半正定値計画 (SDP) に対する内点法プログラムの数値実験

東京工業大学情報理工学研究科数理・計算科学専攻 藤沢 克樹

Katsuki Fujisawa
Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology

The semidefinite programming has various important applications to combinatorial optimization. This paper shows a primal-dual interior-point method for the semidefinite program and numerical experiments on several combinatorial optimization problems. We made a semidefinite programming solver, which we call SDPA. We used the SDPA for solving problems.

1. はじめに

Semidefinite Programming(SDP:半正定値計画) は、線形計画問題の拡張から生まれて、現在理論的には目覚ましい成果が挙げられている．その幾つかをここで紹介すると、線形計画問題の内点法の SDP への拡張 ([1, 15, 16, 21, 23, 27])、組合せ最適化問題への適用 ([1, 8, 9, 18]) などが挙げられる．またこれらの研究と連動しながら、ソフトウェアの開発と数値実験を行う研究も幾つか存在するが ([5, 6, 11, 26])、大規模かつ系統的な数値実験はこれから行われるものと思われる．そこで、本稿では SDP に対する内点法アルゴリズムに簡単に触れたあと、組合せ最適化問題に対する幾つかの適用例について触れる．

2. Semidefinite Programming(SDP)

筆者らが作成した SDP を解くための内点法アルゴリズム SDPA [5] は次の等式標準形の主問題とその双対問題を解く．

$$\text{SDP} \left\{ \begin{array}{lll} \text{主問題} & \min & \displaystyle\sum_{i=1}^{m} c_i x_i \\ & \text{s.t.} & \displaystyle X = \sum_{i=1}^{m} F_i x_i - F_0 \\ & & X \succeq O, \quad X \in \mathcal{S} \\ \text{双対問題} & \max & F_0 \bullet Y \\ & \text{s.t.} & F_i \bullet Y = c_i \ (i = 1, 2, \ldots, m) \\ & & Y \succeq O, \quad Y \in \mathcal{S} \end{array} \right.$$

行列 $F_i (i = 0, 1, \ldots, m)$ とベクトル c は、入力データとしてユーザーが入力する．また初期点 (x^0, X^0, Y^0) を指定することも出来るように設計されている．

$\mathcal{S}$ ： $n \times n$ 対称行列の空間
$F_i \in \mathcal{S}$ $(i = 0, 1, 2, \ldots, m)$ ： 定数行列,
$O \in \mathcal{S}$ ： ゼロ行列

$$c = \begin{pmatrix} c_1 \\ c_2 \\ \cdot \\ c_m \end{pmatrix} \in R^m \text{ ： 定数ベクトル,}$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_m \end{pmatrix} \in R^m \text{ ： 変数ベクトル,}$$

$X \in \mathcal{S},\ Y \in \mathcal{S}$ ： 変数行列,
$U \bullet V$ ： $U,\ V \in \mathcal{S}$ の内積 $\sum_{i=1}^{n}\sum_{j=1}^{n} U_{ij}V_{ij}$
$U \succeq O, U \in \mathcal{S} \iff U \in \mathcal{S}$ が半正定値
$U \succ O, U \in \mathcal{S} \iff U \in \mathcal{S}$ が正定値

(x, X) が主問題の実行可能解（または，最小解）であり，Y が双対問題の実行可能解（または，最大解）であるとき，(x, X, Y) は SDP の実行可能解（または，最適解）であると呼ぶ．以下では，SDP の主問題と双対問題を一まとめにして扱うことが多い．(x, X, Y) が SDP のすべての制約条件を満たすとき，許容解であるという．さらに，許容解 (x, X, Y) が条件 $X \succ O,\ Y \succ O$ を満たすとき，内点許容解と呼ぶ．また，(x, X) が 主問題の最小解で，Y が双対問題の最大解であるとき，(x, X, Y) は SDP の最適解であるという．(x, X, Y) を SDP の許容解とすると，主問題の目的関数値 $\sum_{i=1}^{m} c_i x_i$ と双対問題の目的関数値 $F_0 \bullet Y$ の間に，不等式

$$X \bullet Y = \sum_{i=1}^{m} c_i x_i - F_0 \bullet Y \geq 0 \tag{1}$$

が常に成り立つ．したがって，それらの目的関数値が一致すれば, あるいは，$X \bullet Y = 0$ （相補性条件）が成り立てば，(x, X, Y) は SDP の最適解であることが分かる．以下の双対定理はこの逆を保証している．

