

# Global Optimization Algorithms I Simulated Annealing Algorithm - Application to Calculation of Optical Constants of Al

Aleksandra B. Đurišić

*Faculty of Electrical Engineering, University of Belgrade,  
P.O. Box 816, Belgrade, Yugoslavia*

The detailed description of simulated annealing algorithm is given. Four variations of the original algorithm are presented and their performance was tested on two families of multim minima test functions for up to 100 variables. The application of simulated annealing algorithm for determination of model parameters of optical constants of aluminum is presented.

## I Introduction

Simulated annealing (SA) algorithms represent a class of methods for solving combinatorial optimization problems, based on the analogy to the physical process of annealing. Simulated annealing algorithm was introduced by Kirkpatrick *et al* [1]. Since then, considerable number of papers on the subject were published, dealing either with proof of convergence of the SA technique or proposing the implementation of various modified SA algorithm to solving certain real-world problems like chip floorplaning, wiring, optimum design of electronic devices, neural networks learning process etc.

To understand the analogy of SA with physical process of annealing, let us consider how low energy state of a solid can be achieved. The low energy state is usually highly ordered state such as defect-free crystal lattice. To accomplish this, annealing of the solid is performed: it is heated to high temperature permitting many atomic rearrangements and then slowly cooled until it is frozen into a good crystal. If a function to be minimized, called objective or cost function, is analogous to the energy of a solid, it is reasonable to expect that by simulating slow cooling from high temperatures poor unordered solution transforms to desired, highly optimized solution.

Simulated annealing has its origins in the work of Metropolis *et al* [2], who introduced an algorithm for efficient simulation of a collection of atoms in equilibrium at given temperature. Starting from an arbitrary initial state, the algorithm generates a sequence of changes of variable values termed "moves". If the move results in the decrease of objective function, *i.e.*  $\Delta E < 0$  new state is accepted as initial state for the next move. However, if  $\Delta E > 0$  there is still a non zero probability of accepting

that move, which is a function of  $\Delta E$  and the control parameter  $T$  termed temperature. Probabilistic hill climbing capability of the SA algorithm enables escaping local minima. Let us illustrate the importance of this feature on the simple example, as shown on Fig. 1. If a conventional downhill

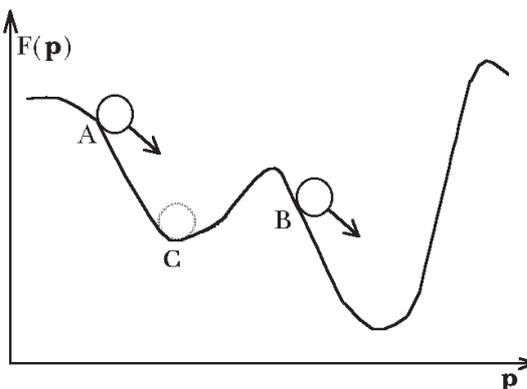


FIG. 1. Illustration of the global optimization problem.

optimization method starts from initial state A, it will end up in local minimum C, failing to find global minimum, which would be located from initial state B. In the case of SA, it does not have necessarily to be so, owing to hill climbing capability of the algorithm. If the initial temperature is high enough, uphill moves that enable evading of local minimum C could be accepted. By decreasing the temperature after equilibrium is reached at each current temperature, probability of accepting an uphill move is decreased.

Since its introduction in 1983, SA has diffused widely into many diverse applications, like integrated circuits design, [1, 3, 4], solving traveling salesman problem [1, 5, 6, 7], conformal optimization of macromolecules [8], model parameter estimation [9, 10, 11, 12, 13, 14, 15], restoration of images [16], neural networks [17] etc. SA has several very attractive features, making it a successful tool for various applications. As first, it is not "greedy" - in other words, it is not easily fooled by the quick payoff achieved by falling into unfavourable minima. It wanders freely among local minima of depth less than about  $T$ . As temperature  $T$  is reduced, the number of minima qualifying for frequent visits is gradually decreased. As second, configuration decisions are in logical order, *i.e.* changes which cause greatest energy differences are possible only at the beginning of the process, when temperature has large values. However, in a number of applications it is imperative to carefully adjust SA algorithm's features like cooling schedule, equilibrium criterion, move-generation procedure etc., to obtain global minimum. Convergence of the SA algorithm was investigated for different cooling schedules and it was proved that algorithm asymptotically converges to global minimum with probability one [16, 18, 19, 20, 21]. Unfortunately, there is no practical guarantee that any of the existing modifications of SA algorithm can solve certain practical problems, so that some changes often have to be introduced in order to improve performance of the algorithm and obtain satisfactory results.

In the following section essentials of SA algorithm are given, as well as brief descriptions of some significant variations in cooling schedule, acceptance probability and move generation procedure. In section III comparison of performance of 4 variations of SA method is given for two families of multiminima test functions, while in section IV application of the acceptance-probability-controlled SA algorithm for determination of model parameters of optical constants of aluminum is demonstrated.

## II Description of the algorithm

Simulated annealing is a procedure that iteratively changes a state of the optimization problem. Flowchart of the algorithm is given on Figure 2. Moves are chosen using a state generation procedure. The decision whether or not to actually make a move is made by acceptance criterion.

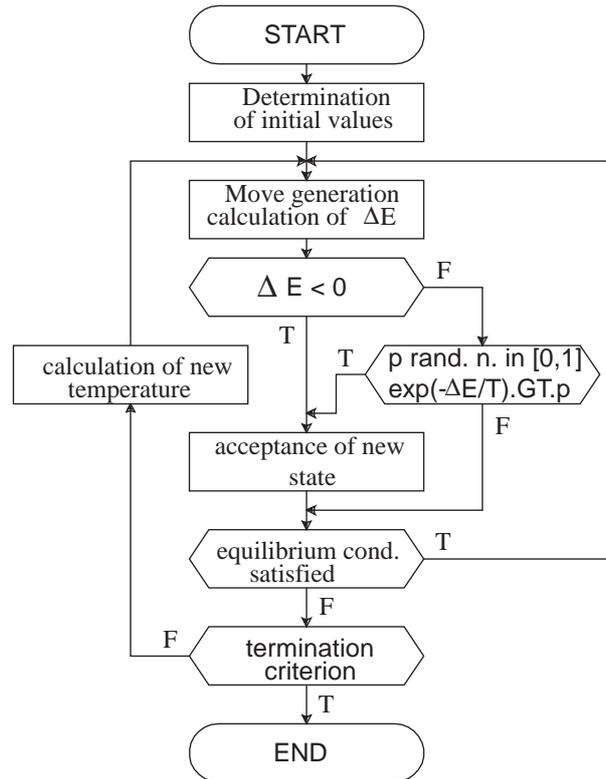


FIG. 2. Flowchart of the SA algorithm.

