大域的 BMI 求解のための分枝限定法

高野 幸雄*・渡辺 隆男**・安田 恵一郎**

Branch and Bound Technique for Global Solution of BMI

Sachio TAKANO*, Takao WATANABE** and Keiichiro YASUDA**

This paper is concerned with branch and bound technique for the global solution of bilinear matrix inequality (BMI) problem. In this paper, more strict estimation method of lower bound of objective function and new bounding operation are proposed. The effectiveness of proposed method is confirmed in computational experiments.

Key Words: BMI problem, global optimization, branch and bound technique, lower bound estimation, bounding operation

1. はじめに

多くの制御系設計問題が双線形行列不等式 (Bilinear Matrix Inequality, 以下 BMI) の制約上で実行可能な点を見つける問題に帰着できることが示されている³⁾。また, BMI は線形行列不等式 (Linear Matrix Inequality, 以下 LMI) の拡張であり, 枠組みとしては LMI よりも柔軟なものである。これらのことから, 実用的な時間内で BMI を解く方法が要望されるが, LMI のように凸計画問題にはならないため大域的な最適化手法が必要とされている。

このような背景の中で, K.C.Goh らは文献 1) で BMI を解く大域的なアプローチとして分枝限定法を構成している。しかしながら, 現実的な規模の問題に対しては計算時間の面から実用的とは言えないようである。大域的な最適化手法としては, 近年, SA (シミュレーテッド・アニーリング) や GA (遺伝的アルゴリズム) 等の確率論的手法が研究されており, その有効性が示されているが, 分枝限定法は有限時間内で大域的な解を得ることを保証している点で, これら他の手法に比べ魅力的となっている。

そこで, 本論文では, 分枝限定法によってより現実的な規模の BMI を実用的な時間内で解くことを目標とし, 文献 1) の分枝限定法の計算効率の向上を図る。本論文では, より厳密な目的関数の下界値の評価法を提案するとともに, 限定

操作に改良を加える。また, 計算機実験によりその有効性を示す。

なお, 本論文と同種の目的の研究として文献 11) が発表されている。本論文で提案する目的関数の下界の計算は, そこで提案されているものと同一である。しかし, これらの研究は独立に行われたものであり, 文献 11) では下界値の他に上界値の提案と, これらの誤差解析を行っている点, 本論文では限定操作に改良を加えている点がそれぞれ異なっている。

2. 問題設定

変数 $(x, y) \in \mathbf{R}^n \times \mathbf{R}^m$ に関して双線形な行列関数

$$F(x, y) := F_{0,0} + \sum_{i=1}^n x_i F_{i,0} + \sum_{j=1}^m y_j F_{0,j} + \sum_{i=1}^n \sum_{j=1}^m x_i y_j F_{i,j} \quad (1)$$

を定義する。ただし, $F_{i,j} = F_{i,j}^T \in \mathbf{R}^{l \times l}$ とする。BMI の一般的な形は

$$F(x, y) < 0 \quad (2)$$

と与えられる。変数に関して双線形な項が存在することが BMI の特徴であり, 変数 x または y を固定すれば LMI になる。ここで, 以下のスカラー関数

$$\phi(x, y) := \lambda_{\max}\{F(x, y)\} \quad (3)$$

を考える。ただし, $\lambda_{\max}$ は行列の最大固有値を表す。 $\phi(x, y)$ が負であることは, (2) の行列不等式が満たされていることと同値である。したがって (2) を解くには, $\phi(x, y)$ を最小化し, その値を負にすることができれば良い。また, 最小値が非負であれば (2) の解は存在しない。そこで以下では問題

$$\text{minimize } \phi(x, y) \quad (4)$$

* 富士ファコム制御 (株) 日野市富士町 1

** 東京都立大学大学院 八王子市南大沢 1-1

* FUJIFACOM Corporation, Hino

** Graduate School of Engineering, Tokyo Metropolitan University, Hachioji

(Received April 1, 1996)

(Revised March 10, 1997)

について考える。

変数に関して線形な行列関数の最大固有値は凸関数となる。したがって線形行列不等式 LMI は凸計画問題として解くことができる。\$(x, y)\$ に関して双線形な行列関数 \$F(x, y)\$ は変数 \$x\$ または \$y\$ を固定すれば線形な行列関数になる。よって \$\phi(x, y)\$ は \$x\$ または \$y\$ を固定すれば凸関数になる。このような関数を双凸関数という。凸関数には局所的な最小解が大域的な最小解であるという良好な特徴を持ち合わせているが、このことは双凸関数には当てはまらない。これより問題 (4) には大域的な最適化手法が必要とされ、このことが BMI を解くことを LMI に比べ難しくしている。

文献 1) では、最小解を含むような十分大きい超直方体

$$Q_0 := \{(x, y) \mid L_x \leq x \leq U_x, L_y \leq y \leq U_y\} \quad (5)$$

を考え、その領域の中での最小化問題

$$\underset{(x,y) \in Q_0}{\text{minimize}} \quad \phi(x, y) \quad (6)$$

を解く分枝限定法を構成しており、有限時間内で大域的な最小値との誤差が与えられた値 \$\epsilon\$ で抑えられる \$\epsilon\$-大域解が得られることを保証している。本論文では、文献 1) の分枝限定法において、より厳密な下界値の評価法を提案し、さらに限定操作に改良を加えることにより、計算効率の向上を図る。

