非線形最適化問題に対する分枝限定法の適用

正道寺　勉

1.　はじめに

本号の特集テーマである「分枝限定法」とその周辺に関する記事は，本誌を振り返っただけでもすでに「組合せ最適化」（1974年9月号，1986年1月号），「数理計画法」（1977年6月号）という特集が企画されており，さらに特別講座［7］やＯＲ学会整数計画法研究部会の総合報告［13］という形でも取り上げられてきた．

巡回セールスマン問題（行商人問題ともいう）に代表される整数計画問題を解くための，代表的手法である分枝限定法は，広く一般に知られている［8］．しかしながら，日本では分枝限定法の非線形最適化問題への適用に関する話題は，これまであまり取り上げられていなかったので，本稿ではこの点に焦点を合わせて述べることにする．

2.　非線形最適化問題

本稿で扱う最適化問題は，次に示す［問題P］によって表現されるが，一般の最適化問題では，Dが有界閉集合であるという制約はない．しかしながら，変数の定義域を定めることにより，Dが有界閉集合でない問題も扱うことができる．

［問題P］　有界閉集合$D\subset \boldsymbol{R}^n$と関数$f: \boldsymbol{R}^n\to \boldsymbol{R}$が与えられたとき，

$$f(\boldsymbol{x}^*)=\max f(\boldsymbol{x})\quad (\text{or} \ \min f(\boldsymbol{x}))$$

を満たすベクトル$\boldsymbol{x}^*\in D$を見つけよ．

ただし，集合Dは$g_i(\boldsymbol{x})\leqq 0\ (i=1,2,\cdots,m)$なる不等式系によって定義される集合で，$g_i: \boldsymbol{R}^n\to \boldsymbol{R}$ $(i=1,2,\cdots m)$である．

［問題P］における集合Dは，一般に複数個の制約条件式によって定まる制約集合（基礎になる空間Xの部分集合）で，許容領域または実行可能領域と呼ばれる．ここでは，基礎になる空間Xをn次元ユークリッド空間$\boldsymbol{R}^n$としているが，現実問題を考える上での距離空間としては自然であろう．

さて，この最適化問題において，目的関数fと制約関数gの少なくとも一方が非線形関数である場合を非線形最適化問題，または非線形計画問題と呼ぶ．特に，目的関数fが凸（凹）関数で，制約関数gが凸関数のとき，この問題を凸（凹）計画問題と呼び，多くのアルゴリズム（目的関数の単峰性が保証されているため，この性質をうまく利用できる）が知られている．本稿で議論する非線形最適化問題は，関数f，gが凸関数でない一般の非線形最適化問題を想定しており，多くの局所解が存在するのが普通である．**4.**で述べる分枝限定法にもとづいた2つの手法は，いずれもこのタイプの問題を解くために考え出されており，凸計画問題に対しても適用できる．

ここで，分枝限定法の考え方を必要とした問題の背景を考えてみよう．解こうとしている問題が難しい場合（たとえば，問題のサイズや関数の非凸性に起因する場合など）には，何とかしてより易しい問題に変換して解こう，という試みがなされる．

元の問題をより易しい問題に変換して解く方法としては，

(a)　元の問題の制約条件や目的関数にともなう制約などを緩めて得られる緩和問題を解くことによって，元の問題の解を得る方法，

(b)　元の問題を分割して得られる部分問題をすべて解くことによって，元の問題の解を得る方法，などがよく知られている．(a)の場合には，緩和問題を解いて解が存

しょうどうじ　つとむ
日本工業大学　工学部　システム工学科
〒345 埼玉県南埼玉郡宮代町学園台4－1

在しなければ，元の問題にも解は存在せず，緩和問題の解が元の問題の許容解であれば解が得られたとする考え方である．

一方，(b)の場合には，元の問題に解が存在しなければ，それを分割した問題にも解は存在せず，もし解が存在する場合には，分割された問題のいずれかの解と一致することから，解が得られるという考え方である．分枝限定法の基本となる考え方の1つに(b)の考え方（**3.**で述べる分枝操作に相当する）があり，最適化問題を解く場合には，緩和法と分枝限定法を組み合せてアルゴリズムを構成することも多い．

