Modeling the optical constants of solids using acceptance-probability-controlled simulated annealing with an adaptive move generation procedure

Aleksandra B. Djurišić,[1] Aleksandar D. Rakić,[2] and Jovan M. Elazar[1]
[1]*Faculty of Electrical Engineering, University of Belgrade, P.O. Box* 816, *Belgrade, Yugoslavia*
[2]*University of Queensland, Department of Electrical and Computer Engineering, St. Lucia QLD 4072, Brisbane, Australia*
(Received 22 April 1996; revised manuscript received 5 August 1996)

The acceptance-probability-controlled simulated annealing with an adaptive move generation procedure, an optimization technique derived from the simulated annealing algorithm, is presented. The adaptive move generation procedure was compared against the random move generation procedure on seven multiminima test functions, as well as on the synthetic data, resembling the optical constants of a metal. In all cases the algorithm proved to have faster convergence and superior escaping from local minima. This algorithm was then applied to fit the model dielectric function to data for platinum and aluminum. [S1063-651X(97)13003-3]

PACS number(s): 02.70.−c, 78.20.Ci, 78.66.−w

I. INTRODUCTION

The interpretation of optical spectra is often accomplished by fitting the model to experimental data. It is usually hard to provide good initial values for adjustable parameters of the model. However, all fitting routines based on classical optimization algorithms, require initial parameter values close to the final values to provide meaningful solution. Practically, it is necessary to rerun the fitting routine many times, while changing the initial model-parameter values, before an acceptable fit is obtained. Even then, there is always a doubt whether the obtained solution is really the global minimum.

In this paper we propose an efficient and fully automatic alternative to conventional fitting routines, an interesting simulated annealing-based technique. Due to the nature of this algorithm, initial parameter values are not required. Unlike its purely numerical counterparts, this seminumerical technique efficiently finds the global minimum and accurately determines all the adjustable parameters of the model, without any external supervision.

The simulated annealing algorithm [1] (SA), has its origins in the work of Metropolis *et al.* [2]. It is based on the analogy with annealing of solids: the function to be minimized, called the cost function, is analogous to the energy, regardless of its physical nature. Based on this analogy a control parameter, called temperature T, with the same units as the cost function is introduced. Starting from an arbitrary initial state, the algorithm generates a sequence of random changes of model parameters, or ''moves'' in parameter state space. Downhill moves are always accepted, while the acceptance probability (AP) of an uphill move is given by Boltzmann distribution $\pi=\exp(-\Delta E/T)$, where ΔE is the change of the cost function and T is the temperature. The simulated annealing algorithm is actually a reiteration of Metropolis algorithms, evaluated at decreasing values of the control parameter T [3]. The literature to date [4–11] describes several different cooling schedules.

In the present work we propose two significant modifications to the SA algorithm for functions of continuous variables. *First,* we use the AP rather than the temperature to control the annealing schedule [12]. That means that the AP is varied in a prescribed manner in time while the temperature is adaptively changed in accordance with the average change in cost function. Therefore, our schedule requires occasionally increasing the temperature to melt the system frozen in local minimum. This feature makes acceptance-probability-controlled simulated annealing (APCSA), superior to classical SA in terms of its ability to escape from local minima. *Second,* we improve the move generation procedure. The generator of changes is the important and, probably, the most problematical element of the SA algorithm [13]. It was already noted [7,12] that in optimization problems with a large number of variables, moves which require changes in all variables can cause the instabilities in the solution. The number of variables to be changed in one move is therefore reduced, and sometimes is chosen randomly [7,14], or even reduced to only one variable per move, as in Ref. [15]. In this paper we propose the adaptive move generation procedure, based on the idea that the frequency of making the move along a certain direction should depend on the sensitivity of the cost function with respect to that variable.

In Sec. III we describe the employed model of the optical constants of metals. Section III is devoted to the description of the APCSA algorithm with the adaptive move generation procedure, while Sec. IV describes tests and application of this algorithm.

