数理解析研究所講究録
1147 巻 2000 年 121-126

Finding All Solutions of Nonlinear Equations Using the Dual Simplex Method

山村 清隆† (Kiyotaka Yamamura), 田中 茂‡ (Shigeru Tanaka)

† Chuo University, 1–13–27, Kasuga, Bunkyo-ku, Tokyo, 112-8551 Japan
yamamura@elect.chuo-u.ac.jp
‡ Gunma University, 1–5–1 Tenjin-cho, Kiryu, 376-8515 Japan

Abstract

An efficient algorithm is proposed for finding all solutions of systems of nonlinear equations using linear programming (LP). This algorithm is based on a simple test (termed the LP test) for nonexistence of a solution to a system of nonlinear equations in a given region. In the LP test, the system of nonlinear equations is transformed into an LP problem, to which the simplex method is applied. Such an LP problem is obtained by surrounding the nonlinear functions by rectangles using interval extensions. In this paper, we introduce the dual simplex method to the LP test, which makes the average number of pivotings per region much smaller (less than one, for example) and makes the algorithm very efficient.

I. Introduction

Finding all solutions of nonlinear equations is an important problem which is widely encountered in science and engineering. In this paper, we consider the problem of finding all solutions of a system of n nonlinear equations:

$$\begin{array}{rcl} f_1(x_1, x_2, \ldots, x_n) & = & 0 \\ f_2(x_1, x_2, \ldots, x_n) & = & 0 \\ & \vdots & \\ f_n(x_1, x_2, \ldots, x_n) & = & 0 \end{array} \tag{1}$$

contained in a bounded rectangular region (box) D in R^n, where $f_1, f_2, \ldots, f_n$ are real-valued nonlinear functions. In vector notation the system (1) will be written as $f(x) = 0$.

As a computational method to find all solutions of nonlinear equations, interval analysis is well-known, and various algorithms based on interval analysis techniques have been developed. Using the interval analysis, all solutions of (1) contained in $D \subset R^n$ can be found with mathematical certainty. However, the computation time of the interval analysis tends to grow exponentially with the dimension n.

In order to improve the computational efficiency of interval analysis, it is necessary to develop a powerful test for nonexistence of a solution to (1) in a given box so that we can exclude many boxes containing no solution at an early stage of the algorithm [1]–[3]. However, the test which was used in the conventional interval analysis is not necessarily a powerful test for excluding boxes.

Recently, a new computational test has been proposed for nonexistence of a solution to the system of nonlinear equations (1) in a region $X \subseteq D$ [1]. This test is termed the LP test. The basic idea of this test is to formulate a linear programming (LP) problem whose feasible region contains all solutions in X. Hence, if the feasible region is empty (which can be easily checked by the simplex method), then X contains no solution, and we can exclude it from further consideration. The LP problem is formulated by replacing the component nonlinear functions in (1) with auxiliary variables and linear inequalities which are obtained by interval extensions. This test is much more powerful than the conventional nonexistence test if the system of nonlinear equations consists of many linear terms and a relatively small number of nonlinear terms. By introducing the LP test to the interval analysis, all solutions of nonlinear equations can be found very efficiently.

However, the computational complexity of LP is generally much larger than that of the conventional test. Although the test is powerful, the algorithm does not become efficient if the computational cost of the test is very large. Therefore, it is important to consider not only the powerfulness of the nonexistence test but also its computational cost to improve the overall efficiency.

In this paper, we propose an efficient algorithm for finding all solutions of systems of nonlinear equations using the dual simplex method. By using the dual simplex method, the number of pivotings needed in the LP test becomes much smaller, which makes the algorithm very efficient.

II. Interval Analysis

In this section, we first summarize the basic procedures of interval analysis.

Intervals will be denoted by capital letters. An n-dimensional interval vector with components $X_i = [a_i, b_i]$ $(i = 1, 2, \ldots, n)$ is denoted by

$$X = (X_1, X_2, \ldots, X_n)^T. \tag{2}$$

Geometrically, X is an n-dimensional box.

