Communications to SIMAI Congress,
ISSN 1827-9015, Vol. 2 (2007)

DOI: 10.1685/CSC06069

EVOLUTIONARY ALGORITHMS FOR DOPING PROFILE OPTIMIZATION IN SEMICONDUCTOR DESIGN

V. Di Stefano

Department of Mathematics,
University of Messina, Italy
E-mail: distefan@dipmat.unime.it

C.R. Drago and C.L.R. Milazzo*

Department of Mathematics and Computer Science,
University of Catania, Italy
E-mail: drago@dmi.unict.it
**E-mail: cmilazzo@dmi.unict.it*

Abstract.

This article, which is a preliminary work, deals with the optimal design of a semiconductor device. The objective is to slightly change the doping profile in order to get a gain in the outflow current. Three different black box algorithms are compared with respect to the computational cost and the quality of the solution. The CRS algorithm seems to be the most competitive allowing to reach the optimum with a low computational cost.

Keywords: Drift Diffusion; Optimization; Doping Profile; Genetic Algorithms

1. Introduction

Contemporary manufacturing is undergoing a rapid evolution. Owing to the sweeping changes in science and technology, research in mathematical and computational sciences plays an important role in creating and improving technologies and management practices of the manufacturing system.

The design of microelectronic devices becomes ever more demanding due to various factors such as substantial increases in frequency and bandwidth and growing requirements regarding functionality and reliability, low power consumption and small size. As a consequence, the design of a microelectronic devices requires advanced models, and the optimization of their performances through robust optimization algorithms.

Along with the increase in complexity of the models, often is not possible to build an analytical representation of the function, and sometimes the derivatives of the function cannot be easily calculated; for this reason it is necessary the use of methods which do not make use of the function gradient. A class of this methods are the so called "black box methods", in which the optimization is performed by generating a minimizing sequence independently of the way the calculations are organized.

In this work we intend to optimize the doping profile of a semiconductor device in order to obtain a gain in the outflow electric current. Due to the difficulty of calculating the derivatives of the function, we decided to test and compare different black box algorithms such us Real coded Genetic Algorithm (RGA), Control Random Search (CRS) and DIviding RECTangles (DIRECT).

For the device simulations we use the Drift Diffusion model, which is today the mostly widely used since it allows an accurate description of the underlying physics in combination with low computational costs. There exists a large amount of literature on this model, which covers questions of the mathematical analysis as well as of the numerical discretization and simulation.[7,11]

The paper is organized as follows. In section 2 is shown the Drift Diffusion Model for semiconductor devices. Section 3 contains the formulation of the optimization problem. In sections 4, 5, 6 are given an overview of the RGA, CRS and DIRECT algorithms respectively. In section 6 are shown the numerical results.

2. The Drift Diffusion Model

The stationary standard Drift Diffusion model for semiconductor devices enclosed in a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, is given by the following balance equations for the electron density n and hole density p, coupled to the Poisson equation for the electrostatic potential V:

$$\begin{aligned} J_n &= q(D_n \nabla n - \mu_n n \nabla V) \\ J_p &= -q(D_p \nabla p + \mu_p p \nabla V) \\ \mathrm{div} J_n &= qR(n,p) \\ \mathrm{div} J_p &= -qR(n,p) \\ \epsilon \triangle V &= q(n - p - C) \end{aligned} \tag{1}$$

where C denotes the doping concentration, J_n and J_p the current densities of electrons and holes, respectively. The parameters D_n, D_p, μ_n, μ_p denote the diffusion coefficients and the mobilities of electrons and holes respectively. The physical constants are the elementary charge q and the materials permittivity constant ϵ. The total current density is given by $J = J_n + J_p$.

In the model, generation-recombination processes are included via the recombination rate $R : \mathbb{R}^2 \to \mathbb{R}$. Commonly employed is the *Shockley-Read-Hall* term

$$R_{SRH} = \frac{np - n_i^2}{\tau_p(n + n_i) + \tau_n(p + n_i)},$$

where the physical constants are the carrier life times τ_n and τ_p and the intrinsic density n_i.

