

模索過程と不動点アルゴリズム

長 尾 昭 哉
鈴 木 牧 雄
村 田 潔

ワルラス以来，競争均衡解の存在問題は，つねに経済学者の注目を集めてきた。特に，1950年代から60年代にかけて，マッケンジー，アロー，ドゥブリューなどが不動点定理を援用することによって，こうした問題に最終的解答を与えたことはよく知られている。

とはいっても，ブラウアー，又は角谷の不動点定理が，その証明において，non-constructive aspect を含んでいる⁽¹⁾のが惜しまれるところであった。実際，ワルラス自身，素朴な方法で均衡解の存在を明らかにした際，解に到達する現実の過程について別に議論する必要を感じ，tâtonnement のアイディアを提出したのである。その意味でスカーフの新しい不動点定理が，必要な精度において近似解を与える手続きを構成しているのは，画期的な意味を有する⁽²⁾。そしてその後の，クーン，ハンセン，マッキノン，メリル，イーヴスなどによる不動点アルゴリズムの展開⁽³⁾が，一部の理論経済学者の興味をひいているのも理由のあることである。

(1) 文献 [9], pp. 12

(2) 文献 [9], pp. 76

(3) 文献 [11], pp. 340

* 小島政和 (東京工大)，酒井泰弘 (筑波大) の両氏から文献に関する貴重な御教示を頂いたことに深く感謝します。もちろん，本稿の欠陥が筆者の責任であることはいうまでもありません。

均衡解を計算する実行可能な方法が開発されたことが、経済理論にもたらすであろうインパクトについては、スカーフ自身の展望があり⁽⁴⁾、マッケンジーのレクチュア⁽⁵⁾もあるのでここに繰り返す必要はない。われわれが注意したいのは、クーンが pivotal method と名付けている⁽⁶⁾ことからわかるように、不動点アルゴリズムが、 $P_i \rightarrow P_{i+1}$ という集合列を創出する反復法に他ならないことである (P は後述のプリミティブ・セット “primitive set” で、線形計画法における基底に相当する)。その意味で、サミュエルソン以来使われてきた微分方程式 (discrete version では定差方程式) アプローチに加えて、模索過程を具体化する新たなでたでが経済学者の自由に供されることになったとも言えよう。

本稿は、不動点アルゴリズムを working tool として導入することによって、ワルラスに始まる「模索過程」の分析を拡張しようとする試みの、一つの予備報告である。具体的にいえば価格調整の微分方程式モデルを、「呼び人」“price caller” の行動を記述するものと解釈する伝統にならって⁽⁷⁾、不動点アルゴリズムをいわば一種の sophisticated price-calling と見ること、又、模索の様式、つまり「呼び人」のスキームと需給対応の若干のトポロジカルな特質との適合関係を調べることである。

I

exposition by cases

本稿では、不動点アルゴリズムのいくつかのタイプを、price calling scheme として順次とり上げる。具体的にいえば、価格空間 (I_a では1次元、 I_b では

(4) 文献 [9], 第1章, 文献 [10]

(5) 文献 [6]

(6) 文献 [4]

(7) 文献 [7]

2次元の単体) 上にコールされる価格の足取りとして、アルゴリズムをイラストレイトする。

I a two-goods economy

森嶋のテキスト “Walras’ Economics” にある、小さな数値例⁽⁸⁾を最初に取り上げよう。図 1 には (森嶋自身が示している図と同じ意味のものであるが)、われわれが naive scheme と呼ぶ価格調整過程 (次の定差方程式に示される「呼び人」の行動) の結果が示されている。

この方法では、ある価格 (ベクトル) に対応する超過需要 (ベクトル) に、単純に反応して次のコールが決められるのであるから、リミティング・サイクルが生ずるのは当然なのである。

$$P_1(t+1) = \left(1 + \max \left[0, 6 - \frac{P_1(t)}{[1 - P_1(t)]^2} [1 - 1.6 P_1(t)], 0 \right] \right)^{-1}$$

定差方程式を適当に微分方程式に変換すれば、——一定の安定条件を満しているとき—— calling は収束する。森嶋はこのことを、模索過程は caller’s skill

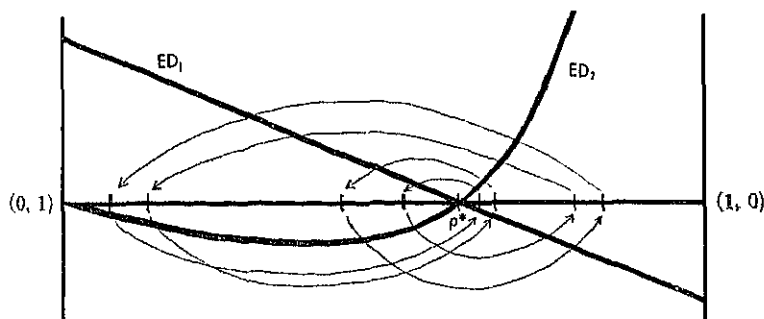


図 1 森嶋の例

(8) 文献 [7], pp. 41

を想定している，というように表現している。しかし，われわれの文脈でいえば，（本来的に離散的な）コール・スキームを improve するという問題は，むしろ価格調整の時間間隔をゼロに近づけるという技巧によって，かくされてしまうのである（これは反復“iteration”のステップ数を限りなく大きくすることを意味する）。

スカーフ，ハンセンによる最初の不動点アルゴリズム——固定格子法“fixed grid method”——の calling⁽⁹⁾ はコーナーから出発し，超過需要の大きさに依存しないステップサイズによって（つまり格子に従って）進む。そして，超過需要が正である間は進み続け，符号が変わったところでストップする。このようにすれば——超過需要の configuration が一定の条件を満たしている限り——均衡価格をはさむ2つのコール（プリミティブ・セット）が与えられることは明らかである。要は，前もって定められたステップ・サイズ（固定格子）によって，最後に均衡点を保持するコール（一般には n 本の価格ベクトル）相互間の距離をコントロールすることにある。



図 2 固定格子法

この方法の泣き所は，ある expedition を終って，近似の程度を高めようとすると，一たんコーナーに帰ってやり直しをしなければならないことである。具体的にいえば，より精密な近似をえるためには，格子サイズを変更しなければならないわけであるが，その際，近似解の近傍から再スタートすることが出来ない。

これにくらべて，ホモトピー法⁽¹⁰⁾は，システムティックに格子サイズを縮

(9) 文献 [9]

(10) 文献 [1], [2]

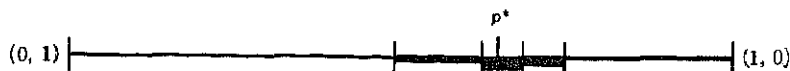


図 3 ホモトピー法

小する巧妙な方法である。これをこの経済における calling として実行したとすれば、次のような経過になる。呼び人は、最初に全区間の中点をコールし、トレーダーの反応を見る。もし（第 1 財の）超過需要が正であれば、彼は右の半分の区間を取り、その中点を次にコールする。再び反応をしらべて、右又は左の半区間から中点を取る。このようにすれば、——多次元の価格空間の場合、必ずしも単調にではないが——格子サイズを次第に小さくし、近似を必要に応じて精密化してゆけることは確実である。

クーンとマッキノンのサンドウィッチ法⁽¹¹⁾——われわれはクーンに従って再スタート法“restart method”という——は、スカーフ、ハンセンの固定格子法に、近似点の近くから再スタートするための、実に巧妙な手段をつけ加えたものである。その着想の秘密は、価格空間を次元の一つだけ高い単体に埋め込む——つまりサンドウィッチする——ことにあり、それについては後述する。