II. MODEL FOR OPTICAL DIELECTRIC FUNCTION

It is well known that the optical properties of solids can be described in terms of complex optical dielectric function $\hat{\epsilon}_r(\omega)=\epsilon_{r1}(\omega)+i\epsilon_{r2}(\omega)$. It was shown [16,17] that $\hat{\epsilon}_r(\omega)$ could be expressed in the form which separates explicitly the intraband effects (usually referred to as free electron effects) from interband effects (usually associated with bound electrons). In this paper the following model is used:

$$\hat{\epsilon}_r(\omega)=\hat{\epsilon}_r^{(f)}(\omega)+\hat{\epsilon}_r^{(b)}(\omega). \tag{1}$$

The intraband part $\hat{\epsilon}_r^{(f)}(\omega)$ of dielectric function is a well known free electron or Drude model

$$\hat{\epsilon}_r^{(f)}(\omega)=1-\frac{\Omega_p^2}{\omega(\omega+i\Gamma_0)}, \tag{2}$$

while the interband part of the dielectric function $\hat{\epsilon}_r^{(b)}(\omega)$ is a simple semiquantum model resembling the Lorentz result for insulators

$$\hat{\epsilon}_r^{(b)}(\omega)=-\sum_{j=1}^{k}\frac{f_j\omega_p^2}{(\omega^2-\omega_j^2)+i\omega\Gamma_j}, \tag{3}$$

where ω_p is the plasma frequency, k is the number of interband transitions with frequency ω_j, oscillator strength f_j and lifetime $1/\Gamma_j$, while $\Omega_p=\sqrt{f_0}\omega_p$ is the plasma frequency associated with intraband transitions, f_0 is oscillator strength for electrons contributing to intraband processes, and Γ_0 is the intraband damping constant.

We used the following objective function for the model parameter estimation:

$$E(\boldsymbol{p})=\sum_{i=1}^{i=N}\left[\left|\frac{\epsilon_{r1}(\omega_i)-\epsilon_{r1}^{\exp}(\omega_i)}{\epsilon_{r1}^{\exp}(\omega_i)}\right|+\left|\frac{\epsilon_{r2}(\omega_i)-\epsilon_{r2}^{\exp}(\omega_i)}{\epsilon_{r2}^{\exp}(\omega_i)}\right|\right]^2. \tag{4}$$

In common situations, at least some of the frequencies of the interband transitions are known from the band structure calculations or can be anticipated from the visible structure optical constants. Therefore, it is possible to make valid assumptions only for the ω_j initial values, while for the other parameters even the order of magnitude is unknown.

III. DESCRIPTION OF THE ALGORITHM

The APCSA algorithm used here has two nested loops. In the outer loop the decrease of the AP is performed directly [12], while the control of the acceptance function is returned to the temperature in the inner loop. The outer loop terminates if the solidification criterion is satisfied [12,6], or if an initially specified maximal number of iterations is reached. The inner loop terminates when the equilibrium condition is satisfied [8]. The quality of the solution obtained by the SA depends not only on the cooling schedule, and move step size, but on the number of parameters to be altered in one iteration as well.

The domain P containing the parameter vector $\boldsymbol{p}=(p(1),p(2),\ldots,p(N))$ is determined by setting the lower and upper boundaries for each parameter, $p_l(k)$ and $p_u(k)$. The efficiency of the generator of changes in the configuration depends largely on two elements: (a) number of variables to be changed in one move, and (b) the move-step adjustment.

As was mentioned above, in optimization problems with a large number of variables, moves which require the change in all variables (here, parameters of the model) can cause instabilities in the solution. The number of variables to be changed in one move is often reduced, and sometimes is chosen randomly [7,14,12], or even reduced to only one variable per move, as in Ref. [15]. However, the random state-generation procedure is far from optimal. We demonstrate here that convergence of the algorithm is accelerated by taking into account the sensitivity of the cost function with respect to the variables. Our algorithm shows the probability of

```
define domain of possible solutions;
discretize the solution space;
set arbitrary initial configuration;
determine initial temperature T^init and average cost function ⟨E_i⟩ at T^init
determine initial frequencies of parameter change;
M=1; J=1;
do while (solidification criterion not satisfied and M<M_max)
  do while (equilibrium condition not satisfied and J<J_max)
    generate move;
    evaluate cost function E_j in new state p_j;
    evaluate change in cost function ΔE_j,i =E_j-E_i;
    apply Metropolis algorithm to determine
    whether to accept move;
    if (move is accepted) then
      update cost function and state, E_i=E_j, p_i=p_j;
      compute D, E_min(M);
    end if
    J=J+1;
  end do
  M=M+1;
  compute ⟨E_acc⟩,⟨|ΔE|_acc⟩;
  compute frequencies of change of parameters;
  decrease step if Δ_k/p(k)<0.005;
  update AP, π_M=π^init exp(-M^2/2σ^2);
  update temperature, T_M = -(⟨|ΔE|_acc⟩ / ln(π_M);
end do
```

FIG. 1. Pseudocode of the APCSA algorithm.

taking the move along a certain coordinate direction proportional to the sensitivity of the cost function with respect to that variable (the adjustable parameter of the model). This improves the mobility of the system, which now shows a preference for steeper slopes in either uphill or downhill directions. Therefore, this generator shows a strong bias toward the moves that cause the greatest energy difference.

