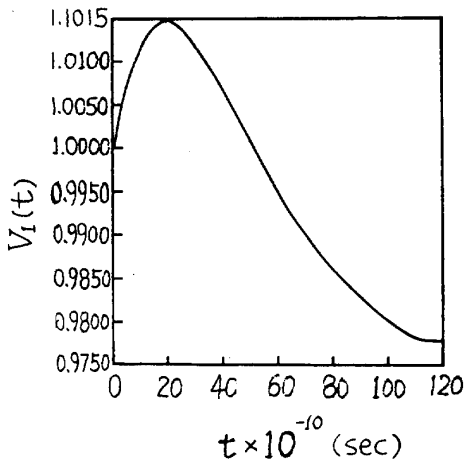

Fig. 3. $V_1(t)$ in Table II.

TABLE I

$t(\times 10^{-10}$ Sec.)	$V_1(t)$	$V_2(t)$	$V_3(t)$
0	1.000000	1.00000	1.000000
1	1.017499	0.973549	1.008943
2	1.032095	0.950667	1.016239
⋮	⋮	⋮	⋮
22	1.105154	0.822897	1.050144
23	1.105622	0.821991	1.050345

TABLE II

$t(\times 10^{-8}u1^{0}$ Sec.)	$V_1(t)$	$V_2(t)$	$V_3(t)$	$V_4(t)$
0	1.000000	1.00000	1.000000	1.000000
1	1.001824	0.999194	1.014673	0.984438
2	1.003504	0.998287	1.028989	0.969018
⋮	⋮	⋮	⋮	⋮
120	0.977810	0.873353	1.488217	0.286843
121	0.977669	0.873099	1.488628	0.285954

These simulation results demonstrate the accuracy of the above analyses and the effectiveness of this proposed approach for the real-time signal processing.

The First Simulation Result

$$D = \begin{bmatrix} 1.65000 & -2.00000 & 0.50000 \\ -2.00000 & 4.65000 & -1.00000 \\ 0.500000 & -1.00000 & 0.90000 \end{bmatrix}$$

$$\lambda_a = 0.6500000$$

$$\lambda_1 = 0.650068$$

$$V_f = [1.106019 \quad 0.821224 \quad 1.050515]^T$$

The Second Simulation Result

$$A = \begin{bmatrix} 0.764400 & 0.025200 & -0.038400 & 0.086400 \\ 0.025200 & 0.794100 & -0.067200 & 0.151200 \\ -0.038400 & -0.067200 & 0.852400 & -0.230400 \\ 0.086400 & 0.151200 & -0.230400 & 1.268400 \end{bmatrix}$$

$$\lambda_a = 0.75000$$

$$\lambda_1 = 0.750227$$

$$V_f = [0.977532 \quad 0.872851 \quad 1.489027 \quad 0.285089]^T$$

References

[1] M. A. Rahman and Y. K. Yu, "Total least square approach for frequency estimation using linear prediction," *IEEE Trans. ASSP*, vol. 35, pp. 1440–1454, 1987.
[2] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. AP*, vol. 34, pp. 276–280, 1986.
[3] J. W. R. Griffiths, "Adaptive array processing, a tutorial," *IEE Proc.*, vol. 130, no. 1, pp. 3–10, 1983.
[4] Z. Banjanian, J. R. Cruz, and D. S. Zrnic, "Eigendecomposition methods for frequency estimation: A unified approach," pp. 2595–2598, *Proc. ICASSP '90*, 1990.
[5] A. Cickocki and R. Unbehauen, "Neural networks for computing eigenvalue and eigenvectors," *Biological Cybern.*, vol. 68, pp. 155–164, 1992.
[6] M. Takeda and J. W. Goodman, "Neural networks for computation: Numerical representation and programming complexity," *Appl. Optics*, vol. 25, pp. 3033–3052, 1986.