3. 分枝限定法による BMI 大域的最適化アプローチ

組み合わせ問題における分枝限定法は良く知られているが、連続の問題に対してもこれを応用することができる。連続の問題に分枝限定法を適用している文献として他には 4), 5), 7) 等がある。連続問題に対する分枝限定法では、ある領域における目的関数の最小値の下界値を計算することが必要である。その下界値が現在までに求めた目的関数の最小値よりも大きければその領域を探索領域から外し、そうでなければその領域をさらに複数の領域に分割する、というのが分枝限定法の大まかな方策である。

ここで、変数に関して線形な行列関数

$$F_L(x, y, W) := F_{0,0} + \sum_{i=1}^n x_i F_i + \sum_{j=1}^m y_j F_j + \sum_{i=1}^n \sum_{j=1}^m w_{i,j} F_{i,j} \quad (7)$$

を定義し、その最大固有値を

$$\phi_L(x, y, W) := \lambda_{\max}\{F_L(x, y, W)\} \quad (8)$$

とする。ここで、\$W\$ は \$w_{i,j}\$ を \$(i, j)\$ 成分とする \$n \times m\$ 変数行列である。\$\phi_L(x, y, W)\$ は凸関数であり、非線形制約 \$W = xy^T\$ を加えると \$\phi(x, y)\$ に等しいが、最小化問題

$$\begin{aligned} &\underset{(x,y) \in Q_0}{\text{minimize}} \quad \phi_L(x, y, W) \\ &\text{subject to} \quad W = xy^T, \end{aligned} \quad (9)$$

はもはや凸計画問題ではなく、問題 (6) に比べ何ら難しさが緩和されることはない。しかしながら、反対に制約 \$W = xy^T\$

を外すことによって、関数 \$\phi_L\$ は \$\phi\$ よりも広い範囲の値をとることができ、したがって \$\phi_L\$ の最小値は \$\phi\$ の最小値の下界値を与えることになる。このことを巧く利用することにより、問題 (6) に対する分枝限定法を構成できる。

本章では問題 (6) を解くための文献 1) の分枝限定法のアルゴリズムを紹介する。

3.1 K.C.Goh らの分枝限定法

超直方体 \$Q_0\$ の中での最小化問題

$$\underset{(x,y) \in Q_0}{\text{minimize}} \quad \phi(x, y) \quad (10)$$

を解く文献 1) の分枝限定法のアルゴリズムを以下に示す。

アルゴリズム BB

- step1 定数 \$\epsilon > 0\$ を定め、\$k := 0, s_0 := \{Q_0\}\$, \$L_0 := LB(Q_0), U_0 := UB(Q_0)\$ とおく。
- step2 \$s_k\$ の中から \$LB(\bar{Q}) = L_k\$ となる \$\bar{Q}\$ を選び、\$s_{k+1} := s_k - \{\bar{Q}\}\$ とする。
- step3 \$\bar{Q}\$ をその最も長い辺の中点で二つに分割し、それぞれ \$\bar{Q}_1, \bar{Q}_2\$ とする。
- step4 \$i = 1, 2\$ に対して以下を行う。
 $\{LB(\bar{Q}_i) \text{ を計算する。}$
 $LB(\bar{Q}_i) \leq U_k \text{ ならば } UB(\bar{Q}_i) \text{ を計算して}$
 $s_{k+1} := s_{k+1} \cup \bar{Q}_i \text{ とする。} \}$
- step5 \$U_{k+1} := \min_{Q \in s_{k+1}} UB(Q)\$ とする。
- step6 \$s_{k+1} := s_{k+1} - \{Q \mid LB(Q) > U_{k+1}\}\$ とする。
- step7 \$L_{k+1} := \min_{Q \in s_{k+1}} LB(Q)\$ とする。
- step8 \$k := k + 1\$ とする。\$U_k - L_k < \epsilon\$ ならば終了。
 $U_k - L_k \geq \epsilon \text{ ならば step2 へ行く。}$

上記のアルゴリズム BB において、\$s_k\$ は探索領域として保留されている超直方体 (最大 \$k\$ 個) の集まりを表す。\$LB(Q)\$, \$UB(Q)\$ は \$\min_{(x,y) \in Q} \phi(x, y)\$ の下界値, 上界値である。\$L_k\$, \$U_k\$ は繰り返し \$k\$ 回後の問題 (10) の下界値, 上界値である。

step2 では、下界値が最小の超直方体を選ばれ、step3 で二つに分割される。step4 では分割された二つの超直方体 \$\bar{Q}_1, \bar{Q}_2\$ に対して \$LB(\bar{Q}_i)\$ が計算される。\$LB(\bar{Q}_i) > U_k\$ であれば \$\bar{Q}_i\$ は捨てられ、それ以外は \$UB(\bar{Q}_i)\$ が計算され \$s_{k+1}\$ に加えられる。上界値 \$UB(Q)\$ は \$\phi(x, y)\$ を超直方体 \$Q\$ 上で局所的に最小化することにより求める。つまり、

$$UB(Q) := \underset{(x,y) \in Q}{\text{local min}} \quad \phi(x, y) \quad (11)$$