The impact of the step size on the quality of the solution has been frequently addressed. Corana *et al.* [15] changed the components of the move-step vector adaptively in order to maintain an acceptance ratio close to 0.5, at all temperatures. Most authors considered it important to decrease the move step during the annealing in order to reduce the fluctuations in the final stage [8,10,11,18,19]. The initial step size has to be comparatively large to provide a sufficient mobility of the algorithm to cover the entire parameter space. Here we adopted the suggestion of Catthoor, de Man, and Vandewalle [8], and reduced the step size in a ''nearly inverse quadratic'' manner. When the ratio $\Delta(k)/p(k)$ is less than 0.005, a further reduction of the move step for that parameter is stopped.

The pseudocode of the APCSA algorithm and the move generation procedure are shown in Figs. 1 and 2, respectively. At each temperature, for each parameter $p(k)$, we determine the average of the absolute change in cost function $\langle|\Delta E|_{\mathbf{k}}\rangle$ by making many random moves for parameter $p(k)$, keeping other parameters fixed. Then, for each k, $k=1,\ldots,N$ we compute the frequency of change $f(k)$ [corresponding to the parameter $p(k)$] using

$$f(k)=0.8\frac{\langle|\Delta E|_{\mathbf{k}}\rangle}{\langle|\Delta E|_{\mathbf{k}}\rangle_{\mathbf{max}}}, \tag{5}$$

where $\langle|\Delta E|_{\mathbf{k}}\rangle_{\mathbf{max}}=\max(\langle|\Delta E|_{\mathbf{k}}\rangle,\ k=1,\ldots,N)$. If the frequency $f(k)$ for changing the parameter $p(k)$ is greater than

```
do k=1, N_par
  N = 1;
  do while (N<N_max)
    generate change of p(k);
    evaluate cost function E_j in new state p_j;
    evaluate change in cost function ΔE_j,i=E_j-E_i;
    update cost function E_i=E_j;
    update state p_i=p_j;
    end if
    N=N+1;
  end do
  compute ⟨|ΔE|⟩_k;
end do
compute ⟨|ΔE|⟩_k,max;
do k=1, N_par
  compute f(k);
end do
```

FIG. 2. Pseudocode of the adaptive move generation procedure.

randomly generated number $p_{ch} \in [0,1]$, parameter $p_i(k)$ is altered to $p_j(k)$,

$$p_j(k) := p_i(k) + r\Delta(k), \tag{6}$$

where r is an integer chosen randomly in the set $(-1,1)$, and $\Delta(k)$ is the step size for parameter $p_i(k)$. If state $p_j(k)$ is outside the specified boundaries $p_l(k)$ and $p_u(k)$, $p_j(k)$ is assigned the value of the nearest boundary.

In each outer loop, the acceptance probability is lowered according to the cooling schedule [12]. Acceptance probabilities depending on the outer loop counter M are given by the normal distribution

$$\pi_M = \pi^{\text{init}} \exp(-M^2/2\sigma^2). \tag{7}$$

The temperature T_M is then determined as

$$T_M = -\frac{\langle |\Delta E|_{\text{acc}} \rangle}{\ln(\pi_M)}, \tag{8}$$

where π_M is the desired acceptance probability, and $\langle |\Delta E|_{\text{acc}} \rangle$ is the average of the absolute change in the cost function at the preceding temperature. This cooling schedule

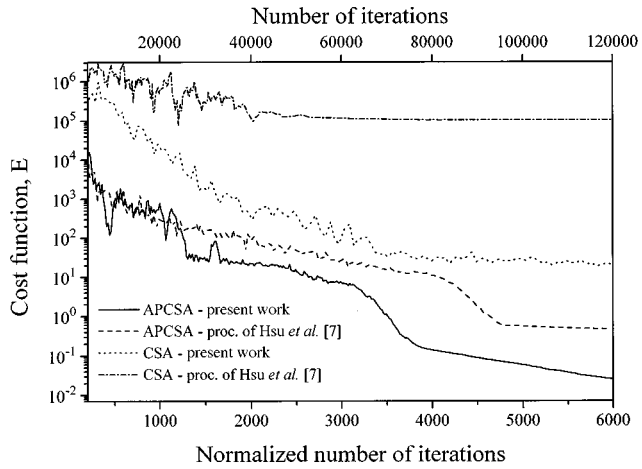

FIG. 3. Cost function vs normalized number of iterations (bottom axis) and number of iterations (top axis) for the function $r(\mathbf{x})$ for $x_i \in [-20,20]$, $i=1$ to 20, with initial values $(-5,5,0,-5,5)\times 4$.

