半正定値計画問題に対する行列補完理論の高速実装

山下　真

半正定値計画問題は基礎的な数理最適化問題の一つであり，幅広い応用問題に用いられる．これらの応用問題を実用的な時間で扱うには，半正定値計画問題をいかに短時間で求解するかが重要である．応用問題から生じる半正定値計画問題では，入力行列が疎行列であることが多く，その構造的な疎性を活用して計算時間短縮を目指すのが行列補完型内点法である．本稿では，行列補完型内点法の計算手法を改良することで，さらなる高速化が得られることを報告する．

キーワード：半正定値計画問題，主双対内点法，行列補完理論，ソフトウェア

1. はじめに

半正定値計画問題（SemiDefinite Programs，以下SDP）は，基礎的な数理最適化問題の一つであり，制御理論や量子理論などをはじめとして数多くの応用を持っている．理論的にも主双対内点法による求解が多項式時間と効率的であることが知られており，主双対内点法を実装したSDPを解くソフトウェアの発展も著しい．また，多項式最適化や組合せ最適化などでは，問題の制約を一部緩和することでSDPに定式化し，このSDPを求解することにより優れた近似解を得る手法も幅広く利用されており，SDPをさらに短時間で求解することには強い需要が存在している．応用問題から生じるSDPの多くは入力行列が疎行列となっており，このような疎性の活用は高速なSDP求解の核心の一つである．

本稿は，SDPに関する研究の中でも，構造的な疎性を活用して主双対内点法の効率を向上させることで短時間での求解を行う，行列補完型内点法に焦点を当てる．SDPに関する一般的な導入については2010年7月号の特集「半正定値計画に対するソルバーと応用例」[7]が詳しいので，そちらも参照してほしい．

SDPの標準形は，一般に以下のような主双対形式で表現できる．

$$
\begin{aligned}
(\mathcal{P}) \quad & \min \ : \ \boldsymbol{A}_0 \bullet \boldsymbol{X} \\
& \text{s.t.} \ : \ \boldsymbol{A}_k \bullet \boldsymbol{X} = b_k (k = 1, \ldots, m), \ \boldsymbol{X} \succeq \boldsymbol{O} \\
(\mathcal{D}) \quad & \max \ : \ \textstyle\sum_{k=1}^{m} b_k z_k \\
& \text{s.t.} \ : \ \textstyle\sum_{k=1}^{m} \boldsymbol{A}_k z_k + \boldsymbol{Y} = \boldsymbol{A}_0, \ \boldsymbol{Y} \succeq \boldsymbol{O}
\end{aligned}
$$

やました　まこと
東京工業大学大学院情報理工学研究科 数理・計算科学専攻
〒152–8552 東京都目黒区大岡山 2–12–1–W8–29

ここで，記号として，$\mathbb{S}^n$ を n 次対称行列とする．主問題 $(\mathcal{P})$ の変数は $\boldsymbol{X} \in \mathbb{S}^n$ であり，制約 $\boldsymbol{X} \succeq \boldsymbol{O}$ は $\boldsymbol{X}$ が半正定値行列であることを示している．なお，$\boldsymbol{X} \succ \boldsymbol{O}$ は $\boldsymbol{X}$ が正定値行列であることを示す．対称行列 $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{S}^n$ の内積は $\boldsymbol{U} \bullet \boldsymbol{V} = \sum_{i=1}^{n} \sum_{j=1}^{n} U_{ij} V_{ij}$ で定義されるとする．また，双対問題 $(\mathcal{D})$ の変数は，$(\boldsymbol{Y}, \boldsymbol{z}) \in \mathbb{S}^n \times \mathbb{R}^m$ である．入力データは $\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{S}^n$ と，$b_1, \ldots, b_m \in \mathbb{R}$ である．

SDPの理論的な計算時間は主問題の制約の本数 m と変数行列 $\boldsymbol{X}$，$\boldsymbol{Y}$ の次数 n で簡便的に見積もることができるが，さまざまな応用における計算では入力行列 $\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{S}^n$ が疎行列（n^2 個の要素のほとんどがゼロであり，非ゼロ要素のみをメモリ上に保持したほうが効率的な行列）であることが多い．（逆にほとんどの要素が非ゼロであり，非ゼロ要素のみでなくすべての要素を保持したほうが計算が容易となる行列は密行列と呼ばれる．）非ゼロ要素の構造として，$\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{S}^n$ の統合疎パターン $\mathcal{A} \in \mathbb{S}^n$ を

$$
\mathcal{A}_{ij} = \begin{cases} 1 & \text{if } [\boldsymbol{A}_k]_{ij} \neq 0 \text{ for some } k = 0, \ldots, m \\ 0 & \text{otherwise} \end{cases}
$$

