Optik - International Journal for Light and Electron Optics 178 (2019) 1016–1022

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.com/locate/ijleo

Original research article

Open-source optimization algorithms for optical design

Furkan E. Sahin

Maxim Integrated Inc., San Jose, CA, 95134, United States

ARTICLE INFO

Keywords:
Optical design
Optimization
Algorithms
Lens design
Cloud computing

ABSTRACT

Driven by the growth of internet, commercial cloud computing now provides on-demand, massively parallel computational power. This cloud processing can be utilized for solving science and engineering problems. Relying on computationally intensive ray-tracing, optical design can greatly benefit from a cloud based optimization approach if suitable parallelizable optimization algorithms can be implemented. Commercial optical design packages have algorithms tuned for optical design. There are also various local and global optimization algorithms for scientific and engineering problems. In this paper, several general purpose, open-source local and global optimization algorithms are evaluated for a lens design problem. The results are compared to results from a commercial optical design package, and some algorithms are shown to give similar performance. This research provides valuable insight into different optimization algorithms and therefore is a first step for taking optical design from desktop computers to cloud computing in order to achieve massively parallel processing in system optimization.

1. Introduction

Optical lens design is the art and science of designing a system that can provide high performance while also meeting constraints such as physical size, cost, manufacturing limitations and tolerances. This is typically achieved through a careful iterative procedure of defining optical constraints and performance targets, constructing a suitable merit function that comprises these constraints and performance targets, minimization of this merit function value to achieve an optimal design, and tolerance analysis of the design to predict the as-built performance. A large number of rays are traced through an optical system to perform optical analyses and calculate the merit function value. Due to the high computational complexity of ray-tracing, computers have been widely used for optical design since their early days [1,2]. However, even using modern high-speed computers with extreme processing power, competent user intervention and guidance typically play a key role in arriving at a well-balanced and optimized design solution [3].

Over the years, commercial optical design software packages have been developed with integrated sequential ray-tracing, lens optimization, tolerancing and complex analysis tools. These software packages have proprietary optimization algorithms developed for efficient and accurate optical design [4–6]. These algorithms may include local or global optimization routines. Local optimization typically requires the starting lens design to be close to the optimal result, but finding this initial lens design is not easy [7]. Therefore, global optimization algorithms have been developed to allow optimization to start from a lens design far from the optimal result. Most lens design programs implement some variant of damped least squares algorithm [8] for local optimization and have proprietary algorithms for global optimization.

Among popular optical design packages, CODE V [4] has a local optimizer called Automatic Design, and a global optimizer called Global Synthesis. OpticStudio (Zemax OpticStudio [5]) has one local optimizer function (that can either use damped least squares or orthogonal descent algorithms for optimization) and two global optimizers called Hammer Optimization and Global Search.

E-mail address: furkan.sahin@maximintegrated.com.

https://doi.org/10.1016/j.ijleo.2018.10.073
Received 6 May 2018; Received in revised form 31 August 2018; Accepted 11 October 2018

SYNOPSYS has optimizers called DSEARCH for fixed focus and ZSEARCH for zoom lenses [6]. These are all proprietary software packages running on Windows operating system.

Commercial optical design packages constantly refine and improve their optimization algorithms. CODE V introduced Step Optimization algorithm in recent years, in order to obtain faster convergence to an optimal solution compared to traditional damped least squares optimization. Innovative optimization techniques and algorithms from the scientific community include PSD III (pseudo-second-derivative) method by Dilworth [9], global optimization with escape function by Isshiki [10] and saddle point method by Bociort [7,11]. PSD III method is implemented in SYNOPSYS software package (D.C. Dilworth is the inventor of the PSD III method and also the developer of the SYNOPSYS software package) and is claimed to be the fastest lens optimization in the world [12]. Even though the PSD III method is described in [9], the exact implementation of the method in SYNOPSYS software is not provided as publicly available source code. Therefore, it could not be included in the comparisons provided in this paper.