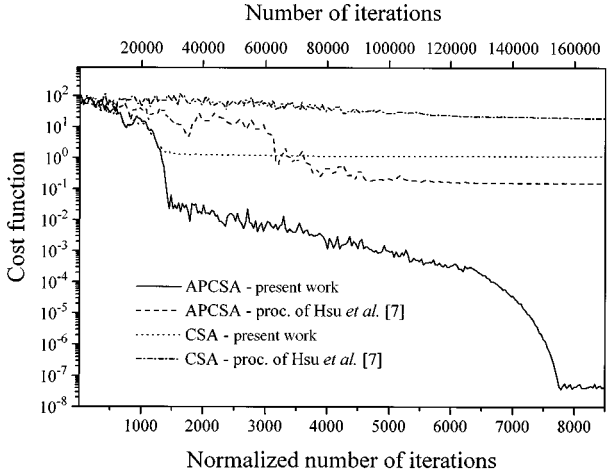

FIG. 4. Cost function vs normalized number of iterations (bottom axis) and number of iterations (top axis) for the function $g(\mathbf{x})$ for $x_i \in [-20,20]$, $i=1$ to 20, with initial values $(-10,5,10,20)\times 5$.

enables occasional rises of the temperature, where the monotonously decreasing function $[-1/\ln(\pi_M)]$ provides the needed average reduction of the temperature.

IV. TESTS AND RESULTS

A. Test functions

In Sec. III we described our APCSA algorithm with the adaptive move generation procedure. As we have already stated, there are two independent features of this algorithm that make it superior to other SA algorithms in terms of convergence rate and robustness: first, the replacement of the temperature control of the cooling schedule with direct control of the acceptance probability; and, second, the introduction of an efficient move generation mechanism. To investigate independently the effect of these two elements, we compared the performance of four algorithms: the APCSA algorithm with the adaptive move generation procedure

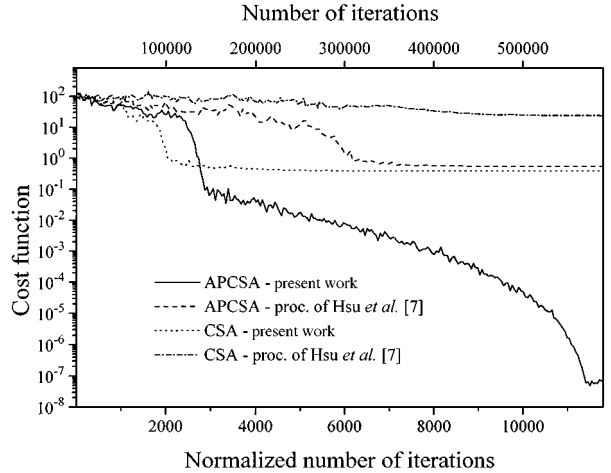

FIG. 5. Cost function vs normalized number of iterations (bottom axis) and number of iterations (top axis) for the function $g(\mathbf{x})$ for $x_i \in [-20,20]$, $i=1$ to 50, with initial values $[(-10,5,10,20)\times 12,-10,5]$.

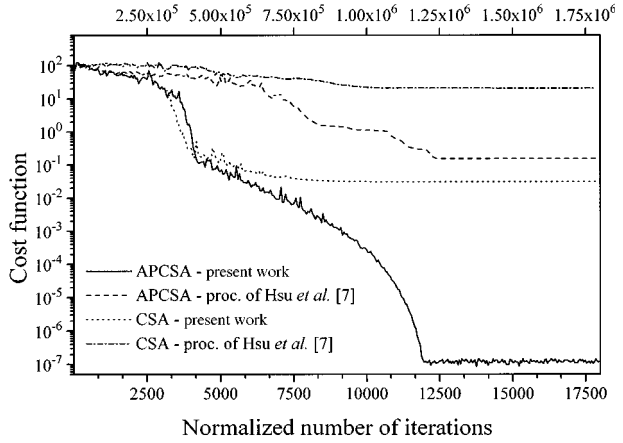

FIG. 6. Cost function vs normalized number of iterations (bottom axis) and number of iterations (top axis) for the function $g(\mathbf{x})$ for $x_i \in [-20,20]$, $i=1$ to 100, with initial values $(-5,5,0,-5,5)\times 20$.

