Model Parameter Identification with SPICE OPUS: a Comparison of Direct Search and Elitistic Genetic Algorithm

Arpad BŰRMEN [*] *Tadej TUMA* [*]

***Abstract* - SPICE is a widely used tool for simulation of analog circuits. Accurate models of devices are crucial for obtaining realistic results from the simulation. Many devices must still be modelled with subcircuit models in order to achieve sufficient accuracy. Parameter values for subcircuit elements are determined by optimisation. Use of SPICE OPUS optimisation capabilities for model parameter identification is presented. The performance of the simple genetic algorithm when used for model parameter identification is evaluated. The effect of elitism on convergence and results is examined for *9* cases of parameter identification and *2* benchmark optimisation cases. The performance of the simple genetic algorithm is compared to the performance of the simplex algorithm. Finally conclusions and guidelines for future work are presented.**

1 Introduction

SPICE [1] is the de-facto standard for analog circuit simulation. As every other CAD tool SPICE too is subject to the known proverb »garbage in-garbage out«. This means that without accurate models it is impossible to obtain realistic simulation results that can be used in the process of circuit design.

Device models are built into the SPICE simulator core for most of the semiconductor devices. Together with model parameters these device models provide the information about behaviour of a particular part or device in the circuit.

These built-in device models however do not always provide sufficient accuracy for simulation of many parts widely used in electronic circuits. Consider for example, the Zener diode, the power MOS FET or the SCR. There is no built-in device model available for them. The solution to this problem is to model a particular part with a subcircuit that represents the part's behaviour.

Model parameter identification is used to obtain the values of model parameters from the data provided by the part manufacturer in the datasheet. There are two criteria by which the optimisation method used for parameter identification is evaluated:

- the quality of the result (i.e. how well the model and the datasheet agree)
- time required to obtain the result

Although a wide variety of optimisation techniques exist, only a few can be successfully applied to circuit optimisation as shown in [2].

* Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, 1001 Ljubljana, Slovenija
http://www.fe.uni-lj.si/spice
E-mail: Arpad.Buermen@fe.uni-lj.si

This paper deals with the simple genetic algorithm (SGA) [3] and the effect of elitism on convergence of the model parameter identification. Genetic algorithms have successfully been applied to a wide variety of optimisation and identification problems. Their main advantage is the fact that they avoid getting caught in local minima. The main handicap of genetic algorithms is their slow convergence. One way to improve this is to add elitism to the genetic algorithm.

The SGA is implemented as one of the available optimisation methods in SPICE OPUS and some encouraging results on benchmark circuit optimisations were reported [2].

In the following sections the list of some of the existing SPICE parameter identification systems is given. Then the modifications to the SPICE programming language NUTMEG and the optimisation subsystem in SPICE OPUS are described. A short description of benchmarks is given. The results obtained from constrained simplex method, SGA and SGA with elitism are compared. Finally the conclusions and guidelines for future work are presented.

2 State of the Art

Many software packages are available for generating SPICE models. Packages included with commercial SPICE simulators like IsSPICE [4] or PSPICE [5] rely on a few characteristic points from the datasheet and so called »rule of thumb« for determining the parameter values for subcircuit elements. MODPEX [6] from Avant! uses some optimisation techniques for parameter identification but there is no support available. Specific software is available for parameter extraction for various models like the BSIM MOS models [7].

Generally, whenever a new model for some real-world device or phenomenon is devised, an article dealing with model parameter extraction for that specific model is published along with custom tailored software that is capable of extracting parameters from measurement results for that specific model.

In this article we present a unified approach to parameter identification for various real-world device models in SPICE. The approach is based on the SPICE OPUS simulator and its optimisation capabilities. SPICE OPUS is an enhancement of the SPICE3f4 simulator including improved scripting language, optimisation facilities and mixed-mode XSPICE extensions. It was developed by the Group for Computer Aided Circuit Design at Faculty of Electrical Engineering, University of Ljubljana. The Windows95/98/NT, LINUX and SOLARIS release can be downloaded from http://fides.fe.uni-lj.si/spice.

