PARALLEL TCAD OPTIMIZATION AND PARAMETER EXTRACTION FOR COMPUTATIONALLY EXPENSIVE OBJECTIVE FUNCTIONS

Clemens Heitzinger, Thomas Binder, and Siegfried Selberherr
Institute for Microelectronics
TU Wien
Gußhausstraße 27–29,
Vienna, Austria
E-mail: heitzinger@iue.tuwien.ac.at

KEYWORDS
Electronics, Optimization, Parallel Methods, Computer-aided analysis, Parameter identification.

ABSTRACT

The SIESTA (Simulation Environment for Semiconductor Technology Analysis) framework is an extensible tool for optimization and inverse modeling of semiconductor devices including dynamic load balancing for taking advantage of several, loosely connected workstations. Because of the increasing computational power available today, the use of evolutionary computation optimizers which usually require a large number of evaluations of the objective functions becomes feasible even for problems with computationally very expensive objective functions. After a brief introduction to the SIESTA framework and its capabilities, we compare the performance of its optimizers at a real world parameter extraction problem and find that for certain problems genetic algorithms and simulated annealing perform better than gradient based optimization.

INTRODUCTION

TCAD (Technology Computer Aided Design) is concerned with the simulation of the manufacturing processes of semiconductor devices and the simulation of device performance. The optimization framework SIESTA (Simulation Environment for Semiconductor Technology Analysis) (Heitzinger and Selberherr 2000; Plasun et al. 1998; Strasser 1999) is a general purpose tool for the optimization of computationally expensive objective functions that often arise in TCAD applications. It provides an extension language and currently four optimizers. This section is a brief introduction to its design and capabilities.

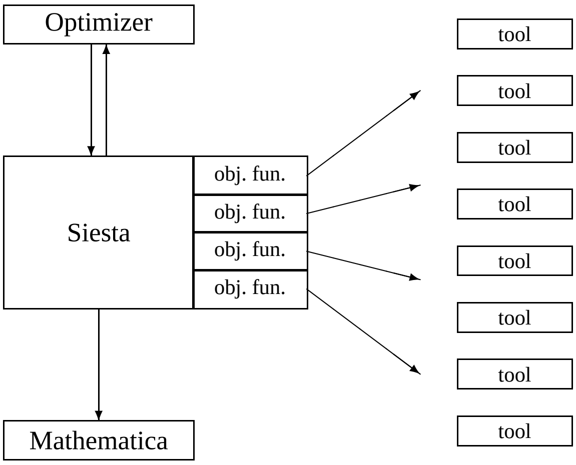

Figure 1: Overview of SIESTA

Figure 1 depicts the control and data flow in a typical optimization run. After setting up an experiment, one of four optimizers is started by SIESTA and its evaluation requests are fulfilled in parallel, where each evaluation entails an arbitrary number of calls of simulation tools on remote machines. In each run the definition of the experiment and the progress of the optimization are saved to files and can be examined from within MATHEMATICA (Wolfram 1991).

SIESTA is written in ANSI Common Lisp using multiprocessing which enables to take advantage of one of the most sophisticated programming languages at the implementation level and at the user level. All language constructs and the SIESTA specific functions are available to the user and can even be used interactively, which facilitates setting up new experiments. The requirements on the software infrastructure installed on the remote machines has been minimized.

In addition to the simulation tools to be used, SIESTA demands to have remote access via `rsh` and `rcp` (or `ssh` and `scp`) and needs a couple of standard UNIX commands.

SIESTA continuously monitors the remote machines and determines available ones and their load averages. Machines with a load average above a certain threshold are not considered for remote tasks until their load average has dropped. Whenever a remote task has to be executed, the machine with the lightest load per CPU is chosen or, if no machine is available, the thread waits until a machine becomes available. Tasks started by SIESTA are accounted for during this process. Several aspects of this algorithm can be influenced by variables and the function for choosing machines can be redefined by the user. Hosts can be disabled temporarily or for certain times every day. Finally all configuration can happen while an optimization is running.

The basic building block for parallel evaluation and load balancing is a macro called `parallel`, which extends the base language transparently and evaluates its arguments in parallel. Another important macro is `with-retries`, which is mostly used for retrying the call of simulation tools which may fail due to software or hardware problems, or because a simulation tool cannot provide a result for certain inputs. This last situation arises more often when applying evolutionary computation optimizers than gradient based ones.