とする。ただし、記号 local min は関数の局所的な最小値を表すものとする。下界値 \$LB(Q)\$ は関数 (8) を以下で示す制約上で最小化することにより計算する。超直方体 \$Q\$ が \$Q := \{(x, y) \mid L_x \leq x \leq U_x, L_y \leq y \leq U_y\}\$ で与えられたとする。ただし、\$U_x = [U_{x_1} \cdots U_{x_n}]^T\$, \$L_x = [L_{x_1} \cdots L_{x_n}]^T\$, \$U_y = [U_{y_1} \cdots U_{y_m}]^T\$, \$L_y = [L_{y_1} \cdots L_{y_m}]^T\$ である。ここで、

$$\begin{aligned} L_{w_{i,j}} &:= \min \{L_{x_i} L_{y_j}, L_{x_i} U_{y_j}, U_{x_i} L_{y_j}, L_{x_i} U_{y_j}\}, \\ U_{w_{i,j}} &:= \max \{L_{x_i} L_{y_j}, L_{x_i} U_{y_j}, U_{x_i} L_{y_j}, L_{x_i} U_{y_j}\} \end{aligned} \quad (12)$$

とすると, $L_{x_i} \leq x_i \leq U_{x_i}$, $L_{y_j} \leq y_j \leq U_{y_j}$ であれば, $L_{w_{i,j}} \leq x_i y_j \leq U_{w_{i,j}}$ が成立する. したがって,

$$S(Q) := \{(x, y, W) \mid (x, y) \in Q, \\ L_{w_{i,j}} \leq w_{i,j} \leq U_{w_{i,j}}\} \quad (13)$$

とすれば,

$$\{(x, y, xy^T) \mid (x, y) \in Q\} \subset S(Q) \quad (14)$$

が成立する. これより $\min_{(x,y,W) \in S(Q)} \phi_L(x, y, W)$ は $\min_{(x,y) \in Q} \phi(x, y)$ の下界値を与える. 下界値 $LB(Q)$ は

$$LB(Q) := \min_{(x,y,W) \in S(Q)} \phi_L(x, y, W) \quad (15)$$

とする. (11), (15) で計算される $UB(Q)$, $LB(Q)$ は以下の条件を満たす¹⁾.

- (i) $LB(Q) \leq \min_{(x,y) \in Q} \phi(x, y) \leq UB(Q)$
- (ii) $UB(Q) - LB(Q) \rightarrow 0$ ($\text{Size}(Q) \rightarrow 0$)
- (ただし, $\text{Size}(Q)$ は Q の最長の辺の長さを表わす.)

これにより, 上記アルゴリズムの有限時間での収束が保証される⁷⁾.

以上が文献1) で構成された分枝限定法である. なお, (11) を計算する方法は文献2), (15) を計算する方法は文献6), 8), 9) 等がある.

4. より厳密な下界値と新たな限定操作

本章では文献1) の分枝限定法において, より厳密な下界値の評価法を示し, 限定操作の改良を考える.

4.1 より厳密な下界値の評価法

以下では, 超直方体 $Q := \{(x, y) \mid L_x \leq x \leq U_x, L_y \leq y \leq U_y\}$ における $\min_{(x,y) \in Q} \phi(x, y)$ の下界値の評価法について考察する. 双曲面を成す集合

$$B(Q) := \{(x, y, xy^T) \mid (x, y) \in Q\} \quad (16)$$

を定義する. ここで, 明らかに,

$$\min_{(x,y) \in Q} \phi(x, y) = \min_{(x,y,W) \in B(Q)} \phi_L(x, y, W) \quad (17)$$

が成立するから, $B(Q)$ を包含する領域上で $\phi_L(x, y, W)$ を最小化することによって下界値 $LB(Q)$ が得られる. また, その領域が小さいほど厳密な下界値になると期待できる. しかしながら, 領域を表す制約が凸制約でなければ, 凸計画で計算できるという利点なくなる. また既存の凸最適化アルゴリズム^{6), 8), 9)} を用いるには線形制約が簡便である.

文献1) では, 制約

$$(x, y, W) \in S(Q) \quad (18)$$

上で最小化することにより $LB(Q)$ を計算している. Fig.1 において, 直方体が領域 $S(Q)$, 斜線の部分が双曲面 $B_{i,j}$ を表す. Fig.1 において, 三点 $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ を通る平面を $A_{i,j}^1$ とすると, $A_{i,j}^1$ を表す式は以下で与えられる.

$$A_{i,j}^1: w_{i,j} = L_{y_j} x_i + L_{x_i} y_j - L_{y_j} L_{x_i} \quad (19)$$

Fig.1 Subset $S(Q)$ and $B_{i,j}$

一方, 双曲面 $B_{i,j}$ を表す式は

$$B_{i,j}: w_{i,j} = x_i y_j \quad (20)$$

で与えられる. ここで, $L_{x_i} \leq x_i$, $L_{y_j} \leq y_j$ より,

$$x_i y_j - (L_{y_j} x_i + L_{x_i} y_j - L_{y_j} L_{x_i}) \\ = (x_i - L_{x_i})(y_j - L_{y_j}) \geq 0 \quad (21)$$