3. 分枝限定法

列挙法(シラミ潰し法)の一種である分枝限定法は，誕生の当初より組合せ最適化問題を効率よく解くために考え出された計算原理で，基本的な考え方はきわめて簡単なものである．この簡単さゆえ頑健性に優れ，広い範囲の問題に対して分枝限定法が適用されており，多くの成功を収めている．組合せ最適化問題は，離散的最適化問題とも呼ばれ，すべての組合せを調べ上げることによって解を得ることができる問題を指す．しかしながら，問題の大きさや複雑さなどによっては，調べるべき組合せの数が天文学的な数になってしまい，高速のスーパー・コンピュータを使ったとしても，すべてを調べつくすのに何年もかかる場合がある．したがって，このような場合には，いくら組合せの数が有限でそれらをすべて調べればよいといっても，現実的には解けないのと同じである．

とはいっても，実際家にしてみれば現実に解かねばならない問題に直面したとき，厳密解は得られなくても，何とかして近似解でもよいから見つけだしたい，と思うのが人の常であろう．このような大規模な問題に対処するため，分枝限定法を利用した近似解法やヒューリスティック解法などが提案されている．分枝限定法に関する一般的な記述は，Lawler-Wood[9]，Mitten[10]，茨木[8]などに詳しく，分枝限定法の基本となる考え方は，次のとおりである．

与えられた［問題P］を直接解くのが難しい場合には，より易しい部分問題（$P^1, P^2, \cdots, P^k$）に分割し，その部分問題をすべて解くことによって［問題P］を間接的に解こう，という考え方である．部分問題 P^i $(i=1,2,\cdots,k)$ は，

［部分問題 P^i］

max $f(\boldsymbol{x})$,

subject to $\boldsymbol{x} \in D \cap X_i$, $(i=1,2,\cdots,k)$.

ただし，$X_i \subset X$ は $(\bigcup_{i=1}^{k} X_i) \supset D$ となるように選ばれる．

で表現され，記号の意味は，［問題P］の場合と同じである．したがって，各［部分問題 P^i］の最適解 $\boldsymbol{x}^{(i)}$ が求まれば，

$$f(\boldsymbol{x}^*) = \max_i \{ f(\boldsymbol{x}^{(i)}) \mid i=1,2,\cdots,k \}$$

を満足する［問題 P］の最適解 $\boldsymbol{x}^*$ が得られる．部分問題 P_i の中で，分割してもなお解くのが困難な問題がある場合には，上で述べた考え方をさらに適用すればよい．ただし，分割だけを繰り返して解こうとすると列挙法と同じになり，膨大な数の部分問題が生成されることになる．そこで，解くべき部分問題の数をできるだけ少なくする工夫をして問題を解こう，というのが分枝限定法の基本的な考え方で，次に示す2つの基本となる操作が分枝限定法という名前の由来にもなっている．

①**分枝操作**：直接解くことがむずかしい問題を，より易しいいくつかの部分問題に分解する操作のことで，分解操作ともいう．

②**限定操作**：分枝操作によって生成された部分問題のうち，どの部分問題を調べ，どの部分問題を捨てるべきか，を判定する操作のこと．

ある問題を部分問題に分解する方法や，分解された部分問題を限定する方法は，数多く存在するであろうことは容易に推測できる．それゆえ，分枝限定法の考えにもとづいた効率のよいアルゴリズムを構築しようとするなら，解こうとしている問題の構造をよく調べ，その構造に適した分枝操作と限定操作を行なう必要がある．

4. 分枝限定法にもとづいたアルゴリズム

本節では，非線形最適化問題を解くために提案された分枝限定法にもとづく2つのアルゴリズムを紹介する．非線形最適化問題に対して，分枝限定法が有効であることを初めて示唆したのは，恐らく Lawler-Wood[9] であろう．

彼らの論文には，次に紹介するアルゴリズムのもとになる考え方がすでに示されている．

4.1 Falk-Soland のアルゴリズム

Falk Soland[3] は，次のような大域的最小化問題を解くためのアルゴリズムを提案した．

［問題 P'］

$\min\ \phi(\boldsymbol{x})=\sum_{i=1}^{m}\phi_i(x_i)$,

subject to $\boldsymbol{x}\in C\cap D\subseteq \boldsymbol{R}^n$.

