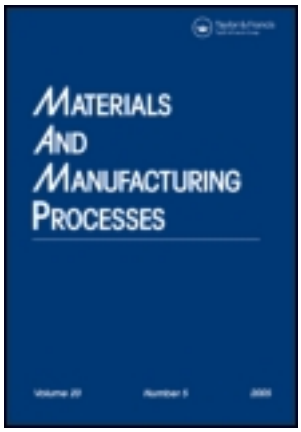


Materials and Manufacturing Processes
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/lmmp20

Hybrid Differential Evolution and Particle Swarm Optimization Approach to Surface-Potential-Based Model Parameter Extraction for Nanoscale MOSFETs

Yiming Li [a b c] & Yu-Hsiang Tseng [b]
[a] Department of Electrical Engineering , National Chiao Tung University , Hsinchu, Taiwan
[b] Institute of Communications Engineering , National Chiao Tung University , Hsinchu, Taiwan
[c] National Nano Device Laboratories , Hsinchu, Taiwan
Published online: 08 Apr 2011.

To cite this article: Yiming Li & Yu-Hsiang Tseng (2011) Hybrid Differential Evolution and Particle Swarm Optimization Approach to Surface-Potential-Based Model Parameter Extraction for Nanoscale MOSFETs, Materials and Manufacturing Processes, 26:3, 388-397, DOI: 10.1080/10426914.2010.526977

To link to this article: http://dx.doi.org/10.1080/10426914.2010.526977

Materials and Manufacturing Processes, 26: 388–397, 2011

ISSN: 1042-6914 print/1532-2475 online
DOI: 10.1080/10426914.2010.526977

Taylor & Francis
Taylor & Francis Group

Hybrid Differential Evolution and Particle Swarm Optimization Approach to Surface-Potential-Based Model Parameter Extraction for Nanoscale MOSFETs

YIMING LI[1,2,3] AND YU-HSIANG TSENG[2]

[1]*Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan*
[2]*Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan*
[3]*National Nano Device Laboratories, Hsinchu, Taiwan*

A set of semiconductor device model and parameters bridges the communities between circuit design and chip fabrication. In this article, we present an intelligent extraction technique for obtaining a set of optimal model parameters of the surface-potential-based PSP model for the sub-45-nm metal-oxide-semiconductor field effect transistors (MOSFETs). The proposed algorithm combines the standard velocity and position update rules in a particle swarm optimization (PSO) algorithm, and the operations of differential mutation and probability crossover from a differential evolution method. This differential approach can increase the diversity of the population and help particles escape from the local optimal solutions. In addition, the adopted fitness function considers not only the error of the $I - V$ curves, but also their first derivatives. Compared with conventional engineering extraction strategy, the hybrid method extracts 14 DC parameters simultaneously for sub-45-nm N-MOSFET devices. The best accuracy and interesting computational efficiency are obtained by several testing cases.

Keywords Differential evolution; Hybrid method; MOSFET; Parameter extraction; Particle swarm optimization; PSP.

INTRODUCTION

Compact models for metal-oxide-semiconductor field effect transistors (MOSFETs) have been indispensable bridges between device fabrication and integrated circuit design over the past two decades [1–3]. Among the state-of-the-art MOSFET compact models [4–9], surface-potential-based models [10–12] are regarded as the advanced ones to contain all relevant physical effects with the aggressive down-scaling of complimentary metal-oxide semiconductor (CMOS) technologies. The PSP model of MOSFETs [11, 12] is one of the most popular surface-potential-based models and has been selected as a new standard for nanoscale CMOS devices [3]. However, the model's parameters should be carefully optimized so that the model can describe the device's electrical characteristics as accurate as possible to measurements. Conventional statistics and local numerical methods, such as curve fitting, regression, and Newton iteration methods [13–16] have been employed for deep-submicron MOSFET parameter extraction. However, these existing methods have several known deficiencies, as follows: (1) poor convergence without good initial guesses; (2) much time is required to seek an optimal solution; (3) difficulty with simultaneous multiobjective optimizations; and (4) limited prediction capability for the sub-45nm MOSFET era. Furthermore, they also require a well-trained device engineer with detailed knowledge of MOSFET models and optimization methods to master each extraction process. Extraction with such local-based methods or tools may increase the product manufacturing cost, prolong the product time to market, complicate the design procedures, and reduce the model accuracy and reliability [17].

The particle swarm optimization (PSO) algorithms [18, 19] were originally devised by Kennedy and Eberhart. It is a cooperative search method inspired by the behavior of birds flocking in search of food. The PSO algorithm features its simple form, easy implementation steps, and fast convergence. PSO has been applied to many real-world problems successfully. However, PSO could fail or become inefficient if the particles get trapped in local minima. One effective method to avoid this premature convergence problem is to increase the diversity of the population. Differential evolution (DE) method [20–22] is also a stochastic population–based optimization method invented by Storn and Price. Its simple structure has drawn much attention. DE employs the differential information to guide its further search, compared with PSO. Methods that combine PSO and DE may pursue fast convergence and high diversity in the population [23–25]. Consequently, computationally robust parameter extraction techniques for searching a set of accurate and reliable parameters of the PSP model [26, 27] are urgent in nowadays nanoscale MOSFETs.

In this article, a hybrid optimization technique [17, 27–29] is successfully implemented for PSP MOSFET compact model parameter extraction. Our approach integrates the operations of differential mutation and probability crossover from DE and the standard velocity and position update rules of PSO. The differential approach taken from DE is used to increase the diversity of the population. Compared

Received July 2, 2010; Accepted September 20, 2010
Address correspondence to Prof. Yiming Li, Department of Electrical Engineering, National Chiao Tung University, 100 Ta-Hsueh Road, Hsinchu 300, Taiwan; E-mail: ymli@faculty.nctu.edu.tw

with a conventional genetic algorithm (GA), DE, and PSO [30–39], the approach exhibits better results in terms of accuracy and efficiency. The error of current-voltage curves and the error of their first derivatives are found to be less than 1% and 2%, respectively. In particular, our hybrid DE-PSO can overcome the difficulty to reduce transconductance (G_m) error. Application of our hybrid DE-PSO to extract parameters of several sub-45nm testing NMOSFETs with different dimension is also successfully obtained in terms of accuracy and computational efficiency.

This article is organized as follows. Section 2 briefs the surface-potential-based PSP model. Section 3 illustrates our parameter extraction configuration and optimization technique. Section 4 shows the results for various tested MOSFETs. Finally, we draw conclusions and suggest future work.

