

## 半正定値計画問題に対するクリロフ部分空間法の適用

中田 和秀 , 張 紹良

東京大学 工学系研究科 物理工学専攻

### 要旨

Lovász 数を半正定値計画問題として定式化し、それを内点法によって多項式時間内で解くことが可能である。この場合、内点法の各ステップにおいて、係数行列が密となる線形方程式系を解くことになる。この線形方程式系の大きさは Lovász 数を計算するグラフの枝の数と等しくなるため、グラフが大規模になるにつれ、線形方程式系も大規模となる。このとき、線形方程式系の解法として直接法を適用することは、計算時間やメモリ使用量の点で困難となる。このため、直接法を利用する代わりに、クリロフ部分空間法を利用することが試みられてきた。本発表では、これらの枠組みのもと、大規模なグラフに対する Lovász 数の計算を行う。このとき、より大きなグラフに対する Lovász 数を計算するため、この問題における特有の性質を生かしたいくつかの計算手法を提案する。最後に、実際に大規模なグラフに対する Lovász 数の計算を行うことにより、これらの手法の有用性について検証する。

## Incorporating Krylov-Subspace Methods for Semidefinite Programs

Kazuhide Nakata , Shao-Liang Zhang

Department of Applied Physics, Graduate School of Engineering, University of Tokyo

### Abstract

As a semidefinite program, the computation of the Lovász number can be solved by a interior-point method in a polynomial time. When one uses a interior-point method for the computation, one has to solve a linear system whose size is the number of edges of the graph and whose coefficient matrix is fully dense in each step of the interior-point method. Therefore, for a huge graph, it is difficult to apply direct methods to the dense linear systems which become huge. In the present paper, we exploit some Krylov-subspace methods for solving the huge and dense linear systems. In terms of the charcatic of the Lovász number's proplem, several special techniques are proposed for solving the linear systems rapidly. Finally, computational experiments are reported to confirm whether these methods work effectively for computing the Lovász number for huge graph.

## 1 はじめに

近年、半正定値計画問題 ( Semidefinite Programming, SDP と略 ) に対して、その解法である主双対内点法に関する理論的な解析、組合せ最適化問題への応用等の様々な研究が行われている [1, 7, 8, etc.]. また、著者等のグループが公開している SDPA (SemiDefinite Programming Algorithm,[2]) をはじめ、SDP を主双対内点法により解くソフトウェアが幾つか開発され、それ

をもとにした数値実験等の研究が盛んに行われている。本研究では、SDP を主双対内点法により解くソフトウェアにより、様々な大規模なグラフに対する Lovász 数の計算を行った。以下では、まず既存のそれらのソフトウェアで、大規模なグラフに対する Lovász 数を計算する場合生ずる問題点について述べる。次に、その問題点の本質的な解決法について説明を行う。さらに、Lovász 数を SDP 問題として計算する場合に現れる特有の性質を活かすことにより、計算時間を減少させることについてを述べる。

## 2 半正定値計画問題

半正定値計画問題の主問題と双対問題は、与えられた  $C \in S^n$ ,  $A_i \in S^n$  ( $1 \leq i \leq m$ ),  $b_i \in \mathbb{R}$  ( $1 \leq i \leq m$ ) により、以下のように定式化される。

$$\left. \begin{array}{ll} \text{主問題：} & \begin{array}{l} \text{最小化} \quad C \bullet X \\ \text{制約条件} \quad A_i \bullet X = b_i \quad (1 \leq i \leq m), \quad X \succeq O. \end{array} \\ \\ \text{双対問題：} & \begin{array}{l} \text{最大化} \quad \sum_{i=1}^m b_i y_i \\ \text{制約条件} \quad \sum_{i=1}^m A_i y_i + Z = C, \quad Z \succeq O. \end{array} \end{array} \right\} \quad (1)$$

ただし、 $S^n$  は  $n \times n$  の実対称行列の全体集合とする。任意の  $X, Z \in \mathbb{R}^{n \times n}$  に対して、 $X \bullet Z$  は  $X$  と  $Z$  の内積を意味し、 $\text{Tr } X^T Z$  ( $X^T Z$  のトレース) で定義される。 $X \succeq O$  は  $X \in S^n$  が半正定値、つまり任意の  $u \in \mathbb{R}^n$  に対し  $u^T X u \geq 0$  であることを示す。