In the following we will only consider regimes in which we can assume the validity of the Einstein relations:

$$D_n = U_T \mu_n, \qquad D_p = U_T \mu_p.$$

where $U_T = K_B T_L / q$ is the thermal voltage at the temperature T. K_B denotes the Boltzmann constant.

In high field applications the mobilities depend on the electric field $E = -\nabla V$, by the following Caughey-Thomas formula relations:

$$\mu_{n,p} = \mu^0_{n,p} \left[1 + \left(\frac{\mu^0_{n,p}|E|}{v_s} \right)^2 \right]^{-1/2},$$

where μ^0_n, μ^0_p stand for the mobilities of the field-independent scattering models and v_s is the saturation velocity.

To get well posed problem, system (1) has to be supplemented with appropriate boundary conditions.

We assume that the boundary $\partial\Omega$ splits into two disjoint parts Γ_N and Γ_D, where Γ_D models the Ohmic contacts of the device and Γ_N the insulating parts. Let ν denotes the outward normal vector along the boundary. Firstly, assuming charge neutrality and thermal equilibrium at the Ohmic contacts Γ_D, and, secondly, zero current flow and vanishing electric field at the insulating part Γ_N, yields to the following boundary conditions:

$$\begin{aligned} n = n_D, \quad p = p_D, \quad V = V_D \quad &\text{on } \Gamma_D, \\ \nabla n \cdot \nu = \nabla p \cdot \nu = \nabla V \cdot \nu = 0 \quad &\text{on } \Gamma_N, \end{aligned}$$

where $n = n_D$, $p = p_D$, $V = V_D$ are the $H^1(\Omega)$-extension of

$$n_D = \frac{C + \sqrt{C^2 + 4n_i^2}}{2}, \qquad p_D = \frac{-C + \sqrt{C^2 + 4n_i^2}}{2},$$
$$V_D = -U_T log\left(\frac{n_D}{n_i}\right) + U, \qquad \text{on } \Gamma_D.$$

Here, U denotes the applied voltage.

For the sake of a smoother presentation we assume in the following that the device considered is operated near thermal equilibrium. Thus we assume that no generation–recombination effects are present, i.e. $R \equiv 0$, and the mobilities μ_n, μ_p are constant. In order to get dimensionless equations, we perform the following scaling:

$$\begin{gathered} n \to C_m\tilde{n}, \qquad p \to C_m\tilde{p}, \qquad x \to L\tilde{x}, \\ C \to C_m\tilde{C}, \qquad V \to U_T\tilde{V}, \qquad J_{n,p} \to \tfrac{qU_TC_m\mu_0}{L}\tilde{J}_{n,p}, \end{gathered}$$

where L denotes a characteristic device length, C_m the maximal absolute value of the background doping profile and μ_0 a characteristic value for the mobilities.

Introducing the Debye length

$$\lambda^2 = \frac{\epsilon\, U_T}{qC_mL^2},$$

the scaled equations reads:

$$\begin{aligned} \triangle n - \operatorname{div}(n\nabla V) &= 0 \\ \triangle p + \operatorname{div}(p\nabla V) &= 0 \\ \lambda^2 \triangle V &= n - p - C. \end{aligned}$$

The Dirichlet boundary conditions transform to

$$n_D = \frac{C + \sqrt{C^2 + 4\delta^4}}{2}, \qquad p_D = \frac{-C + \sqrt{C^2 + 4\delta^4}}{2},$$
$$V_D = -log\left(\frac{n_D}{\delta^2}\right) + U, \qquad \text{on } \Gamma_D,$$

where $\delta^2 = n_i/C_m$ denotes the scaled intrinsic density.