となる. よって, 双曲面 $B_{i,j}$ は平面 $A_{i,j}^1$ の上側に存在する. つぎに, 三点 $b_{i,j}$, $c_{i,j}$, $d_{i,j}$ を通る平面を $A_{i,j}^2$, 三点 $a_{i,j}$, $b_{i,j}$, $d_{i,j}$ を通る平面を $A_{i,j}^3$, 三点 $a_{i,j}$, $c_{i,j}$, $d_{i,j}$ を通る平面を $A_{i,j}^4$ とすると, これらの平面を表す式は

$$A_{i,j}^2: w_{i,j} = U_{y_j} x_i + U_{x_i} y_j - U_{y_j} U_{x_i}, \\ A_{i,j}^3: w_{i,j} = U_{y_j} x_i + L_{x_i} y_j - U_{y_j} L_{x_i}, \\ A_{i,j}^4: w_{i,j} = L_{y_j} x_i + U_{x_i} y_j - L_{y_j} U_{x_i} \quad (22)$$

で与えられる. 上と同様の議論によって, 双曲面 $B_{i,j}$ は $A_{i,j}^2$ の上側, $A_{i,j}^3$ の下側, $A_{i,j}^4$ の下側に存在することが示される. したがって, 領域

$$\hat{S}(Q) := \{(x, y, W) \mid (x, y) \in Q, \\ \left. \begin{aligned} w_{i,j} &\geq L_{y_j} x_i + L_{x_i} y_j - L_{y_j} L_{x_i}, \\ w_{i,j} &\geq U_{y_j} x_i + U_{x_i} y_j - U_{y_j} U_{x_i}, \\ w_{i,j} &\leq U_{y_j} x_i + L_{x_i} y_j - U_{y_j} L_{x_i}, \\ w_{i,j} &\leq L_{y_j} x_i + U_{x_i} y_j - L_{y_j} U_{x_i} \end{aligned} \right\} \quad (23)$$

は, 集合 $B(Q)$ を含む. また Fig.1 よりも明らかであるが, $\hat{S}(Q)$ は $S(Q)$ に含まれる. つまり, 以下が成立する.

$$B(Q) \subset \hat{S}(Q) \subset S(Q) \quad (24)$$

これより,

$$\min_{(x,y) \in Q} \phi(x, y) = \min_{(x,y,W) \in B(Q)} \phi_L(x, y, W) \\ \geq \min_{(x,y,W) \in \hat{S}(Q)} \phi_L(x, y, W) \\ \geq \min_{(x,y,W) \in S(Q)} \phi_L(x, y, W) \quad (25)$$

が成立する. よって, アルゴリズム BB において,

$$LB(Q) := \min_{(x,y,W) \in \hat{S}(Q)} \phi_L(x, y, W) \quad (26)$$

とすれば、より厳密な下界値が計算される。

$\hat{S}(Q)$ は端点 $a_{i,j}$, $b_{i,j}$, $c_{i,j}$, $d_{i,j}$ を含む最小容積の多面体であるから、 $B(Q)$ の凸包となっている。よって、線形制約によって表現できる、 $B(Q)$ を含む最小の領域である。なお、文献 4) では $L_{x_i} \leq x_i \leq U_{x_i}$, $L_{y_j} \leq y_j \leq U_{y_j}$ における $x_i y_j$ の凸下包 (Convex Lower Envelope) と凹上包 (Concave Upper Envelope) を

$$\begin{aligned} \text{Vex}\{x_i y_j\} &= \max\{L_{y_j} x_i + L_{x_i} y_j - L_{y_j} L_{x_i}, \\ &\quad U_{y_j} x_i + U_{x_i} y_j - U_{y_j} U_{x_i}\}, \\ \text{Cav}\{x_i y_j\} &= \min\{U_{y_j} x_i + L_{x_i} y_j - U_{y_j} L_{x_i}, \\ &\quad L_{y_j} x_i + U_{x_i} y_j - L_{y_j} U_{x_i}\} \end{aligned} \quad (27)$$

と与えており、これらを用いれば $\hat{S}(Q)$ は

$$\begin{aligned} \hat{S}(Q) &= \{(x, y, W) \mid (x, y) \in Q, \\ &\quad \text{Vex}\{x_i y_j\} \leq w_{i,j} \leq \text{Cav}\{x_i y_j\}\} \end{aligned} \quad (28)$$

とも表現できる。ただし、Vex, Cav はそれぞれ凸下包、凹上包を表す。

4.2 限定操作の改良

アルゴリズム BB においては

$$LB(Q) > U_k \quad (29)$$

が満たされたときのみ Q は削除され、それ以外は (29) を満たすまで分割されることになる。分割されることにより漸近的下界値の精度が高まり、(29) を満たす可能性が高くなる。しかし、アルゴリズム BB では分割によってのみ下界値の精度を高くすることができる。このことから、分枝の数 (Q の数) が増大することが懸念される。分枝数の増加は計算効率低下の原因になる。また、一般に計算機の記憶容量には限界があり、分枝数は制限される。これらのことから、大域解を取り逃がさない範囲で Q を分割せずに縮小することが望ましいと考える。そこで、本節では新たな限定操作として、 Q を分割せずに縮小する方法を示す。

