Design of a Gradient-Index Beam Shaping System via a Genetic Algorithm Optimization Method

Neal C. Evans and David L. Shealy*
University of Alabama at Birmingham
Department of Physics, 1530 3rd Avenue South, CH310
Birmingham, AL 35294-1170 USA

ABSTRACT

Geometrical optics - the laws of reflection and refraction, ray tracing, conservation of energy within a bundle of rays, and the condition of constant optical path length - provides a foundation for design of laser beam shaping systems. This paper explores the use of machine learning techniques, concentrating on genetic algorithms, to design laser beam shaping systems using geometrical optics. Specifically, a three-element GRIN laser beam shaping system has been designed to expand and transform a Gaussian input beam profile into one with a uniform irradiance profile. Solution to this problem involves the constrained optimization of a merit function involving a mix of discrete and continuous parameters. The merit function involves terms that measure the deviation of the output beam diameter, divergence, and irradiance from target values. The continuous parameters include the distances between the lens elements, the thickness, and radii of the lens elements. The discrete parameters include the GRIN glass types from a manufacturer's database, the gradient direction of the GRIN elements (positive or negative), and the actual number of lens elements in the system (one to four).

Keywords: beaming shaping; optical design; optimization, genetic algorithm, GRIN, gradient index

1. INTRODUCTION

Geometrical optics has been used for a number of years to design optical systems which shape the irradiance profile of a laser beam.[1, 2, 3, 4, 5, 6] These geometrical methods of beam shaping have generally involved solving differential equations for the contours of the optics. Wang and Shealy[7] have presented a method to compute the GRIN profiles of a two-lens beam shaping system. However, their GRIN profiles can not be associated with available GRIN glass types, which motivated the current work to determine whether a genetic algorithm based optimization method can be used to design a laser beam shaping system with catalog GRIN elements. The present work uses the design equations of the geometrical methods[1] of laser beam shaping as a basis for constructing a merit function used during the GA optimization of GRIN beam shaping systems.

Recent developments of machine learning techniques, such as genetic algorithms (GAs), have transformed the prospect of solving difficult optimization problems. Though there are a number of variations of GAs, they all share a central theme: their search strategy borrows concepts from natural selection and genetics.[8] These techniques endow GAs with several unique features, as described by Goldberg:[9]

- GAs work with a coding of the parameter set, not with the parameters themselves;
- GAs search from a population of points, not a single point;
- GAs use payoff …[merit function] information only, not derivatives or other auxiliary knowledge;
- GAs use probabilistic transitions rules, not deterministic rules.

These characterizations suggest using GAs to design to laser beam shaping systems when analytical methods are difficult to apply or other optimization techniques are inefficient or fail to yield good solutions. Many conventional optimization techniques work only within a continuous parameter space, since they are driven by first- and second-order derivatives. The ability to choose from discrete parameters is a particularly powerful feature of the GA, relative to other optimization codes. As a first step, Evans and Shealy[10, 11] have shown that GA optimization methods can determine the shapes of one- or two-aspherical lenses used in laser beam shaping systems.

* Correspondence: D.L. Shealy. Other author information: N.C.E.: Email: nce@uab.edu; Telephone: 205-934-4736; Fax: 205-934-8042. D.L.S.: E-mail: dls@uab.edu ; Telephone: 205-934-8068; Fax: 205-934-8042.

In this paper, GAs are used to design two laser beam shaping systems. The first system contains two plano-aspheric lenses and is used to expand and transform a Gaussian input beam into a top-hat output beam. This system is similar to one designed, fabricated and tested by Jiang, Shealy, and Martin.[12, 13, 14] The second problem solves not only for the attributes of the lens elements, but also for the GRIN glass types from a catalog and the number of elements used in the beam shaping system.[15] It is interesting to note that the GRIN glass type and the number of elements in a beam shaping system can only have discrete finite values. Section 2 describes a specific the merit function that has been used with a genetic algorithm optimization method to design several laser beam shaping systems. Section 3 describes application of the genetic algorithm optimization method to design a two plano-aspheric lens system and a three element GRIN beam shaping system.

2. GENETIC ALGORITHM OPTIMIZATION METHOD*

This section links geometrical optics into the merit function used to design laser beam shaping systems via GA optimization. Developing a fast, accurate means of computing the irradiance (energy per unit area per unit time) profile at different locations within an optical system is important to using GAs to design laser beam shaping systems. Conservation of radiant energy within a bundle of rays[16] is used to develop a computational procedure for evaluating the irradiance within an optical system. If no energy is dissipated by the system, then the total energy entering the system is given by:

$$E = \int_I \sigma(\rho) \left(\hat{\mathbf{n}}^{in}(\rho) \cdot \mathbf{v}^{in}(\rho) \right) da = \int_O u(\mathrm{P}) \left(\hat{\mathbf{n}}^{out}(\mathrm{P}) \cdot \mathbf{v}^{out}(\mathrm{P}) \right) dA, \tag{1}$$

where $u(\mathrm{P})$ is the irradiance function on the output surface O, and $\sigma(\rho)$ is the irradiance function on the input plane I, which is assumed to be Gaussian in the paper:

$$\sigma(\rho_i) = \exp\left(-2\left(\frac{\rho_i}{\rho_N} \right)^2 \right). \tag{2}$$

The input beam waist is ρ_N. See Figure 1 for illustration of the terms in Eq. (1).

Balancing the radiant energy striking differential ring da with the radiant energy exiting differential ring dA gives

$$u(\mathrm{P}) = \sigma(\rho) \frac{\cos(i^{in}(\rho))}{\cos(i^{out}(\mathrm{P}))} \left(\frac{2\pi\ \rho_i\ d\rho_i}{2\pi\ \mathrm{P}_i dS_i} \right) = \sigma(\rho) \frac{\cos(i^{in}(\rho))}{\cos(i^{out}(\mathrm{P}))} \left(\frac{2\pi\ \rho_i\ d\rho_i}{\frac{2\pi\ \mathrm{P}_i d\mathrm{P}_i}{\cos\chi_i^{out}}} \right), \tag{3}$$

where dS_i is an element of length on the output surface and is equal to $d\mathrm{P}_i / \cos\chi_i^{out}$ for the configuration shown in Figure 1. To evaluate $u(P)$, N rays are traced through the system. For these applications, $N = 200$ gives adequate resolution for the input and output profiles. Each ray enters parallel to the optical (Z-) axis at a specified height, ρ_i, where the set of ρ_i are distributed equally across the radius of the input plane according to the following function:

$$\rho_i = \left(\frac{r}{N} \right) i, \quad i = 0 \ldots N. \tag{4}$$

Each ray will exit the system and strike a point on the output surface at P_i, as shown in Figure 1. Assuming $d\rho_i = \rho_i - \rho_{i-1}$, then Eq. (3) can be written in terms of these discrete variables as

---

* See Appendix of the paper for a brief summary of the GA terminology, optimization algorithms, and methods.

$$u(\mathrm{P}_i) = \sigma\left(\rho_i\right)\left(\frac{\cos(i_i^{in})\,\rho_i\left(\rho_i - \rho_{i-1}\right)\cos(\chi_i^{out})}{\cos(i_i^{out})\,\mathrm{P}_i(\mathrm{P}_i - \mathrm{P}_{i-1})}\right). \tag{5}$$

Subscripts are introduced into these equations to emphasize the numerical nature of the solution the irradiance along a ray, Eq.(5). Equations (2) and (5), along with the ray trace array, provide an accurate means of calculating the beam irradiance profile over any reasonable surface. The accuracy of this method has been verified by calculating the profiles for several benchmark systems.[12] Now, a merit function can be developed based on Eq. (5) which allows the GA to distinguish between systems with uniform and non-uniform output intensity profiles.