The algorithm has two nested loops. In the inner one, Metropolis algorithm at fixed temperature is performed until the equilibrium state is reached, while in the outer one update of the temperature is executed. The algorithm is determined by: initial temperature setting procedure, equilibrium condition (inner loop stopping criterion), termination criterion, acceptance criterion, move generation procedure and cooling schedule. Details are given as follows.

## II.1 Initial temperature

Initial temperature should be high enough to allow exploration of the whole objective function domain. It is usually set to arbitrary value, considered high enough. However, this approach often leads to larger computational time, due to unnecessarily spent time at too high temperatures. Therefore, in algorithms investigated here, procedure of Rees and Ball [7] was employed. Making many random changes to the objective function  $E$ , average objective function  $\langle E \rangle_\infty$  and the average of the absolute change  $\langle |\Delta E| \rangle_\infty$  is calculated, which correspond to infinite temperature when all moves would be accepted. The initial temperature, corresponding to initial acceptance probability AP (under the hypothesis that AP is given by Boltzmann distribution)  $\pi^{init} = 0.90$  is given with

$$T^{init} = -\frac{\langle |\Delta E| \rangle_\infty}{\ln(\pi^{init})} \approx 10 \langle |\Delta E| \rangle_\infty \quad (1)$$

## II.2 Inner loop stopping criterion

In the plain SA, introduced by Kirkpatrick *et al* [1], inner loop was terminated if number of accepted moves equaled ten times number of variables, or if number of moves equaled hundred times number of variables. This criterion is still frequently used in a number of practical applications, although it is far from optimal one. In other words, equilibrium is often reached after fewer moves, so that algorithm

spends to much time at some temperatures. Wasting time and effort can be prevented if inner loop stopping criterion is connected with the convergence of the entity  $D$ , as suggested in [6], given by

$$D = \frac{1}{n} \sum_{i=1}^n \exp [(\langle E \rangle - E(\mathbf{p}_i^{\text{acc}}))/T_m] \quad (2)$$

where summation is performed over accepted states  $\mathbf{p}_i^{\text{acc}}$  at the temperature  $T_m$ .  $\langle E \rangle$  is the average objective function at the preceding temperature  $T_{m-1}$  given by

$$\langle E \rangle = \frac{1}{N_{\text{acc}}} \sum_{i=1}^{N_{\text{acc}}} E(\mathbf{p}_i^{\text{acc}}) \quad (3)$$

where  $N_{\text{acc}}$  is the number of accepted states at  $T_{m-1}$ , while  $E(\mathbf{p}_i^{\text{acc}})$  is the objective function corresponding to accepted states  $\mathbf{p}_i^{\text{acc}}$ . Equilibrium is achieved when the absolute value between two consecutive values of  $D$  is less than specified number  $\delta$ ,  $|D_n - D_{n-1}|/D_{n-1} < \delta$ . However, it should be pointed out that there have been reported non-equilibrium SA algorithms, where temperature is reduced even though equilibrium is not reached [5].

### II.3 Termination criterion

Outer loop is commonly terminated if specified number of iterations is reached, or if next temperature is less than specified final temperature  $T_0$ . In this paper, algorithms employing slightly modified solidification criterion of Doria *et al* [22] were investigated. At each temperature the lowest obtained value of the objective function is recorded. When the absolute values of the relative difference between the current minimal objective function and three preceding ones were within  $\epsilon$  of each other, the simulation was stopped.

### II.4 Acceptance criterion

Probability of accepting an uphill move is usually given with Boltzmann distribution  $\pi = \min(1, \exp(-\Delta E/T))$ , where  $\Delta E$  is the change in the cost function, and  $T$  is temperature. Recently, a number of different acceptance criteria and corresponding cooling schedules have been proposed for improving the performance of the SA algorithm. Several of them will be described here. Most proposed AP slightly differ in behavior from Boltzmann's, like AP with factor  $\beta$  *i.e.*  $\pi = \min(1, \exp(-\Delta E/\beta T))$  [27], where  $\beta$  is a constant that relates temperature to objective function value, or  $\pi = \min(1, (1 + \exp(-\Delta E/T))^{-1})$  [28]. But, there are also AP that are quite different from Boltzmann's, like Glauber AP [5], where AP for both uphill and downhill moves is given by

$$\pi = \frac{\exp(-\Delta E/k_B T)}{1 + \exp(-\Delta E/k_B T)} \quad (4)$$

where  $k_B$  is a given constant. It can be observed that at high temperatures all moves will be accepted with probability 0.5. With decrease of the temperature, AP of downhill moves approaches 1, while AP for uphill moves approaches 0. Recently, various AP have been proposed which are based on Tsallis statistics, built from general entropy. For instance, in [12] was proposed generalized AP given by

$$\pi(s_{t_s}) = \min(1, (1 - (1 - q_a)\Delta\epsilon^2/T(t_s))^{1/(1-q_a)}) \quad (5)$$

where  $\Delta\epsilon^2$  is the difference of the sum over squared deviations achieved with parameter set  $s_{t_s}$  at time  $t_s$ . The parameter  $q_a \in [-10, 1)$  enables avoiding numerical instability. More complicated variant, also originated from Tsallis statistics [8] is given with

$$P = \min \left[ 1, \left( \frac{1 - (1 - q(T))\beta E_{new}}{1 - (1 - q(T))\beta E_{old}} \right)^{q(T)/(1-q(T))} \right] \quad (6)$$

and

$$\lim_{T \rightarrow 0} q(T) = 1 \quad (7)$$

where  $q(T)$  is monotonically decreasing function of temperature. In [8] is also stated that form of AP which is widely used in simulations of spin glasses systems is given with  $p = 1/2[1 - \tanh(\beta\Delta E/2)]$ , which has the same behavior as previous expression. However, since different AP does not solve certain problems of SA algorithms, which are described in detail in the part discussing cooling schedule, in algorithms investigated in this paper Boltzmann's AP was employed.