Finally a license management allows users to adjust the number of licenses of commercial tools to be used by the framework while an optimization is running. This means that a certain number of licenses can be reserved for interactive work.

AVAILABLE OPTIMIZERS

In this section, we shortly describe the optimizers currently available in SIESTA, namely two gradient based optimizers (Donopt, Lmmin) (Kelley 1999) and two stochastic global optimizers (Genopt, Siman).

Genopt

The interface to GAlib (Wall 1994), a C++ library for genetic optimization, is called genopt. It provides standard selection, crossover, mutation, scaling, and termination methods (Michalewicz 1996).

For our experiments we mainly use the following setup, because it provides good results in an acceptable amount of computation time. Since all parameters are reals chosen from intervals, we represent them as floating point numbers, and not as binary vectors as favoured in early genetic optimization. We use a mutation operator which adds a random number from a normal distribution, more precisely, $x \in [a, b]$ is changed to $\min(\max(N(x, \sigma), a), b)$, where σ depends on the length of the interval.

As crossover operators we use two point and uniform crossover. Most populations consist of about 40 to 50 individuals. Constraints handling is done using the popular penalty method, i.e., the scores of states which do not fulfill given constraints are increased by prescribed amounts. In SIESTA, the constraints can be defined as arbitrary functions using all the functionality of the framework.

Siman

Simulated annealing (Beasley et al. 1993; Rutenbar 1989) was invented by Kirkpatrick in 1982 and is a modified version of the hill climbing algorithm. Starting from a random point in the search space, a random move is made. If this move yields a better point, it is accepted. If it yields a worse point, it is accepted only with a certain probability $p(t)$ which depends on the time t. The function $p(t)$ is initially close to 1, but gradually reduces towards 0 in analogy to the cooling of a solid. Hence initially any moves are accepted, but as the temperature reduces, the probability of accepting a negative move is lowered. Negative moves are essential sometimes if local extrema are to be escaped, but obviously too many negative moves will simply lead away from the optimum. Versions like fast re-annealing, adaptive annealing and parallel annealing have been developed. In our framework we provide an interface to an external implementation (Ingber 1996).

Donopt

This gradient based optimizer (Plasun 1999; Strasser 1999) minimizes a scalar value and supports equality and inequality constraints. It is based on donlp2 (Spellucci 1995; Spellucci 1996).

Lmmin

The Levenberg-Marquardt algorithm (Marquardt 1963) is an efficient method to solve nonlinear least squares problems, and is therefore well suited for inverse modeling tasks. SIESTA provides an interface to the implementation found in the MINPACK (Moré et al. 1980; Moré et al. 1984) project.

The parameter values are chosen from prescribed intervals. However, arbitrary constraints are not supported by this optimizer. The step size used for the gradient computation and a tolerance value acting as termination criterion can be adjusted.

INVERSE MODELING

Many models in TCAD applications contain free parameters which depend on properties of the device material and have to be calibrated using measurements. Usually vectors of measured values are fit to characteristic curves of the device in question.

It is not obvious which goal function should be used in an inverse modeling experiment where the distance between two vectors (where one is a measurement) is to be minimized, and several functions have been used, e.g. (Cartuyvels et al. 1993; Young et al. 1996).

In the following $m \in \mathbb{R}^n$ is the measured vector, and $s \in \mathbb{R}^n$ is a vector of simulation results. We define the relative error of two vectors as the vector resulting when applying the relative error function elementwise, i.e., $r_k := (s_k - m_k)/m_k$ for $r = \mathrm{RE}(s, m)$ being the vector of relative errors. The quadratic mean M_2 of a vector $x \in \mathbb{R}^n$ is defined as

$$M_2(x) := \sqrt{\frac{\sum_{k=1}^n x_k^2}{n}},$$

and the weighted quadratic mean $M_{2,w}$ with weights w_k is defined as

$$M_{2,w}(x) := \sqrt{\frac{\sum_{k=1}^n w_k x_k^2}{\sum_{k=1}^n w_k}},$$

where $w_k > 0$ for all k. In this regard, the reader is referred to (Hardy 1952) for properties of mean values. Of course, $\|.\|_2$ and M_2 are equivalent norms. However, the quadratic mean is easier to interpret since the number of comparison points does not influence the value.

