Local versus Global Optimization of Electron Lens System Design

Nezhad, Neda Hesam Mahmoudi; Niasar, Mohamad Ghaffarian; Hagen, Cornelis Wouter; Kruit, Pieter

DOI
10.1109/ICOA49421.2020.9094475

Publication date
2020

Document Version
Final published version

Published in
6th International Conference on Optimization and Applications, ICOA 2020 - Proceedings

Local versus Global Optimization of Electron Lens System Design

Neda Hesam Mahmoudi Nezhad(1)* , Mohamad Ghaffarian Niasar(2) , Cornelis Wouter Hagen(1) and Pieter Kruit(1)
(1) Imaging Physics, Charged Particle Optics Group, Delft University of Technology, The Netherlands
(2) Faculty of Electrical Engineering, DC Systems, Energy Conversion and Storage, Delft University of Technology, The Netherlands
*n.hesammahmoudinezhad@tudelft.nl

***Abstract*— In electron optics, the design of electron lens systems is still a challenge. To optimize such systems, the objective function which should be calculated, depends on the electric potential distribution in the space created by the lenses. To obtain the electric potential, the existing methods are generally based on some mathematical techniques which need to mesh the space of the lens system and derive the electric potential at all mesh points. Hence, calculation of the objective function for such systems are computationally expensive. Therefore, applying a fully automatic optimization routine has not yet been feasible, especially for lens systems with many free variables. Hence, the study of objective-function landscape of such problems has not yet been performed.**

One of the questions of interest for optical designers, that has not been studied in the literature, is whether this problem can be solved by a local optimizer or is it necessary to apply a global optimizer. Recently we succeeded in implementing a method (based on a so-called SOEM (Second Order Electrode Method) technique) which calculates the electric potential in a fast and reasonably accurate way. In this paper, that method, is implemented to perform the study of local versus global optimization for electron lens design. The global optimization method here is performed by GA (Genetic Algorithm). The objective function is taken to be the probe size of the electron beams at the image plane.

The results of our study show that the objective function of this problem has many local minima and the optimization of such problems cannot be handled by a local optimizer. GA is shown to perform well by overcoming these multiple-local minima to arrive at a global minima.

Keywords—Electron Lens Design, Global Optimization, Local Optimization, Genetic Algorithms, SOEM (Second Order Electrode Method)

I. INTRODUCTION

Optimization of multi electrode lens systems, is still a challenging task for electron-optical designers. The challenge is due to the fact that the objective function for such problems (i.e. spot size) cannot be analytically formulated. The existing methods that numerically calculate the objective function, are based on methods such as finite element method, boundary element method, or finite difference method that all need to mesh the space of electrode lens systems. Calculation of optical parameters on all these mesh points is then needed to derive the objective function for each system. Using such calculation intensive methods in an optimization routine significantly increases the optimization time. It causes running a fully-automatic optimization routine to be not easily feasible, especially when many free parameters are involved.

Due to the problem mentioned above, to our knowledge, there is not yet a fully automated optimization routine which can handle the optimization of a multi electrode lens system, having all its geometry and voltages as free parameters. Therefore, studying the objective function landscape of such problems, even the question of whether the optimization of multi-electrode lens systems are a global or local optimization problem, have still not been performed and answered in the literature. This topic is of interest to electron optical designers and can be very useful in speeding up the optimization process in electron lens design.

Recently we have introduced and implemented successfully a fast method to calculate the electric potential (hereafter we call it potential), and therefore the objective function, in electron-lens design problems [1] (based on a so-called Second Order Electrode Method (SOEM) method, first proposed by Adriaanse and Barth in 1989 [2-4]). Once we had developed such an automated and fast routine, we decided to use that to perform the above mentioned study on the electron lens design. The main intention of this work is to investigate the objective function landscape of a multi-electrode lens design system, and whether or not it has many local minima, and if this problem can be handled by a local optimizer or not.