## II.5 Move-generation procedure

The domain  $P$  containing the parameter vector  $\mathbf{p} = (p(1), p(2), \dots, p(N))$  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 configuration depends largely on two elements: a) number of variables to be changed in one move, and b) move-step adjustment.

As it was mentioned above, in the optimization problems with the large number of variables, moves which require the change in all variables (here, parameters of the model) can cause the instabilities in the solution. The number of variables to be changed in one move is often reduced, and sometimes is chosen randomly [9, 24, 25], or even reduced to only one variable per move, as in the Ref. [23]. However, the random state-generation procedure is far from optimal. In this paper is demonstrated that convergence of the algorithm is accelerated by taking into account the sensitivity of the objective function with respect to the change of individual variables. The algorithm makes the probability of taking the move along the certain coordinate direction proportional to the sensitivity of the cost function with respect to that variable. This improves the mobility of the system which now shows preference for the steeper slopes in either uphill or downhill direction. Therefore, this generator shows strong bias towards the moves that cause the greatest energy difference. Let us describe in detail adaptive move-generation procedure. At each temperature, for each parameter  $p(k)$ , we determine the average of the absolute change in cost function  $\langle |\Delta E|_k \rangle$  by making a lots of random moves for parameter  $p(k)$ , keeping other parameters fixed. Then, for each  $k$ ,  $k = 1, \dots, N$  we compute the frequency of change  $f(k)$  (corresponding to the parameter  $p(k)$ ) using:

$$f(k) = 0.8 \frac{\langle |\Delta E|_k \rangle}{\langle |\Delta E|_k \rangle_{\max}} \quad (8)$$

where  $\langle |\Delta E|_k \rangle_{\max} = \max(\langle |\Delta E|_k \rangle, k = 1, \dots, N)$ . If the frequency  $f(k)$  of changing the parameter  $p(k)$  is greater than randomly generated number  $p_{ch} \in [0, 1]$ , parameter  $p_i(k)$  is altered to  $p_j(k)$ ,

$$p_j(k) := p_i(k) + r \Delta(k), \quad (9)$$

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 new state  $p_j(k)$  is outside the specified boundaries  $p_l(k)$  and  $p_u(k)$ ,  $p_j(k)$  is assigned a value of the nearest boundary.

The impact of the step size on the quality of the solution was frequently addressed. Corana *et al.*[23] changed the components of the move-step vector adaptively in order to maintain the acceptance ratio close to 0.5, at all temperatures. Most of the authors consider it important to decrease the move-step during the annealing in order to reduce the fluctuations in the final stage [6, 28, 26, 29]. The initial

step size has to be comparatively large to provide the sufficient mobility of the algorithm to cover the entire parameter space. Therefore, there is a direct connection between temperature scales and move scales. At high temperatures most large moves and essentially all small moves are accepted. This suggests a strategy for adapting the move set being used to the temperature scale. It can be done, as suggested in [29] if selected move classes satisfy  $\langle |\Delta E| \rangle \approx T$  to within standard distribution of  $|\Delta E|$  and  $\langle |\Delta E| \rangle < [\langle E(T) \rangle^2 - \langle E^2(T) \rangle]^{1/2}$ . However, this criterion haven't find many practical applications. On the contrary, combination of Cauchy-Lorentz move distribution  $G(x) = T(t)/(T(t)^2 + x^2)$  coupled with fast annealing schedule, as proposed recently in [28], have become widely applied. However, since temperature usually depends on time, *i.e.* number of executed outer loops, move-step size can be in a simple manner directly connected with time, as suggested by Catthoor *et al* [6]. Therefore, here is adopted the suggestion of Catthoor *et al.* and reduced the step size in "nearly inverse quadratic" manner. When the ratio  $\Delta(k)/p(k)$  is less than 0.005 further reduction of move-step for that parameter is stopped.

## II.6 Cooling schedule

The plan for changing the temperature with time is termed cooling schedule. The standard, exponential cooling schedule  $T_m = \alpha T_{m-1}$  is widely accepted [1, 3, 11, 12, 13, 22, 24, 30], where  $\alpha$  has fixed value between 0.9-0.98. Geman and Geman [16] proved that for high enough initial temperature global minimum is obtained asymptotically, if cooling schedule is not faster than logarithmic one,  $T(t) = T^{init}/\ln(1+t)$ . Logarithmic cooling schedule was compared to exponential and decrement ( $T(m) = T(m-1) - \gamma$ ) one, and employed in various applications [7, 32, 33]. Szu and Hartley introduced "fast simulated annealing" [28], with cooling schedule

$$T(t) = \frac{T(t-1)}{1+t} \quad (10)$$

This cooling schedule is usually applied together with Cauchy-Lorentz distribution in move generation [27, 28]. For different AP described above, corresponding cooling schedules have been devised. For instance, to Glauber AP corresponds the following cooling schedule [5]

$$T_{new} = \frac{T_{old}}{1 + \frac{T_{old}(1+\delta)}{3\sigma}} \quad (11)$$

where  $\delta$  controls cooling rate, while  $\sigma$  is standard deviation of all configurations at current temperature  $T_{old}$ . In [12] parameter distribution probabilities are quenched according to individual parameter annealing temperatures  $T_s$ . Naturally, this is highly impractical for large number of parameters. There are even SA algorithms that do not have analytical cooling schedule. For instance, in [30] annealing schedule is represented by  $\{(T_0, n_0), (T_1, n_1), \dots, (T_m, n_m)\}$  - a list of decreasing temperatures  $T_i$  and the required number of moves  $n_i$  to reach equilibrium at temperature  $T_i$ .

