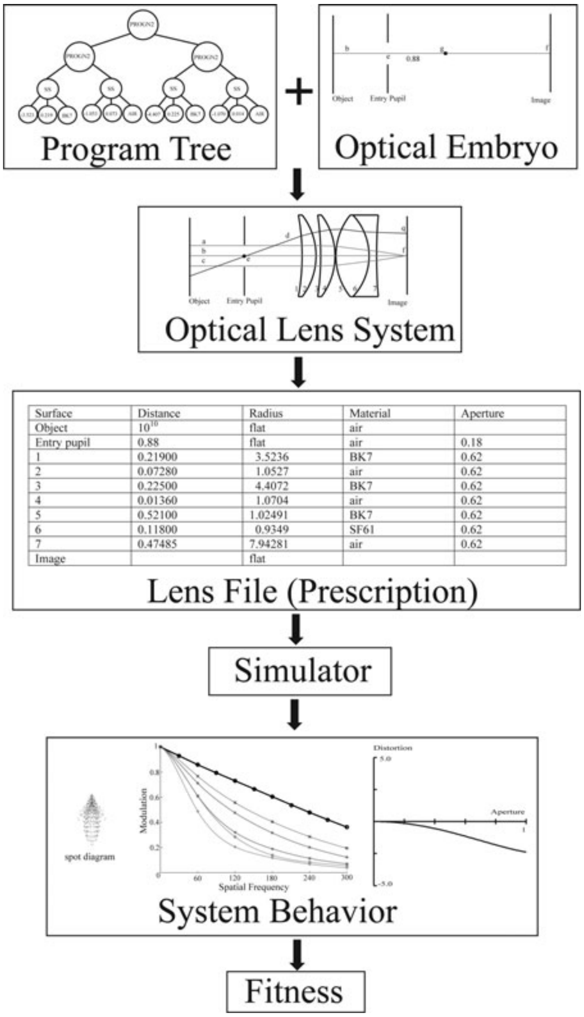


Fig. 12. Program tree for first two lenses of Tackaberry–Muller lens system.

during the run of genetic programming (i.e., the fitness measure must enable two candidates to be compared so that one is identified as being better than the other). The fitness measures for each of the problems presented in this paper are based on the level of performance of the lens system described in the particular patent. The fitness measures for each of the problems in this paper have the same general structure.

Figure 13 shows the steps in ascertaining the fitness of an individual program tree that may be encountered during the run of genetic programming. When developmental genetic programming it used, the ascertainment of fitness begins with the developmental process. First, the functions in the program tree are executed and applied to an optical embryo to create an optical lens system (as will be explained in additional detail momentarily). Second, the resulting optical lens system is reexpressed as a prescription (lens file) specifying the numerical and nonnumerical parameters associated with each surface in the lens system. Third, the prescription (lens file) is fed into a simulator to obtain the optical behavior and characteristics of the lens system. Fourth, a value of fitness is calculated from the system's behavior and characteristics.

Once a classical optical system is specified by means of its prescription (lens file), many of its optical properties can be calculated by tracing the path of light rays of various wavelengths through the system. The ray tracing analysis yields a set of characteristics of interest to optical designers, including distortion, astigmatism, and chromatic aberration. Ray-tracing analysis by hand is extremely time consuming. Ray tracing is typically performed nowadays by optical simulation software (e.g., OSLO, Zemax, Code V, KOJAC). KOJAC is a public-domain educational software package for optical ray tracing originally authored by Olivier Scherler and currently maintained by Olivier Ripoll. We developed our own lens analysis simulator based on KOJAC to evaluate the performance of candidate lens systems. We wrote code to use the ray traces produced by KOJAC to compute relevant optical characteristics, and in addition, wrote our own code for the image analysis. We embedded all of this code in our genetic programming system operating on our Beowulf-style parallel cluster computer. We used a commercially available software package (OSLO from Lambda Research) running on a single workstation for postrun validation of the final results obtained from the cluster computer.



Surface	Distance	Radius	Material	Aperture
Object	10^{10}	flat	air	
Entry pupil	0.88	flat	air	0.18
1	0.21900	3.5236	BK7	0.62
2	0.07280	1.0527	air	0.62
3	0.22500	4.4072	BK7	0.62
4	0.01360	1.0704	air	0.62
5	0.52100	1.02491	BK7	0.62
6	0.11800	0.9349	SF61	0.62
7	0.47485	7.94281	air	0.62
Image		flat		

Fig. 13. Steps for ascertaining the fitness of an individual program tree.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

Fitness is measured by embedding each candidate lens system into a fixed structure called the *test fixture*. The OBJ, IMS, and EP lines of the prescription together constitute the test fixture for evaluating a particular candidate lens system. For the eyepieces discussed in this paper, the distance between the OBJ surface and the EP in the test fixture is fixed at infinity. The distance between the EP and the first surface (*eye relief*) is fixed for each particular problem and appears as the distance on the line for the EP in the prescription (e.g., the eye relief is 0.88 mm in Table 1). The concept of test fixture used in this paper for optical systems is directly analogous to the test fixture used in connection with the automatic synthesis of electrical circuits by means of genetic programming (Koza, Bennett, Andre, & Keane, 1999; Koza, Keane, Streeter, Mydlowec, Yu, & Lanza, 2003).

The representation scheme used in this paper for applying genetic programming to the field of optical design is similar to that encountered in numerous other fields of design in that it permits the creation of various types of pathological structures that are of no practical interest. In the course of discussing the fitness measure below, we describe how we heavily penalize, slightly penalize, or repair various pathological structures.

The fitness measure is, for practical design problems in most fields, multiobjective in the sense that it combines two or more different elements. Trade-offs between multiple competing considerations are common to most fields of engineering design, regardless of whether the design is being accomplished by evolutionary search, other types of search, or human design. The different elements of the fitness measure are typically in competition with one another to some degree. This is indeed the case for optical design.

Because of the multiple conflicting objectives inherent in problems of optical design, the fitness measure is divided into four distinct phases. Broadly speaking, the four phases sequentially consider whether

- the system is not degenerate;
- the system's paraxial aberration performance is at least as good as that of the target (patented) system;
- the system's image-forming quality is at least as good as that of the target (patented) system; and
- the system is parsimonious, that is, it has the same or fewer lenses as the target (patented) system.

Numerical and Boolean values are both computed for each of the four phases. That is, fitness is a vector of eight items. The numerical value is used to rank the goodness of a candidate individual. The Boolean value indicates whether the individual satisfies certain minimum performance requirements specified for the phase. In comparing two individuals, an individual satisfying the minimum requirements of a phase (i.e., has a positive Boolean value) is considered superior to any individual that does not. In addition, an individual with a lower numerical value for a particular phase is considered superior to an individual with a higher numerical value for that phase.

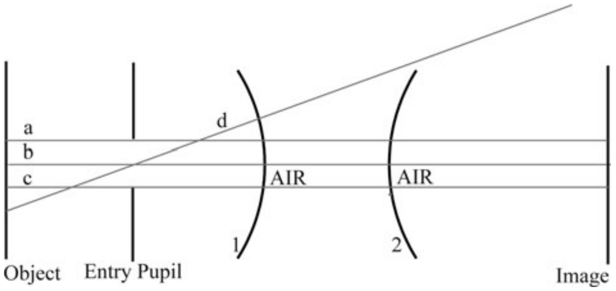


Fig. 14. All-air optical system.

To conserve computer time, the fitness evaluation of an individual not satisfying the minimum requirements of a particular phase (i.e., has a negative Boolean value) is short circuited (i.e., simply not evaluated in later phases).

As will be seen, construction of the fitness measure requires consideration of the technical requirements of the field of optical design.

The goal of the first phase of the fitness measure is to determine if a candidate lens system has two particular pathological characteristics, namely, whether the lens system is "all air" or whether glass abuts the IMS. Figure 14 shows an all-air system. An all-air system is created when air is specified as the material associated with every surface. Such an individual is of no interest, and receives an extremely high numerical penalty value for fitness (10^{100}) that effectively eliminates it from further serious consideration.

Figure 15 shows a lens system in which glass abuts the IMS. Such an individual is considered infeasible for the problems under consideration here, and this individual accordingly receives a large numerical penalty (500,000), reflecting its infeasibility. Some search is conducted outside the range of feasible structures in the hope that a desirable feasible individual may subsequently emerge. A large infeasibility penalty greatly reduces the likelihood that an individual will be selected to participate in a genetic operation; however, it does not entirely exclude an individual from further consideration. Infeasible individuals participate (to a diminished degree) in the genetic search.

For the first phase, the Boolean value is negative for an individual with a numerical value of fitness of 10^{100} and

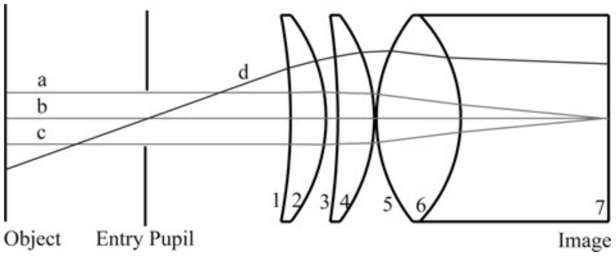


Fig. 15. Suspension passive physical structure design alternative 3.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

positive otherwise (i.e., those with a numerical value of 0 or 500,000). If a lens system does not have either of the above two pathologies, it is assigned a numerical fitness of 0 for the first phase.