3. Problem Formulation

Let Γ_0 be a portion of the Ohmic contacts Γ_D at which we can measure the total current density J. At this contact we prescribe a gained current I_g and allow deviations from a reference doping profile $\overline{C}$ to gain this current flow. Especially we intend to minimize the cost functional of the form

$$Q = \left(\int_{\Gamma_0} J ds - I_g \right)^2 + \varepsilon \int_{\Omega} (C - \overline{C})^2 dx, \tag{1}$$

where $\varepsilon > 0$ is a parameter which balances the two contributions of the functional, the one given by the current and the one given by the doping profile. The reference doping profile $\overline{C}$ can be approximated by a Gaussian distribution:

$$\overline{C} = \frac{\alpha}{\sqrt{2\pi}\sigma} exp\left(- \frac{x^2}{2\sigma^2} \right),$$

where σ is the standard deviation and α is the width of the curve.

4. The Real coded Genetic Algorithm

Genetic Algorithms (GAs) are search procedures based on the mechanism of natural genetics and natural selection. They combine the concept of artificial survival of fittest with genetic operators abstracted from nature to form a robust search mechanism.

The application of GAs to Optimization problems is becoming of challenging interest, due to the possibility of obtaining global optimal solutions and of being applicable also to non differentiable (noisy) functions.

Real coded Genetic Algorithm (RGA) differs from most optimization techniques because of its global searching effectuated by one population of solutions rather than from one single solution.

The RGA search starts with the creation of a random initial population of chromosomes, i.e., potential solutions to the problem. Then, these individuals are evaluated in terms of their "fitness" values, i.e., their corresponding objective function values. In this work the fitness function used is the functional (1). The fitness of each chromosome is calculated and the minimum fitness F_{min} and the average fitness F_{avg} will be found.

The offsprings are the new chromosomes obtained from crossover and mutation. Crossover is a structured recombination operation obtained by exchanging genes of two parents. Mutation is the occasional random alteration of genes.

The crossover process randomly selects two parents to exchange genes with a crossover rate P_c. The location of the gene within the chromosome is called "locus". The crossover point is also randomly chosen from the loci. If one (or both) offspring is infeasible, another mate will be chosen again for crossover. In particular in RGA are used a *Single Point Crossover* or *Simulated Binary Crossover*.

The mutation process randomly selects one parent with a mutation rate P_m. If the offspring is infeasible, another parent will be chosen until a feasible solution can be obtained. For mutation in RGA, are used the *Binary Mutation* or the *Polynomial Mutation* to restore lost or unexpected genetic material into a population to prevent the premature convergence of the algorithm to a sub-optimal solutions and to guaranty diversity among solutions. In fact, lack of such diversity would lead to a reduction of the search space spanned by the RGA, and consequently to a degradation of its optimization performance. Now, the 2p chromosomes (p parents and p offspring) are then ranked in ascending order according to their fitness values. "b" individuals with the best fitness are kept as the parents for the next generation. Other individuals in the combined population of size $(2p-b)$ have to compete by adopting the "tournament selection" approach to get selected in the next generation.

The process of generating new trials with the best fitness will be continued until the fitness values are optimized or the maximum generation number is reached.

5. The Controlled Random Search Algorithm

Controlled Random Search (CRS) is a global optimization algorithm similar to a genetic one, which has an advantage of a relatively lower possibility of converging to a local minimum point than the optimization algorithms based upon general gradient.

The process of finding a minimal point using CRS is as follows.

Given a function of n variables, an initial search domain V is defined by specifying limits to each variable. Controlled random search randomly generates a preset number of search points N called the candidate solutions within the search region of V, and consequently begins the search.

The objective function of each search point is obtained, and the location information of each search point and the objective function value are stored in an array A. A new search point P is then selected based upon the set of candidate solutions in each of the next iterations. The objective function value of point P is calculated when the search point P exists within the search range V. The objective function value calculated at point P f_P is compared with the objective function value at point M f_M as the maximum objective function value of the N number of points stored in array A. If $f_P < f_M$, then point M is eliminated from the set of candidate solutions and point P is included into the set. However, if $f_P > f_M$, point P is discarded and a new search point is selected based upon the set of candidate solutions. In this way the current candidate solutions tend to cluster around the minima lower than the current value of f_M as the algorithm proceeds.