On the other hand, many general purpose local and global optimization algorithms have been developed for a variety of scientific and engineering problems [13,14]. These open-source algorithms can be used for optical design as well. Once a merit function of a lens design is constructed, these algorithms can perturb the optimization variables to minimize the merit function value. With many options available, it is not straightforward to select a suitable algorithm. Algorithms also have many parameters that can be adjusted to produce fast and accurate convergence for a specific optimization problem.

With advances in design, manufacturing and testing, modern lenses are more complex than traditional designs. Lens designs have more optimization variables, either with more lens elements or with high-order aspherical surfaces. Lens design optimization is further sophisticated with new techniques, such as integrating manufacturing tolerances into optimization in order to achieve minimal performance degradation with as-built lenses [15,16] or incorporating computational photography steps into lens design stage [17–19].

In this paper, several open-source optimization algorithms are implemented for a simple lens design problem, *i.e.* optimization of a triplet lens, and a comparative review of these algorithms is presented. With surface curvature, element and airspace thickness values and element glass selection, even the triplet lens is not a trivial optimization problem. Although most lens design problems have more optimization variables, optimization of the triplet provides valuable insight about characteristics of suitable optimization algorithms.

Ray-tracing, merit function generation and evaluation operations are achieved with a commercial optical design software package (Zemax OpticStudio [5]). External optimization algorithms are implemented in Python programming language and interfaced to OpticStudio.

New parallelizable optimization algorithms that can run on scalable cloud computing systems provide a next-generation solution to optical design optimization problems. Using appropriate optimization algorithms and with a massively parallel, cloud computing approach, optical design optimization will be significantly accelerated. The local and global optimization algorithms tested in this paper are open-source and can also be implemented with a population-based generalized island model for parallelization [20]. They can therefore be quickly programmed for cloud processing platforms in order to take advantage of multiprocessor architectures.

2. Lens design optimization

The purpose of optical lens design is to find an optimal set of lens parameters that satisfy a set of system (*e.g.* effective focal length, *efl*) and physical (*e.g.* element center thickness) constraints and also provide high optical performance (*i.e.* low aberrations). Variables of the optimization problem are the parameters of the lens, such as surface curvatures, element thicknesses and spacings, element materials and surface aspherical coefficients. In order to achieve an optimal design, typically a least squares merit function (*MF*) is defined as the sum of many optimization operands, and the optimizer tries to minimize this merit function.

One way to define the merit function (*MF*) is:

$$\mathrm{MF}^2 = \frac{\sum W_i (P_i - T_i)^2}{\sum W_i} \tag{1}$$

in which, W is the weight, P is the present value, T is the target value of a particular operand and i is the subscript that indicates different optimization operands [5].

The optical designer preferably starts the optimization process with a lens design that is close to the optimal solution, so that an optimal solution can be reached with local optimization. This initial lens form can be inspired from a traditional design or a patent lens. Then the designer determines optimization variables in the system, and constructs the merit function. The merit function operands can either be system and physical parameters or image quality related operands. The operand values are usually calculated by tracing many rays through the lens.

As an example, the target *efl* value for the lens can be defined by the optical designer and a corresponding optimization weight can be assigned. If the *efl* of the initial design is not equal to the target *efl*, then this disparity will contribute to the merit function. As the optimization progresses, if the *efl* is close to the target *efl*, then the contribution to the merit function value will be reduced. Similarly, one can also define image quality related operands. As an example, in order to minimize the on-axis spot size, many rays at 0° object field angle can be traced through the lens. Ideally, all these rays should focus at the center of the image plane. The physical distance between each ray's position at the image plane and the center of the image plane can be defined as operands of the merit function. Once the optimizer minimizes this merit function, all the on-axis rays should be focused to a tight spot. With commercial optical design software, there can be optimization functions to optimize for spot size, wavefront error or the modulation transfer function

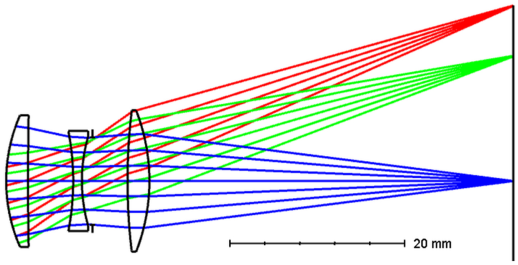

Fig. 1. Schematic view of the Cooke triplet lens design, from [22].