In interval analysis, the following procedure is performed recursively, beginning with the initial box $X = D$. At each level, we analyze the box X. If there is no solution of (1) in X, then we exclude it from further consideration. If there is a unique solution of (1) in X, then we compute it by some iterative method. In the field of interval analysis, computationally verifiable sufficient conditions for nonexistence, existence and uniqueness of a solution in X have been developed. If these conditions are not satisfied and the existence or nonexistence of a solution in X cannot be checked, then split X in some appropriately chosen coordinate direction(s) to form two (or more) new boxes; we then continue the above procedure with one of these boxes, and put the other one(s) on a stack for later consideration. Thus, we can find all solutions of (1) contained in $D \subset R^n$ with mathematical certainty.

The nonexistence of a solution in X can be checked by using interval extensions. If in $f(x_1, x_2, \ldots, x_n)$ the variables x_i are replaced by intervals X_i and the arithmetic operations are replaced by the corresponding interval operations (for example, $x_1 + x_2$ is replaced by $X_1 + X_2 = [a_1 + a_2, b_1 + b_2]$), then an interval-valued vector function $F(X)$ is obtained which is called the interval extension of $f(x)$. It is known that $F(X)$ contains the range of $f(x)$ over X. Hence, if

$$0 \notin F(X) \tag{3}$$

holds, then there is no solution of (1) in X. This is the computationally verifiable sufficient condition for nonexistence of a solution to (1) in X, which is used as the nonexistence test in conventional interval analysis [1].

However, (3) is not necessarily a powerful test for excluding boxes. In order to improve the computational efficiency of interval analysis, it is necessary to develop a powerful test which can exclude many boxes containing no solution at an early stage.

In practical problems, the system of nonlinear equations often consists of many linear terms and a relatively small number of nonlinear terms. For such systems, the LP test is much more powerful than the conventional test (3).

[1] As another computationally verifiable sufficient condition for nonexistence of a solution to (1) in X, $K(X) \cap X = \phi$ is known where $K(X)$ is the Krawczyk operator.

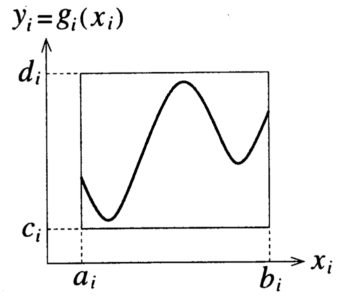

Fig. 1 In the LP test, nonlinear functions are surrounded by rectangles.

III. LP Test

In this section, we review the LP test proposed in [1]. Consider a system of n nonlinear equations:

$$f(x) \triangleq Pg(x) + Qx - s = 0 \tag{4}$$

where $x = (x_1, x_2, \ldots, x_n)^T \in R^n$ is a variable vector, $s = (s_1, s_2, \ldots, s_n)^T \in R^n$ is a constant vector, P and Q are $n \times n$ constant matrices, and $g(x) = [g_1(x_1), g_2(x_2), \ldots, g_n(x_n)]^T$ is a continuous nonlinear function with component functions $g_i(x_i) : R^1 \to R^1$ $(i = 1, 2, \ldots, n)$. Note that the discussion of this paper is easily extended to more general cases (e.g., the case where $f(x)$ contains nonseparable functions of more than one variable). As for details, see [1].

In the LP test, we first compute the interval extensions of $g_i(x_i)$ over $[a_i, b_i]$ $(i = 1, 2, \ldots, n)$. Let the interval extension be $[c_i, d_i]$. Then, we introduce auxiliary variables y_i $(i = 1, 2, \ldots, n)$ and put $y_i = g_i(x_i)$. If $a_i \le x_i \le b_i$, then $c_i \le y_i \le d_i$.

Now we replace each nonlinear function $g_i(x_i)$ in (4) by the auxiliary variable y_i and the linear inequality $c_i \le y_i \le d_i$, and consider the LP problem:

$$\begin{aligned} &\max \ \text{(arbitrary constant)} \\ &\text{subject to} \\ &\quad Py + Qx - s = 0 \qquad\qquad\qquad (5) \\ &\quad a_i \le x_i \le b_i, \qquad i = 1, 2, \ldots, n \\ &\quad c_i \le y_i \le d_i, \qquad i = 1, 2, \ldots, n \end{aligned}$$

where $y = (y_1, y_2, \ldots, y_n)^T \in R^n$. Geometrically, the inequality constraints in (5) imply that the component nonlinear functions $g_i(x_i)$ are surrounded by rectangles as shown in Fig. 1. Then, we apply the simplex method to (5).