反復 k における大域解の上界値が U_k で与えられているとする。このとき、大域解の存在する範囲は

$$C_B := \{(x, y) \mid U_k I - F(x, y) > 0\} \quad (30)$$

と表現できる。領域

$$C_L := \{(x, y, W) \mid U_k I - F_L(x, y, W) > 0\} \quad (31)$$

を定義すると C_B は

$$C_B = \{(x, y) \mid (x, y, W) \in C_L, W = xy^T\} \quad (32)$$

とも表現できる。つぎに、

$$\begin{aligned} \bar{C}_B(Q) &:= \{(x, y) \mid (x, y, W) \in C_L \cap B(Q)\}, \\ \bar{C}_S(Q) &:= \{(x, y) \mid (x, y, W) \in C_L \cap \hat{S}(Q)\} \end{aligned} \quad (33)$$

を定義すれば、 $B(Q) \subset \hat{S}(Q)$ より、

$$\bar{C}_B(Q) \subseteq \bar{C}_S(Q) \subseteq Q \quad (34)$$

Fig. 2 Hyper-Rectangle Q and $\hat{Q}$

が成り立つ。 $\hat{S}(Q)$, C_L は凸領域であるから、 $\bar{C}_S(Q)$ も凸領域である。しかし、 $\bar{C}_B(Q)$ は一般に凸とはいえない。 $\bar{C}_B(Q)$ に外接する超直方体を計算できれば理想的であるが、これは問題 (8) と同程度に難しいと考えられる。これに対して、 $\bar{C}_S(Q)$ に外接する超直方体は、以下に示すように $2(n+m)$ 回の凸計画問題を解くことによって計算される。

(i) 凸計画による Q の縮小

超直方体

$$\hat{Q} = \{(x, y) \mid \hat{L}_x \leq x \leq \hat{U}_x, \hat{L}_y \leq y \leq \hat{U}_y\} \quad (35)$$

が $\bar{C}_S(Q)$ に外接しているとすれば、

$$\begin{aligned} \hat{L}_{x_i} &= \min_{(x, y, W) \in C_L \cap \hat{S}(Q)} x_i, & \hat{U}_{x_i} &= \max_{(x, y, W) \in C_L \cap \hat{S}(Q)} x_i, \\ \hat{L}_{y_j} &= \min_{(x, y, W) \in C_L \cap \hat{S}(Q)} y_j, & \hat{U}_{y_j} &= \max_{(x, y, W) \in C_L \cap \hat{S}(Q)} y_j \end{aligned}$$

が成り立つ。ところで、制約

$$(x, y, W) \in C_L \cap \hat{S}(Q) \quad (36)$$

は LMI 制約によって表現される。すなわち、超直方体 $\hat{Q}$ は、LMI 制約上で線形関数を最小化する凸計画問題

$$\begin{aligned} &\text{minimize} && c^T z \\ &\text{subject to} && F_L(z) > 0 \end{aligned} \quad (37)$$

を $2(n+m)$ 回解くことによって求められる。ただし、 z は x_i , y_j , $w_{i,j}$ を要素とする $n+m+nm$ 次元空間のベクトルとし、 c は一つの要素のみが 1 または -1 の値をとり他の要素が 0 の値をとる $n+m+nm$ 次元空間の定数ベクトルとする。このようにして求められる $\hat{Q}$ と Q の関係を Fig. 2 に図示する。

実際に計算する場合には、一つの凸計画を解いた後に Q を縮小できる可能性があり、したがって、つぎの凸計画を解く際に、その縮小された Q を用いたほうが効率が良い。つまり、凸計画が解かれるごとに逐次 Q を更新することで、 $\hat{Q}$ よりも小さい超直方体になる。また、縮小された超直方体に対して、 $2(n+m)$ 回の凸計画を解く同じ手順を施すことによって更に縮小できる可能性もある。

以上のことを考慮して以下に Q を縮小する手順をまとめる。

手順 1

step 1 定数 $\epsilon > 0$, 最大繰り返し数 $k_{\max}$ を定める。

$k := 0$, $\hat{Q} := Q$ とする。

Fig. 3 Procedure 2

step 2 $\hat{Q}_0 := \hat{Q}$ とする.

step 3 $i = 1, \dots, n$ に対して以下を順に行う.

$\{\hat{L}_{x_i} := \min_{(x,y,W) \in C_L \cap \hat{S}(\hat{Q})} x_i \text{ として } \hat{Q} \text{ を更新.}$
 $\hat{U}_{x_i} := \max_{(x,y,W) \in C_L \cap \hat{S}(\hat{Q})} x_i \text{ として } \hat{Q} \text{ を更新.}\}$

step 4 $j = 1, \dots, m$ に対して以下を順に行う.