A Novel Algorithm for DC Analysis of Piecewise-Linear Circuits: Popcorn

Satılmış Topçu, Ogan Ocalı, Abdullah Atalar, and Mehmet A. Tan

***Abstract*—A fast and convergent iteration method for piecewise-linear analysis of nonlinear resistive circuits is presented. Most of the existing algorithms are applicable only to a limited class of circuits. In general, they are either not convergent or too slow for large circuits. The new algorithm presented in the paper is much more efficient than the existing ones and can be applied to any piecewise-linear circuit. It is based on the piecewise-linear version of the Newton-Raphson algorithm. As opposed to the Newton-Raphson method, the new algorithm is globally convergent from an arbitrary starting point. It is simple to understand and it can be easily programmed. Some numerical examples are given in order to demonstrate the effectiveness of the proposed algorithm in terms of the amount of computation.**

I. Introduction

DC analysis of nonlinear resistive circuits is one of the basic problems in the computer-aided design of electronic circuits. Various methods are available for the solution of this problem. These methods can be classified into two major groups. One is based on an iterative

Manuscript received March 29, 1993; revised March 24, 1994. This paper was recommended by Associate Editor Martin Hasler.

The authors are with the Electrical and Electronics Engineering Department, Bilkent University, 06533 Bilkent, Ankara, Turkey.

IEEE Log Number 9403404.

algorithm which is applied directly to the nonlinear circuit equations. The well-known method in this group is the Newton-Raphson method [1]–[3]. The second group is based on the piecewise-linear (PWL) analysis which has been investigated by many researchers due to its computational efficiency [4]–[15].

In the PWL analysis, a nonlinear resistive circuit can be described by

$$\mathbf{f}(\mathbf{x}) = \mathbf{y} \tag{1}$$

where $\mathbf{f}(\cdot)$ is a continuous PWL mapping from $\mathbf{R}^n$ into itself, $\mathbf{x}$ is a point in $\mathbf{R}^n$ and represents a set of chosen circuit variables and $\mathbf{y}$ is an arbitrary point in $\mathbf{R}^n$ which represents the inputs to the circuit. The operating region of every nonlinear element is divided into a finite number of segments. Hence, the space $\mathbf{R}^n$ is divided into N linear regions bounded by hyperplanes where N is a very large number. The system of PWL equations in (1) can be expressed by the following set of linear simultaneous equations

$$\mathbf{A}_l\mathbf{x} + \mathbf{w}_l = \mathbf{y}, \quad \text{for } \sigma_l, \quad l = 1, 2, \ldots, N \tag{2}$$

where $\mathbf{A}_l$ is a constant $n \times n$ matrix (called Jacobian matrix for convenience) and $\mathbf{w}_l$ is a constant n-vector. They characterize the circuit in linear region σ_l. To find all solutions of (1), one may solve n linear simultaneous equations in (2) for each of N linear regions to find $x^{(l)}$ and decide whether $x^{(l)}$ lies within the considered linear region, σ_l. If $x^{(l)}$ lies within σ_l, it is a valid solution. This method is conceptually simple and finds all existing solutions, but it is computationally complex. Recently, a number of authors have proposed various methods to decrease the number of linear regions, N, by a sign test. One of these methods [4] requires more than $O(Nn^2)$ multiplications. Moreover, the sign test is not a simple procedure. A more efficient method is proposed in [5]. Nishi [6] has proposed a method in which the number of multiplications required to find all solutions of (2) is $O(Nn)$. Although the method developed in [7] seems to be the best, it is computationally impractical for large PWL circuits. For example, if the circuit contains 1000 MOS transistors each of which is modeled with 4 segments, then there are 4^{1000} (approximately 10^{600}) linear regions. If the sign test requires at least one multiplication for each linear region, it will take much more than billions of years on today's supercomputers to find the solutions by using these methods.

In this paper, we present a new algorithm, which we call *popcorn*, shown to be more efficient than the existing algorithms of the same generality. This algorithm is globally convergent for a general class of PWL resistive circuits with no restrictions. It is simple and can be easily programmed. The method of PWL analysis of nonlinear resistive circuits is reviewed in section II. The popcorn algorithm is presented in section III. Some numerical examples are given in section IV to illustrate the effectiveness of the algorithm.

II. Piecewise-Linear Analysis

In PWL analysis, the well-known technique due to Katzenelson [8] has been originally applied to the circuits with two-terminal elements which are strictly monotonic. The PWL approach was further extended to include the resistive circuits of much broader class [9]–[15]. In particular, Fujisawa and Kuh [11] have shown that the Katzenelson's algorithm can be applied to (1) and it always converges to a solution as long as the equation has a unique solution. Fujisawa, Kuh, and Ohtsuki [12] have shown that if all the Jacobian matrix determinants $\det\mathbf{A}_l$, $l = 1, 2, \ldots, N$ in (2) have the same sign, then there exists at least one solution to the equation $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ and the algorithm also converges. This property is referred to as *the sign condition.* This restriction of the sign condition was later removed in the generalized Katzenelson's method [13], [15].