As is well-known, the simplex method consists of Phase I and Phase II. In Phase I, we find a basic feasible solution using artificial variables. In Phase II, we optimize the objective function starting with the basic feasible solution obtained by Phase I. If there is no feasible solution, then Phase I terminates with that information.

If the feasible region of (5) is empty, then we can conclude that there is no solution of (5) in X because all solutions of (4) which exist in X satisfy the constraints in (5). This test is called the LP test. Since there is much good software for the simplex method, the implementation of the LP test is very easy. It has been shown that the interval algorithm using the LP test (which will be called the LP test algorithm for short) is much more efficient than the original interval algorithm [1],[2].

Now let us examine the size of the tableau in the LP test. In the implementation of the simplex method to (5), we apply the variable transformation $\bar{x}_i = x_i - a_i$ and $\bar{y}_i = y_i - c_i$ so that the LP problem becomes the form with non-negativity constraints:

$$\begin{aligned}
&\max \ (\text{arbitrary constant})\\
&\text{subject to}\\
&\quad P\bar{y} + Q\bar{x} - \bar{s} = 0\\
&\quad \bar{x}_i \le b_i - a_i, \qquad i = 1,2,\ldots,n\\
&\quad \bar{y}_i \le d_i - c_i, \qquad i = 1,2,\ldots,n\\
&\quad \bar{x}_i \ge 0, \ \ \bar{y}_i \ge 0, \qquad i = 1,2,\ldots,n
\end{aligned} \tag{6}$$

where $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)^T \in R^n$ and $\bar{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n)^T \in R^n$. This LP problem has n equality constraints and $2n$ inequality constraints (excluding the non-negativity constraints). Introducing the slack variables $\bar{\lambda}_i$ and $\bar{\mu}_i$ $(i = 1,2,\ldots,n)$, (6) is transformed into a standard form:

$$\begin{aligned}
&\max \ (\text{arbitrary constant})\\
&\text{subject to}\\
&\quad P\bar{y} + Q\bar{x} - \bar{s} = 0\\
&\quad \bar{x}_i + \bar{\lambda}_i = b_i - a_i, \qquad i = 1,2,\ldots,n\\
&\quad \bar{y}_i + \bar{\mu}_i = d_i - c_i, \qquad i = 1,2,\ldots,n\\
&\quad \bar{x}_i \ge 0, \ \bar{y}_i \ge 0, \ \bar{\lambda}_i \ge 0, \ \bar{\mu}_i \ge 0, \quad i = 1,2,\ldots,n.
\end{aligned} \tag{7}$$

In Phase I, we introduce artificial variables to obtain an initial basic feasible solution. Since $b_i - a_i > 0$ and $d_i - c_i > 0$ $(i = 1,2,\ldots,n)$ hold, we only need to introduce n artificial variables for the first n equality constraints. Hence, the size of the tableau is $(3n+1) \times (2n+1)$.

IV. LP Test Using the Dual Simplex Method

In the LP test algorithm, the simplex method is performed on many regions. If we start the simplex method always from the beginning of Phase I, then the total number of pivotings becomes very large because the simplex method requires many pivotings for large scale problems. In this section, we

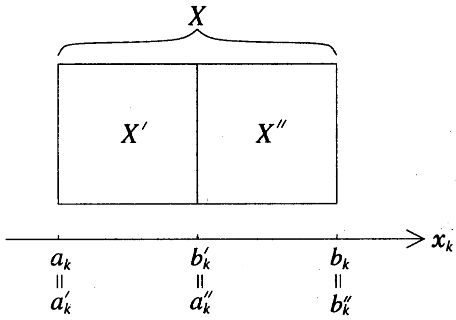

Fig. 2 The LP test can be performed by the dual simplex method from the second region.

show that the LP test can be performed very efficiently in a small number of pivotings by using the dual simplex method.