## 3 主双対内点法

SDP は線形計画問題の対称行列問題への拡張として捉えることができ、線形計画問題の主双対内点法をそれに拡張した内点法で理論的に多項式時間で解くことが可能である ([5], [8] を参照)。この主双対内点法では、探索方向の計算過程においてある種の Newton 法を使用しており、このとき大きさ  $m$  の線形方程式系

$$Bx = s \quad (2)$$

を解く必要が生じる。ここで、線形方程式系 (2) の係数行列  $B$  の各要素は、制約行列  $A_i$  ( $1 \leq i \leq m$ ) と内点法の反復点  $X, Z$  を使い

$$B_{ij} = A_j \bullet X A_i Z^{-1} \quad (1 \leq i, j \leq m)$$

と定義される (本稿では、幾つか提案されている探索方向の中でも HRVW/KSH/M 方向に限定して述べる [7])。一般に  $X$  や  $Z^{-1}$  は密行列となるため、たとえ制約行列  $A_i$  ( $1 \leq i \leq m$ ) が疎の場合でも、係数行列  $B$  は密になるという性質がある。我々のグループが開発した SDPA[2] を含む SDP を主双対内点法により解く多くの既存のソフトウェアでは、この線形方程式系 (2) の解法として Cholesky 分解や修正 Cholesky 分解等の直接法を利用してきた。

## 4 Lovász 数

Lovász 数とはグラフに対し定義される正数であり、そのグラフの安定数の上界となり、さらにクリークカバー数の下界となる。グラフの安定数やクリークカバー数の計算問題が NP-hard であるのに対し、Lovász 数の計算は多項式時間内で可能であることが知られている。最近、この Lovász 数を半正定値計画問題として定式化し、それを内点法によって解く方法が提案されている。 $V = \{1, 2, \dots, n\}$  を頂点集合、 $E \subset \{(i, j) : i, j \in V, i < j\}$  を枝集合としたとき、 $G = (V, E)$  によって無向グラフを定義する。 $E^c$  を  $E$  の補集合  $\{(i, j) : (i, j) \notin E, i, j \in V, i < j\}$  とすると、このグラフの Lovász 数は次の SDP 問題の最適値と等しくなる ([4] など)。

$$\left. \begin{array}{l}
 \text{主問題： 最大化} \quad \mathbf{E}_{all} \bullet \mathbf{X} \\
 \text{制約条件} \quad \mathbf{E}_{ij} \bullet \mathbf{X} = 0 \ ((i, j) \in E^c), \\
 \quad \quad \quad \mathbf{I} \bullet \mathbf{X} = 1, \mathbf{X} \succeq \mathbf{O}. \\
 \\
 \text{双対問題： 最小化} \quad y_0 \\
 \text{制約条件} \quad y_0 \mathbf{I} + \sum_{(i,j) \in E^c} y_{ij} \mathbf{E}_{ij} - \mathbf{Z} = \mathbf{E}_{all}, \mathbf{Z} \succeq \mathbf{O}.
 \end{array} \right\} \quad (3)$$

ただし、行列  $\mathbf{X}, \mathbf{Z}, \mathbf{I}, \mathbf{E}_{all}, \mathbf{E}_{ij}$  は、すべて  $n \times n$  の対称行列であり、 $\mathbf{I}$  は単位行列、 $\mathbf{E}_{all}$  は各要素が全て 1 の行列、 $\mathbf{E}_{ij}$  は  $(i, j)$  と  $(j, i)$  要素が 1 で残りの要素が 0 の行列を表す。

$|E|$  を枝の数とすると、この SDP 問題 (3) の主問題の制約式の数  $m$  は  $n(n-1)/2 - |E| + 1$  となる。従って  $m$  は  $O(n^2)$  であるため、 $n$  に比べ  $m$  が極めて大きくなる可能性がある。このような SDP 問題を主双対内点法で解く場合、線形方程式系 (2) の解法として従来使われてきた直接法を利用すると、行列分解にかかる計算時間や行列  $B$  を保持するメモリ量が極めて大きくなり、最適解の計算が困難になることが予想される。

表 1 は、実際にいくつかの SDP 問題 (3) を SDPA[2] によって解いたときの、計算時間とメモリ使用量である。SDPA の終了条件として、双対残差 (最適値との差の上界) が 0.1 以下とした。

表 1: Lovász 数を SDPA によって計算したときの計算時間と使用メモリ

$n$	100	100	100
$m$	1024	2029	3992
$Bx = s$ の計算時間	71.1 秒	575.6 秒	5140.6 秒
その他の部分の計算時間	4.6 秒	4.9 秒	5.3 秒
総計算時間	75.7 秒	580.5 秒	5145.9 秒
係数行列 $B$ の確保	8.0MB	31.4MB	121.6MB
その他の部分の使用メモリ	7.1MB	9.9MB	20.4MB
総使用メモリ	15.1MB	41.3MB	142.0MB

表 1 より、計算時間の大部分は線形方程式系  $Bx = s$  の計算に費やされており、特に  $m$  が大きな場合顕著であることが確認できる。また、使用メモリの大部分も係数行列  $B$  の確保に費やされている。つまり、 $m$  が大きくなる SDP 問題 (3) は、計算時間やメモリ使用量の点で既存の方法では解くことが困難であるといえる。(より詳細な実験結果については [3] を参照)。