(MTF). These functions can handle automatic generation of individual operands and assignment of the operand weights.

In optimization and optical analyses, there should be sufficient rays to sample the aperture, the field of view and the spectrum (*e.g.* visible spectrum) to achieve an accurate model of the optical system. Ray failure is a common issue, especially with wide field of view or fast (*i.e.* low *f/number*) lenses. In this case, a particular ray in the selected ray bundle does not reach the image plane. This could be due to either the ray falling outside a lens' clear aperture or total internal refraction at a lens surface. In the case of a ray failure, the merit function for the lens is not defined, making the optimization procedure more complicated [21].

3. Lens design optimization problem

In order to evaluate and compare various general purpose open-source local and global optimization algorithms, a basic lens design problem was constructed. The lens design starting point for the local optimization problem was the Cooke triplet as shown in Fig. 1 (with the prescription listed in Table 1, obtained from [22]). This lens design has a 50 mm effective focal length and is *f/4*, with a corresponding entrance pupil diameter of 12.5 mm.

The design was entered into OpticStudio in sequential mode. The wavelengths for the lens were specified to be the dFC wavelengths (588 nm, 486 nm and 656 nm), covering the visible spectrum. The fields were specified as real image height of 0 mm, 12.5 mm and 20 mm at the image plane, which corresponds to a full field of view of about 44°. The specified wavelengths and fields were weighed equally for optimization and subsequent analyses.

OpticStudio's Optimization Wizard was used to generate an RMS Wavefront Error merit function for optimization. In addition to operands for optical aberrations, lens glass and airspace center thickness values were constrained from 1 mm to 2.5 mm and from 0.5 mm to 7.5 mm as operands of the merit function. Weights for the glass and airspace thickness constraints were changed to 10 (from the default value of 1) in order to penalize optimization results that violate these boundary requirements. The back radius of the third lens was set as an *f/number* solve in OpticStudio to keep the *f/number* at *f/4*, and the thickness parameter for the airspace after the third lens was set as a marginal ray height solve to focus the lens at the paraxial focus. As a reference for future optimization results, OpticStudio calculated a merit function value of 0.1191 for this lens with the generated RMS Wavefront Error merit function.

4. Optimization algorithms

OpticStudio's Python programming interface was utilized to evaluate different open-source optimization algorithms. In this mode, a Python program can change parameters and settings of an OpticStudio lens file, optical simulations and analyses can also be performed. At a given iteration of the Python optimization program, the program generates a set of values for the optimization variables of the lens design. These variable values are passed to OpticStudio, OpticStudio performs ray-tracing on this lens design, calculates the corresponding merit function value and passes the merit function value back to the Python optimization program. Therefore, OpticStudio was only used as a blackbox for ray-tracing and merit function calculation, while optimization was entirely handled by the Python programs. As the interface between the Python program and OpticStudio introduces some computational

Table 1
Cooke triplet lens design optimization starting point for minimizing the RMS wavefront error of the lens. Variable parameters are indicated with (V) and solve parameters are indicated with (S).

Comment	Radius	Thickness	Material
Object	Infinity	Infinity	
First lens	18.609 (V)	2.5 (V)	SK16
	460.967 (V)	5.157 (V)	
Second lens	−29.934 (V)	1 (V)	F6
	18.605 (V)	1.207 (V)	
Aperture stop	Infinity	4.007 (V)	
Third lens	67.594 (V)	2.5 (V)	SK16
	−22.162 (S)	41.605 (S)	

overhead, instead of reporting total CPU time for optimization, "Number of Merit Function Value Calculations" is provided as a metric to evaluate and compare different optimization algorithms.

