数理解析研究所講究録
第880巻1994年167-175

多変数関数を一変数関数の和で表現するアルゴリズム

群馬大学工学部情報工学科　山村清隆 (Kiyotaka Yamamura)

1. まえがき

最近，多くの分野で非線形に関する研究が非常に活発になっている．特に工学では，非線形性は“避けるべきもの”から“積極的に利用して革新的な新技術を発展させるためのもの”へと変貌しつつあり，非線形に関する研究は今後ますます重要視されるものと予想される．

非線形システム解析における基礎的問題に，非線形方程式の求解問題がある．非線形方程式を効率よく解く手法を確立することは実用的にも重要なテーマであるが，非線形性に伴う種々の困難のため効率のよい万能的解法は見つかっておらず，本質的に難しい問題となっている．

過去，様々な非線形方程式の数値解法が提案されてきたが，それらの中には非線形関数の特定の性質 (単調性，ヤコビ行列のスパース性など) を利用することにより効率化を図るものも多い．そのような非線形関数の性質で最近多くの研究が行なわれているものの一つに，分離性がある．関数 $\boldsymbol{f}$: $\boldsymbol{R}^n \to \boldsymbol{R}^m$ が

$$\boldsymbol{f}(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} \boldsymbol{f}^i(x_i)$$

$$\boldsymbol{f}^i : \boldsymbol{R}^1 \to \boldsymbol{R}^m \quad (i = 1, 2, \cdots, n)$$

の形で表せるとき，$\boldsymbol{f}$ は分離可能 (separable) であるという．また，このような性質を変数分離可能性，あるいは簡単に分離性 (separability) という．すなわち，一変数関数の和の形で表現できる多変数関数が分離可能関数である．

分離性の概念は1978年にKojima(1)により最初に導入され，その後Todd(2), Garcia and Zangwill(3), Saigal(4), Yamamura and Horiuchiら(5)-(29)により数値解析アルゴリズムの効率化に活用した研究が行なわれている． 分離性活用の効果は極めて劇的である．例えば，n 次元非線形関数の区分的線形近似は一般に単体分割上で行われるが，分離可能な関数に対してはその区分的線形近似を直方体分割上で行えることが示されている(1)．各直方体は $n!$ 個の単体を含むことから，分離性の活用により一つの直方体内に含まれる領域数は

$$n! \to 1$$

と大幅に減少されることがわかる．その他，分離性の活用により区分的線形近似の関数評価回数が $10^n \to 10$ (文献(19)), サインテスト施行回数が $2^n \to 2$ (文献(19)), 線形方程式の求解回数が100億 → 4,865 (文献(19), (22)), 数値微分における関数評価回数が $n+1 \to 2$ (文献(17)), 計算時間が数か月 → 数十秒，数百年 → 数十分 (文献(19)-(29)) になるなど，アルゴリズムの計算複雑度を飛躍的に改善できる例が多数報告され

ている. このように分離性の活用は,非線形システムを効率よく解析するための強力な武器となる可能性を秘めている.

しかし分離性自体は, 関数の性質としては比較的強い条件であり,そのため分離性活用の適用範囲は狭いものと解釈されてきた. (このことが, 分離性に関する研究が広く普及しなかった理由の一つとなっている.) 上記のアルゴリズムも,応用分野としては電子回路解析, 非線形計画問題, ニューラルネットワークなど, 分離可能な非線形方程式が直接派生する分野に限定されてきた. しかし分離性活用は上記のように非線形システム解析の計算複雑度を本質的に改善するアイデアであり,その適用範囲を拡張することは, 非常に重要な課題であるものと考えられる.