と定義する．ただし，本稿では表記を明確にするために，行列 $\boldsymbol{A}$ の (i, j) 成分を，A_{ij} だけでなく $[\boldsymbol{A}]_{ij}$ と記すこともある．

統合疎パターンの例として，量子化学におけるスピングラスから生じるSDP [4] について，$n = 3{,}375$ の場合の統合疎パターンで1となる要素を図示したものが図1である．全要素数 $n^2 = 11{,}390{,}625$ に対して，非ゼロ要素の数は23,625個であり0.20%に留まる．さらに図1から，統合疎パターンが構造的なパターン

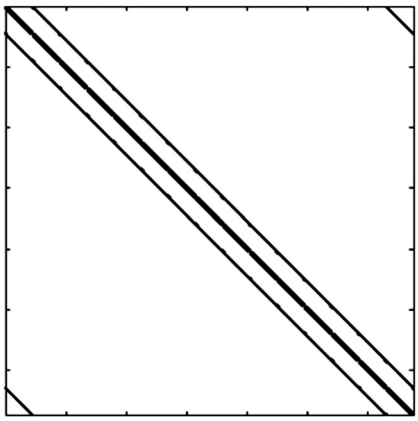

図 1　スピングラスの SDP における統合疎パターン

をなしていることも見て取れる．

このような構造的な疎パターンを利用して主双対内点法の計算時間短縮を進めるのが，行列補完型内点法 [2, 5] である．行列補完型内点法は SDPA-C (SDPA Completion method) というソフトウェアに実装され，組合せ最適化から生じる SDP などを短時間で求解することに成功している [5]．行列補完型内点法では，統合疎パターンに基づいて変数行列 $\boldsymbol{X}$，$\boldsymbol{Y}$ を複数の行列に分解して保持する．しかしながら，行列補完型内点法を用いても従来の内点法同様に，Schur 補完行列の計算が主要な計算ボトルネックである．

本稿では，複数の行列に分解する分解式を修正することで，Schur 補完行列の計算時間を短縮した内容を報告する．従来の計算式では通常の Cholesky 分解ではなかったため CHOLMOD [1] などの Cholesky 分解ソフトウェアを直接利用することができなかった．今回の改良された分解式では，逆行列の Cholesky 分解に注目することで，これらのソフトウェアを活用できることを見いだし，計算時間の短縮につながった．

本稿の構成として，行列補完型内点法について概説した後，分解式の改良について述べる．また，マルチスレッドを用いることで，スピングラスなどの SDP について，さらなる計算時間短縮が得られることを数値実験結果を通して示す．ただし，紙面の都合で数式の一部は天下り的に紹介しており，第 2 節の数式の詳細については，[2, 5] を参照してほしい．

以下では，行列 $\boldsymbol{X}$ について行集合 $I \subset \{1, 2, \ldots, n\}$ と列集合 $J \subset \{1, 2, \ldots, n\}$ による小行列を $\boldsymbol{X}_{I,J}$ と表すこととする．具体的には，$\boldsymbol{X}_{\{2,8\},\{3,5\}} = \begin{pmatrix} X_{23} & X_{25} \\ X_{83} & X_{85} \end{pmatrix}$ のような小行列である．さらに，行集合と列集合が一致するときには，$\boldsymbol{X}_I := \boldsymbol{X}_{I,I}$ で小行列を表す．

2. 内点法と行列補完理論

主双対内点法は $\mathcal{F} = \{(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{z}) \in \mathbb{S}^n \times \mathbb{S}^n \times \mathbb{R}^m : \boldsymbol{X} \succ \boldsymbol{O}, \boldsymbol{Y} \succ \boldsymbol{O}\}$ という内点の集合の中で点列を生成し，主問題 $(\mathcal{P})$ と双対問題 $(\mathcal{D})$ の最適解に近づけていく反復法である．以下に簡単な枠組みを示すが，[7] やその参考文献などに詳しい．

主双対内点法の枠組み

1. $(\boldsymbol{X}^0, \boldsymbol{Y}^0, \boldsymbol{z}^0)$ を $\mathcal{F}$ から適切に選ぶ．また，パラメータ γ を $0 < \gamma < 1$ から適切に選択する．反復回数を $h = 0$ とする．

2. $(\boldsymbol{X}^h, \boldsymbol{Y}^h, \boldsymbol{z}^h)$ が終了条件を満たせば，$(\boldsymbol{X}^h, \boldsymbol{Y}^h, \boldsymbol{z}^h)$ を $(\mathcal{P})$ と $(\mathcal{D})$ の最適解として出力して終了．

3. 修正ニュートン法により探索方向 $(\Delta\boldsymbol{X}, \Delta\boldsymbol{Y}, \Delta\boldsymbol{z})$ を計算する．