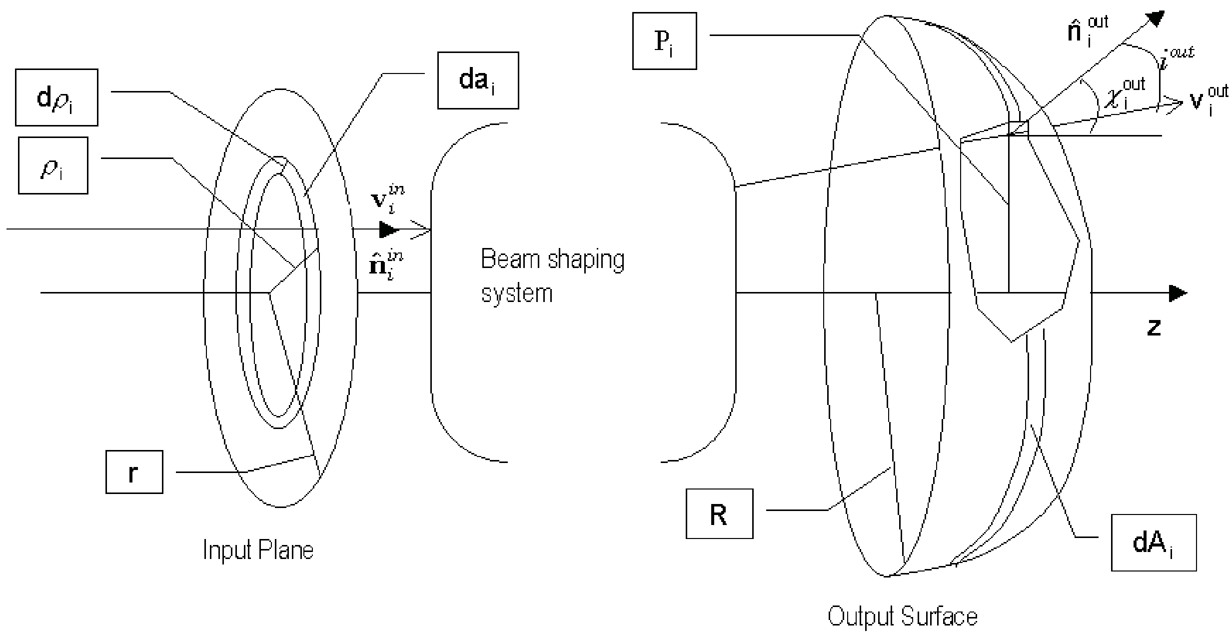

Figure 1. Geometrical configuration of beam expander with input and output surfaces.

The merit function is used by the GA optimization algorithms to distinguish between "good" systems and "bad" systems. The GA will find those systems with the highest value of the merit function. The merit function for these applications contains terms that quantify the diameter, the degree of collimation, and the uniformity of the irradiance profile of the output beam. To accomplish these objectives, the following merit function is defined:

$$M = M_{\mathrm{Diameter}} M_{\mathrm{Collimation}} M_{\mathrm{Uniformity}} \tag{6}$$

where

$$M_{\mathrm{Diameter}} = \exp\left[-s\left(\mathrm{P}_{\mathrm{Target}} - \mathrm{P}_N\right)^2\right]; \tag{7}$$

$$M_{\mathrm{Collimation}} = \exp\left[-\left(1 - \prod_{i=1}^{N}\cos^Q\left(\gamma_i\right)\right)\right]; \tag{8}$$

$$M_{\text{Uniformity}} = \frac{1}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[u(\mathrm{P}_i) - \left(\frac{1}{N}\sum_{i=1}^{N}u(\mathrm{P}_i)\right)\right]^2}} . \qquad (9)$$

In Eq.(7), P_N is the radial height (as measured from the optical axis) of the marginal ray on the output surface, and $\mathrm{P}_{\text{Target}}$ is the desired radius of the output beam. The exponential function is chosen since one wants the merit function to peak sharply at $P_N = \mathrm{P}_{\text{Target}}$, which indicates a "good" system during the GA optimization process. (See Ref. 15 for a detailed discussion this GA optimization process and construction of a suitable merit function to use when designing laser beam shaping systems.) The exponential accomplishes this nicely, but functions other than the exponential may have been chosen. The coefficient s determines the sensitivity of the merit function to the exit pupil radius constraint. That is, the smaller the value of s, the broader the exponent function becomes. In this example, s was set to 0.01. This value has been adjusted on occasion while the GA is executing to insure that the pupil radius constraint is satisfied. The merit function M also includes a term that favors each ray being perpendicular to the Output Plane (i.e., a collimated output beam). In Eq. (8), $\cos^Q(\gamma_i)$ is the cosine of angle γ_i that ray i makes with the optical axis. The exponent Q adjusts the sensitivity of the merit function to nonparallel [$\cos(\gamma_i) \neq 1$] rays. For this application, Q is set at six. If each ray is parallel to the optical axis, then $M_{\text{Collimation}} = 1$. If not, then Eq. (8) is less than one and reduces the merit function, which penalizes the set of system configuration parameters during the GA optimization process. The merit function M also includes a term that favors a uniform irradiance output beam profile. In Eq. (9), $\bar{u} = \frac{1}{N}\sum_{i=1}^{N}u(\mathrm{P}_i)$ is the "mean" of the values of the output intensity function, $u(\mathrm{P}_i)$ over N points on the output surface. As the beam profile on the output surface becomes more uniform, the denominator of Eq. (9) approaches zero, and $M_{\text{Uniformity}}$ increases substantially.

In summary, the merit function rewards those systems which tend to increase the value of M and penalize systems with smaller values of M as the GA optimization searches throughout both the discrete and continuous parameter space.

3. APPLICATIONS

The application of a GA generally must satisfy two prerequisites. First, one must identify those parameters that fundamentally characterize the system. Second, one must identify those features of a system which best describe the fitness (or "merit") of the system. This could be one particular attribute, such as focal length, or, on the other extreme, could involve the blending of many different attributes, each with different weights and measures of influence. Using the methods summarized above, one can adapt the GA to solve a range of problems. The beam shaping systems presented below illustrate how to build a method of using GAs to design simple to more complex beam shaping systems. Design of two beam shaping systems is discussed below - a two-plano-aspheric lens system and a three-GRIN lens system.

