凸2次計画問題に対する容量法の高速度化[†]

佐藤 泰司*・沖田 豪**・伊藤 雅***

Fast Capacity Method for Solving Convex Quadratic Programming Problem

Taiji SATOH*, Tsuyoshi OKITA** and Masaru ITOH***

Quadratic Programming (QP) problem is a natural generalization of well-known Linear Programming (LP) problem. QP problem is formulated as a problem to minimize a linear objective function subject to a system of linear equalities and inequalities.

The continuation method is a powerful tool for solving a system of nonlinear equations. In the continuation method, the continuation parameter β is introduced into the original problem P and a parameterized problem $P(\beta)$ is constructed. Assuming that the optimal solution to $P(\beta)$, for $\beta = 0$, can be trivially or easily found, and that the optimal solution to $P(\beta)$, for $\beta = 1$, will be the desired solution to the original problem P , a trajectory of the solution to $P(\beta)$ is followed by increasing the value of β from 0 to 1.

In this paper, based on the continuation method, a path following method called the capacity method is constructed for solving QP problem. The basic idea of the capacity method is to follow a trajectory of Kuhn-Tucker point when the value of β is increased from 0 to 1. It is shown that the trajectory is expressed as a piecewise-linear function of β . Therefore the optimal solution can be found very efficiently by following the piecewise-linear trajectory.

In this paper, firstly, the assumption that the right-hand-side vector of the constraint equations must be positive, which restricts the applicability of the capacity method, is dropped and a method for transforming any QP problem to the standard form of the capacity method is presented. Secondly, the basis matrix of QP is factorized by using an LDL^T factorization, and the factorization is efficiently updated by using matrix modification techniques. Then the proposed method is applied to some test problems, and numerical results indicate the effectiveness of the method.

Key Words: optimization, quadratic programming, continuation method, capacity method

1. はじめに

凸2次計画問題は線形計画問題の自然な拡張であり、線形の制約条件下で凸2次関数を最小化する問題とし

て定式化される。このような凸2次計画問題に定式化される問題は、最小2乗推定問題や経済学におけるポートフォリオ問題¹⁾、数理計画手法では2次計画問題を繰り返し解くことにより一般の非線形計画問題を解こうとする逐次2次計画法²⁾の部分問題として現われるため、これを効率的に解くことは重要な問題である。

凸2次計画問題の解法としては、Bealの方法³⁾、Lemkeの方法⁴⁾、Wolfeの方法⁵⁾などが提案されている。これらの手法は線形計画問題に対するシンプレックス法を拡張した相補掃出しに基づいており、すべての主変数および双対変数を同時に考慮したシンプレックス表のうえで掃出しを行っている。

本研究では凸2次計画問題の解法として連続変形法⁶⁾の考え方を適用した容量法⁷⁾を取り上げる。この方法

[†] 計測自動制御学会中国支部30周年記念学術講演会で発表 (1993・12)

* 山口大学工学部 宇部市常盤台2557

** 広島市立大学情報科学部 広島市安佐南区沼田町大塚151-5

*** 愛知工業大学工学部 豊田市八草町八千草1247

* Faculty of Engineering, Yamaguchi University, Ube

** Faculty of Information Science, Hiroshima City University, Hiroshima

*** Faculty of Engineering, Aichi Institute of Technology, Toyota

(Received May 29, 1995)

(Revised November 1, 1995)

の基本的な考え方はパラメータの変化に対する Kuhn-Tucker 点を追跡することにより最適解を求めようとするものであり、自明な解から出発して大きさの変わる基底行列を更新しながら最適解に至る方法である。

本研究では、まず、容量法の適用範囲をせばめる原因であった制約条件の右辺が正であるという仮定をはずし、任意の凸 2 次計画問題を容量法の標準形に変換する方法を提案する⁸⁾。さらに、2 次計画問題の基底行列を LDL^T 分解⁹⁾し、その次元の変化に従い LDL^T 分解を逐次更新することにより基底変換の効率化を図る。また、テスト問題に対する数値計算により、基底行列の LDL^T 分解を行わない場合に比べて、計算時間を約 1/3 にできることを示す。最後に、提案手法を相補掃出しによる Lemke の方法と比較し、2.5 倍程度の高速化が達成できることを示す。

2. 定式化

容量法の原論文⁷⁾では以下の形の凸 2 次計画問題を考えている（以下、この問題を容量法の標準形と呼ぶ）。

$$\text{Min. } f(x) = c^T x + \frac{1}{2} x^T C x \quad (1)$$

$$\text{sub. to } Ax \leq b \quad (2)$$

$$e^T x \leq \beta^* \quad (3)$$

$$x \geq 0 \quad (4)$$

ここで、 $A \in R^{m \times n}$, $x \in R^n$, $b \in R^m$, $c \in R^n$, $C \in R^{n \times n}$ であり、 C は正定値対称行列、 b は正ベクトル ($b > 0$) である。また、 $e \in R^n$ はすべての要素が 1 のベクトルであり、 $\beta^* > 0$ とする。

上記の標準形において (3) 式は容量制約と呼ばれ、各変数 x_j の総和が β^* でおさえられるという制約である。与えられた問題にこのような制約がない場合には容量制約は解を追跡するための人為的な制約であると考え、 β^* の値を十分に大きく設定すればよい。