4. $\mathcal{F}$ に留まる最大ステップ長 $\alpha_{\max}$ を $\alpha_{\max} = \max\{\alpha \geq 0 : \boldsymbol{X}^h + \alpha\Delta\boldsymbol{X} \succeq \boldsymbol{O}, \boldsymbol{Y}^h + \alpha\Delta\boldsymbol{Y} \succeq \boldsymbol{O}\}$ で求める．

5. $(\boldsymbol{X}^{h+1}, \boldsymbol{Y}^{h+1}, \boldsymbol{z}^{h+1}) = (\boldsymbol{X}^h, \boldsymbol{Y}^h, \boldsymbol{z}^h) + \gamma\alpha(\Delta\boldsymbol{X}, \Delta\boldsymbol{Y}, \Delta\boldsymbol{z})$ として反復点を更新し，$h = h + 1$ として 2. に戻る．

上記の枠組みで多くの計算時間が集中するのが，探索方向 $(\Delta\boldsymbol{X}, \Delta\boldsymbol{Y}, \Delta\boldsymbol{z})$ の計算であり，特に $\Delta\boldsymbol{z}$ を得るために解く $\boldsymbol{B}\Delta\boldsymbol{z} = \boldsymbol{r}$ という連立方程式の係数行列 $\boldsymbol{B}$ の計算が主なボトルネックである．この係数行列 $\boldsymbol{B} \in \mathbb{S}^m$ は Schur 補完行列と呼ばれることが多く，その要素は，$B_{ij} = (\boldsymbol{X}^h\boldsymbol{A}_i(\boldsymbol{Y}^h)^{-1}) \bullet \boldsymbol{A}_j$ で計算される．以下では各反復での Schur 補完行列の計算については，反復回数 h を省略し，

$$B_{ij} = (\boldsymbol{X}\boldsymbol{A}_i\boldsymbol{Y}^{-1}) \bullet \boldsymbol{A}_j \tag{1}$$

と表記する．

ここで，$\boldsymbol{Y}$ は双対問題 $(\mathcal{D})$ の制約より，$\boldsymbol{Y} = \boldsymbol{A}_0 - \sum_{k=1}^m \boldsymbol{A}_k z_k$ である．よって，$\mathcal{A}_{ij} = 0$ ならば $Y_{ij} = 0$ となり，$\boldsymbol{Y}$ は統合疎パターンの疎性を継承している．しかしながら，評価式 (1) に現れる $\boldsymbol{X}$，$\boldsymbol{Y}^{-1}$ は一般にすべての要素を非ゼロと扱うべき密行列となり，従来の内点法では統合疎パターンから得られる疎

性を $\boldsymbol{X}$，$\boldsymbol{Y}^{-1}$ に用いることができない．

行列補完型内点法の基本的なアイデアは，$\boldsymbol{X}$，$\boldsymbol{Y}^{-1}$ の代わりに，$\boldsymbol{X}$，$\boldsymbol{Y}$ を複数の小さい行列に分解することで統合疎パターンの疎性を可能な限り維持して，評価式 (1) を計算することである．

例えば，$n=3$ で統合疎パターンと変数行列が

$$\mathcal{A}=\begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix},\quad \boldsymbol{X}=\begin{pmatrix}X_{11}&X_{12}&X_{13}\\X_{12}&X_{22}&X_{23}\\X_{13}&X_{23}&X_{33}\end{pmatrix} \tag{2}$$

であるとき，内積の性質から行列 $\boldsymbol{X}$ の X_{13} 要素は $\boldsymbol{A}_k \bullet \boldsymbol{X}=b_k$ の制約に関係しない．さらに，$\boldsymbol{X}_{\{1,2\}}=\begin{pmatrix}X_{11}&X_{12}\\X_{12}&X_{22}\end{pmatrix}$，$\boldsymbol{X}_{\{2,3\}}=\begin{pmatrix}X_{22}&X_{23}\\X_{23}&X_{33}\end{pmatrix}$ の 2 つの小行列の値が決まっており，$\boldsymbol{X}_{\{1,2\}}\succ\boldsymbol{O},\boldsymbol{X}_{\{2,3\}}\succ\boldsymbol{O}$ の正定値条件が満たされれば，$X_{13}=X_{12}X_{33}^{-1}X_{23}$ と $\{(i,j):\mathcal{A}_{ij}=0\}$ の値を適切に補完することで，$\boldsymbol{X}$ 全体を正定値 ($\boldsymbol{X}\succ\boldsymbol{O}$) にすることができる．このことを行列補完と呼ぶ．つまり，この例では，$\boldsymbol{X}$ 全体で計算する必要はなく，統合疎パターンから導出される小行列のみで計算を進めることが可能である．