3.1. TWO PLANO-ASPHERIC LENS BEAM SHAPING SYSTEM

This problem is inspired by the system designed, built, and tested by Jiang, Shealy, and Martin in Refs.12, 13, and 14. The system has two plano-aspheric lenses, which shape an incoming Gaussian beam into an outgoing beam with a uniform irradiance profile, as shown in Figure 2. The system also expands the beam from 8 mm(ρ_N) to 12 mm $(\mathrm{P}_{\text{Target}})$. Both incoming and outgoing beams are parallel to the optical axis. The right surface of the first lens and the left surface of the second lens accomplish the beam expansion, collimation, and irradiance redistribution. In Ref. 12, the lens shaping surfaces are aspherical, described the optics surface equation:

$$z(h) = \frac{c\, h^2}{1 + \sqrt{1-(1+k)\, c^2 h^2}} + \sum_{j=2}^{6} A_{2j} h^{2j}, \tag{10}$$

where z is the sag of the surface; h is the radial height; c is the curvature of the surface; k is the conic constant and $A_4 \ldots A_{12}$ are aspherical deformation coefficients.

The choice of Eq. (10) for the surface sag allows the GA great flexibility in determining the surface contour of the two aspheric lens surfaces, depending on the number of deformation coefficients included in the optimization process. Furthermore, it is desirable to provide the GA with a large (multidimensional) parameter space to explore, since this is where GAs are especially powerful. The GA starts by randomly choosing, within predetermined constraints, values for the seven surface-shape parameters to be optimized. In each generation, ten individuals are produced. The parameters for each individual are passed to a ray-tracing routine (CODE V[17] is used for ray tracing in this application) where N rays are traced, and then, the value of the merit function is calculated for each individual system as described in section 2. The GA is given 14 parameters to optimize in this problem: c, k, A_4, A_6, A_8, A_{10}, and A_{12} for the two aspheric surfaces of each lens in the system.

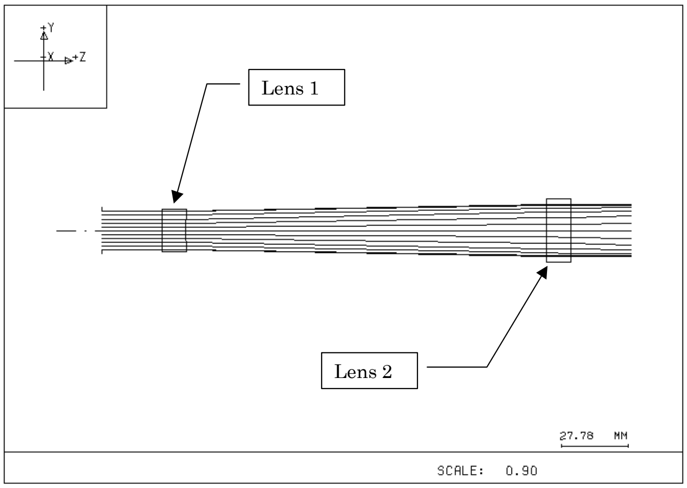

Figure 2. Two-lens beam shaper system with ray trace

The GA code was executed simultaneously on four Sun Sparcs, all running Solaris 2.6. The constraints were modified in real-time so that each instantiation of the GA code could search different regimes of the parameter space. Once it became apparent that a particular regime contained better solutions, the constraints were narrowed on all machines to search that regime more thoroughly. Also, when one machine found an individual that was substantially superior to the best individuals on the other three machines, the code on the three other machines was reinitialized using a restart file from the machine with the superior individual. This amounts to a primitive form of parallel processing. Total processing time was not rigorously recorded but was on the order of 50 hours.

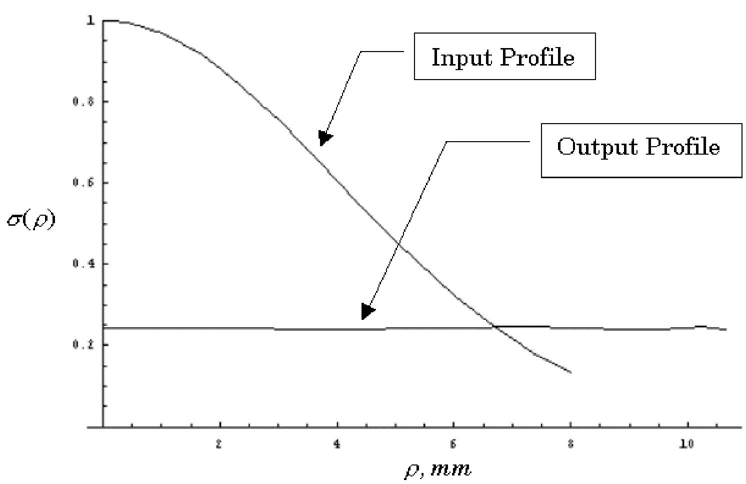

Figure 3. Input and output irradiance profiles for the two-lens beam shaper.

The fastest machine of the four, a Sun Ultra 1-170 with 64M of RAM, found the best individual. The search was stopped when no significantly better individuals were found over a period of several hours. The GA found a solution, which is presented in Table 1 and Table 2. Figure 3 compares the irradiance profiles of the input and output beam for this lens system. Integrating input beam profile, $\sigma(\rho)$, over the input plane yields 86.9 units. The mean value of the output beam profile, $\overline{u}$ = 0.242 rays per mm^2, with a standard deviation of 1.86 x 10^{-3} rays/mm^2, or 0.8% of $\overline{u}$. Integrating this mean value of the output irradiance over a plane perpendicular to the optical axis yields a value of 87.0 units. Therefore, conservation of energy is maintained for the designed beam shaping system to within the convergence accepted in this GA optimization.

Table 1. Two-lens Shaper System Parameters

Parameter	Value
Wavelength	589.00 nm
Radius of the input beam (entrance pupil diameter)	8.00 mm
Radius of the output aperture	10.7 mm
Glass type for two lens elements	CaF_2
Index of ambient medium (air)	1.0
Gaussian constant $2/\rho_N^2$ in Eq. (2)	0.031 mm^{-2}
Object distance	Infinity

Table 2. Two-lens Shaper Lens Element Parameters

Parameter	First lens element		Second element	
	Left surface	Right surface	Left surface	Right surface
Thickness	10 mm	150 mm	10 mm	25 mm
Vertex radius ($1/c$)	Infinity	100.59 mm	-100.01 mm	infinity
Surface type	spherical	Aspherical	Aspherical	spherical
Conic constant (k)		-0.922971	-0.375469	
A_4 (mm^{-3})		0.843226×10^{-5}	0.617298×10^{-4}	
A_6 (mm^{-5})		$-.664541 \times 10^{-6}$	$-.960417 \times 10^{-7}$	
A_8 (mm^{-7})		0.504624×10^{-8}	$-.164098 \times 10^{-8}$	
A_{10} (mm^{-9})		0.274667×10^{-14}	0.687429×10^{-11}	
A_{12} (mm^{-11})		$-.999878 \times 10^{-13}$	0.466018×10^{-14}	

3.2. THREE GRIN ELEMENT BEAM SHAPING SYSTEM

Although the previous example illustrates the GA's ability to arrive at solutions for a well-understood problem, the first example does not fully harness the *creative* ability of the GA method. If the GA were given more flexibility, including choosing the actual number and types of GRIN elements, for example, rather than just their shapes, could the GA actually produce a solution in a reasonable amount of time? In searching for a problem to address this question, there are two encumbrances of the GRIN system developed by Wang and Shealy.[7] First, their GRIN beam shaping system requires a connector piece of glass to permit the marginal ray to trace parallel to optical axis, and then, their GRIN profiles can not be associated with available glass types. Evans has shown[11, 15] that GA optimization can find catalog glasses for the GRIN elements of the beam shaping system considered by Wang and Shealy. However, the resulting connector element is long and allows rays to focus within the connector, that is an undesirable feature from a cost and fabrication perspective, since the connector could be damaged if a high-power laser is used.