The objective function (spot size) calculated by SOEM has some deviation from its precise value [5], which could be derived using Finite Element Method (FEM), however, it is shown in our previous work [1], that the trend of decrement of spot size that resulted from optimization based on SOEM is almost the same as the one calculated accurately based on FEM (using COMSOL Multi-physics software, version 5.3a). The formulation of objective function calculation (derivation of spot size from the potential) in SOEM, mathematically, is also similar to the one calculated by FEM. The main intention of this work was not to run the optimization to accurately ascertain the best optimized system, but simply to study the local versus global optimization, so this can be performed by the fast, approximate SOEM method. Therefore, all optimization for this study is performed based on SOEM. Genetic Algorithm (GA) [6] is used as a global optimizer to make this study.

This optimization problem also has a constraint. In order to apply the constraint two approaches are taken. The outcome of these approaches are also analyzed and illustrated in this paper.

The paper is structured as follows. In section II, Optimization parameters, namely objective function, free variables, bounds and constraints for multi-electrode lens optimization problems, are described. Implementation of local and global optimization in MATLAB (Matrix Laboratory, version r2016 b), as well as constraint implementation are addressed in section III. study of local versus global optimization is described and analyzed in section IV. Section V contains the conclusion of this study.

978-1-7281-6654-4/20/$31.00©2020 European Union

II. Optimization Parameters

A. Objective Function

In electron-optical imaging systems, the main application of electrode lenses is bending and focusing a bunch of electron beams by electric fields for the purpose of image formation. The higher the resolution of the image, the better the electrode-lens system. To achieve an image with higher resolution, a so-called optical parameter of "spot size" (the cross section area of a bunch of electron beams passing through the system, at the image plane), should be minimized. Thereby, this optimization problem is to obtain the system with the smallest spot size at the image side. So, the spot size (at the image side) is taken as the objective function.

The spot size, assuming the lens only suffers from third order axial aberrations such as chromatic and spherical aberrations and that the source image contribution is negligible, can be estimated as [7]:

$$D_s^2 = (0.18\, C_s\, \alpha^3)^2 + (0.6\, C_c\, \alpha\, \frac{\Delta U}{U})^2 \qquad (1)$$

where, D_s stands for spot size, C_s and C_c are spherical and chromatic aberration coefficients, respectively. This estimation is presumed to be valid for our case-study. α is the half opening angle of the beam (chosen to be 10 milliradian in this case-study), ΔU is the energy spread of the electron source (taken as 1eV) and U is the acceleration energy (equal to the potential at the image plane = 1kV), these values are typical for low voltage electron beam systems.

To get the spot size (D_s) at the image side, all parameters mentioned in equation (1) should be calculated at the image side. Here, first C_s and C_c are calculated at the object side, then using the magnification (M), they are converted to their correspondent values at the image side.

C_c and C_s at the object side, can be calculated as a function of axial potential (ø), its first and second derivatives (ø' and ø''), and the imaging principle ray of $r_a(z)$ starting on-axis, from the object side, with a slope of 1 as [8]:

$$C_s = \frac{1}{16ø_0^{1/2}} \int_{z=0}^{L} ø^{1/2} \left\{ \left(\frac{5}{4}\left(\frac{ø''}{ø}\right)^2 + \frac{5}{24}\left(\frac{ø'}{ø}\right)^4\right) r_\alpha^{\,4} + \frac{14}{3}\left(\frac{ø'}{ø}\right)^3 r_\alpha^{\,3} r'_\alpha - \frac{3}{2}\left(\frac{ø'}{ø}\right)^2 r_\alpha^{\,2} r'^{\,2}_\alpha \right\} dz \qquad (2)$$

$$C_c = -ø_0^{1/2} \int_{z=0}^{L} \left(\frac{3ø'^2}{8ø^{\frac{5}{2}}}\right) r_\alpha^{\,2}\, dz \qquad (3)$$

where $ø_0$ refers to the potential at the object side.