The PSP model for MOSFETs

PSP is the latest and the most advanced compact MOSFET model, compared with the charge-based BSIM model [3]. It was developed by integrating and enhancing the best electrical and physical properties of the two surface-potential-based models SP (developed at the Pennsylvania State University) [10] and MOS Model 11 [7] (developed by Philips Research). The PSP has been selected as a new industry standard for the next generation compact MOSFET model by the Compact Modeling Council [3]. The drain current of PSP model is given below:

$$I_{DS} = BETN \cdot C_{ox} \cdot F_{\Delta L} \cdot \frac{q_{im}^*}{G_{vsat}} \cdot \Delta\psi, \tag{1}$$

where *BETN* is a model parameter, C_{ox} is oxide capacitance, q_{im}^* is the effective inversion charge, and $\Delta\psi$ is the difference between drain end surface potential and source end surface potential. G_{vsat} and $F_{\Delta L}$ will be discussed later. Next, we briefly introduce some essential equations along with model parameters. More detailed introduction and complete model equations can be found in [12]. The bulk potential ϕ_B, which serves as a reference potential, is given by

$$\phi_B = DPHIB + 2\phi_T \ln(NEFF/n_i), \tag{2}$$

where *DPHIB* and *NEFF* are model parameters to be extracted. *NEFF* is related to substrate doping and *DPHIB* modulates bulk potential shift. Drain-induced barrier lowering (DIBL) is a significant effect in nanoscale MOSFETs. The model parameter associated with DIBL is *CF* and it controls the influence of drain bias on potential barrier

$$\Delta V_G = CF \cdot V_{dsx}. \tag{3}$$

Interface states also have effects on the surface potential; in PSP it is given by

$$\psi_{ss} = \phi_T \cdot \left(1 + CT \cdot \frac{T_{KR}}{T_{KD}}\right) \cdot x_s \tag{4}$$

$$\psi_{sd} = \phi_T \cdot \left(1 + CT \cdot \frac{T_{KR}}{T_{KD}}\right) \cdot x_d, \tag{5}$$

where *CT* is a model parameter associated with interface states. The effective drain-source voltage is modeled by the smooth function

$$V_{dse} = \frac{V_{DS}}{[1 + (V_{DS}/V_{dsat})^{AX}]^{1/AX}}. \tag{6}$$

Model parameter *AX* determines the smoothness of the transition. The following are equations related to the G_{vsat} term in drain current model. They combine the mobility model, velocity saturation effect, and series resistance effect. *FETA*, *BETN*, *RS*, *THESAT* are model parameters to be extracted:

$$E_{eff} = E_{eff0} \cdot (q_{bm} + 2 \cdot FETA \cdot q_{im}), \tag{7}$$

$$\rho_s = 2 \cdot BETN \cdot RS \cdot \rho_b \cdot \rho_g \cdot q_{im}, \tag{8}$$

$$G_{mob} = \frac{1 + (\mu_E \cdot E_{eff})^{\theta_\mu} + C_s \cdot \left(\frac{q_{bm}}{q_{im}+q_{bm}}\right)^2 + \rho_s}{\mu_x}, \tag{9}$$

$$\theta_{sat}^* = \frac{THESAT}{G_{mob,s} \cdot G_{\Delta L}}, \tag{10}$$

$$z_{sat} = (\theta_{sat}^* \cdot \Delta\psi)^2, \tag{11}$$

and

$$G_{vsat} = \frac{G_{mob} \cdot G_{\Delta L}}{2} \cdot (1 + \sqrt{1 + 2 \cdot z_{sat}}). \tag{12}$$

The following are equations related to $F_{\Delta L}$ term in the drain current model. They account for channel length modulation (CLM) effect. *ALP*, *ALP1*, *ALP2*, and *VP* are model parameters to be extracted:

$$T_1 = \ln\left(\frac{1 + \frac{V_{DS}-\Delta\psi}{VP}}{1 + \frac{V_{dse}-\Delta\psi}{VP}}\right), \tag{13}$$

$$T_2 = \ln\left(1 + \frac{V_{dsx} - \Delta\psi}{VP}\right), \tag{14}$$

$$\Delta L/L = ALP \cdot T_1, \tag{15}$$

$$\Delta L_1/L = \left[ALP + \frac{ALP1}{q_{im}^*} \cdot R_1\right] \cdot T_1 + ALP2 \cdot q_{bm} \cdot R_2^2 \cdot T_2, \tag{16}$$

$$G_{\Delta L} = \frac{1}{1 + \Delta L/L + (\Delta L/L)^2}, \tag{17}$$

and

$$F_{\Delta L} = [1 + \Delta L_1/L + (\Delta L_1/L)^2] \cdot G_{\Delta L}. \tag{18}$$

TABLE 1.—A set of PSP model parameters to be extracted in this work.

Parameter	Physical meaning	Min	Max
NEFF	Substrate doping	1.0 e + 20	1.0 e + 26
DPHIB	Offset of body potential	−0.3	0.3
CT	Interface states factor	0	5
CF	DIBL Parameter	0	1
BETN	Product of channel aspect ratio and zero-field mobility	1.0 e − 4	10
XCOR	Non-universality parameter	0	1
FETA	Effective field parameter	0	10
RS	Source/drain series resistance	1	1000
THESAT	Velocity saturation parameter	0	15
AX	Linear/saturation transition factor	1	5
ALP	CLM pre-factor	0	3
ALP1	CLM enhancement factor above threshold	0	1
ALP2	CLM enhancement factor below threshold	0	1
VP	CLM logarithmic dependence factor	1.0 e − 10	1

Table 1 shows the PSP model parameters extracted in this work together with their physical meanings, minimal values, and maximal values.

Different physical effects of MOSFETs have different influence on the $I - V$ characteristics of the devices. Since the derivation of PSP model is close to the physical prototype, the model parameters relevant to different physical effects also have different effects on different operation regions of MOSFET. For example, the doping-related parameter would affect the whole regions of $I - V$ characteristics. The parameter related to source/drain series resistance is dominant when MOSFET is operated in low drain bias together with high gate bias. The influence of parameters corresponding to CLM effects emerges when MOSFET is operated under high drain bias.

According to the above observations, we generally would not extract all parameters simultaneously. Instead, we divide the parameter extraction procedure into several steps. In each step, some specific parameters are adjusted to fit limited range of $I - V$ curves using optimization methods. The parameter extraction procedure used in this work is described in Table 2 which is mainly based on the steps [12] with some modifications. First, the magnitude and shape of $I_d - V_{gs}$ curves under low drain bias is roughly formed in the step 1. Next, the subthreshold behaviors of $I_d - V_{gs}$ curves are optimized in the step 2. In the step 3, the mobility-related and series-resistance-related parameters are optimized. In the step 4, parameters corresponding to velocity saturation and channel length modulation are determined. Finally, in the step 5, all parameters are fine tuned to fit the overall $I - V$ curves. If the specified stopping criterion is not reached, parameter extraction procedure repeats.

TABLE 2.—The extraction procedure of PSP model parameters.