(APCSA1); the APCSA algorithm which uses the random move generation procedure, proposed by Hsu, Chang, and Chan [7](APCSA2); corresponding temperature-controlled SA algorithms with an adaptive move-generation procedure (CSA1); and a random move-generation procedure (CSA2). Algorithms have been tested on a set of seven multiminima test functions. The first five functions have been taken from the literature, while the last two functions are new.

The first two tests were made on the Rosenbrock function in four and 20 dimensions ($R4$ and $R20$), given by

$$r(\mathbf{x})=\sum_{i=1}^{n} 100(x_{i+1}-x_i^2)^2+(1-x_i)^2. \tag{9}$$

This function was investigated by Pronzato *et al.* [20] for two, and by Corana *et al.* [15] for two and four variables. We defined the admissible domain of the function $R4$ as $x_i \in [-200,200]$, $i=1$ and 4, and for $R20$ as $x_i \in [-20,20]$, $i=1$ and 20. It was interesting to compare our results for $R4$ with the results of Corana *et al.* Their SA locates the global minimum for all the starting points tried, achieving the final cost function value of approximately 10^{-7} in all cases. However, their algorithm required 1.3 mil-

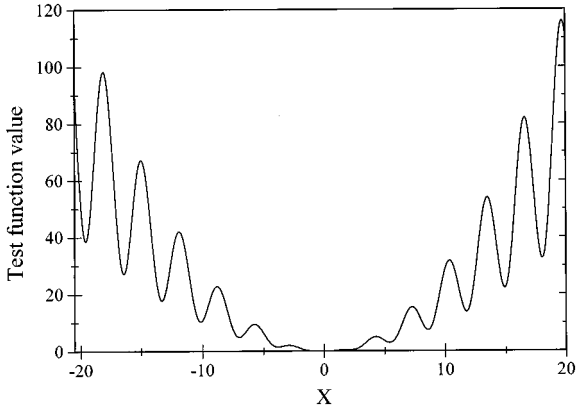

FIG. 7. Section of function $f(\mathbf{x})$ along one axis for $a=0.2$, $b=0.1$, and $c=2$.

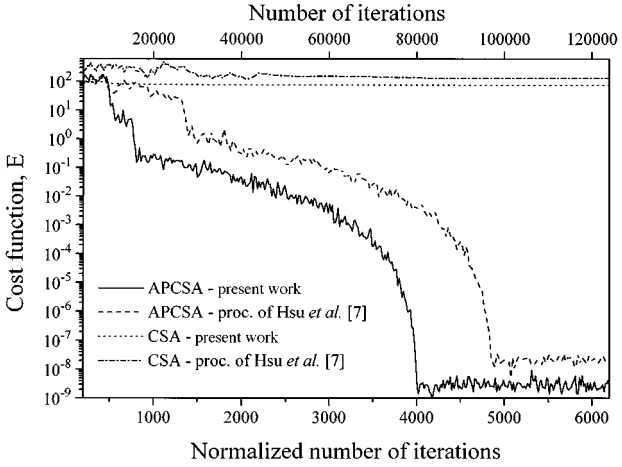

FIG. 8. Cost function vs normalized number of iterations (bottom axis) and number of iterations (top axis) for the function $f(\mathbf{x})$ for $x_i \in [-20,20]$, $i=1$ to 20, with initial values $(-10,15,10,-15)\times 5$.

lion function evaluations to reach the minimum. Our APCSA1 algorithm required less than one-third of this number to achieve a cost function value of approximately 10^{-4}, being actually limited by the final value of the move step vector. Obviously, this solution is equally good. All the coordinates of the minimum are correct to the fourth significant digit, and from this point finding the minimum with an arbitrary precision is a trivial task for any gradient method. Further, we compared the APCSA1, APCSA2, CSA1, and CSA2 on the Rosenbrock function in 20 dimensions ($R20$). Results obtained for $R20$ are presented in Fig. 3. This shows the cost function vs normalized number of iterations, i.e., the number of iterations divided by the number of variables (bottom axis) and the number of iterations (top axis) for investigated algorithms. Obviously, with the increased number of variables, the advantages of our adaptive move-generation procedure and direct control of the acceptance probability became more pronounced.