ここで，Dは不等式系の制約式によって定まるコンパクト凸集合で，Cは$C=\{\boldsymbol{x}\in\boldsymbol{R}^n \mid \alpha_i\leqq x_i\leqq\beta_i,\ i=1,2,\cdots n\}$ と定義される $\boldsymbol{R}^n$ の矩形部分集合である．

彼らの提案したアルゴリズムは，前述の緩和問題に変換する方法と分枝限定法を利用しており，アルゴリズムの中で凸包絡線 (convex envelope：エピグラフと凸包の概念にもとづいている)が重要な役割を演じている．すなわち，分離可能な目的関数 (separable objective function：凸である必要はない) を凸包絡線による凸関数に置き換えることにより，目的関数を緩和している．Falk[2]による凸包絡線の説明は，以下のとおりである．ϕをある集合$C\subseteq\boldsymbol{R}^n$上で定義された任意の関数であるとしよう．集合：

$[\phi,\ C]=\{(\xi,\ \boldsymbol{x})\mid\xi\geqq\phi(\boldsymbol{x}),\ \boldsymbol{x}\in C\}$

はϕのエピグラフと呼ばれ，ϕのグラフ上とその上方にある $\boldsymbol{R}^{n+1}$ 上のすべての点から成り立っている．それゆえ，$[\phi,\ C]$ の凸包 $[\phi,\ C]^C$ は $[\phi,\ C]$ を含む $\boldsymbol{R}^{n+1}$ 上の最小の凸集合である．この集合の下側部分がΨのグラフ，すなわち，

$D^{\Psi}=\{\boldsymbol{x}\mid(\xi,\ \boldsymbol{x})\in[\phi,\ C]^C$ for some $\xi\}$

なる領域 D^{Ψ} で定義されるϕの凸包絡線：

$\Psi(\boldsymbol{x})=\inf\{\xi\mid(\xi,\ \boldsymbol{x})\in[\phi,\ C]^C\}$

を形成する．**図 1**は，一変数関数に対する凸包絡線を表わしたものである．もし関数ϕが凹関数なら，その凸包絡線Ψは与えられた関数のグラフの端点を通る 1 次関数である．

また，問題が解をもつためには，$C\cap D$ が空集合でないという条件の下で，D がコンパクト凸集合かつ ϕ, g がC上で下半連続であり，ϕがC上で分離可能であることを仮定している．

アルゴリズムは，一連の実行可能解の点列 $\{\boldsymbol{x}^k\}$ を生成し，$\boldsymbol{x}^k$ は $C\cap D$ 上の区分的凸関数の最小化を含む問題 P^kの解となっている．問題P^kは，分枝限定法により問題 P^{k-1} から生成され，分枝操作はCをより小さな矩形に分割することに対応している．一方，限定操作は分割された矩形領域とDとの共通集合におけるϕの下界値を決定した上で，さらに調べるべき部分問題を選定することに対応している．

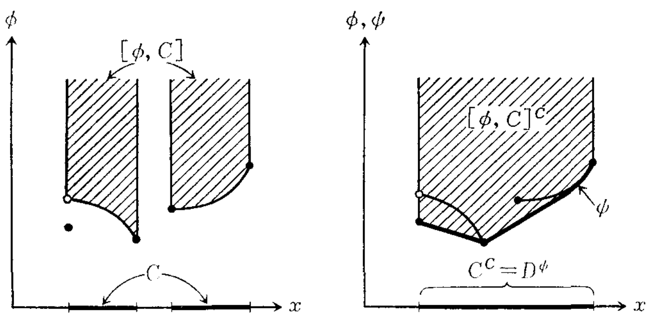

図 1　一変数関数に対する凸包絡線

さて，アルゴリズムの具体的な説明に入ることにしよう．最初に考える問題 P^1は，元の問題 P のことで次に示すような形をしている．

［問題 P^1］

$\min\ \Psi^1(\boldsymbol{x})$,

subject to $\boldsymbol{x}\in C\cap D$.