$\{\hat{L}_{y_j} := \min_{(x,y,W) \in C_L \cap \hat{S}(\hat{Q})} y_j \text{ として } \hat{Q} \text{ を更新.}$
 $\hat{U}_{y_j} := \max_{(x,y,W) \in C_L \cap \hat{S}(\hat{Q})} y_j \text{ として } \hat{Q} \text{ を更新.}\}$

step 5 $k := k + 1$ とする. $k \geq k_{\max}$ あるいは,

$1 - \text{Vol}(\hat{Q}_0)/\text{Vol}(\hat{Q}) < \epsilon$ ならば終了.

それ以外ならば step 2 へ行く.

ただし,

$$\text{Vol}(Q) := \left\{ \prod_{i=1}^n (U_{x_i} - L_{x_i}) \right\} \left\{ \prod_{j=1}^m (U_{y_j} - L_{y_j}) \right\}.$$

なお, 上記の手順で凸計画が解かれるごとに $\hat{Q}$ が更新されるので, 途中で $C_L \cap \hat{S}(\hat{Q})$ が空集合になることがある. このときは $\hat{Q}$ の中には大域解は存在せず, よって $\hat{Q}$ は削除できる.

この手順によって Q は分割をせずに縮小できるが, 同じ値を持つ大域解が二つ以上離れて存在する場合には, それらを含む Q は, それらの大域解を含む最小の超直方体より小さくすることはできない. したがって, 縮小する操作だけでなく, 分割する操作も必要である.

手順1は $LB(Q)$ を計算するのに比べ必要となる計算量が多い. また, 手順1は上界値 U_k が更新されたときに特に効果が大きい. これらのことから, 上界値 U_k が大きく更新されたときのみ手順1を行うのが効果的であるが, その時点で分枝の数 (Q の数) が多く必要となる計算量も多いので, 以下に示す手順2を行うのが適当であると考え.

手順2

step 1 すべての Q_i を含む最小の超直方体

$Q_A := \{(x, y) \mid L_x \leq x \leq U_x, L_y \leq y \leq U_y\}$
を求める.

step 2 Q_A に対して手順1を行い, 縮小された超直方体を $\hat{Q}_A$ とする.

step 3 $Q_i := Q_i \cap \hat{Q}_A$ とする.

手順2の様子を Fig. 3 に示す.

以上述べた手順は $LB(Q)$ を計算するのと同一規模の凸計画を $2(n+m)$ 回解くことが必要である. 以下では $LB(Q)$ を計算するよりも少ない計算量で縮小する方法を述べる.

(ii) 楕円体近似による Q の縮小

領域 $C_L \cap \hat{S}(Q)$ は LMI の実行可能集合として表現される. LMI の実行可能集合については, それを含むような楕円体の計算法が文献6), 8) で与えられている. 楕円体に外接する超直方体は比較的容易に計算できる. 以下に, 文献6) の LMI 実行可能集合を包含する楕円体の計算法とその楕円体に外接する超直方体の計算法を示す.

LMI の実行可能集合

$$\mathbf{X} := \left\{ z \mid F_L(z) = F_0 + \sum_{i=1}^{n+m+nm} z_i F_i > 0 \right\} \quad (38)$$

を定義する. ただし, $F_i = F_i^T \in \mathbb{R}^{l \times l}$ とする. $\mathbf{X}$ を包含する, つまり, $\mathbf{X} \subseteq E_{out}$ を満たす楕円体は

$$E_{out} = \{ z \mid (z - z^*)^T H (z - z^*) \leq l(l-1) \} \quad (39)$$

で与えられる⁶⁾. ただし, z^* は $\mathbf{X}$ の解析的中心

$$z^* := \underset{z \in \mathbf{X}}{\text{argmax}} \det F_L(z) \quad (40)$$

である. また, H の (i, j) 要素は,

$$H_{i,j} = \text{Trace}\{F_L(z^*)^{-1} F_i F_L(z^*)^{-1} F_j\} \quad (41)$$

である. 楕円体 E_{out} に外接する

$$Q_{E_{out}} = \{ z \mid L_z \leq z \leq U_z \} \quad (42)$$

で表される超直方体は

$$L_{z_i} = z_i^* - \sqrt{l(l-1) a_i^T H^{-1} a_i},$$

$$U_{z_i} = z_i^* + \sqrt{l(l-1) a_i^T H^{-1} a_i} \quad (43)$$

として計算される. ただし, a_i は i 番目の要素を1とし, 他の要素を0とする $n+m+nm$ 次元空間のベクトルである.

これらを用いて $C_L \cap \hat{S}(Q)$ を含む $n+m+nm$ 次元空間の超直方体

$$Q_L = \{(x, y, W) \mid L_{x_i} \leq x_i \leq U_{x_i},$$

$$L_{y_j} \leq y_j \leq U_{y_j},$$

$$L_{w_{i,j}} \leq w_{i,j} \leq U_{w_{i,j}}\} \quad (44)$$

を計算することができ,

$$C_L \cap B(Q) \subseteq C_L \cap \hat{S}(Q) \subseteq Q_L \quad (45)$$

が成り立つ. したがって,

$$\hat{Q} = \{(x, y) \mid (x, y, W) \in Q_L\}$$

$$= \{(x, y) \mid L_x \leq x \leq U_x, L_y \leq y \leq U_y\} \quad (46)$$

とすれば,

$$\bar{C}_B(Q) \subseteq \hat{Q} \quad (47)$$

が成り立つ. このままでは $\hat{Q}$ が Q よりも大きくなることがあるので $\hat{Q} \cap Q$ を用いればよい. このようにして求められる $\hat{Q}$ と Q の関係を Fig. 4 に図示する.