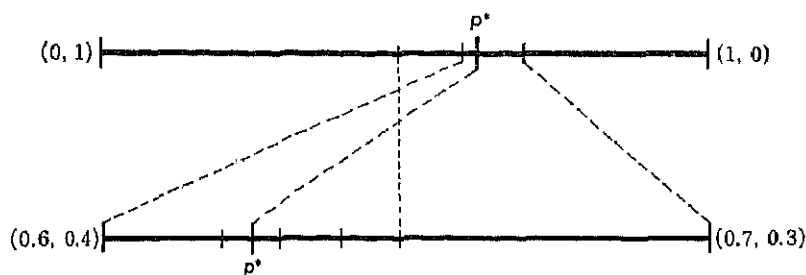


図 4 再スタート法

(11) 文献 [5]

I b three-goods economy

2次元の価格空間における expedition——均衡の模索過程——として、pivotal method のはたらきを見るために、われわれは次のようなケースを取り上げる。(1) ordinary case⁽¹²⁾, (2) Scarf's "global instability" case⁽¹³⁾, (3) highly perplexing case⁽¹⁴⁾ の3つである。

まず、弱公理を満たす、正常な需給市場について、ホモトピー法を実行すると、図 5-1, 2, 3 のように、一進一退しながら模索を進めてゆく。

次にスカーフの反例としてよく知られている、不安定ケースをコールしてみよう。定差方程式による模索の場合は、予想される通り、どの点から出発しても目標をそれてしまう(図 6)。森嶋の意味での熟練した「呼び人」の行動を記述するという微分方程式に変換すれば、均衡から離れていくことはなくなる。しかし、均衡に近づくわけでもない。

固定格子法によれば、図 7 の示すように、しばらくは単体の辺にそって進み、次にほぼまっすぐに均衡価格に向う。

再スタート法の場合は、若干の小さな回り道をくり返ししながら、格子サイズを変更し、精度を高めてゆく(図 8)。この場合には、ホモトピー法が図 9 から分るように最適である。

$$(12) \quad ED(1) = -6 + \frac{P(3)^2}{2P(1)^2} + \frac{P(2)P(3)\{24P(1)+48P(2)-P(3)\}}{4P(1)\{P(1)\cdot P(2)+P(2)P(3)+P(3)P(1)\}}$$

$$ED(2) = -12 + \frac{P(3)^2}{2P(2)^2} + \frac{P(1)P(3)\{24P(1)+48P(2)-P(3)\}}{4P(2)\{P(1)P(2)+P(2)P(3)+P(3)P(1)\}}$$

$$ED(3) = \frac{-(P(1)+P(2))P(3)}{2P(1)P(2)} + \frac{24P(1)P(2)\{P(1)+2P(2)\} + \{P(1)+P(2)\}P(3)^2}{4P(3)\{P(1)P(2)+P(2)P(3)+P(3)P(1)\}}$$

(13) 文献 [8]