ここで，$\Psi^1(\boldsymbol{x})$はC上で定まるϕの凸包絡線で，$D=\{\boldsymbol{x}\mid g_i(\boldsymbol{x})\leqq 0,\ i=1,2,\cdots,m\}$ である．

もし，g_i が凸関数であれば，問題 P^1は凸計画問題となり，簡単に解を得ることができる．仮に，$\boldsymbol{x}^1$ が問題P^1の解であるとしよう．このとき，もし $\phi(\boldsymbol{x}^1)=\Psi^1(\boldsymbol{x}^1)$なら，$\Psi^1(\boldsymbol{x}^1)$ は $C\cap D$ 上での $\phi(\boldsymbol{x}^1)$ の下界を与えるので，$\boldsymbol{x}^1$ は元の問題 P の解である．しかしながら，一般には$\Psi^1(\boldsymbol{x}^1)<\phi(\boldsymbol{x}^1)$なので，集合$C$を分枝限定法を用いて2つ（もしくはそれ以上）の矩形部分集合に分割し，再び分割された矩形部分集合上における $\phi(\boldsymbol{x})$ の凸包絡線が最小化される．アルゴリズムのあるステップにおいて，$\phi(\boldsymbol{x}^*)=\min_{kj}\{\phi(\boldsymbol{x}^{kj})\}$ であるとしよう．このとき$\boldsymbol{x}^*$は，$\boldsymbol{x}^{kj}$ に対応しているものとする．$\phi(\boldsymbol{x}^*)$はこの時点において，問題Pの最適値（最小値）の上界値であるから，問題P^{kj}の最適値が $\phi(\boldsymbol{x}^*)$より大きいか等しければ問題P^{kj}をさらに分割する必要はない．すなわち，その中で最も小さい値に対応するCの矩形部分集合をさらに分割するという手続きを繰り返し，

$\phi(\boldsymbol{x}^{kj})=\Psi^{kj}(\boldsymbol{x}^{kj})$,　$(j=1,2,\cdots,p_k)$

$\boldsymbol{x}^{kj}\in C\cap D$

となる$\boldsymbol{x}^{kj}$が得られれば，アルゴリズムは終了する．ここで，kはアルゴリズムの第kステップを表わし，p_kは第kステップにおいて生成されたCの矩形部分集合の個数を表わす．したがって，$\boldsymbol{x}^{kj}\in C^{kj}(j=1,2,\cdots,p_k)$ は $C^{kj}\cap D$ において，C^{kj} 上で定義された ϕ の凸包絡線Ψ^{kj}を最小にする値で，問題 P^{kj} の 1 つの解である．

このアルゴリズムは，Soland [21] によって制約集合Dが非凸集合であっても使えるように拡張された．Horst[4, 5] は，Soland のアルゴリズムを基にして，新しい分枝限定法タイプのアルゴリズムを提案している．主な相違点は，制約領域を矩形分割する替わりにコンパクト分割を行ない，凸包絡線の使用を要求していないことである．さらに，目的関数の分離可能性をも要求していない．Falk-Soland と Horst のアルゴリズムの収束性が Benson [1] によって議論されている．

最近，Horst-Tuy [6] は Horst のアルゴリズムを一般化したアルゴリズムの概念を提案している．

4.2 Lipschitz 定数を用いたアルゴリズム(注1)

関数のクラスが Lipschitz 連続である場合の非線形最適化問題に対しても，やはり分枝限定法にもとづいたアルゴリズム [20, 14, 15, 17, 12] が提案されており，Lipschitz 定数を用いて部分問題を限定しているのが特徴である．アルゴリズムが対象としている問題は，**2.** の[問題P] における関数f, gが$f: \boldsymbol{R}^n \to \boldsymbol{R}$, $g_i: \boldsymbol{R}^n \to \boldsymbol{R}$ $(i=1,2,\cdots,m)$で，共に Lipschitz 連続関数を仮定しており，各変数の定義域があらかじめ，$x_i \in [a_i,\ b_i]$ と定められていることを仮定している．また，等号制約条件式が含まれている場合には，不等号制約条件式を用いて等価な問題に変換できる．（常に変換可能というわけではないが，実際問題を考える上では問題なく変換できることが多い）

次に，アルゴリズムを述べるために必要な定義と定理を述べておこう．

定義1　空でない集合 $X \subset \boldsymbol{R}^n$ の任意の部分集合 S に対して，関数 f と部分集合 S にのみ依存する，ある定数 $K>0$ が存在し，任意の2点 $\boldsymbol{x}_1$, $\boldsymbol{x}_2 \in S$ が与えられたとき，

$$|f(\boldsymbol{x}_2)-f(\boldsymbol{x}_1)| \leqq K\|\boldsymbol{x}_2-\boldsymbol{x}_1\|,$$