本論文では, 与えられた分離不可能な多変数関数を分離可能形で表現するアルゴリズムを提案する. すなわち, 実用レベルで現れる多変数関数 (四則演算, 単項演算, べき乗の組合せからなる関数) を補助変数の導入により一変数関数の和の形で表現するアルゴリズムを提案する. そのための基本概念として, 関数の計算過程を記述する有向グラフである計算グラフを利用する. 本論文のアルゴリズムを用いれば, ほとんどの実用的クラスの分離不可能な非線形関数を, 等価で分離可能な非線形関数に変換することができる. 従って上記の分離性を活用したアルゴリズムの適用範囲を大幅に拡張することが可能となる. また, 微分方程式や精度保証, 区間解析, ニューラルネットワークの性能評価, 非線形計画問題など, 理工学上の多くの分野で新しい可能性を開くことが期待される.

2. 基本的手順

紙面の都合から, 本論文ではアルゴリズムの概略だけを示す. 詳細については文献 (30) を参照されたい.

実用レベルで現れる多変数関数の大部分は, 式で書くことのできる関数, すなわち四則演算, 単項演算, べき乗の組合せからなる関数である. 本論文ではこのような関数を対象とする.

上記の関数の計算過程で現れる分離不可能な関数形には, 次のような三つのタイプがある. 但し, g および g^i は $\boldsymbol{R}^1$ から $\boldsymbol{R}^1$ への関数とする.

[関数形 **1**] 乗除算 $g^1(x_1) \times \cdots \times g^k(x_k)$
(例: x_1x_2, x_1/x_2)

[関数形 **2**] 単項演算
$g(g^1(x_1) + \cdots + g^k(x_k))$
(例: $\cos(x_1+x_2)$, $(x_1+x_2)^2$)

[関数形 **3**] べき乗 $g^1(x_1)^{g^2(x_2)}$
(例: $(x_1+1)^{x_2}$)

関数形 1 の関数は, 次のようにして分離形に変換することができる. いくつかの具体例を示す. 但し, y_i は補助変数である.

例 1 $\quad x_1x_2$

$$\longrightarrow \ (y_1^2 - x_1^2 - x_2^2)/2$$
$$y_1 = x_1 + x_2$$

例 2　$x_1x_2x_3$

$$\longrightarrow\ y_2x_3$$
$$y_2 = x_1x_2$$
$$\longrightarrow\ (y_3^2 - y_2^2 - x_3^2)/2$$
$$y_1 = x_1 + x_2$$
$$y_2 = (y_1^2 - x_1^2 - x_2^2)/2$$
$$y_3 = y_2 + x_3$$

例 3　$x_1x_2x_3x_4$

$$\longrightarrow\ y_4x_4$$
$$y_4 = x_1x_2x_3$$
$$\longrightarrow\ (y_5^2 - y_4^2 - x_4^2)/2$$
$$y_1 = x_1 + x_2$$
$$y_2 = (y_1^2 - x_1^2 - x_2^2)/2$$
$$y_3 = y_2 + x_3$$
$$y_4 = (y_3^2 - y_2^2 - x_3^2)/2$$
$$y_5 = y_4 + x_4$$

同様に, $x_1x_2\cdots x_k$ は $2k-3$ 個の補助変数の導入により分離形に変換することができる.

但し, 常に $g^i(x_i) > 0$ が成り立つ場合は, 次のように一つの補助変数で分離形に変換することができる.

例 4　$(x_1^2+1)(x_2^2+2)$

$$\longrightarrow\ \exp(y_1)$$
$$y_1 = \log(x_1^2+1) + \log(x_2^2+2)$$

例 5　$g^1(x_1)g^2(x_2)\cdots g^k(x_k)$

$$\longrightarrow\ \exp(y_1)$$
$$y_1 = \sum_{i=1}^{k} \log(g^i(x_i))$$

次に, 関数形 2 は, 次のようにして分離形に変換することができる.