Consider that we have performed the LP test on a box X where the feasible region is not empty and have obtained an optimal (in the sense of Phase I) feasible tableau for (7). Then, we divide X into two boxes X' and X'' that are adjacent to each other in the x_k-direction as shown in Fig. 2. Then, we perform the LP test for X' and X''. However, for these boxes, we need not perform the simplex method from the beginning of Phase I. Instead, we perform the following procedure.

We first perform the LP test for X'. Let $[c'_k, d'_k]$ be the interval extension of $g_k(x_k)$ over $[a'_k, b'_k]$. Note that $c'_k \ge c_k$ and $d'_k \le d_k$ hold. In the LP test, we introduce auxiliary variables y_i $(i = 1,2,\ldots,n)$ and consider the LP problem:

$$\begin{aligned}
&\max \ (\text{arbitrary constant})\\
&\text{subject to}\\
&\quad Py + Qx - s = 0\\
&\quad a_i \le x_i \le b_i, \qquad i = 1,2,\ldots,n, \ i \ne k\\
&\quad a'_k \le x_k \le b'_k\\
&\quad c_i \le y_i \le d_i, \qquad i = 1,2,\ldots,n, \ i \ne k\\
&\quad c'_k \le y_k \le d'_k.
\end{aligned} \tag{8}$$

Applying the variable transformation $\bar{x}_i = x_i - a_i$ $(i \ne k)$, $\tilde{x}_k = x_k - a'_k$, $\bar{y}_i = y_i - c_i$ $(i \ne k)$, and $\tilde{y}_k = y_k - c'_k$, and introducing the slack variables, (8) is transformed into a standard form:

$$\begin{aligned}
&\max \ (\text{arbitrary constant})\\
&\text{subject to}\\
&\quad P\tilde{y} + Q\tilde{x} - \tilde{s} = 0\\
&\quad \bar{x}_i + \bar{\lambda}_i = b_i - a_i, \qquad i = 1,2,\ldots,n, \ i \ne k\\
&\quad \tilde{x}_k + \tilde{\lambda}_k = b'_k - a'_k\\
&\quad \bar{y}_i + \bar{\mu}_i = d_i - c_i, \qquad i = 1,2,\ldots,n, \ i \ne k\\
&\quad \tilde{y}_k + \tilde{\mu}_k = d'_k - c'_k
\end{aligned} \tag{9}$$

where $\tilde{x} = (\bar{x}_1, \ldots, \bar{x}_{k-1}, \tilde{x}_k, \bar{x}_{k+1}, \ldots, \bar{x}_n)^T \in R^n$ and $\tilde{y} = (\bar{y}_1, \ldots, \bar{y}_{k-1}, \tilde{y}_k, \bar{y}_{k+1}, \ldots, \bar{y}_n)^T \in R^n$. In (9), the non-negativity constraints are omitted. Notice that the constraints in (9) and those in (7) are different only in the constant terms.

From the relations $\tilde{x}_k = \bar{x}_k$, $\tilde{\lambda}_k = \bar{\lambda}_k - (b_k - b'_k)$, $\tilde{y}_k = \bar{y}_k - (c'_k - c_k)$, and $\tilde{\mu}_k = \bar{\mu}_k - (d_k - d'_k)$, the dual feasible tableau for (9) is easily obtained by modifying the final (optimal in the sense of Phase I) feasible tableau for (7) a little. For example, if the optimal feasible tableau for (7) is

$$
\begin{array}{c||c|ccccc}
 & & & & \bar{\mu}_k & & \\
\hline\hline
 & \vdots & & & \vdots & & \\
\bar{x}_k & \alpha & & \cdots & \gamma & \cdots & \\
 & \vdots & & & \vdots & & \\
\bar{\lambda}_k & \beta & & \cdots & -\gamma & \cdots & \\
 & \vdots & & & \vdots & & \\
\bar{y}_k & d_k - c_k & & 0 \ldots 0 & -1 & 0 \ldots 0 & \\
 & \vdots & & & \vdots & & \\
\hline\hline
\end{array}
$$