定理 2..1. [22] SDP に内点許容解が存在すると仮定する（Slater の制約想定 [19]）．このとき，

1. SDP に最適解が存在する．
2. SDP の許容解 (x^*, X^*, Y^*) が最適解であるための必要十分条件は主問題と双対問題の目的関数値が一致することである．すなわち，

$$X^* \bullet Y^* = \sum_{i=1}^{m} c_i x_i^* - F_0 \bullet Y^* = 0$$

この双対定理は SDP の理論の中核をなしている．

3. 主・双対内点法の SDP への拡張

主・双対内点法（[14, 20, 25]）は，Karmarkar 法 [13] 以後に発展した内点法のなかで，理論的にも実用的にも最も優れた方法であるといえる．

非常に多くの研究がなされており，超大規模な線形計画問題を高速に解く計算手法として定着している．日本語の文献としては [28] 等がある．文献 [16] で主・双対内点法の基本構造を SDP に拡張している．

この方法で主要な役割を果たしているのは内点許容領域の内部を通って最適解に収束する “中心パス（center path, central trajectory）” である．

SDP に対しては，

$$F_{cen} = \{(x, X, Y) \in F_{++} \ : \ XY = \mu I \ \ \exists \mu > 0\}$$

で定義される．ここで，

$$F_{++} = \left\{ (x, X, Y) : \begin{array}{l} X = \sum_{i=1}^{m} F_i x_i - F_0, X \succ O, \\ F_i \bullet Y = c_i \ (i = 1, 2, \ldots, m), Y \succ O, \end{array} \right\}$$

は SDP の内点許容領域を表す．

以下に中心パスの性質を示す．

- $(x, X, Y) \in F_{cen}$ であれば，$XY = \mu I$ なるパラメータ $\mu > 0$ は $\mu = X \bullet Y/n$ で与えられる．
- $F_{++} \neq \emptyset$ のもとで，中心パス F_{cen} は内点許容領域内の滑らかな曲線となり，$\mu \to 0$ のとき SDP の最適解に収束する．

主・双対内点法ではパラメータ $\mu > 0$ を減少させながら，中心パス F_{cen} を数値的に追跡し，SDP の近似最適解に到達する．

次に、SDP に対する主・双対内点法アルゴリズムの概要を示す．

手順０． まず、条件 $X^0 \succ O$, $Y^0 \succ O$ を満たす初期点 (x^0, X^0, Y^0) を選ぶ (許容解でなくてもよいことに注意)．

手順１． 方向探索パラメータ $\beta \in [0,\ 1]$ を選んで（例えば $\beta = 0.1$），$\mu = \beta X^k \bullet Y^k/n$ とおく．

手順２． パラメータ $\mu > 0$ をもつ中心パス上の点

$$(x^k + w, X^k + U, Y^k + V) \in F_{cen}$$

を近似するためのニュートン方程式をたてる．

$$\left. \begin{array}{l} (X^k + U) = \sum_{i=1}^{m} F_i (x_i^k + w_i) - F_0, \\ F_i \bullet (Y^k + V) = c_i \ (i = 1, 2, \ldots, m), \\ X^k Y^k + X^k V + U Y^k = \mu I, \\ U \in \mathcal{S},\ V \in \mathcal{S},\ w \in R^m. \end{array} \right\} \tag{2}$$

手順３． ニュートン方程式を解き，探索方向ベクトル (w, U, V) を計算する．(手順１で探索方向パラメータ $\beta \in [0,\ 1]$ を 1 に近く取るほど，探索方向ベクトル (w, U, V) は中心を向き，0 に近く取るほど SDP の解の方向を向く．)

手順４． 新しい反復点 $(x^{k+1}, X^{k+1}, Y^{k+1})$ を条件

$$\begin{aligned} x^{k+1} = x^k + \alpha_p w,\ X^{k+1} &= X^k + \alpha_p U \succ O, \\ Y^{k+1} &= Y^k + \alpha_d V \succ O, \end{aligned}$$

を満たすように定める．ただし，$\alpha_p, \alpha_d > 0$ はステップ長．(ステップ長をあまり大きく取ると新しい反復点が境界に近づきすぎて，以後の反復で困難を生ずる．また，小さすぎると収束までに多くの反復回数を要する．)

方向探索パラメータ β とステップ長 α_p, α_d を適当に選ぶことにより，大域的な収束性を保証することができる．また，ポテンシャル関数を評価関数として組み合わせて使うことも可能である．