Step	Optimized parameters	Fitting target
1	NEFF, DPHIB, BETN, FETA, RS	$I_d - V_{gs}$ curves
2	NEFF, DPHIB, CT	Subthreshold region in $I_d - V_{gs}$ curves
3	BETN, XCOR, FETA, RS	Strong inversion region in $I_d - V_{gs}$ curves
4	THESAT, ALP1, ALP2, VP, AX, CF	$I_d - V_{ds}$ curves
5	All above parameters	Both $I_d - V_{gs}$ and $I_d - V_{ds}$ curves

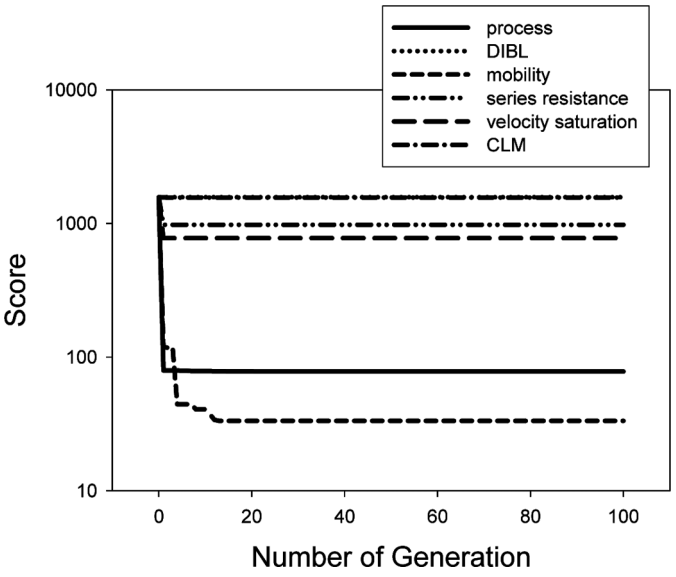


FIGURE 1.—Plot of fitness score versus the number of generation for the sensitivity analysis of six group-wised PSP model parameters.

Figure 1 shows sensitivity analysis of grouped model parameters. The sensitivity analysis can help us understand the degree of effects of different groups of parameters on the $I - V$ characteristics. In our experiment, only one group of parameters is adjusted at a time while other groups of parameter are locked. It can be found that the mobility and process related parameters affect the $I - V$ characteristics the most. DIBL, series resistance, velocity saturation, and CLM have less effect on the $I - V$ characteristics.

THE SYSTEM CONFIGURATION AND EXTRACTION TECHNIQUES

Figure 2 illustrates the architecture of the program implemented in this study. The program can be divided into three layers: the I/O layer, the kernel layer, and the model layer. The I/O layer deals with loading the measurement data, output of extracted parameters file, and output of $I - V$ plot files which are simulated results using the extracted parameters. The kernel layer includes flow control, error calculation, and the function of numerical differentiation for $I - V$ curves. Most importantly, several optimization algorithms are also implemented in the kernel layer. Model layer consists of a set of model equations and model parameter information which contains the name, default value, maximum value, and minimum value of each parameter. In addition, the function of feeding parameter set into model equation is also developed in this layer. Data can be passed from layer to layer if it is necessary. For example, the kernel would pass a trial set of parameters to the model layer, and the corresponding simulation results would be sent back to the kernel during the optimization process.

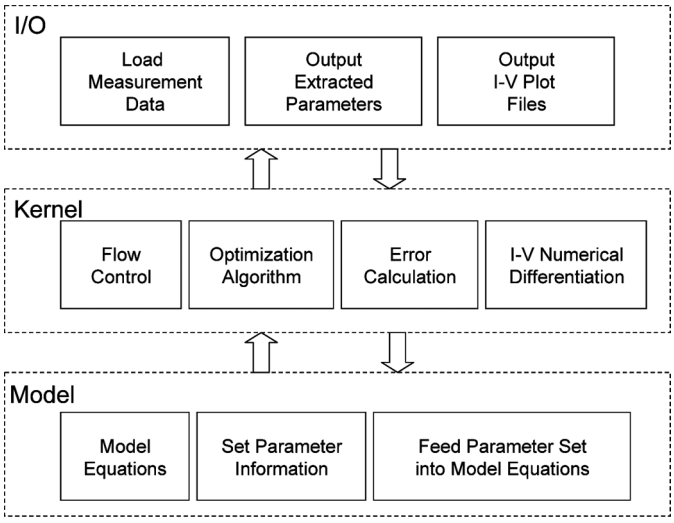


FIGURE 2.—The system architecture of the parameter extraction program implemented in the tool.

Differential Evolution (DE)

Differential evolution (DE) starts with a random population and generates new offspring by forming a trial vector of each parent individual (target vector) of the population. DE is composed of three operations: mutation, crossover, and selection. The mutation operator creates mutant individuals by adding the weighted difference between two individuals to a third individual. Specifically, for each individual x_i^t, a mutant individual is generated according to

$$v_i^{t+1} = x_{r_1}^t + F \cdot (x_{r_2}^t - x_{r_3}^t), \tag{19}$$

where $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$ are chosen randomly and are different from the running index i. $F \in (0, 2]$ is a real and constant factor that controls the amplification of the differential variation. Crossover operator is used to increase the diversity of the population. The trial individual

$$u_i^{t+1} = (u_{i,1}^{t+1}, \ldots, u_{i,D}^{t+1}), \tag{20}$$

is formed where

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{if } rand(j) \le CR \\ x_{i,j}^t, & \text{otherwise} \end{cases}, \quad j = 1, 2, \ldots, D, \tag{21}$$

$rand(j)$ is the jth uniform random number distributed within $[0, 1]$. $CR \in [0, 1]$ is the user-defined crossover rate. The selection operator adopts the greedy selection scheme. The trial individual u_i^{t+1} is compared to target individual x_i^{t+1}. If the fitness of the trial individual is lower than that of the target individual, the target individual will be replaced by the trial individual.

PSO

For an optimization problem with n variables, a fixed number of particles are initially generated and randomly spread over the search space. The position of each particle is a potential solution of the optimization problem. Each particle has its own velocity, and it can fly to next position in the search space according to its velocity. For an n dimensional problem, the position and velocity vector for the ith particle in the population can be represented as

$$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}), \tag{22}$$

$$v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,n}). \tag{23}$$

In the flying process, each particle can memorize the best position which has ever been (it is a locally best position), and knows the best position of the whole populations have ever been, which is a globally best position. By part of individual experience and part of group's experience, particles move towards the desired solution. For each iteration, the velocity of particle follows

$$v_i^{t+1} = w v_i^t + C_1 \cdot r_1 \cdot (p_i^t - x_i^t) + C_2 \cdot r_2 \cdot (g^t - x_i^t). \tag{24}$$

Then, the position of particle is updated by

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{25}$$

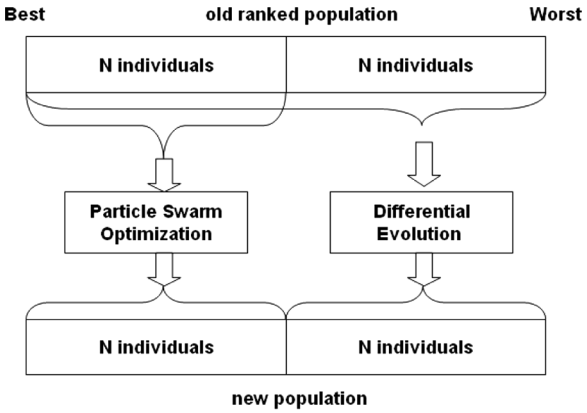
where t is the iteration number, i is the particle index, p denotes the local best position, and g denotes the global best position. r_1 and r_2 are random numbers uniformly distributed in the interval of $[0, 1]$ which act as the stochastic sources in the algorithm. The constant w is so-called the inertia weight of particle which represents the influence of previous velocity of particle on its new velocity. C_1 and C_2 control the effects of local and global guides, respectively. Notably, a convergence of PSO algorithm could be ensured with considering $w = 0.7298$, $C_1 = C_2 = 1.49618$ [19].