3. 連続変形法

連続変形法⁶⁾においては、与えられた問題 P にスカラパラメータ θ を導入し、パラメータ化された問題 $P(\theta)$ を構成する。パラメータ化された問題においては、 $\theta = 0$ とした問題 $P(\theta = 0)$ が自明な解をもち、 $\theta = 1$ とした問題 $P(\theta = 1)$ が元の問題 P と一致するものとする。このとき、 $0 \leq \theta \leq 1$ に対して $P(\theta)$ の最適解の軌跡が存在し、 θ の値を 0 とした自明な解から θ の値を連続的に増加させて $\theta = 1$ とすることができれば、 $P(\theta)$ の最適解の軌跡を追跡することにより元の問題 P が解けることになる。

このような連続変形法の考え方を凸 2 次計画問題に適用した容量法では、(3) 式の β^* をパラメータ β で置き換えた以下の問題を考える。

$$\text{Min. } f(x) = c^T x + \frac{1}{2} x^T C x \quad (5)$$

$$\text{sub. to } Ax \leq b \quad (6)$$

$$e^T x \leq \beta \quad (7)$$

$$x \geq 0 \quad (8)$$

(5) ~ (8) 式の問題において $\beta = 0$ とすれば自明な解 $x = 0$ が得られる。この状態から β の値を増加させ、それに対応する解を追跡し、 β を本来の値 β^* にすることができれば、元の問題が解けたことになる。ここで、(5) ~ (8) 式の問題は凸計画問題であるから Kuhn-Tucker 条件は最適性の必要十分条件となり、 β の値の変化に対する Kuhn-Tucker 点を追跡すれば元の問題の最適解が得られる。

4. 標準形への変換

制約条件の右辺ベクトルに対する原論文⁷⁾の仮定 ($b > 0$) は $\beta = 0$ に対する自明な解 $x = 0$ が実行可能解であることを保証するためのものである。しかし、このような仮定は容量法の適用範囲をせばめることになる。そこで、以下では自明な解 $x = 0$ が線形制約の内点となるように与えられた任意の凸 2 次計画問題の制約条件を変換する方法を提案する。

変数 x に非負性がない場合には、線形計画法と同様に $x = x^+ - x^-$, $x^+ \geq 0$, $x^- \geq 0$ とおけば変数に非負性がついた標準形となる。また、等式制約 $Ax = b$ は 2 本の不等式制約 $Ax \leq b$, $-Ax \leq -b$ で置き換えられる。したがって、次式のように右辺ベクトルが非正である場合について標準形に変換する手続きを与えれば十分である（一般には $b_i \leq 0$ なる制約に対してのみ以下の変換を行えばよいが、説明を簡単にするために $b \leq 0$ とした）。

$$Ax \leq b, \quad b \leq 0 \quad (9)$$

(9) 式に対して $x = 0$ としたときは実行不能であるので、適当な正ベクトル $0 < \eta \in R^m$ およびスカラー変数 z を導入し、つぎのような制約を考える。

$$Ax + (\eta - b)z \leq \eta \quad (10)$$

(10) 式は $x = 0$, $z = 0$ に対して実行可能であり、また、 $z = 1$ のとき (9) 式と一致する。したがって、最適解において $z = 1$ となるように制約条件および目的関数を設定すればよい。以上より、右辺ベクトル b が非正の場合の標準形はつぎのようになる。

Min.

$$f(x) = \begin{bmatrix} c^T & -M \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x^T & z \end{bmatrix} \begin{bmatrix} C & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \quad (11)$$

sub. to

$$\begin{bmatrix} A & \eta - b \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \leq \begin{bmatrix} \eta \\ 1 \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} e^T & 1 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \leq \beta^* + 1 \quad (13)$$

$$\begin{bmatrix} x \\ z \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (14)$$

ここで、(11) 式の線形係数の M は十分大きな正数であり、2次係数の (2,2) ブロックの要素 1 は行列の正定値性を保証するためのものである。この定式化では $0 \leq z \leq 1$ の制約条件の下で目的関数における z の線形係数を負に設定しており、 z の線形係数の絶対値 (M) が十分大きいとすれば、目的関数を小さくするためには z を大きくした方が有利となり、 z には上限制約があるために最適解においては $z = 1$ となる。また、十分大きな M に対して最適解において $z < 1$ である場合には、与えられた問題は実行可能解をもたない。

5. 容量法の導出

5.1 Kuhn-Tucker 条件

以下では (5) ~ (8) 式の問題を考え、変数 x を正の値をとっている基底 (basis) $x_B > 0$ とゼロとなっている非基底 (nonbasis) $x_N = 0$ に分け、(6) 式を等号が成立しているアクティブ (active) な部分 $A_A x = b_A$ とインアクティブ (inactive) な部分 $A_I x < b_I$ に分割する。この状況を行列表示すれば次式となる。

$$\begin{bmatrix} A_{AB} & A_{AN} \\ A_{IB} & A_{IN} \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} b_A \\ b_I \end{bmatrix}, \quad \begin{bmatrix} x_B \\ x_N \end{bmatrix} > \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (15)$$

また、容量制約 (7) 式は常にアクティブであるものとする (アクティブでないとすれば β の変化に対して解 x は変化しないので、その解が最適となり手続きが終了する)。