## 5 クリロフ部分空間法の適用

前節で述べた問題点を克服するため、線形方程式系 (2) に対し反復解法を適用することを試みる。その場合、係数行列を陽に保持せず計算を行うことが可能となる。また、少ない反復回数で線形方程式系 (2) の近似解  $\hat{x}$  を得ることが出来るならば、全体の計算時間を減らすことが期待できる。中田等 [6] では、既に線形方程式系 (2) に対し共役勾配法や共役残差法などのクリロフ部分空間法を適用したソフトウェア SDPA-KS(SemiDefinite Programming Algorithm-Krylov Subspace method) の開発を行っている。本研究では、以下で述べるような SDP 問題 (3) に特有の性質を利用することにより、計算効率を高めるための工夫を組み入れた。

- $E_{ij}((i, j) \in E^c)$  や  $I$  の疎性の利用

制約行列  $E_{ij}((i, j) \in E^c)$  は大きさ  $n \times n$  の対称行列であるが、非ゼロ要素はわずか 2 個しかない。また、制約行列  $I$  においても非ゼロ要素は  $n$  個である。これらの行列を疎形式で保持し、さらにその疎性を計算に利用することにより、大幅に計算時間やメモリ量を減らした。

- $E_{ij}((i, j) \in E^c)$  と  $I$  の直交性の利用

線形方程式系 (2) をクリロフ部分空間法により近似的に解く場合、主双対内点法でその誤差を補正する必要がある。中田等 [6] では、最小二乗問題に定式化することにより、その補正を行っている。SDP 問題 (3) では、制約行列  $E_{ij}((i, j) \in E^c)$  や  $I$  に対し、

$$\begin{aligned} E_{ij} \bullet E_{kl} &= 0 \quad ((i, j), (k, l) \in E^c, (i, j) \neq (k, l)) \\ I \bullet E_{ij} &= 0 \quad ((i, j) \in E^c) \end{aligned}$$

という直交性が成り立つため、実際上最小二乗問題の計算を省くことに成功した。

- 実行可能内点の導出

主双対内点法は実行可能内点  $(A_i \bullet X = b_i (1 \leq i \leq m), \sum_{i=1}^m A_i y_i + Z = C)$  を満たす正定値行列  $X, Z$  とベクトル  $y$  からスタートしたほうが、最適解に速く収束することが知られている。一般に、実行可能内点を見つけることは容易ではないが、SDP 問題 (3) では、以下のような  $X, y, Z$  が実行可能内点となる。

$$X = \frac{1}{n}I, y_0 = 10n, y_{ij} = 0 \quad ((i, j) \in E^c), Z = 10nI - E.$$

この実行可能内点を主双対内点法の初期点として利用することにより、最適解への収束性を向上させた。

表 2 は、表 1 と同じ問題を SDPA-KS によって解いたときの、計算時間と使用メモリである。クリロフ部分空間法として共役勾配法を使用しており、終了条件は表 1 と同様に実行可能解の双対残差が 0.1 以下とした。

表 1 と表 3 から、特に  $m$  が大きな SDP 問題 (3) に対し、総計算時間・使用メモリの両方において SDPA-KS は SDPA より優れていることが確認できる。

## 6 前処理

次に、主双対内点法の反復回数とクリロフ部分空間法の収束性について観察する。図 1 は、横軸に主双対内点法の各反復における双対残差、縦軸にその反復点に到達するまでの累積実行時間

表 2: Lovász 数を SDPA-KS によって計算したときの計算時間と使用メモリ

$n$	100	100	100
$m$	1024	2029	3992
$Bx = s$ の計算時間	19.7 秒	27.0 秒	47.4 秒
その他の部分の計算時間	4.5 秒	4.8 秒	5.7 秒
総計算時間	24.2 秒	31.8 秒	53.1 秒
総使用メモリ	5.5MB	6.1MB	7.4MB

をとったグラフである。SDPA では内点法の 1 反復で双対残差はほぼ一定の割合で減り計算時間もほぼ同じであるため、双対残差の対数と累積計算時間はほぼ比例している。しかし、SDPA-KS では内点法の 1 反復で双対残差はほぼ一定の割合で減るものの、反復回数は双対残差が小さくなるにつれて、各反復での計算時間が指数的に増大していることがわかる。結局 SDPA-KS は SDPA に比べ、双対残差がある程度大きな精度の悪い近似最適解を高速に求めることができる。一方、SDPA-KS で高精度の近似最適解を求めることは、計算時間の点から困難である。