例 6　$(x_1+x_2)^2$

$$\longrightarrow\ y_1^2$$
$$y_1 = x_1 + x_2$$

例 7　$\sin(x_1 + x_2^2 + x_3^3)$

$$\longrightarrow\ \sin(y_1)$$
$$y_1 = x_1 + x_2^2 + x_3^3$$

例 8　$(\cos(x_1^2 + x_2^2))^2$

$$\longrightarrow\ (\cos(y_1))^2$$
$$y_1 = x_1^2 + x_2^2$$

この場合, 分離形への変換に必要な補助変数の数は 1 個である.

また関数形 3 は, $x_1 > 0$ を仮定すれば次のように変換できる.

例 9　$x_1^{x_2}$

$$\longrightarrow\ \exp(y_2)$$
$$y_2 = x_2\log(x_1)$$
$$\longrightarrow\ \exp(y_2)$$
$$y_1 = \log(x_1) + x_2$$
$$y_2 = y_1^2 - (\log(x_1))^2 - x_2^2$$

このときの補助変数の数は 2 個であるが, さらに $x_1 > 1$, $x_2 > 0$ を仮定すれば, 以下に示すように補助変数は 1 個ですむ.

例 10　$x_1^{x_2}$

$$\longrightarrow\ \exp(\exp(y_1))$$
$$y_1 = \log(\log(x_1)) + \log(x_2)$$

本論文で提案するアルゴリズムは, 与えられた多変数関数の各計算過程で, 以上述べた分離形への変換を自動的に行なうものである.

3. アルゴリズムの概略

本論文で対象とする四則演算, 単項演算, べき乗の組合せからなる関数は, 一般にその計算過程を計算グラフで表すことができる[31]. 関数を計算グラフで記述することにより、その構造と計算過程を明確にすることができる. その各計算過程で, 2章で示した三つのタイプの変換を行ない, それにより関数全体を分離形へと変換する.

ここで, 各計算過程における演算を, 次のような五つの形に分類する. ここで c は定数, g は単項演算, f_L, f_R は部分計算グラフ（一つの定数頂点のみからなる場合を除く）を表すものとする.

(i) $f = f_L + f_R,\ \ f = f_L - f_R$

(ii) $f = c * f_R,\ \ f = f_L * c,\ \ f = f_L / c$

(iii) $f = c / f_R,\ \ f = f_L^c,\ \ f = c^{f_R}$

(iv) $f = f_L * f_R,\ \ f = f_L / f_R$

(v) $f = g(f_L)$

(vi) $f = f_L^{f_R}$

それぞれに対する分離形への変換の基本的手順は以下の通りである.

1. (i) または (ii) の場合

このとき, f_L と f_R が分離可能であれば, f は分離可能となる. f_L と f_R が分離可能でない場合は, f_L と f_R をそれぞれ分離形に変換することにより, f を分離形に変換できる.

2. (iii) の場合

このとき, f_L あるいは f_R が一つの変数のみからなる関数であれば, f は分離可能となる. また二つ以上の変数を含む場合も, 補助変数を導入して

$$
\begin{aligned}
f = c/f_R \quad &\longrightarrow \quad f = c/y_1 \\
&\qquad\quad\ \ y_1 = f_R \\
f = f_L^c \quad &\longrightarrow \quad f = y_1^c \\
&\qquad\quad\ \ y_1 = f_L
\end{aligned}
$$

と置けば, あとは f_L, f_R を分離形に変換することにより, f 全体が分離形となる.

3. (iv) の場合

このとき, f_L と f_R の両方が一つの変数からなる関数であれば, これは2章で述べた関数形1になる. また, f_L と f_R の少なくとも一方が二つ以上の変数を含む場合は, 次のように補助変数を導入することにより, 関数形1に変換できる.

$$
\begin{aligned}
f = f_L * f_R \quad &\longrightarrow \quad f = y_1 * y_2 \\
&\qquad\quad\ \ y_1 = f_L \\
&\qquad\quad\ \ y_2 = f_R
\end{aligned}
$$

関数形1は, 2章で示したように

$$
\begin{aligned}
f = y_1 * y_2 \quad &\longrightarrow \quad f = (y_3^2 - y_1^2 - y_2^2)/2 \\
&\qquad\quad\ \ y_3 = y_1 + y_2
\end{aligned}
$$

と変換できるので, あとは f_L と f_R のそれぞれを分離形に変換すれば, 関数全体を分離形に変換することができる.