then the dual feasible tableau for (9) is

$$
\begin{array}{c||c|ccc}
 & & & \tilde{\mu}_k & \\
\hline\hline
 & \vdots & & \vdots & \\
\tilde{x}_k & \alpha + \gamma(d_k - d'_k) & \cdots & \gamma & \cdots \\
 & \vdots & & \vdots & \\
\tilde{\lambda}_k & \beta - (b_k - b'_k) - \gamma(d_k - d'_k) & \cdots & -\gamma & \cdots \\
 & \vdots & & \vdots & \\
\tilde{y}_k & d'_k - c'_k & 0 \ldots 0 & -1 & 0 \ldots 0 \\
 & \vdots & & \vdots & \\
\hline\hline
\end{array}
$$

that differs from the previous tableau only in the constant column (leftmost column).

Starting from this tableau (that may be feasible or infeasible), we can perform the dual simplex method and check the existence of the feasible region of (9). In general, this dual simplex method requires only a few pivotings (often no pivoting) per region. The same procedure is possible also for X''. Hence, the number of pivotings is substantially decreased from the second region.

V. Numerical Examples

We introduced the LP test using the dual simplex method to the piecewise-linear (PWL) version of the LP test algorithm proposed in [4] and implemented the new algorithm using the programming language C on a Sun Ultra Enterprise 3000 (248MHz). We applied the algorithm to the systems of PWL equations discussed in the numerical examples of [4]. In this section, we show the computational results. We will denote the total number of linear regions as L, the number of solutions obtained by the algorithm as S, the computation time as T, and the average number of pivotings per region as P. We used the same initial regions as those used in [4].

Table 1: Computation time T and the average number of pivotings per region P (Example 1).

			Ref.[5]	Ref.[4]		Proposed	
n	L	S	T (s)	T (s)	P	T (s)	P
10	10^{10}	7	46	0.7	10	0.1	0.5
12	10^{12}	5	1 423	1.8	13	0.2	0.5
14	10^{14}	5	98 378	4.0	16	0.4	0.4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	10^{20}	7	∞	31	24	2	0.4
30	10^{30}	7	∞	334	37	13	0.4
40	10^{40}	7	∞	859	48	27	0.4
50	10^{50}	9	∞	4 776	66	110	0.4
60	10^{60}	9	∞	11 824	80	224	0.4
70	10^{70}	9	∞	24 554	95	398	0.4
80	10^{80}	7	∞	28 168	101	451	0.5
90	10^{90}	7	∞	115 376	124	1 433	0.4
100	10^{100}	9	∞	∞	–	2 166	0.4
110	10^{110}	9	∞	∞	–	3 860	0.5
120	10^{120}	9	∞	∞	–	5 551	0.5
130	10^{130}	9	∞	∞	–	7 694	0.5
140	10^{140}	9	∞	∞	–	10 994	0.5
150	10^{150}	11	∞	∞	–	14 584	0.5

Example 1: We first consider a system of n nonlinear equations:

$$g(x_i) + x_1 + x_2 + \ldots + x_n - i = 0, \qquad i = 1, 2, \ldots, n$$

where

$$g(x_i) = 2.5x_i^3 - 10.5x_i^2 + 11.8x_i$$

which describes a nonlinear resistive circuit containing n tunnel diodes [4]. The number of variables is n, which is changed from $n = 10$ to $n = 150$. The nonlinear function $g(x_i)$ is approximated by a PWL function with ten segments. Hence, the number of linear regions L is 10^{10}–10^{150}.

Table 1 compares the computation time of the sign test algorithm in [5], the original LP test algorithm in [4], and the proposed algorithm, where ∞ denotes that it could not be computed in practical computation time. Table 1 also compares the average number of pivotings per region of the original LP test algorithm and the proposed algorithm. It is seen that the average number of pivotings per region is less than one in the proposed algorithm, which makes the algorithm about n times faster than the original algorithm.