There exists also a PWL version of the Newton-Raphson method [2]. However, it is well-known that for the continuous case the Newton-Raphson method may not converge depending on the initial guess. The same situation may occur in PWL case, if the initial linear region is not close enough to the linear region of the solution. The divergence can be in the form of a cyclic repetition of two or more virtual linear regions. We have observed that the PWL version of the Newton-Raphson method may not converge for some circuits, particularly with multiple solutions. We have tested this method 100 times on a 128-bit shift register circuit which contains 2580 MOS transistors using different initial linear regions. It has converged in only 22 trials, but the convergence speed was very high. Hence, the PWL Newton-Raphson method does not guarantee convergence, but if it does converge, it is extremely fast. We have developed a new algorithm as described in the next section, by modifying the PWL Newton-Raphson method to avoid its major drawback, i.e., divergence.

III. The Popcorn Algorithm

The new algorithm is described as follows:

1) Initially, choose an arbitrary linear region, let's say, σ_k. $\sigma_k = \{a_{k,1}, a_{k,2}, \ldots, a_{k,m}\}$ where $a_{k,j}$ represents the segment for the jth element in the kth iteration. Set $k = 0$.
2) Compute $\mathbf{x}_{k+1}$ from

$$\mathbf{x}_{k+1} = \mathbf{A}_k^{-1}(\mathbf{y} - \mathbf{w}_k)$$

Check if $\mathbf{x}_{k+1}$ lies in σ_k. If so, STOP; $\mathbf{x}_{k+1}$ is the solution. Otherwise, CONTINUE.
3) Let $\sigma'_{k+1} = \{a'_{k+1,1}, a'_{k+1,2}, \ldots, a'_{k+1,m}\}$ be the linear region where $\mathbf{x}_{k+1}$ lies. The next linear region $\sigma_{k+1} = \{a_{k+1,1}, a_{k+1,2}, \ldots, a_{k+1,m}\}$ is chosen as follows: For $j = 1, 2, \ldots, m$
If $a'_{k+1,j} = a_{k,j}$ then

$$a_{k+1,j} = \begin{cases} a'_{k+1,j} & \text{with probability } 1 - q \\ \text{Any other segment} & \text{with probability } q,\ 0 < q < 1 \end{cases}$$

If $a'_{k+1,j} \neq a_{k,j}$ then

$$a_{k+1,j} = \begin{cases} a'_{k+1,j} & \text{with probability } 1 - p \\ \text{Any other segment} & \text{with probability } p,\ 0 < p < 1 \end{cases}$$

4) Set $k = k + 1$. Go to step 2.

A segment may not be chosen, albeit with a very small probability q, even though the present solution satisfies the limits of the assumed segment. If the solution does not satisfy the assumed segment, the segment in which the present solution lies is chosen with a high probability $(1 - p)$. With a small probability p any other segment is chosen. Here the other segments are chosen with equal likelihood. The segment selection procedure for each nonlinear device is independent of the other nonlinear devices. Note that, if $p = 0$ and $q = 0$ then the algorithm becomes identical to the PWL Newton-Raphson algorithm. For $q = 0$, we have constructed a counterexample circuit with no convergence. That circuit, shown in Fig. 1, contains two voltage-controlled voltage sources and two tunnel diodes modeled by 3 PWL segments. This circuit has a unique solution, but the algorithm described above cannot find the solution if $q = 0$. It must be noted that both the PWL Newton-Raphson and the Katzenelson algorithms fail for this circuit, unless the initial linear region happens to be the correct one.

The popcorn algorithm assures the convergence for any initial guess since the algorithm tries all of the linear regions eventually,

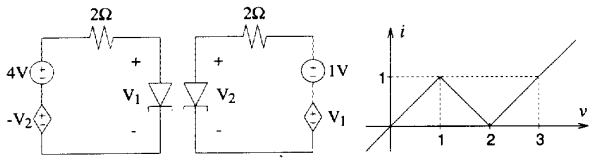

Fig. 1. Tunnel diode circuit and the i–v characteristics of the tunnel diodes.

until it converges. Having such a feature, it resembles the well-known simulated annealing algorithm without a cooling procedure [16]. The convergence proof is trivial, since the probability of visiting the linear region containing the solution is nonzero. In the worst case, the algorithm visits all linear regions and convergence is always assured. This simple proof does not tell us how fast the algorithm converges, it merely shows that it is convergent for sufficiently many iterations. In each iteration, the PWL Newton-Raphson method selects a new linear region to be used in the next iteration and our algorithm makes a random perturbation on that linear region by means of the parameters p and q to prevent divergence. Obviously, these parameters should be appropriately selected to improve the speed of the algorithm. We have made many experiments for different type and size of circuits by changing the values of p and q. The results obtained have been very encouraging as can be concluded from the numerical examples given in the following section.