が常に成り立てば，関数 $f: \boldsymbol{R}^n \to \boldsymbol{R}$ は Lipschitz 連続であるという．ここで，$\|\cdot\|$ はユークリッド・ノルムである．□

定義1の幾何学的な解釈は，$\boldsymbol{R}^1$上に有界部分集合$S \subset X$と任意の2点 $x_1, x_2 \in S$ が与えられたとき，S上において関数fのグラフが有限の傾きKを持つ2つの1次関数 $y=f(x_1) \pm K|x_2-x_1|$ で挟まれることを意味する．すなわち，定義1の式を満たす適当な定数 $K>0$ が見つけられれば，関数fの集合S上での大域的最大値の上界値と大域的最小値の下界値が得られることになる．この事実より，集合Sを分割することで$|x_2-x_1|$を小さくすることができ，関数fの上界値，下界値を改善することができる．以上のことは，次元を拡張しても成立する．

Lipschitz 定数を用いたアルゴリズムは，上で述べたことを実現するため，分枝限定法の考え方をもとに構成されている．アルゴリズムの中で重要な役割を演じているのが Lipschitz 定数（計算方法は参考文献 [15] を参照）であり，部分問題を限定するための根拠となる定理1も Lipschitz 定数に依存している．

定義2　区間の直積集合を $C=\prod_{i=1}^{n}[a_i,\ b_i]$，制約条件式によって定まる制約集合を

$$D=\{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{R}^n,\ g_j(\boldsymbol{x}) \leqq 0,\ (j=1,2,\cdots,m)\},$$

そして実行可能集合を $F=\{\boldsymbol{x} \mid \boldsymbol{x} \in C \cap D,\ \boldsymbol{x} \in \boldsymbol{R}^n\}$ と定義する．□

$F=C$ でない場合には，実行可能集合Fを見つけて，Fのみを分割するのは一般に困難なので，本アルゴリズムでは区間の直積集合Cに着目し，Cを分割する過程においてFを見つけだす，という方法を取っている．正確には，各反復時点において分割された小領域で，Fを覆うことのできる最小の被覆集合 $A \supset F$ を見つけており [18]，集合Aは実行可能集合Fを緩和している．(**4.3**の図2と図3を比較されたい)

定義3　Cの分割は二分法（各変数の区間を半分に分割する方法）で行ない，分割された領域を分割小領域と呼び，$C^{kj}(j=1,2,\cdots,m_k)$で記す．その中心点を分割小領域の代表点と呼び，$P^{kj}(j=1,2,\cdots,m_k)$ で記す．ここで，kはアルゴリズムの第k反復であることを示し，m_kは第k反復において生成された分割小領域の個数を示す．m_k個の分割小領域の間には，$C=\cup_k(\cup_j C^{kj})$，$C^{kj} \cap C^{kj'}=\phi(j \neq j')$という関係が常に成り立っている．□

定義4　矩形集合の径（有界な点集合の径とは，その集合に属する2点間の距離の上限をいう）は，対角線の長さであるから，半径はその1/2である．したがって，分割小領域 C^{kj} の代表点 P^{kj} を中心とする最大半径 M_kを，次のように定義する．

$$M^k=\{\sum_{i=1}^{n}(\tau_i/2^{k+1})^2\}^{1/2}.$$

ただし，$\tau_i=b_i-a_i(i=1,2,\cdots,n)$である．□

定理1　目的関数fの Lipschitz 定数を $\mathrm{Lip}(f)$，分割小領域C^{kj}の代表点P^{kj}での関数値を$f^{kj}=f(P^{kj})$，それまでに得られている目的関数の最大値をf^{k*}としたとき，アルゴリズムの第k反復におけるf^{kj}に対して，

$$f^{k*}>f^{kj}+\mathrm{Lip}(f)\,M_k,\ (j=1,2,\cdots,m_k)$$

(注1)　本項の研究の一部は，昭和62年度学内特別研究費の援助による．

となる f^{kj} が存在すれば，点 P^{kj} を含む C^{kj} 内には，f^{k*} よりも大きな値は存在しない．□

（証明） 任意の点 $Q\in C^{kj}$ をとれば，定義1より

$$f(Q)\leqq f^{kj}+\mathrm{Lip}(f)\,M_k,\ \ (j=1,2,\cdots,m_k)$$

が成立する．もし，定理1の式が成立するとすれば，$f^{kj}+\mathrm{Lip}(f)\,M_k<f^{k*}$，$(j=1,2,\cdots,m_k)$ であり，上の2つの式から $f(Q)<f^{k*}$ となる．■