A further three tests were performed on the test function of Alufi-Pentini, Parisi, and Zirilli [21], later used by Dek-

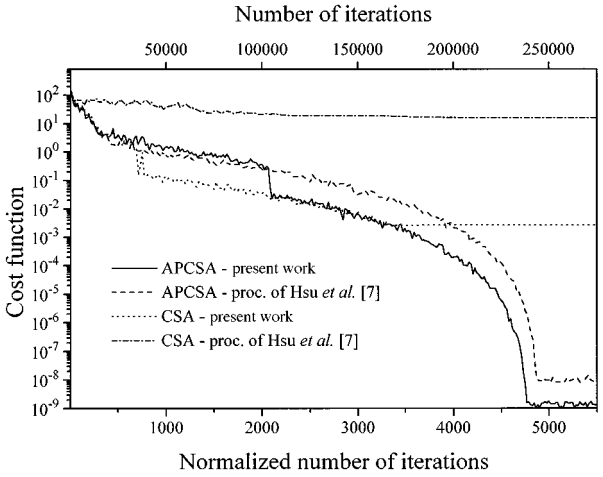

FIG. 9. Cost function vs normalized number of iterations (bottom axis) and number of iterations (top axis) for the function $f(\mathbf{x})$ for $x_i \in [-10,10]$, $i=1$ to 50, with initial values $(-5,5,0,-5,5)\times 10$.

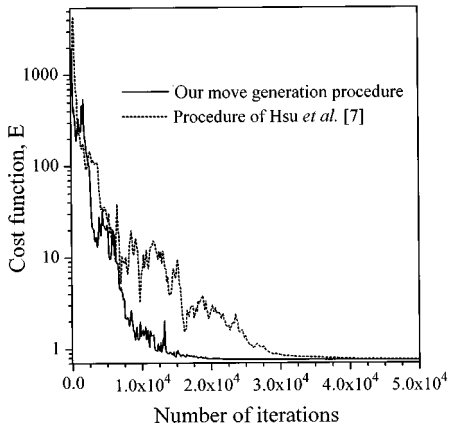

FIG. 10. Cost function vs number of iterations for APCSA algorithm with adaptive move generation procedure (solid line) and APCSA algorithm with procedure of Hsu, Chang, and Chan.

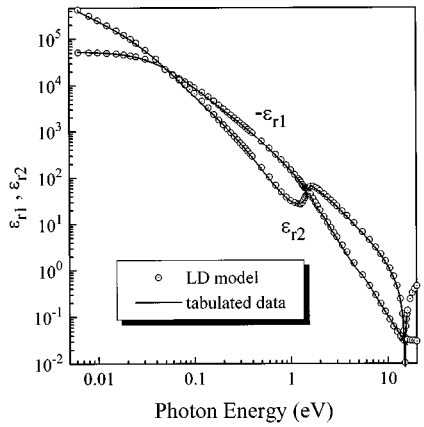

FIG. 11. Platinum: comparison of the tabulated dielectric function (from Ref. [23] — open circles) and model dielectric function calculated in this study (solid line).

kers and Aarts [22] (their test problem $P8$),

$$g(\mathbf{x})=\frac{\pi}{n}\left\{k_1\sin^2\pi y_1+\sum_{i=1}^{n-1}(y_i-k_2)^2(1+k_1\sin^2\pi y_{i+1})+(y_n-k_2)^2\right\}, \tag{10}$$

where $y_i=1+0.25(x_i+1)$, $k_1=10$, $k_2=1$, and $x_i \in[-10,10]$, $i=1$ and n. This function has roughly 5^n local minima. In the cited references, it was tested for three variables. No test of the global optimizing algorithm is complete unless it includes problems with a very large number of variables. The case of 50 and 100 dimensions, which is much more than reported in the literature so far, should be regarded as such. Therefore, we tested our algorithm on problem $P8$ in 20, 50, and 100 dimensions. The obtained results are presented in Figs. 4, 5, and 6, respectively. Finally, we designed the test function, given by

$$f(\mathbf{x})=\sum_{i=1}^{n} ax_i^2+bx_i^2\mathrm{sinc}x_i\,. \tag{11}$$

Section of $f(\mathbf{x})$ along one axis for values of $a=0.2$, $b=0.1$, and $c=2$ is shown in Fig. 7. This function resembles in certain features the paraboloid $q_n(\mathbf{x})$ of Corana *et al.* [15], but actually is a more severe test for SA algorithms. It has wider minima than $q_n(\mathbf{x})$, and has no vertical edges. Small width of the rectangular holes in Corana *et al.* is $q_n(\mathbf{x})$ (width equals 0.1 for variables in interval $[-10\,000, 10\,000]$) prevents such local minima from trapping the algorithm in the early phase of the annealing when move-step size is still large. Figures 8 and 9 show the obtained results for $n=20$ and 50.