楕円体近似による方法は, 解析的中心 z^* と正定対称行列 H を計算する必要があるが, $LB(Q)$ を計算する際に, 解析的中心を追跡するアルゴリズム⁶⁾を用いれば, その過程でそれらは求めることができ, 計算量の増分はわずかですむ.

Fig. 4 Hyper-Rectangle Q and $\hat{Q}$

5. 計算機実験

本章では、4章で述べた改良案の有効性を検証するために、以下の二つの問題に対して計算機実験を行った。

Example 1)

$$\begin{aligned} &\text{minimize} \quad \lambda_{\max}\{F_1(x, y)\} \\ &\text{subject to} \quad -10 \leq x \leq 10, -10 \leq y \leq 10 \end{aligned}$$

$$F_1(x, y) :=$$

$$\begin{aligned} &\begin{bmatrix} -10 & -0.5 & -2 \\ -0.5 & 4.5 & 0 \\ -2 & 0 & 0 \end{bmatrix} + x \begin{bmatrix} -1.8 & -0.1 & -0.4 \\ -0.1 & 1.2 & -1 \\ -0.4 & -1 & 0 \end{bmatrix} \\ &+ y \begin{bmatrix} 9 & 0.5 & 0 \\ 0.5 & 0 & -3 \\ 0 & -3 & -1 \end{bmatrix} + xy \begin{bmatrix} 0 & 0 & 2 \\ 0 & -5.5 & 3 \\ 2 & 3 & 0 \end{bmatrix} \end{aligned}$$

Example 2)

$$\begin{aligned} &\text{minimize} \quad \lambda_{\max}\{F_2(x, y)\} \\ &\text{subject to} \quad -10 \leq x_i \leq 10, -10 \leq y_j \leq 10 \\ &\quad \text{for } i = 1, 2, j = 1, 2, 3 \end{aligned}$$

$$F_2(x, y) := \begin{bmatrix} (A + bf)P + P(A + bf)^T & 0 \\ 0 & -P \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & -1 \\ 2 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 0.2 \end{bmatrix},$$

$$f = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T, \quad P = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \end{bmatrix}$$

Example 1) は文献 1) から引用している。Example 2) は安定化状態フィードバックを求める問題に相当する。これら二つの問題に対して以下の 4 つの方法を比較した。

- BB_{KCG} : 文献 1) の分枝限定法。
- BB_{LB} : 下界値評価を改善した方法。
- BB_{LB+E} : 下界値評価の改善と、 $LB(Q)$ を計算した後に、 Q が保留される場合には楕円体近似による限定操作を Q に対して行う方法。

(a) Progress of U_k and L_k (b) Progress of difference $U_k - L_k$

(c) Number of rectangles

— BB_{KCG}, - - - BB_{LB},
- · - · - BB_{LB+E}, ····· BB_{LB+E+M}

Fig. 5 Results for Example 1)

Table 1 Results of the computational experiments

	Method	No. of Partial Problems			Max. No. of Rects.	No. of Iterations	CPU time (second)
		BMILOC	LMIGLO	MINCX			
Exp. 1)	BB _{KCG}	212	287	—	69	143	552
	BB _{LB}	15	29	—	1	14	54
	BB _{LB+E}	10	15	—	2	7	36
	BB _{LB+E+M}	5	7	16	2	3	47
Exp. 2)	BB _{KCG}	15316~	20001~	—	5313~	10000~	68063~
	BB _{LB}	361	677	—	61	338	2336
	BB _{LB+E}	221	387	—	53	193	1483
	BB _{LB+E+M}	102	181	40	34	90	849

BB_{LB+E+M} : 方法 BB_{LB+E} において, U_k が更新された場合に手順 2 ($k_{\max} = 2, \epsilon = 0.1$) を行う方法.

方法 BB_{LB+E+M} においての U_k が更新したかの判定は

$$\bar{U} - U_k > 0.1(U_k - L_k) \quad (48)$$

を用いた. ただし, $\bar{U}$ は前回手順 2 を行った際の U_k の値とする. 手順 2 はアルゴリズム BB において, $k = 0$ では step1 の後に行い, $k > 0$ では step7 の後に行った. なお, $UB(Q)$ を計算するのには文献 2) の内点法を用い, $LB(Q)$ は文献 6) の内点法を用いた. 楕円体近似による限定操作を行う際には $LB(Q)$ を計算する際にその過程で z^* と H を計算した. 手順 1 の凸計画問題を解くには, LMI Control Toolbox¹⁰⁾ のコマンド `mincx` を用いた. 収束判定は $U_k - L_k < 0.0001$ を用いた. 計算にはパーソナルコンピュータ (PPC603e 100MHz) を使用した. Example 2) において BB_{KCG} は 10000 ステップを越えたので計算を打ち切った.