(2) の例は小さい例であり，統合疎パターンからどのように小行列に分解されるかを見るために，以下では $n=7$ の場合のもう少し大きな統合疎パターンを取り上げる．

$$\mathcal{A}=\begin{array}{c}\\1\\2\\3\\4\\5\\6\\7\end{array}\begin{array}{c}\begin{array}{ccccccc}1&2&3&4&5&6&7\end{array}\\\begin{pmatrix}1&1&&1&&&\\1&1&1&&&&1\\&1&1&1&&&1\\1&&1&1&1&&\\&&&1&1&1&\\&&&&1&1&1\\&1&1&&&1&1\end{pmatrix}\end{array} \tag{3}$$

どのような複数の小行列に分解すれば $\boldsymbol{X}$ として正定値行列に行列補完できるかどうかは，グラフ理論におけるコーダルグラフの性質と密接に関係している．まず，行列 $\boldsymbol{X}$ が正定値であることと，ある正則な下三角行列 $\overline{\boldsymbol{L}}$ で $\boldsymbol{X}=\overline{\boldsymbol{L}}\,\overline{\boldsymbol{L}}^T$ の積に $\boldsymbol{X}$ を Cholesky 分解できることは同値である．しかし，一般に $X_{ij}=0$ となる (i,j) についても $\overline{\boldsymbol{L}}_{ij}\neq 0$ となることがあり，この現象は fill-in と呼ばれる．コーダルグラフを用いると fill-in となる (i,j) が特定でき，詳細は紙面の都合で省略するが，これが正定値行列に行列補完できることに関係している．

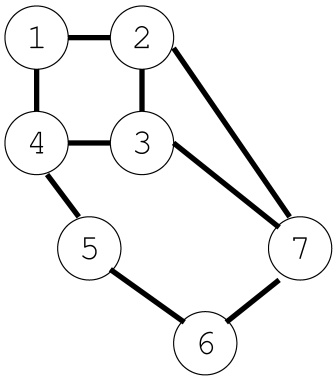

図 2　統合疎パターン (3) に対応するグラフ

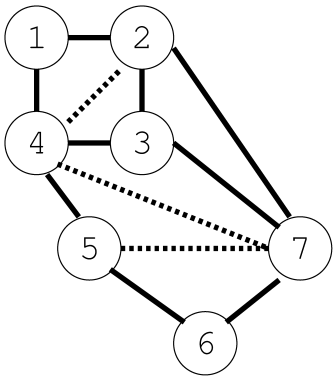

図 3　統合疎パターン (3) を拡張したコーダルグラフ

(3) の統合疎パターンの各行と各列を頂点とし，$\mathcal{A}$ で 1 となっている要素を枝として表現したグラフが図 2 である．あるグラフがコーダル (chordal) グラフであるとは，4 以上のすべてのサイクルに弦 (chord) があることである．図 2 は $1-2-3-4-1$ と $3-7-6-5-4-3$ のそれぞれ 4，5 本の枝で構成されるサイクルに弦がないためコーダルグラフではない．

ここで，コーダルグラフにするために $2-4, 4-7, 5-7$ に枝を計 3 本追加してできたのが，図 3 である．図 3 には極大クリークとなる 4 つの頂点集合があり，それぞれ $C_1=\{1,2,4\}$，$C_2=\{2,3,4,7\}$，$C_3=\{4,5,7\}$，$C_4=\{5,6,7\}$ である．頂点集合がクリークであるとは，その頂点集合の任意の 2 つの頂点の間に枝があることであり，極大クリークとはほかのクリークに含まれないクリークであることである．

コーダルグラフから疎パターン行列に戻して生成されるのが拡張疎パターン $\mathcal{E}$ である．この例では以下の (4) となり，四角で囲まれた 1 が追加された枝に対応している．さらに，四角で囲まれた 1 は Cholesky 分

解で fill-in が起こる位置の要素でもある．

$$\mathcal{E} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1 & 1 & & 1 & & & \\ 1 & 1 & 1 & 1 & & & 1 \\ & 1 & 1 & 1 & & & 1 \\ 1 & 1 & 1 & 1 & 1 & & 1 \\ & & & 1 & 1 & 1 & 1 \\ & & & & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \tag{4}$$

一般に極大クリークが $C_1, \ldots, C_\ell$ と ℓ 個得られたときに，拡張疎パターン $\mathcal{E}$ の非ゼロ要素は，極大クリークによって以下のように求めることができる．

$$\mathcal{E}_{ij} = 1 \Leftrightarrow (i,j) \in \bigcup_{r=1}^{\ell} C_r \times C_r$$

このことは，拡張疎パターン $\mathcal{E}$ の非ゼロ要素が $\boldsymbol{X}_{C_1}, \ldots, \boldsymbol{X}_{C_\ell}$ によってカバーされる，ということでもある．ここで Grone ら [3] の行列補完理論により，$\boldsymbol{X}_{C_r} \succ \boldsymbol{O}$ $(r = 1, \ldots, \ell)$ を満たせば，$\boldsymbol{X}_{C_1}, \ldots, \boldsymbol{X}_{C_\ell}$ から正定値行列 $\widehat{\boldsymbol{X}}$ に補完することが可能となる．この行列補完の具体的な計算方法は [2, 5] で示されており，

$$\widehat{\boldsymbol{X}} = \boldsymbol{L}^T \boldsymbol{D} \boldsymbol{L} \tag{5}$$