PyGMO (Python Parallel Global Multiobjective Optimizer) open-source optimization package [14] was used for optimization in order to have a uniform interface and therefore to achieve a fair comparison of different algorithms. PyGMO provides internal optimization algorithms and interfaces to various other optimizers.

PyGMO is a population-based optimization suite. A population consists of individuals, which are potential solutions to an optimization problem. A population is optimized with a selected algorithm until an optimal solution or a certain termination criterion is reached. PyGMO also supports parallel optimization and interactions of the populations through a generalized island model [23]. However, since OpticStudio license only allows for two program sessions to be simultaneously active, parallelization was not implemented.

4.1. Local optimization

Different local optimization algorithms were evaluated for wavefront optimization of the starting Cooke triplet lens design. As the Cooke triplet is well optimized to begin with, local optimization just reoptimizes the design to minimize the RMS wavefront error merit function.

Most of the local optimization algorithms in PyGMO are wrappers to optimizers from the NLOpt nonlinear optimization library [24]. NLOpt includes both derivative-free (blackbox) and gradient-based optimization algorithms. If an algorithm is gradient-based, numerical derivatives were calculated by PyGMO. Detailed descriptions of the algorithms and references are provided on the libraries' websites [14,24].

Even though PyGMO is a population based optimization suite, for local optimization a population with a single individual was used. Initially this single individual was set to the lens starting point, as listed in Table 1. Default optimization parameters as set in PyGMO were used, including criteria for optimization termination.

The lens design has 11 variables; 5 of these are lens radius of curvature, 3 are lens center thickness and the other 3 are airspace thickness parameters, as listed in Table 1. If an optimization algorithm supported boundary values explicitly, the following boundary values were specified; the boundaries for curvature (curvature is defined as 1/(radius of curvature) for a surface) were -0.25 and 0.25 (which equals to radii of either less than -4 mm or greater than 4 mm), the boundaries for lens center and airspace thickness were 1 mm to 2.5 mm and 0.5 mm to 7.5 mm, respectively. Even if an algorithm did not support boundary values, these conditions are imposed implicitly as part of the merit function in OpticStudio.

Local optimization results achieved with different algorithms are summarized in Table 2. Once an optimizer reaches a low merit function value, it continues to perturb the lens trying to make it better. After searching around the minima for a while, if a lower merit function value cannot be found, the optimization is terminated. Therefore, "Number of Merit Function Value Calculations to Reach Optimal Result First" is reported in the table to show how many iterations are needed for the initial arrival at the minima.

As a benchmark, the same lens design was also optimized in OpticStudio with the internal "Local Optimization" function. "Damped Least Squares" (DLS) algorithm arrived at an optimization result of 0.0686 and "Orthogonal Descent" (OD) algorithm arrived at an optimization result of 0.0698 from the initial value of 0.1191. Most of the local optimization algorithms reached the same local optimization result as OpticStudio's Local Optimization with DLS algorithm, except for PRAXIS and T-Newton algorithms. However, total number of merit function value calculations before terminating optimization and merit function value calculations to reach the optimal result for the first time are different. Considering these two factors, Nelder-Mead Simplex [25] and SLSQP [26] algorithms can be stated as the most accurate and efficient optimization algorithms among the algorithms listed in Table 2 for this local optimization problem. Nelder-Mead Simplex [25] algorithm is a derivative-free, nonlinear, local optimization algorithm,

Table 2
Results for local optimization algorithms. Algorithms with an asterisk (*) are gradient-based algorithms. Merit function value calculations for gradient-based algorithms include merit function value calculations for numerical derivatives.