The probability of the candidate solutions ultimately converging to the global minimum depends on the value of N, on the complexity of the objective function, and on how the initial candidate solutions are chosen.

The method of determining the new candidate solution to be included in the set in each iteration affects the performance of CRS.

Price[8] suggests the following method when there are n input variables. At each iteration, $n+1$ distinct points $R_1, ..., R_{n+1}$ are randomly chosen from the current candidate solutions, and these constitute a simplex in n-space. The point R_{n+1} is arbitrarily taken as the pole (designated vertex) of the simplex, and the next candidate solution point is calculated with the following equation.

$$\mathbf{P} = 2\mathbf{G} - \mathbf{R}_{n+1}.$$

Where, $\mathbf{P}$, $\mathbf{G}$ and $\mathbf{R}$ represent the position vectors of the corresponding points.

Price[9] modified the algorithm to speed up convergence without significantly reducing the global search capability. In the modified algorithm, R_1 is always the point L, which has the least function value, and n points are randomly chosen from $N - 1$ points. Hence, L can never be the pole of the simplex.

6. The DIviding RECTangles Algorithm

The DIviding RECTangles (DIRECT) optimization algorithm was created in order to solve difficult global optimization problems with bound constraints and real-valued objective function.
DIRECT is a sample algorithm. That is, it requires no knowledge of the objective function gradient. Instead, the algorithm samples points in the domain, and uses the information it has obtained to decide where to search next. A global search algorithm like DIRECT can be very useful when the objective function is a "black box" function or a simulation. The DIRECT algorithm will globally converge to the minimal value of the objective function. DIRECT was designed to overcome some of the problems that Lipschitzian Optimization encounters. DIRECT begins the optimization by transforming the domain of the problem into the unit hyper-cube. That is,

$$\Omega = \{x \in R^N : 0 \leq x_i \leq 1\}.$$

The algorithm works in this normalized space, referring to the original space only when making function calls. The center of this space is c_1, and we begin by finding $f(c_1)$. The next step is to divide this hyper-cube. We do this by evaluating the function at the points $c_1 \pm \delta \mathbf{e_i}$, where δ is one-third the side-length of the hyper-cube, and $\mathbf{e_i}$ is the ith unit vector. The DIRECT algorithm chooses to leave the best function values in the largest space; therefore we define:

$$w_i = min(f(c_1 + \delta e_i), f(c_1 + \delta e_i)), \quad 1 \leq \ i \leq \ N,$$

and divide the dimension with the smallest w_i into thirds, so that $c_1 \pm \delta \mathbf{e_i}$ are the centers of the new hyper-rectangles. This pattern is repeated for all dimensions on the "center hyper-rectangle", choosing the next dimension by determining the next smallest w_i. The algorithm now begins its loop of identifying potentially optimal hyper-rectangles, dividing these rectangles appropriately, and sampling at their centers. Once an hyper-rectangle has been identified as potentially optimal, DIRECT divides this hyper-rectangle into smaller hyper-rectangles. This restriction ensures that the rectangles will shrink on every dimension. If the hyper-rectangle is an hyper-cube, then the division will be done along all sides, as was the case with the initial step.

7. Numerical Results

The purpose of this paper is to modify the doping reference profile for which an amplification of the total current by 50% is gained, i.e. we set $I_g = \overline{J} * 1.50$, where $\overline{J}$ is a reference current value.
For the simulations it has been chosen a $P - N$ silicon diode, 1 μm long and the applied bias voltage was $V = 0.259$ Volt.
The reference current value is

$$\overline{J} = 7.0337 \cdot 10^{-4}$$

obtained in correspondence of $\sigma = 0.6$ and $\alpha = 5.0$.
The state system was discretized by a variant of the well known exponentially fitted Scharfetter-Gummel scheme.[11] The computations were performed on a uniform grid of 101 points.