Example 2: We next consider a system of n nonlinear equations:

$$x_i - \frac{1}{2n}\left(\sum_{j=1}^{n} x_j^3 + i\right) = 0, \qquad i = 1, 2, \ldots, n.$$

Table 2: Computation time T and the average number of pivotings per region P (Example 2).

			Ref.[5]	Ref.[4]		Proposed	
n	L	S	T (s)	T (s)	P	T (s)	P
10	10^{10}	3	154	0.8	11	0.1	0.5
12	10^{12}	3	5 031	1.9	14	0.2	0.5
14	10^{14}	3	174 757	4.0	17	0.4	0.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	10^{20}	3	∞	25	25	2	0.5
30	10^{30}	3	∞	205	40	8	0.6
40	10^{40}	3	∞	843	53	26	0.6
50	10^{50}	3	∞	2 579	67	63	0.6
60	10^{60}	3	∞	6 350	82	133	0.6
70	10^{70}	3	∞	13 684	98	255	0.7
80	10^{80}	3	∞	26 508	109	431	0.6
90	10^{90}	3	∞	48 433	127	702	0.7
100	10^{100}	3	∞	82 974	137	1 073	0.7
110	10^{110}	3	∞	134 388	155	1 597	0.7
120	10^{120}	3	∞	∞	–	2 254	0.7
130	10^{130}	3	∞	∞	–	3 013	0.7
140	10^{140}	3	∞	∞	–	4 184	0.7
150	10^{150}	3	∞	∞	–	5 401	0.7

Table 3: Computation time of the algorithms (transistor circuits).

				Ref.[4]		Proposed	
Circuit	n	L	S	T (s)	P	T (s)	P
Fig. 3	8	10^8	9	0.48	20	0.13	3.5
Fig. 4	9	10^9	3	0.32	16	0.11	2.8
Fig. 5	15	10^{15}	11	39.0	28	5.40	1.8

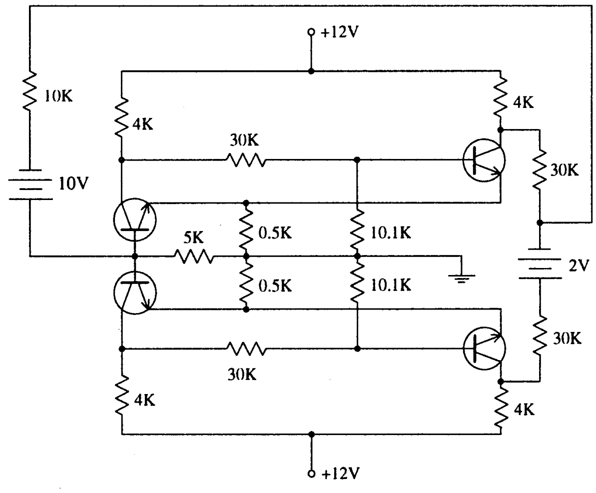

Fig. 3 Transistor circuit 1.

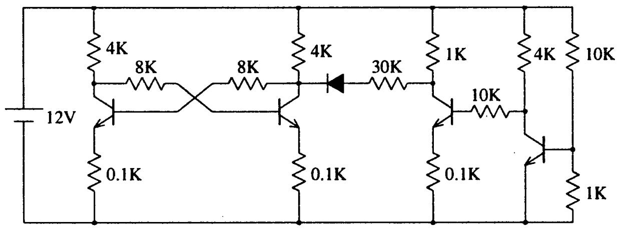

Fig. 4 Transistor circuit 2.

discussed in Example 3 of [4]. The initial region is $([-2.5, 2.5], \ldots, [-2.5, 2.5])^T$. The nonlinear function x_j^3 is approximated by a PWL function with ten segments. Table 2 shows the result of computation.

Example 3: We finally consider systems of nonlinear equations which describe the transistor circuits shown in Figs. 3–5 [4]. These circuits are described by systems of 8, 9, 15 (resp.) nonlinear equations of the form (4). Table 3 shows the result of computation.

VI. Conclusion

In this paper, an efficient algorithm has been proposed for finding all solutions of nonlinear equations using the dual simplex method. The proposed test is not only very powerful but also efficient and requires only a few pivotings per region. Therefore, it will be of great use in practical applications.