The Hybrid DE-PSO Algorithm

The hybrid DE-PSO algorithm combines the standard velocity and position update rules of PSO and the operations of differential mutation and probability crossover from DE. Figure 3(a) shows a flowchart of the hybrid DE-PSO algorithm. For all iterations, individuals are first sorted according to their fitness values. The individuals in the better half of the population proceed as PSO. On the other hand, the remaining part of new population is generated by modifying the better half of the population using differential evolution.

Programming procedures for the hybrid DE-PSO, PSO, and DE algorithms are shown in Figs. 3(b–d), respectively, where m_i represents the mutant particle created by adding the weighted position difference between two particles to a third particle. r_1, r_2, r_3 are randomly chosen within $1, 2, \ldots, N$ and are different from each other. F is a real and constant factor that controls the amplification of the differential variation. CR is the crossover rate. In this work, we use $F = 0.8$ and $CR = 0.3$ [21].

The sorting of each iteration preserves the better obtained solutions. If the solution generated by differential evolution lies within the better half of the population, it will proceed



(a)

Hybrid DE-PSO Algorithm

Population size $2N$

Dimension D

Maximum iteration number t_{max}

Randomly initialize positions and velocities for N particles

for i = 1 to $2N$ **do**

  for j = 1 to D **do**

  $diff = max(j)\text{-}min(j)$

  $x_{i,j} = min(j) + rand() \times diff$

  $v_{i,j} = (2 \times rand() - 1) \times 0.5 \times diff$

  end for

end for

Evaluate fitness for each particle

for t = 1 to t_{max} **do**

  Sort particles according to their fitness value

  (i = 1 denotes the best fit particle; $i = N$ denotes the worse particle)

  for i = 1 to N **do**

    Differential Evolution (*i*)

  end for

  for i = 1 to N **do**

    Particle Swarm Optimization (*i*)

  end for

  Evaluate fitness for each particle

  Update global best and local best position

  end for

  Output the final global best position

(b)

Particle Swarm Optimization (*i*)

for j = 1 to D **do**

  $v_{i,j} = w \bullet v_{i,j} + C_1 \bullet rand() \bullet (p_{i,j} - x_{i,j}) + C_2 \bullet rand() \bullet (g_j - x_{i,j})$

  $x_{i,j} = x_{i,j} + v_{i,j}$

  if $x_{i,j} < min(j)$ **then**

    $x_{i,j} = min(j)$

    $v_{i,j} = 0.1 \bullet rand() \bullet (max(j)\text{-}min(j))$

  end if

  if $x_{i,j} > max(j)$ **then**

    $x_{i,j} = max(j)$

    $v_{i,j} = -0.1 \bullet rand() \bullet (max(j)\text{-}min(j))$

  end if

end for

(c)

Differential Evolution (*i*)

for j = 1 to D **do**

  $m_{i,j} = x_{r1,j} + F \bullet (x_{r2,j} - x_{r3,j})$

where r_1, r_2, r_3 are randomly chosen within 1,2,…,N and are different from each other

end for

$x_{i+N} = x_i$

for j = 1 to D **do**

  if $rand() < CR$ **then**

    $x_{i+N,j} = m_{i,j}$

  end if

end for

(d)

FIGURE 3.—Continued.

FIGURE 3.—(a) Flowchart of the hybrid DE-PSO algorithm and programming procedures corresponding to the (b) hybrid DE-PSO, (c) PSO, (d) and DE algorithms, respectively.

with PSO. In differential evolution, the individuals in the worse half of the population have chances to be selected to create the mutant individual. This will help increase the diversity of the population. In addition, the difference between individuals decreases with evolution. As a result, the differential evolution technique can help both global search at the beginning of the evolution and local search near the end of the evolution.

The Fitness Functions

Fitness functions are used to evaluate the quality of solutions. Most importantly, the fitness scores of solutions guide the search directions in all evolutionary based algorithms investigated in this work. The measured data contains a set of drain current-gate voltage ($I_d - V_{gs}$) and drain current-drain voltage ($I_d - V_{ds}$) curves. Our goal is to minimize the error between the model-generated data and measured data. Therefore, we define the fitness function as the root mean square error of current-voltage ($I - V$) curves. For analog design, the model is even required to produce accurate derivatives of current. Therefore, the error of transconductance $G_m = \partial I_d / \partial V_{gs}$ and output conductance $G_{ds} = \partial I_d / \partial V_{ds}$ should be considered in the fitness function. We define the fitness function as follows:

$$f = f_1 + f_2 + f_3 + f_4, \tag{26}$$

where

$$f_1 = \frac{\sqrt{\sum_{V_{gs}} (I_d^{\text{mod}} - I_d^{\text{exp}})^2 / N}}{Max(I_d^{\text{exp}})}, \tag{27}$$

$$f_2 = \frac{\sqrt{\sum_{V_{gs}} (G_m^{\text{mod}} - G_m^{\text{exp}})^2 / N}}{Max(G_m^{\text{exp}})}, \quad (28)$$

$$f_3 = \frac{\sqrt{\sum_{V_{ds}} (I_d^{\text{mod}} - I_d^{\text{exp}})^2 / N}}{Max(I_d^{\text{exp}})}, \quad (29)$$

and

$$f_4 = \frac{\sqrt{\sum_{V_{ds}} (G_{ds}^{\text{mod}} - G_{ds}^{\text{exp}})^2 / N}}{Max(G_{ds}^{\text{exp}})}. \quad (30)$$

N is the number of measured data points. The superscripts "mod" and "exp" represents the model-generated values and experimental data, respectively. The meanings of f_1, f_2, f_3, and f_4 are the RMS errors of $I_d - V_{gs}$ curves, G_m curves, $I_d - V_{ds}$ curves, and G_{ds} curves, respectively.

Results and discussion

Figure 4 compares the fitness score with respect to the generation for with and without applying parameter extraction procedure when PSO is implemented. The parameter extraction procedure is described in Table 2. As shown in Fig. 4, the fitness score oscillates up and down when parameter extraction procedure is applied. In a single step of parameter extraction procedure, a local region or subset of $I - V$ curves are optimized. However, this may cause the error of other region of $I - V$ curves to increase. Similar results can be found when DE-PSO is implemented, as shown in Fig. 5. This indicates that it is not easy to clearly decouple parameters corresponding to different $I - V$ curves. Therefore, parameter extraction with several steps may not be an efficient way to extract surface-potential-based model parameters.