4. (v) の場合

f_L が一つの変数からなる場合は, f はそのままで分離可能である. f_L が二つ以上の変数からなる場合は, 関数形2となるので,

$$f = g(f_L) \quad \longrightarrow \quad f = g(y_1)$$
$$y_1 = f_L$$

とおく．後は f_L を分離形に変換すれば，全体として分離可能な形になる．

5．(vi) の場合

f_L と f_R の両方が一つの変数からなる場合は，f は関数形 3 となる．f_L と f_R が二つ以上の変数からなる場合も，

$$f = f_L^{f_R} \quad \longrightarrow \quad f = y_1^{y_2}$$
$$y_1 = f_L$$
$$y_2 = f_R$$

と置くことにより，関数形 3 に変換できる．関数形 3 は 2 章で示したように

$$f = y_1^{y_2} \quad \longrightarrow \quad f = \exp(y_4)$$
$$y_3 = \log(y_1) + y_2$$
$$y_4 = y_3^2 - (\log(y_1))^2 - y_2^2$$

と変換できるので，f_L と f_R をそれぞれ分離形に変換すれば，関数全体を分離形に変換できる．

以上の説明で，「f_L と f_R を分離形に変換すれば」という表現がいくつかでたが，f_L あるいは f_R を f とみなせば，その極大頂点部は (i) ～ (vi) のいずれかとなる．従って 1. ～ 5. を f_L と f_R について繰り返すことにより，もとの関数 f 全体を分離形に変換することができる．

以上より，分離形への変換は，計算グラフの頂点を極大頂点からトップダウン的に探索しながら各部分を変換していけばよいことがわかる．この頂点の探索では，まず頂点の演算が (i) ～ (vi) のうちどれに該当するかを判別する必要がある．しかし演算の種類を調べるだけでは，(ii), (iii), (iv) の乗算，除算を区別することはできない．そこである部分計算グラフが変数を含まない（定数と演算だけを含む）かどうかを容易に判別できるように予め設定しておく必要がある．また，f_L と f_R が一つの変数だけを含む場合，そのことも判別できるように設定しておく必要がある．そこで，このような判別を容易にできるようにするため，頂点に次のようなラベル付けを行なう．

まず，すべての変数頂点 x_i $(i = 1, \cdots, n)$ には i，定数頂点には C，演算が "+" または "−" である演算頂点には S，その他の頂点には N をラベル付けする．ここで S は「分離可能 (separable) な関数形」，N は「分離不可能 (nonseparable) かもしれない関数形」を表す．但し，その後のラベルの更新により，N は「分離不可能な関数形」を表すようになる．

次に，変数頂点を含まない部分計算グラフ（すなわち定数頂点と演算頂点のみを含む部分計算グラフ）の極大頂点のラベルを C とし，また x_i 頂点だけしか変数頂点を含まない部分計算グラフの極大頂点のラベルを i とする．（詳細は文献 (30) 参照．）

このようにラベル付けすれば，ある部分計算グラフが変数を全く含まないか，一つの変数のみを含むか，あるいは二つ以上の変数を含むかをその極大頂点のラベルによって判別することができる．

このようなラベル付けを行なった後のアルゴリズムの概略は次の通りである．

[Step 1]　f が (iii) ～ (vi) の関数形のいずれかの場合，f_L，f_R を補助変数に置いて計算グラフを分解する．これを f_L，f_R に対して同様に繰り返す．