Algorithm	Total number of merit function value calculations	Number of merit function value calculations to reach optimal result first	Optimization result
BOBYQA	26,624	24,611	0.0686
NEWUOA	131,322	117,643	0.0686
PRAXIS	750	375	0.0770
Nelder-Mead Simplex	12,635	9603	0.0686
SBPLX	250,276	104,626	0.0686
MMA (Method of Moving Asymptotes)*	263,905	145,489	0.0686
CCSA*	373,897	164,185	0.0686
SLSQP*	2958	1829	0.0686
Low-storage BFGS*	17,569	15,433	0.0686
T-Newton*	17,137	2089	0.0775
T-Newton w/ Preconditioning and Restart*	221,425	130,777	0.0686

Table 3
Results for global optimization algorithms for a lens design with pre-selected lens materials.

Algorithm	Total number of merit function value calculations	Number of merit function value calculations to reach optimal result first	Optimization result
Differential Evolution	40,500	36,824	0.0686
Particle Swarm Optimization	500,000	375,260	0.0686
Self-Adaptive Differential Evolution	72,000	53,436	0.0686
Self-Adaptive DE (DE1220)	75,000	60,600	0.0686
Simple Evolutionary Algorithm	20,000	19,834	0.2615
Simple Genetic Algorithm	500,000	494,306	0.2060
Simulated Annealing	11,000	9010	0.3603
CMAES	40,000	25,485	0.3921
Monotonic Basin Hopping – SLSQP	14,910	2630	0.0686
Monotonic Basin Hopping – Nelder-Mead Simplex	37,812	14,722	0.0686

whereas SLSQP [26] algorithm is a gradient-based, sequential quadratic programming algorithm for local optimization. SLSQP reaches the optimal result with significantly less merit function value calculations compared to the Nelder-Mead Simplex algorithm.

4.2. Global optimization with pre-selected lens materials

In order to test various global optimization algorithms, the local optimization problem was converted to a global optimization problem. Namely, the starting point lens design was moved away from the local optimization result. As the first step, 200 random lenses were produced by generating a random number for each of the lens variables within the specified bounds. Once again, *f/number* solve (to define the radius of curvature for the second surface of the last element) and paraxial focus were set in OpticStudio. The lens glasses were the same as the ones listed in Table 1.

Ray failure can be a common problem for complex lenses. In this case lenses were randomly produced, therefore ray-tracing was not possible with some of the generated lenses. With ray failure the merit function is not defined and OpticStudio returns a very high merit function value as a warning that the computation is aborted. The merit function for each lens design was calculated and the 25 triplet lens designs with the lowest merit function values were selected. These 25 lens designs were specified as the members of the starting population for the different algorithms. The merit function values for these top 25 lens designs ranged from 0.9738 to 2.0945.

This same initial population containing the 25 lens designs was then optimized with different global optimization algorithms for up to 20,000 generations. The default parameters as set in PyGMO for the optimization algorithms were used, and the optimization results are listed in Table 3.

Monotonic Basin Hopping (MBH) [27] is a global optimization algorithm, however it relies on other algorithms for local optimization. As the best results for local optimization were achieved with derivative-free Nelder-Mead Simplex and gradient-based SLSQP algorithms, these two algorithms were used as local optimizers with the MBH algorithm. MBH algorithm includes random perturbations of local optimization results with the goal of converging to a global minimum.

As presented in Table 3, most global optimization algorithms reached the minimum merit function value of 0.0686, except for Simple Evolutionary, Simple Genetic, Simulated Annealing, and CMAES algorithms.

Among the global optimization algorithms (implemented with the default parameters as given in PyGMO), MBH algorithm with SLSQP as the local optimizer can be considered to be the most efficient algorithm, as the optimal result was first achieved with only 2630 merit function value calculations, and the optimization terminated after 14,910 merit function value calculations. These numbers include merit function value calculations for estimating the first-order derivative.

4.3. Global optimization with optimal lens material selection

In lens design, chromatic aberration correction is achieved with an optimal selection of lens optical materials. As a more general and more difficult lens global optimization problem, the optimal selection of lens materials can be handled by the optimization algorithm. For this comprehensive optimization problem, lens materials were also considered as optimization variables.