It can be concluded that the APCSA algorithm with the adaptive move generation procedure in all cases achieves the lowest cost function value. Furthermore, an adaptive move generation procedure not only improves the performance of the APCSA algorithm, but also significantly ameliorates the CSA algorithm, so that the APCSA algorithm with the random move generation procedure and the CSA algorithm with the adaptive move generation procedure perform similarly, but the results for both algorithms depend on the initial values. In some cases the CSA algorithm with our move generation procedure can obtain the same order of magnitude of the cost function as the APCSA algorithm, or even a slightly lower value, as shown in Fig. 5. The CSA algorithm of Hsu, Chang, and Chan [7], with a random move generation procedure, is in all cases inferior, i.e., it fails to locate global minimum, and achieves the highest cost function value, far from the near-optimal one.

B. Synthetic data

The impact of the state generation procedure on the convergence and quality of the solution is also investigated on functions describing the model for the optical constants of metals. Two sets of synthetic data were generated. The first of these is a set of $\hat{\epsilon}_r^{\mathrm{synt}}(\omega)$ values generated from the model described by Eqs. (2) and (3). Model parameters (called ''target parameters'') are chosen to produce $\hat{\epsilon}_r(\omega)$ values resembling the experimental data for a metal; the second set is obtained by corrupting the first set of $\hat{\epsilon}_r^{\mathrm{synt}}(\omega)$ data with additive Monte Carlo-generated Gaussian noise with frequency dependent variance as described in [12]. Fitting both sets of synthetic data (with and without noise) to the model was performed starting from several points in parameter space, far from the target position. These experiments con-

TABLE I. Parameter values for platinum.

j	0	1	2	3	4	5	6
f_j	1.10	1.44	3.99	0.17	3.14	4.89	12.8
Γ_j	0.08	0.79	4.17	1.26	5.66	14.3	12.0
ω_j	0	0.85	2.39	6.21	9.60	12.6	19.3

TABLE II. Parameter values for aluminum.

j	0	1	2	3	4
f_j	0.498	0.248	0.045	0.196	0.010
Γ_j	0.044	0.304	0.288	1.502	2.794
ω_j	0	0.133	1.546	1.802	5.707

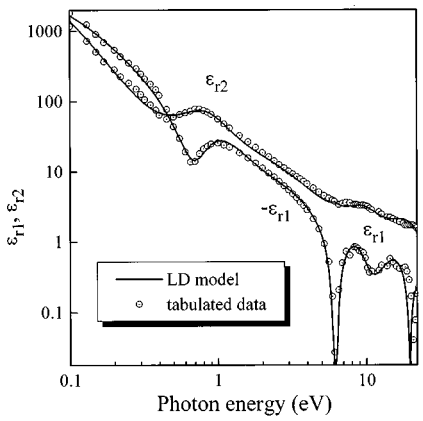

FIG. 12. Aluminum: comparison of the tabulated dielectric function (from Ref. [33] —open circles) and model dielectric function calculated in this study (solid line).

firmed that the initial value of the parameter vector has no impact on the final solution. Figure 10 shows the cost function vs the number of iterations for both algorithms. Although both APCSA algorithms finally converge toward the same cost function value, the algorithm with the adaptive move-generation procedure described here shows significantly faster convergence. Keeping in mind the future applications of this algorithm in real time parametrization problems, the importance of the convergence acceleration is obvious.

C. Application to platinum and aluminum

To illustrate the fitting algorithm described above, we used the data for two metals having qualitatively different optical spectra: platinum as a representative of the transition metals, and aluminum as a metal exhibiting nearly free-electron behavior. We used the tabulation of the optical constants of Pt given in [23] based on the study of Weaver [24]. Weaver used reflectance [25–27] and transmittance [28] data from a number of sources to obtain n and k by the Kramers-Krönig technique. Interband transitions for platinum are expected, according to [29], at about 6.3, 7.8, 9.3, and 10.8 eV. Structure in optical constants [24] is evident at about 7.4, 9.8, and 19.9 eV. Like the other transition metals, platinum possesses a characteristic minimum in ϵ_{r2} (in the case of Pt located near 13 eV) with an additional structure at higher energy. We used six oscillators to model the optical spectrum of Pt in the region between 0.2 and 20 eV.

The obtained parameters are presented in Table I. The oscillator strength values correspond to the plasma frequency $\hbar\omega_p = 5.14$ eV [30]. Figure 11 shows $\epsilon_{r1}(\omega)$ and $\epsilon_{r2}(\omega)$ for platinum. Tabulated data are shown for comparison.