We tested the optimization algorithms using for the parameters σ and α the following ranges:

$$\sigma \in [0.1, 0.7] \quad \alpha \in [1.0, 7.0].$$

To test the algorithms, it has been necessary to carry out many simulations by varying the characteristic parameters of each algorithm.

In order to obtain the best results of the optimization it was necessary for RGA to choose the number of the initial population d, the number of generations gen, which corresponds to the number of iterations of the algorithm, the crossover and the mutation probability p_c and p_m; for CRS it was necessary to fix the number of search points N; for DIRECT we had to choose the maximum number of the iterations *maxiter*, the maximum number of the evaluations of the objective function *maxeval*, the maximum number of the rectangles divisions *maxdeep*.

The following table shows the best results of the optimizations

Algorithm	Opt σ	Opt α	Opt J	Funct	Feval
RGA	0.7	5.14487	$10.4389 \cdot 10^{-4}$	$4.8444 \cdot 10^{-5}$	10100
CRS	0.7889	6.98781	$10.4463 \cdot 10^{-4}$	$4.3104 \cdot 10^{-5}$	314
DIRECT	0.6982	5.10836	$10.4510 \cdot 10^{-4}$	$4.8593 \cdot 10^{-5}$	367

where Opt σ and Opt α are the best value of the parameters, Opt J is the corresponding value of the optimized current, Funct stands for the value of the objective function, and Feval is the total number of the evaluations of the objective function.

These results are obtained using the following characteristic parameters of each algorithm:

- RGA: $d = 100$, $gen = 100$, $p_c = 0.9$, $p_m = 0.4$;
- CRS: $N = 50$;
- DIRECT: *maxiter*=100, *maxeval*=350, *maxdeep*=400.

The following pictures show the points distribution and the evolution of the objective function.

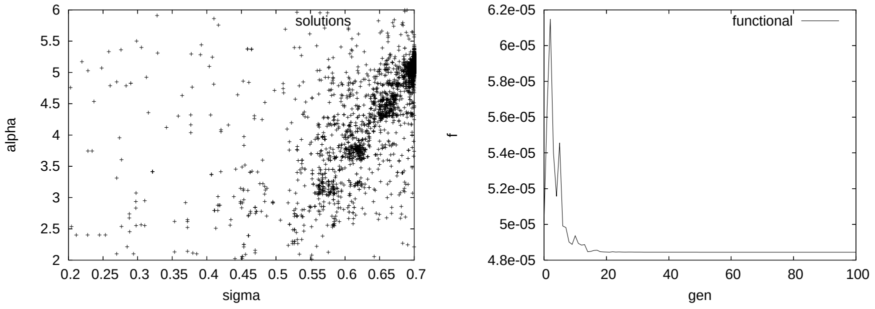

Fig. 1. Points distribution and evolution of the objective function referred to the number of the geration, obtained using the RGA algorithm.

Conclusions

As can be seen in the previous table all the algorithms are able to obtain a global optimal solution, in correspondence of which the value of the objective function is very

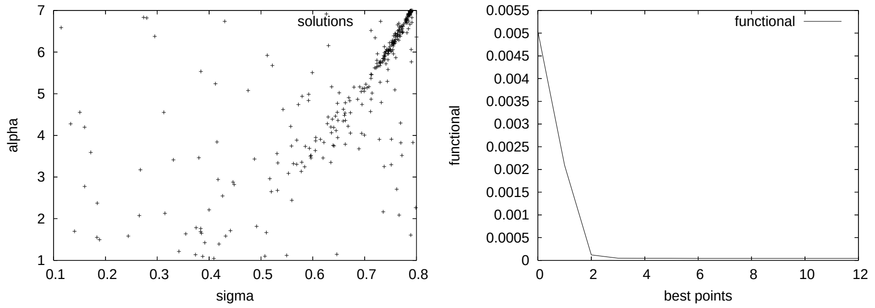

Fig. 2. Points distribution and evolution of the objective function referred to the number of the research point, obtained using the CRS algorithm.