(14) 近く別に報告する

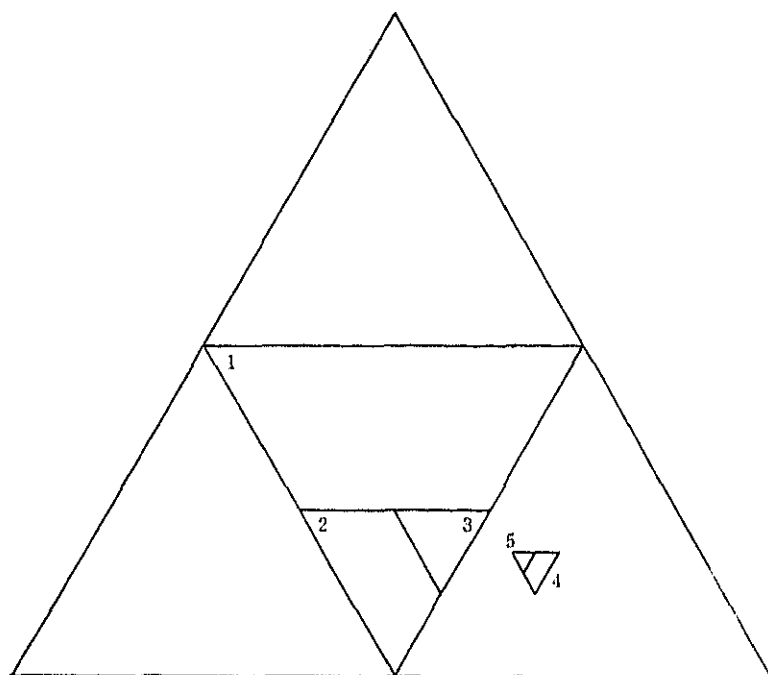


図 5-1 ordinary case のホモトピー法による探索

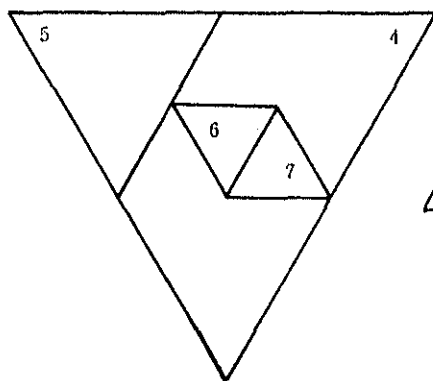


図 5-2 続き

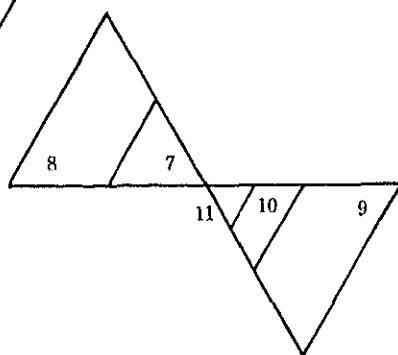


図 5-3 続き

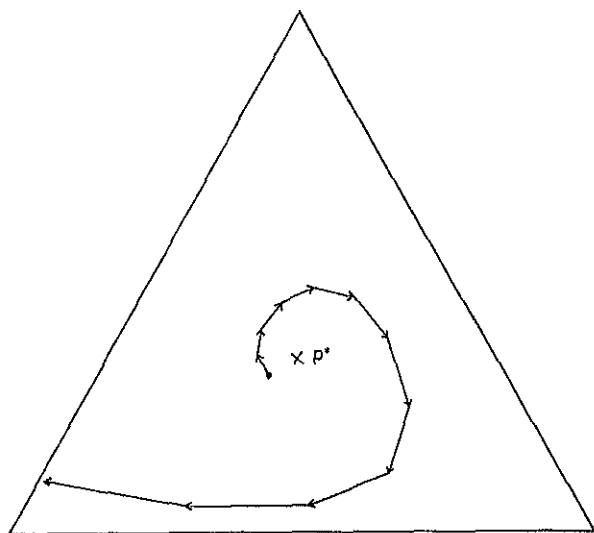


図 6 不安定ケースの定差方程式による模索

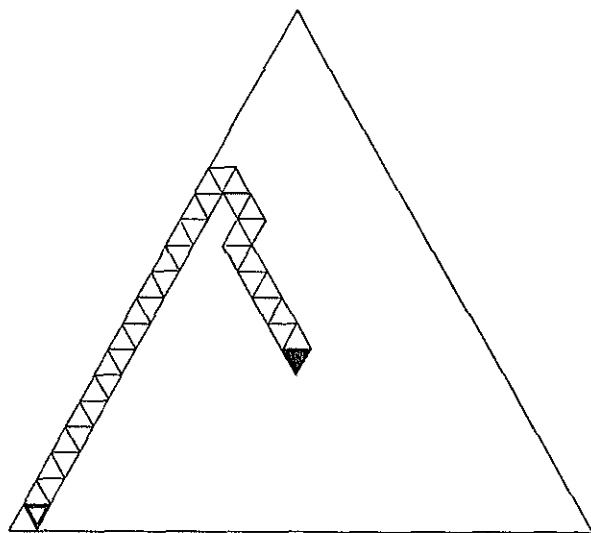


図 7 不安定ケースの固定格子法による模索

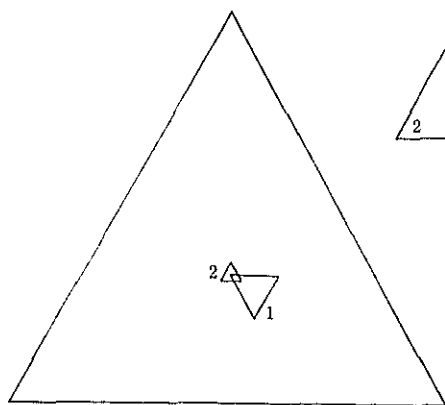


図 8-1 不安定ケースの再スタート法による探索

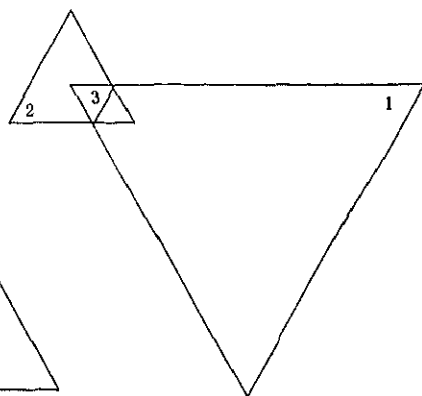


図 8-2 続き

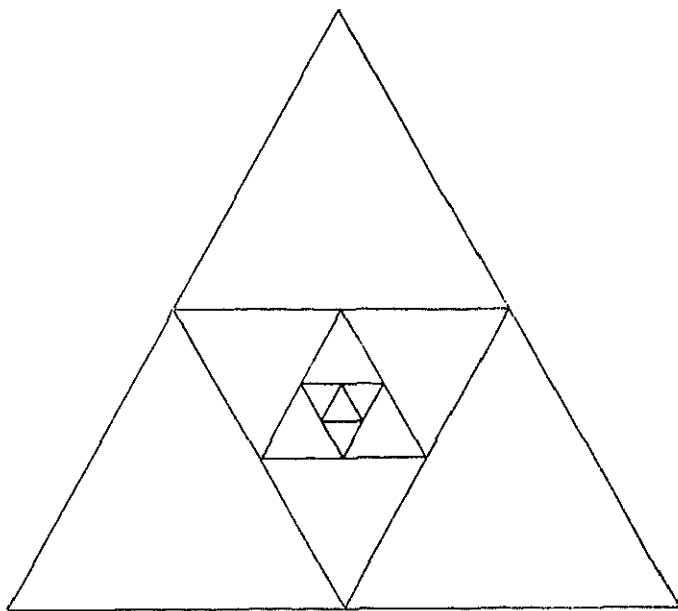


図 9 不安定ケースのホモトピー法による探索

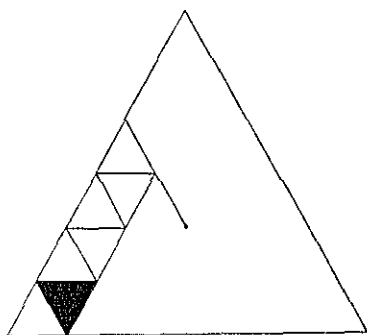


図 10-1 perplexing case の再スタート法
による模索

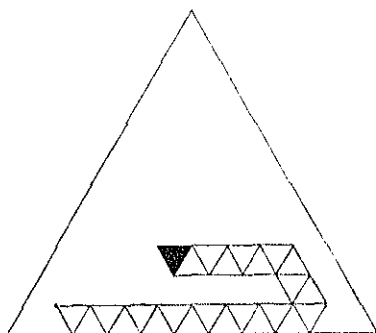


図 10-2 続き

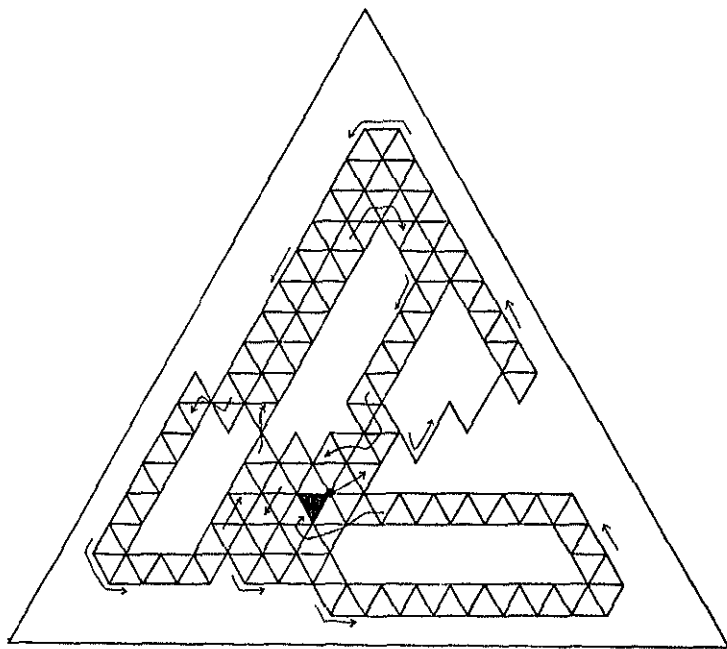


図 10-3 続き

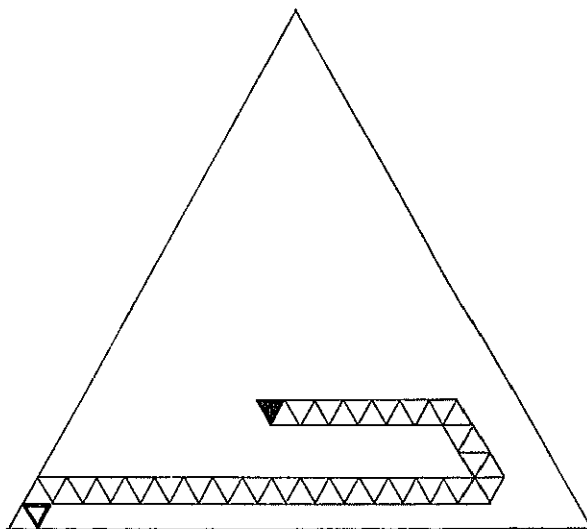


図 11 perplexing case の固定格子法による模索

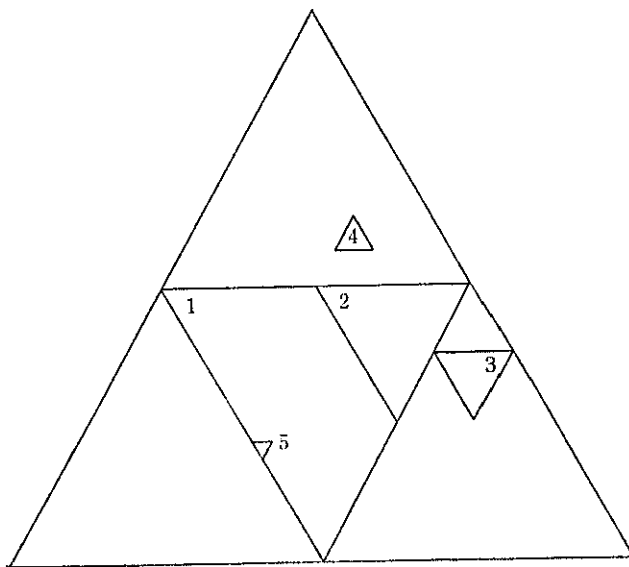


図 12 perplexing case のホモトピー法による模索

ひねくれたトレーダーのケースについて、再スタート法はトラブルを起す(図 10)。つまり、最終的に必ず均衡解を近似するという点では、クーン、マッキノンの論文に保証されている性質を実際に示すのであるが、格子サイズを変更するごとに、せっかくなつかんだ近似解からそれてしまい、遠い回り道をして、やっと目標に達する。この意味で、同法がスカーフ、ハンセンのオリジナルな方法(図 11)にくらべて優れている、再スタートのメリットをなくしてしまうのである。このような、強い意味での不安定をひき起す位相的な性質を確かめ、分類することを次の課題にしたいと思う⁽¹⁶⁾。この例については、ホモトピー法の優位が明らかで、トラブルは生じない(図 12)。

II

basic concepts

前節でわれわれは、不動点アルゴリズムによる価格調整を *sophisticated calling* と呼んだ。その意図は、この方法が近似解法として *gradient* 法や *Newton-Raphthton* 法と異っている特質を強調することにあった。それは、単体分割 “*simplicial subdivision*” の着想をうけつぐ正規格子 “*regular grid*” を導くラベル付け “*labelling*” の方法にある。以下、これらの概念を要約すると共に、循環辞書式順序 “*cyclic lexicographic ordering*” などにもふれる。

II a

triangulation

本稿では、 R^n 内の単体 “*simplex*” S からそれ自身の中への連続写像について不動点を近似的に計算することを考えるものとする。さしあたり、 S の中にあるベクトルのある大きなリストが与えられているものとしよう。リスト

(15) 近く別に報告する

内のベクトルはお互いに同じ第 i 成分をもたないものとする。

いま, これらのベクトルの部分集合

$$V = \{x^{j1}, x^{j2}, \dots, x^{jn}\}$$

に対して,

$$\begin{aligned} x_1^j &> \min [x_1^{j1}, x_1^{j2}, \dots, x_1^{jn}] \\ &\vdots \\ x_n^j &> \min [x_n^{j1}, x_n^{j2}, \dots, x_n^{jn}] \end{aligned}$$

という条件を満たす $x^j \in S$ が存在しないとき, V をプリミティブ・セットと定義する。

実際には, 正規格子ベクトルでアルゴリズムをすすめていくが, この場合当然なこととして同じ第 i 成分をもつベクトルを扱うことになる。そこで, これを解決するためには, 適当な退化処理法 “tie-breaking rule” が必要になってくる。まず, 分母 D の正規格子のベクトルを

$$\begin{aligned} x &= (m_1/D, m_2/D, \dots, m_n/D) \\ m_i &\geq 0 \text{ for } i=1, \dots, n \quad \sum_i m_i = D \end{aligned}$$

と表わすことにする。このとき, S 上の任意のベクトル