Thus, a new problem may be defined: is there a system that shapes an input Gaussian beam to a top-hat beam profile in manner similar to that of the original Wang and Shealy GRIN system, but without requiring a connector piece between the GRIN elements? That is, is there a shaping system that uses GRIN elements separated only by ambient air? Furthermore, if the GA is no longer limited by specifying so many aspects of the optical system *a priori*, such as the actual number of elements present in the system, can the GA harness this greater flexibility in determining the makeup of the optical system to solve the problem? To do this, the GA not only must optimize surface shapes of the GRIN elements and their spacing, but also must determine the actual number of elements in the solution, up to a certain limit (four, in this case). In essence, to solve this problem, the GA must do the same things that a human optical designer would. The number of GRIN elements must be determined, the type of GRIN material for each element must be picked, and so on. This new type of problem further distinguishes the GA method from deterministic methods (i.e., those that rely on derivatives and a smooth, continuous merit function) since the merit function required for this problem is a complicated mix of discrete and continuous parameters.

The merit function is constructed in a similar manner as described in section 2. It contains three key terms: one term that measures the uniformity of the output profile, one term that insures that all rays exit the system parallel to the optical axis, and finally, one term that penalizes systems that do not have the specified radius on the output surface (4 mm, in this case). One unique aspect of this application is the way that the GA is allowed is select the actual number of elements for each test system. This variable, N_{Element}, may be one of the following integers: one, two, three, or four. An idiosyncrasy of the particular GA code[18] used in these examples is that parameters to be optimized are defined by default as 8-bit real numbers. The parameter in the code that corresponds to N_{Element} (an integer) is $P(i,1)$ (an 8-bit real number), where i refers to the one of the ten individuals systems in a generation. For this problem, there are actually 26 parameters to be optimized, and the population size of each generation is 10. Thus, P is a array with 10 rows and 26 columns. In order to translate $P(i,1)$ into an integer within the desired range, the limits are set as follows:

$$0.50 \le P(i,1) \le 4.49\,; \tag{11}$$

and, the following function is used for translation:

$$N_{ele}(i) = \text{int}\left[P(i,1)\right], \tag{12}$$

where $\text{int}(a)$ is a function that rounds a to the closest integer. Using Eqs. (11) and (12), an integer between and including one and four is chosen, each with equal probability.

It is this method that is used to define discrete parameters for the GA in all cases that require them. In this problem, there are eight other discrete parameters: the GRIN glass types and the GRIN direction for each of the (up to) four elements. These glass types are selected from four LightPath gradient glass types.[19] See lines 34 and 40 in Appendix A of Ref. 15 for an example of this coding. See Table 3 for a description of all parameters optimized in this problem. It should be noted that if the GA chooses less than four elements for a particular individual, the remaining surfaces are made into dummy surface in the code. For example, if $P(i,1) = 3$, then the following occurs: first, the left and right surface of element four is set to have an infinite radius of curvature (i.e., the surfaces are made flat). Then, the thickness of element four is set to zero and the glass type is set to air. The net result is that element four has no effect on the optical system, while the surfaces for element four remained defined for future use, should they become necessary. This method simplifies coding of the problem and makes evaluation of the merit function more efficient. See Appendix B of Ref 15 for an example of this in code.

The code was executed on a Sun Ultra 1 170 with 64 M of RAM as before. Using this setup, it took an average of 7.80 seconds for the GA to completely evaluate the merit function for each of the 10 test beam shaping systems (individuals) in a generation. The code was allowed to run until it reached generation number 12,367, resulting in an effective total run-time of 26.8 hours. The merit function, which is measured in arbitrary units, peaked at a value of $M = 101.21$. The fact that this value represents the best solution to the problem can be seen in Figure 4, where it is clear that M_{best} asymptotically approaches a value of about 102. The system with a merit function value of $M = 101.21$ is represented in Figure 5. The system has three elements, all with spherical surfaces. Most importantly, this solution reshapes the Gaussian input profile into a irradiance profile with uniform intensity on the output surface. The input and output profiles are shown in Figure 6. The parameters for the systems are given in Table 4 and Table 5. The mean value of the output irradiance shown in Figure 6 is $\overline{u} = 4.55 \times 10^{-2}$ rays per mm^2 with a standard deviation of 1.70×10^{-3} or 3.7% of $\overline{u}$. Integrating this mean value of the irradiance over the output plane yields 21.9 units. Similarly, integrating the $\sigma(\rho)$ over the input plane yields 21.7 units. Therefore, conservation of energy is maintained for this GRIN beam shaping system to within the convergence accepted by this GA optimization.

To review, the GA method produced a GRIN beam shaping system that only has spherical surfaces, requires no connectors, and uses GRIN lenses from a catalog of an established GRIN manufacturer. The GA solved a problem that would be difficult to solve using analytical methods or conventional optimization techniques, since the merit function contains discrete parameters (for example, picking the GRIN glass type from a predefined set of GRIN elements). Furthermore, the GA was presented with a unique problem that has not been solved before and was allowed a certain degree of creativity in producing a solution. The result is a three-element system that is wholly the creation of the GA and has a form that could not have been anticipated before the fact.

Table 3. Optimized Parameters for the Free-form GA-Designed GRIN Shaper Problem.