Acknowledgement

The authors wish to thank Prof. Tetsuro Yamamoto of Ehime University for his valuable suggestions.

References

[1] K. Yamamura, H. Kawata, and A. Tokue, "Interval solution of nonlinear equations using linear programming," *BIT*, vol. 38, no. 1, pp. 186–199, March 1998.

[2] K. Yamamura and M. Nishizawa, "Finding all solutions of a class of nonlinear equations using an improved LP test," *Japan J. Indust. Appl. Math.*, vol. 16, no. 3, pp. 349–368, Oct. 1999.

[3] K. Yamamura, "Finding all solutions of nonlinear equations using linear combinations of functions," *Reliable Computing*, vol. 6, no. 2, 2000.

[4] K. Yamamura and K. Yomogita, "Finding all solutions of piecewise-linear resistive circuits using an LP test," *IEEE Trans. Circuits and Systems-I*, vol. 47, 2000.

[5] K. Yamamura and M. Mishina, "An algorithm for finding all solutions of piecewise-linear resistive circuits," *Int. J. Cir. Theor. Appl.*, vol. 24, pp. 223–231, March 1996.

[6] K. Yamamura and K. Horiuchi, "A globally and quadratically convergent algorithm for solving nonlinear resistive networks," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 9, no. 5, pp. 487–499, May 1990.

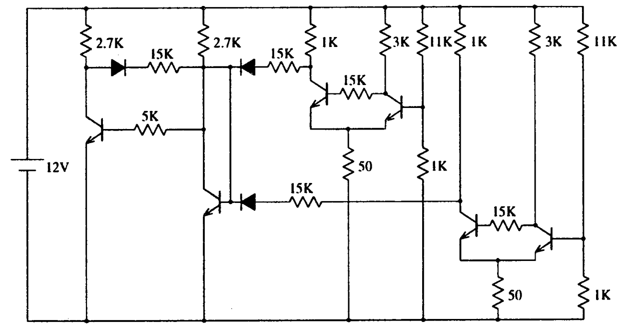

Fig. 5 Transistor circuit 3.

[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, no. 3, pp. 213–221, March 1992.

[8] K. Yamamura, "On piecewise-linear approximation of nonlinear mappings containing Gummel-Poon models or Shichman-Hodges models," *IEEE Trans. Circuits Syst.-I*, vol. 39, no. 8, pp. 694–697, Aug. 1992.

[9] K. Yamamura, "Simple algorithms for tracing solution curves," *IEEE Trans. Circuits Syst.-I*, vol. 40, no. 8, pp. 537–541, Aug. 1993.

[10] K. Yamamura, "Finding all solutions of piecewise-linear resistive circuits using simple sign tests," *IEEE Trans. Circuits Syst.-I*, vol. 40, no. 8, pp. 546–551, Aug. 1993.

[11] K. Yamamura, "An algorithm for representing functions of many variables by superpositions of functions of one variable and addition," *IEEE Trans. Circuits Syst.-I*, vol. 43, no. 4, pp. 338–340, April 1996.

[12] K. Yamamura and T. Ohshima, "Finding all solutions of piecewise-linear resistive circuits using linear programming," *IEEE Trans. Circuits Syst.-I*, vol. 45, no. 4, pp. 434–445, April 1998.

[13] K. Yamamura, T. Sekiguchi, and Y. Inoue, "A fixed-point homotopy method for solving modified nodal equations," *IEEE Trans. Circuits Syst.-I*, vol. 46, no. 6, pp. 654–665, June 1999.

[14] M. Nakata, T. Dobashi, Y. Inakuma, and K. Yamamura, "Coexistence curve of polystyrene in methylcyclohexane. X. Two-phase coexistence curves for ternary solutions near the tricritical compositions," *J. Chemical Physics*, vol. 111, no. 14, pp. 6617–6624, Oct. 1999.

[15] M. Suzuki, T. Dobashi, Y. Mikawa, K. Yamamura, and M. Nakata, "Reentrant three-phase equilibrium of homologous polystyrene solution," to appear in Physics Letters A.