の形式で与えられる．ここで，$\boldsymbol{L}$ は正則な下三角行列であり，正則な下三角行列の $\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_{\ell-1}$ による積 $\boldsymbol{L} = \boldsymbol{L}_{\ell-1} \cdots \boldsymbol{L}_2 \boldsymbol{L}_1$ で構成される．これらの要素は，極大クリークから得られる集合

$$S_r := C_r \backslash (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_{r+1})$$
$$U_r := C_r \cap (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_{r+1})$$

を用いて

$$[\boldsymbol{L}_r]_{ij} = \begin{cases} 1 & (i = j) \\ \left[\boldsymbol{X}_{U_r U_r}^{-1} \boldsymbol{X}_{U_r S_r}\right]_{ij} & (i \in U_r, j \in S_r) \\ 0 & \text{otherwise} \end{cases}$$

で求まる．また，$\boldsymbol{D}$ は対角ブロック構造を持つ正定値行列で

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{S_1 S_1} & & & \\ & \boldsymbol{D}_{S_2 S_2} & & \\ & & \ddots & \\ & & & \boldsymbol{D}_{S_\ell S_\ell} \end{pmatrix}$$

となり，それぞれの対角ブロックは，

$$\boldsymbol{D}_{S_r S_r} = \begin{cases} \boldsymbol{X}_{S_r S_r} - \boldsymbol{X}_{S_r U_r} \boldsymbol{X}_{U_r U_r}^{-1} \boldsymbol{X}_{U_r S_r} & (r = 1, 2, \ldots, \ell - 1) \\ \boldsymbol{X}_{S_\ell S_\ell} & (r = \ell) \end{cases}$$

で与えられる．

上の (5) で得られた $\widehat{\boldsymbol{X}}$ は $\widehat{\boldsymbol{X}} \succ \boldsymbol{O}$，$\widehat{\boldsymbol{X}}_{C_r} = \boldsymbol{X}_{C_r}$ $(r = 1, \ldots, \ell)$ を満たし，$\boldsymbol{X}_{C_r} (r = 1, \ldots, \ell)$ から正定値行列 $\widehat{\boldsymbol{X}}$ に補完される．しかしながら，(2) の例で $X_{13} \neq 0$ となることからも推測できるとおり，$\widehat{\boldsymbol{X}}$ は $\mathcal{E}$ の疎パターンを失い，すべての要素を非ゼロと扱う密行列として計算を行わなければならない．このことは，Schur 補完行列の評価式 (1) に現れる双対問題側の行列 $\boldsymbol{Y}^{-1}$ でも同様であり，$\boldsymbol{Y}^{-1}$ も密行列である．

その一方で行列 $\boldsymbol{L}^{-1}$ は，その構築の過程から $\mathcal{E}$ の構造を継承できることが [2, 5] で示されている．つまり，$\mathcal{E}_{ij} = 0$ であれば，$[\boldsymbol{L}^{-1}]_{ij} = 0$ である．双対問題側でも同様に $\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{M}^T$ と下三角行列 $\boldsymbol{M}$ に Cholesky 分解すると，$\mathcal{E}_{ij} = 0$ であれば，$M_{ij} = 0$ となる．

このことから，$\mathcal{E}$ の非ゼロ要素数が n^2 に対して数パーセント未満に留まるようなスピングラスの SDP においては，密行列の $\widehat{\boldsymbol{X}}$ や $\boldsymbol{Y}^{-1}$ を扱うよりも拡張疎パターン $\mathcal{E}$ にのみ非ゼロ要素を持つ $\boldsymbol{L}^{-1}, \boldsymbol{M}$ を計算に用いるほうが効果的と考えられる．例えば，ベクトル $\boldsymbol{v} \in \mathbb{R}^n$ に対して $\boldsymbol{w} := \widehat{\boldsymbol{X}}\boldsymbol{v}$ を計算するのであれば，$\widehat{\boldsymbol{X}}$ を密行列として計算するよりも $\boldsymbol{L}^{-1}\boldsymbol{w}_1 = \boldsymbol{v}, \boldsymbol{w}_2 = \boldsymbol{D}\boldsymbol{w}_1, \boldsymbol{L}^{-T}\boldsymbol{w} = \boldsymbol{w}_2$ の 3 段階の計算のほうが計算コストの面で有利である．特に，$\boldsymbol{L}^{-1}$ が下三角であるため，$\boldsymbol{L}^{-1}\boldsymbol{w}_1 = \boldsymbol{v}$ は前進代入だけで $\boldsymbol{w}_1$ が得られる．