Para-meter	Parameter description	Parameter type	Parameter limits20
1	Number of elements	Discrete	1-4 (integer)
2	Radius of curvature of left surface of Element 1	Continuous	-100 to 100
3	Radius of curvature of right surface of Element 1	Continuous	-100 to 100
4	Thickness of Element 1	Continuous	1 to 10
5	Distance between Element 1 and Element 2	Continuous	1 to 10
6	GRIN glass type for Element 1	Discrete	1-6 (integer)
7	Positive or negative GRIN for Element 1	Discrete	0, negative or 1, positive
8	Radius of curvature of left surface of Element 2	Continuous	-100 to 100
9	Radius of curvature of right surface of Element 2	Continuous	-100 to 100
10	Thickness of Element 2	Continuous	1 to 10
11	Distance between Element 2 and Element 3	Continuous	1 to 10
12	GRIN glass type for Element 2	Discrete	1-6 (integer)
13	Positive or Negative GRIN for Element 2	Discrete	0, negative or 1, positive
14	Radius of curvature of left surface of Element 3	Continuous	-100 to 100
15	Radius of curvature of right surface of Element 3	Continuous	-100 to 100
16	Thickness of Element 3	Continuous	1 to 10
17	Distance between Element 3 and Element 4	Continuous	1 to 10
18	GRIN glass type for Element 3	Discrete	1-6 (integer)
19	Positive or negative GRIN for Element 3	Discrete	0, negative or 1, positive
20	Radius of curvature of left surface of Element 4	Continuous	-100 to 100
21	Radius of curvature of right surface of Element 4	Continuous	-100 to 100
22	Thickness of Element 4	Continuous	1 to 10
23	Distance between Element 4 and Surface 10 (a dummy surface)	Continuous	1 to 10
24	GRIN glass type for Element 4	Discrete	1-6 (integer)
25	Positive or negative GRIN for Element 4	Discrete	0, negative or 1, positive
26	Distance from Surface 10 (a dummy surface) to the output plane	Continuous	1 to 100

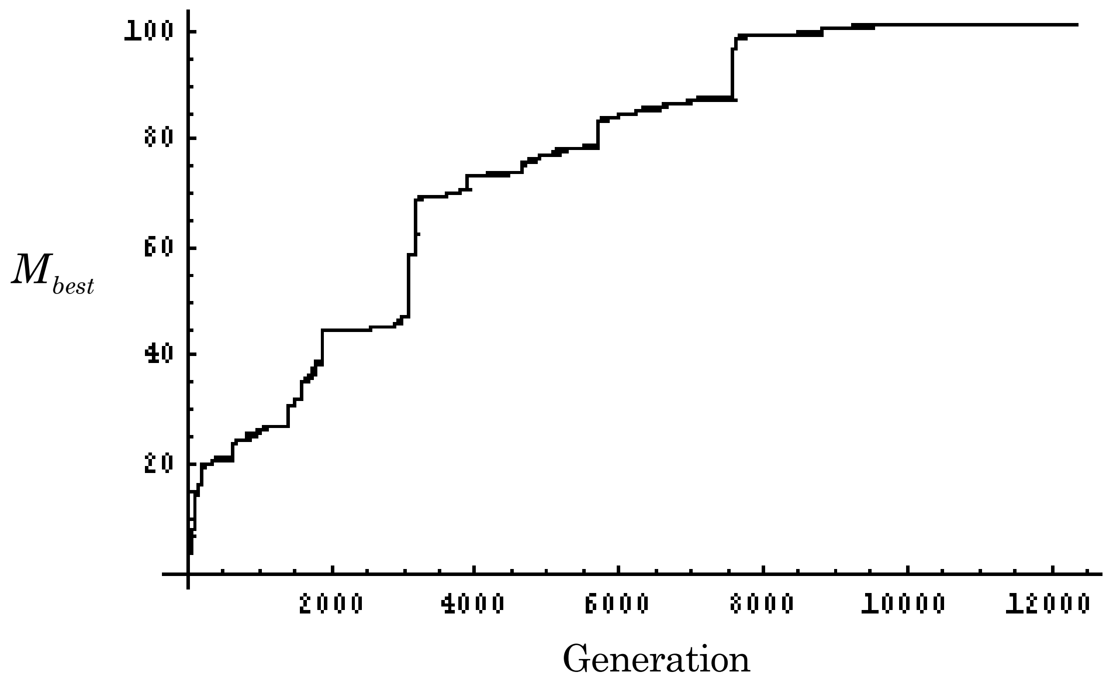

Figure 4. A plot showing the best individual in a generation as a function of generation.

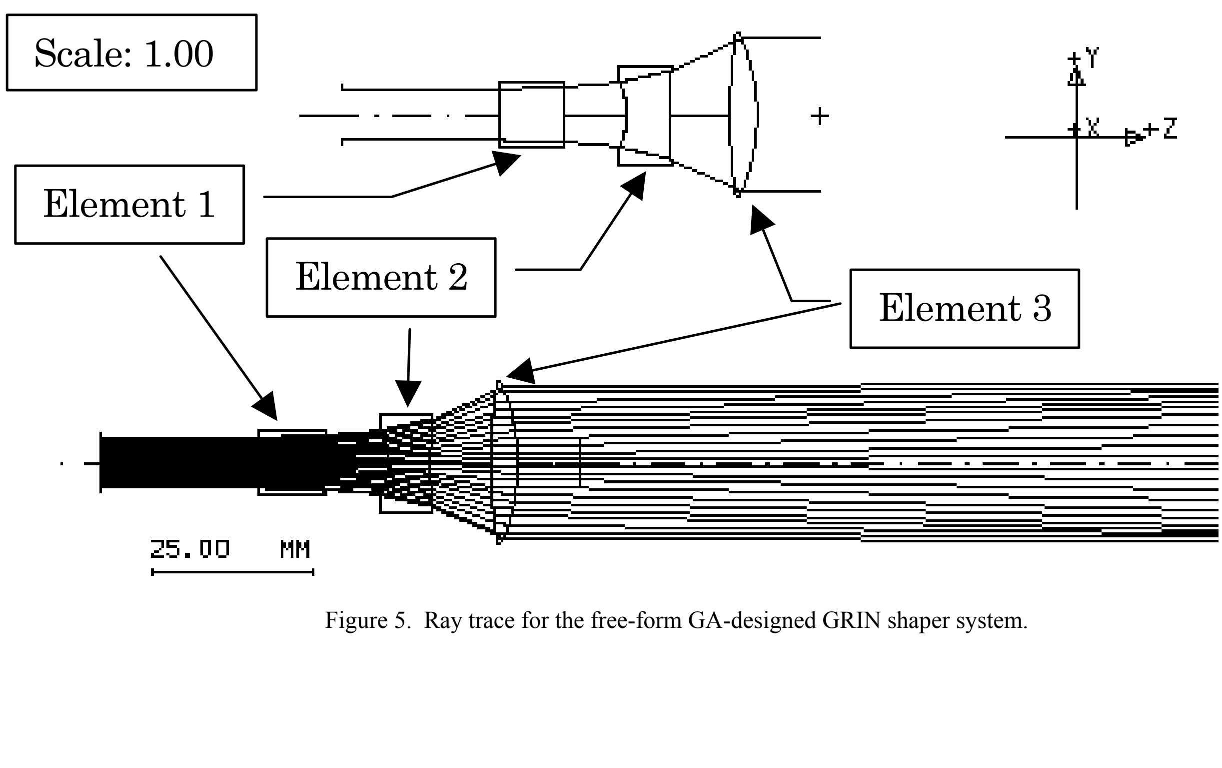

Figure 5. Ray trace for the free-form GA-designed GRIN shaper system.