The goal of the second phase of the fitness measure is to determine if the values of aberration and paraxial coefficients are at least as good as those of the target system, without vignetting (i.e., physically blocking some percentage of the incoming light), without modifying the system F-number (i.e., the ratio of the size of the effective EP to the focal length of the system), and without excessive distortion.

The second phase of the fitness measure involves an analysis of 62 rays. The 62 rays are considered in three separate groups:

- the system's two principle rays (i.e., the axial ray and chief ray),
- 20 rays traced along a radial axis to determine the distortion curve for the lens system, and
- 40 rays (20 tangential and 20 sagittal) from the full FOV OBJ position to analyze vignetting.

The first of these groups of rays is used to derive the system's aberration and paraxial coefficients, the second is used to measure distortion, and the third verifies that the system's aberration and paraxial performance was not achieved through the artifice of vignetting.

We first consider the system's two principle rays. Two pathological situations sometimes arise in tracing the axial and chief rays.

First, Figure 16 shows a divergent lens system in which the chief ray, d, does not hit the IMS at all. Ray traces also fail in other ways. For example, a ray trace is considered to fail if the light rays do not hit all surfaces of the system (including the final IMS) within the window determined by the eyepiece's maximum image size or if a ray hits the surfaces out of order. If either the axial or chief ray traces fail, the individual receives an extremely high numerical penalty value for fitness (10^{100}).

Second, Figure 17 shows a lens system in which the back focal length is negative (i.e., the image ends up inside the lens system). Such an individual receives a large numerical penalty equal to 500,000 plus 500,000 times the distance by which the back focal length is offset.

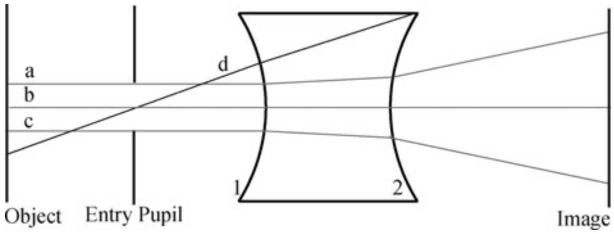


Fig. 16. Chief ray fails to hit IMS.

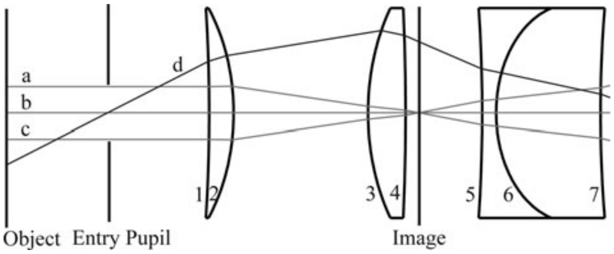


Fig. 17. Negative back focal length.

In the second phase, if a lens system does not have either of the above two pathologies, its performance is compared to that of the target (patented) lens system and a weighted sum of the deviations between the systems is computed. Specifically, the weighted sum is

- 10,000 times each of the Seidel aberration deviations, namely, spherical aberration, coma, astigmatism, field curvature (Petzval sum), and Seidel distortion;
- 20,000 times each of the chromatic deviations, namely, axial chromatic and lateral chromatic;
- if the back focal length is less than the target value, 10 times the absolute deviation of the back focal length from the target value; and
- if the Petzval radius is less than the target value, 25 times the absolute deviation of Petzval radius from the target value and 100 times the absolute deviation of the effective focal length from the target value.

The particular weightings above were chosen to approximately equalize, based on our own advance estimates, the relative influence of deviations from the various elements of the fitness measure. This approach is patterned after our recent work with multiobjective fitness measures in the field of automatic circuit synthesis (Koza, Jones, Keane, Streeter, & Al-Sakran 2004).

We now consider the 20 rays traced along a radial axis to determine the distortion curve for the lens system. For this analysis, 20 rays are traced along the OBJ tangential axis of the system, starting from the axial ray and extending out to the chief ray.

The distortion percentage is calculated at each position as the percent deviation from where the ray strikes the image plane and where it would strike an ideal image. An ideal image is one where the following constraints are met: if the coordinates of the OBJ point O from which the ray being cast originates are (h_y, h_x) and those of the point O' where the ray terminates on the image plane are (h_y', h_x'), the image of O onto O' will be considered ideal if $h_y' = mh_y$ and $h_x' = mh_x$, where m is the magnification of the image and is independent of h_y and h_x. In our case, m is a paraxial constant determined by dividing the image height (point q in Fig. 1) ascertained by the chief ray intersection of the image plane by the corresponding height in the OBJ plane where the chief ray originated. The distortion curve is then comprised of the distortion percentage values at each of the 20 rays along the

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

tangential axis of the system. The system is penalized for deviation at two specific points of the curve, with the first being at the chief ray's image height and the second being the maximum distortion value along the length of the curve defined by the 19 other ray traces. The numerical fitness value for the second phase is incremented by 100 times the deviation of each of these values with those corresponding in the target system (but only if the distortion is greater than that the target system).

Whenever a search (genetic or otherwise) is conducted under the direction of a fitness measure, there is the possibility that improvements can be achieved in a way that is of no practical interest. Experience indicates that a low value of the above weighted sum can be obtained through the artifice of modifying the system's *F* number or by vignetting. To counter these two forms of what is often called "cheating," the above weighted sum is adjusted in one of two ways.

For the vignetting analysis, we consider 40 rays (20 tangential and 20 sagittal) traced along the tangential and sagittal planes from the axis to the edge of the EP originating from the full FOV of the OBJ. The system is assumed to be symmetric across each of the axis. If any ray does not hit all surfaces of the system (including the final IMS) within the window determined by the eyepiece's maximum image size or if a ray hits the surfaces out of order, the system is said to *vignette*. The vignetting percentage is defined as the percentage of rays that vignette. The numerical fitness value accumulated so far for the second phase is divided by 100% minus this percentage (i.e., it multiplies the fitness value accumulated so far).

The *F*-number of the system is defined as the effective focal length divided by the aperture of the EP. It is easier to minimize the numerical value of fitness (because of aberrations) from the second phase if the *F*-number is larger because the proportion of light the system processes is decreased.

For the second phase, the Boolean value is positive if the first seven values of the weighted sum (i.e., the five Seidel aberrations and the two chromatic deviations) equal 0, if the individual has not been assigned numerical value of fitness of 10^{100}, if the effective focal length is not greater than 110% of the target value for effective focal length, and if no vignetting has occurred. Otherwise, the Boolean value is negative.

Returning to the discussion of pathological cases, certain combinations of values for distance and the radius of curvature may result in a lens system in which two lenses physically intersect. For example, Figure 18 shows a physical intersection of the two lenses defined by surfaces 5, 6, and 7. In such cases, we determine the maximum aperture that avoids the intersecting glass. We then pass the repaired individual (with reduced aperture) to the simulator. Experience indicates that the simulator will often successfully evaluate such individuals. As usual, if either chief ray or the axial ray falls beyond the maximum aperture, the numerical fitness value for the second phase is set to 10^{100} for the candidate individual.

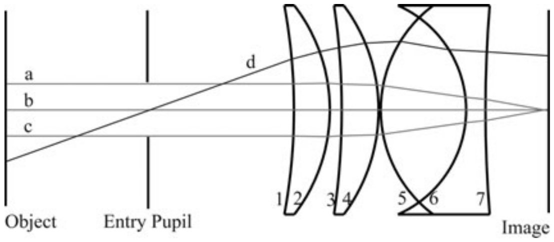


Fig. 18. Intersecting lenses within a doublet.

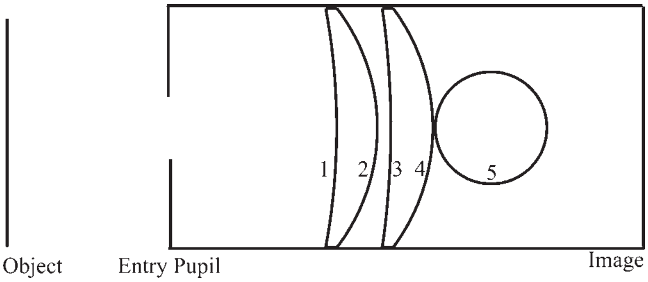


Fig. 19. A spherical marble.

Figure 19 shows another pathological case, called a *marble*. In this figure, the diameter of sphere 5 is smaller than the housing for the lens system, so that the sphere hovers in space.

Similarly, in Figure 20, two surfaces (5 and 6) define a hovering lens whose diameter is smaller than the housing for the lens system. Because lenses cannot hover in space, an opaque mounting is added above and below the lens. The mounting redefines (and reduces) the aperture. If either chief ray or the axial ray falls beyond the maximum aperture of surfaces 5 and 6, the numerical fitness value for the second phase is set to 10^{100} for the candidate individual.

The goal of the third phase of the fitness measure is to establish that the system's image-forming quality to be at least as good as that of the target (patented) system.

A 17 × 17 grid is overlaid on the EP and a ray is shot through the corner defining each grid position contained inside the EP. A three-color spot diagram is then formed and evaluated. Several other system metrics are derived from these data, including modulation transfer functions (MTFs) and point spread functions.