これらの分割に対応して Kuhn-Tucker 条件を分割し、停留点条件・実行可能性・相補性・最適性を導出すれば、次式を満たす解は Kuhn-Tucker 点となる。ここで、 y

は $Ax \leq b$ に対するスラック変数である。また、 δ , γ , λ は各々 $Ax \leq b$, $x \geq 0$, $e^T x \leq \beta$ に対するラグランジュ乗数である。

$$c_B + C_{BB} x_B + A_{AB}^T \delta_A + e_B \lambda = 0 \quad (16)$$

$$A_{AB} x_B = b_A \quad (17)$$

$$e_B^T x_B = \beta \quad (18)$$

$$x_B \geq 0, x_N = 0, y_A = 0, y_I \geq 0 \quad (19)$$

$$\lambda(e_B^T x_B - \beta) = 0, \lambda \geq 0 \quad (20)$$

$$\delta_A^T y_A + \delta_I^T y_I = 0, \delta_A \geq 0, \delta_I = 0 \quad (21)$$

$$\gamma_B^T x_B + \gamma_N^T x_N = 0, \gamma_B = 0, \gamma_N \geq 0 \quad (22)$$

ここで、 y_I および γ_N は次式で定義される。

$$y_I = b_I - A_{IB} x_B \quad (23)$$

$$\gamma_N = c_N + C_{NB} x_B + A_{AN}^T \delta_A + e_N \lambda \quad (24)$$

Kuhn-Tucker 条件のうち等式条件 (16) ~ (18) 式は x_B , δ_A , λ に関する連立 1 次方程式である。これを行列表示すれば次式となり、この係数行列が凸 2 次計画問題の基底行列となる。

$$\begin{bmatrix} C_{BB} & A_{AB}^T & e_B \\ A_{AB} & O & 0 \\ e_B^T & 0^T & 0 \end{bmatrix} \begin{bmatrix} x_B \\ \delta_A \\ \lambda \end{bmatrix} = \begin{bmatrix} -c_B \\ b_A \\ \beta \end{bmatrix} \quad (25)$$

付録の定理 A.2 より、この係数行列が正則であるための必要十分条件は行列

$$\begin{bmatrix} A_{AB} \\ e_B^T \end{bmatrix} \quad (26)$$

がフル行ランクをもつことである (正則でないときの処理は後述する)。したがって、この条件が成立するときには (25) 式を数値的に解くことができる。このとき x_B , δ_A , λ は次式のように β の線形関数となり、 β の変化に対して Kuhn-Tucker 点を追跡することができる。

$$\begin{aligned} \begin{bmatrix} x_B \\ \delta_A \\ \lambda \end{bmatrix} &= \begin{bmatrix} C_{BB} & A_{AB}^T & e_B \\ A_{AB} & O & 0 \\ e_B^T & 0^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} -c_B \\ b_A \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} C_{BB} & A_{AB}^T & e_B \\ A_{AB} & O & 0 \\ e_B^T & 0^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \beta \\ &\quad + \begin{bmatrix} C_{BB} & A_{AB}^T & e_B \\ A_{AB} & O & 0 \\ e_B^T & 0^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} -c_B \\ b_A \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} t \\ v \\ \tau \end{bmatrix} \beta + \begin{bmatrix} u \\ w \\ \sigma \end{bmatrix} \quad (27) \end{aligned}$$

5.2 基底変換

パラメータ β を変化させた場合に変数 x_B , δ_A , λ は (27) 式に従って動き, これにより変数 y_I および γ_N は (23), (24) 式に従って動く. このとき成立しなくなる可能性のある条件は容量制約の最適性 $\lambda \geq 0$, 基底変数の実行可能性 $x_B \geq 0$, アクティブ制約の最適性 $\delta_A \geq 0$, 非基底変数の最適性 $\gamma_N \geq 0$ およびインアクティブ制約の実行可能性 $y_I \geq 0$ のいずれかである. 以下, それぞれの場合について考察する.

容量制約に対する双対変数 λ がゼロとなる場合

(27) 式において $\tau < 0$ の場合には β の増加により λ が減少し, $\beta_0 = -\sigma/\tau$ のとき λ はゼロとなる. また, それ以上に β を増加させれば λ は負となり, 目的関数が増加に転じることになる. この場合には $\lambda = 0$ となった時点で容量制約をインアクティブとする.

基底変数 x_B がゼロとなる場合

(27) 式において $t_i < 0$ の場合には β の増加により x_{Bi} が減少し, $\beta_i = -u_i/t_i$ のとき x_{Bi} がゼロとなる. この場合には, 実行可能性を維持するために対応する変数 x_{Bi} を非基底変数とし, その値をゼロで限定する.

双対変数 δ_A がゼロとなる場合

(27) 式において $v_j < 0$ の場合には β の増加により δ_{Aj} が減少し, $\beta_j = -w_j/v_j$ のとき δ_{Aj} がゼロとなる. λ がゼロとなる場合と同様に, β の増加に対してアクティブ制約に対する双対変数 δ_{Aj} が減少してゼロとなる場合であるから, 最適性を維持するためには対応するアクティブ制約をインアクティブとし, その双対変数の値をゼロとする.