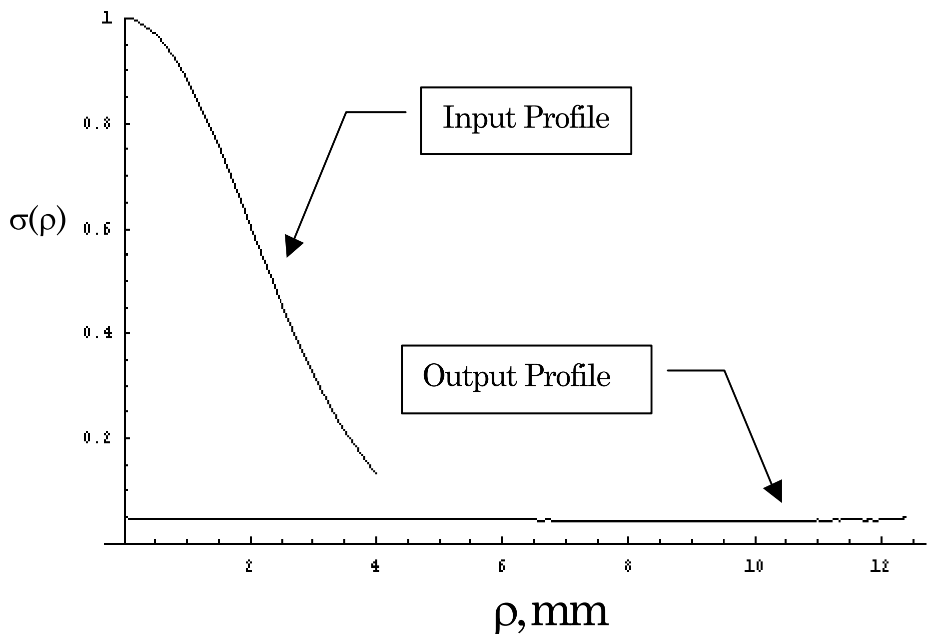

Figure 6. Input and output irradiance profiles for the free-form GA-designed GRIN Shaper.

Table 4. Free-form GA-Designed GRIN Shaper Parameters

Parameter	Value
Wavelength	589.00 nm
Radius of the input beam (entrance pupil diameter)	4.00 mm
Radius of the output aperture	12.4 mm
Index of ambient medium (air)	1.0
Gaussian constant $2/\rho_N^2$ in Eq. (2)	$0.035\ \mathrm{mm}^{-2}$
Number of elements	3
Distance from Surface 10 to output plane (parameter 26 in Table 3)	100 mm
Object distance	Infinity

Table 5. Free-form GA-Designed GRIN Shaper Lens Parameters

Parameter	First element		Second element		Third element	
	Left surface	Right surface	Left surface	Right surface	Left surface	Right surface
Thickness, mm	9.99	10.0	6.77	9.48	4.25	9.71
Vertex radius ($1/c$), mm	-61.6	80.4	-12.5	100	87.5	-30.3
Surface type	spherical	spherical	spherical	Spherical	spherical	spherical
Glass Type (UDG C1)[21]	3		2		1	
GRIN Direction	negative		positive		negative	

4. SUMMARY AND CONCLUSIONS

This work started with the following assumption: there are some problems in geometrical optics that cannot be solved using either analytical methods, such as the differential equation method,[2, 12, 13, 14] or conventional computational optimization techniques, such as the simplex algorithm.[22] There are numerous reasons that account for this, ranging from merit functions that contain non-continuous parameters to problems that simply have no solution. To this end, the following hypothesis was proposed: special machine learning codes - in particular, GAs - can be adapted to solve some of these problems. The first step in testing this hypothesis is to establish that GAs can indeed produce solutions to problems in geometrical optics. This was done by addressing a few well-understood problems from the literature, which have been solved by other methods. In addition to establishing a proving ground, these exercises shed light on the myriad subtle qualities that govern the GA optimization method.

The first problem (two-plano-aspheric beam shaper) presented here has been solved by more conventional methods by Jiang, Shealy, and Martin via an analytical differential equation method. [2, 12, 13, 14] The GA method indeed produced solutions to these problems and did so within a reasonable amount of time. Yet, this solution requires highly aspherical lens surfaces. This greater flexibility makes finding a solution easier, both for the GA method and analytical methods. However, choosing aspherical surfaces comes at a price; they are both expensive to manufacture and difficult to build to precise specifications.

With this in mind, Wang and Shealy[7] endeavored to produce a beam shaping system similar to Jiang, Shealy, and Martin's[12] two-beam shaper but with the additional constraint that only the solution contains only spherical elements. To compensate for the decreased flexibility as a result of requiring spherical surface shapes, axial gradient-index glasses were used for the shaping elements. Using a differential equation design method, Wang and Shealy produced the desired shaping system. While the system performs its primary function of shaping a Gaussian input beam into a uniform profile on the output surface, it has an undesirable feature. Namely, the parameters defining the index of refraction function for each of the GRIN elements were allowed to vary continuously, resulting in GRIN elements that could not be matched with elements found in established GRIN manufacturer catalogs. That is, the GRIN elements must be custom fabricated, adding significantly to the cost required to build that system.

To address this issue, the GA was given the task of producing a system similar to the system designed by Wang and Shealy, except that the GRIN elements were to be chosen from a predefined set.[11, 15] This set is simply a catalog of elements from a popular GRIN manufacturer. LightPath[19] was used in this case, though any catalog would have been sufficient. With this constraint, the problem contains parameters that can only assume certain values and, hence, do not vary smoothly. This aspect makes this problem difficult to solve with methods that rely on continuous merit functions, which eliminates the differential equation method used by Wang and Shealy. This is the first problem where the GA was used to produce a truly unique solution to a difficult problem and is described in Ref. 15. The system found by the GA contains only spherical surfaces, has GRIN elements chosen by the GA from a catalog, and shapes the Gaussian input beam into one with a uniform irradiance profile on the output surface, as desired. However, it also has one unanticipated feature. The marginal rays actually focus within the connector. New terms might be added to the merit function to insure that rays do not focus within the connector, after which the entire optimization process could be restarted to find a new solution. Rather than follow this course, however, one finds that the accidental discovery of this feature shows the way to a new, more ambitious problem: the total elimination of the connector.

In the free-form GA-designed laser beam shaping system discussed in section 3.2, the goal was to produce a system that has the following features. First, it must contain elements with spherical surfaces only. Second, it must contain elements that are chosen from a GRIN catalog. Third, it must require no connectors, and the rays should not focus within any elements (since such focusing of a high-powered laser could destroy the element). Finally, of course, it must shape the beam into a uniform output profile. In presenting the problem to the GA, no assumptions were made as to the number of elements required to solve the problem. The GA was allowed to function as a human optical designer would - trying two elements in this system, four in another, and evaluating the merit function in each case to see what improvements, if any, resulted. This problem demonstrates more than any other presented here how the GA is an example of machine learning. Essentially, information about past successes is preserved in the best individual of each iteration, and this information persists from generation from generation. Progress is made, often in small steps, but sometimes in great leaps (see Figure 4). The final result is a system not conceived by the human designer, but a system that, nevertheless, functions as required.