Figure 21a–c shows gray scale versions of a three-color spot diagram for the Tackaberry–Muller system. The rays from each FOV position in the figure are traced for each of three

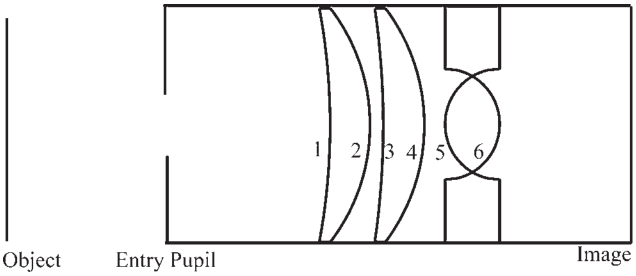


Fig. 20. Another example of a marble.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

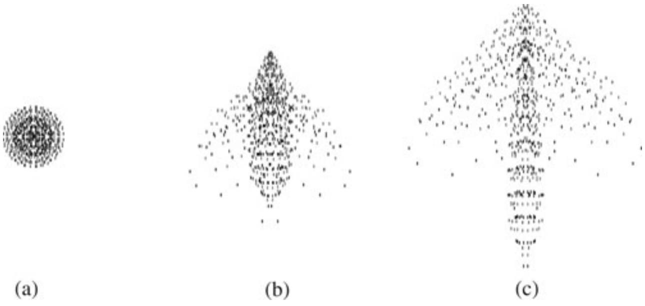


Fig. 21. Gray scale versions of spot diagrams.

wavelengths (486, 588, and 656 nm) and projected through the system onto the image plane. Figure 21a shows the trace from the axial point. Figure 21b shows the trace from the 70% of the FOV. Figure 21c shows the full FOV performance.

The spot diagram provides a measure of the deviation resulting from the compound error of the chosen lens aberration contributions. An ideal diffraction limit spot size (corresponding to the minimum spot size that can be discernable when diffraction effects are taken into account) is determined for the system and the root mean square (RMS) error for the ray intercept deviations is calculated. The numerical fitness value for the third phase is incremented by 200, 340, and 400 times the difference between the target RMS error for the axial, 70% FOV and full FOV, respectively. These different penalty multipliers reflect the increasing difficulty in attaining the desired performance.

An MTF measures the contrast and resolution differences between the OBJ and the image for a specified plane and FOV. Figure 22 shows the six MTFs for the Tackaberry–Muller system in the tangential and sagittal planes of each of the three FOV positions used for the spot diagrams. Each curve is sampled at 30 increments of 10 cycles/mm across the target system. We obtain the same six MTFs for the candidate lens system. The numerical fitness value for the third phase is further incremented by a sum over the 30 points for each of the six MTFs. Each addend in this sum consists of a weight and the amount (if any) by which the MTF of the candidate individual is worse than the MTF of the target system. There is no contribution if the candidate individual is better than the target system. The weights are 0.5 for the sagittal-axial, 1.0 for the tangential-axial, 1.0 for the sagittal 70% field, 2.0 for the tangential 70% field, 2.0 for the sagittal full field, and 4.0 for the tangential full field.

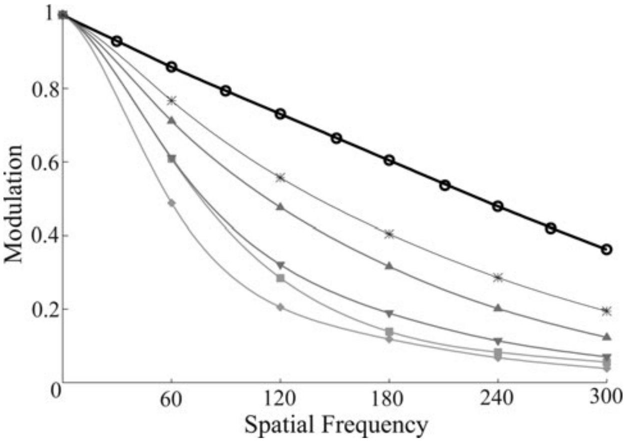


Fig. 22. Modulation transfer functions.

For the third phase, the Boolean value is positive if the values for the RMS error for the spot diagram are equal or less than those of the target system and the weighted cumulative error across the MTF curves is no larger than 5%.

The goal of the fourth phase of the fitness measure is parsimony. The parsimony penalty is the sum of 20 times the number of lenses plus the width of the lens system (its footprint) measured in millimeters.

If desired, the cost of the glass (found in the glass catalog) can be considered in this fourth phase.

For the fourth phase, the Boolean value is positive if the lens system has a better value of the parsimony penalty than the target (patented) system, but negative otherwise.

To conserve computer time, if an "all-air" optical system is encountered in the process of creating the initial generation of the population (generation 0), it is immediately deleted from the population and a replacement is created.

All fitness calculations are done with double precision arithmetic.

The runs of genetic programming described in this paper starts with an initial population that is randomly created from the available functions and terminals. About 94.6% of the randomly created individuals are pathological in some way and cannot be simulated. If we retained these individuals in the population for generation 0 (say, penalizing them heavily because of their unsimulatability), we would be limiting the genetic material available for crossovers and mutation to only about 1/20 of what it might be. Thus, in creating the initial random generation, we continue generating the required number of individuals until 100% of the generation 0 is simulatable. Because simulatable individuals tend to breed simulatable offspring (an observation applicable to the design of analog electrical circuits, antenna, and controllers as well as the design of optical lens systems), the issue of unsimulatability is not a substantial concern in later generations.

7.4. Control parameters

The fourth preparatory step in applying genetic programming to a problem entails specifying the control parameters for the run.

The crossover operation is usually predominant in runs of genetic programming (typically being performed at rates such as 80 or 90%). However, because there is only one active function (SS) in the function set used for the optical design work described in this paper and all arguments of this one function are terminals, perturbation of parameters (numerical and nonnumerical) is far more important than usual. Accordingly, we performed numerical parameter mutation at a rate of

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

32%, glass mutation at a rate of 22%, lens splitting at a rate of 10%, crossover at a rate of 32%, reproduction at a rate of 3%, and subtree mutation at a rate of 1%.

The standard deviation for the Gaussian distributions used for numerical parameter mutation was 0.1. In glass mutation, there is a 10% probability of air mutating into a type of glass and vice versa.

The maximum number of points in a program tree is 1000. In creating generation 0, the `SS` function was twice as likely as the `PROGN2` function.

Tournament selection (with a group size of 2) was used. Elitism (i.e., the automatic reproduction of the single best individual on each processing node) was used. In crossover, the probability of choosing a terminal is 40% and a nonterminal is 60%.

We used the Schott Glass Catalog (containing 199 types of glass).

We partitioned our Beowulf-style computer cluster into five parts for the problems of optical design in this paper. The population size was substantially the same for all problems in this paper (2000 individuals for each of 134 processing nodes for a total of 268,000 individuals). Slight differences in population arose because of the unequal number of active nodes in each of the five partitions.

7.5. Termination

The fifth preparatory step in applying genetic programming to a problem consists of specifying the termination criterion and the method of designating the result of the run. In general, the termination criterion may include a maximum number of generations to be run as well as a problem-specific success predicate. For the work described in this paper, we manually monitored the run and terminated the run when the values of fitness for numerous successive best of generation individuals appeared to have reached a plateau. The single best so far individual was then harvested and designated as the result of the run. For each run, the stated result was the smallest fully compliant individual produced.

7.6. Preparatory steps

Table 2 summarizes the preparatory steps for runs of genetic programming used to automatically synthesize optical lens systems that duplicate (or improve upon) the functionality of the patented optical lens systems presented in this paper.

8. RESULTS FOR THE FOUR PATENTED OPTICAL LENS SYSTEMS

As will be seen in this section, one of the genetically evolved lens systems infringed a previously issued patent, whereas the others were noninfringing novel designs that duplicated (or improved upon) the performance specifications contained in the patents. The noninfringing novel designs can be considered new inventions.

Table 2. *Preparatory steps for runs of genetic programming*

Objective:	Design an optical lens system that duplicates (or improves upon) the functionality of the patented optical lens system.
Test fixture:	A lens system (with no lenses) containing an object surface (OBJ), an image surface (IMS) and an entry pupil (EP).
Developmental process:	A turtle starts on the axis at a specified distance from the entry pupil (the *eye relief*) and inserts spherical surfaces and materials as the turtle moves to the right.
Program architecture:	One result-producing branch. Automatically defined functions and architecture-altering operations are not used.
Function set:	`F = {SS, PROGN2}.` The three-argument SS function inserts a spherical surface and material into a developing optical system. The two-argument `PROGN2` function is a connective function.
Terminal set:	The terminal set for establishing distance and radius values consists of perturbable numerical values. The terminal set for establishing the type of material consists of the names of all of the permissible materials.
Fitness:	A multiobjective fitness measure incorporating various optical characteristics obtained from a ray-tracing simulator.
Parameters:	Population size $M = 268{,}000$.
Result designation:	Best so far individual
Termination:	Manual termination of the run when the values of fitness for numerous successive best of generation individuals appeared to have reached a plateau

8.1. Nagler eyepiece

Figure 23 shows the patented Nagler lens system (Nagler 1985). This system has four groups of lenses containing a total of six lenses.

Each run of genetic programming in this paper starts with an initial population (generation 0) in which each individual in the population is randomly created from the available functions and terminals. That is, the runs of genetic programming start *ab initio* (as opposed to starting with a human-designed good lens system).