For p being a vector of parameters to be fit, minimizing

$$f_1(p) := M_2(\mathrm{RE}(s(p), m))$$

or

$$f_2(p) := M_2(\mathrm{RE}(\log s(p), \log m))$$

(here the logarithm is applied elementwise) is a natural and advantageous formulation of the problem of parameter extraction. Variations of $s(p)$ or m over many magnitudes do not have an ill effect as compared to other functions working with absolute errors, and by using the weight factors f_1 and f_2 can be adjusted to individual needs.

INVERSE MODELING OF A STORAGE CELL

The goal of this example is to show that a typical class of optimization problems, namely inverse modeling or parameter extraction problems, can be automatically solved using evolutionary computation optimizers. In this real world problem, we extract six parameters from the drain currents of the select transistor of a storage cell and try to fit two transfer characteristics (two bulk voltages, two times 27 points) in the process.

For the purpose of this optimization, we treat the measurements and simulated vectors as one vector with 54 components and try to minimize $f(p) := M_2(\mathrm{RE}(s(p), m))$. MINIMOS NT (Binder et al. 1998) was used as the device simulator and Table 1 shows the six parameters, their intervals, and their default values. The default values were used as starting values for the gradient based optimizer.

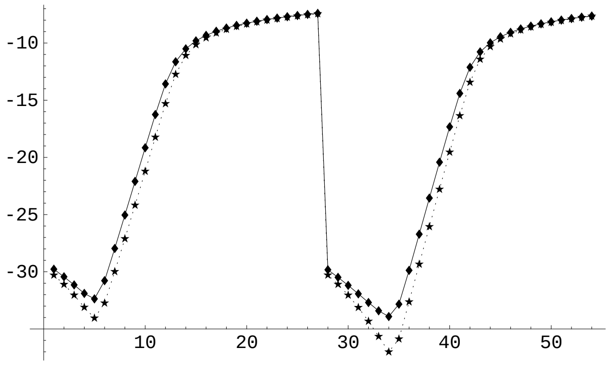

Figure 3: Logarithmic Plot of Best Result by Optimizer Donopt, Score 0.579942.
Solid Line: Measurement;
Dashed Line: Simulation

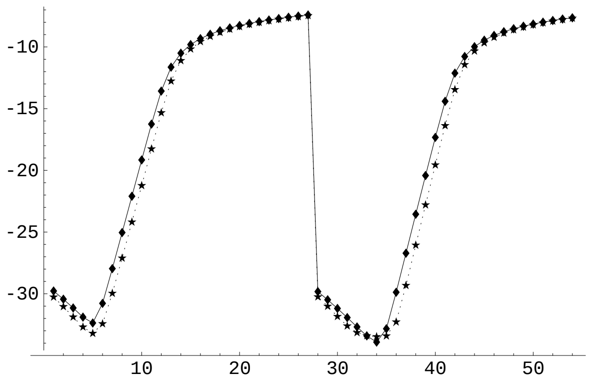

Figure 4: Logarithmic Plot of Overall Best Result after 500 Evalutions, Optimizer Siman, Score 0.517803.
Solid Line: Measurement;
Dashed Line: Simulation

The variable `ew` is the work function of the gate material, and the variable `sr` is the source resistance. The other variables pertain to the Shockley–Read–Hall model (Binder et al. 1998, page 71).

Table 2 shows the three optimizers used (cf. Section) and their configuration which is the default configuration in all three cases.

Figure 2 shows the progress of the three optimizers. For all optimization runs we used a cluster of fifteen workstations