APPENDIX: SUMMARY OF GA OPTIMIZATION

Generally, the idea behind optimization is that one has some function f that can be evaluated easily. This function is expressed in terms of several variables, which may be discrete or continuous. One wishes to find the values of these variables for which f assumes either a maximum or minimum value. The difficulty of the problem is related to whether one is searching for *local* extrema, of which there may be many, or the *global* extrema, which represent the absolute best solutions. The complexity of the problem is related to the number of variables which make up f, in addition to the ease with which f can be calculated.[23] The greater the complexity of the problem, the longer it takes to arrive at a solution. Thus, search algorithms that arrive at solutions quickly are to be coveted, which is evident by the voluminous amount of research regarding the subject present in the literature.[24, 25] In this work, the GA search method is presented as one of these treasured methods, but it should be noted that other algorithms exist which may produce similar, if not superior performance. Two popular alternative methods are simulated annealing and the Tabu search.[26] A discussion of the theory of optimization as it applies to optics is presented in Ref. 15.

Though there are myriad optimization techniques to choose from, methods such as GAs and simulated annealing are of particular interest because of their ability to solve combinatorial minimization problems. The key feature of such problems is that one or more of the parameters that make up the merit function (which is to be maximized) are discrete, in the sense that they only can assume particular values from a predefined set of allowable values. Thus, instead of an N-dimensional space made up of N continuous parameters, one is presented with a parameter space whose complexity is factorially large—so large in fact that it cannot be completely explored.[27] Without a continuous merit function, concepts such as "downhill" and "uphill" lose their meaning and other optimization techniques, such as the simplex method,[28] can no longer be applied. For example, the GRIN beam shaping system presented in Section 3.2 was optimized using only the glass types for the lens elements from a fixed set of GRIN materials found in a manufacturer's catalog. It is here that GAs and simulating annealing techniques excel, though they also can be applied to problems that are purely continuous. In the literature, there are numerous examples of problems solved using these techniques,[29, 30] as well as research that compares the performance of one or more of these methods on the same class of problems.[31, 32, 33, 34] It seems that of the three methods discussed here, no one method is necessarily more efficient than the others, though it does appear that GAs and the Tabu search tend to arrive at solutions more quickly than simulated annealing methods, at least in the papers cited here.

It should be noted that several commercial optical design and analysis packages implement these techniques in their optimization routines to varying degrees. OSLO,[35] for example, uses an "adaptive simulating annealing" method. ZEMAX[36] and CODE V[17] also contain proprietary global optimization methods. The problem with these implementations, among other things, is that the merit functions in these packages are oriented towards imaging systems (ZEMAX, however, allows for user-defined merit functions computed by macros or an external programming interface), limiting one's ability to manipulate the merit function for one's own purpose. Also, since the makers of these packages keep their optimization codes proprietary, one's ability to customize those routines is all but eliminated.

Since GAs are based on a biological paradigm, much of the GA nomenclature is borrowed directly from evolutionary biology. The reader may find it useful to have some of this jargon expressed in terms more familiar to the optics community. As discussed above, GAs produce a finite number of test solutions to a problem. Individually, these solutions are referred to as "organisms" (or just "individuals"), and collectively as a "generation." A "generation" is essentially an iteration. With each iteration, the merit function, M, is evaluated for each member of a generation. There may be a few as five or as many as hundreds of individuals per generation, depending on the code used and how it is configured. For the applications explored here, there are typically five or ten individuals in a generation. A new generation of child systems is produced from the genetic material of the parent generation (the specifics of this process are described below). An individual's genetic code represents a particular system prescription. For example, in the two-lens beam shaper/expander example presented in section 3.1, seven parameters, that collectively define one surface of the beam-shaping element, are concatenated into a string (i.e., genetic code). Thus, with each iteration, five or ten new system prescriptions are produced and their respective merit functions evaluated.

The GA method developed here is based on a micro-GA code.[18, 37, 38] Micro-GAs have several features that distinguish them from other GA codes. The most prominent of these features is the fact that micro-GAs can operate efficiently with small generation sizes (on the order of 10 individuals per generation). This is important for the applications presented here, since evaluation of the merit function for each individual is a very time-consuming process. The micro-GA is able to work with small generation sizes by checking for "stagnancy" in each generation it produces. Stagnancy is determined by taking the average value of the merit function for all the individuals in a generation, $\bar{M}(t)$, and comparing it to average merit functions values for N parent generations, $\bar{M}(t-1)$, $\bar{M}(t-2)$, ..., $\bar{M}(t-N)$. If these values do not differ significantly (a parameter which can be set in the micro-GA, and is usually "tweaked" at the beginning of a problem to produce the most efficient code), then the population is defined as stagnant. Essentially, when stagnancy is detected, the code assumes that the GA is stuck in a local minima and attempts to add some randomness to the process. When such a situation arises, the GA picks the best of the individuals in a generation, kills the remaining and replaces them with new, randomly selected individuals in the child generation.

"Reproduction" is defined as producing a child generation of new individuals from the genetic material of a parent generation. The individual with the highest value of M in a particular generation is most likely to have its genetic material passed on to the next generation. The "genetic material" for a particular individual is defined by concatenating the binary value for each parameter to be optimized into a binary string (consists of only ones and zeros). See Figure 7 for an example. New generations are created by a "crossover operator", which swaps chunks of genetic material (strings) between two or more individuals in a generation. With the Micro-GA code, crossovers are done in a manner that maintains individual "alleles". An allele is the particular value that a parameter assumes, expressed in string format.[39] When a crossover occurs, the strings that represent alleles are not broken into pieces, but are transferred from one individual to another intact. Another operation the GA performs is "mutation." Here, the GA randomly selects one or several bits in an allele, and changes the state of these bits. Since the string is binary, this amounts to operating on the bit with "Not" (not 0 = 1, not 1 = 0). Mutation adds a built-in randomness to the GA method, which helps the GA avoid local minima. Because of the stagnancy-checking feature it employs, the Micro-GA allows one to avoid the constant tweaking of GA parameters (e.g., crossover and mutation rates) which is often necessary with conventional routines.

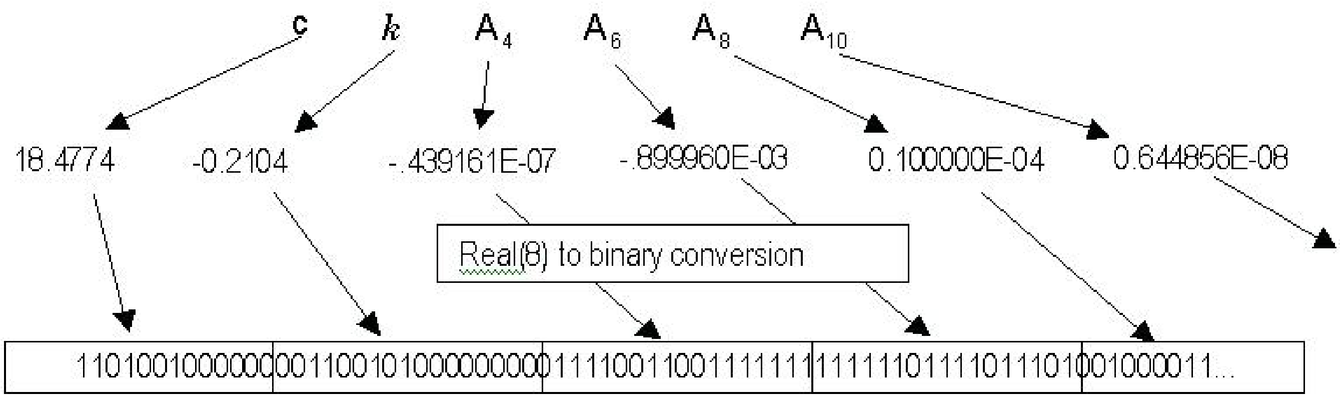

Figure 7. Example of the genetic material for a single individual. Values (Real*8) for each parameter are converted into binary strings, which are in turn concatenated into one long string, the genetic material for that individual (from Ref.**10**). Individual alleles (the values that surface coefficients, such as, curvature, conic constant, and aspherical deformation constants, expressed in binary form) are kept intact when crossovers occur.