Step 1 実行後，計算グラフ内の分離不可能な関数形は関数形 1 と関数形 3 だけとなる．

[Step 2]　関数形 1 と関数形 3 の関数を，補助変数を導入して分離する．

Step 1 を実行した後，Step 2 を実行すれば，計算グラフ全体を分離形に変換できる．

この後の詳細については，文献 (30) を参照されたい．文献 (30) のアルゴリズムでは，さらにラベル P を導入し，補助変数の数を少なくする工夫行なっている．ここで P は「補助変数に置くことをペンディング (pending) し，その頂点を通り抜ける (pass through)」の意味である．

なお，このアルゴリズムの計算量は変換後の計算グラフの頂点数のオーダーとなることが示されている．

4. 実験例

本論文で提案したアルゴリズムを C 言語でプログラミングし，Sun S-4/1 ワークステーション上で実験を行なった．このプログラムでは，関数式を入力すれば，変換式，補助方程式，補助変数の数，変換前の計算グラフのデータと頂点数，変換後の計算グラフの頂点数が出力されるようになっている．

このアルゴリズムを多くの多変数関数に対して適用した．それらすべての場合について，分離形への変換が実現できた．以下に実験結果の例を示す．

$$
\begin{aligned}
& f = (x_1^2 + 3 * x_1) * x_2 - \sin(x_1 + x_2 * x_3) \\
&\quad + x_1 - \sin(x_4) \\
\rightarrow\ & f = (y_2^2 - (x_1^2 + 3 * x_1)^2 - x_2^2)/2 \\
&\qquad - \sin(y_1) + x_1 - \sin(x_4) \\
& y_1 = x_1 + (y_3^2 - x_2^2 - x_3^2)/2 \\
& y_2 = x_1^2 + 3 * x_1 + x_2 \\
& y_3 = x_2 + x_3
\end{aligned}
$$

$$
\begin{aligned}
& f = x_1^2 + 2/\sin(x_1) + x_1 * x_2 * x_3 - 3 * x_2 \\
&\quad + x_4^2/x_2 \\
\rightarrow\ & f = x_1^2 + 2\sin(x_1) + (y_3^2 - y_1^2 - x_3^2)/2 \\
&\qquad -3 * x_2 + (y_4^2 - (x_4^2)^2 - 1/x_2^2)/2 \\
& y_1 = (y_2^2 - x_1^2 - x_2^2)/2 \\
& y_2 = x_1 + x_2 \\
& y_3 = y_1 + x_3 \\
& y_4 = x_4^2 + 1/x_2
\end{aligned}
$$

$$
\begin{aligned}
& f = 5 * x_1/(x_2 + 5 * x_1 + x_3^2) + x_5^2 * x_1/3 \\
\rightarrow\ & f = (y_2^2 - (5 * x_1)^2 - 1/y_1^2)/2 \\
&\qquad + (y_3^2 - (x_5^2)^2 - x_1^2)/2/3 \\
& y_1 = x_2 + 5 * x_1 + x_3^2 \\
& y_2 = 5 * x_1 + 1/y_1 \\
& y_3 = x_5^2 + x_1
\end{aligned}
$$

$$
\begin{aligned}
& f = 5 * (x_1 + x_2 + x_3)/(x_1^{x_2} + x_3 + x_4 + x_5)^2 \\
&\quad + \cos(x_2) + 2 \\
\rightarrow\ & f = (y_4^2 - y_1^2 - 1/(y_2^2)^2)/2 + \cos(x_2) + 2 \\
& y_1 = 5 * (x_1 + x_2 + x_3) \\
& y_2 = \exp(y_3) + x_3 + x_4 + x_5 \\
& y_3 = (y_5^2 - \log(x_1)^2 - x_2^2)/2 \\
& y_4 = y_1 + 1/y_2^2 \\
& y_5 = \log(x_1) + x_2
\end{aligned}
$$