$$\begin{aligned} x' &= (m_1'/D, m_2'/D, \dots, m_n'/D) \\ x'' &= (m_1''/D, m_2''/D, \dots, m_n''/D) \end{aligned}$$

について, x' が x'' よりも辞書式により大 ($x' >_L x''$) ということを,

$$(m_1' - m_1'', \dots, m_n' - m_n'', m_1' - m_1'', \dots, m_{i-1}' - m_{i-1}'')$$

の最初の 0 でない成分が正のとき, と定義し, この順序付けを辞書式順序と呼ぶ。

Example:

$$x' = (.26, .18, .33, .11, .12)$$

$$x'' = (.15, .20, .33, .20, .12)$$

上の二つのベクトルを考えると, $x_s' <_L x_s''$ 。なぜなら, $x_s' - x_s'' = 0$, $x_1' - x_1'' < 0$ 。また, $x_s' >_L x_s''$ 。なぜなら, $x_s' - x_s'' = 0$, $x_1' - x_1'' > 0$ 。

プリミティブ・セットは, 辞書式順序を退化処理法とすることにより, 正規格子をベクトルのリストとしても同様に定義できる。プリミティブ・セットは正規格子を頂点とした単体分割となる。

では, この場合のプリミティブ・セットはどのようなものであろうか。今, n 個のベクトルの分子だけ (分母は D) の行列を

$$M = \begin{bmatrix} m_1^1 & \cdots & m_1^n \\ \vdots & & \vdots \\ m_n^1 & \cdots & m_n^n \end{bmatrix}$$

と表わし, $I(j)$ を, 整数 $1, \dots, n$ によって表示された第 j 列目のベクトルに対応する順列 “permutation” とする。このとき, M が,

$$m_i^j = m_i^{j-1} + 1 \quad i = I(j) - 1 \quad (\text{ただし, } I(j) = 1 \text{ の時, } i = n)$$

$$m_i^j = m_i^{j-1} - 1 \quad i = I(j)$$

$$m_i^j = m_i^{j-1} \quad \text{その他}$$

$$\text{for } j = 1, \dots, n \quad (\text{ただし, } j = 1 \text{ の時, } j-1 = n \text{ とする})$$

という条件⁽¹⁶⁾を満たすならば, M/D はプリミティブ・セットとなり, このような組み立てを正規分割スキーム “triangulation scheme” という。

Example:

$$D=100, \quad n=5$$

ベクトル $(10, 20, 31, 9, 30)'$ と順列 $I(j) = (2, 5, 3, 1, 4)$ が与えられたとき, このベクトルを第 1 列とするプリミティブ・セットは次のようになる。

(16) クーンが示した条件は, $+1$ と -1 の位置が, これとは丁度逆になっている。
文献 [3]

$$\begin{bmatrix} 10 & 10 & 10 & \underline{9} & 9 \\ \underline{20} & 20 & 21 & 21 & 21 \\ 31 & 31 & \underline{30} & 30 & 31 \\ 9 & 10 & 10 & 10 & \underline{9} \\ 30 & \underline{29} & 29 & 30 & 30 \end{bmatrix}$$

各行の最小値に下線を引いたが、その行の成分がそこから右に向って辞書式により大となっていることがわかる。

II b labelling (scalar label)⁽¹⁷⁾

ラベル付けとは、 S 内のベクトルに、整数 $1, \dots, n$ を何らかの規則で対応させることである。ところで、Sperner's Lemma は、単体分割の各頂点に、次に述べるラベル付け（好ましいラベル付け “proper labelling”）がなされると、完全にラベル付けされた “completely labeled”（各頂点が異なってラベル付けされた）部分単体が奇数個取れることを示している。実際、適当な規則でラベル付けし、完全にラベル付けされたプリミティブ・セットを見つけることが、不動点アルゴリズムに他ならない。

x^j を S 内の任意のベクトル、 $f(x)$ を $S \rightarrow S$ の連続写像とすると、 x^j のラベルを $L(x^j)$ として、

$$\text{a) } f_i(x^j) - x_i^j \geq 0 \quad (\text{i})$$

あるいは逆に

$$\text{b) } f_i(x^j) - x_i^j \leq 0 \quad (\text{ii})$$

という条件を満しているとき、 $L(x^j) = i$ とすればよい。（このような i が複数個あるときは、最小の i をとっても良いし、

(17) 文献 [9], 第3章, [5]