3 Methods

In order to guide the optimisation procedure a cost function must be defined. Since the original Berkeley SPICE scripting language (NUTMEG) doesn't provide sufficient means for describing an arbitrary cost function the following modifications were made:

- Interpolation of vectors from different plots had to be fixed since it was broken in the original Berkeley release of SPICE.
- Cursor functions for performing measurements on simulation results in similar manner as on digital storage oscilloscopes were added.
- Some simple functions like *min(), max(), sum(),* etc. were implemented as NUTMEG internals in order to make vector manipulation easier.
- Handling of vectors was thoroughly revised and fixed.
- Access to model and instance parameters was simplified and is now handled by the let statement in the same manner as other vectors are.
- Since the DC characteristic is one of the most common sources of information in the datasheets DC analysis in SPICE OPUS was improved in order to facilitate logarithmic sweep and arbitrary circuit parameter sweep.

The *'optimize'* command syntax was slightly extended in order to provide following features:

- Logarithmic compression of search space for a parameter was implemented in the *'optimize'* command to aid optimisation convergence for parameters whose explicit constraints span several decades.
- Elitism was added to the SGA.

According to past research dealing with application of various optimisation techniques to simulated circuits [2] only constrained simplex and Hooke-Jeeves search from the family of direct search methods [8] have given satisfactory results in conjunction with a circuit simulator on benchmark circuits. Since Hooke-Jeeves search requires an initial guess it is somewhat less appropriate for parameter identification. Therefore only constrained simplex [9] search is considered in this article and its performance is compared to the one of the SGA without elitism and SGA with elitism.

Since constrained simplex search and SGA both start with a random set of points (initial simplex, initial population) only explicit constraints are sufficient for starting the search.

For the constrained simplex search the number of points in the simplex was set to twice the number of parameters. Optimisation was stopped when the relative simplex size was less that *0.1%*.

For the GA Gray encoding was used to encode parameter values in chromosomes. Population size was twice the number of parameters. Chromosome length was set to *4* bytes per parameter (more than $4 \cdot 10^9$ discrete levels per parameter's range). Probability of mutation was set to *0.02* and crossover probability was set to *0.6*. A fixed number of generations was evolved for each test example.

Elitism was introduced into the population by automatically copying the best individual into the next generation. In order to avoid getting caught in local minima a higher probability of mutation (*0.05*) was used for the SGA with elitism.

The fitness of individuals was based on the cost function in such manner that the fitness of the individual with the lowest cost function value was *5* times greater that the fitness of the individual with the highest cost function value in the population.

4 Results

The results are summarized in Table 1. For each test example the number of cost function evaluations is given along with the lowest cost function value found by the method.

4.1 *Shottlky diode wih guard ring*

SB020 Shottky diode forward characteristic was fitted to the model in Figure 1. Rs is a temperature dependent resistor. Parameters to be identified were *IS*, *N*, *EG* and *XTI* for both diodes and *Rs* and *TC1* for the resistor. *EG* and *XTI* were restricted to a band around typical values of a Si junction diode for Dguard (*EG=0.8-1.6*, *XTI=2.9-3.1*) and of a Shottky diode for D (*EG=0.3-0.8*, *XTI=1.9-2.1*). Explicit constraints on other parameters were set to a wider range. Logarithmic compression was applied to all parameters except *TC1* of the resistor.

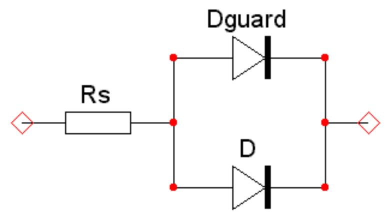

Figure 1: model used for fitting the DC characteristic of a Shottky diode with a guard ring.

The cost function was defined as the root mean square error calculated as the difference between values obtained from the model and desired values. Since there are two curves in the forward characteristic (one for each temperature) the cumulative cost function was defined as the sum of root mean square errors of individual curves.

180 generations were evolved. The quality of the result from the SGA with elitism is slightly worse than the the quality of the result from the constrained simplex method. SGA without elitism delivered significantly worse performace than SGA with elitism.

4.2 *Zener diode breakdown*

Differential resistance in breakdown of various BZX84 Zener diodes was modelled (*2.7V*, *5.6V*, *10V*, *22V*, *36V* and *51V*). The model used is shown in Figure

2. In our examples only two parallel branches were used. Rs_i, V_i and N_i (for diode D_i) in each branch along with Rs were identified. Logarithmic compression was applied to all parameters. Since the differential resistance values are given in log scale the error for the cost function was calculated in the same manner as in previous example except that the the difference between $\log_{10}$ values was used.