双対変数 γ_N がゼロとなる場合

非基底変数に対する双対変数 γ_N を β の関数で表すために次式を計算する.

$$\begin{aligned}\gamma_N &= c_N + C_{NB}x_B + A_{AN}^T\delta_A + e_N\lambda \\ &= c_N + C_{NB}(t\beta + u) + A_{AN}^T(v\beta + w) \\ &\quad + e_N(\tau\beta + \sigma) \\ &= (C_{NB}t + A_{AN}^Tv + e_N\tau)\beta \\ &\quad + (c_N + C_{NB}u + A_{AN}^Tw + e_N\sigma) \\ &= r\beta + s\end{aligned}\quad (28)$$

このとき, $r_k < 0$ の場合には β の増加により γ_{Nk} が減少し, $\beta_k = -s_k/r_k$ のとき γ_{Nk} がゼロとなる. このとき, ゼロで限定されていた非基底変数を限定からはずして基底変数とすることにより, 目的関数をさらに減少させることができる.

スラック変数 y_I がゼロとなる場合

スラック変数 y_I を β の関数で表すために次式を計算する.

$$\begin{aligned}y_I &= b_I - A_{IB}x_B = b_I - A_{IB}(t\beta + u) \\ &= (-A_{IB}t)\beta + (b_I - A_{IB}u) \\ &= p\beta + q\end{aligned}\quad (29)$$

このとき, $p_h < 0$ の場合には β の増加により y_{Ih} が減少し, $\beta_h = -q_h/p_h$ のとき y_I がゼロとなる. これは, β の増加に対してスラック変数 y_I が減少してゼロとなる場合であり, x_B がゼロとなる場合と同様に実行可能性を維持するために, 制約を新たにアクティブ制約の集合に加える.

最適性および実行可能性を維持するために, 上述の手続きで求めた β_0 , β_i , β_j , β_k , β_h および β^* の内で最も小さい値

$$\hat{\beta} = \min\{\beta_0, \beta_i, \beta_j, \beta_k, \beta_h, \beta^*\} \quad (30)$$

を求め, それに対応する基底変換を行えば, β の値をさらに増加させることができる. このとき, β^* が選ばれれば $\beta = \beta^*$ に対応する解が与えられた問題の最適解となる. また, β_0 が選ばれ, $\beta_0 < \beta^*$ ならば最適解において容量制約はアクティブではなく, $\beta = \beta_0$ に対応する解が最適解となる.

5.3 退化の処理

(26) 式がフル行ランクをもたない場合には β の変化に対して解を追跡することができない. この状況は, 基底変数の数とアクティブ制約の数が一致して方程式系に自由度が残っていない場合と, 有効制約の中に冗長な条件がある場合である. そこで, そのような場合には有効制約の数を減らすか基底変数の数を増やす手続きを行う. 有効制約から制約を削除した場合

現在の有効制約から制約条件を 1 本ずつ取り除き, 対応する連立 1 次方程式 (25) 式を解き, (27) 式の線形関数を求める. このとき, β の値を現在の値から増加した場合に取り除いた制約条件が満たされない場合には, 実行不能となるのでその制約を取り除くことはできない. また, 取り除く制約条件の候補が複数存在する場合には, (27) 式のラグランジュ乗数 λ を現在の β において評価し, その値が最大のものを選ぶ.

基底変数に変数を追加した場合

現在の基底変数に非基底変数のうちのいずれかを追加し, 対応する連立 1 次方程式 (25) 式を解き, (27) 式の線形関数を求める. このとき, β の値を現在の値から増加した場合に追加した変数が負となる場合には, 実行不能となるのでその変数を基底に追加することはできない. また, 追加する変数の候補が複数存在する

場合には、(27) 式のラグランジュ乗数 λ を現在の β において評価し、その値が最大のものを選ぶ。

制約条件を削除するか基底変数を追加するかの判定

制約条件を削除した場合と基底変数を追加した場合のラグランジュ乗数 λ を比較し、その値の大きい方に対応する処理を行い、通常の処理に戻る。ここで、制約条件を削除することも基底変数を追加することもできない場合には現在の解が最適解であり、最適解において容量制約はアクティブとはならない。

5.4 アルゴリズム

初期値として $\beta = 0$ とおいた解は x_B および y_A が空であり、退化した基底となっている。このときには5.3節の退化の処理における基底変数の追加を行う。この処理はアルゴリズムの Step 0 で c_j が最小となる変数を基底に入れることに対応している。

アルゴリズム

Step 0 : $\beta = 0$ とおく。目的関数の線形係数の最小値

$$c_k = \min_{j=1,2,\dots,n} c_j$$

を求め、 $x_B = [x_k]$ とする（同じ値のものが複数ある場合にはそれらすべてを基底に入れる）。また、容量制約以外の制約はインアクティブとする。このとき $c_k > 0$ ならば $x = 0$ が最適解である。

Step 1 : (25) 式の連立1次方程式を解き、 x_B 、 δ_A 、 λ を β の線形関数で表す。

Step 2 : 5.2 節の手続きに従って (30) 式の $\hat{\beta}$ を求め、 $\beta = \hat{\beta}$ として対応する基底変換を行う。