The above mentioned principle ray ($r_a(z)$) is derived by ray tracing numerically, using the equation of motion of electrons, in the paraxial approximation [8]:

$$\frac{1+\varepsilon}{1+2\varepsilon} ø(z).r_a''(z) + \frac{1}{2} ø'(z).r_a'(z) + \frac{1}{4} ø''(z).r_a(z) = 0 \qquad (4)$$

The spot size at the image side is calculated by inserting Cs and Cc at the image side into equation 1. From now on, where the spot size is mentioned, it means the spot size at the image side.

Spot size, as shown above, can be derived from the axial potential and its derivatives. For multi electrode lenses, with many free parameters, to reach a satisfactory result, this function should be evaluated for thousands of systems within an optimization routine [1]. As the potential calculation part is the most time-consuming part of the objective function evaluation, a fast method of potential calculation for this study is needed. Therefore, SOEM has been applied for the potential calculation part of this research.

B. Free Variables and Bounds

To explain the optimization parameters such as free variables, bounds and constraints for multi-electrode lens systems more specifically, an example of a system of multi-electrode lenses with 6 lenses is taken as the case-study. This choice was made arbitrarily, simply as an example which has enough complexity as a type of multi-electrode lens system. Changing it to other numbers of electrodes is straightforward. A schematic of this system in 3D is shown in Fig. 1. As the system is rotationally symmetrical, it can be converted and solved as a 2D problem (Fig. 2).

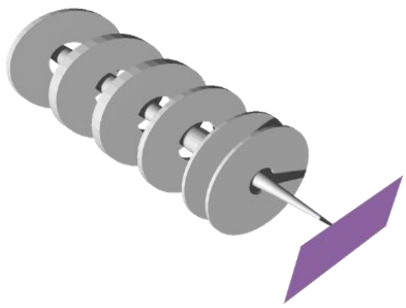

Fig. 1. Schematic of the rotationally symmetrical electrostatic lens system including 6 electrodes in 3D.

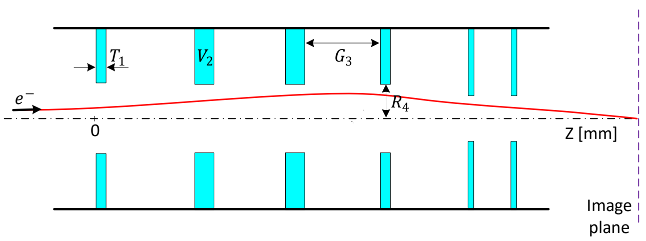

Fig. 2. Schematic of the rotationally symmetrical electrostatic lens system including 6 electrodes in 2D. T_i, R_i, and V_i, correspond to thicknesses, radii and voltages at each electrode, G_i, are gaps between two sequential electrodes.

C. Constraints

The constraint is related to the image position. It is aimed to get the image at a fixed position Xc. For our case-study, this distance is taken to be 3.5 mm from the entrance plane of the first electrode (shown in Fig. 3). A deviation up to a few micrometer (~ 3 μm) is acceptable in our case-study. Therefore the constraint is determined as:

$$3.48\, mm < Xc < 3.52\, mm \qquad (5)$$

III. Implementaion in MATLAB

A. Applying Global and Local Optimizers

Having defined the optimization parameters, a global and local optimizer can now be implemented. Genetic Algorithm (GA) and a so–called "fmincon" local optimizer, in MATLAB, are used as the global and local optimizer, respectively. The choice of using GA as a global optimizer was based on our experience of its successful performance on optical lens system design [9] and recently in electron lens system design [1]. There are different optimizer solvers which can be implemented as the local optimizer in MATLAB. Our optimization problem is a nonlinear problem. In MATLAB, “fmincon” is offered as a suitable optimizer for such a problem. This optimizer does not guarantee to find the global minima, but does guarantee to arrive at the local minima of the objective function-landscape’s valleys, dependent on the point where it is initiated. Therefore, this optimizer is used as a local optimizer for this case-study.

B. Constraint Implementation

Generally, an optimization program can be divided into two major parts. The function which generates systems (here we call it “Optimization function” and in this work this is performed by GA or “fmincon”), and the “objective function calculation” part. An optimization program with constraints includes the third additional part; the “constraint evaluation” function.