We further investigate the feasibility to extract parameters without extraction procedure. In other words, we extract all 14 direct current (DC) parameters in a single step. GA, DE, PSO, and the proposed hybrid DE-PSO algorithm are implemented in our optimization kernel. In our experiments, the population size is fixed to 50, and the maximum number of generations is set to 2000. Thirty independent runs of were carried out for each algorithm. The fitness score is defined as shown in Eqs. (26)–(30), which is the sum of $I_d - V_{gs}$ error, $I_d - V_{ds}$ error, G_m error, and G_{ds} error.

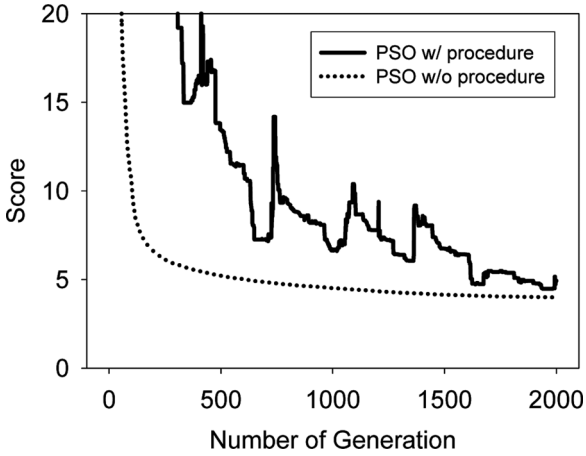


FIGURE 4.—The fitness score versus the number of generation for the optimization with (solid line) and without (dashed line) applying parameter extraction procedure when PSO is implemented.

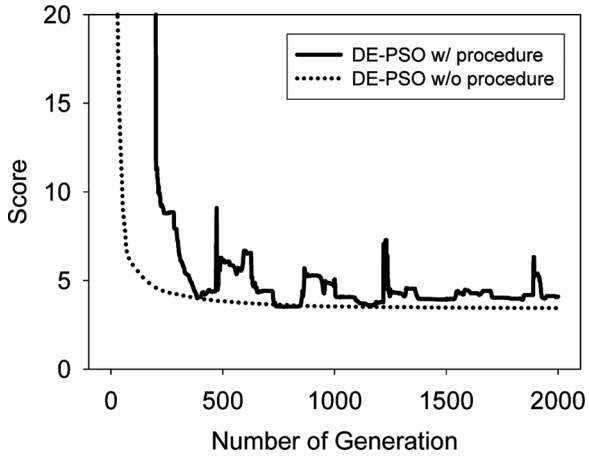


FIGURE 5.—The fitness score versus the generation for the optimization with (solid line) and without (dashed line) applying parameter extraction procedure when DE-PSO is implemented.

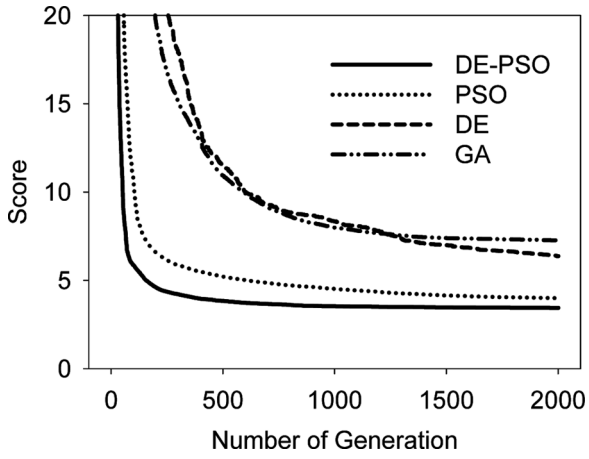


FIGURE 6.—Plot of the score versus the number of generations among different algorithms.

TABLE 3.—Comparison of successful runs and the number of generations among different extraction algorithms.

	GA	DE	PSO	DE-PSO
S	2	1	24	30
Max	1967	–	1922	824
Median	–	1766	579	158
Min	310	–	163	56
Average	1139	1766	1249	241

TABLE 4.—A comparison of accuracy with respect to different extraction algorithm.

	GA (%)	DE (%)	PSO (%)	DE-PSO (%)
$\varepsilon_{\max}$ of $I_d - V_{gs}$	3.81	1.93	0.80	0.55
$\varepsilon_{\min}$ of $I_d - V_{gs}$	0.50	0.50	0.41	0.41
ε_{av} of $I_d - V_{gs}$	0.90	0.82	0.54	0.45
$\varepsilon_{\max}$ of G_m	7.84	7.17	2.46	1.65
$\varepsilon_{\min}$ of G_m	1.73	1.80	1.44	1.43
ε_{av} of G_m	3.76	2.78	1.74	1.49
$\varepsilon_{\max}$ of $I_d - V_{ds}$	6.59	2.02	1.05	0.63
$\varepsilon_{\min}$ of $I_d - V_{ds}$	0.55	0.88	0.39	0.40
ε_{av} of $I_d - V_{ds}$	1.07	1.36	0.58	0.44
$\varepsilon_{\max}$ of G_{ds}	7.43	1.89	1.42	1.15
$\varepsilon_{\min}$ of G_{ds}	1.05	1.11	1.03	0.98
ε_{av} of G_{ds}	1.54	1.41	1.14	1.08

TABLE 5.—Final extracted results of devices with different dimensions.

		RMS errors					
Length (nm)	Width (nm)	$I_d - V_{gs}$ (%)	G_m (%)	$I_d - V_{ds}$ (%)	G_{ds} (%)	Number of generations	Successful runs
16	16	0.46	1.60	0.40	0.70	241	30
32	5000	0.23	1.37	0.86	1.00	250	30
32	32	0.28	1.01	0.56	0.91	328	30
45	110	0.32	1.43	0.82	1.44	476	22
45	1000	0.20	0.82	0.53	0.97	358	30

Figure 6 shows the fitness score convergence behavior of the proposed hybrid DE-PSO in comparison with conventional PSO, DE, and GA. The tested case was N-MOSFET device with $W/L = 16\,\text{nm}/16\,\text{nm}$. The demonstrated results are with average results of 30 independent runs. It can be found that DE-PSO can achieve much lower fitness score than DE and GA after 2000 generations. In addition, DE-PSO reaches the same good fitness score as PSO in fewer generations. The probable reason is that the mutation and crossover scheme facilitates the global search ability of PSO and the selection scheme reduces resource waste on poor individuals.

Table 3 lists the comparison of successful runs and the number of generations among different extraction algorithms. S denotes the number of successful runs. A run is considered to be successful if it is able to reach the specified goals. That is, $I_d - V_{gs}$ error within 1%, $I_d - V_{ds}$ error within 1%, G_m error within 2%, and G_{ds} error within 2%. The maximum, median, minimum, and the average number of generation of successful runs are also listed in the table. It is found that the successful runs are less than 2 for GA and DE. On the other hand, all 30 runs can reach the specified goal successfully when DE-PSO is applied. Besides, the number of generations required for successful runs are significantly fewer for DE-PSO than that for PSO. This result indicates that the hybrid DE-PSO is more effective and efficient than other evolutionary methods.