Step 3 : Step 2 で得られた基底が退化している場合には、5.3 節の退化の処理を行った後、Step 1 へ戻る。

6. 基底行列の三角化分解

(25) 式の連立1次方程式の解法としては係数行列の LU 分解を行い、前進代入・後退代入を行うコレスキー法が効率的である¹⁰⁾。また、容量法の係数行列は対称行列であり、 β の変化により基底変換が起こった場合には係数行列は1行1列増加または減少する。そこで、各イテレーションにおいて連立1次方程式をコレスキー法により直接解くのではなく、係数行列を LDL^T 分解⁹⁾ しておき、それを逐次更新することにより解法の高速度を図る。以下では、第 k イテレーションにおける連立1次方程式の係数行列を

$$G^{(k)} = \begin{bmatrix} C_{BB} & A_{AB}^T & e_B \\ A_{AB} & O & 0 \\ e_B^T & 0^T & 0 \end{bmatrix} \quad (31)$$

とし、この $G^{(k)}$ の LDL^T 分解が次式のように得られているものとする。

$$G^{(k)} = L^{(k)} D^{(k)} L^{(k)T} \quad (32)$$

6.1 係数行列の次元が増加する場合

非基底であった変数が新たに基底になる場合およびインアクティブ制約が新たにアクティブとなる場合には連立1次方程式の次元が1つ増加する。拡大された行および列を現在の $G^{(k)}$ の後ろに追加するものとするれば、 $G^{(k+1)}$ は

$$G^{(k+1)} = \begin{bmatrix} G^{(k)} & g \\ g^T & \zeta \end{bmatrix} \quad (33)$$

と表わされる。これに対応して LDL^T 分解も以下のよう1行1列拡大される。

$$\begin{aligned} G^{(k+1)} &= L^{(k+1)} D^{(k+1)} L^{(k+1)T} \\ &= \begin{bmatrix} L^{(k)} & 0 \\ \ell^T & 1 \end{bmatrix} \begin{bmatrix} D^{(k)} & 0 \\ 0^T & d \end{bmatrix} \begin{bmatrix} L^{(k)T} & \ell \\ 0^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} L^{(k)} D^{(k)} L^{(k)T} & L^{(k)} D^{(k)} \ell \\ \ell^T D^{(k)} L^{(k)T} & d + \ell^T D^{(k)} \ell \end{bmatrix} \end{aligned} \quad (34)$$

上式より ℓ および d に関する方程式として

$$L^{(k)} D^{(k)} \ell = g \quad (35)$$

$$d = \zeta - \ell^T D^{(k)} \ell \quad (36)$$

が得られる。 $D^{(k)}$ が対角行列であり $L^{(k)}$ が下三角行列であるので、 g が与えられれば (35) 式は前進代入により効率的に解くことができる。また、ここで得られた ℓ を (36) 式に代入すれば対角要素 d が求まる。

6.2 係数行列の次元が減少する場合

正であった基底変数がゼロとなる場合およびアクティブ制約がインアクティブとなる場合には連立1次方程式の次元が1つ減少する。 $G^{(k)}$ から第 j 行および第 j 列が取り除かれるものとし、 $G^{(k)}$ の部分行列を G_1 、 G_2 、 G_3 とする。

$$G^{(k)} = \begin{bmatrix} G_1 & \vdots & G_2 \\ \cdots & \cdot & \cdots \\ G_2^T & \vdots & G_3 \end{bmatrix} \leftarrow j\text{行目} \quad (37)$$

このとき、 $G^{(k+1)}$ は次式で定義される。

$$G^{(k+1)} = \begin{bmatrix} G_1 & G_2 \\ G_2^T & G_3 \end{bmatrix} \quad (38)$$

(37) 式の分割に対応して $L^{(k)}$ の行を分割すれば

$$L^{(k)} = \begin{bmatrix} L_1 & & \\ \cdots & \cdot & \\ & & L_2 \end{bmatrix} \leftarrow j\text{行目} \quad (39)$$

となる. この行列の j 行目を取り除いた行列を $\hat{L}$ とする.

$$\hat{L} = \begin{bmatrix} L_1 & \\ & L_2 \end{bmatrix} \quad (40)$$

この $\hat{L}$ を用いれば $G^{(k+1)}$ は次式となる.

$$G^{(k+1)} = \hat{L} D \hat{L}^T \quad (41)$$

ここで, $\hat{L}$ は右上部分の非ゼロ要素は対角のすぐ上の要素のみであるような下 Hessenberg 行列である. そこで, 以下では $\hat{L}$ を下三角行列に変換する手続きを考える. そのために, (40) 式を縦方向に分割して

$$\hat{L} = \begin{bmatrix} \ddots & & \\ \bar{L}_1 & \ell & \\ & & \bar{L}_2 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & & \\ & d & \\ & & D_2 \end{bmatrix} \quad (42)$$

と表わすことにすれば, 新しい LDL^T 分解として

$$L^{(k+1)} = \begin{bmatrix} L_1 & \\ & \bar{L}_2 \end{bmatrix}, \quad D^{(k+1)} = \begin{bmatrix} D_1 & \\ & \bar{D}_2 \end{bmatrix} \quad (43)$$

が得られる. ただし, $\bar{L}_2$ および $\bar{D}_2$ は次式で与えられる.

$$\bar{L}_2 \bar{D}_2 \bar{L}_2^T = \bar{L}_2 D_2 \bar{L}_2^T + d \ell \ell^T \quad (44)$$

この (44) 式は LDL^T 分解の rank one modification となっているので, 効率的に更新することができる (付録 B 参照).