All of the above algorithms have two major problems. First, these schedules imply only the reduction of temperature with time. For instance, in metallurgical annealing it is not unusual to inspect the structure of the system during the annealing process in order to detect the polycrystalline state, and to increase temperature to remelt it. Kirkpatrick *et al* [1] suggested that quantity analogous to specific heat should be introduced to indicate that freezing has begun and hence that very slow cooling is required. Application of such criterion would enable one to determine when parameter controlling cooling rate  $\alpha$  should be increased. In such manner, annealing can start with smaller values of  $\alpha$ , *i.e.* faster cooling rate, thus reducing the necessary computational time. However, computing of effective specific heat also requires time and memory. There are cooling schedules that enable switching between  $\alpha_{min}$  and  $\alpha_{max}$  [29], or gradually changing value of  $\alpha$  towards  $\alpha_{min}$  or  $\alpha_{max}$ , depending on the demands of the system [4, 6].

Still, these cooling schedules also employ only cooling, though with variable cooling rate. Nevertheless, need for the occasional heating of the system was recognized. For example, adaptive SA proposed in [31] employs both increasing and decreasing of temperature, depending on the ratio of uphill and downhill moves. Matsuba [17] varied temperature according to decrement in the objective function  $E$ , thus allowing increases of the temperature when needed. However, proposed solution required substantial analytical effort even for the simplest one-dimensional problem, so it is obvious that could be of no use for real-size problems.

All previously described cooling schedule, although they sometimes can cope with the first problem, do not solve the second one. The second problem is related to the traditional choice of temperature as control parameter. The temperature is varied in one way or another during the simulation in order to reduce AP of uphill moves, thus confining the system in vicinity of the global minimum. However, it is evident that really important parameter is not the temperature itself, but the *acceptance probability*. Let us describe the concept of acceptance-probability-controlled simulated annealing (APCSA).

In each outer loop acceptance probability is lowered according to the cooling schedule [9, 10]. Acceptance probabilities depending on the outer loop counter  $M$  are given by the normal distribution:

$$\pi_M = \pi^{\text{init}} \exp(-M^2/2\sigma^2). \quad (12)$$

The temperature  $T_M$  is then determined as

$$T_M = - \frac{\langle |\Delta E|_{\text{acc}} \rangle}{\ln(\pi_M)}. \quad (13)$$

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

It is evident that this cooling schedule increases the temperature when the values of  $\langle |\Delta E|_{\text{acc}} \rangle$  are significant. Also, it should be noted that memory of connection between  $T$  and  $\langle |\Delta E|_{\text{acc}} \rangle$  established in the initial temperature determination process can be easily lost with standard temperature controlled cooling schedules, especially in regions where objective function has valleys with steep walls or deep pits. In the APCSA algorithm, correspondence between temperature and the average of absolute change in the objective function is reestablished at the end of each inner loop. This feature is crucial, since AP depends not only on temperature, but also on the change of the objective function. Therefore, this feature, together with the ability of occasional rises of the temperature, enables more efficient escaping local minima and faster convergence of APCSA algorithm compared to the standard SA.

In the following section the APCSA algorithm with adaptive move generation procedure (APCSA1) was severely tested and compared its performances with the performances of APCSA algorithm with random move generation procedure (APCSA2) of Hsu *et al.* [24], and to the SA algorithm proposed by Catthoor *et al.* [6] (CSA) with both move generation procedures stated above (CSA1, CSA2).

### III Tests and results

Two families of multim minima test functions were employed. The first family of multim minima functions is the function used by Aluffi-Pentini *et al.* [34] and Dekkers *et al.* [19].

$$g(\mathbf{x}) = \frac{\pi}{n} [k_1 \sin^2 \pi y_1 + \sum_{i=1}^n (y_i - k_2)^2 \cdot (1 + k_1 \sin^2 \pi y_{i+1}) + (y_n - k_2)^2] \quad (14)$$

where  $y_i = 1 + 0.25(x_i + 1)$ ,  $k_1 = 10$ ,  $k_2 = 1$ , and  $x_i \in [-10, 10]$ ,  $i = 1, n$ . This function has roughly  $5^n$  local minima. In cited references, this function was tested for three variables. We performed tests with 20, 50 and 100 variables. Function  $g$  of two variables is shown on Fig. 3. Obtained results are presented on Fig. 4, Fig. 5 and Fig. 6, respectively.

The second family of multim minima functions, also investigated by Aluffi-Pentini *et al.* [34] and Dekkers and Aarts [19], is given by

$$h(\mathbf{x}) = k_3 \left\{ \sin^2(\pi k_4 x_1) + \sum_{i=1}^{n-1} (x_i - k_5)^2 [1 + k_6 \sin^2(\pi k_4 x_{i+1})] + (x_n - k_5)^2 [1 + k_6 \sin^2(\pi k_7 x_n)] \right\} \quad (15)$$

where  $k_3 = 0.1$ ,  $k_4 = 3$ ,  $k_5 = 1$ ,  $k_6 = 1$  and  $k_7 = 2$ . In [19] this function, whose two-dimensional segment is shown on Fig 7, was investigated for five variables  $x_i \in [-5, 5]$ ,  $i = 1..5$ , *i.e.* in the area where there are roughly  $15^5$  minima. In this paper, family of test functions  $h(\mathbf{x})$  was investigated for 20, 50 and 100 variables  $x_i \in [-10, 10]$ ,  $i = 1, n$ , and obtained results are shown on Fig. 8, Fig. 9 and Fig. 10, respectively.

It can be observed that APCSA algorithm with the adaptive move generation procedure in all cases achieves the lowest cost function value. What's more, new adaptive move generation procedure not only improves performances of the APCSA algorithm, but also significantly ameliorates CSA algorithm so that APCSA2 algorithm and CSA1 algorithm have similar performances, but results of both algorithms depend on the initial values. In some cases CSA1 algorithm with new move generation procedure can obtain the same order of magnitude of the cost function as APCSA2. CSA2 - algorithm with random move generation procedure of Hsu *et al.* [24] in all cases shows the worst performances, *i.e.* fails to locate global minimum, and achieves the highest cost function value, far from near-optimal one.