Table 4 lists the accuracy comparison among different extraction algorithms. The maximum, minimum, and average errors are denoted by $\varepsilon_{\max}$, $\varepsilon_{\min}$, and ε_{av}, respectively. It is found that GA and DE have difficulty reducing G_m error. A few runs of PSO also fail due to slightly larger G_m error. One competitive advantage of the proposed DE-PSO algorithm is the method can find better solutions and keep the minimal G_m error.

Table 5 lists the final extracted results of devices of different dimensions obtained by the hybrid DE-PSO algorithm, including the $I_d - V_{gs}$ error, G_m error, $I_d - V_{ds}$ error, G_{ds} error. The average number of generations required to reach our specified goal and the number of successful runs are also summarized in the Table 5. By the verification of devices of different dimensions, the hybrid DE-PSO shows excellent numerical results in terms of the solution accuracy, computational efficiency and effectiveness for the parameter extraction of PSP MOSFET model.

Figures 7–11 illustrate the simulated results with the final extracted parameters for the tested NMOS devices with

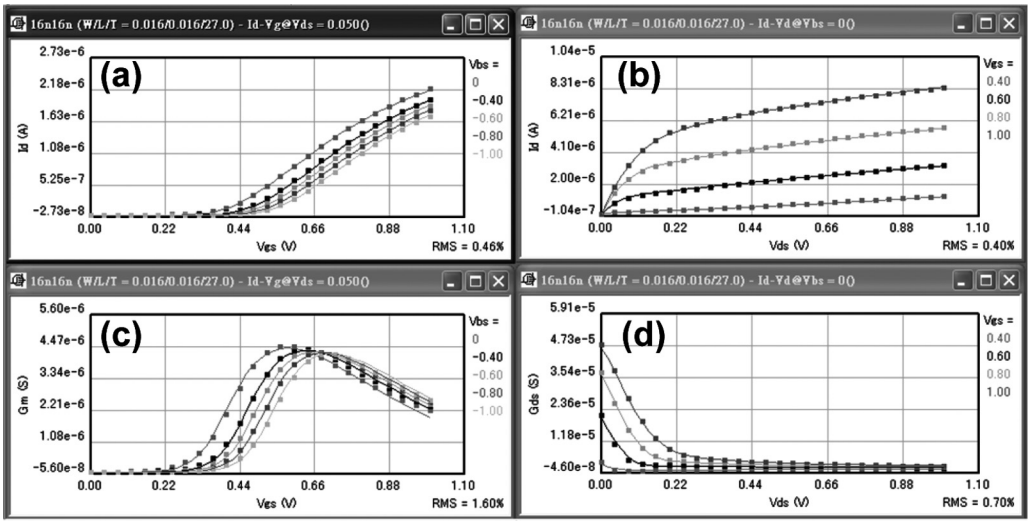


FIGURE 7.—Measured and simulated characteristics for $W/L = 16\,\text{nm}/16\,\text{nm}$ NMOS device using parameters extracted by DE-PSO algorithm: (a) $I_d - V_{gs}$ at $V_{ds} = 0.05\,\text{V}$; (b) $I_d - V_{ds}$ at $V_{bs} = 0\,\text{V}$; (c) transconductance; and (d) output conductance.

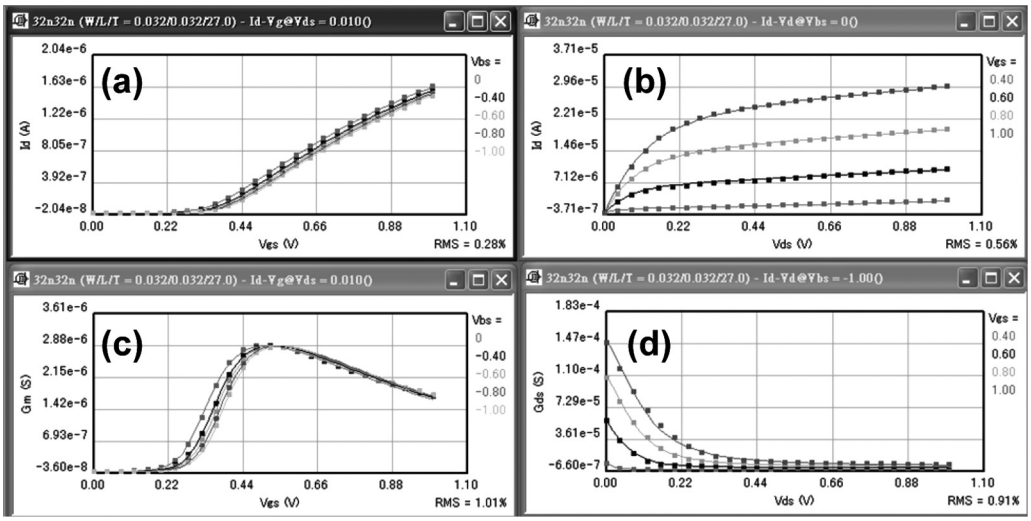


FIGURE 8.—Measured and simulated characteristics for $W/L = 32\,\text{nm}/32\,\text{nm}$ NMOS device using parameters extracted by DE-PSO algorithm: (a) $I_d - V_{gs}$ at $V_{ds} = 0.05\,V$; (b) $I_d - V_{ds}$ at $V_{bs} = 0\,V$; (c) transconductance; and (d) output conductance.

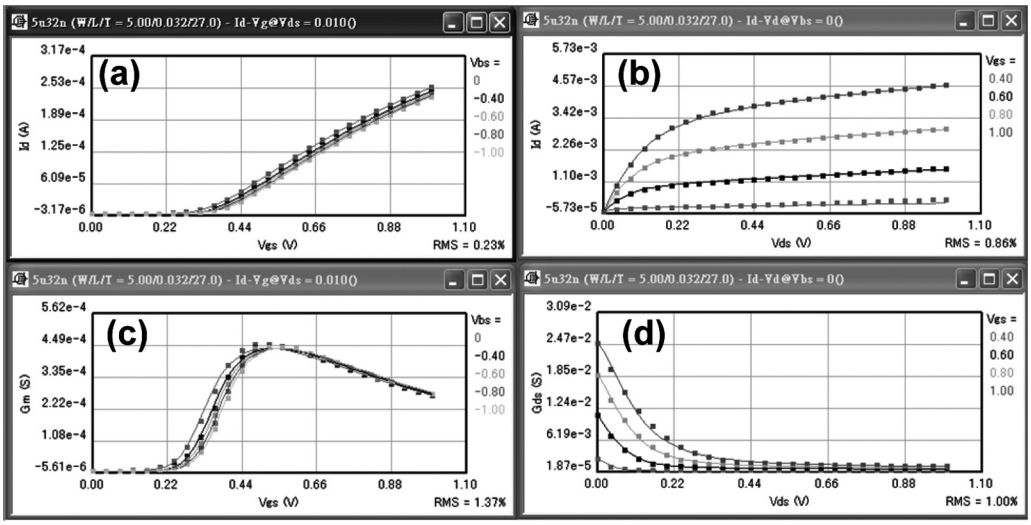


FIGURE 9.—Measured and simulated characteristics for $W/L = 5\,\mu\text{m}/32\,\text{nm}$ NMOS device using parameters extracted by DE-PSO algorithm: (a) $I_d - V_{gs}$ at $V_{ds} = 0.05\,\text{V}$; (b) $I_d - V_{ds}$ at $V_{bs} = 0V$; (c) transconductance; and (d) output conductance.

various device dimensions. The solid lines are the simulated results and the dotted lines are the measured data. It is obvious that the simulated $I - V$ characteristics are in good agreement with the measured data as well as their first derivatives.