4. SDPA における入力行列のデータ構造

SDPA ではブロック対角なデータ行列の入力およびそれらの内部演算を組み入れている．ブロック数とブロック構造ベクトルを用いて，定数行列 $F_0,\ F_1,\ \ldots,\ F_m$ に共通なブロックデータ構造を表わす．一般に，

$$\left.\begin{array}{rcl} F &=& \begin{pmatrix} B_1 & O & O & \ldots & O \\ O & B_2 & O & \ldots & O \\ \cdot & \cdot & \cdot & \ldots & O \\ O & O & O & \ldots & B_\ell \end{pmatrix}, \\ B_i &:& p_i \times p_i \text{ 対称行列 } (i = 1, 2, \ldots, \ell) \end{array}\right\} \tag{3}$$

とすると，この対称行列のブロック数 nBLOCK およびブロック構造ベクトル bLOCKsTRUCT は以下のように定義される．

$$\begin{aligned} \text{nBLOCK} &= \ell, \\ \text{bLOCKsTRUCT} &= (\beta_1,\ \beta_2,\ \ldots,\ \beta_\ell), \\ \beta_i &= \begin{cases} p_i & B_i \text{ が通常の対称行列のとき} \\ -p_i & B_i \text{ が対角行列のとき} \end{cases} \end{aligned}$$

例えば，

$$\begin{pmatrix} 1 & 2 & 3 & 0 & 0 & 0 & 0 \\ 2 & 4 & 5 & 0 & 0 & 0 & 0 \\ 3 & 5 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \tag{4}$$

では

$$\begin{aligned} \text{nBLOCK} &= 3, \\ \text{bLOCKsTRUCT} &= (3,\ 2,\ -2) \end{aligned}$$

となる．なお、入力行列 B_i は、密行列 (通常の 2 次元配列) と疎行列の 2 種類に対応することができる．

5. 組合せ最適化問題への応用について

多くの組合せ最適化問題は 0-1 整数計画問題に定式化できる．

$$\begin{aligned} \text{IP} \quad & \text{目的} \quad a_0^T x \ \rightarrow \ \text{最小化} \\ & \text{条件} \quad x \in P = \left\{ x : \begin{array}{l} Ax \leq b, \\ x_j = 0 \text{ or } 1 \ (j = 1, 2, \ldots, n) \end{array} \right\} \end{aligned}$$

この問題を解くために，許容領域 P を緩和した線形計画問題

$$\begin{aligned} \text{LP による緩和問題} \quad & \text{目的} \quad a_0^T x \ \rightarrow \ \text{最小化} \\ & \text{条件} \quad x \in \hat{P} = \left\{ x : \begin{array}{l} Ax \leq b, \\ 0 \leq x_j \leq 1 \ (j = 1, 2, \ldots, n) \end{array} \right\} \end{aligned}$$

を考える場合が多い．明らかに，$P \subset \hat{P}$ であり，LP による緩和問題の最小値は IP の最小値の下界を与える．P と $\hat{P}$ が近似的に等しい場合には，緩和問題の最小解が IP の近似最小解になる．一般には，$\hat{P}$ と P は隔たりが大きいので，$\hat{P}$ を削り取って P に近づけるための切除平面を入れた線形計画問題を作る必要がある．この種の方法は切除平面法と呼ばれている．

SDP を利用すると $P \subset \tilde{P} \subset \hat{P}$ を満たし，P をよりよく近似する凸集合 $\tilde{P}$ を構成できる．

$$\begin{aligned} \text{SDP による緩和問題} \quad & \text{目的} \quad a_0^T x \ \rightarrow \ \text{最小化} \\ & \text{条件} \quad x \in \tilde{P}. \end{aligned}$$

$\tilde{P}$ はもはや多面体ではないので，単体法を適用することはできないが，内点法によって解くことができる．最大安定集合問題 (最大クリーク問題) 等のグラフ上の組合せ最適化問題に対して理論的な研究が盛んに行われている．詳しくは [1, 8, 9, 18] 参照．今回は、その中から最大カット問題、グラフ分割問題および最大クリーク問題を選択し、解くべき元問題の大きさと SDP に定式化したときのメモリの消費量との関係、また実行時間などを調べる．