IV. Numerical Examples

We have implemented the popcorn algorithm in C programming language and analyzed various CMOS, ECL and analog bipolar circuits. Let us describe the CMOS example circuits briefly. **Counter** is a combinational circuit which finds the number of one's in a 128-bit input. The circuit **lfsr** is a linear feedback shift register which produces pseudo-random binary numbers. **Sh128** is a 128-bit shift register circuit consisting of master-slave flip-flops. **Pgen** is a pulse generating circuit. **Addcs** circuit is a carry-select adder. The circuit **rsync** is used to produce a synchronization pulse. **Add18** is an 18-bit adder circuit. **Status** is a 5-bit register circuit which can be loaded in series or in parallel. **Sh5** is a 5-bit shift register circuit. The results given below are obtained using approximately 6,500 hours of CPU time on a number of SUN *Sparc*-2+ workstations.

Let us define $\bar{q}$ as equal to q times the number of nonlinear elements in a given circuit. The MOS transistors are modeled with 4 PWL segments representing the cutoff, saturation, linear, and reverse saturation states. The average number of iterations, $\bar{k}$, for the example circuits are shown in Fig. 2 as a function of p while $\bar{q}$ is kept constant at 0.005. The plots in Fig. 2 are obtained by taking the mean of more than 200 simulation results for every circuit at chosen values of p and $\bar{q}$. The mean value does not change more than 5% after 200 simulations have been performed. As it is seen from Fig. 2, for all of the circuits except for the combinational circuits such as **counter**, **addcs**, and **add18**, $\bar{k}$ reaches a minimum around $p = 0.2$ and it increases sharply as the value of p goes to 0 or 0.5. For combinational circuits, $\bar{k}$ increases monotonically with p. For 4-segment PWL MOSFET model, we can say that the parameter p can be safely set to a value between 0.1 and 0.3. The standard deviation in the required number of iterations is smaller than half of the mean in the range $0.1 \leq p \leq 0.3$.

We have also analyzed the example circuits using 9-segment PWL model for MOS transistors. In the 9-segment model, there are 4 segments in the linear region, and 2 segments each in the saturation and reverse saturation regions. The average of more than 200 simulation results for each circuit is given in Fig. 3. It is observed

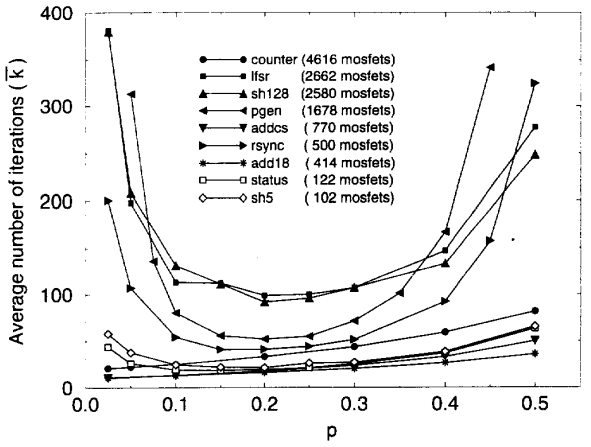

Fig. 2. Average number of iterations required for the popcorn algorithm as a function of p using $\bar{q} = 0.005$ and 4-segment PWL MOSFET model.