以下に，アルゴリズムの概要を示す．

step 1 Lipschitz 定数，変数の定義域，収束判定のための定数 $\varepsilon>0$，反復回数：$k\leftarrow 1$ を入力する．C の最大半径 M_0，C の代表点 P^{11} における関数値 f^{0*} を計算する．$M_0=\{\sum_{i=1}^{n}(\tau_i/2)^2\}^{1/2}(\tau_i=b_i-a_i,\ i=1,2,\cdots,n)$，$f^{0*}=f(P^{11})$．

step 2 n 次元直積空間Dを二分法により分割する．分割された個数を m_k とする．

step 3 $C^{kj}\cap D\ (j=1,2,\cdots,m_k)$ が空集合であるかどうかの判定を行ない，空集合の場合には対応する分割小領域を捨てる．さもなければ，保存しておき step 4 へゆく．保存すべき分割小領域が1つもなければ，アルゴリズムを終了する．

step 4 分割小領域C^{kj}の代表点P^{kj}での目的関数の値 $f^{kj}=f(P^{kj})$を計算する．

step 5 $f^{k*}=\max_{kj}f^{kj}(j=1,2,\cdots,m_k)$を求め，$f^{k*}=\max\{f^{k*},\ f^{k-1,*}\}$ を大域的最大値の下界値（いわゆる暫定値）とする．

step 6 分割小領域の最大半径 $M_k=M_{k-1}/2$ を計算する．

step 7 $\{\mathrm{Lip}(f)\cdot M_k/|f^{k*}|\}\leqq\varepsilon$

を満足すれば，アルゴリズムを終了する．さもなければ，step 8 へゆく．

step 8 上界値テスト：定理1より，

$$f^{k*}\leqq f(P^{kj})+\mathrm{Lip}(f)\cdot M_k$$

を満たす探索点 P^{kj} を含む分割小領域をすべて見つけ，$k\leftarrow k+1$ と置き step 2 へ戻る．□

step 7の収束判定条件は，このアルゴリズムが ε-近似アルゴリズムであることを示している．その理由は，目的関数 f の大域的最大値を f^*（以下の議論においては $f^*\neq 0$ を仮定している）としたとき，$f^{k*}\leqq f^*\leqq f^{k*}+\mathrm{Lip}(f)\cdot M_k$ が定義1より成立している．上の式を変形することにより，相対誤差は，$(f^*-f^{k*})/|f^*|\leqq\mathrm{Lip}(f)\cdot M_k/|f^*|$ となる．さらに，$f^{k*}\leqq f^*$ であることより，$\mathrm{Lip}(f)\cdot M_k/|f^*|\leqq\mathrm{Lip}(f)\cdot M_k/|f^{k*}|$ が成立し，相対誤差の上限値を各ステップごとに計算できる．したがって，アルゴリズムは step 7 の収束判定条件：$\{\mathrm{Lip}(f)\cdot M_k/|f^{k*}|\}\leqq\varepsilon$ において，$(f^*-f^{k*})/|f^*|\leqq\varepsilon$ を満たす近似最適値 f^{k*} を得ることができる．

ここで，アルゴリズムの理解を容易にするため，2変数関数を例にとって説明しよう．本アルゴリズムが適用できる関数は，Lipschitz 連続関数のクラスであると前に述べたが，厳密には次に示す［例題1］のように，目的関数や制約関数が$f(x,y)=|g(x,y)|$なる合成関数で表わされる場合でも，関数 g が Lipschitz 連続であればよい．また，目的関数や制約関数がLipschitz連続関数からなる合成関数でもよく，Lipschitz 連続関数の和，差，積，商で表わされる場合にも適用できるので，関数のクラスはもう少し広くなる．アルゴリズムの動きを説明するための例題は，次のとおりである．