5.1. 問題の定義

単純無向グラフ $G = (V, E)$ が与えられたときに、各枝を $(i, j) \in E$、枝に対する非負の重みを c_{ij} で表すこととする．$n = |V|$ は、グラフ分割問題の場合、偶数であると仮定し、このとき点集合 V の分割とは、$L \cap R = \emptyset$, $L \cup R = V$ となるような点集合の対 (L, R) である．

最大カット問題とは、$c(L, R) = \sum_{i \in L, j \in R} c_{ij}$ を最大にするような、対 (L, R) を見つける問題である．
グラフ分割問題とは、$|L| = |R| = n/2$ で $c(L, R) = \sum_{i \in L, j \in R} c_{ij}$ を最小にするような、対 (L, R) を見つける問題である．
最大クリーク問題とは、同様にグラフが与えられたときに位数が最大になるような完全部分グラフを見つける問題である．

5.2. 最大カット問題

最大カット問題に対する定式化は Goemans and Williamson [7] が良く知られている．$x = (x_i) \in \Re^n$ を vertex-incidence ベクトルとして次のように定義する．

$$x_i = \begin{cases} 1 & i \in L \\ -1 & i \in R. \end{cases}$$

このとき、最大カット問題は次のような二次の $0-1$ 整数計画として定式化することができる．

$$\begin{array}{ll} \max & \frac{1}{2}\sum_{i<j} c_{ij}(1 - x_i x_j) \\ \text{s.t.} & x_i \in \{-1, 1\} \qquad \text{for } i. \end{array}$$

ここで、$X = xx^T$ つまり $X_{ij} = x_i x_j$ と置き換えることにより 次の SDP 緩和問題に定式化することができる．$X_{ii} = x_i x_i = x_i^2 = 1$ であり、また $X_{ij} = x_i x_j$ であるので、$X \succeq O$ でかつ $\mathrm{rank}(X) = 1$ でなければならない．つまり、下の定式化は $\mathrm{rank}(X) = 1$ という条件を緩和している．

$$\begin{array}{ll} \max & \frac{1}{2}\sum_{i<j} c_{ij}(1 - X_{ij}) \\ \text{s.t.} & X \succeq O \\ & X_{ii} = 1 \qquad \text{for } i. \end{array}$$

今回の数値実験では、Johnson et al. [12] がグラフ分割問題に用いたグラフを中心に用いて、最大カット問題に対して数値実験を行う．なおソフトウェアは C++ を用いて作成された SDPA [5] を用いる．なお計算機は SONY の NEWS-5000WI (搭載メモリ 128Mbyte, 135MIPS, 71SPEC-int92, 77SPEC-fp92) を用いて行う．

Table 1: 最大カット問題における SDP 緩和の実験結果

グラフ	実行時間 (sec.)	メモリ (Mbyte)	上界	下界
g10.05	0.2	0.3	18.3	18
g20.10	1.4	0.5	69.3	68
g40.20	18.6	1.1	251.2	248
g50.25	43.5	1.4	378.8	371
g124.02	1711.7	5.1	138.9	137
g124.05	472.9	5.1	263.2	256
g124.10	495.9	5.1	456.8	446
g124.20	499.9	5.1	837.0	834
g250.05	6373.3	23	531.2	502
g500.20	133892.5	81	3556.5	3372

g10.05 とは、点数が 10 で、点の平均次数が 5 であることを示している．下界は組合せ最適化問題に対する代表的な近似解法の一つである tabu search [4] で求めている．最大カット問題の場合は、$n = m = |V|$ であり、n が増加していくにつれて実行時間もメモリの消費量も急激に増加していくのが見て取れる．

5.3. グラフ分割問題

グラフ分割問題も最大カット問題と類似した問題なので同様に定式化することができる [24]. 最大カット問題と同様に、$x = (x_i) \in \Re^n$ を vertex-incidence ベクトルとして次のように定義する。

$$x_i = \begin{cases} 1 & i \in L \\ -1 & i \in R. \end{cases} \tag{5}$$

このとき、グラフ分割問題も次のような二次の $0-1$ 整数計画問題として定式化することができる.

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i<j} c_{ij}(1 - x_i x_j) \\ \text{s.t.} & \sum_i x_i = 0 \\ & x_i \in \{-1, 1\} \qquad \text{for } i. \end{array}$$

ここでも最大カット問題と同様に、 $X = xx^T$ つまり $X_{ij} = x_i x_j$ と置き換えることにより SDP 緩和問題に定式化することができる。

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i<j} c_{ij}(1 - X_{ij}) \\ \text{s.t.} & X \succeq 0 \\ & J \bullet X = 0 \\ & X_{ii} = 1 \qquad \text{for } i. \end{array}$$

$J \in \mathcal{S}$ は全ての要素が 1 の行列であり, $J \bullet X = 0$ は行列 J と X の内積が 0 であることを示している.