$$a) \max_i [f_i(x^j) - x_i^j]$$

または,

$$b) \min_i [f_i(x^j) - x_i^j]$$

となる i をとつても、やはり好ましいラベル付けである。) またスラック・ベクトル “slack vector” については,

$$a) L(x^j) = i \quad \text{ただし, } x_i^j = 0 \quad (i)'$$

$$b) L(x^j) \neq i \quad \text{ただし, } x_i^j = 0 \quad (ii)'$$

とすればよい。

S 上の格子の目を次第に細かくしていくと、完全にラベル付けされたプリミティブ・セットの列は、極限において一つのベクトル x^* に収束するであろう。このとき、写像の連続性から x^* はすべての i に関して、(a) の場合)

$$f_i(x^*) \geq x_i^*$$

である。ところが,

$$\sum_i f_i(x^*) = \sum_i x_i^*$$

であるから,

$$f_i(x^*) = x_i^* \quad \text{for all } i$$

したがって、 x^* は不動点であることがわかる。

実際、写像が連続であり、 S がコンパクトであるから、 x, x' をそれぞれメッシュ “mesh” が十分に細かいときの完全にラベル付けされたプリミティブ・セットの頂点とすると、ある $\varepsilon > 0$ に対し

$$|x' - x| \leq \delta$$

のとき

$$|f(x') - f(x)| \leq \varepsilon$$

を満たす $\delta > 0$ をとることができる。ここで

$$f_i(x') - x_i' \geq 0$$

故に,

$$\begin{aligned} f_i(x) - x_i &= \{f_i(x) - f(x')\} + \{x_i' - x_i\} + \{f_i(x') - x_i'\} \\ &\geq -\varepsilon - \delta \end{aligned}$$

さらに

$$\begin{aligned} f_i(x) - x_i &= -\sum_{j \neq i} \{f_j(x) - x_j\} \\ &\leq (n-1)(\varepsilon + \delta) \end{aligned}$$

したがって

$$|f(x) - x| \leq (n-1)(\varepsilon + \delta)$$

II c

labelling (vector label)⁽¹⁸⁾

前節のラベルはスカラーであったが、ここでは、 S 上のベクトル x^j に、ベクトル a^j を対応させるラベル付けについて述べる。このようなラベル付けの場合、 $b \geq 0$ とし、 $Ay=b$ の非負解が有界であるとする、対応するラベルベクトル (a^1, \dots, a^n) が $Ay=b$ の実行可能基底 “feasible basis” となっているプリミティブ・セットが存在する。これは、前に述べた完全にラベル付けされたプリミティブ・セットの一般化である。好ましいラベル付けの一例を示しながら話を進める。

A と b を

- a) スラック・ベクトル: x^j の第 i 成分が 0 のとき、 a^j を第 i 単位ベクトルとする。
- b) インテリア・ベクトル “interior vector”: $S \rightarrow S$ の点対集合優半連続写像 $x \rightarrow \Phi(x)$ を考え、 $\varphi^j = (\varphi_1^j, \dots, \varphi_n^j)'$ を $\varphi^j \in \Phi(x^j)$ の任意のベクトルとする。このとき、 x^j には $a^j = (\varphi_1^j - x_1^j + 1, \dots, \varphi_n^j - x_n^j + 1)'$ を対応さ

(18) 文献 [9], 第 4 章

せる。

$$c) \quad b = (1, \dots, 1)'$$

というルールで定める。このとき、完全にラベル付けされたプリミティブ・セット (x^j, \dots, x^n) が、不動点の近似値を表わすことを示す。

$$A = \begin{bmatrix} 1 & \cdot & 0 & \varphi_1^{n+1} - x_1^{n+1} + 1 \cdots \varphi_1^k - x_1^k + 1 \\ \cdot & \cdot & \cdot & \vdots \\ 0 & \cdot & 1 & \varphi_n^{n+1} - x_n^{n+1} + 1 \cdots \varphi_n^k - x_n^k + 1 \end{bmatrix}$$

とすると、 $Ay = b$ は、

$$\begin{aligned} y_1 & \cdot \quad + \sum_{j=n+1}^k y_j (\varphi_1^j - x_1^j + 1) = 1 \\ & \cdot \quad \vdots \\ & \cdot \quad y_n + \sum_{j=n+1}^k y_j (\varphi_n^j - x_n^j + 1) = 1 \end{aligned}$$

$$y_i \geq 0 \quad j \neq i \quad (\text{for } i=1, \dots, n) \quad \text{のとき等号成立}$$

と表わせる。次に、今迄のように格子サイズを十分大きくしていくと、極限では、

a) 実行可能基底を形成するプリミティブ・セットは一つのベクトル \hat{x} に収束する。

b) それらのインテリア・ベクトルに対応する A は、 $(\hat{\phi}_1^j - \hat{x}_1 + 1, \dots, \hat{\phi}_n^j - \hat{x}_n + 1)$ に収束する。

c) 正のウェイト y は非負のウェイト \hat{y} に収束する。

これを、上と同じように表わすと、

$$\begin{aligned} \hat{y}_1 & \cdot \quad + \sum_i \hat{y}_i (\hat{\phi}_1^i - \hat{x}_1 + 1) = 1 \\ & \cdot \quad \vdots \\ & \cdot \quad \hat{y}_n + \sum_i \hat{y}_i (\hat{\phi}_n^i - \hat{x}_n + 1) = 1 \end{aligned} \tag{iii}$$

$$\hat{y}_i \geq 0 \quad \hat{x}_i > 0 \quad (\text{for } i=1, \dots, n) \quad \text{のとき等号成立}$$

$$\hat{\phi}^i \in \Phi(\hat{x}), \quad \hat{y}_i \geq 0$$

ここで、 $x_i > 0$ の数を $N (N \leq n)$ とすると、

$$\sum y_{\mu} (\sum_{x_i > 0} \phi_i^{\mu} - \sum_{x_i > 0} x_i) + N \sum y_{\mu} = N$$

ところが、

$$\sum_{x_i > 0} x_i = 1$$

$$\sum_{x_i > 0} \phi_i^{\mu} \leq 1$$

故に、

$$N \sum y_{\mu} \geq N, \quad \sum y_{\mu} \geq 1$$

一方、

$$\sum y_i + n \sum y_{\mu} = n$$

故に、

$$\sum y_{\mu} \leq 1$$

したがって

$$\sum y_{\mu} = 1$$

$$y_i = 0 \quad \text{for } i = 1, \dots, n$$

これを (iii) 式に代入すると、

$$\sum y_{\mu} \phi_1^{\mu} = x_1$$

$$\vdots$$

$$\sum y_{\mu} \phi_n^{\mu} = x_n$$

$\phi^{\mu} \in \Phi(x)$, また、 $\Phi(x)$ は凸であるから、

$$x \in \Phi(x)$$

となる。これは、角谷の不動点である。

II d

replacement operation

ある一つの頂点をプリミティブ・セットから除き、残りの頂点はそのままに

して、新しい一つの頂点を入れ、新しいプリミティブ・セットを作る。すると、それはもとのプリミティブ・セットとフェイス“face”を共有し隣接している。(ある一つのフェイス(ドア)を共有する部分単体(へや)は、たかだか二個であるから) 実際のアゴリズムでは、この置き換え“replacement step”は、プリミティブ・セットの第 j 列が除かれるとき、 $I(j)$ と $I(j+1)$ を入れ換えるという操作である。

次のような $n \times n$ 行列を考える。

$$A = \begin{pmatrix} -1 & +1 & 0 & 0 \\ 0 & -1 & +1 & \vdots \\ \vdots & 0 & -1 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & +1 \\ 0 & \vdots & \vdots & \vdots & \vdots \\ +1 & 0 & 0 & -1 \end{pmatrix}$$