CONCLUSIONS

In this article, we have successfully proposed a hybrid DE-PSO algorithm for surface-potential-based PSP model parameter extraction of nanoscale MOSFETs'. Based on the combination of the standard velocity and position update rules of PSO and the operations of differential mutation and probability crossover from DE, our method successfully extracts 14 DC parameters for a single MOSFET device simultaneously. The accuracy and efficiency of the algorithm has been obtained, verified and reported in this study for sub-45 nm NMOSFETs, through several testing cases which have one set of $I_d - V_{gs}$ and $I_d - V_{ds}$ curves. For more sets of $I - V$ curves, it needs to be investigated in our future work. Compared with GA, DE, and PSO, DE-PSO is particularly able to overcome the difficulty to obtain accurate G_m characteristics. The hybrid DE-PSO algorithm is expected to be a promising method to extract model parameters for different kinds of semiconductor devices. It was shown recently that PSO is actually not a new paradigm but an elitist real-coded genetic algorithm [39]. Also, the dimensional problems of Eqs. (24) and (25) have pointed out. Nevertheless, it will not affect any of the reported results. We are currently studying other operation schemes for population in DE and PSO including alternating generations for more robust methodologies.

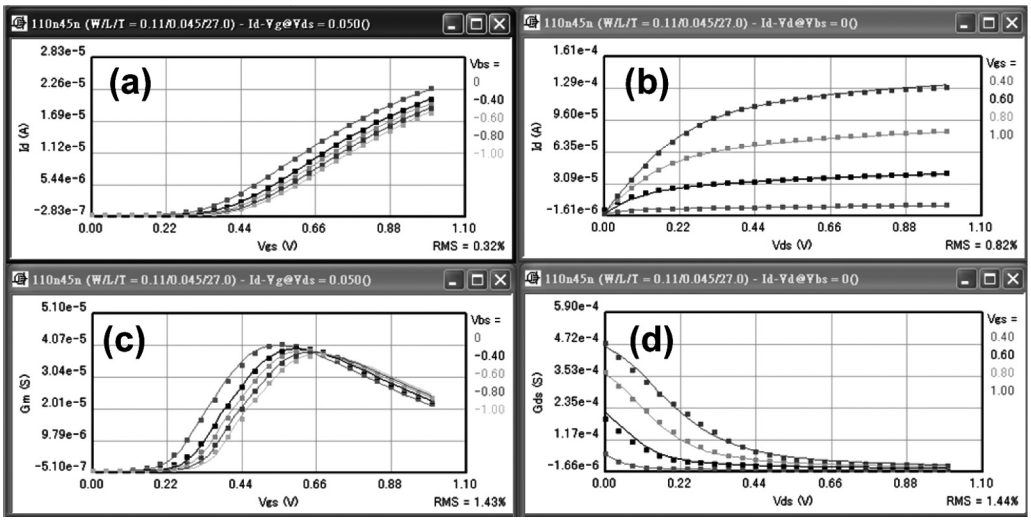


FIGURE 10.—Measured and simulated characteristics for $W/L = 110\,\text{nm}/65\,\text{nm}$ NMOS device using parameters extracted by DE-PSO algorithm: (a) $I_d - V_{gs}$ at $V_{ds} = 0.05\,V$; (b) $I_d - V_{ds}$ at $V_{bs} = 0V$; (c) transconductance; and (d) output conductance.

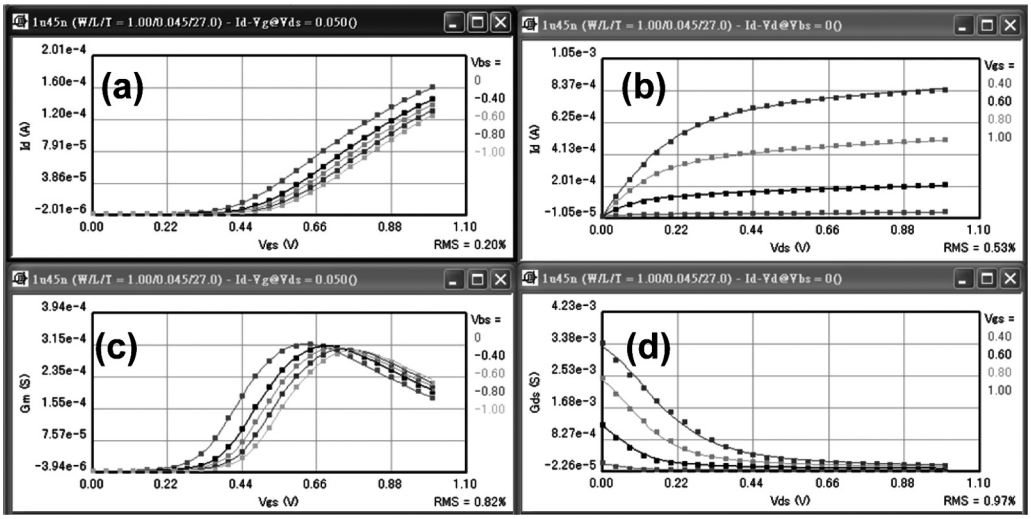


FIGURE 11.—Measured and simulated characteristics for $W/L = 1\,\mu\text{m}/45\,\text{nm}$ NMOS device using parameters extracted by DE-PSO algorithm: (a) $I_d - V_{gs}$ at $V_{ds} = 0.05\,V$; (b) $I_d - V_{ds}$ at $V_{bs} = 0\,V$; (c) transconductance; and (d) output conductance.

ACKNOWLEDGMENTS

This work was supported in part by Taiwan National Science Council (NSC) under Contracts No. NSC-97-2221-E-009-154-MY2 and No. NSC-99-2221-E-009-175, and by the Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan under a 2009-2010 grant.