このことを Schur 補完行列の評価式 (1) に反映させると，

$$\begin{aligned} B_{ij} &= (\widehat{\boldsymbol{X}}\boldsymbol{A}_i\boldsymbol{Y}^{-1}) \bullet \boldsymbol{A}_j \\ &= \sum_{k=1}^{n} \boldsymbol{e}_k^T (\widehat{\boldsymbol{X}}\boldsymbol{A}_i\boldsymbol{Y}^{-1}\boldsymbol{A}_j)\boldsymbol{e}_k \\ &= \sum_{k=1}^{n} (\widehat{\boldsymbol{X}}\boldsymbol{e}_k)^T \boldsymbol{A}_i (\boldsymbol{Y}^{-1}[\boldsymbol{A}_j]_{*k}) \\ &= \sum_{k=1}^{n} ((\boldsymbol{L}^{-T})^{-1}\boldsymbol{D}(\boldsymbol{L}^{-1})^{-1}\boldsymbol{e}_k)^T \boldsymbol{A}_i (\boldsymbol{M}^{-T}\boldsymbol{M}^{-1}[\boldsymbol{A}_j]_{*k}) \end{aligned}$$

と評価式を変形できる．ただし，$\boldsymbol{e}_k$ は k 要素だけ 1 の単位ベクトルであり，$[\boldsymbol{A}_j]_{*k}$ は $\boldsymbol{A}_j$ の第 k 列ベクトル，つまり $\boldsymbol{A}_j\boldsymbol{e}_k$ である．

行列補完型内点法は，このように計算式を変形することで，密行列 $\widehat{\boldsymbol{X}}$，$\boldsymbol{Y}^{-1}$ を構築することなく，拡張疎

パターン $\mathcal{E}$ の要素のみを計算に用いて主双対内点法を実行するタイプの内点法である．スピングラスから発生する SDP は，図 1 に見たように $\mathcal{A}$ が構造的な疎パターンをなしており，さらに対応するグラフを作ると，枝を追加せずともコーダルグラフになるため，$\mathcal{A}=\mathcal{E}$ という特徴もあわせ持つことがわかる．よって，サイズが大きくなるほどに行列補完型内点法の効果が得られやすい SDP である．

3. 分解式の改良と高速化

これまでの行列補完型内点法の難点は，$\widehat{\boldsymbol{X}}$ の分解式 (5) において，対角ブロック行列 $\boldsymbol{D}$ が存在するために通常の Cholesky 分解 $\widehat{\boldsymbol{X}}=\overline{\boldsymbol{L}}\,\overline{\boldsymbol{L}}^T$ の形式にできず，CHOLMOD [1] に代表されるような Cholesky 分解のためのソフトウェアを利用できないことにある．

この状況を克服するために注目したのが，$\widehat{\boldsymbol{X}}$ の逆行列の Cholesky 分解

$$\widehat{\boldsymbol{X}}^{-1}=\widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^T \tag{6}$$

である．$\widehat{\boldsymbol{L}}$ は $\widehat{\boldsymbol{X}}^{-1}$ を経由することなく，$\boldsymbol{X}_{C_r}(r=1,\ldots,\ell)$ から以下の手順によって直接求められる．

1. $\boldsymbol{X}_{C_r}^{-1}=\boldsymbol{L}_r\boldsymbol{L}_r^T$ に Cholesky 分解 $(r=1,\ldots,\ell)$
2. $\widehat{\boldsymbol{L}}_{C_rS_r}$ に $[\boldsymbol{L}_r]_{C_rS_r}$ を代入 $(r=1,\ldots,\ell)$

例えば，(2) の例は $C_1=\{1,2\},C_2=\{2,3\}$ を極大クリークとして持つ $n=3$ の場合と考えられる．この場合も，統合疎パターン $\mathcal{A}$ に対応するグラフがコーダルグラフであることから拡張疎パターン $\mathcal{E}$ は $\mathcal{A}$ に一致する．ここで，$\boldsymbol{X}_{C_1}\succ\boldsymbol{O},\boldsymbol{X}_{C_2}\succ\boldsymbol{O}$ のときには，

$$\begin{aligned}
&\boldsymbol{X}_{C_1}^{-1}=\boldsymbol{L}_1\boldsymbol{L}_1^T,\quad \boldsymbol{X}_{C_2}^{-1}=\boldsymbol{L}_2\boldsymbol{L}_2^T\\
\Rightarrow&\begin{pmatrix}X_{11}&X_{12}\\X_{12}&X_{22}\end{pmatrix}^{-1}=\begin{pmatrix}\ell_{11}&\\\ell_{12}&\ell_{22}\end{pmatrix}\begin{pmatrix}\ell_{11}&\ell_{12}\\&\ell_{22}\end{pmatrix},\\
&\begin{pmatrix}X_{22}&X_{23}\\X_{23}&X_{33}\end{pmatrix}^{-1}=\begin{pmatrix}m_{22}&\\m_{23}&m_{33}\end{pmatrix}\begin{pmatrix}m_{22}&m_{23}\\&m_{33}\end{pmatrix}
\end{aligned}$$