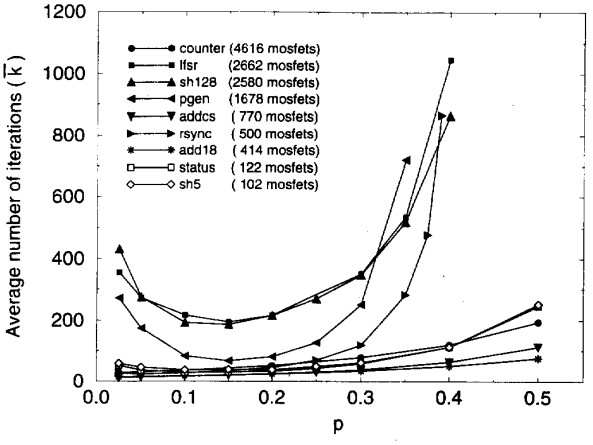

Fig. 3. Average number of iterations required for the popcorn algorithm as a function of p using $\bar{q} = 0.005$ and 9-segment PWL MOSFET model.

that the results for both 4-segment and 9-segment models have similar characteristics. The number of iterations is approximately doubled for 9-segment model. As it is seen from Fig. 3, for 9-segment PWL MOSFET model, the minimum occurs around $p = 0.15$ and the parameter p can be set to a value between 0.05 and 0.25.

In order to find a suitable value for $\bar{q}$, we have analyzed the same circuits by setting $p = 0.2$ and changing the value of $\bar{q}$. Fig. 4 shows $\bar{k}$ as a function of $\bar{q}$ using 4-segment PWL MOSFET model. As it can be seen from Fig. 4, for all of the circuits except for the tunnel diode circuit, $\bar{k}$ increases as $\bar{q}$ approaches unity. The tunnel diode circuit, however, needs a $\bar{q}$ value close to unity to converge quickly. Therefore, a compromising value of the parameter $\bar{q}$ can be chosen between 0.02 and 0.5. The standard deviation is not larger than half of the mean value in this range.

We have chosen some example circuits to make a performance comparison between the popcorn, PWL Newton-Raphson and the Katzenelson algorithms. First, we have used the circuit **rsync** which has multiple solutions. We have set $p = 0.2$ and $\bar{q} = 0.1$ in the popcorn algorithm. The results for this circuit are given in Fig. 5. The vertical axis in Fig. 5 represents the number of transistors which could not find the correct segment at the corresponding iteration. When this number becomes zero, it means that the solution is found.

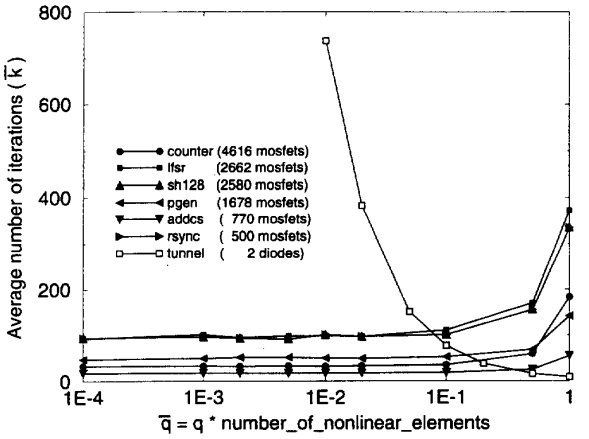

Fig. 4. Average number of iterations required for the popcorn algorithm as a function of $\bar{q}$ using $p = 0.2$ and 4-segment PWL MOSFET model.

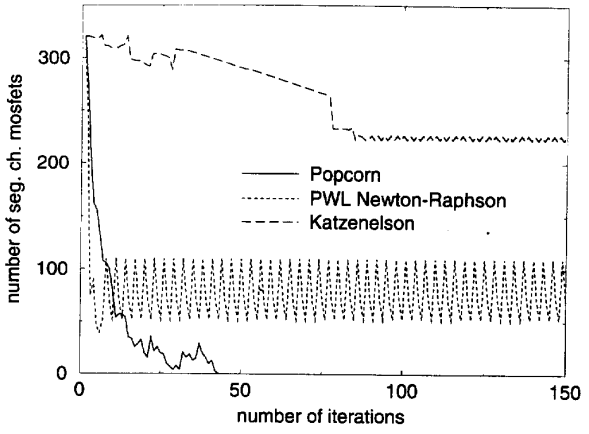

Fig. 5. The results of the popcorn, PWL Newton-Raphson, and the Katzenelson algorithms for the circuit **rsync** with 500 MOS transistors.