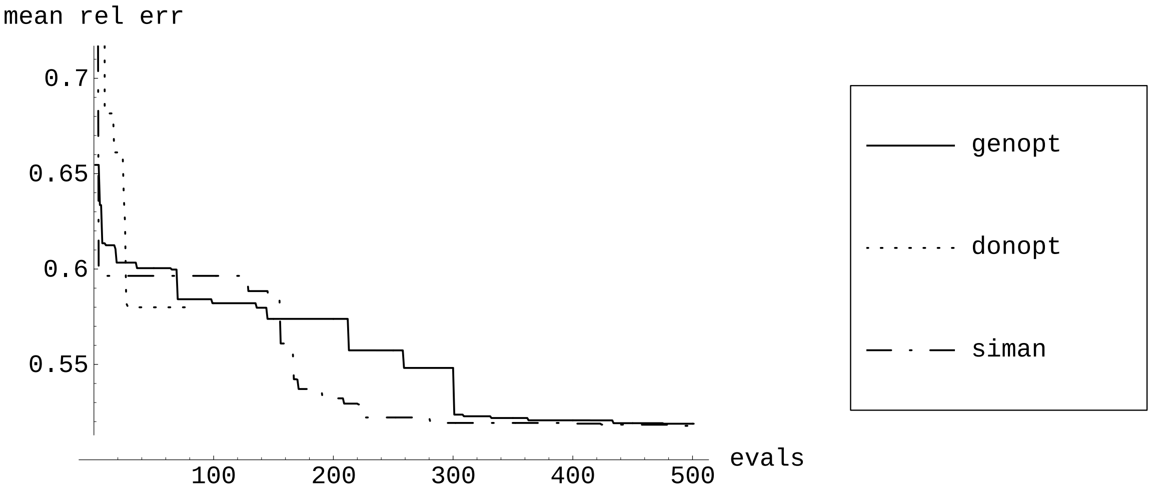

Figure 2: Optimization Progress

Name	Symbol	Interval	Default Value
`et`	E_T	$[0, 2]$	0
`ew`		$[-0.6, -0.4]$	-0.425
`nt`	N_T	$[10^{12}, 10^{16}]$	10^{13}
`sr`		$[100, 500]$	200
`srve`	S_n	$[10, 10000]$	5000
`ste`	$\sigma_{T,n}$	$[10^{-20}, 10^{-17}]$	10^{-19}

Table 1: Parameters

with twenty CPUs and dynamic load balancing. We note that the gradient based optimizer does not yield a good result (cf. also Figure 3), although its initial progress is fast. Furthermore, the evaluations of the genetic algorithm, Genopt, are better parallelized on the twenty CPUs than those of simulated annealing, Siman, and thus in terms of wall clock time elapsed, the genetic algorithm is the fastest optimizer.

Figure 3 shows the best fitting simulated transfer characteristics (for two bulk voltages, left and right half) found by the gradient based optimizer, Donopt, yielding a value of the objective function of 0.58. The agreement in the range of the points numbered 1–10 and 28–38 is mediocre. The next figure, Figure 4, shows the best fitting vector found after at most 500 evaluations with each optimizer. Siman yields good agreement and a value of 0.517803. Both plots are logarithmic and the comparison points are numbered from 1 to 54.

After performing the inverse modeling step the calibrated model was used for minimizing the leakage current of the storage cell and thus improving its performance.

Name of Parameter	Value
Optimizer Genopt:	
algorithm type	steady state
population size	50
probability of replacement	0.7
probability of crossover	0.8
crossover type	two point
probability of mutation	0.2
number of best genomes	50
Optimizer Siman:	
block	15
block max	30
moving average	3
Optimizer Donopt:	
del0	0.5

Table 2: Configuration of the Three Optimizers

CONCLUSION AND OUTLOOK

The real world example shows that global stochastic optimizers can be successfully used on a moderate cluster of workstations even for computationally expensive objective functions where one evaluation takes several minutes. The direction of further work includes specializing evolutionary computation optimizers for TCAD problems and using approximations via generalized Bernstein polynomials for solving optimization problems and design for manufacturability problems.

ACKNOWLEDGEMENTS

The authors acknowledge support from the “Christian Doppler Forschungsgesellschaft”, Vienna, Austria.

REFERENCES

Beasley, D.; D.R. Bull; and R.R. Martin, 1993, ”An Overview of Genetic Algorithms: Part 2, Research Topics”. *University Computing*, 15:170–181.

Binder, T.; K. Dragosits; T. Grasser; R. Klima; M. Knaipp; H. Kosina; R. Mlekus; V. Palankovski; M. Rottinger; G. Schrom; S. Selberherr; and M. Stockinger, 1998, *MINIMOS-NT User’s Guide*. Institut für Mikroelektronik.

Cartuyvels, R.; R. Booth; S. Kubicek; L. Dupas; and K.M. De Meyer, 1993, ”A Powerful TCAD System Including Advanced RSM Techniques for Various Engineering Optimization Problems”. In Selberherr, S.; H. Stippel; and E. Strasser, editors, *Simulation of Semiconductor Devices and Processes*, volume 5, pages 29–32, Wien. Springer.

Hardy, G.; J.E. Littlewood; and G. Pólya, 1952, *Inequalities*. Cambridge University Press, 2nd edition.

Heitzinger, C. and S. Selberherr, 2000, ”An Extensible TCAD Optimization Framework Combining Gradient Based and Genetic Optimizers”. In *Proc. SPIE International Symposium on Microelectronics and Assembly: Design, Modeling, and Simulation in Microelectronics*, pages 279–289, Singapore.