Both the Drude model [31,32] and the semiquantum model [33,34] were often employed for the parametrization of the optical constants of aluminum. For fitting we used the tabulated intrinsic optical constants of aluminum from the recent study of Rakić [33]. Interband transitions are expected at about 0.4, 1.5, 2.1, and 4.5 eV. Final parameter values are presented in Table II. The values of the oscillator strengths correspond to the plasma frequency $\hbar\omega_p = 14.98$ eV [33]. Figure 12 shows excellent agreement between tabulated (open circles) and model (solid line) dielectric functions of aluminum.

V. CONCLUSION

Our principal aim was to improve the APCSA algorithm for the purpose of modeling the optical constants of solids. The algorithm with the adaptive move generation procedure showed faster convergence compared to the procedure with randomly reduced number of parameters to be changed in one iteration. This algorithm was employed for fitting the model dielectric function to the data for platinum and aluminum. We obtained good agreement between the model and the experimental results.

[1] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Science **220**, 671 (1983).
[2] N. Metropolis *et al.*, J. Chem. Phys. **21**, 1087 (1953).
[3] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines* (Wiley, New York, 1989).
[4] S. Geman and D. Geman, IEEE Trans. Pattern Anal. Machine Intell. **PAMI-6**, 721 (1984).
[5] R. A. Rutenbar, IEEE Circuits Dev. Mag. **5**, 19 (1989).
[6] M. M. Doria, J. E. Gubernatis, and D. Rainer, Phys. Rev. B **41**, 6335 (1990).
[7] F. Hsu, P.-R. Chang, and K.-K. Chan, IEEE Trans. Antennas Propagat. **41**, 1195 (1993).
[8] F. Catthoor, H. de Man, and J. Vandewalle, VLSI J. **6**, 147 (1988).
[9] G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, IEEE J. Solid-State Circuits **25**, 707 (1990).
[10] H. H. Szu and R. L. Hartley, Phys. Lett. A **122**, 157 (1987).
[11] H. H. Szu and R. L. Hartley, Proc. IEEE **75**, 1538 (1987).
[12] A. D. Rakić, J. M. Elazar, and A. B. Djurišić, Phys. Rev. E **52**, 6862 (1995).
[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran*, 2nd ed. (Cambridge University Press, Cambridge, 1992).
[14] J. Wu, T. Kondo, and R. Ito, Jpn. J. Appl. Phys. **32**, 5576 (1993).
[15] A. Corana, M. Marchesi, C. Martini, and S. Ridella, ACM T. Math. Software **13**, 262 (1987).
[16] H. Ehrenreich and H. R. Philipp, Phys. Rev. **128**, 1622 (1962).
[17] H. Ehrenreich, H. R. Philipp, and B. Segall, Phys. Rev. **132**, 1918 (1963).
[18] C. P. Chang and T. H. Lee, Opt. Lett. **15**, 595 (1990).
[19] S. R. White, in *Proceedings of the IEEE International Conference on Computer Design* (IEEE Computer Society, New York, 1984), pp. 646–651.
[20] L. Pronzato, E. Walter, A. Venot, and J.-F. Lebruchec, Math. Comput. Simul. **26**, 412 (1984).
[21] F. Aluffi-Pentini, V. Parisi, and F. Zirilli, J. Optim. Theory Appl. **47**, 1 (1985).
[22] A. Dekkers and E. Aarts, Math. Progr. **50**, 367 (1991).
[23] D. W. Lynch and W. R. Hunter, in *Handbook of Optical Con-*

stants of Solids, edited by E. D. Palik (Academic, Orlando, FL, 1985), pp. 275–367.
[24] J. H. Weaver, Phys. Rev. B **11**, 1416 (1975).
[25] A. Y.-C. Yu, W. E. Spicer, and G. Hass, Phys. Rev. **171**, 834 (1968).
[26] A. Seignac and S. Robin, Solid State Commun. **11**, 217 (1972).
[27] G. Hass and W. R. Hunter, in *Space Optics*, edited by B. J. Thompson and R. R. Shanon (National Academy of Sciences, Washington, D.C., 1974), pp. 525–553.
[28] R. Haensel, K. Radler, B. Sonntag, and C. Kunz, Solid State Commun. **7**, 1495 (1969).
[29] N. V. Smith, Phys. Rev. B **9**, 1365 (1974).
[30] M. A. Ordal *et al.*, Appl. Opt. **24**, 4493 (1985).
[31] M. I. Marković and A. D. Rakić, Appl. Opt. **29**, 3479 (1990).
[32] M. I. Marković and A. D. Rakić, Opt. Laser Technol. **22**, 394 (1990).
[33] A. D. Rakić, Appl. Opt. **34**, 4755 (1995).
[34] C. J. Powell, J. Opt. Soc. Am. **60**, 78 (1970).