7. 数値計算

本計算方法の有効性を検証するために, $(m, n) = (10, 10)$ から $(m, n) = (100, 100)$ までの問題を各イテレーションで連立 1 次方程式をコレスキー法により直接解いた場合と LDL^T 分解を更新した場合との比較を行った. テスト問題の係数は区間 $[-5, 5]$ の一様乱数を発生させて設定した. また, 2 次係数行列は正則な下三角行列を L として $C = LL^T$ により求め, 右辺ベクトル b は問題が実行可能解をもつように設定した. なお, 以下に示す計算時間は各サイズごとにランダムに生成した 10 個の問題に対する時間の平均である. また, 使用した計算機は SUN Sparc Station 10/30 であり, プログラムには C 言語を用いた.

提案手法により連立 1 次方程式の係数行列を LDL^T 分解し, それを更新した場合の計算時間を Table 1 に

示す. また, 容量法の連立 1 次方程式を各イテレーションでコレスキー法により直接解いた場合の計算時間を提案手法を基準とした比で Table 2 に示す. さらに, 与えられた問題を相補掃出しによる Lemke の方法で解いた場合の計算時間を同様に提案手法を基準とした比で Table 3 に示す.

まず, Table 2 より, 問題のサイズが小さいところでは三角化分解の更新による計算時間の改善はあまり顕著ではないが, 大きくなるにつれて計算時間の改善率が大きくなり, $(m, n) = (100, 100)$ 程度では約 3 倍の高速化が実現されていることがわかる. また, Table 1 より, 計算時間も中規模の問題である $(m, n) = (100, 100)$ 程度に対して約 4 秒であり, 実用面からみても十分高速な方法となっている.

つぎに, Table 3 より, 問題のサイズが大きくなると提案手法は Lemke の方法に比べて 2.5 倍程度の高速化が実現されていることがわかる. ここで, n が小さく m が大きな問題に対して計算時間の比が非常に大きくなっているのは, Lemke の方法では常に $n + m$ 次元のタブローを取り扱っているのに対し, 容量法の連立 1 次方程式は高々 $2n$ 次元であるためと考えられる.

8. おわりに

本研究では連続変形法の考え方をを用いた凸 2 次計画問題に対する容量法を考察した. まず, 容量法の適用範囲を限定していた原論文における右辺ベクトルに関する仮定 ($b > 0$) をはずし, 任意の凸 2 次計画問題を容量法の標準形に変換する方法を示した. さらに, 基底行列を LDL^T 分解し, 係数行列が拡大・縮小された場合の LDL^T 分解の効率的な更新方法を示した. また, 本手法を用いた数値計算により約 3 倍の高速化が達成できることが示された. この数値計算では係数行列が密な場合を想定しているが, 問題の規模がさらに大きくなり係数行列が疎になった場合に対してスパース処理技法を導入することが今後の課題として考えられる. さらに, この方法を拡張することにより, 容量制約の上限値 β^* が変化した場合のパラメトリック分析を行うこともできる.

参考文献

- 1) 八巻, 矢部: 教養講座 非線形計画法 (2): モデリングと応用, オペレーションズ・リサーチ, 39-8, 665/669 (1994)
- 2) S-P Han: Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems, *Mathematical Programming*, 11, 263/282 (1976)
- 3) 古林 隆: 線形計画法入門, 産業図書 (1980)
- 4) 今野 浩: 線形計画法, 日科技連 (1978)
- 5) A. L. Peressini, F. E. Sullivan and J. J. Uhl, Jr.: *The Mathematics of Nonlinear Programming*, Springer-Verlag, New York (1993)

Table 1 Computation Time (Factorization Update)

$m \backslash n$	CPU Time [sec.] (Factorization Update)									
	10	20	30	40	50	60	70	80	90	100
10	0.03	0.05	0.10	0.17	0.26	0.35	0.51	0.73	1.10	1.40
20	0.07	0.10	0.16	0.27	0.37	0.46	0.67	0.89	1.27	1.63
30	0.08	0.16	0.25	0.43	0.48	0.79	0.84	1.56	1.64	2.16
40	0.13	0.24	0.34	0.47	0.63	0.89	1.46	1.52	1.73	2.38
50	0.13	0.34	0.39	0.60	0.92	0.92	1.32	1.77	2.20	2.60
60	0.15	0.40	0.54	0.65	0.87	1.28	1.52	2.01	3.38	3.00
70	0.18	0.40	0.56	0.79	1.15	1.34	1.90	2.35	3.73	3.50
80	0.19	0.51	0.64	1.02	1.25	1.57	1.97	2.53	3.47	3.63
90	0.25	0.59	0.70	1.10	1.53	1.76	2.76	2.85	3.43	4.19
100	0.21	0.63	0.90	1.18	1.63	2.31	2.71	3.44	3.70	4.33

Table 2 Ratio of Computation Time (Direct / Update)