と要素ごとに書き下すと，$S_1=\{1\},S_2=\{2,3\}$ であることから $\widehat{\boldsymbol{L}}$ は

$$\widehat{\boldsymbol{L}}=\begin{pmatrix}\ell_{11}&&\\\ell_{12}&m_{22}&\\0&m_{23}&m_{33}\end{pmatrix}$$

と得られる．(5) によって得られる $\widehat{\boldsymbol{X}}$ と比較すると，$\widehat{\boldsymbol{X}}^{-1}=\widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^T$ が成り立つことが確認できる．この考え方は，統合疎パターンと拡張疎パターンが一致しないような一般の状況にも拡張でき，上記手順で $\widehat{\boldsymbol{L}}$ が求められることが証明できる．

さらに，この例では，$[\widehat{\boldsymbol{L}}]_{13}=0$ となっているが，これは偶然ではなく (2) の例で $\mathcal{E}_{13}=0$ であることに起因している．実際に $\widehat{\boldsymbol{L}}$ は拡張疎パターン $\mathcal{E}$ の疎性を維持しており，密行列の $\widehat{\boldsymbol{X}}$ よりも計算コストの点で有利である．

なお，上記手順では C_r，S_r の情報が必要となるが，これは CHOLMOD のアルゴリズム supernodal 法から直接得られる．主双対内点法は反復計算であるが，C_r,S_r は反復の途中で変わることがないため，反復計算に入る前の前処理として，まずは入力行列 $\boldsymbol{A}_0,\boldsymbol{A}_1,\ldots,\boldsymbol{A}_m$ から統合疎パターン $\mathcal{A}$ を構成し，supernodal 法を適用することで C_r,S_r を得ておく．各反復では $\boldsymbol{X}_{C_r}$ の情報を更新し $\boldsymbol{L}_r$ を計算することで，CHOLMOD のデータ構造を直接利用しながら $\widehat{\boldsymbol{L}}$ をメモリ上に保持することが可能となる．Schur 補完行列の計算は，$\widehat{\boldsymbol{X}}^{-1}=\widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^T$ の分解より

$$B_{ij}=\sum_{k=1}^{n}(\widehat{\boldsymbol{L}}^{-T}\widehat{\boldsymbol{L}}^{-1}\boldsymbol{e}_k)^T\boldsymbol{A}_i(\boldsymbol{M}^{-T}\boldsymbol{M}^{-1}[\boldsymbol{A}_j]_{*k})$$

となるが，$\widehat{\boldsymbol{L}}$ を CHOLMOD のデータ構造のまま保持することで，$\widehat{\boldsymbol{L}}^{-T}\widehat{\boldsymbol{L}}^{-1}\boldsymbol{e}_k,\boldsymbol{M}^{-T}\boldsymbol{M}^{-1}[\boldsymbol{A}_j]_{*k}$ の計算も CHOLMOD のルーチンを活用できる．

従来の分解式 (5) を実装した既存ソフトウェアが SDPA-C Version 6 （以下 SDPA-C6） であり，新しい分解式 (6) の導入によってソフトウェア全体を一新したのが SDPA-C Version 7（以下 SDPA-C7） である．表 1 は，SDPA-C6 と SDPA-C7 の SDP の求解に必要な計算時間をまとめたものである．SDPA-C7 の列にある括弧付き数字は，後述するマルチスレッドに関するものである．数値実験は，3.00 GHz の CPU (Xeon X5365) と 48 GB のメモリ空間を搭載した Linux 上で行った．

表 1 で解いた SDP は 300×10 の格子ネットワーク上で生成した最大クリーク問題から発生しているものであり，変数行列 $\boldsymbol{X}$，$\boldsymbol{Y}$ の次元は $n=300\times10=3{,}000$ である．この問題では，極大クリークは 438 個生成され，その最大の要素数 $\max\{|C_1|,|C_2|,\ldots,|C_\ell|\}$ は 59 である．ただし，$|C|$ は集合 C の要素数を表す．表 1 を見ると Schur 補完行列の計算が 4205 秒から 2094 秒と半分以下の時間に短縮され，新しい分解式 (6) の優位性が見て取れる．これによって，全体の時間としても 4729.28 秒かかっていた SDP を 2515.50 秒で求解できている．

また，SDPA-C7 では新しい分解式とともにマルチスレッド計算も導入されている．最近の CPU では CPU

表 1 最大クリーク問題に対する計算時間（単位は秒）

	SDPA-C6	SDPA-C7(1)	SDPA-C7(4)
Schur 補完行列の評価時間	4205.70	2094.03	731.98(2.86x)
全体の時間	4729.28	2515.50	889.15(2.82x)