The “Optimization function” part generates systems starting with those randomly created in GA, whereas in “fmincom” it starts from an initially given system. Afterwards the process continues in GA by generating the new systems using GA parameters such as crossover, mutation, population, generation, etc. and ends based on its stopping criteria (that can be determined as the maximum number of generation, execution time, value of the objective function). In “fmincon” this process is performed based on the objective functions` gradient, controlled by several limiting parameters such as iterations, tolerance, etc. The boundaries of the problem (maximum and minimum ranges of the geometry and voltages of the systems) should also be determined in this part.

Based on our previous experience of using GA in optimization of such systems having constraints [1], also considering the fact that the intention of this investigation was not to get the best optimized system, but to achieve some results out of GA which can be used as comparison between a global and a local optimizer, the GA parameters are chosen to be: population = 50 and maximum generation = 100. Crossover and Mutation are selected to be 'crossoverarithmetic' and 'mutationadaptfeasible', respectively. Elitism is taken to be 1.

In a standard way, for the optimization routine with constraints, the generated systems are then imported to the second part, i.e. “constraint evaluation”, to check whether or not the constraints are satisfied. The third part, the “objective function calculation”, calculates and evaluates the objective function (spot size) for the systems which had already satisfied the constraints.

However, due to the nature of gradient-based “fmincon” and the possibility of having a non-smooth objective function, in order to ensure the constraints be well-satisfied, the constraints are applied within the objective evaluation part and not by implementing the constraint function as a separate function in MATLAB. To have both GA and “fmincon” run in a similar fashion, for the sake of having fair comparison, this strategy is applied to both. This approach is presented below, and schematically shown in Fig. 3.

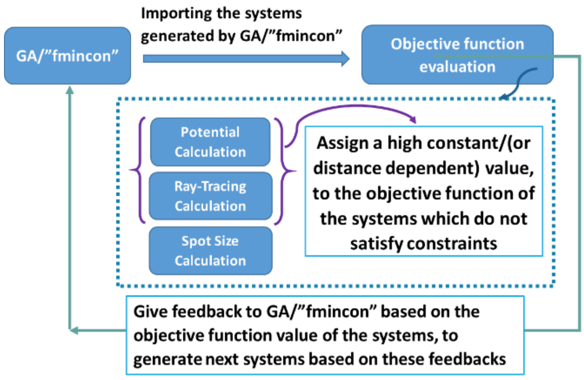

Fig. 3. Schematic of the optimization routine, when the constraints are brought inside the body of objective function evaluation code.

After the first part (“optimization function”) generates the system/systems, they are evaluated for the constraint. If the constraint is not satisfied, a high constant value is assigned to the objective function, and the function ends, otherwise, it continues to calculate the real image plane position. Therefore, the systems which do not satisfy the constraint automatically would have a very small chance of being chosen as the parents to bring to the next generations for breeding the offspring in GA, or to be among the next selected systems in “fmincon”.

To choose the high constant value, it should be noted that this value should be high enough that it exceeds the real feasible values for the systems which are within the constraints. Otherwise, these systems could enter the selection process by their non-real low objective function values.

As the spot size for the systems in which their image plane is within the ranges of $3.48\ mm \leq X_c \leq\ 3.52$ mm, mostly is below 40 nm, taking a value of 100 can be a proper value. Therefore, this approach (here called approach A) is applied by assigning a constant value of 100 as the objective function value for systems which are not within the constraints:

Approach A:

If $3.48\ mm \leq X_c \leq\ 3.52$ mm : Spot size = its real value

If $X_c >\ 3.52$ mm : $\text{Spot size} = 100$

If $X_c < 3.48\ mm$: $\text{Spot size} = 100$

However, by this approach, all systems which are not within the ranges of the constraint, no matter how far they are out of the range, get a similar value of 100. It sounds more efficient to give a high value that depends on the distance of them from the border of the constraint. The spot size based on this approach, considering the value of maximum feasible spot size (i.e. = 40), is formulated as:

Approach B:

$3.48\ mm \leq X_c \leq\ 3.52$ mm : Spot size = real value

If $X_c >\ 3.52$ mm : $\text{Spot size} = 50 * (Xc - 3.52 + 1)$

If $X_c < 3.48\ mm$: $\text{Spot size} = 50 * (3.48 - Xc + 1)$

Choosing this format of formulation and taking value of 1 to the offset, and 50 for multiplication factor was arbitrary, simply to ensure that it takes a value above 40. Taking other values may speed up the optimization further and can be played with. However, for now this format and value, was working properly and efficient enough for this study.