## IV Application to optical constants of aluminum

For fitting the optical constants of aluminum, algorithm with best performances APCSA was used. The both Drude [35, 36] and the Lorentz-Drude model [37, 38] were often employed for the parametrization of the optical constants of aluminum. According to the Lorentz-Drude model, the dielectric permittivity function is described with

$$\hat{\epsilon}_\infty(\omega) = \hat{\epsilon}_\infty^{(f)}(\omega) + \hat{\epsilon}_\infty^{(b)}(\omega). \quad (16)$$

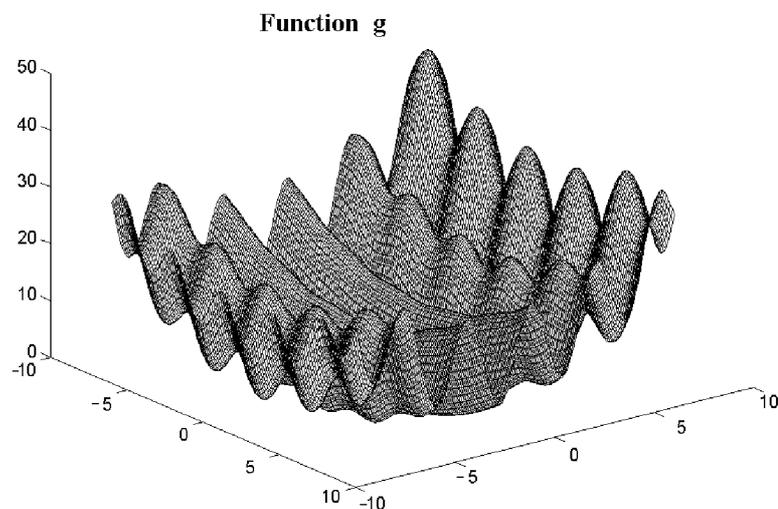


FIG. 3. Multim minima test function  $g$  of two variables.

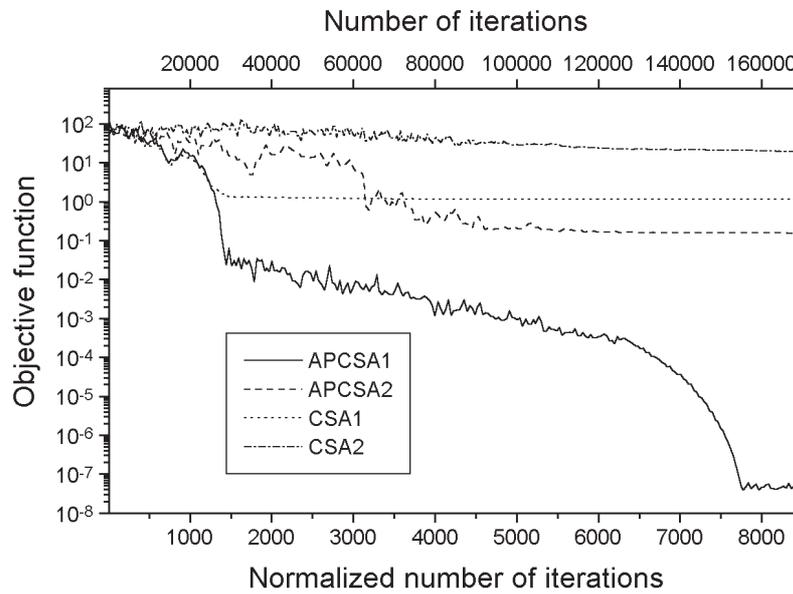


FIG. 4. Cost function vs. normalized number of iterations (bottom axis) and number of iterations (top axis) for the function  $g$  of  $x_i \in [-10, 10], i = 1, 20$ , with initial values  $(-5, 5, 0, -5, 5) \times 5$ .

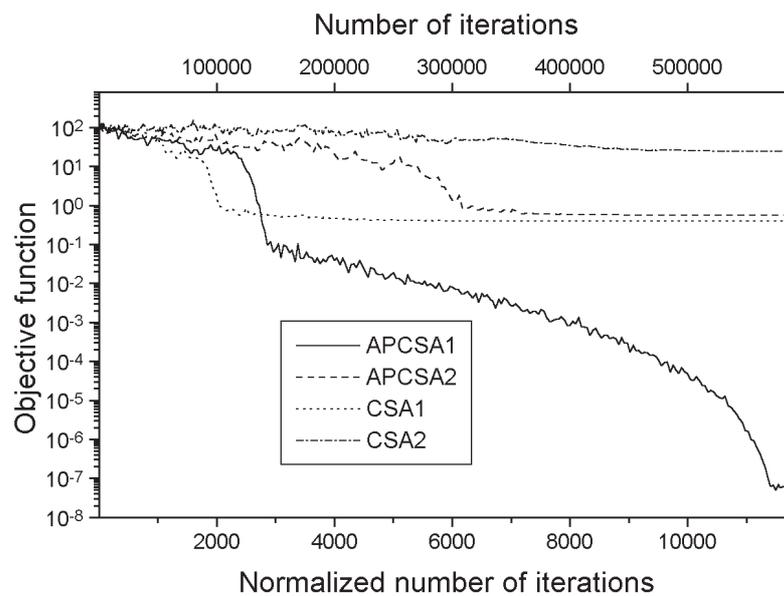


FIG. 5. Cost function vs. normalized number of iterations (bottom axis) and number of iterations (top axis) for the function  $g$  of  $x_i \in [-10, 10], i = 1, 50$ , with initial values  $(-5, 5, 0, -5, 5) \times 10$ .