表 2 スピングラスに対する計算時間（単位は秒）

p	n	クリーク数 (ℓ)	平均クリークサイズ	SDPA-C7	SDPA7
10	1000	155	25.69	20.77	11.85
15	3375	191	29.97	336.40	560.40
18	5832	1118	28.13	2136.88	1522.60
20	8000	1737	25.75	4552.10	3726.03
25	15625	3173	29.50	24913.20	26023.67

の中に複数の計算コアがあることが多く，それぞれの計算コアに別の計算をスレッドとして担当させることで並列計算が可能となっている．

Schur 補完行列では，B_{ij} の評価は j に関する各列ごとに独立している．列ごとの計算単位に分解し異なるスレッドに担当させることで SDPA-C7 ではマルチスレッドによる並列計算を行っている．スレッドの割り当てについては，4 スレッド使用可能な場合には，まず $\boldsymbol{B}$ の 1 列目から 4 列目を第 1 スレッドから第 4 スレッドに割り当てる．そのあとで，$\boldsymbol{B}$ の 5 列目以降の計算は，割り当てられた列の計算が先に終了したスレッドから順に割り当てることとしている．実際のソフトウェア実装では，スレッドの衝突を防ぐために内部的な線形代数演算のスレッド数の管理などが全体の性能向上のために欠かせない．

このマルチスレッドの効果は，表 1 の SDPA-C7(1) と SDPA-C7(4) の列を比較することで確認できる．SDPA-C7(4) では 4 スレッドによる並列計算を行っている．4 スレッドの使用によって，Schur 補完行列の計算では，2.86 倍の高速化，全体としても 2.82 倍の高速化を達成しており，比較的良好な並列計算の効果を得ている．

次に，スピングラスから生じる SDP の計算時間をまとめたものが表 2 である．ここでは行列補完型ではない通常の主双対内点法を実装した SDPA [6] についても計算時間を掲載している．なお，SDPA-C7, SDPA7 については 4 スレッドを使用した．スピングラスの SDP は，パラメータ p から生成されるが，p の 3 乗が変数行列の次数 n となる．48 GB のメモリ空間で生成できる SDP としては，表 2 の最下行にある $p = 25$ が最大である．また，平均クリークサイズは $\sum_{r=1}^{\ell} |C_r|/\ell$ で求めている．

表 2 の結果より，p が小さいときには SDPA7 が SDPA-C7 より高速であるが，p が 18, 20 と大きくなるにしたがって差は縮まり，$p = 25$ では逆転して SDPA-C7 が SDPA7 よりも短時間で求解している．このことは，p が大きくなっても平均クリークサイズが増加しないことによる．行列補完型内点法では極大クリークのサイズに合わせて変数行列が小行列に分解されるため，極大クリークが小さいほど有利となるのである．

4. 最後に

本稿では，行列補完型内点法の要点である変数行列の分解式を改良することで計算時間の短縮を行った，という内容をまとめた．数値実験に見るように，構造的な疎性パターンを強く持つ SDP では効果が高い．今後の研究の方向性の一つとして，疎性パターンをあらかじめ解析することで，従来型の内点法と行列補完型内点法のどちらを実行するか自動選択できるようなメカニズムの検討がある．

なお，本稿で紹介したソフトウェア SDPA-C7 は，SDPA のサイト http://sdpa.sourceforge.net からフリーソフトとしてダウンロードして利用可能である．

謝辞 本研究は，公益財団法人日本科学協会の笹川科学研究助成によって実施したものです．

参考文献

[1] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, Algorithm 887: CHOLMOD: supernodal sparse Cholesky factorization and update/downdate, *ACM Transactions on Mathematical Software*, **35**(3), Article No. 22, 2009.

[2] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, Exploiting sparsity in semidefinite programming via

matrix completion I: general framework, *SIAM Journal on Optimization*, **11**(3), 647–674, 2000.
[3] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz, Positive definite completions of partial hermitian matrices, *Linear Algebra and its Applications*, **58**, 109–124, 1984.
[4] COPhy Junior Research Group, Spin glass server, http://www.informatik.uni-koeln.de/spinglass/.
[5] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results, *Mathematical Programming, Series B*, **95**, 303–327, 2003.
[6] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakta, and M. Nakata, Latest developments in the SDPA family for solving large-scale SDPs, In M. F. Anjos and J. B. Lasserre, editors, *Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*, chapter 24, 687–714. Springer, NY, USA, 2011.
[7] 藤澤克樹，福田光浩，中田和秀，中田真秀，脇隼人，山下真，特集　半正定値計画に対するソルバーと応用例，オペレーションズ・リサーチ，**55**(7), 386–424, 2010.