すると、プリミティブ・セットの第 j 列は、

$$m_i^j = m_i^{j-1} + A_i^{I(j)} \quad \text{for } i=1, \dots, n$$

と表わせる。そこで、プリミティブ・セット内の任意の連続する三つの列を考え、それぞれのベクトルに対応する順列を I_0, I_1, I_2 とする。そのとき、各要素は、

$$j_0 \text{ 列: } m_i^{j_0}$$

$$j_1 \text{ 列: } m_i^{j_0} + A_i^{I_1}$$

$$j_2 \text{ 列: } m_i^{j_0} + A_i^{I_1} + A_i^{I_2}$$

$$\text{for } i=1, \dots, n$$

となる。この場合、第 j_1 列が除かれるとすると、置き換え操作によって、

$$j_0 \text{ 列: } m_i^{j_0}$$

$$j_1 \text{ 列: } m_i^{j_0} + A_i^{I_2}$$

j_3 列: $m_i^{j_0} + d_i^{j_1} + d_i^{j_2}$

for $i=1, \dots, n$

と変わる。これは、プリミティブ・セットのあるベクトルを除くとき、その隣接する二つのベクトルの要素を足し合わせ、それから除かれるベクトルの要素を引くことと同値である。

ハンセンは、本質的に全く同じ操作を違う形で示しているが⁽¹⁹⁾、次元数が大きくなった場合、計算時間に関してはより効率的であろう。

II e

fixed-grid method

1) initial primitive set

出発点のプリミティブ・セットは、分子だけで表わすと、格子サイズ D , n 次元のとき、

$$M = \begin{bmatrix} D-n+1 & D-n+2 & \cdots & D-n+2 \\ 1 & 0 & 1 & \cdots & 1 \\ \vdots & 1 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}$$

となり、これは単体 S の頂点 $(1, 0, \dots, 0)'$ 付近のプリミティブ・セットである。この場合、 M の第2列から第 n 列はスラック・ベクトルであるから ((i)' より)、第1列がラベル1でない限りある一つのベクトルだけが重複し、完全にラベル付けされたプリミティブ・セットを見つけるために除かれる。(図 13)

2) algorithm

スカラー・ラベル (i) とベクトル・ラベルのそれぞれについて、アルゴリズム

(19) 文献 [9], 第6章

ムをフロー・チャート⁽²⁰⁾で示す。(図 14, 15)

II f restart method⁽²¹⁾

再スタート法は、固定格子法で扱った単体を1つ次元の高い単体に埋め込んでアルゴリズムを進める。図 16 のように、不動点がある単体 H_1 を H_0 と H_2 ではさむことからサンドウィッチ法とも呼ばれる。単体 H_0, H_1, H_2 上のベクトルは、 $n+1$ 次元で表わされ、その分子は、

$$x^j \in H_0 \text{ について} \quad m_{n+1}^j = 0, \quad \sum_{i=1}^n m_i^j = D+1$$

$$x^j \in H_1 \text{ について} \quad m_{n+1}^j = 1, \quad \sum_{i=1}^n m_i^j = D$$

$$x^j \in H_2 \text{ について} \quad m_{n+1}^j = 2, \quad \sum_{i=1}^n m_i^j = D-1$$

と書ける。

1) labelling

(20) フローチャートの語句の説明をしておく。

JIN: プリミティブ・セットに置き換えで入れられたベクトルの列番号

JOUT: 置き換えによって、プリミティブ・セットからはき出されるベクトルの列番号

AIN: ビボット・ステップによって実行可能基底に入っている列ラベルベクトル

AOUT: 実行可能基底から、ビボット・ステップによってはき出される列ラベルベクトル

A^k : プリミティブ・セット $\{X^1, \dots, X^k, \dots, X^n\}$ 内のベクトル X^k に対応するラベルベクトル

E : 単位行列

e^i : 第 i 単位ベクトル

(21) 文献 [5]

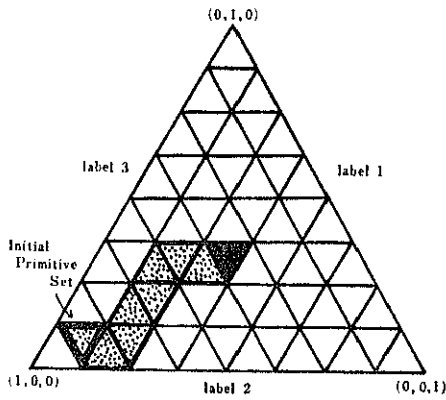


図 13 固定格子法

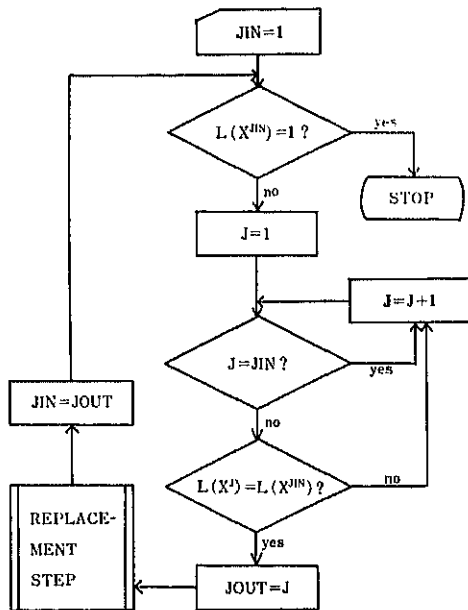


図 14 アルゴリズム (スカラー・ラベル)

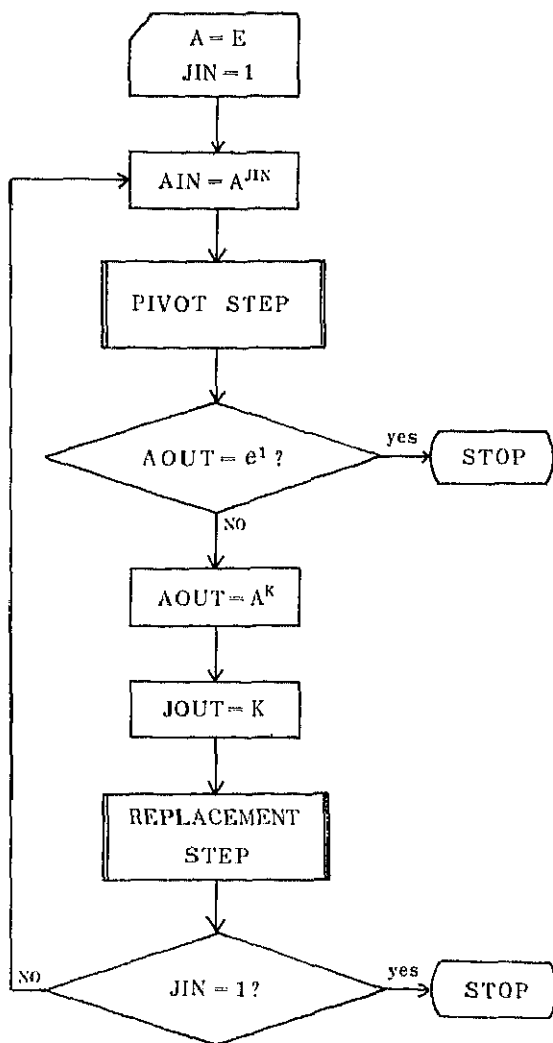


図 15 アルゴリズム (ベクトル・ラベル)