REFERENCES

1. Arora, N.D. *MOSFET Models for VLSI Circuit Simulation: Theory and Practice*; Springer-Verlag: New York, 1993.
2. Tsividis, Y. *Operation and Modeling of the MOS Transistor*; McGraw-Hill: New York, 1999.
3. Compact Model Council. Available at: http://www.eigroup.org/cmc/
4. Pao, H.C.; Sah, C.T. Effects of diffusion current on characteristics of metal-oxide (insulator)-semiconductor transistors. Solid State Electronics **1966**, *9*, 927–937.
5. Brews, J.R. A charge-sheet model of the MOSFET. Solid-State Electronics **1978**, *21*, 345–355.
6. Suetake, M.; Suematsu, K.; Nagakura, H.; Miura-Mattausch, M.; Mattausch, H.J.; Kumashiro, S.; Yamaguchi, T.; Odanaka, S.; Nakayama, N. HiSIM: A drift-diffusion-based advanced MOSFET model for circuit simulation with easy parameter extraction. Proc. of Simulation of Semiconductor Processes and Devices (SISPAD) 2000, 261–264.
7. van Langevelde, R.; Scholten, A.J., Klaassen, D.B.M. MOS Model 11, level 1101 NL-UR 2002/802, Philips Electron. N. V., 2002. Available at: http://www.nxp.com/models/mos_models/model11/ (accessed March 15, 2011).

8. Bendix, P.; Rakers, P.; Wagh, P.; Lemaitre, L.; Grabinski, W.; McAndrew, C.C.; Gu, X.; Gildenblat, G. RF distortion analysis with compact MOSFET models. Proc. of Custom Integrated Circuit Conference 2004; 9–12.
9. Xi, X.J. BSIM5 MOSFET Model. Proc. of Solid-State and Integrated Circuit Technology 2004; 920–923.
10. Gildenblat, G.; Wang, H.; Chen, T.L.; Gu, X.; Cai, X. SP: An advanced surface-potential-based compact MOSFET model. IEEE Journal of Solid-State Circuits 2004; 1394–1406.
11. Gildenblat, G.; Xin Li; Wu, W.; Hailing Wang; Jha, A.; van Langevelde, R.; Smit, G.D.J.; Scholten, A.J.; Klaassen, D.B.M. PSP: An Advanced Surface-Potential-Based MOSFET Model for Circuit Simulation. IEEE Trans. on Electron Devices **2006**, *53*, 1970–1993.
12. PSP 102.1 User's Manual Available at: http://pspmodel.asu.edu/ (accessed March 15, 2011).
13. Marquardt, D.W. An algorithm for least squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics **1963**, *11*, 431–441.
14. Doganis, K.; Scharfetter D.L. General optimization and extraction of IC device model parameters. IEEE Trans. Electron Devices **1983**, *30*, 1219–1228.
15. Wang, S.J.; Lee, J.Y.; Chang, C.Y. An efficient and reliable approach for semiconductor device parameter extraction. IEEE Trans. Computer-Aided Design **1986**, *6*, 170–178.
16. Sharma, M.; Arora, N.D. OPTIMA: A nonlinear model parameter extraction program with statistical confidence region algorithms IEEE trans. Computer-Aided Design **1993**, *12*, 982–987.
17. Li, Y.; Cho, Y.-Y. Intelligent BSIM4 model parameter extraction for Sub-100nm MOSFETs era. Japanese Journal of Applied Physics **2004**, *43*, 1717–1722.
18. Kennedy, J.; Eberhart, R. Particle swarm optimization. Proc. of the IEEE International Conference on Neural Networks, Piscataway, NJ, 1995; 1942–1948.
19. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in multidimensional complex space. IEEE Transactions on Evolutionary Computation **2002**, *6*, 58–73.
20. Storn, R.; Price, K. Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces. Journal of Global Optimization **1997**, *11*, 341–359.
21. Gamperle, R.; Dmuller, S.; Koumoutsakos, P. A parameter study for the differential evolution. Proc. of International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, 2002; 293–298.
22. Wang, K.; Ye, M. Parameter determination of Schottky-barrier diode model using differential evolution. Solid-State Electronics **2009**, *53*, 234–240.
23. Das, S.; Abraham, A.; Konar, A. Particle swarm optimization and differential evolution algorithms: Technical analysis. Applications and Hybridization Perspectives, Studies in Computational Intelligence **2008**, *116*, 1–38.
24. Pant, M.; Thangaraj, R.; Grosan, C.; Abraham, A. Hybrid differential evolution – particle swarm optimization algorithm for solving global optimization problems. Proc. IEEE Third International Conference on Digital Information Management **2008**, 18–24.
25. Zhanga, C.; Ning, J.; Lu, S.; Ouyang, D.; Ding, T. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. Operations Research Letters **2009**, *37*, 117–122.
26. Tseng, Y.H.; Li, Y. A parameter extractor for PSP compact modeling of Nano-CMOS devices. Proc. of IEEE Workshop on Compact Modeling **2008**, 61–66.
27. Zhou, Q.; Yao, W.; Wu, W.; Li, X.; Zhu, Z.; Gildenblat, G. Parameter extraction for the PSP MOSFET model by the combination of genetic and Levenberg-Marquardt algorithms. Proc. of IEEE ICMTS **2009**, 137–142.
28. Li, Y. Parallel genetic algorithm for intelligent model parameter extraction of metal-oxide- semiconductor field effect transistors. Materials and Manufacturing Processes **2009**, *24*, 243–249.
29. Li, Y. Hybrid Intelligent approach for modeling and optimization of semiconductor devices and nanostructures. Computational Material Science **2009**, *45*, 41–51.
30. Goldberg, D.E. *Genetic Algorithm in Search, Optimization and Machine Learning*; Addison-Wesley: New York, 1989.
31. Chelouah, R.; Siarry, P. A continuous genetic algorithm designed for the global optimization of multimodel functions. Journal of Heuristics **2000**. *6*, 191–213.
32. Cai, X.; Wang, H.; Gu, X.; Gildenblat, G.; Bendix, P. Application of the genetic algorithm to compact MOSFET model development and parameter extraction. Nanotech **2003**, *2*, 314–317.
33. Li, Y.; Cho, Y.Y.; Wang, C.S.; Huang, K.Y. A genetic algorithm approach to InGaP/GaAs HBT parameters extraction and RF characterization. Japanese Journal of Applied Physics. **2003**, *42*, 2371–2374.
34. Li, Y. An automatic parameter extraction technique for advanced CMOS device modeling using genetic algorithm. Microelectronic Engineering **2007**, *84*, 260–272.
35. Li, Y.; Yu, S.-M. A coupled-simulation-and-optimization approach to nanodevice fabrication with minimization of electrical characteristics fluctuation. IEEE Transactions on Semiconductor Manufacturing **2007**, *20*, 432–438.
36. Li, Y. Hybrid intelligent approach for modelling and optimization of semiconductor devices and nanostructures. Computational Materials Science **2009**, *45*, 41–51.
37. Li, Y.; Yu, S.-M.; Li, Y.-L. Intelligent optical proximity correction using genetic algorithm with model- and rule-based approaches. Computational Materials Science **2009**, *45*, 65–76.
38. Paszkowicz, W. Genetic algorithms, a nature-inspired tool: Survey of applications in materials science and related fields. Materials and Manufacturing Processes **2009**, *24*, 174–197.
39. Chakraborti, N.; Das, S.; Jayakanth, R.; Pekoz, R.; Erkoç, Ş. Genetic algorithms applied to Li+ ions contained in carbon nanotubes: An investigation using particle swarm optimization and differential evolution along with molecular dynamics. Materials and Manufacturing Processes **2007**, *22*, 562–569.