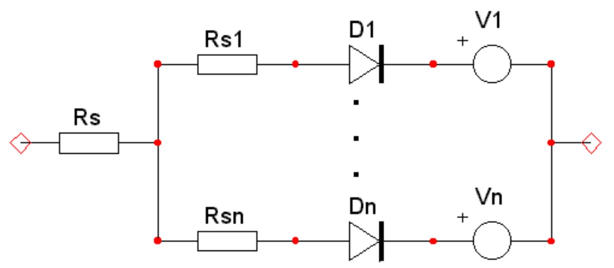

Figure 2: Zener diode differential resistance model.

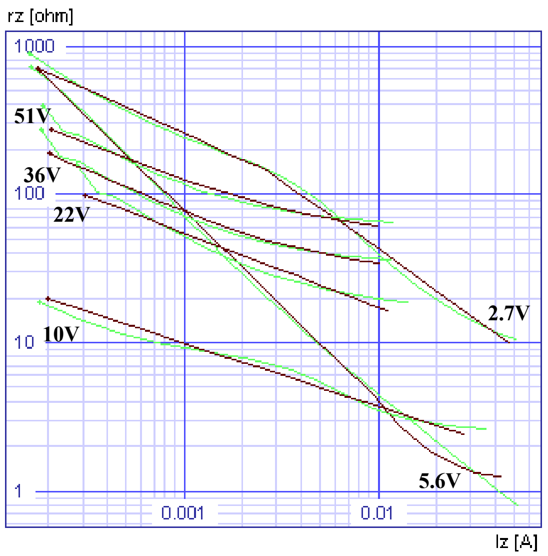

Figure 3: result of model parameter identification for the BZX84 Zener diode family for SGA with elitism. Bright curves represent results obtained from models.

For the SGA test *200* generations were evolved. In all *6* cases the SGA performs worse than constrained simplex. However with the addition of elitism the behaviour of SGA improves. The result from the SGA with elitism is in *2* of the Zener diode fitting cases slightly worse, in *3* cases the same and in one case even better than the result obtained from the constrained simplex method.

4.3 MOS level 3

MOS level *3* model parameters were identified for two MOS transistors (*1µm* and *0.25µm*). The measured data were obtained from [10]. Explicit constraints constraints were the same as in [10].

For all parameters except VTO, ETA, THETA and KAPPA logarithmic compression was used. For the *0.1µm* MOS parameter XJ was fixed to $2 \cdot 10^{-7}$. For the *0.25µm* MOS only the $I_D(V_{DS})$ curves were used although the $I_D(V_{GS})$ curve was also available.

Since measurements comprised multiple $I_D(V_{GS})$ curves the same definition of cumulative cost function was used as for the Shottky diode example.

400 generations were evolved for both examples in the SGA test. The performance of the SGA with elitism is somewhat worse than the performance of the constrained simplex method but still shows significant improvement over SGA without elitism.

4.4 Linearisation of an amplifier

To additionally test the performance of the SGA with elitism two further examples were considered. Both of them are described in [2].

The first one is the linearisation of a transistor amplifier. For this example the population size was increased to *10* (since twice the number of parameters was not sufficient for a population) and *15* generations were evolved.

The quality of the result obtained by SGA is almost identical as the quality of the result obtained by the constrained simplex method. Again elitism significantly improved the convergence of the SGA.

4.5 Triangle to sine converter

All settings are the same as in [2] except that logarithmic compression is added to all parameters.

In a SGA run *250* generations were evolved. This time SGA delivered acceptable performance when compared to constrained simplex method. Elitism slightly improved the result although plain SGA performed well enough.

5 Conclusions

The use of SGA with elitism in SPICE OPUS for model parameter identification was demonstrated on 9 examples. The results show that inferior convergence properties of SGA can be improved by adding a simple modification such as elitism. SGA with elitism exhibited significant improvement over SGA and its performance could be compared to that of the constrained simplex method.