Two important constants of a particular optical material are the refractive index (n) and the Abbe number (V). Typically, glass refractive indices lie between 1.5 and 1.9, Abbe numbers roughly range from 20 to 70. However, any random value of refractive index and Abbe number within these bounds might not necessarily correspond to a physical glass. Therefore, considering these two parameters as continuous variables during optimization might result in a lens design that can't be realized. To circumvent this problem, refractive index and Abbe number were modeled as continuous variables during optimization and OpticStudio's built-in glass substitution function was used at each step.

OpticStudio's glass substitution function substitutes the closest optical material to a given refractive index and Abbe number pair from a list of pre-selected materials. As part of global optimization, the optimization algorithm determined values for the refractive index and Abbe number for a lens element, and the closest physical glass to these values was determined by OpticStudio using this substitution function. With the addition of refractive index and Abbe number as variables for each lens element, the total number of

Table 4
Results for global optimization algorithms for a lens design including optimal glass selection. Glasses selected by the algorithms are also listed.

Algorithm	Total number of merit function value calculations	Number of merit function value calculations to reach optimal result first	Optimization result	Selected glasses (refractive index/Abbe number)
OpticStudio Global Optimization with DLS Algorithm	500,000	–	0.0426	N-LAK34 (1.729/54.50), SF10 (1.728/28.41), N-LASF45 (1.801/34.97)
Differential Evolution	345,750	318,602	0.0416	N-LASF40 (1.834/37.30), SF10 (1.728/28.41), N-LASF44 (1.804/46.50)
Particle Swarm Optimization	500,000	417,606	0.0415	N-LASF44 (1.804/46.50), N-SF11 (1.785/25.68), N-LASF40 (1.834/37.30)
Monotonic Basin Hopping – SLSQP*	47,858	5,284	0.1769	N-LASF40 (1.834/37.30), N-SF2 (1.648/33.82), N-LASF40 (1.834/37.30)
Monotonic Basin Hopping – Nelder-Mead Simplex	120,081	29,057	0.0515	N-LAK34 (1.729/54.50), N-SF2 (1.648/33.82), N-LAK34 (1.729/54.50)

optimization variables increased to 17. Boundaries for refractive index and Abbe number were set from 1.5 to 1.9 and from 20 to 70, respectively. In OpticStudio, Schott preferred glasses [28] were selected for glass substitution, resulting in 85 different glass options.

Similar to the previous global optimization problem, initially 200 random lenses were generated. The 25 lenses with the lowest merit function values (ranging between 0.9603 and 1.8214) were specified as the members of the starting population for the different optimization algorithms. This same initial population containing the 25 initial lenses were optimized with different global optimization algorithms for up to 20,000 generations, with the default optimization parameters. The final optimization results and the selected glasses are listed in Table 4. Differential Evolution algorithm reached a merit function value of 0.0416 after 318,602 merit function evaluations, while Particle Swarm Optimization algorithm reached a merit function value of 0.0415 after 417,606 merit function evaluations, with the final lens design shown in Fig. 2. This particular design has a biconvex lens as the first element and a meniscus lens as the last element whereas the initial design shown in Fig. 1 has a meniscus lens as the first element and a biconvex lens as the last element.

As a comparison, the same lens design was also optimized in OpticStudio with “Global Optimization” using the DLS algorithm. The optimization was stopped after 500,000 systems were generated, and the optimal merit function value was 0.0426. The selected glasses for these three optimization results are similar. It is interesting to note that PyGMO's two algorithms arrived at a smaller merit function value compared to OpticStudio in the limited optimization run. However, with longer runs of optimization, OpticStudio can reach lower merit function values.