上界は最大カット問題と同様に tabu search [4] で求めた近似解である. この場合 $n = |V|, m = |V| + 1$ であるが、行列 J が要素が全て 1 の行列であるために、最大カット問題と比較してメモリの消費量がやや多くなっている. なお、最大カット問題とグラフ分割問題と共に、グラフの枝数 ($|E|$) が、直接 n, m と関係しないために (F_0 のみに関係する)、あまり実行時間等に影響を及ぼしていない. なお [6] では、制約式に疎行列が多いなどの問題の構造を活用し、さらに高速に問題を解くことに成功している.

5.4. 最大クリーク問題

最大クリーク問題に対しては、次の SDP 緩和問題 [8] がよく知られている.

$$\begin{array}{ll} \max & J \bullet X \\ \text{s.t.} & H_{ij} \bullet X = 0 \quad \forall (ij) \notin E \\ & X \bullet I = 1 \\ & X \succeq O. \end{array}$$

行列 H_{ij} は、(ij) 及び (ji) 要素のみが 1 であとの要素は全て 0 の行列である.

Table 2: グラフ分割問題における SDP 緩和の実験結果

グラフ	実行時間 (sec.)	メモリ (Mbyte)	下界	上界
g10.05	0.2	0.3	8.9	9
g20.10	1.7	0.5	40.2	42
g40.20	21.7	1.3	149.4	155
g50.25	45.6	2.0	238.6	249
g124.02	732.5	6.4	6.8	13
g124.05	725.7	6.4	44.3	63
g124.10	706.2	6.4	147.2	178
g124.20	706.0	6.4	401.4	449
g250.02	8564.0	24.8	15.4	29
g250.05	8630.8	24.8	81.9	114
g250.10	7755.9	24.8	303.5	357
g250.20	8426.6	24.8	747.3	828
g500.02	121165.2	82	25.3	49
g500.05	94071.8	82	156.1	218
g500.10	79922.8	82	513.0	626
g500.20	74016.0	82	1567.0	1744

Table 3: 最大クリーク問題における SDP 緩和の実験結果

グラフ	実行時間 (sec.)	メモリ (Mbyte)	上界	最適解
g10.05	0.2	0.4	3.0	3
g20.10	5.2	1.0	5.0	5
g30.15	36.3	2.1	7.0	7
g50.45	302.5	2.6	21.2	21
g60.54	758.5	3.1	22.4	21
g70.63	628.8	7.0	25.3	24
johnson8-2-4	118.9	1.6	4.0	4
johnson8-4-4	10009.6	11.1	14.0	14
C125.9	9767.4	19.5	37.8	34

ランダムに作成したグラフおよび DIMACS (**ftp://dimacs.rutgers.edu**) のベンチマーク問題からグラフを使用して、数値実験を行う。

下の 3 つの問題は、DIMACS のベンチマーク問題である．なお最適解は、同じく DIMACS から得られる最大クリーク問題に対する陰的列挙法のプログラム (dfmax.c) を用いて算出した．この場合は、$n = |V|, m = \frac{n(n-1)}{2} - |E| + 1$ である．つまり、グラフの点数が同じならば枝数が多いほうが、制約式の数 m が少なく解きやすくなる．

6. 結論と今後の課題

本稿では、SDP の数値実験結果を幾つか紹介して、問題の規模と実行時間やメモリの消費量との関係を示した．最近では、SDP にも predictor-corrector 法 [17] が取り入れられたり、また探索方

向 [2, 11, 16, 21, 23] も複数提案されるなど、アルゴリズムも発展を続けており、SDPA などのソフトウェアにも随時反映されている．そのため、大規模かつ系統的な数値実験はこれから行われていくものと推測される．本稿では、最新の実験結果には触れられなかったが、例えば [6] では、疎行列などの問題の構造を有効に活用して大幅な高速化を達成している．また SDP には、組合せ最適化だけではなく、システムと制御への応用 [3, 10, 27] も知られており、この分野においても SDP の適用と数値実験が望まれる．

References

[1] W. F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM J. on Optimization* to appear.