TABLE I

Circuit	# nonlinear elements	$\bar{k}$		
		PWL Newton-Raphson	Popcorn	Katzenelson
Control (CMOS)	176 mosfet	6	9	73
Add18 (CMOS)	414 mosfet	11	18	146
Addcs (CMOS)	770 mosfet	10	17	74
Counter (CMOS)	4616 mosfet	20	33	565
4-bit FA (ECL)	102 bjt, 34 diode	12	35	186
Opamp	26 bjt	17	281	X

It is seen from Fig. 5 that the PWL Newton-Raphson method and the Katzenelson algorithm fail in finding any of the multiple solutions. However, the popcorn algorithm has converged to one of the solutions in each trial. Second, we have chosen several CMOS, ECL and analog bipolar circuits. Table I gives $\bar{k}$ for these circuits by using three different algorithms. **Opamp** is a noninverting amplifier circuit containing 741 operational amplifier. The MOS transistors and BJT's are modeled with 4 segments while the diodes are modeled using 2 segments. In the popcorn algorithm, the parameter values are chosen to be $p = 0.2$ and $\bar{q} = 0.1$. As it is seen from Table I, the speed of PWL Newton-Raphson method is better than the speed of popcorn algorithm. However, the Katzenelson algorithm is relatively slow compared to the popcorn algorithm.

V. Conclusion

An efficient algorithm for finding DC solution of *large* PWL resistive circuits has been proposed. The algorithm is an extension of the piecewise-linear version of the Newton-Raphson method. The main feature of our approach is to insert some randomness into the PWL Newton-Raphson method to guarantee convergence without sacrificing the speed. The degree of randomness in the algorithm is controlled by the parameters p and q. We have found appropriate values for these parameters using the large number of trials on the example circuits. In the case of multiple DC solutions, the algorithm reaches to one of the solutions in each trial. This algorithm can also be adapted to the continuous case by modifying the Newton-Raphson algorithm.

References

[1] F. H. Branin Jr. and H. H. Wang, "A fast reliable iteration method for dc analysis of nonlinear networks," *Proc. IEEE*, vol. 55, pp. 1819–1826, Nov. 1967.

[2] L. O. Chua and P. M. Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques.* Englewood Cliffs, NJ: Prentice Hall, 1975.

[3] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design.* New York: Van Nostrand Reinhold, 1983.

[4] L. O. Chua and R.L.P. Ying, "Finding all solutions of piecewise-linear circuits," *Int. J. Circuit Theory Appl.*, vol. 10, pp. 201–229, July 1982.

[5] Q. Huang and R. W. Liu, "A simple algorithm for finding all solutions of piecewise-linear networks," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 600–609, Apr. 1989.

[6] T. Nishi, "An efficient method to find all solutions of piecewise-linear resistive circuits," in *Proc. IEEE Int. Symp. Circuits and Syst.*, pp. 2052–2055, May 1989.

[7] K. Yamamura and M. Ochiai, "An efficient algorithm for finding all solutions of piecewise-linear resistive circuits," *IEEE Trans. Circuits Syst.-I*, vol. 39, pp. 213–221, Mar. 1992.

[8] J. Katzenelson, "An algorithm for solving nonlinear resistor networks," *Bell Syst. Tech. J.*, vol. 44, pp. 1605–1620, Oct. 1965.

[9] T. Ohtsuki and N. Yoshida, "DC analysis of nonlinear networks based on generalized piecewise-linear characterization," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 146–152, Jan. 1971.

[10] E. S. Kuh and I. N. Hajj, "Nonlinear circuit theory: Resistive networks," *Proc. IEEE*, vol. 59, pp. 340–355, Mar. 1971.

[11] T. Fujisawa and E. S. Kuh, "Piecewise-linear theory of nonlinear networks," *SIAM J. Appl. Math.*, vol. 22, pp. 307–328, Mar. 1972.

[12] T. Fujisawa, E. S. Kuh, and T. Ohtsuki, "A sparse matrix method for analysis of piecewise-linear resistive networks," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 571–584, Nov. 1972.

[13] T. Ohtsuki, T. Fujisawa, and S. Kumagai, "Existence theorems and a solution algorithm for piecewise-linear resistive networks," *SIAM J. Math. Anal.*, vol. 8, pp. 69–99, Feb. 1977.

[14] M. J. Chien and E. S. Kuh, "Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 305–317, June 1977.

[15] S. M. Lee and K. S. Chao, "Multiple solutions of piecewise-linear resistive networks," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 84–89, Feb. 1983.

[16] P. J. M. van Laarhoven and E. H. L. Aarts *Simulated Annealing: Theory and Applications.* Lancaster: D. Reidel, 1987.