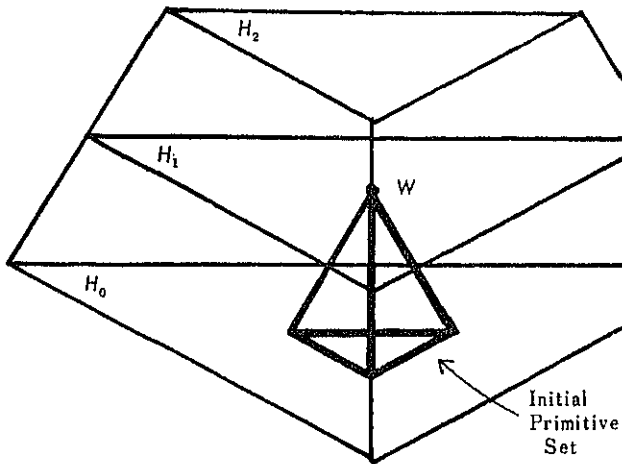


図 16 再スタート法

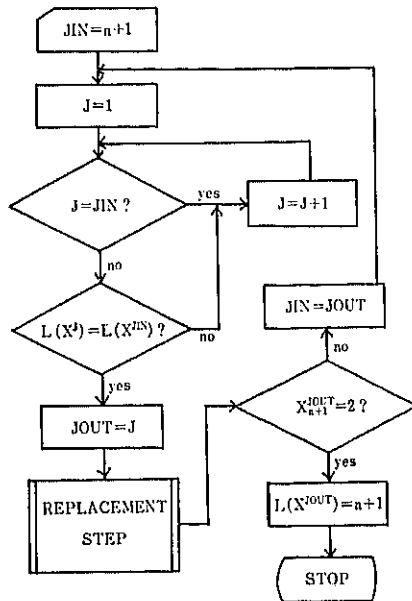


図 17 アルゴリズム

- ① ある特定のベクトル $W \in H_1$ を定める。 $x \in H_0$ については,

$$L(x) = K \quad \text{if } (x_K - W_K) \geq (x_i - W_i) \text{ for } i=1, \dots, n$$

$$\text{and } K \leq i \quad \text{if } (x_K - W_K) = (x_i - W_i)$$

- ② $x \in H_1$ については, (ii) 式にしたがう。for $i=1, \dots, n$

- ③ $x \in H_2$ については, $L(x) = n+1$

2) initial primitive set

- ① 格子サイズパラメータ D' を定める。

- ② $W \in H_1$ について, $m^W = (D'/n, \dots, D'/n, 1)$ とする。 $(D'/n$ が整数でない場合は, 近似の整数とする。)

- ③ $m_i^j = m_i^W + E_i^j - E_i^{n+1}$ (E は $n+1$ 次元単位行列)

この出発点のプリミティブ・セットでは, H_1 に属するベクトルが一つ (W) と, H_0 に属するベクトルが n 個ある。このとき, H_0 に属するベクトルは, $1, \dots, n$ のラベルを一つずつ持ち, W のラベルは, $1, \dots, n$ のうちの一つであるから, 不完全にラベル付けされた単体である。除かれるべきベクトルは, ラベルが一つだけしか重なっていないので一意に決まる。

3) algorithm

アルゴリズムはフロー・チャート⁽²⁰⁾で示す。(図 17)

もし, 近似の精度が十分でないなら, 格子をより細かくし, 最終のへやの内部に新たに W をとり, 再スタートすればよい。

II g homotopy method⁽²²⁾

イーヴスのアルゴリズムの特徴は, $S_0, S_1, \dots, (S_K$ は $D=2^K$ の正規単体分割の族) を R^n の正象限 R^{n+} に埋め込み, その錐体そのものを正規分割する

(22) 文献 [1], [2]

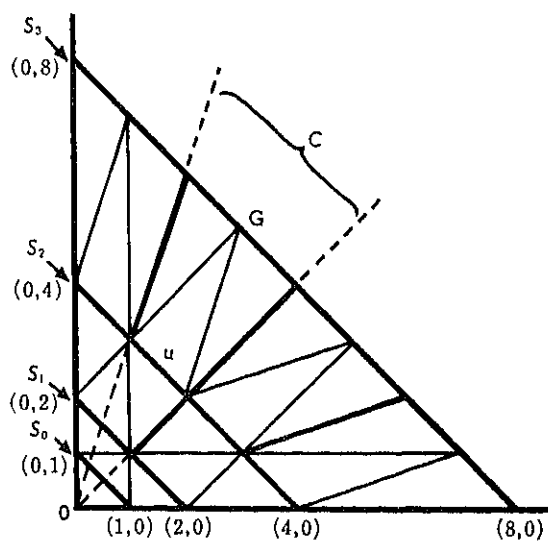


図 18 ホモトピー法

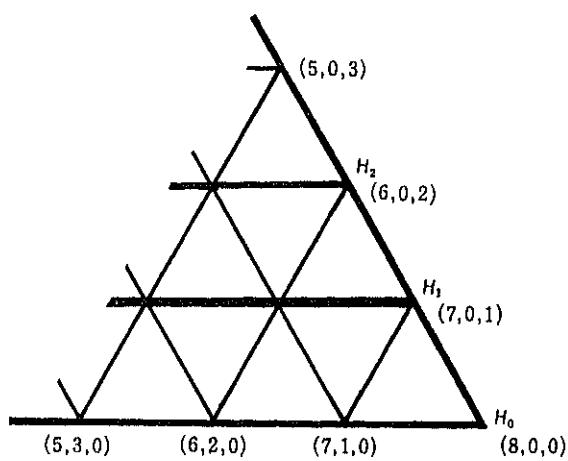


図 19 再スタート法

ことにある。図 18, 19 より明らかなように、再スタート法とちがひ、 S_K の格子サイズは一樣でない。近似値の精度は、 S_K 内の n 単体 u から S_{K+1} 内の u' へ徐々に移動することによって高められる ($K=0, 1, 2, \dots$)。通常、 u' は、原点と u の各頂点を結ぶ直線によって形成される錐体 C と S_{K+1} との共通部分 G の中にある。すなわち、 u を 2^{n-1} 個に正規分割した (格子サイズを $1/2$ にした) 部分単体の 1 つが u' になるのである。

置き換えを示す前に、記号を定義する。

$$R^{n++} = \{x | x \in R^{n+}, \sum x_i \leq 2, x_i \text{ は非負整数}\}$$

u : S_K 内の n 部分単体

$$\text{ただし, } u^{j+1} = u^j + P(\beta_j) \quad \text{for } j=1, \dots, n-1$$

$$P_i(j) = \begin{cases} -1 & i=j \\ 1 & i=j+1 \\ 0 & \text{その他} \end{cases}$$

v : R^{n++} 内の $n+1$ 部分単体

$$\text{ただし, } v^{j+1} = v^j + q(\gamma_j) \quad \text{for } j=1, \dots, n$$

$$q_i(j) = \begin{cases} -1 & i=j \\ 1 & i=j+1, i \leq n \\ 0 & \text{その他} \end{cases}$$

$$Tu(v) = u \cdot v$$

u と v はそれぞれ、 $u = [u^1, \beta]$, $v = [v^1, \gamma]$ と表わせる。

○ replacement step

$$(1) \quad v^* \in R^{n++}$$

$$v^* = v^{n+1} + v^2 - v^1 \quad \text{JOUT} = 1$$

$$v^* = v^{\text{JOUT}-1} + v^{\text{JOUT}+1} - v^{\text{JOUT}} \quad 2 \leq \text{JOUT} \leq n$$

$$v^* = v^n + v^1 - v^{n+1} \quad \text{JOUT} = n+1$$

これを $[v^i, \gamma]$ で表わせば,

$$v^* = [v^2, (\gamma_2, \dots, \gamma_n, \gamma_1)] \quad \text{JOUT} = 1$$

$$v^* = [v^1, (\gamma_1, \dots, \gamma_{\text{JOUT}}, \gamma_{\text{JOUT}-1}, \dots, \gamma_n)] \quad 2 \leq \text{JOUT} \leq n$$

$$v^* = [v^1 - g(\gamma_n), (\gamma_n, \gamma_1, \dots, \gamma_2)] \quad \text{JOUT} = n+1$$