[2] W. F. Alizadeh, J.-P.A. Haeberly and M.L. Overton, "Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results," Report 721, Computer Science Dept. New York University, New York, May 1996.

[3] S. Boyd, L. E. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory,* (SIAM, Philadelphia, 1994).

[4] 藤沢 克樹, 久保 幹雄, 森戸 晋. Tabu search のグラフ分割への適用と実験的解析. 電気学会論文誌, 114-C(4):430–437, 1994.

[5] K. Fujisawa, M. Kojima and K. Nakata, "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, December 1995, Revised August 1996. Available via anonymous ftp at ftp.is.titech.ac.jp in pub/OpRes/software/SDPA.

[6] K. Fujisawa, M. Kojima and K. Nakata, "Exploiting Sparsity in Primal-Dual Interior-Point Methods for Semidefinite Programming," Technical Report B-324, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan, January 1997, Revised April 1997. Available via anonymous ftp at ftp.is.titech.ac.jp in pub/OpRes/articles.

[7] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. 1994. unpublished manuscript.

[8] M, Grötschel, Lovász and A. Schrijver, "Polynomial algorithms for perfect graphs," *Annals of Discrete Mathematics* 21 (1984) 325-356.

[9] M, Grötschel, Lovász and A. Schrijver, *Geometric algorithms and combinatorial optimization* (Springer, New York, 1988).

[10] 原辰次, "ロバスト制御理論の動向 – 実用的な制御計設計法を目指して –", Proceedings of the 38th Annual Conference of the ISCIE (1994) 25–27.

[11] C. Helmberg, F. Rendl, R.J. Vanderbei and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM Journal on Optimization* **6** (1996) 342–361.

[12] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation, part I, graph partitioning. *Operations Research*, 37:865–892, 1989.

[13] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica* **4** (1984) 373–395.

[14] M. Kojima, S. Mizuno and A. Yoshise, "A primal-dual interior point algorithm for linear programming," In N. Megiddo, ed., *Progress in Mathematical Programming, Interior-Point and Related Methods* (Springer-Verlag, New York, 1989) 29–47.

[15] M. Kojima, S. Kojima and S. Hara, "Linear algebra for semidefinite programming," Research Reports B-290, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, October 1994.

[16] M. Kojima, S. Shindoh and S. Hara, "Interior-point methods for the monotone linear complementarity problems in symmetric matrices," Research Reports B-282, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, April 1994.

[17] M. Kojima, M. Shida and S. Shindoh, "Local Convergence of Predictor-Corrector Infeasible-Interior-Point Algorithms for SDPs and SDLCPs" Research Report #306, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan, December 1995, Revised October 1996.

[18] L. Lovász and A. Schrijver, "Cones of matrices and set functions and 0-1 optimization," *SIAM J. on Optimization* 1 (1991) 166-190.

[19] O. L. Mangasarian, *Nonlinear Programming* (McGraw-Hill, New York, 1969).

[20] N. Megiddo, "Pathways to the optimal set in linear programming," In N. Megiddo, ed., *Progress in Mathematical Programming, Interior-Point and Related Methods* (Springer-Verlag, New York, 1989) 131–158.

[21] R.D.C. Monteiro, "Primal-Dual Path Following Algorithms for Semidefinite Programming," to appear in *SIAM Journal on Optimization*.

[22] Y. E. Nesterov and A. S.Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming* (SIAM, Philadelphia, 1993).

[23] Y. E. Nesterov and M. J. Todd, "Self-scaled cones and interior-point methods in nonlinear programming," Catholic University of Lonvain, Lonvain-la-Neuve, Belgium, 1994.

[24] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph-bisection problems. 1993. unpublished manuscript.

[25] K. Tanabe, "Centered Newton method for mathematical programming," In M. Iri and K. Yajima, eds., *System Modeling and Optimization* (Springer-Verlag, New York, 1988) 197-206.

[26] K.C. Toh, M.J. Todd and R.H. Tütüncü, "SDPT3 – a MATLAB software package for semidefinite programming," Dept. of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore, December 1996. Available at http://www.math.nus.sg/ mattohkc/index.html.

[27] L. Vandenberghe and S. Boyd, "A primal-dual potential reduction method for problems involving matrix inequalities," *Mathematical Programming, Series B* to appear.

[28] 吉瀬章子, "凸計画問題に対する最適化手法 – 内点法と解析中心", システム／制御／情報, Vol.3, No.3. 155–160, 1994.