To take the more efficient constraint implementation approach, these two methods (i.e. approach A and B) are tested and compared. As this comparison is performed only to select the better constraint implementation approach for the rest of the study, therefore, it was sufficient to apply only one of the optimization methods. GA is implemented for this search. Since GA, as a global optimizer, is assumed to cover systems having more variety with larger differences in their objective function values, it is better for such an evaluation and data analysis. The results of this comparison are discussed in the following section.

C. Comparison of two approaches on constraint implementation

As GA is a semi-heuristic search optimization technique, the optimizations have been run for each approach (approach A and B) 10 times to provide statistically more reliable data. Each time GA is run for 5000 systems (Population=50 and Generation=100). As representative of the 10 runs, the plots for three runs are presented in Fig 4. The first row (a, b and c) is related to approach A, the second row (d, e and f) corresponds to approach B.

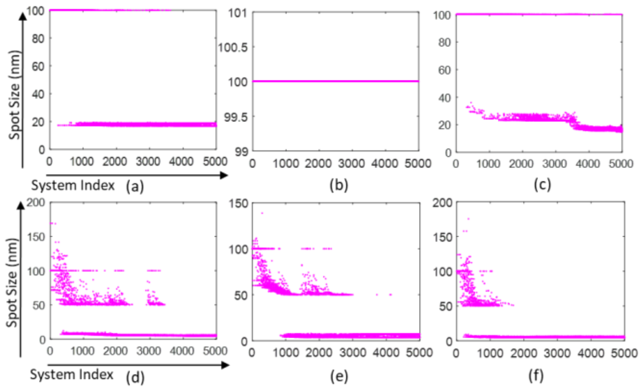

Fig. 4. Three runs related to implementation of two different approaches A and B (the first row (i.e. (a), (b) and (c)) are results of approach A, the second row (i.e. (d), (e) and (f)) are related to approach B).

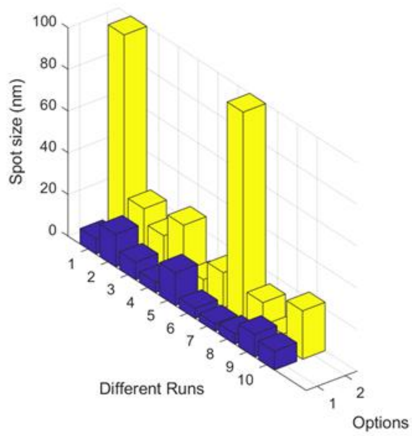

Fig. 5. Result comparison from implementation of approach A and B (shown by options 2 and 1 in the figure, respectively). The plot represents the data related to the minimum value of the objective function which is found after 5000 system iterations for 10 different runs.

As can be seen at a glance from both Fig. 4 and Fig. 5, approach B could get better results. The minimum objective functions reached by approach B after 5000 iterations (shown by the blue columns in Fig 5) has in general, lower values compared to the ones from approach A (yellow columns). For some runs, GA could not even find a system which satisfies the constraint, within evaluation of 5000 systems (Fig. 4(b) or first and 7th column in Fig 5).

Hence, in total, it is clear from the figure, that approach B could, on average, find a much better system. It can be concluded that, using a value depending on the distance from the border of the constraints (approach B) helps the optimization to get better results. Therefore approach B is taken as the constraint implementation approach for our problem.