$$f = 2/x_1 - x_3^{\log(x_2)} - 4/(x_1 - x_2 - x_3^3)$$
$$\begin{aligned}\to\quad & f = 2/x_1 - \exp(y_2) - 4/y_1\\ & y_1 = x_1 - x_2 - x_3^3\\ & y_2 = (y_3^2 - \log(x_2)^2 - \log(x_3)^2)/2\\ & y_3 = \log(x_2) + \log(x_3)\end{aligned}$$

$$f = x_1 * x_2 * \sin(x_1 * \log(\cos(x_3) + 1) * \exp(x_1 * \tan(x_2 + x_3)))$$
$$\begin{aligned}\to\quad & f = (y_{10}^2 - y_1^2 - \sin(y_5)^2)/2\\ & y_1 = (y_6^2 - x_1^2 - x_2^2)/2\\ & y_2 = (y_7^2 - x_1^2 - \log(\cos(x_3) + 1)^2)/2\\ & y_3 = x_2 + x_3\\ & y_4 = (y_8^2 - x_1^2 - \tan(y_3)^2)/2\\ & y_5 = (y_9^2 - y_2^2 - \exp(y_4)^2)/2\\ & y_6 = x_1 + x_2\\ & y_7 = x_1 + \log(\cos(x_3) + 1)\\ & y_8 = x_1 + \tan(y_3)\\ & y_9 = y_2 + \exp(y_4)\\ & y_{10} = y_1 + \sin(y_5)\end{aligned}$$

$$f = (x_1 + x_2 + 2) * (x_1 - x_2)/4 - \sin(x_1)^2 + \exp(\log(x_1^{x_2})) + \cosh(x_3)$$
$$\begin{aligned}\to\quad & f = (y_4^2 - y_1^2 - y_2^2)/2/4 - \sin(x_1)^2 + \exp(\log(y_3)) + \cosh(x_3)\\ & y_1 = x_1 + x_2 + 2\\ & y_2 = x_1 - x_2\\ & y_3 = \exp(y_5)\\ & y_4 = y_1 + y_2\\ & y_5 = (y_6^2 - \log(x_1)^2 - x_2^2)/2\\ & y_6 = \log(x_1) + x_2\end{aligned}$$

また, これらの関数を掛け合わせたより複雑な関数に対してもアルゴリズムを適用した. その場合も, 分離形への変換が実現できた.

5. むすび

本論文では, 多変数関数を一変数関数の和の形で表現するアルゴリズムを提案した. 従来, 非線形方程式の数値解析の効率化は, 変数を消去して方程式系を小さくすることにより行なわれる場合が多かった. しかし非線形方程式は圧縮すると非線形性がより強くなるため, 計算時間はかえって増大してしまう場合が少なくない. 本研究の思想は, 補助変数を導入して逆に非線形方程式を大きくし, その特殊な構造を活用することにより, 最終的な効率化を目指すものである. このような思想は以前からあったが, 計算複雑度が NP の問題に対するアプローチとして, 今後ますます重要視されるものと予想される.

最後に, 分離性の研究を進めるにあたり, 以下の御指摘と御教示を東京大学の伊理正夫教授と早稲田大学の大附辰夫教授から独立に頂いた. 1900 年のパリにおける国際数学者会議で, ドイツの数学者ヒルベルト (D. Hilbert) が提起した有名な 23 個の問題のうち, 第 13 問題は次のようなものであった[32].

3 変数のすべての解析関数は, 2 変数の連続関数を合成して得られるであろうか.