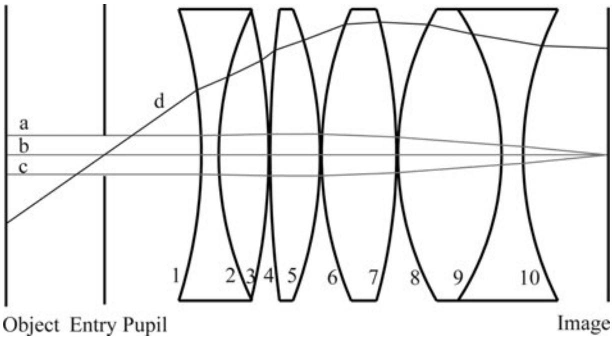


Fig. 23. Nagler patent.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

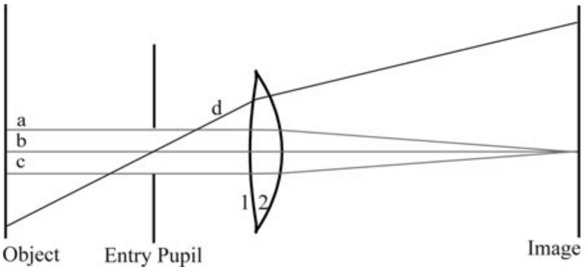


Fig. 24. Best of generation 0 for the Nagler problem.

For nontrivial problems, the individuals in the population at generation 0 of a run of genetic programming are invariably poor in terms of satisfying the problem's requirements. This was indeed the case for the Nagler problem. The best of generation individual from the initial random population (generation 0) and consisted of a lens system with one lens (Fig. 24). This individual has a numerical value of fitness of 222 for the second phase of the fitness measure but a negative Boolean value (i.e., it does not satisfy the requirements of the second phase of the fitness measure).

The number of lenses in the best lens system produced by a run of genetic programming is not prespecified by the human user. Instead, the number of lenses (as well as their topological arrangement and the numerical and nonnumerical parameters associated with each lens) emerge during the evolutionary process in an automated way. The first best of generation individual with two groups of lens occurred in generation 29 (with a numerical value of fitness of 102 and a negative Boolean value for the second phase of the fitness measure). That is, the best of generation lens system from generation 29 has better performance than the best of generation individual from generation 0, although it has not satisfied the performance requirements of the second phase of the fitness measure.

Although the best of generation individual from generation 0 is poor in terms of satisfying the problem's requirements, the single lens provides a toehold that enables the evolutionary process to proceed.

A two-group topology (with four lenses) appeared as the best of generation individual of generation 35 (Fig. 25). This individual had a numerical value of fitness of 57 and a negative Boolean value for the second phase of the fitness measure).

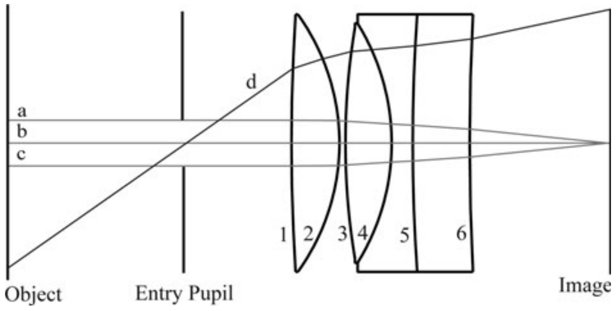


Fig. 25. Best of generation 35 for the Nagler problem.

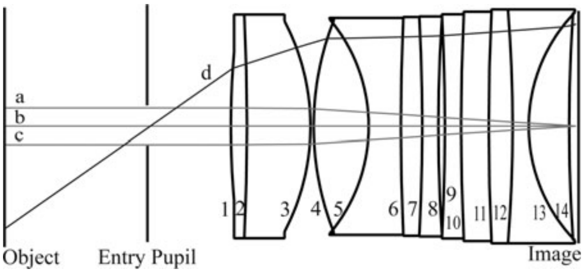


Fig. 26. Best of generation 124 for the Nagler problem.

As the run continued, best of run individuals emerged with three and four groups of lenses (with the largest group containing four lenses).

The first best of generation individual that has a numerical value of fitness of 0 and a positive Boolean value for the second phase of the fitness measure appeared in generation 124. It has a numerical value of fitness of 204 and a negative Boolean value for the third phase of the fitness measure. This lens system (Fig. 26) consisted of three groups, with two, four, and five lenses in the three groups, respectively.

Different two- and three-group arrangements appeared over approximately the next 100 generations, whereas three- and four-group arrangements dominated for approximately 150 generations.

Figure 27 shows the best of generation individual of generation 382. This individual is the last pace-setting individual with a numerical value of fitness of 21 and a negative Boolean value for the third phase of the fitness measure.

The first best of generation individual with a numerical value of fitness of 0 and a positive Boolean value for the third phase of the fitness measure appeared in generation 330. This individual consisted of a singlet and a triplet and has a parsimony fitness of 104. Parsimony was thus relevant for all later pace-setting individuals. As the run continued, the total size of the system and the remaining residual errors were reduced.

Figure 28 shows the best of run lens system from generation 495 of the Nagler problem. This lens system has two groups of lenses, one singlet and one triplet.

Table 3 shows the prescription (lens file) for the best of run individual from generation 495. Note that there is an air surface (surface 1) at the front of the lens system (surface 1 in the table). Thus, the air that was built into the system in the form

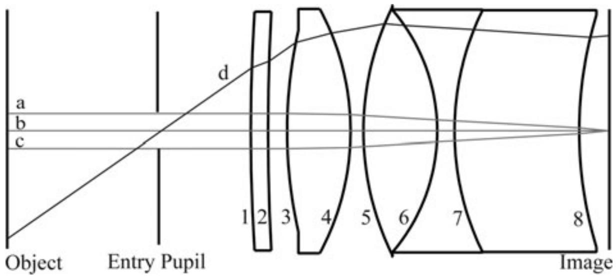


Fig. 27. Best of generation 382 for the Nagler problem.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

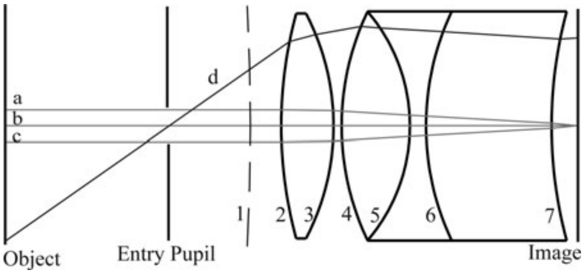


Fig. 28. Best of run lens system from generation 495 for the Nagler problem.

of the eye relief (the 0.5 mm of air on the "EP" line of the table) is added to the air created by this air surface (the 0.19 mm of air for surface 1). The result is to increase the eye relief by 38% (from 0.5 to 0.69). Note that the radius of curvature is irrelevant when a surface is air.

Table 4 compares characteristics of the best of run individual from generation 495 with those of the patented lens system for the Nagler problem. As can be seen in the table, the evolved individual is comparable to the Nagler lens system for each characteristic in the table (and superior in all but five instances).

Note that in Table 4, the patented lens systems always scores a value of zero for all of the MTF measurements, and that a value of zero for the evolved solution indicates that the evolved solution is equal to, or superior to, the patented system in that respect. In addition, note that the last nine lines of Table 4 (and other similar comparison tables later in this paper) are measured quantities that are not part of the fitness measure.

The best of run individual from generation 495 for the Nagler problem differs considerably from Nagler's patented invention (Nagler, 1985), and therefore does not infringe the patent. However, as shown in Table 5, the inventor's design goals, as stated in the patent, are achieved. As can be seen in the table, the Nagler patent focused on a wide FOV and a specific performance (low astigmatism).

Thus, the best of run lens individual from generation 495 accomplished the inventor's design goals in addition to being superior to the lens system in the Nagler patent for each characteristic in the Table 4. Therefore, the best of run lens individual from generation 495 is a noninfringing novel design that duplicates (and indeed improves upon) the performance specifications for invention, that is, the evolved lens system can be considered as a new invention.

Table 3. *Lens file for best of run lens system from generation 495 for the Nagler problem*

Surface	Distance	Radius	Material	Aperture
OBJ	10^{10}	Flat	Air	
EP	0.5	Flat	Air	0.098615
1	0.190362	2.651701	Air	0.448719
2	0.322134	−14.948329	LAK21	0.691494
3	0.050000	−1.493656	Air	0.691494
4	0.420184	1.627528	LAFN21	0.704655
5	0.100000	−1.079922	SF57	0.704655
6	0.773284	1.648485	LASFN15	0.704655
7	0.161123	2.723813	Air	0.704655
IMS		Flat		0.687515

Table 4. *Comparison of characteristics of the patented Nagler system and best of run individual from generation 495*

	Nagler Patent	Evolved Solution
Spherical aberration	−0.000467	−0.000419
Coma	0.000365	0.000017097
Astigmatism	0.004094	0.003000
Petzval	−0.012140	−0.011745
Distortion	−0.214273	−0.205194
Distortion (%)	21.655	21.421
Max distortion (%)	21.655	21.421
Axial chromatic	0.000243	0.000119
Lateral chromatic	−0.001698	−0.001372
Spot RMS		
Axial error	0.0	0.0
70% FOV error	0.40212	0.39901
Full field error	0.54564	0.54424
MTF axial		
Tangential	0	0
Sagittal	0	0
MTF 70%		
Tangential	0	0
Sagittal	0	0
MTF full		
Tangential	0	(Residual error)
Sagittal	0	0
Peak–valley OPD axial	0.01971	0.01772
RMS OPD axial	0.006045	0.005431
Peak–valley OPD 70%	0.3067	0.3181
RMS OPD 70%	0.07773	0.07909
Peak–valley OPD full	0.5511	0.5985
RMS OPD full	0.1393	0.1473
0 Peak PSF		
Axial	0.999	0.999
70% field	0.755	0.755
Full field	0.5	0.27

The best of run lens individual from generation 495 has the unexpected benefit of having two fewer lenses. That is, the portion of the fitness measure devoted to parsimony was successful in directing the genetic search to a parsimonious lens system that satisfied the problem's requirements.