Example 1) における U_k と L_k の時間推移, $U_k - L_k$ の時間推移, 分枝の数の時間推移を Fig. 5 に示す. Example 2) については Fig. 6 に示す. また, 収束に要した時間, 解かれた問題の数等を Table 1 にまとめる. Table 1 において, BMILOC は (11) を計算する問題, LMIGLO は (15) または (26) を計算する問題, MINCX は問題 (37) を表す.

5.1 考察

提案する下界値の評価法を用いることで, 両問題において 10 倍以上の高速化が達成されている. Fig. 5(b), Fig. 6(b) より, 必要とされる精度が高くなる程この差は顕著になると予想できる. 提案する限定操作による効果については, BMILOC, LMIGLO, MINCX の問題を解くのに要する時間の割合に依存するので, 方法 BB_{LB}, BB_{LB+E}, BB_{LB+E+M} の間で, 一般的に優劣を断定するのは難しい. 本計算機実験では, Example 1) においては BB_{LB+E}, Example 2) においては BB_{LB+E+M} が最短時間で収束した. 収束に要するステップ数については両問題で BB_{LB+E+M} が最小となった.

6. おわりに

BMI 大域的求解のためのアプローチとして, K.C.Goh らが分枝限定法を構成しているが, 本論文ではより厳密な目的

(a) Progress of U_k and L_k

(b) Progress of difference $U_k - L_k$

(c) Number of rectangles

— BB_{KCG}, - - - BB_{LB},
 ····· BB_{LB+E}, - · - · BB_{LB+E+M}

Fig. 6 Results for Example 2)

関数の下界値の評価法を提案し、さらに新たな限定操作を加えた。また、計算機実験によってこれらの有効性を検証した。

これらの改善によって計算効率が向上したが、筆者らの経験では、いまだに現実的規模のBMIを解くにはしばしば多大な計算時間を必要とする。本分枝限定法は単純な分枝操作をおこなっており、分枝操作には改善の余地がある。この改善によってさらなる計算時間の短縮化が期待できるが、これは今後の課題とする。

参 考 文 献

- 1) K.C.Goh, M.G.Safonov and G.P.Papavassilopoulos: A Global Optimization Approach for the BMI Problem, Proc. of CDC, 2009/2014 (1994)
- 2) K.C.Goh, L.Turan, M.G.Safonov, G.P.Papavassilopoulos and J.H.Ly: Biaffine Matrix Inequality Properties and Computational Methods, Proc. of ACC, 850/855 (1994)
- 3) M.G.Safonov, K.G.Goh and J.H.Ly: Control System Synthesis via Bilinear Matrix Inequality, Proc. of ACC, 45/49 (1994)
- 4) Faiz A.Al-Khayyal and James E. Falk: Jointly Constrained Biconvex Programming, Maths. Ops. Res., **8**, 273-286 (1983)
- 5) Faiz A.Al-Khayyal: Jointly Constrained Bilinear Programming and Related Problem: An Overview, Computers Math. Applic., **19**, 53/62 (1990)
- 6) S.Boyd and L.E.Ghaoui: Method of Centers for Minimizing Generalized Eigenvalues, Linear Algebra and Its Applications, **188-189**, 63/111 (1993)
- 7) V.Balakrishnan and S.Boyd: Global Optimization in Control System Analysis and Design, Control and Dynamic Systems, **53**, 1/55 (1992)
- 8) Yuri Nesterov and Arkadii Nemirovskii: Interior-Point Polynomial Algorithms in Convex Programming, SIAM (1994)
- 9) Arkadii Nemirovskii and Pascal Gahinet: The Projective Method for Solving Linear Matrix Inequalities, Proc. of ACC, 840/844 (1994)
- 10) Pascal Gahinet, Arkadii Nemirovskii, Alan J.Laub, Mahmoud Chilali: LMI Control Toolbox For Use with MATLAB, The MathWorks, Inc. (1995)
- 11) 星島, 藤岡: BMI 固有値問題の解の上界および下界とその性質, 第40回システム制御情報学会研究発表講演会論文集, 453/454 (1996)

[著 者 紹 介]

高 野 幸 雄 (正会員)

1996年3月, 東京都立大学大学院工学研究科修士課程電気工学専攻修了。同年4月富士ファコム制御株式会社入社, 現在に至る。在学中はLMI及びBMIによる制御系設計に関する研究に従事。

渡 辺 隆 男 (正会員)

1994年3月, 東京都立大学大学院工学研究科修士課程電気工学専攻修了。同年4月同大学工学部助手, 現在同大学院工学研究科助手。ロバスト制御の研究に従事。IEEE, 電気学会の会員。

安 田 恵 一 郎 (正会員)

1989年3月, 北海道大学大学院工学研究科博士課程電気工学専攻修了。同年4月東京都立大学工学部助手, 91年同大学工学部助教授となり現在同大学院工学研究科助教授。主として, 大規模システムの計画・運用・制御に関する研究に従事(工学博士)。電気学会の会員。