［例題1］

$|x^2-y^2|\to\max,$

s.t. $x^2+y^2\leqq 1,\ (x-1)^2+y^2\leqq 1,$

$-1\leqq x\leqq 1,\ -1\leqq y\leqq 1.$

図2は，目的関数の等高線と2つの制約式を表わしており，影をつけたところが実行可能集合である．**［例題1］**は，最適解 $(x,y)=(1,0)$ において最大値1をとることは図2からも明らかである．**図3～図5**は，**［例題1］**にアルゴリズムを適用したとき，各反復時点において，さらに調べるべき分割小領域（影をつけた領域）がどこであるかを示しており，反復回数が増加するにしたがって，調べるべき分割小領域の和集合が，次第に小さくなっていく様子がよくわかる(注2)．このアルゴリズムは，収束判定条件を満足した時点で停止し，最大値を含む領域の代表点とその点での関数値が得られ，目的関数が同一最大値を複数個含む場合にも適用できる．同一最大値もしくは最大値と似通った関数値が，複数個存在する場合には，残された領域の連結成分を数えることにより，アルゴリズムが停止した時点において，いくつの峰が残されているのかがわかり，なおかつその領域の座標もわかるという利点がある．

以上，筆者の提案したアルゴリズムの概要について述べたが，水野[11]はこのアルゴリズムにもとづいて，目的関数のグラジェントがリプシッツ連続であるクラスに対し，分割の仕方に工夫を加えた方法を提案している．

（注2） 図4と図5では，見やすくするため区間領域の全域にわたって格子を切っているが，アルゴリズムでは前のイタレーションにおいて残った領域のみを分割している．

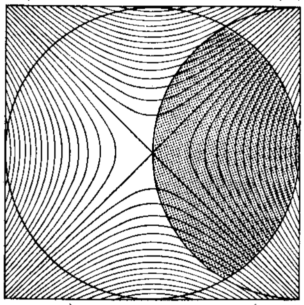

図 2 目的関数の等高線と制約関数

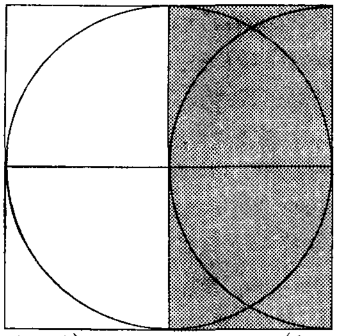

図 3 2nd iteration において残った分割小領域の状態

4.3 分枝限定法を用いたアルゴリズムの考察

非線形最適化問題を分枝限定法によって解くアルゴリズムは，連続変数を分割するため分割操作が無限となり，一般的には近似解法である．本稿で紹介したアルゴリズムは，いずれも厳密解を与えるのはむずかしいが，大域的最適解の ε–近似解を与えることは保証している．Falk–Soland のアルゴリズムの欠点は，多変数関数の凸包絡線を簡単に見つけられないことであり，筆者の提案したアルゴリズムの欠点は，変数の数が多くなった場合，一般に調べるべき部分問題が指数的に増大する傾向にあることである．

一方，2つのアルゴリズムを分枝限定法の立場から眺めてみれば，Falk–Soland のアルゴリズムは目的関数を凸関数に変換しているため深さ優先探索法になっており，筆者の提案したアルゴリズムは，各反復時点において同時に複個の峰を探索しているため，幅優先探索法になっている．しかしながら，記憶容量その他の兼合いで，深さ優先探索法やそれらを組み合わせた方法も構成することができる[19]．また，分枝限定法が使えるための条件は，変数の定義域があらかじめ定まっていなければならないことである．この制約は，一見大きな欠点のように見えるが，現実の問題を解く上では，変数に何らかの制約（たとえば，装置の設計上，設定できる温度や圧力の範囲が制限される）がついていることの方が多いので，欠点とはならない[16]．数値実験結果は，紙数の制約から省略したが，興味のある方は参考文献[14]～[19]を参照されたい．

5. おわりに

大域的最適解の近似解を得るためのアルゴリズムは，今のところ分枝限定法を基礎にした方法が最も有効であるように思われる．また，分枝限定法にもとづいたアルゴリズムは，大域的最適値の上界と下界を正確に求める