この問題は 1950 年代後半にコルモゴロフ (A. N. Kolmogorov) とアーノルド (V. I. Arnol'd) により肯定的に証明された[33]. 彼らの証明によれば, n 次元立方体の上で定義された n 変数の任意の連続関数は, 1 変数の実数値連続関数 χ_i, φ_{ij} により

$$f(x_1, \cdots, x_n) = \sum_{i=1}^{2n+1} \chi_i\Big(\sum_{j=1}^{n} \varphi_{ij}(x_j)\Big)$$

の形で表現される. 換言すれば, この証明

は「n 変数の任意の連続関数は，高々 $2n+1$ 個の補助変数の導入により，常に 1 変数の関数とただ 1 種類の関数 — 加法 — とだけでことをすませることができる」ことを示している．

本研究の方向は，ヒルベルトの第 13 問題に対するコルモゴロフとアーノルドの存在証明に，実用レベルで具体的な構成アルゴリズムを与える形となっている．このような方向の研究は，デバイスモデルを組み込んだ回路解析におけるアルゴリズムの implementation の単純化や，どのような関数形が丸め誤差の意味で数値計算上安定となるかの判定などにおいて重要になるものと思われる（大附教授の御指摘による）．また，代数幾何学との関連も予想される（九州大学古賀利郎教授の御指摘による）．今後，詳細な検討を行なっていきたい．

謝辞　本研究を進めるに当たり，貴重な御指摘や激励の言葉を頂きました九州大学古賀利郎教授，東京大学伊理正夫教授，早稲田大学大附辰夫教授に感謝致します．また，日頃から御指導頂きます早稲田大学堀内和夫教授，大石進一教授に感謝致します．

文　献

(1) Kojima, M., “On the homotopic approach to systems of equations with separable mappings,” in *Math. Programming Study 7*, Balinski, M. L. and Cottle, R. W., eds. North-Holland, Amsterdam, the Netherlands, Feb. 1978, pp. 170-184.

(2) Todd, M. J., “Exploiting structure in piecewise-linear homotopy algorithms for solving equations,” *Math. Programming*, vol. 18, no. 3, pp. 233-247, May 1980.

(3) Garcia, C. B. and Zangwill, W. I., *Pathways to Solutions, Fixed-Points, and Equilibria*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

(4) Saigal, R., “A homotopy for solving large, sparse and structured fixed point problems,” *Math. Oper. Res.*, vol. 8, no. 4, pp. 557-578, Nov. 1983.

(5) 山村清隆，堀内和夫，“ニュートンホモトピーと直方体分割を用いた非線形回路網の直流解析法，” 信学論 (A), vol. J71-A, no. 9, pp. 1756-1759, Sep. 1988.

(6) 山村清隆，堀内和夫，“非線形回路解析におけるホモトピー法の収束性について，” 信学論 (A), vol. J72-A, no. 1, pp. 156-159, Jan. 1989.

(7) 福山健次郎，山村清隆，堀内和夫，“変数分離が不可能な非線形方程式に対する直方体法の適用，” 信学論 (A), vol. J72-A, no. 3, pp. 625-627, Mar. 1989.

(8) Yamamura, K. and Horiuchi, K., “Quadratic convergence of the homotopy method using a rectanglar subdivision,” *Trans. IEICE*, vol. E72, no. 3, pp. 188-193, Mar. 1989.

(9) Yamamura, K. and Horiuchi, K., “Solving nonlinear resistive networks by a homotopy method using a rectanglar subdivision,” *Trans. IEICE*, vol. E72, no. 5, pp. 584-594, May 1989.

(10) 山村清隆，落合信，“非線形方程式の変数分離可能性を利用した整数ラベリング法，” 信学論 (A), vol. J72-A, no. 11, pp. 1807-1813, Nov. 1989.

(11) Yamamura, K. and Kiyoi, M., “A piecewise-linear homotopy method with the use of the Newton homotopy and a polyhedral subdivision,” *Trans. IEICE*, vol. E73, no. 1, pp. 140-148, Jan. 1990.

(12) 山村清隆，福山健次郎，堀内和夫，“記憶のない離散的通信路に対する直方体分割を用いた通信路容量の計算法，” 信学論 (A), vol. J73-A, no. 3, pp. 576-583, Mar. 1990.

(13) Yamamura, K. and Horiuchi, K., "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.