5. Discussion

In this paper, several general purpose open-source optimization algorithms were evaluated for local and global optimization of a triplet lens design. In particular, for local optimization, gradient-based SLSQP and derivative-free Nelder-Mead Simplex were the top two algorithms in quickly reaching an optimal solution. However, SLSQP algorithm finished optimization with only 2958 merit function value calculations whereas Nelder-Mead algorithm terminated optimization with 12,635 merit function value calculations. Even though estimating numerical derivatives requires additional merit value calculations, the gradient-based nature of the SLSQP algorithm allows for more efficient optimization.

The benefit of gradient-based optimization is also evident in global optimization. Considering the global optimization problem with pre-selected lens glasses, using these two algorithms with added global search through Monotonic Basin Hopping again yielded optimal results efficiently. However, once again the SLSQP algorithm is much more efficient.

Introducing optimal selection of lens glasses breaks the continuous nature of the problem, as physical lens glasses can only take certain refractive index and Abbe number values. For this case, Differential Evolution and Particle Swarm Optimization algorithms gave optimal results. Other algorithms arrived at local optima.

Versions of genetic and simulated annealing algorithms are used in commercial optical design programs. However, with the implementation using PyGMO, they could not reach optimal results in global optimization. In lens design, not only the algorithm, but also the selection of the parameters of a given algorithm is critical. In commercial software, the choice of the optimal parameters is done on the basis of extensive previous experience. In this paper, a first order result is achieved with the default parameters of the optimization algorithms set in PyGMO optimization package. Tuning of the algorithm parameters can lead to more efficient convergence. The results provided in this paper form the basis of selecting an optimization algorithm for further parameter tuning of these open-source algorithms to accomplish fast and accurate optical design optimization.

6. Conclusions and future work

Lens design is a complex and cumbersome optimization procedure to minimize optical aberrations while system level constraints are satisfied. Several commercial software packages are available with integrated algorithms developed and tuned specifically for lens design optimization. On the other hand, there are also general purpose open-source optimization algorithms.

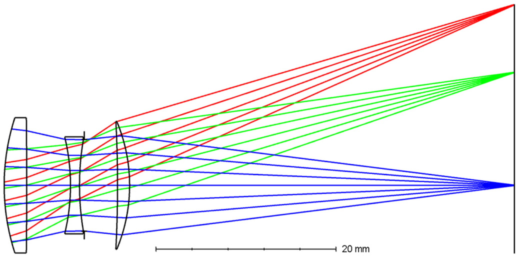

Fig. 2. Schematic view of the final lens design after global optimization with Particle Swarm Optimization algorithm, including optimal lens materials selection.

Several of these general purpose open-source optimization algorithms were evaluated for local and global optimization of a triplet lens design. It was discovered that having a gradient-based algorithm is useful for local optimization and global optimization when variables are continuous. Considering a more general global optimization problem of also selecting optimal lens materials, Differential Evolution and Particle Swarm Optimization algorithms gave optimal results.

Typically, parameters of optimization algorithms need to be configured for a given problem. With tuning of the algorithm parameters, one can achieve more efficient convergence. Once initial parameter tuning is achieved, optimization of more complex lens systems can be demonstrated. More complex lens systems may include lenses with lower *f/number*, larger field of view, greater number of lens elements and elements with aspherical surfaces.

In this paper, OpticStudio's ray-tracing, merit function calculation, and glass substitution routines were used for their reliability and simple interface through the Python programming language. Future work may include developing an open-source, robust ray-tracing package and self programming of the merit function calculation functionality. One example is “Pyrate – Optical Raytracing Based on Python” software by Hartung et al. [29]. With this approach, parallelization of the optimization algorithms can be achieved. Through parallelization, lens optimization can run over multiple processors and on massively parallel cloud computing to achieve optimal results much quicker than currently used methods.

Conflict of interest

None declared.