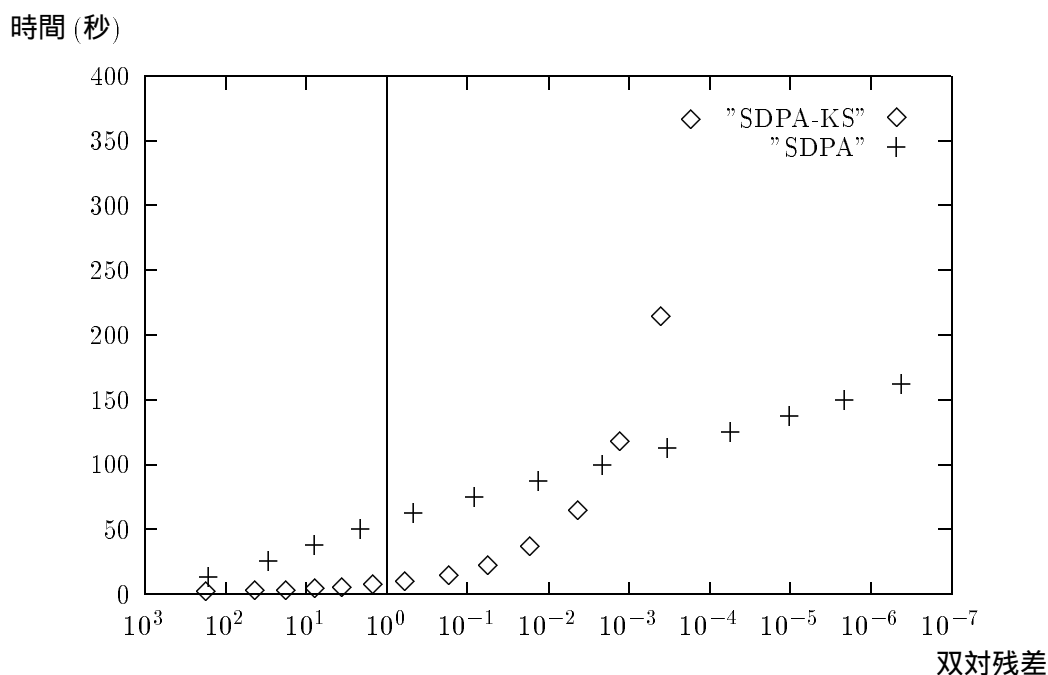


図 1: SDPA と SDPA-KS の双対残差 - 累積計算 グラフ

SDPA-KS ではより厳密な最適解を求めるためには、クリロフ部分空間法の収束性を高める必要がある。一般に反復解法の収束性は線形方程式系の係数行列の固有値に依存するため、以下で説明する前処理付きのクリロフ部分空間法が収束性を高めるのに有効である。これは、ある正則行列  $M_1, M_2$  を前処理行列とし、線形方程式系 (2) を解く代わりに

$$(M_1^{-1} B M_2^{-1})(M_2 x) = (M_1^{-1} s)$$

という線形方程式系を解くことに相当する。このとき行列  $M_1^{-1} B M_2^{-1}$  が単位行列に近似してお

れば、反復解法の収束が速くなる。ただし、前処理行列  $M_1, M_2$  はその行列を係数とする線形方程式系  $M_1 x = h, M_2 x = h$  の解が高速に計算出来るものでなくてはならない。本研究では、SDP 問題 (3) に対し、係数行列  $B$  の構造を解析することにより、この線形方程式系に対する特殊な前処理を適用する。

## 参考文献

- [1] Alizadeh, F., Haeberly, J.-P.A. and Overton, M.L.(1998) “Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results,” *SIAM Journal on Optimization* **8** 746–768.
- [2] Fujisawa, K., Kojima, M. and Nakata, K. (1996). “SDPA(Semidefinite Programming Algorithm) – User’s Manual –,” Technical Report B-308, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan. Available at <ftp://ftp.is.titech.ac.jp/pub/OpRes/software/SDPA>.
- [3] Fujisawa, K., Fukuda, M., Kojima, M. and Nakata, K. (1997) “Numerical Evaluation of SDPA(SemiDefinite Programming Algorithm),” Research Report B-330, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan.
- [4] Helmberg, C., Rendl, F., Vanderbei R.J. and Wolkowicz, H. (1996) “An interior-point method for semidefinite programming,” *SIAM Journal on Optimization* **6** 342–361.
- [5] 小島政和 (1996). “半正定値計画問題と内点法,” 応用数理, Vol 6, 16–25.
- [6] 中田和秀, 藤沢克樹, 小島政和 (1998). “半正定値計画問題に対する主双対内点法における共役勾配法の実装,” 統計数理 第 46 巻 第 2 号 297–316.
- [7] Todd, M. (1997). “On search directions in interior-point methods for semidefinite programming,” Technical Report No.1205, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853-3801.
- [8] Vandenberghe, L. and Boyd, S. (1996). “Semidefinite Programming,” *SIAM Review* **38** 49–95.