イーヴスのアルゴリズムは, u の上に v の埋め込みをしているのであるから, この場合, v の replace だけを行なえばよい。

ところが, (1) の操作を行なった結果, v の定義にもとる v^* が得られる場合がある。 v に関してみれば, 以下の3つのケースが起るが, それぞれについて若干解説し, 例を示す。

$$(2) \quad \sum v_i^* > 2 \quad [\text{JOUT} = n+1, \gamma_n = n]$$

$\gamma_n = n$ であるから, v^{n+1} のレベル ($\sum_j v_j^{n+1}$) は1, 他 (v^1, \dots, v^n) は2である。 v^{n+1} を (1) に従って計算された新しい v^* と入れかえることは, レベル2のみの $v^i (i=1, \dots, n)$ から成るフェイスを共有するもう1つのへやへ移ることである。この時, 入ってきた頂点のベクトルのレベルは, 1でも2でもなく, 2より大きくなる。しかし, 定義より v のレベルは1か2でなければならぬため, u のレベルを一段上げる ($\sum_j u_j^i (i=1, \dots, n)$ を2倍にする) ことにより, 置き換えを続ける。つまり, (2) は, より細かい格子に移ったことを意味している。

Example: $\text{JOUT} = n+1, \gamma_n = n$

$$\begin{aligned} u &= \begin{pmatrix} 4 & 3 & 3 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{pmatrix} & v &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & Tu(v) &= \begin{pmatrix} 7 & 6 & 6 & 3 \\ 5 & 6 & 5 & 3 \\ 4 & 4 & 5 & 2 \end{pmatrix} \\ & & \gamma &= (1 \ 2 \ 3) \\ u' &= \begin{pmatrix} 7 & 6 & 6 \\ 5 & 6 & 5 \\ 4 & 4 & 5 \end{pmatrix} & v' &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} & Tu'(v') &= \begin{pmatrix} 13 & 7 & 6 & 6 \\ 10 & 5 & 6 & 5 \\ 9 & 4 & 4 & 5 \end{pmatrix} \\ & & \gamma' &= (3 \ 1 \ 2) \end{aligned}$$

$\gamma_1=n$ の u は一意的に決り、 $v'=(1,0,\cdots,0,1)'$ であるため、 v^* は $(v'+v^*)$ で求められる。

(3) $v^*=0$ [JOUT=1, $\gamma_1=n$]

(3) は (2) の逆で、レベルが1以下のベクトルが頂点に入ってきてしまう場合である。しかし、やはり v のレベルはより小さく出来ないため ($v=0$ と出来ないため)、 u のレベルを一段下げる ($\sum_j u_j^i (i=1,\cdots,n)$ を 2^{-1} 倍する)。つまり、(3) は格子サイズを上げ、ある u をベースにしてアルゴリズムを進めたが、近似値となるプリミティブ・セットが見出されないため、他の u に移り、アルゴリズムを続行するためのステップの1つである。

Example: JOUT=1, $\gamma_1=n$

$$u=\begin{pmatrix} 4 & 3 & 3 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{pmatrix} \quad v=\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad Tu(v)=\begin{pmatrix} 7 & 4 & 3 & 3 \\ 4 & 2 & 3 & 2 \\ 5 & 2 & 2 & 3 \end{pmatrix}$$

$$\gamma=(3 \quad 1 \quad 2)$$

$$u'=\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \quad v'=\begin{pmatrix} 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad Tu'(v')=\begin{pmatrix} 4 & 3 & 3 & 2 \\ 2 & 3 & 2 & 1 \\ 2 & 2 & 3 & 1 \end{pmatrix}$$

$$\gamma'=(1 \quad 0 \quad 3)$$

$\gamma_1=n$ の時、 v は一意的に決るが、新しい v' は、 $\gamma_n=n$ だけが条件で、 u の値に依存する。

(4) $v^*<0$ for some $j=1,\cdots,n$

v^* は、レベルは1あるいは2であるが、 v の定義にかなう $n(n+3)/2$ 個のベクトルでない場合である。置き換えを続けるには、定義を満たすある v' を埋め込んだ時、新しいへやが表現できるような、新しい u' に移らなければならない。

Example: JOUT=2

$$u = \begin{pmatrix} 4 & 3 & 3 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{pmatrix} \quad v = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 \end{pmatrix} \quad Tu(v) = \begin{pmatrix} 7 & 6 & 6 & 3 \\ 4 & 5 & 4 & 2 \\ 5 & 5 & 6 & 3 \end{pmatrix}$$

$$\gamma = (1 \ 2 \ 3)$$

$$u' = \begin{pmatrix} 4 & 3 & 3 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{pmatrix} \quad v' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 \end{pmatrix} \quad Tu'(v') = \begin{pmatrix} 7 & 7 & 6 & 3 \\ 4 & 3 & 4 & 2 \\ 5 & 6 & 6 & 3 \end{pmatrix}$$

$$\gamma' = (1 \ 2 \ 3)$$

この場合は, $v=v'$ であるが, v が変化する場合もある。

参 考 文 献

- [1] B. C. Eaves, Homotopies for Computation of Fixed Points, *Mathematical Programming* 3, 1972, pp. 1-22.
- [2] R. Jansen and A. S. Louter, An Efficient Way of Programming Eaves' Fixed Point Algorithm, in: *Numerical Solution of Highly Nonlinear Problems*, ed. W. Foaster, North-Holland, Amsterdam, 1980, pp. 115-167.
- [3] H. W. Kuhn, Simplicial Approximation of Fixed Points, *Proceedings of the National Academy of Sciences, U. S. A.* 61, 1968, pp. 1238-1242.
- [4] H. W. Kuhn, How to Compute Economic Equilibria by Pivotal Methods, in: *Computing Equilibria: How and Why*, eds. Jerzy Łoś and Maria W. Łoś, North-Holland, Amsterdam, 1976, pp. 21-38.
- [5] H. W. Kuhn and J. G. Mackinnon, Sandwich Method for Finding Fixed Points, *Journal of Optimization Theory and Applications* 17, 1975, pp. 189-204.
- [6] L. W. McKenzie, Why Compute Economic Equilibria?, in: *Computing Equilibria: How and Why*, eds. Jerzy Łoś and Maria W. Łoś, North-Holland, Amsterdam, 1976, pp. 3-19.
- [7] M. Morishima, *Walras' Economics: a pure theory of capital and money*, Cambridge Univ. Press, Cambridge, 1977.

- [8] H. E. Scarf, Some Examples of Global Instability of the Competitive Equilibrium, *International Economic Review* 1, 1960, pp. 157-172.
- [9] H. E. Scarf (with the collaboration of T. Hansen), *The Computation of Economic Equilibria*, Yale Univ. Press, New Haven, 1973.
- [10] H. E. Scarf, Introduction, in: *Fixed Points: Algorithms and Applications*, ed. S. Karamardian, Academic Press, New York, 1977, pp. 1-10.
- [11] A. Takayama, *Mathematical Economics*, Dryden Press, 1974.