IV. LOCAL VERSUS GLOBAL OPTIMIZATION

By selecting the constraint implementation method, now our study of global versus local optimization can be performed. To conduct this study, the following steps are taken. First, the optimization routine using the GA for the selected specific number of generations and populations is executed. The constraint on image plane, is applied using approach B. Some points which were intermediate systems found by GA are taken and fed to "fmincon" as the initial points to start with. "fmincon" is executed from each point until it converges to a solution. This is shown schematically in Fig. 6.

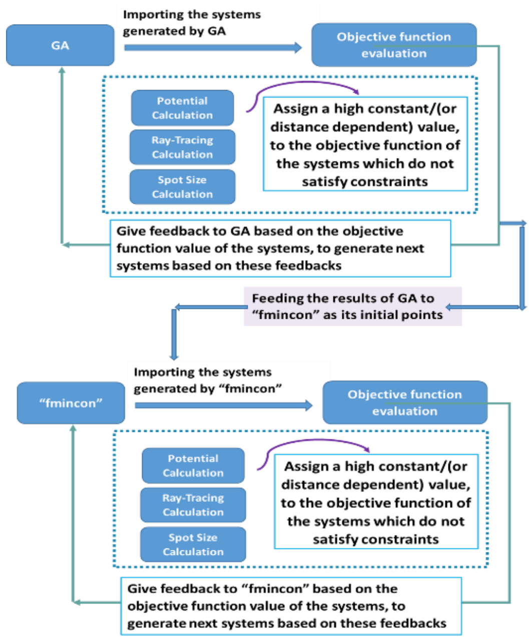

Fig.6. Schematic of optimization implementation for combined "fmincon" and GA.

The total number of systems to be evaluated with GA, is taken to be 5000 (population=50, generation=100).

To analyse, the data from one of the GA runs (Fig. 4(d)) out of 10 previous runs is taken as a sample. Fig. 7 presents this sample. The top plot, includes the objective function data for the systems which are within the constraints, together with those which have not satisfied the constraints (the one in which their values are above 40 nm). The bottom plot shows the data only for those which satisfied the constraint, by omitting the systems which were not within constraint.

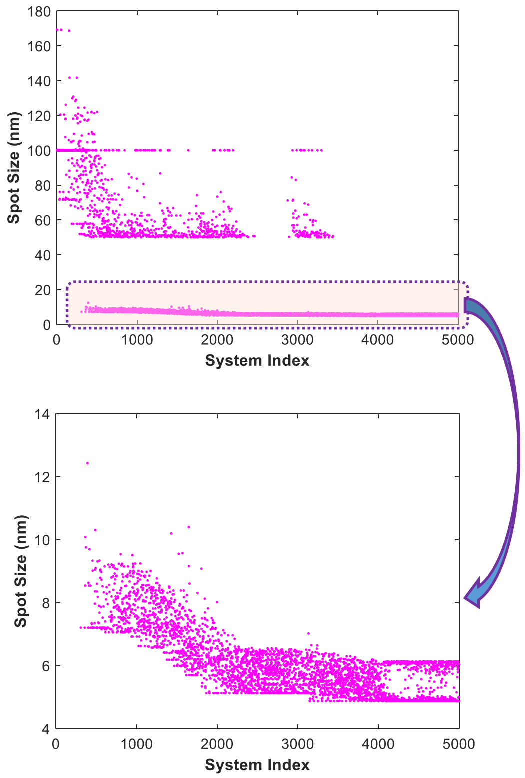

Fig. 7. A run using GA with 5000 evaluated systems (top): spot size versus iterations, for all systems including the ones which have not satisfied the constraints, (bottom): spot size versus iterations only for the systems which had satisfied the constraints.

Four different intermediate points are taken from this run (points A, B, C and D, shown in Fig. 8), and fed to "fmincon" as its initial system to start with.