$m \backslash n$	Ratio of Computation Time (Direct / Update)									
	10	20	30	40	50	60	70	80	90	100
10	1.33	1.40	1.50	1.59	1.58	1.80	1.76	1.92	2.10	2.22
20	1.29	1.70	1.56	1.44	1.73	1.78	2.10	2.06	2.23	2.32
30	1.38	1.75	1.88	1.53	2.13	1.62	2.25	1.72	2.43	2.55
40	1.38	1.71	1.76	2.04	2.11	1.90	1.79	2.57	2.36	2.63
50	1.38	1.82	2.23	2.05	2.05	2.39	2.63	2.69	2.70	2.82
60	1.40	1.78	2.24	2.31	2.52	2.45	2.70	2.91	2.14	3.01
70	1.50	1.90	2.27	2.39	2.55	2.69	2.51	2.79	2.52	3.07
80	1.47	2.02	2.25	2.58	2.66	2.79	2.91	3.07	2.64	3.29
90	1.72	1.92	2.21	2.51	2.75	2.82	2.55	3.48	3.29	3.41
100	1.38	2.02	2.26	2.53	2.77	2.50	3.12	3.32	3.30	3.35

Table 3 Ratio of Computation Time (Lemke / Update)

$m \backslash n$	Ratio of Computation Time (Lemke / Update)									
	10	20	30	40	50	60	70	80	90	100
10	1.00	1.40	1.50	1.59	1.69	1.91	1.82	1.89	1.71	1.67
20	1.57	1.40	1.50	1.52	1.68	2.02	1.90	2.12	1.91	1.86
30	2.88	1.94	1.68	1.42	1.90	1.59	2.17	1.54	1.87	1.81
40	3.54	2.17	2.03	1.83	1.92	1.91	1.50	1.87	2.14	1.91
50	4.85	3.21	2.46	1.78	1.66	2.17	2.05	2.04	2.02	2.17
60	8.00	3.83	4.00	3.95	2.18	2.07	2.14	2.22	1.54	2.11
70	7.78	6.03	5.84	4.09	2.30	3.16	2.08	2.18	1.64	2.09
80	11.11	6.31	5.97	3.48	2.57	2.75	2.35	2.42	2.11	2.31
90	11.16	6.88	8.10	4.45	3.35	3.79	2.41	2.36	2.39	2.32
100	21.81	8.14	7.88	6.68	3.69	2.55	3.22	2.20	2.46	2.67

- 6) S. L. Richter and R. A. DeCarlo, : Continuation Methods: Theory and Applications, *IEEE Trans. Automat. Contr.*, AC-28-6, 660/665 (1983)
- 7) H. S. Houthakker : "The Capacity Method of Quadratic Programming," *Econometrica*, Vol. 28, No. 1, pp. 62-87 (1960)
- 8) 佐藤, 沖田 : 凸2次計画問題に対する容量法の拡張, 計測自動制御学会中国支部30周年記念学術講演会, 146/147 (1993)
- 9) P. E. Gill and W. Murray : Modification of Matrix Factorizations after a Rank-one Change, *The State of the Art in Numerical Analysis*, D. Jacobs ed., Academic Press, 55/83 (1977)
- 10) 伊理, 阪田 : 応用数学3=マトリックス, 培風館 (1982).
- 11) Hestenes, *Calculus of Variations and Optimal Control Theory*, John Wiley (1967)

《付 録》

A. 係数行列の正則性

連続変形法に現れる連立1次方程式の係数行列

$$\begin{bmatrix} C_{BB} & A_{AB}^T & e_B \\ A_{AB} & O & 0 \\ e_B^T & 0^T & 0 \end{bmatrix} \quad (\text{A.1})$$

の正則性に関する定理を示す. 以下では簡単のため部分行列の添字は省略する.

《定理 A.1》 正定値行列 $C \in R^{n \times n}$ および行列 $A \in R^{m \times n}$ ($m \leq n$) に対して

$$\begin{bmatrix} C & A^T \\ A & O \end{bmatrix} \quad (\text{A.2})$$

が正則であるための必要十分条件は A がフル行ランクをもつこと ($\text{rank}\{A\} = m$) である.

証明: C は正定値であり, それゆえ正則であるから

$$\det \begin{bmatrix} C & A^T \\ A & O \end{bmatrix} = \det C \det(-AC^{-1}A^T)$$

が成立する. したがって,

$$\begin{bmatrix} C & A^T \\ A & O \end{bmatrix} \text{が正則}$$

$$\iff AC^{-1}A^T \text{が正則}$$

$$\iff AC^{-1}A^T \text{が正定}$$

$$\iff A^T y \neq 0, \forall y \neq 0 \text{ } (C^{-1} \text{が正定})$$

$$\iff \text{rank}\{A^T\} = m$$

となる. $\square$

また, この定理はもう少し広い形で知られている. すなわち, C は必ずしも全空間で正定値である必要はなく, $Ax = 0$ なる任意の $x \neq 0$ に対して正定値であれば定理の内容は成り立つ¹¹⁾. (注1)

(A.1) 式の C の正定値性は仮定より目的関数の2次係数行列が正定値であることと正定値行列の主座小行列は正定値であることから直接得られる. したがって, 以下の定理が成立する.