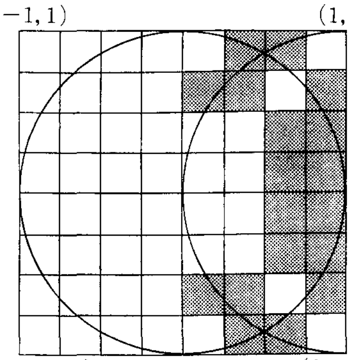

図 4 4th iteration において残った分割小領域の状態

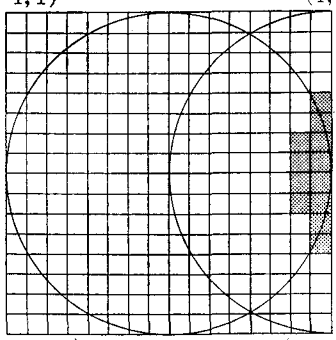

図 5 5th iteration において残った分割小領域の状態

ことにより，大域的最適解を含むシャープな区間を見つけるためのインタバル・アナリシスとも一脈相通じるところがある．アルゴリズムを構築するさいには，問題の構造をよく調べ，その問題特有の分枝限定法を考えた上で，コンピュータを使用することによる情報のロスを最小限に食い止めなければならない．すなわち，目的関数や制約関数などの形状を表わすアナログ情報を，コンピュータで扱えるディジタル情報を用いて，いかにして表現するかが良いアルゴリズムを構築するためのもう1つの鍵である．

最後に，京都大学の茨木俊秀先生から貴重なコメントを賜りましたことを心から感謝いたします．

参考文献

[1] Benson, H. P., "On the Convergence of Two Branch-and-Bound Algorithms for Nonconvex Programming Problems", *JOTA*, **36**, pp. 129-134(1982).

[2] Falk, J. E., "Lagrange Multipliers and Nonconvex Programs", *SIAM J. Control*, **7**, pp. 534-545(1969).

[3] —— and Soland, R. M., "An Algorithm for Separable Nonconvex Programming Problems", *Management Sci.*, **15**, 550-569(1969).

[4] Horst, R., "An Algorithm for Nonconvex Programming Problems", *Math. Prog.*, **10**, pp. 312-321(1976).

[5] ——, "A General Class of Branch-and-Bound Methods in Global Optimization with Some New Approaches for Concave Minimization", *JOTA*, **51**, pp. 271-291(1986).

[6] ——, and Tuy, H., "On the Convergence of Global Methods in Multiextremal Optimization", *JOTA*, **54**, pp. 253-271(1987).

[7] 茨木俊秀，"整数計画法(1)—(5)"，オペレーションズ・リサーチ，1970年9月号～1971年1月号連載．

[8] ——，「組合せ最適化-分枝限定法を中心として-」講座・数理計画法8，産業図書，1983．

[9] Lawler, E. L. and Wood, D. E,. "Branch-and-Bound Methods : A Survey" *Oper. Res.*, **14**, pp. 699-719(1966).

[10] Mitten, L. G., "Branch-and-Bound Methods :General Formulation and Properties", *Oper. Res.*, **18**, pp. 24-34(1970).

[11] 水野眞治，"分枝限定法をもちいた方程式の解法と関数の最小化"，*JORSJ*, **30**, pp. 41-57(1987).

[12] Pintér, J., "Globally Convergent Methods fot n-Dimensional Multiextremal Optimization", *Optimization*, **17**, pp. 187-202(1986).

[13] 整数計画法研究部会，"整数/組合せ計画法の現状(その1)—(その6)"，オペレーションズ・リサーチ，1978年11月号～1979年1月号，1979年5月号～7月号．

[14] 正道寺勉，"一変数多峰性関数に対する最適値探索法の研究"，*JORSJ*, **19**, pp. 295-307(1976).

[15] ——，"多変数多峰性関数に対する最適値探索法の研究"，*JORSJ*, **20**, pp. 311-320(1977).

[16] "非線形システムの最適化に関する研究"，IEレビュー，**19**, pp. 93-98(1978).

[17] ——，"Lipschitzアルゴリズムの分枝限定法としての一考察"，日本経営工学会秋期研究発表会予稿集，pp. 167-168(1979).

[18] ——，"制約条件付き非線形計画問題のための一解法"，日本工業大学研究報告，**16**, pp. 109-115(1986).

[19] ——，"非線形最適化問題を解くためのハイブリッド法について"，日本経営工学会春期研究発表会予稿集，pp. 98-99(1987).

[20] Shubert, B. O., "A Sequential Method Seeking the Global Maximum of a Function", *SIAM J. Numer, Anal.*, **9**, pp. 379-388(1972).

[21] Soland, R. M., "An Algorithm for Separable Nonconvex Programming Problems II : Nonconvex Constraints", *Management Sci.*, **17**, pp. 759-773(1971).