REFERENCES

1. D. L. Shealy, "Geometrical Methods," in *Laser Beam Shaping: Theory and Techniques,* Marcel Dekker, Inc., New York, in print.
2. D.L. Shealy, "Theory of Geometrical Methods for Design of Laser Beam Shaping Systems," *Proc. SPIE* **4095-01**, 2000.
3. P.W. Malyak, "Two-mirror unobscured optical system for reshaping the irradiance distribution of a laser beam," *Appl. Opt.* **31**, pp. 4377-4383, 1992.
4. P.W. Rhodes and D.L. Shealy, "Refractive optical systems for irradiance redistribution of collimated radiation: their design and analysis," *Appl. Opt.* **19**, pp. 3545-3553, 1980.
5. D. F. Cornwell, *Non-projective transformations in optics*, Ph.D. Dissertation, University of Miami, Coral Gables, 1980.
6. J.L. Kreuzer, "Coherent light optical system yielding an output beam of desired intensity distribution at a desired equiphase surface," U. S. Patent 3,476,463, 4 November, 1969.
7. C. Wang and D.L. Shealy, "Design of gradient-index lens systems for laser beam shaping," *Appl. Opt.* **32**, pp. 4763-4769, 1993.
8. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, pp. 1-6, Addison-Wesley, Reading, MA, 1989.
9. D. E. Goldberg, p. 7.

10. N.C. Evans and D.L. Shealy, "Design and optimization of an irradiance profile shaping system with a genetic algorithm method," *Appl. Opt.* **37**, pp. 5216-5221, 1998.
11. N.C. Evans and D.L. Shealy, "Optimization-based Techniques for Laser Shaping Optics," in *Laser Beam Shaping: Theory and Techniques,* Marcel Dekker, Inc., New York, in print.
12. W. Jiang, D. L. Shealy, and J. C. Martin, "Design and testing of a refractive reshaping system," in *Current Developments in Optical* Design *and Optical Engineering III*, Robert E. Fischer and Warren J. Smith, eds., *Proc. SPIE* **2000**, pp. 64-75, 1993.
13. W. Jiang, *Application of a laser beam profile reshaper to enhance performance of holographic projection systems*, Ph.D. Dissertation, The University of Alabama at Birmingham, 1993.
14. W. Jiang and D.L. Shealy, "Development and Testing of a Refractive Laser Beam Shaping System," *Proc. SPIE* **4095-19**, 2000.
15. N.C. Evans, *Genetic Algorithm Optimization Methods in Geometrical Optics,* Ph.D. Dissertation, The University of Alabama at Birmingham, 1999.
16. M. Born and E. Wolf, *Principles of Optics*, Fifth Edition, p. 115, Pergamon Press, New York, 1975.
17. CODE V is a registered trademark of Optical Research Associates, 3280 E. Foothill Blvd., Pasadena, CA 91107.
18. See the Naval Research Lab's GA archive, http://www.aic.nrl.navy.mil/galist/, for a comprehensive source of GA codes in several different languages and implementations. This site also is the home of a very useful mailing list in which one may find the current state of the art in GA techniques and theory.
19. The LightPath user-defined GRIN glass catalog within CODE V was used. For more information on this glass catalog, contact Optical Research Associates or LightPath Technologies Inc., 6820 Academy Parkway, NE, Albuquerque, NM 87109.
20. For continuous parameters, numbers are real and are in units of mm.
21. UDG C1 is a CODE V variable that refers to "User-defined glass constant 1". C1 corresponds to a particular GRIN in the CODE V Lightpath catalog. The CODE V Lightpath GRIN glass catalog can be obtained from Optical Research Associates.[17]
22. G. B. Dantzig, *Linear Programming and Extensions*, Princeton U.P., Princeton, N. J., 1963
23. W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Chap. 10, pp. 387-388, Cambridge U.P., Cambridge, U.K., 1992.
24. Dennis, J.E. and Schnabel, R.B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
25. Brent, R.P., *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
26. L. B. Morales, "Scheduling a bridge club by tabu search," Mathematics Magazine **70**, pp. 287-290, 1997.
27. W. H. Press, pp. 436-437.
28. G. B. Dantzig, *Linear Programming and Extensions*, Princeton U.P., Princeton, N. J., 1963.
29. T. Etzion and P.R.J. Ostergand, "Greedy and heuristic algorithms for codes and colorings," *IEEE Trans. on Information Theory* **44**, pp. 382-388, 1988.
30. R.H.J.M. Otten and L.P.P.P. van Ginneken, *The Annealing Algorithm*, Kluwer, Boston, 1989.
31. R. R. Brooks, S. S. Iyengar and J. Chen, 'Self-calibration of a noisy mulitple-sensor system with genetic algorithms," in *Self-Calibrated Intelligent Optical Sensors and Systems*, Anbo Wang, ed., *Proc. SPIE* **2594**, pp. 20-38, 1996.
32. H. Kim, Y. Hayashi and K. Nara, "An algorithm for thermal unit maintenance scheduling through combined use of GA, SA and TS," *IEEE Trans. on Power Systems* **12**, pp. 329-335, 1997.
33. F. S. Wen and C. S. Chang, "Tabu search approach to alarm processing in power systems," *IEEE Proceedings: Generations, Transmission and Distribution* **144**, pp. 31-38, 1997.
34. H. J. Jensen, *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological System,* pp. 34-71, Cambridge U. P., New York, 1998.
35. OSLO is a registered trademark of Sinclair Optics, Inc., 6780 Palmyra Road, Fairport, NY, 14450.
36. ZEMAX is a registered trademark of Focus Software, Inc., P.O. Box 18228, Tucson, Arizona, 85731.
37. The GA code is based on ga164.f, D.L. Carroll's FORTRAN Genetic Algorithm Driver. See http://www.staff.uiuc.edu/~carroll/ga.html to download ga164.f, along with documentation and useful information on its implementation. One may find several sites related to ga164.f by using it as a keyword on popular search engines. Ga164.f is based on a micro-GA method. See Ref. 38 for more information on micro-GAs.
38. K. Krishnakumar, "Micro-genetic algorithms for stationary and non-stationary function optimization," *SPIE: Intelligent Control and Adaptive Systems* **1196**, 1989.
39. D. E. Goldberg, p. 21.