Table 5. *Design goals of the Nagler patent*

Nagler Patent	Evolved Solution
Very wide field of view (>70°)	Field of view extends to 70°
Reduced astigmatism at the edge of the field	Lens system has very low astigmatism in image
Low aberration contributions	Evolved individual has equivalent performance
Six lenses	Only four lenses

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

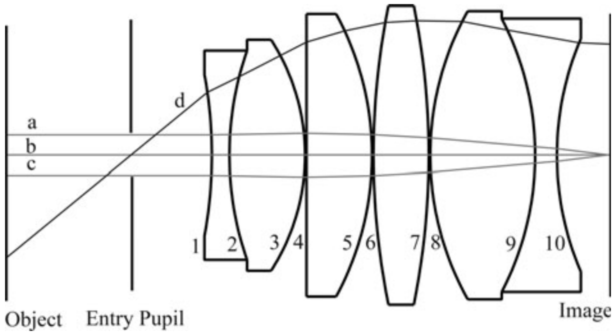


Fig. 29. Scidmore patent.

Another unexpected benefit is the 38% larger eye relief of the best of run lens individual from generation 495. This benefit is remarkable because this characteristic was not an explicit part of the fitness measure. It is an example of the fact that the evolutionary process sometimes rewards the user with something beyond that which was explicitly requested by the fitness measure.

8.2. Scidmore eyepiece

Figure 29 shows the patented Scidmore lens system (Scidmore, 1968). This system has four groups of lenses containing a total of six lenses.

The best of generation individual of generation 0 of the Scidmore problem has a fitness of 777 and consisted of a lens system with one lens (Fig. 30). Similarly, the best of generation individuals for the next seven generations has one lens; the best of generation individuals between generations 8 and 31 have two lenses; and the best of generation individual until generation 49 has three lenses. A wide variety of topologies appeared between generations 50 and 98, most having four groups of lenses.

Figure 30 shows the best of run lens system from generation 194 for the Scidmore problem.

Table 6 shows the lens file for best of run lens system from generation 194 for the Scidmore problem.

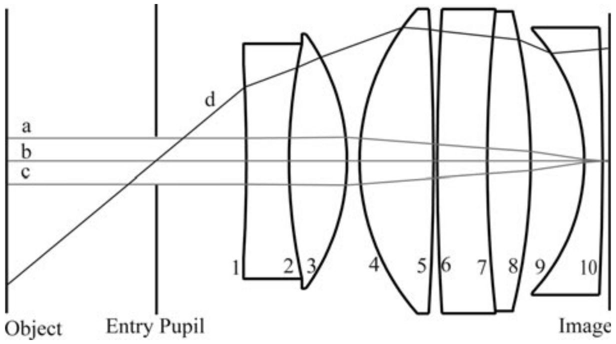


Fig. 30. Best of run lens system from generation 194 for the Scidmore problem.

Table 6. *Lens file for the best of run individual from generation 194 for the Scidmore problem*

Surface	Distance	Radius	Material	Aperture
OBJ	10^{10}	Flat	Air	
EP	0.493359	Flat	Air	0.125218
1	0.232378	−9.30484	SF59	0.501373
2	0.316047	2.973292	LAFN21	0.651134
3	0.068638	−1.20292	Air	0.719252
4	0.403009	1.271851	PSK52	0.880533
5	0.02	−13.7423	Air	0.880533
6	0.272930	13.23272	LAF2	0.93062
7	0.236531	7.667174	LASFN31	0.93062
8	0.292384	−3.490137	Air	0.897563
9	0.1	−1.062511	SF53	0.851483
10	0.001989	−14.95792	Air	0.724261
IMS		Flat		0.8391

The genetically evolved lens system is comparable to the patented lens system for each of the characteristics listed in Table 4 and, in fact, is superior in all but four instances.

The best of run lens individual from generation 194 for the Scidmore problem differs considerably from the Scidmore patent (Scidmore, 1968), and therefore, does not infringe the patent. However, as shown in Table 7, the inventor's design goals, as stated in the patent, are achieved. As can be seen in the table, the Scidmore patent focused on achieving a minimum acceptable performance level over a wide FOV and minimizing the number of surfaces for manufacturing efficiency.

Thus, the best of run lens individual from generation 194 is a noninfringing novel design that duplicates (and indeed improves upon) the performance specifications for invention, that is, the evolved lens system can be considered as a new invention.

8.3. Koizumi–Watanabe eyepiece

Figure 31 shows the patented Koizumi–Watanabe lens system (Koizumi & Watanabe, 2000), Figure 32 displays the best of run lens system created by our run of genetic programming in generation 1026 on the Koizumi–Watanabe problem, and Table 8 provides the lens file for best of run lens system from generation 1026 for the Koizumi–Watanabe problem.

The genetically evolved lens system is comparable to the patented lens system for each of the characteristics listed in Table 4 and, in fact, is superior in all but four instances.

Table 7. *Design goals of the Scidmore patent*

Scidmore Patent	Evolved Solution
Very wide field of view (>80°)	Field of view extends to 80°
Minimize lens elements (six lens elements in patent)	Evolved individual has six lens elements
Low aberration contributions	Evolved individual has performance equivalent to patent

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

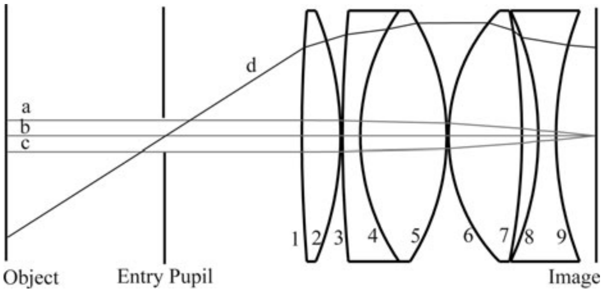


Fig. 31. Koizumi–Watanabe patent.

The best of run lens individual from generation 1026 for the Koizumi–Watanabe problem differs considerably from the patent (Koizumi & Watanabe, 2000), and therefore, does not infringe the patent. The best of run lens individual from generation 1026 accomplished the inventors' design goals (Table 9), and is therefore a noninfringing novel design that duplicates (and indeed improves upon) the performance specifications for invention—that is, the evolved lens system can be considered as a new invention.

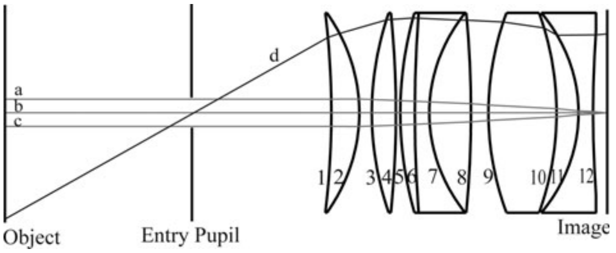


Fig. 32. Best of run lens system from generation 1026 for the Koizumi–Watanabe problem.

Table 8. *Lens file for the best of run individual from generation 1026 for the Koizumi–Watanabe problem*

Surface	Distance	Radius	Material	Aperture
OBJ	10^{10}	10^{10}	Air	
EP	0.925247	10^{10}	Air	0.106126
1	0.022066	−5.936786	Air	0.695573
2	0.192813	−7.362778	LAK23C	0.709631
3	0.086376	−1.473264	Air	0.790685
4	0.166211	2.663955	SK13	0.820276
5	0.030000	−8.744855	Air	0.837020
6	0.100000	3.159402	LAK8	0.840131
7	0.100000	11.381255	SF57	0.835117
8	0.287737	1.354558	LAK31	0.829935
9	0.120000	−9.312223	Air	0.829286
10	0.469817	2.506716	LASFN30	0.821380
11	0.151433	−2.591650	Air	0.773951
12	0.100000	−1.314657	SF58	0.735645
13	0.092526	9.419787	Air	0.651290
IMS		10^{10}		0.642290

8.4. Tackaberry–Muller and Konig eyepiece

Figure 1 shows the patented Tackaberry–Muller lens system (Tackaberry & Muller, 1958), and Table 1 shows the lens file for the Tackaberry–Muller patent. The 1958 Tackaberry–Muller patent cites the 1940 Konig patent (Konig, 1940) and is a special case of it.

Figure 33 illustrates the best of run lens system from generation 241 for the Tackaberry–Muller problem. This lens system has only three groups and a total of four lenses. Table 10 shows the lens file for best of run lens system from generation 241 for the Tackaberry–Muller problem.

The genetically evolved lens system is comparable to the patented lens system for each of the characteristics listed in Table 4 and, in fact, is superior in all but four instances.

The best of run lens individual from generation 241 for the Tackaberry–Muller problem differs considerably from the patent (Tackaberry & Muller, 1958), and therefore, does not infringe the patent. However, as shown in Table 11, the inventors' focus was on achieving a minimum acceptable performance level over a wide FOV and minimizing the number of surfaces for manufacturing efficiency. The inventors' design goals, as stated in the patent, are achieved by best of run lens individual from generation 241.

Table 12 compares that the best of run lens system from generation 241 with claim 1 of the Konig patent (Konig, 1940). As can be seen, the best of run lens system from generation 241 infringes the Konig patent.

In previously published work (Al-Sakran, Koza, & Jones, 2005), we used a fitness measure that considered fewer characteristics than the fitness measure used herein. In the earlier work, the topology of the best of run individual (Fig. 34) matched that of the topology of the Tackaberry–Muller. That is, in that earlier work, genetic programming succeeded in evolving a lens system that infringed claim 1 of the Konig patent (Konig, 1940). In other words, genetic programming succeeded, in the earlier work, in reinventing a previously patented invention.