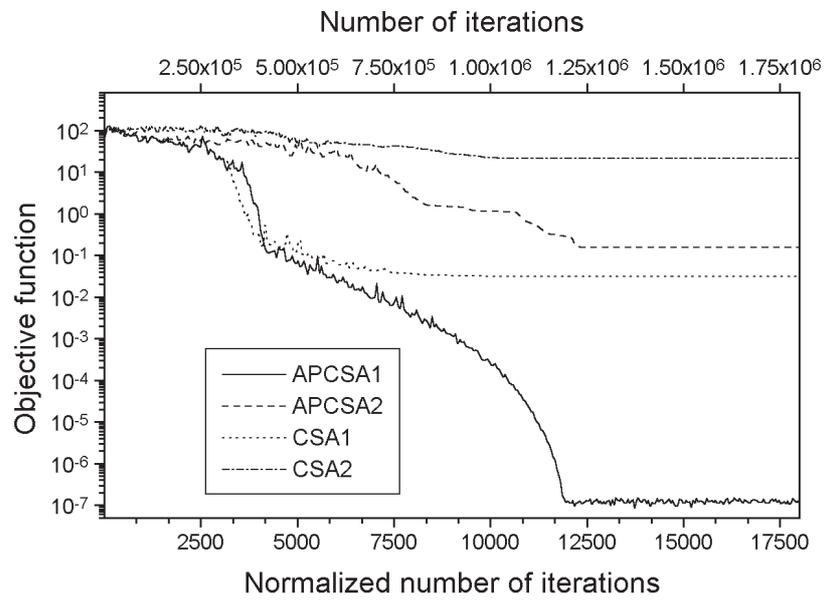


FIG. 6. Cost function vs. normalized number of iterations (bottom axis) and number of iterations (top axis) for the function  $g$  of  $x_i \in [-10, 10]$ ,  $i = 1, 100$ , with initial values  $(-5, 5, 0, -5, 5) \times 20$ .

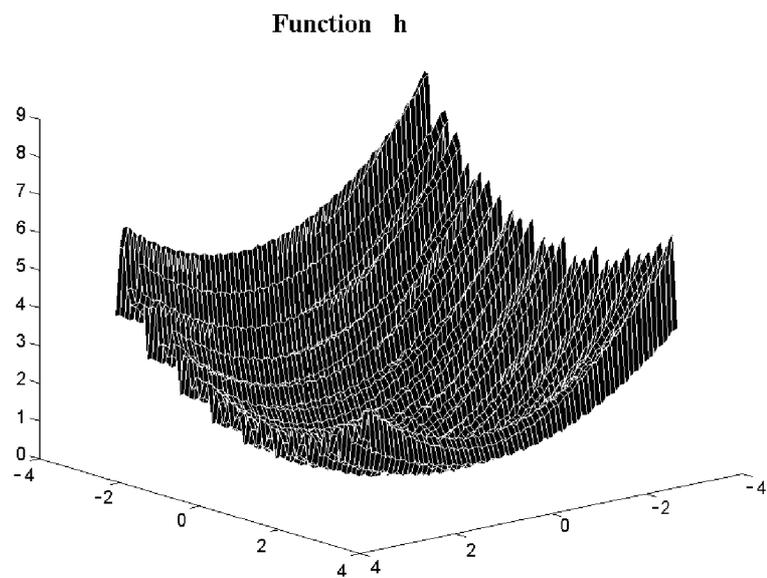


FIG. 7. Multim minima test function  $h$  of two variables.

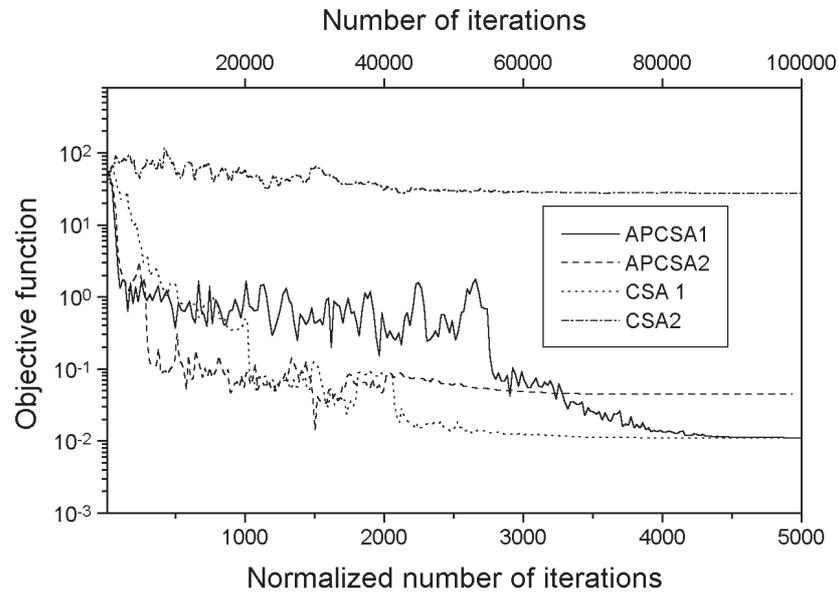


FIG. 8. Cost function vs. normalized number of iterations (bottom axis) and number of iterations (top axis) for the function  $h$  of  $x_i \in [-10, 10], i = 1, 20$ , with initial values  $(-5, 5, 0, -5, 5) \times 5$ .