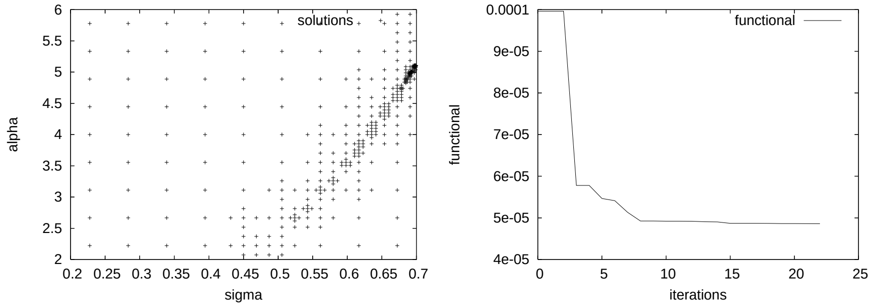

Fig. 3. Points distribution and evolution of the objective function referred to the number of the iteration, obtained using the DIRECT algorithm.

small. So the main difference in the performance of the algorithms is the number of the objective function evaluations which determines the computational time. In particular, RGA needs 10100 evaluations of the objective function, CRS reaches its optimal solution after 314 evaluations while DIRECT does it after 367 evaluations.

So we can conclude that for this problem Controlled Random Search algorithm seems to be the most competitive search algorithm in terms of quality solution and computational cost.

Future work will focus on the study of more sophisticated devices and on the formulation of the optimization problem by using a combination of Gaussian distributions for the doping profile and in such a way a larger number of control variables.

REFERENCES

1. P. J. Ballester and J. N. Carter, *Genetic and Evolutionary Computation Conference*, in *LNCS*, **2723**, Springer, 706–717 (2003).
2. K. Deb, *A population-based algorithm-generator for real-parameter optimization*, in *Soft Computing-A Fusion of Foundations, Kethodologies and Applications*, **9**, 236–253 (2005).
3. D. E. Finkel and C. T. Kelley, *An Adaptive Restart Implementation of DIRECT*, in *Kluwer Academic Publishers*, (2004).
4. D. R. Jones, C. D. Perttunen and B. E. Stuckman, *Lipschitzian Optimization without the Lipschitz constant*, in *Journal of Optimization Theory and Applications*, 79–157 (October 1993).
5. D. Kim, M. Kang and S.Rhee, *Determination of Optimal Welding Conditions with a*

Controlled Random Search Procedure, in *Welding Journal*, 125–130 (August 2005).

6. W.M. Lin, F.S. Cheng and M.T. Tsay, *Distribution feeder reconfiguration with refined genetic algorithm*, in *IEE Proc. Gener. Transm. Distrib.*, **147(6)**, 349–354 (November 2000).
7. P.A. Markowich, *The stationary semiconductor devices equations, eds.* Springer-Verlag, Wien (1986).
8. W. Price, *A controlled random search procedure for global optimization*, in *The Computer Journal*, **20(4)**, 367–370 (1977)
9. W. Price, *Global optimization by controlled random search*, in *Journal of Optimization Theory and Applications*, **74(3)**, 333–348 (1983)
10. S. Rinaudo, F. Moschella, and A.M. Anile, *Controlled Random Search Parallel Algorithm for Global Optimization with Distributed Processes on Multivendor CPUs*, in *proc "Progress in Industrial Mathematics at ECMI 98", eds. L. Arkeryd, J. Bergh, P. Brenner and R. Petterson, B.G. Teubner, Stuttgart-Leipzig*, (1999).
11. D.L. Scharfetter, and H.K. Gummel, *Large signal analysis of a silicon real diode oscillator*, in *IEEE Trans. Electr. Dev.*, **15**, 64–77 (1969).
12. I. G. Tsoulos, I. E.Lagaris *Genetically Controlled Random Search: a global Method for Continuous Multidimensional functions*, in *Computer Physics Communications* **174**, 152–159, (2006).