《定理 A.2》 正定値行列 $C \in R^{n \times n}$, 行列 $A \in R^{m \times n}$ およびすべての要素が1のベクトル $e \in R^n$ ($m < n$) に対して, 行列

$$\begin{bmatrix} C & A^T & e \\ A & O & 0 \\ e^T & 0^T & 0 \end{bmatrix} \quad (\text{A.3})$$

が正則であるための必要十分条件は行列

$$\begin{bmatrix} A \\ e^T \end{bmatrix} \quad (\text{A.4})$$

がフル行ランクをもつことである. また, $m = n$ の場合には (A.3) 式の行列は正則ではない. $\square$

B. LDL^T 分解の rank one modification

ここでは LDL^T 分解の rank one modification について説明する⁹⁾. 対称行列 B が次式のように LDL^T 分解されているものとする.

$$B = LDL^T \quad (\text{B.1})$$

この B がスカラー α とベクトル y により

$$\bar{B} = B + \alpha yy^T \quad (\text{B.2})$$

と rank one modification されたときの LDL^T 分解

$$\bar{B} = \bar{L}\bar{D}\bar{L}^T \quad (\text{B.3})$$

を元の行列の LDL^T 分解から効率的に求める. ベクトル p を

$$Lp = y \quad (\text{B.4})$$

と定義すれば, $\bar{B}$ は p を用いて

$$\bar{B} = LDL^T + \alpha yy^T = L(D + \alpha pp^T)L^T \quad (\text{B.5})$$

と表わされる. ここで, $D + \alpha pp^T$ を LDL^T 分解すれば

$$D + \alpha pp^T = \tilde{L}\tilde{D}\tilde{L}^T \quad (\text{B.6})$$

と書ける. したがって,

$$\bar{B} = L\tilde{L}\tilde{D}\tilde{L}^T L^T \quad (\text{B.7})$$

となり,

$$\tilde{L} = L\tilde{L} \quad (\text{B.8})$$

$$\tilde{D} = \tilde{D} \quad (\text{B.9})$$

とおけば, $\bar{B}$ の LDL^T 分解が得られることがわかる.

以下では $O(n^2)$ の手間で $\tilde{L}\tilde{L}$ を求める手続きを与える. ここで, $\tilde{L}$ は次式の形をしている⁹⁾.

$$\tilde{L} = \begin{bmatrix} 1 & & & & \\ \beta_1 p_2 & 1 & & & \\ \beta_1 p_3 & \beta_2 p_3 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \beta_1 p_n & \beta_2 p_n & \beta_3 p_n & \cdots & 1 \end{bmatrix} \quad (\text{B.10})$$

そこで, L と $\tilde{L}$ の第1列 $\tilde{\ell}_1$ との積を書き下せば次式が得られる. ただし, ℓ_1 は L の第1列である.

$$\begin{aligned} L\tilde{\ell}_1 &= L \begin{bmatrix} 1 \\ \beta_1 p_2 \\ \beta_1 p_3 \\ \vdots \\ \beta_1 p_n \end{bmatrix} = \beta_1 L \begin{bmatrix} 1/\beta_1 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{bmatrix} \\ &= \beta_1 (Lp + \ell_1/\beta_1 - \ell_1 p_1) \\ &= \beta_1 y + (1 - \beta_1 p_1)\ell_1 \\ &= \ell_1 + \beta_1(y - p_1 \ell_1) \end{aligned} \quad (\text{B.11})$$

これより, つぎのアルゴリズムが得られる. ここで, ℓ_{rj} は L の (r, j) 要素を表わす.

Algorithm (rank one modification)

Step 1: $\alpha_1 = \alpha$, $w^{(1)} = y$ とする.

(注1) この定理に対してコメントをいただきました校閲者に感謝いたします.

Step 2: $j = 1, 2, \dots, n$ に対して次式を計算する.

$$p_j = w_j^{(j)}$$

$$\bar{d}_j = d_j + \alpha_j p_j^2$$

$$\beta_j = \alpha_j p_j / \bar{d}_j$$

$$\alpha_{j+1} = \alpha_j d_j / \bar{d}_j$$

Substep: $r = j + 1, j + 2, \dots, n$ に対して次式を計算する.

$$w_r^{(j+1)} = w_r^{(j)} - p_j \ell_{rj}$$

$$\bar{\ell}_{rj} = \ell_{rj} + \beta_j w_r^{(j+1)}$$

[著 者 紹 介]

佐 藤 泰 司 (正会員)

1980年広島大学工学部第2類(電気系)卒業, 85年広島大学大学院工学研究科博士課程修了, 同年広島大学工学部助手, 91年山口大学工学部電気電子工学科助教授, 現在に至る. 工学博士. おもに, 非線形最適化, 組合せ計画などの研究に従事. 電気学会, 日本OR学会, IEEEなどの会員.

沖 田 豪 (正会員)

1961年広島大学工学部電気工学科卒業, 64年東北大学大学院工学研究科修士課程修了, 同年広島大学工学部助手, 同講師, 同助教授を経て, 78年月山口大学工学部教授, 95年広島市立大学情報科学部教授, 現在に至る. 工学博士. おもに, システム同定の研究に従事. 電気学会, システム制御情報学会などの会員.

伊 藤 雅

1984年愛知工業大学工学部経営工学科卒業, 89年広島大学大学院工学研究科博士課程修了, 同年広島県立大学経営学部講師, 93年愛知工業大学工学部経営工学科講師, 現在に至る. 工学博士. おもに, 数理計画, データ圧縮などの研究に従事. 電気学会, 電子情報通信学会, IEEEなどの会員.