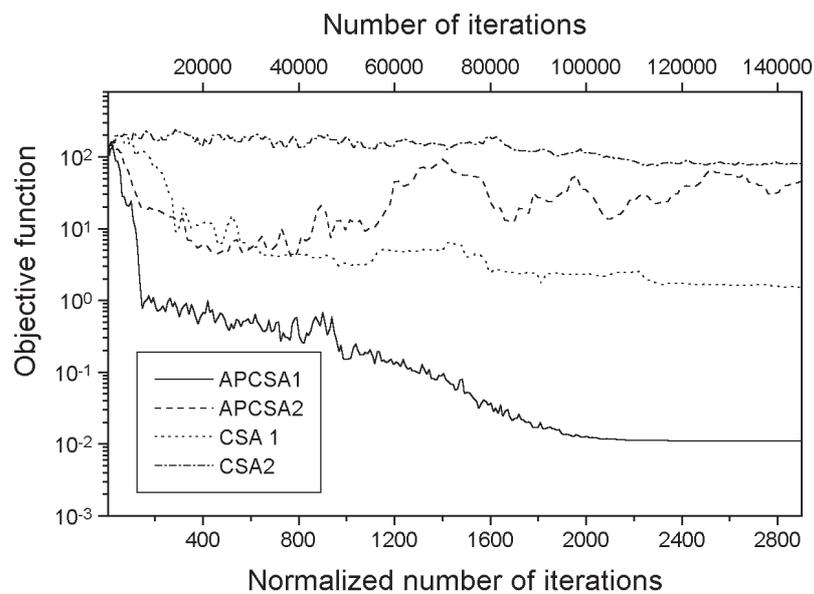


FIG. 9. Cost function vs. normalized number of iterations (bottom axis) and number of iterations (top axis) for the function  $h$  of  $x_i \in [-10, 10], i = 1, 50$ , with initial values  $(-5, 5, 0, -5, 5) \times 10$ .

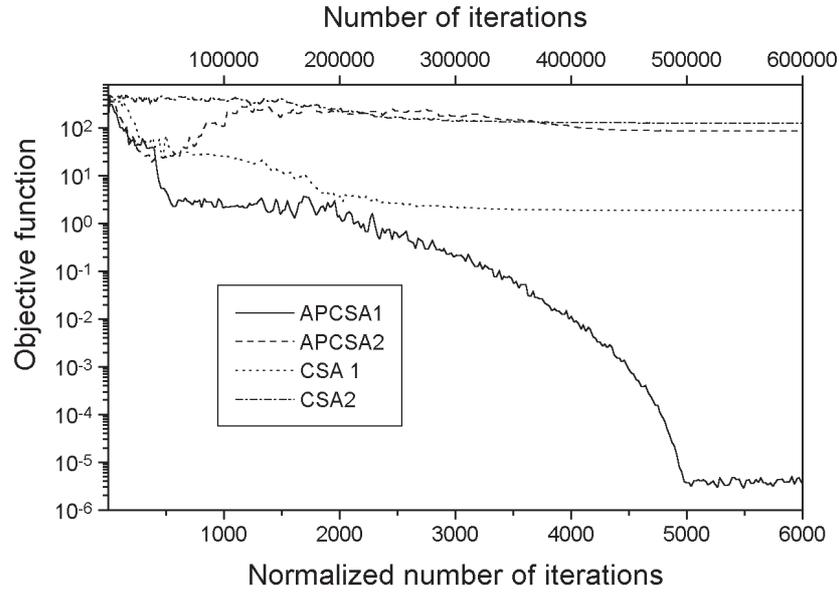


FIG. 10. Cost function vs. normalized number of iterations (bottom axis) and number of iterations (top axis) for the function  $h$  of  $x_i \in [-10, 10]$ ,  $i = 1, 100$ , with initial values  $(-5, 5, 0, -5, 5) \times 20$ .

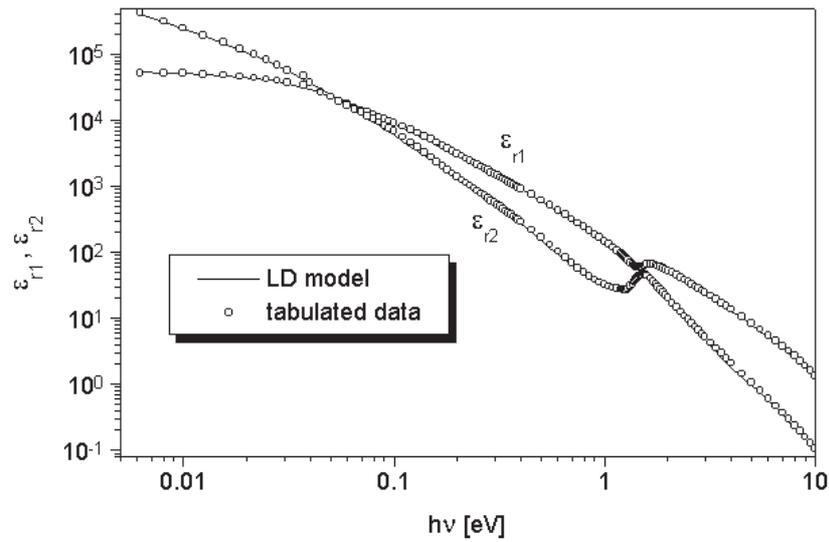


FIG. 11. Aluminum: comparison of the tabulated dielectric function (from Ref. [37] - open circles) and model dielectric function calculated in this study (solid line).

In the above equation, contributions of intraband transitions (free electron effects) are separated from interband transitions (bound electron effects), as shown in [39, 40]. The intraband part  $\hat{\epsilon}_r^{(f)}(\omega)$  of dielectric function is the well known free electron or Drude model

$$\hat{\epsilon}_r^{(f)}(\omega) = 1 - \frac{\Omega_p^2}{\omega(\omega + i, 0)}, \quad (17)$$

while the interband part of the dielectric function  $\hat{\epsilon}_r^{(b)}(\omega)$  is given with the Lorentz equation for insulators

$$\hat{\epsilon}_r^{(b)}(\omega) = - \sum_{j=1}^k \frac{f_j \omega_p^2}{(\omega^2 - \omega_j^2) + i\omega, j}, \quad (18)$$

where  $\omega_p$  is the plasma frequency,  $k$  is the number of interband transitions with frequency  $\omega_j$ , 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 in intraband processes, and  $\gamma_0$  is the intraband damping constant.

The following objective function was used for the model parameter estimation:

$$E(\mathbf{p}) = \sum_{i=1}^{i=N} \left[ \left| \frac{\epsilon_{r1}(\omega_i) - \epsilon_{r1}^{\text{exp}}(\omega_i)}{\epsilon_{r1}^{\text{exp}}(\omega_i)} \right| + \left| \frac{\epsilon_{r2}(\omega_i) - \epsilon_{r2}^{\text{exp}}(\omega_i)}{\epsilon_{r2}^{\text{exp}}(\omega_i)} \right| \right]^2. \quad (19)$$

The tabulated intrinsic optical constants of aluminum from the recent study of Rakić[37] were employed for fitting. Interband transitions are expected at about 0.4eV, 1.5eV, 2.1eV and 4.5eV. Final parameter values are presented in Table 1. The values of the oscillator strengths correspond to the plasma frequency  $\hbar\omega_p = 14.98$  eV [37]. Figure 11 shows excellent agreement between tabulated (open circuits) and model (solid line) dielectric function of aluminum.