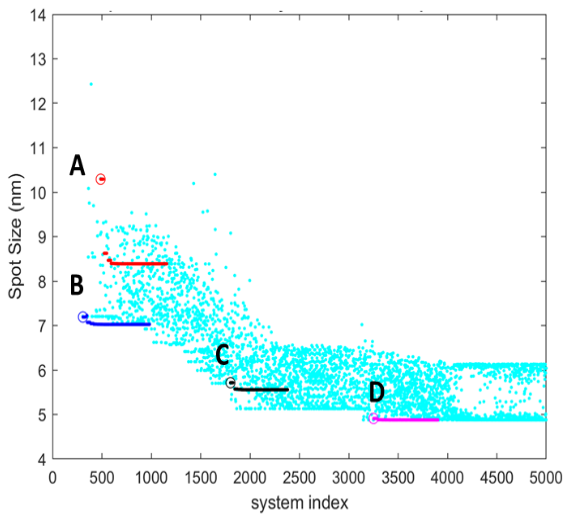

Fig. 8. Spot size versus iterations, for "fmincon" and GA.

If the optimization problem is a global one, it is expected that "fmincon" can arrive at the same solution as GA. However, as can be seen, each system could be improved only slightly by "fmincon"and could not arrive at the last point which had been found by GA (Fig. 8) (point A: reached from 10.35 nm to 8.39 nm, point B from 7.19 nm to 7.02 nm, point C from 5.72 nm to 5.55 nm, point D from 4.91 nm to 4.87 nm). It shows that this problem is a global optimization problem and not a local one, and there are multiple local minima in the objective function landscape of this problem.

It also illustrates that GA can properly perform this optimization problem, by starting from randomly generated systems having high spot size, overcoming many local minima, reaching to a satisfactory result of a system with very small spot size.

Moreover, looking at the last point tested by "fmincon" (point D in Fig. 8), it is seen that this point could also not be improved further by the "fmincon" than the point where GA converged. This shows that GA could act by itself as a good local optimizer and bring the point from different valleys of objective functions to their local point.

V. Conclusion

In this work, an investigation on local versus global optimization has been performed to find out whether the objective function landscape of electron lens system optimization is either a global or a local problem. The results show that the search space of this problem has multiple local minima . For example, starting from a probe size of 10.35 nm, local optimizer reaches a probe size of 8.39 nm The global optimizer, starting from a probe size of 10.35 nm, reaches a probe size of 4.87 nm. A local optimizer, therefore, is not sufficient to find a satisfactory result of such problems and a global optimizer should be used instead.

It is also shown that a Genetic Algorithm acts as a powerful global optimizer, which could handle this complex multi-dimensional objective function problem having many local minima. However, a local optimizer might be used in addition to GA, to speed up the process of finding the global point. Our next plan is to study this possibility and to determine the best possible combination of these two optimizations.

References

[1] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C. Hagen and P. Kruit , "Multi-electrode lens optimization using genetic algorithms", Int. Journal of Modern Physics A, Vol. 34 , 1942020 (16 p.), 2019.

[2] J.P. Adriaanse, H.W.G. van der Steen, and J.E. Barth, "Practical optimization of electrostatic lenses", J.Vac. Sci. Technol. B7, pp. 651-666, 1989.

[3] M.A.J. Van der Stam, J.E. Barth, P. Kruit., "Design of a Multi-Mode Transport Lens with Optimization Program SOEM", SPIE Vol. 2014 Charged Particle Optics, pp.45-56, 1993.

[4] J.E. Barth, H.W.G. van der Steen and J. Chmelik, "Improvement to the electrostatic lens optimization method SOEM", proceeding of SPIE vol. 2522, pp. 128-137, 1995.

[5] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, T. Janssen, C. Hagen and P. Kruit, Optimization of electrostatic lens systems using genetic algorithms, in Recent Trends in Charged Particle Optics and Surface Physics Instrumentation , pp. 24–26, 2018.

[6] Mitchell Melanie, An Introduction to Genetic Algorithms, The MIT Press, Cambridge, London, 1999.

[7] J.E. Barth, P. Kruit, "Addition of different contributions to the charged particle probe size", Optik101, pp. 101-109, 1996 .

[8] Miklos Szilagyi, Electron and Ion Optics, Plenum Press, New York, 1988.

[9] N. Hesam Mahmoudi Nezhad, Optical system optimization using genetic algorithms, Master's thesis, Delft University of Technology, 2014.