Ingber, L., 1996, ”Adaptive Simulated Annealing (ASA): Lessons Learned”. *J. Control and Cybernetics*, 25(1):33–54.

Kelley, C.T., 1999, *Iterative Methods for Optimization*. SIAM, Philadelphia.

Marquardt, D.W., 1963, ”An algorithm for the estimation of nonlinear parameters”. *Soc. Ind. Appl. Maths. J.*, 11:431–441.

Michalewicz, Z., 1996, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.

Moré, J.J.; B.S. Garbow; and K.E. Hillstrom, 1980, Users Guide for MINPACK-1. Argonne National Laboratory Report ANL-80-74, Argone, IL.

Moré, J.J.; D.C. Sorensen; K.E. Hillstrom; and B.S. Garbow, 1984, *The MINPACK Project*. Sources and Development of Mathematical Software. Prentice-Hall, Englewood Clifs, NJ.

Plasun, R.; M. Stockinger; R. Strasser; and S. Selberherr, August 1998, ”Simulation Based Optimization Environment and its Application to Semiconductor Devices”. In *Intl. Conf. on Applied Modelling and Simulation*, pages 313–316, Honolulu, Hawaii, USA.

Plasun, R., 1999, *Optimization of VLSI Semiconductor Devices*. Dissertation, Technische Universität Wien. http://www.iue.tuwien.ac.at/diss/plasun/diss-new/diss.html.

Rutenbar, R.A., January 1989, ”Simulated Annealing Algorithms: an Overview”. *IEEE Circuits & Devices*, pages 19–26.

Spellucci, P., 1995, donlp2 Users Guide. part of the netlib project.

Spellucci, P., June 1996, Solving General Convex QP Problems via an Exact Quadratic Augmented Lagrangian with Bound Constraints. http://www.mathematik.th-darmstadt.de/ags/ag8/spellucci.

Strasser, R., 1999, *Rigorous TCAD Investigations on Semiconductor Fabrication Technology*. Dissertation, Technische Universität Wien. http://www.iue.tuwien.ac.at/diss/strasser/diss-new/diss.html.

Wall, Matthew, 1994, GAlib: A C++ Genetic Algorithm Library. http://lancet.mit.edu/ga/.

Wolfram, S., 1991, *Mathematica – A System for Doing Mathematics by Computer*. Addison-Wesley, 2nd edition.

Young, R.; F. Morehead; and S. Fischer, 1996, ”Calibrating a Complex Process Simulator for Predicting Device Characteristics”. In *First Intl. Workshop on Statistical Metrology*, Honolulu.

AUTHOR BIOGRAPHY

CLEMENS HEITZINGER was born in Linz, Austria, in 1974. After the compulsory military service he studied Technical Mathematics at the University of Technology in Vienna, Austria, where he received the degree of Diplom-Ingenieur (with honors) in 1999. He joined the Institute for Microelectronics (TU Vienna) in February 2000, where he is currently working for his doctoral degree. From March to May 2001 he held a position as visiting researcher at the Sony Technology Center in Hon-Atsugi (Tokyo, Japan). His scientific interests include symmetry methods for solving partial differential equations and the optimization of semiconductor devices.

THOMAS BINDER was born in Bad Ischl, Austria, in 1969. He studied electrical engineering and computer science at the Technische Universität Wien, where he received the degree of ‘Diplomingenieur’ in December 1996. During his studies he was working on several software projects mainly in the CAD, geodesy and security fields. In March 1997 he joined the Institut für Mikroelektronik, where he is currently working on his doctoral degree. In autumn 1998 he held a visiting research position at Sony, Atsugi, Japan. His scientific interests include data modeling, algorithms, software engineering and semiconductor technology in general.

SIEGFRIED SELBERHERR was born in Klosterneuburg, Austria, in 1955. He received the degree of ‘Diplomingenieur’ in electrical engineering and the doctoral degree in technical sciences from the ‘Technische Universität Wien’ in 1978 and 1981, respectively. Dr. Selberherr has been holding the ‘venia docendi’ on ‘Computer-Aided Design’ since 1984. Since 1988 he has been the head of the ‘Institut für Mikroelektronik’ and since 1999 he has been dean of the ‘Fakultät für Elektrotechnik’. His current topics are modeling and simulation of problems for microelectronics engineering.