References

[1] D.P. Feder, Automatic lens design methods, J. OSA A 47 (10) (1957) 902–912.
[2] C. Wynne, Lens designing by electronic digital computer: I, Proc. Phys. Soc. 73 (5) (1959) 777.
[3] R.C. Juergens, The Sample Problem: A Comparative Study of Lens Design Programs and Users, (1980), https://doi.org/10.1117/12.959102.
[4] Inc. Synopsys, Code V Reference Manual, (2015).
[5] L.L.C. Zemax, OpticStudio User Manual, (2017).
[6] D. Dilworth, The SYNOPSYS Lens Design Program, Optical Systems Design Inc., 2009.
[7] F. Bociort, P. van Grol, Finding starting points analytically for optical system optimization, Proc. SPIE-Int. Soc. Opt. Eng. 8486 (2012) 84860B-1–84860B-5.
[8] M. van Turnhout, F. Bociort, Instabilities and fractal basins of attraction in optical system optimization, Optics Express 17 (1) (2009) 314–328, https://doi.org/10.1364/OE.17.000314 http://www.opticsexpress.org/abstract.cfm?URI=oe-17-1-314.
[9] D.C. Dilworth, Automatic lens optimization: recent improvements, 1985 International Lens Design Conference, SPIE, 1986, pp. 191–196.
[10] M. Isshiki, H. Ono, K. Hiraga, J. Ishikawa, S. Nakadate, Lens design: global optimization with escape function, Opt. Rev. 2 (6) (1995) 463–470.
[11] F. Bociort, E. van Driel, A. Serebriakov, Networks of local minima in optical system optimization, Optics Lett. 29 (2) (2004) 189–191.
[12] OSD Inc, SYNOPSYS: The Optimization Challenge, (2018) http://www.osdoptics.com/optimization-challenge.html.
[13] L.M. Rios, N.V. Sahinidis, Derivative-free optimization: a review of algorithms and comparison of software implementations, J. Global Optim. 56 (3) (2013) 1247–1293.
[14] ESA, PyGMO: Python Parallel Global Multiobjective Optimizer, (2017) https://esa.github.io/pagmo2/index.html.
[15] J.P. McGuire Jr., T.G. Kuper, Approaching direct optimization of as-built lens performance, Proc. of SPIE, vol. 8487 (2012) 84870D-1.
[16] F.E. Sahin, Lens design for active alignment of mobile phone cameras, Opt. Eng. 56 (6) (2017) 065102.
[17] F. Heide, M. Rouf, M.B. Hullin, B. Labitzke, W. Heidrich, A. Kolb, High-quality computational imaging through simple lenses, ACM Trans. Graphics (TOG) 32 (5) (2013) 149.
[18] W. Li, X. Yin, Y. Liu, M. Zhang, Computational imaging through chromatic aberration corrected simple lenses, J. Mod. Optics 64 (20) (2017) 2211–2220.
[19] F. Sahin, A. Tanguay, Distortion optimization for wide-angle computational cameras, Optics Express 26 (5) (2018) 5478–5487.
[20] D. Izzo, M. Ruciński, F. Biscani, The generalized island model, Parallel Architectures and Bioinspired Algorithms, Springer, 2012, pp. 151–169.
[21] W. Smith, Modern Lens Design, McGraw-Hill, 2004.
[22] M. Kidger, Intermediate Optical Design, SPIE Opt. Eng. Press, 2004.
[23] F. Biscani, D. Izzo, C. H. Yam, A global optimisation toolbox for massively parallel engineering optimisation, preprint arXiv:1004.3824.
[24] S.G. Johnson, The NLopt Nonlinear-Optimization Package, (2011) http://ab-initio.mit.edu/nlopt.
[25] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (4) (1965) 308–313, https://doi.org/10.1093/comjnl/7.4.308.
[26] D. Kraft, Algorithm 733: Tomp-fortran modules for optimal control calculations, ACM Trans. Math. Softw. (TOMS) 20 (3) (1994) 262–281.
[27] D. Wales, J. Doye, Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms, J. Phys. Chem. A 101 (28) (1997) 5111–5116.
[28] S. Inc, Optical Glass Data Sheets, (2017).
[29] J. Hartung, M. Esslinger, Pyrate – optical raytracing based on python.