(14) 山村清隆, 福山健次郎, 堀内和夫, "直方体分割を用いたホモトピー法による制約条件付通信路容量の計算," 信学論 (A), vol. J73-A, no. 7, pp. 1286-1289, July 1990.

(15) 山村清隆, 新井かおり, 清井雅広, "非線形計画問題に対する区分的線形ホモトピー法," 信学論 (A), vol. J74-A, no. 3, pp. 515-523, Mar. 1991.

(16) Yamamura, K., Katou, K. and Ochiai, M., "Improving the efficiency of integer labeling methods for solving systems of nonlinear equations," *Trans. IEICE*, vol. E74, no. 6, pp. 1463-1470, June 1991.

(17) 山村清隆, 清井雅広, "計算グラフを用いた非線形関数の分離性検出アルゴリズム," 信学論 (A), vol. J74-A, no. 12, pp. 1755-1765, Dec. 1991.

(18) Yamamura, K., "Exploiting separability in numerical analysis of nonlinear systems," *IEICE Trans. Fundamentals*, vol. E75-A, no. 3, pp. 285-293, Mar. 1992.

(19) Yamamura, K. and Ochiai, M., "An efficient algorithm for finding all solutions of piecewise-linear resistive circuits," *IEEE Trans. Circuits Syst.*, vol. 39, no. 3, pp. 213-221, Mar. 1992.

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

(21) Yamamura, K. and Kiyoi, M., "Detecting separability of nonlinear mappings using computational graphs," *IEICE Trans. Fundamentals*, vol. E75-A, no. 12, pp. 1820-1825, Dec. 1992.

(22) Yamamura, K., "Finding all solutions of piecewise-linear resistive circuits using simple sign tests," *IEEE Trans. Circuits Syst.*, vol. 40, no. 8, Aug. 1993.

(23) Yamamura, K., "Simple algorithms for tracing solution curves," *IEEE Trans. Circuits Syst.*, vol. 40, no. 8, Aug. 1993.

(24) Yamamura, K., "A simple algorithm for finding all solutions of piecewise-linear resistive circuits," *IEICE Trans. Fundamentals*, vol. E76-A, no. 10, pp. 1812-1821, Oct. 1993.

(25) Yamamura, K., "Piecewise-linear analysis of nonlinear resistive networks containing Gummel-Poon models or Shichman- Hodges models," *IEICE Trans. Fundamentals*, vol. E77-A, no. 1, Jan. 1994.

(26) Yamamura, K., "A sign test for finding all solutions of piecewise-linear resistive circuits," *IEICE Trans. Fundamentals*, vol. E77-A, no. 1, Jan. 1994.

(27) Yamamura, K., "Finding all solutions of piecewise-linear resistive circuits containing neither voltage nor current controlled resistors," submitted to *IEICE Trans. Fundamentals.*

(28) Yamamura, K., "Finding all solutions of piecewise-linear resistive circuits containing sophisticated transistor models," submitted to *IEEE Trans. Circuits Syst.*

(29) Yamamura, K., "A Katzenelson-like algorithm for solving nonlinear resistive networks," submitted to *IEICE Trans. Fundamentals.*

(30) 山村清隆, 村山泰子, "多変数関数を一変数関数の和で表現するアルゴリズム," 信学技報, CAS93-51, 52, NLP93-39, 40, pp. 67-82, June 1993.

(31) Iri, M., "Simultaneous computation of functions, partial derivatives and estimates of rounding errors — complexity and practicality —," *Japan J. Appl. Math.*, vol. 1, no. 2, pp. 223-252, 1984.

(32) 国沢清典, 梅垣寿春, "情報理論の進歩 — エントロピー理論の発展 —," 岩波書店, 1965.

(33) Tikhomirov, V. M. (ed.), *Selected Works of A. N. Kolmogorov: Volume I — Mathematics and Mechanics* —, Kluwer Academic Publishers, Dordrecht/Boston/London, 1991.