Zener diode reverse characteristic and triangle to sine converter example are essentialy a search for a piecewise linear approximation to desired response. Since in both cases SGA with elitism performed well it seems that SGA is suited for cases where piecewise linear fit to experimental data is sought.

When dealing with cost functions with many local minima (MOS fitting) SGA with elitism finds a solution which is usually a local minimum. This fact suggests the implementation of a mechanism for trading convergence speed for capability of avoiding local minima. This could be achieved by randomly limiting the direct transfer of the best individual to the next generation.

Test Case	No. of parameters	Cost function evaluations			Minimal cost function found		
		Constrained simplex	SGA	SGA+ elitism	Constrained simplex	SGA	SGA+ elitism
Zener 2.7	7	1135	2814	2614	0.0274	0.1201	0.0384
Zener 5.6	7	1681	2814	2614	0.0405	0.1434	0.0495
Zener 10	7	893	2814	2614	0.0809	0.0812	0.0450
Zener 22	7	840	2814	2614	0.0450	0.0452	0.0450
Zener 36	7	892	2814	2614	0.0204	0.0204	0.0204
Zener 51	7	743	2814	2614	0.0245	0.0300	0.0245
Shottky SB020	10	2584	3440	3620	0.0837	0.1597	0.0900
MOS 1um	8	3375	5616	5266	94.55u	636.0u	244.9u
MOS 0.25um	9	2954	6318	5968	83.02u	627.9u	181.0u
Amp. Lin.	2	74	160	145	0.8737m	1.414m	0.8790m
Triangle to sine	12	2829	6024	5774	4.328u	6.712u	4.944u

Table 1: performance comparison for constrained simplex, SGA and SGA with elitism.

SGA in SPICE OPUS could be improved further by implementing adaptive mutation rate control which would increase the mutation rate if no change in cost function was detected for a certain number of generations. The mutation rate would decrease if the cost function was changing rapidly.

The addition of elitism increased the speed of convergence (or, in other words, the result obtained by SGA with elitism for a fixed number of generations is significantly better than the result obtained by plain SGA). The number of cost function evaluations required to obtain a result comparable to the result of constrained simplex search is still *2-3* times greater than with constrained simplex.

SGA with elitism is no substitute for the constrained simplex method which has been from our experience the absolute winner in virtually all optimisation cases involving circuit optimisation based on results from a simulator. One should consider SGA as a preprocessing step for other methods. From the population distribution in parameter space the feasible regions for search could be identified. Then other methods could be employed to find minima in those regions.

References

[1] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, "SPICE3 Version 3f4 User's Manual", University of California, Berkeley, California, 1989.

[2] J. Puhan, T.Tuma, I. Fajfar, "Optimisation Methods in SPICE, a Comparison", Proceedings of the ECCTD'99, vol. 1, 1999, pp. 1279-1282.

[3] David E. Goldberg, "Genetic Algorithms in Search, Optimisation & Machine Learning", Addison-Wesley Publishing, 1989.

[4] "SpiceMod: SPICE Models for Semiconductors", file spicemod.htm at http://www.intusoft.com/, 2001.

[5] "PSpice Model Editor", file pspicedsr9.asp#editor at http://www.pspice.com/product_info/, 2001.

[6] "MODPEX Modeling Tool", file modpex.html at http://www.symmetry.com/, 2001.

[7] J. R. Pierret, "A MOS Parameter Extraction Program for the BSIM Model", Rep. No. ERL-M84/99, Electronics Research Laboratory, University of California, Berkeley, 1984.

[8] R. M. Lewis, V. Torczon, M. W. Trosset, "Direct Search Methods: Then and Now", Journal of Computational and Applied Mathematics, vol. 124, no. 1-2, 2000, pp. 191-207.

[9] M. J. Box, "A new method of constrained optimisation and comparison with other methods", Computer Journal, vol. 7, 1965, pp. 42 – 52.

[10] T. Sakurai, B. Lin, A. R. Newton, "Fast Simulated Diffusion: An Optimization Algorithm for Multiminimum Problems and Its Application to MOSFET Model Parameter Extraction", IEEE Transactions on Computer Aided Design, vol. II, no. 2, 1992, 1992, pp 228-234.

[11] G. Massobrio, P. Antognetti, "Semiconductor Device Modeling with SPICE", McGraw-Hill, 1993.