Table 13 compares the features of the best of run lens system from that earlier run with claim 1 of the Tackaberry–Muller patent (Tackaberry & Muller, 1958). The only difference (a slightly out of range radius of curvature) is apparently because of the improved performance of the genetically evolved design compared to the patented 1958 design.

Table 9. *Design goals of the Koizumi-Watanabe patent*

Koizumi–Watanabe Patent	Evolved Solution
Wide field of view (>55°)	Field of view extends to 65°
Less than 6% distortion through full field of view	Max distortion of 5.13%
Less than or equal to six lens elements	Evolved solution has seven lenses
Low aberration contributions	Evolved solution has equivalent performance to embodiment

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

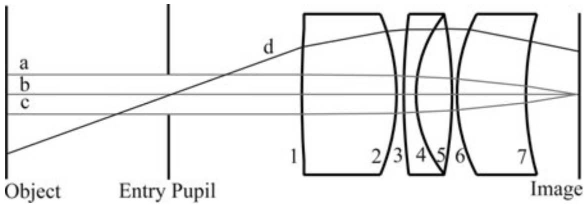


Fig. 33. Best of run lens system from generation 241 for the Tackaberry–Muller problem.

Additional information may be found in Al-Sakran, Koza, and Jones (2005); Jones, Al-Sakran, and Koza (2005); and Koza, Al-Sakran, and Jones (2005*a*, 2005*b*).

8.5. Three common features of the four results

The above results have three features in common, namely,

- genetic programming starts small,
- genetic programming breeds simulatability, and
- genetic programming engineers around existing patents and creates novel designs more frequently than infringements.

The initial population (generation 0) for a run of genetic programming is typically created using a probabilistic growth process for program trees. Generally, this growth process is structured so as to guarantee the generation of program trees for generation 0 that span a wide range of sizes and shapes. There is a loose correlation between program size and the number of operative parts in the structure (but, of course, it is entirely possible for a large program tree to have a small number of operative parts).

The best of generation structure of generation 0 tends to have a small number of operative parts.

We suggest that the reason that better individuals at the beginning of a run of genetic programming tend to have just one (or just a few) operative parts is that the probability (in the initial random growth process) of creating a multipart structure

Table 10. *Lens file for the best of run individual from generation 241 for the Tackaberry Muller–problem*

Surface	Distance	Radius	Material	Aperture
OBJ	10^{10}	Flat	Air	
EP	0.88	Flat	Air	0.18
1	0.179862	Flat	Air	0.63
2	0.769713	8.177274	KZFS7A	0.63
3	0.058810	−1.670709	Air	0.63
4	0.100000	5.153155	SF59	0.63
5	0.292354	1.021859	LAFN21	0.63
6	0.040000	−3.279045	Air	0.63
7	0.560156	1.404301	LASFN30	0.63
8	0.429366	2.410695	Air	0.63
IMS		Flat		0.357227

Table 11. *Design goals of the Tackaberry–Muller patent*

Tackaberry–Muller Patent	Evolved Solution
Eyepiece having a flat field	Evolved system has a flatter field
Full correction for color within the eyepiece itself	Evolved system has chromatic aberrations corrected to a higher degree than patented system
Four lens elements	Evolved system has four lenses
“Ample” eye relief	Evolved system has greater eye relief than the patented system

Table 12. *Comparison of claims of the Konig patent and the best of run lens system from generation 241*

Claim 1 of Konig Patent	Evolved Solution
“An optical system for telescope eyepieces, comprising a front, a medial and a rear element, said elements being convergent and axially spaced by air,”	The evolved solution contains three convergent elements (two single lenses and one doublet lens) and they are separated by air.
“the sum of the distances apart of said elements being at most one-third of the focal length of said system,”	The focal length of the evolved system is 0.992298 and the sum of the distances is 0.09881 (∼1/10).
“said rear element being a single lens, the numerical value of the curvature of the rear surface of said lens being smaller than the numerical value of the refractive power of said lens,”	The curvature of the rear element (the lens defined by surfaces 1 and 2) is 0.12229 (i.e., 1/8.177274) and its refractive power is 0.49017 (computed by the standard textbook formula).
“said medial element consisting of at least one lens and at most two lenses,”	The medial element (the lenses defined by surfaces 4–6) is a doublet lens.
“said front element consisting of at least one lens,”	The front element (surfaces 7 and 8) consists of a single lens.
“the front lens of said medial element and that lens of said front element which faces this front lens of said medial element being convergent,”	These two lenses (viz., the front of the medial lens defined by surfaces 5 and 6 and the singlet defined by surfaces 7 and 8) are both convergent.
“at least one optically effective surface of one of said two convergent lenses being a cemented surface,”	The doublet lens defined by surfaces 4–6 has a common surface 5.
“the refractive power of one cemented surface of said two convergent lenses amounting to at least eleven twentieths of the algebraic sum of the refractive powers of all cemented surfaces of said convergent lenses,”	The refractive power of the specified surface is 0.16068. It is also the only common surface and hence amounts to the entire sum described.
“the numerical value of last said sum being greater than one twelfth of the sum of the numerical values of the curvatures of those surfaces of said convergent lenses which face each other.”	One-twelfth of the sum corresponds to 0.033928 and 0.16068 is greater.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

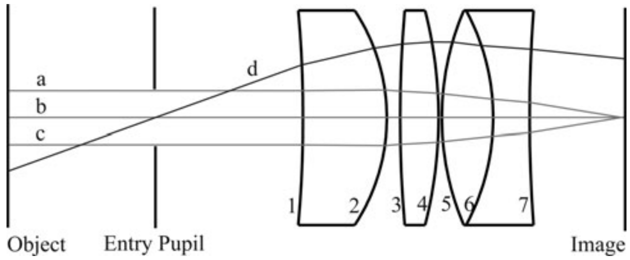


Fig. 34. Earlier evolved individual for the Tackaberry–Muller problem.

with reasonably high fitness is small. The probability is small because, in multipart structures, the parts must be appropriately connected together and the numerical parameters associated with each part must lie in certain limited, appropriately coordinated ranges.

When simulators are employed in conjunction with genetic programming, unsimulatable individuals are typically assigned a severe penalty. Because Darwinian selection is used at each step of genetic programming, the practical effect of this penalty is that unsimulatable individuals are almost never selected to be reproduced into the next generation or to participate in crossover and mutation.

Of course, culling out unsimulatable individuals and eliminating their influence on future generations does not itself guarantee that simulatable parents will beget simulatable offspring. At the time when we began our work on the automatic synthesis of optical lens systems by means of genetic programming, it was not known whether genetic programming would beget simulatable offspring.

Table 13. *Comparison of claims of the Tackaberry–Muller patent and of best of run lens system*

Claim 1 of Tackaberry–Muller Patent	Evolved Solution
"A telescopic eyepiece adequately corrected for color"	The genetically evolved solution is a slight improvement over the patented design.
"consisting of three convergent components"	There are three convergent components.
"formed of four lenses"	There is a total of four lenses.
"the front component being a doublet"	The front component is a doublet.
"comprising a divergent lens of relatively low dispersion glass"	The first material of the doublet is SF58, a flint (low dispersion glass).
"and a convergent lens of crown glass"	The second material of the doublet is LAK16A, a crown glass (convergent lens)
"the dispersive indices of the two lenses of the doublet having a ratio lying between .415 and .445"	The dispersive indices are 21.51 and 51.78, with a ratio of .415.
"and the radius of curvature of the internal contact surfaces lying between .86 F and 1.01 F,"	The radius of curvature is 1.3.
"the other convergent components being single lenses of crown glass."	The material for the other two single lens is SSK3, a crown glass.

Table 14. *Percentage of unsimulatable individuals in generations 0 and 1*

Problem	Generation 0	Generation 10
Nagler	76	9
Skidmore	80	11
Tackaberry–Muller	64	10
Koizumi–Watanabe	82	14

Table 14 shows the percentage of unsimulatable individuals at generation 0 and at a typical early generation (i.e., generation 10) of the runs reported in this paper. As can be seen, the vast majority of individuals in generation 0 are unsimulatable. However, after only a small number of generations, most of the individuals are simulatable.

We consider the fact that genetic programming breeds simulatability to be a very surprising recurring feature of the work discussed because new individuals are introduced into the population by probabilistic operations (i.e., crossover, mutation). However, experience indicates there is a high probability that the offspring produced by crossover will be simulatable (Koza, Al-Sakran, & Jones, 2005*a*, 2005*b*).

As of this writing, genetic programming has been used to automatically synthesize multiple previously patented inventions (as opposed to single previously patented inventions) in two domains. In addition to the optical lens systems discussed in this paper, genetic programming has previously duplicated the functionality of six previously patented 21st century analog circuit designs (Koza, this issue). Acknowledging the small amount of data, we believe it is nonetheless worth mentioning that, in both the group of optical patents and the group of analog circuit patents, genetic programming infringed only one of the patents. Specifically, genetic programming created an infringing circuit for the Irving–Kolb circuit patent and created an infringing optical system for the Konig optical patent. We suggest that the reason that there is only one infringement in each of these two groups of patents is that there are many ways to satisfy engineering specifications. Genetic programming conducts a search for a satisfactory solution, but it has no *a priori* knowledge about (and, hence, no particular preference for) the solution that a human may have patented in the past. In addition, a patent is a solution for a problem, but it is not necessarily (or even usually) an optimal solution. Genetic programming seeks a solution that optimizes its fitness measure.