TABLE 1. Parameter values for aluminum.

j	0	1	2	3	4
$f_j$	0.498	0.248	0.045	0.196	0.010
$\gamma_j$	0.044	0.304	0.288	1.502	2.794
$\omega_j$	0	0.133	1.546	1.802	5.707

## V Conclusion

Simulated annealing has been widely used technique for global optimization. A number of modifications of the algorithm have been devised, to improve the convergence and ability of escaping local minima. Some of the significant modifications have been described. To illustrate the effectiveness of SA approach in finding the global minimum, four different SA algorithms were tested on two families of multim minima test functions. After finding out which algorithm shows the best performance, its application to determination of the model parameters of optical constants of aluminum was demonstrated. Obtained excellent agreement between experimental and calculated data once again proves the ability of chosen algorithm to locate the global minimum of the objective function, which in this case was the sum of squared relative differences between experimental and calculated data.

## References

- [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] R. A. Rutenbar, *IEEE Circuits Dev. Mag.* **5**, 19 (1989).
- [4] F. Romeo, A. S. Vincentelli and C. Sechen, *Proc. IEEE Internat. Confer. Computer Design*, (1984).
- [5] M. F. Cardoso, R. L. Salcedo and S. F. Azeredo, *Ind. Eng. Chem. Res.*, **33**, 1908, (1994).
- [6] F. Catthoor, H. de Man, and J. Vandewalle, *INTEGRATION, the VLSI journal* **6**, 147 (1988).
- [7] S. Rees and R. C. Ball, *J. Phys. A: Math. Gen.*, **20**, 1239, (1989).
- [8] I. Andricioaei and J. E. Straub, *Phys. Rev. E*, **53**, R3055, (1996).
- [9] A. D. Rakić, J. M. Elazar, and A. B. Djurišić, *Phys. Rev. E* **52**, 6862 (1995).
- [10] A. B. Djurišić, A. D. Rakić and J. M. Elazar, *Phys. Rev. E* **55**, 4797 (1997).
- [11] M. K. Vai, D. Ng and S. Prasad, *Elec. Lett.*, **26**, 892, (1990).
- [12] J. Schulte, *Phys. Rev. E*, **53**, R1348, (1996).
- [13] M. K. Vai, S. Prasad, N. C. Li and F. Kai, *IEEE Trans. on Electron Devices*, **36**, 761, (1989).
- [14] R. Hara, T. Iwamoto and K. Kyuma, *J. Appl. Phys.*, **32**, L68, (1993).
- [15] T. H. Tan and W.-W. Chag, *Elec. Lett.*, **32**, 1256, (1996).
- [16] S. Geman and D. Geman, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-6**, 721 (1984).
- [17] I. Matsuba, *Phys. Rev. A* **39**, 2635 (1989).
- [18] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines* (John Wiley, New York, 1989).
- [19] A. Dekkers and E. Aarts, *Mathematical programming*, **50**, 367 (1991).
- [20] V. Granville, M. Krivanek and J.-P. Rason, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **16**, 652, (1994).
- [21] U. Faigle and W. Kern, *SIAM J. Control and Optimization*, **29**, 153, (1991).
- [22] M. M. Doria, J. E. Gubernatis, and D. Rainer, *Phys. Rev. B* **41**, 6335 (1990).
- [23] A. Corana, M. Marchesi, C. Martini, and S. Ridella, *ACM T. Math. Software* **13**, 262 (1987).
- [24] F. Hsu, P.-R. Chang, and K.-K. Chan, *IEEE Trans. Antennas Propagat.* **41**, 1195 (1993).
- [25] J. Wu, T. Kondo, and R. Ito, *Jpn. J. Appl. Phys.* **32**, 5576 (1993).
- [26] C. P. Chang and T. H. Lee, *Opt. Lett.* **15**, 595 (1990).
- [27] C. S. Koh, J. A. Mohammed and S.-Y. Haha, *IEEE Trans. on Magnetics*, **30**, 3644, (1994).
- [28] H. H. Szu and R. L. Hartley, *Phys. Lett. A* **122**, 157 (1987).
- [29] S. R. White, in *Proc. IEEE Internat. Confer. Computer Design* (IEEE Computer Society Press, New York, 1984), pp. 646–651.
- [30] I. Ziskind and M. Wax, *IEEE Trans. Antennas Propagat.* **38**, 1111, (1990).
- [31] W. Youhua, Y. Weili and Z. Guangsheng, *IEEE Trans. on Magnetics*, **32**, 1214, (1996).
- [32] S. Shinomoto and V. Kabashima, *J. Phys. A Math. Gen.*, **24**, L141, (1991).
- [33] K. H. Hoffman and P. Salamon, *J. Phys. A Math. Gen.*, **23**, 3511, (1990).
- [34] F. Aluffi-Pentini, V. Parisi and F. Zirilli, *Journal of optimization theory and applications*, **47**, 1 (1985).
- [35] M. I. Marković and A. D. Rakić, *Appl. Opt.* **29**, 3479 (1990).
- [36] M. I. Marković and A. D. Rakić, *Opt. Laser Technol.* **22**, 394 (1990).
- [37] A. D. Rakić, *Appl. Opt.* **34**, 4755 (1995).
- [38] C. J. Powell, *J. Opt. Soc. Am.* **60**, 78 (1970).

- [39] H. Ehrenreich and H. R. Philipp, Phys. Rev. **128**, 1622 (1962).
- [40] H. Ehrenreich, H. R. Philipp, and B. Segall, Phys. Rev. **132**, 1918 (1963).

(Received September 1997.)