8.6. Human-competitiveness of the results

Referring to the eight criteria for establishing that an automatically created result is competitive with a human-produced result (Koza, Bennett, Andre, & Keane 1999; Koza, Keane, Yu, Bennett, & Mydlowec 2000), the rediscovery by genetic

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

programming of an optical lens system duplicating the functionality of the each of the above patented inventions satisfies the following two of the eight criteria:

A. The result was patented as an invention in the past, is an improvement over a patented invention, or would qualify today as a patentable new invention.

. . .

F. The result is equal to or better than a result that was considered an achievement in its field at the time it was first discovered.

8.7. High return for human effort expended

The question arises, in connection with any automated problem-solving method, as to what is delivered by the automated operation of the problem-solving method in comparison to the amount of knowledge, information, analysis, and intelligence that is presupplied by the human employing the method.

We define the *artificial/intelligence* (*AI*) *ratio* of a problem-solving method as the ratio of that which is delivered by the automated operation of the *artificial* method to the amount of *intelligence* that is supplied by the human applying the method to a particular problem.

The AI ratio is especially pertinent to methods for getting computers to automatically solve problems because it measures the value added by the artificial problem-solving method. Manifestly, the aim of the fields of artificial intelligence, machine learning, and automated design is to generate human-competitive results with a high AI ratio.

Ascertaining the AI ratio for the problems presented in this paper requires measuring the amount of A that is delivered by the automated problem-solving method (i.e., genetic programming) in relation to the amount of I that is supplied by the human user (i.e., in the form of the preparatory steps prior to the launch of the run of genetic programming).

In performing this qualitative comparison, we start with the amount of A. The result produced by genetic programming for the problems presented in this paper is considered to be human-competitive for the two reasons mentioned in the previous section. Because the results are human-competitive, there is a high amount of A in the numerator of the AI ratio.

Measuring the amount of I requires the discipline of drawing an explicit line between that delivered by the artificial system and that supplied by the intelligent human user. We imposed this discipline when we detailed the human-supplied preparatory steps for this problem.

We did not employ any deep knowledge about optical lens systems in selecting the terminals and functions to be used by genetic programming in synthesizing optical systems. In fact, the function and terminal sets were based on the widely used and well-established format for optical prescriptions and lens files for optical analysis software. That is, the terminal set and the function set incorporated only platitudinous information about optical lens systems.

The fitness measure is constructed directly from the high-level statement of the problem's requirements, as presented by the inventor in their patent documents. Clearly, technical knowledge of the field of optics is required to construct a quantative measure for evaluating candidate individuals. However, construction of a fitness measure is a precondition to doing optical design: whether it is done with the aid of conventional nongenetic optimization or search methods or done by humans.

The remaining preparatory steps (concerning control parameters, the termination criterion, and the method of result designation) were unremarkable administrative steps.

In summary, the preparatory steps were straightforward and uncomplicated translations of a high-level statement of the problem. That is, the human user supplied only *de minimus* knowledge about the field of optical design. Hence, there is only a small amount of I in the denominator of the AI ratio.

The high amount of A in the numerator of the AI ratio in conjunction with the small amount of I in the denominator means that the AI ratio is high for the genetically evolved solutions to the problems of optical design presented in this paper.

8.8. Routineness of the results

Generality is a precondition to what we mean when we say that an automated problem-solving method is "routine." Once the generality of a method is established, "routineness" means that relatively little human effort is required to get the method to successfully handle new problems within a particular domain and to successfully handle new problems from a different domain. The ease of making the transition to new problems lies at the heart of what we mean by routine.

The routineness of the transition from problem to problem is clear from the problems of optical design presented in this paper. All the problems were run using the same software, the same computer hardware, and the same settings of the minor control parameters. As we moved from one problem to the next, the only substantial change is the specification of what needs to be done. For the work described in this paper, this amounted to changing the target values in the fitness measure to match the performance levels specified in the patent involved and adjusting the test fixture to reflect the specific problem. This specification is based on each inventor's statement of performance of each patented circuit. As stated in *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (Koza, 1992*a*), "Structure arises from fitness."

In summary, the designs were created in a substantially similar and routine way, indicating that the approach described in the paper can be readily applied to other problems in the field of optical design.

8.9. Summary

In summary, genetic programming delivered routine, high-return, human-competitive solutions to the problems of optical design presented in this paper.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

9. PARALLEL COMPUTING SYSTEM

The problems described in this paper were run on a Beowulf-style parallel cluster computer system built in 1999 (Sterling, Salmon, Becker, & Savarese, 1999). Each node of the cluster contains of a 350-MHz Pentium II processor accompanied by 64 MB of RAM. The nodes communicate with one another by 100 Mbps Ethernet. The Pentium system has a 350-MHz Pentium II computer as host. The nodes and the host use the Linux operating system. There is no hard disk associated with the processing nodes. Instead, the operating system, the genetic programming system, and the data (notably the population) reside in RAM.

The entire system consists of 1000 processors. Two processors are housed in each of the 500 physical boxes of the 1000-Pentium system.

The system was partitioned into five parts for the work described in this paper. Each 200-processor part of the system is organized into a 2 × 2 toroidal arrangement of groups, with each group containing 50 processors (25 physical boxes). A population of size 2000 resides at each processor.

The "island" version of parallel genetic programming was employed. In the island version of parallel genetic programming, semi-isolated subpopulations (demes) of individuals reside at each processor. The time-consuming evaluation of fitness and the relatively quick genetic operations are performed separately on each processor. The time required to evaluate fitness varies considerably from individual to individual, with larger and more complex structures generally requiring more time to simulate. Generations are run asynchronously on each processor. This asynchronous approach guarantees full usage of the computational resources of each processor because no computational resources are wasted in an attempt to synchronize activity on a global basis. As each processor (asynchronously) completes the generation on which it is currently working, emigrants from that processor's population are dispatched to four toroidally adjacent processors. The emigrants wait in a buffer at their destination processor until that processor reaches the end of its current generation.

The four processors that are toroidally adjacent to a given processor may be of three types. One of the four toroidally adjacent processors is always in the same physical box. The second case is that the toroidally adjacent is in the same group of 50 processors. The third case is that the toroidally adjacent is in a different group of 50 processors. The rate of migration varies depending on these three cases. The rate of migration is 10% if the destination is within the same physical box, 5% if the destination is within the same group of 50 processors, and 2% if the destination is within a different group of 50 processors. The emigrants are selected randomly for emigration within a physical box of within a group of 50 processors. The emigrants are selected probabilistically based on fitness for emigration between the different groups of 50 processors. The result is that the two populations residing in the same physical box are highly intermixed; the populations within a group of 50 processors are intermixed to an intermediate degree; and the four groups of 50 are more lightly intermixed.

Fan failures and hard disk failures are the major operational problems associated with Beowulf-style commodity off the shelf (COTS) cluster computers. In our case, we have no hard disks, so fan failures have been the major operational problem. In the 5 years of operation of our 1000-processor system, fan failures (concentrated in the third year) caused about 24% of the processors to burn out. We have now replaced all the fans. The result of this damage is that there are now dead processors scattered at random in the system. Emigrants are only sent to processors that are known to be functional.

10. COMPUTER TIME, MOORE'S LAW, AND THE FUTURE OF GENETIC PROGRAMMING FOR AUTOMATED DESIGN

Table 15 shows the amount of computer time required for the runs of genetic programming presented in this paper. There is considerable variation in the size and complexity of the optical lens system satisfying the requirements of each problem. Moreover, we have made only one run of each problem, so we do not have statistically valid averages for any of the problems. Having said that, the average time required to process a single individual is about 0.9 s; the average number of processed individuals is about 128,000,000; and the average length of the run is 239 h (about 10 days).

Remembering that the processors are 350-MHz Pentium II processors and that inexpensive computers available at the time of this writing operated at about 3.5 GHz, the average length of a run on contemporary equipment would be about 23.9 h (about 1 day).

Alternatively, the average number of processors used for the runs in the table is 129, which is equivalent to a cluster of about 13 inexpensive contemporary (2005) computers.

Table 15. *Computer time required for runs*

Problem	Population	Generations	Individuals Processed	Run Duration (h)	Time per Individual (s)
Nagler	268,000	496	132,928,000	150	0.544
Skidmore	282,000	195	54,990,000	104	0.96
Tackaberry–Muller	218,000	242	52,756,000	151	1.123
Koizumi–Watanabe	264,000	1027	271,128,000	551	0.966
Average	258,000	490	127,950,500	239	0.898

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

Techniques of genetic and evolutionary computation generally require significant computational resources to solve nontrivial problems. This statement is generally true *a fortiori* when the results are human competitive.

Fortunately, because of three factors, the computer time necessary to achieve human-competitive results has become increasingly available in recent years.

First, the speed of commercially available single computers continues to double approximately every 18 months in accordance with Moore's law (Moore, 1996). Moore's law is currently expected to continue (and possibly accelerate) over the next decade. This exponential growth in computational power is equivalent to 2 orders of magnitude per decade: 10,000 to 1 over 20 years and 1,000,000 to 1 over 30 years.

Second, techniques of genetic and evolutionary computation in general (and genetic programming in particular) are especially amenable to efficient (and almost effortless) parallelization (Koza, Bennett, Andre, & Keane 1999*a*).

Third, affordable parallel cluster computer systems can be assembled with relative ease in the Beowulf-style using COTS hardware (Sterling, Salmon, Becker, & Savarese, 1999).

The combined effect of the second and third factors above is that increased computational power may be realized both at the rates called for by Moore's law and though parallelization.

Referring again to the computer time required for the runs of genetic programming presented in this paper (Table 15), the

- the average length of a run on contemporary equipment would be about 24 h,
- the average length of a run in 2015 would be 2.4 h,
- the average length of a run in 2025 would be 0.24 h (14.4 min), and
- the average length of a run in 2035 would be 0.024 h (1.44 min or 86.4 s).

Looking at automated design in terms of the increasing complexity of problems that may be solved in the future,

- a problem 100 times more complex may be solvable in 2015,
- a problem 10,000 times more complex may be solvable in 2025, and
- a problem 1,000,000 times more complex may be solvable in 2035.

11. CONCLUSION

This paper described how genetic programming was used to automatically synthesize complete designs for four optical lens systems. The automated synthesis was done *ab initio*, that is, without starting from a preexisting good design and without prespecifying the number of lenses, the topological arrangement of the lenses, or the numerical or nonnumerical parameters associated with any lens. The paper described how genetic programming created lens systems that duplicated the functionality (or infringed) four previously patented lens systems, including one 21st century patent. The genetically evolved designs are instances of human-competitive results produced by genetic programming in the field of optical design.

ACKNOWLEDGMENT

Alexander Scheeline of the University of Illinois made helpful comments on various versions of this work.

REFERENCES

Alander, J.T. (2000). *An Indexed Bibliography of Genetic Algorithms in Optics and Image Processing—Draft August 16, 2000*, Report 94-1-OPTICS. Vassa, Finland: University of Vaasa, Department of Information Technology and Production Economics.

Al-Sakran, S.H., Koza, J.R., & Jones, L.W. (2005). Automated re-invention of a previously patented optical lens system using genetic programming. *Genetic Programming: 8th European Conf., EuroGP 2005, Lecture Notes in Computer Science* (Keijzer, M., Tettamanzi, A., Collet, P., van Hemert, J., & Tomassini, M., Eds.), Vol. 3447, pp. 25–37. Heidelberg: Springer–Verlag.

Banzhaf, W., Nordin, P., Keller, R.E., & Francone, F.D. (1998). *Genetic Programming—An Introduction.* San Francisco, CA/Heidelberg: Morgan Kaufmann/dpunkt.

Beaulieu, J., Gagné, C., & Parizeau, M. (2002). Lens system design and reengineering experimentations with genetic algorithms and genetic programming. *Proc. 2002 Genetic and Evolutionary Computation Conf.* (Langdon, W.B., Cantu-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E., & Jonoska, N., Eds.), pp. 155–162. San Francisco, CA: Morgan Kaufmann.

Comisky, W., Yu, J., & Koza, J. (2000). Automatic synthesis of a wire antenna using genetic programming. *Late-Breaking Papers at the 2000 Genetic and Evolutionary Computation Conf.*, pp. 179–186.

Fischer, R.E., & Tadic-Galeb, B. (2000). *Optical System Design.* New York: McGraw–Hill.

Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Jones, L.W., Al-Sakran, S.H., & Koza, J.R. (2005). Automated design of a previously patented aspherical optical lens system by means of genetic programming. In *Genetic Programming Theory and Practice III* (Yu, G., Worzel, W., & Riolo, R., Eds.), Chap. 3, pp. 33–48. New York: Springer.

Keane, M.A., Koza, J.R., & Streeter, M.J. (2005). *Apparatus for improved general-purpose PID and non-PID controllers.* US Patent 6,847,851. Filed July 12, 2002. Issued January 25, 2005.

Koizumi, N., & Watanabe, N. (2000). *Wide-field eyepiece.* US Patent 6,069,750. Filed August 17, 1998. Issued May 30, 2000.

Konig, A. (1940). *Telescope eyepiece.* US Patent 2,206,195. Filed in Germany December 24, 1937. Filed in United States December 14, 1938. Issued July 2, 1940.

Koza, J.R. (1990). *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Technical Report STAN-CS-90-1314, Stanford University, Computer Science Department.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: MIT Press.

Koza, J.R. (1994*a*). *Genetic Programming II: Automatic Discovery of Reusable Programs.* Cambridge, MA: MIT Press.

Koza, J.R. (1994*b*). *Genetic Programming II Videotape: The Next Generation.* Cambridge, MA: MIT Press.

Koza, J.R., Al-Sakran, S.H., & Jones, L.W. (2005*a*). Automated re-invention of six patented optical lens systems using genetic programming. *Proc. Genetic and Evolutionary Computation Conf. (GECCO–2005)* (Beyer, H.-G., et al., Eds.), pp. 1953–1960. New York: ACM Press.

Koza, J.R., Al-Sakran, S.H., & Jones, L.W. (2005*b*). Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits. *Proc. 2005 NASA/DoD Conf. Evolvable*

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press

Hardware (Lohn, J., et al., Eds.), pp. 205–212. Los Alamitos, CA: IEEE Computer Society Press.

Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1996*a*). Toward evolution of electronic animals using genetic programming. *Artificial Life V: Proc. 5th Int. Workshop on the Synthesis and Simulation of Living Systems* (Langton, C.G., & Shimohara, K., Eds.), pp. 327–334. Cambridge, MA: MIT Press.

Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1996*b*). Four problems for which a computer program evolved by genetic programming is competitive with human performance. *Proc. 1996 IEEE Int. Conf. Evolutionary Computation*, pp. 1–10. New York: IEEE Press.

Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1996*c*). Automated design of both the topology and sizing of analog electrical circuits using genetic programming. *Proc. Artificial Intelligence in Design'96* (Gero, J.S., & Sudweeks, F., Eds.), pp. 151–170. Dordrecht: Kluwer Academic.

Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1996*d*). Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming. *Genetic Programming 1996: Proc. First Annual Conf.* (Koza, J.R., Goldberg, D.E., Fogel, D.B., & Riolo, R.L., Eds.), pp. 123–131. Cambridge, MA: MIT Press.

Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1996*e*). Reuse, parameterized reuse, and hierarchical reuse of substructures in evolving electrical circuits using genetic programming. *Proc. Int. Conf. Evolvable Systems: From Biology to Hardware (ICES–96), Lecture Notes in Computer Science* (Higuchi, T., Iwata, M., & Liu, W., Eds.), Vol. *1259*, pp. 312–326. Berlin: Springer–Verlag.

Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving.* San Francisco, CA: Morgan Kaufmann.

Koza, J.R., Bennett III, F.H., Andre, D., Keane, M.A., & Brave, S. (1999). *Genetic Programming III Videotape: Human-Competitive Machine Intelligence.* San Francisco, CA: Morgan Kaufmann.

Koza, J.R., Jones, L.W., Keane, M.A., Streeter, M.J., & Al-Sakran, S.H. 2004. Toward automated design of industrial-strength analog circuits by means of genetic programming. In *Genetic Programming Theory and Practice II* (O'Reilly, U.-M., Riolo, R.L., Yu, G., & Worzel, W., Eds.), pp. 121–142. Boston: Kluwer Academic.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., & Lanza, G. (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence.* New York: Kluwer Academic.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G., & Fletcher, D. (2003). *Genetic IV Video: Routine Human-Competitive Machine Intelligence.* New York: Kluwer Academic.

Koza, J.R., Keane, M.A., Yu, J., Bennett III, F.H., & Mydlowec, W. (2000). Automatic creation of human-competitive programs and controllers by means of genetic programming. *Genetic Programming and Evolvable Machines 1*, 121–164.

Koza, J.R., & Rice, J.P. (1992). *Genetic Programming: The Movie.* Cambridge, MA: MIT Press.

Langdon, W.B., & Poli, R. (2002). *Foundations of Genetic Programming.* New York: Springer–Verlag.

Moore, G.E. (1996). Can Moore's law continue indefinitely? *Computerworld Leadership Series 2*(*6*), 2–7

Nagler, A. (1985). *Wide angle eyepiece.* US Patent 4,525,035. Filed January 5, 1984. Issued January 25, 1985.

Scidmore, W.H. (1968). *Wide angle eyepiece.* US Patent 3,390,935. Filed August 9, 1965. Issued July 2, 1968.

Smith, W.J. (1992). *Modern Lens Design: A Resource Manual.* Boston: McGraw–Hill.

Smith, W.J. (2000). *Modern Optical Engineering*, 3rd ed. New York: McGraw–Hill.

Sterling, T.L., Salmon, J., & Becker, D.J., & Savarese, D.F. (1999). *How to Build a Beowulf: A Guide to Implementation and Application of PC Clusters.* Cambridge, MA: MIT Press.

Tackaberry, R.B., & Muller, R.M. (1958). *Telescope eyepiece system.* US Patent 2,829,560. Filed October 15, 1956. Issued April 8, 1958.

John R. Koza is a Consulting Professor in the Biomedical Informatics Program in the Department of Medicine and the Department of Electrical Engineering at Stanford University. He is the author of the 1992 book *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, three subsequent books on genetic programming, and over 250 papers on genetic programming, genetic algorithms, and artificial life.

Sameer Al-Sakran received an MS in scientific computing and computational mathematics from Stanford University in 2003. He worked at Genetic Programming Inc. between 2003 and 2007, performing research on genetic programming and applications to automated design of analog electronic and optical systems. He is currently working on music recommendation, similarity, and other data-crunching problems at imeem.com.

Lee Jones is a Research Programmer at Genetic Programming Inc. in Mountain View, CA.

https://doi.org/10.1017/S0890060408000176 Published online by Cambridge University Press