

双対問題に対する内点法

山下 浩

1. はじめに

線形計画問題に対する Karmarkar の画期的な解法の発見以後, 多くの類似のアルゴリズムが提案されている。それらは大別すると次のように分類できる。

- (i) プロジェクティブ・スケーリングを利用したもの [7, 4, 13]
- (ii) アフィン・スケーリングを利用したもの [1, 2, 3, 5, 9]
- (iii) センター・トラジェクトリの追跡によるもの [8]
- (iv) Newton 法 [6, 10, 11, 12]

これらの方法は, それぞれ異なったアイデアにより実現され, 固有の長所・短所をもっている。筆者の個人的な意見としては, ポテンシャル関数を最小化するためにプロジェクティブあるいはアフィン・スケーリングを利用した方法が色々な意味で将来性があると思われる。

本稿では筆者の提案した (i) に属する方法の概略といつかの数値実験について論ずる。詳しくは [13, 14] を参照されたい。又本稿の数値実験の結果は数理システムの荒川貴道氏との共同研究の結果であることを付記しておく。

2. 問題の設定と射影空間の導入

線形計画問題:

$$(D) \quad \begin{array}{l} \text{最小化} \quad c^t x, \quad x \in R^n \\ \text{条件} \quad A^t x \geq b \end{array}$$

を考える。ここで $c \in R^n$, $A = [a_1, \dots, a_m] \in R^{n \times m}$, $b \in R^m$ である。よく知られているように問題 (D) は標準形の問題:

$$(P) \quad \begin{array}{l} \text{最大化} \quad b^t y, \quad y \in R^m \\ \text{条件} \quad A y = c, \quad y \geq 0 \end{array}$$

の双対問題である。それぞれの問題の許容領域を Π , Ω と表わす。

$$(2.1) \quad \Pi = \{x \in R^n \mid A^t x \geq b\}$$

やました ひろし (株)数理システム

〒160 新宿区新宿 2-5-3

$$(2.2) \quad \Omega = \{y \in R^m \mid A y = c, \quad y \geq 0\}$$

ここでは問題 (D) を解くための内点法を考える。そこで以下の仮定をおく。

- (i) Π の内部は空でない。
- (ii) $\text{rank } A = n$
- (iii) c_0^* を最適値とすると c_0^* の下界値 $c_0 (\leq c_0^*)$ が知られている。
- (iv) 解集合 $\Pi^* = \{x \in \Pi \mid c^t x = c_0^*\}$ は有界である。

最適解を得るために, 伊理・今井 [6] によって導入されたポテンシャル関数

$$(2.3) \quad F(x; c_0) = \frac{(c^t x - c_0)^{m+1}}{\prod_{i=1}^m (a_i^t x - b_i)}$$

の最小化を考える。 $F(x; c_0) (> 0)$ の十分小さな値を与える (x, c_0) を得れば x と c_0 は $x^* \in \Pi^*$ と c_0^* の十分良い近似値であると考えることができる。伊理・今井は F の最小化の方法として Newton 法を提案したが, ここでは射影空間を導入して多項式オーダーの解法を得ることが可能であることを示す。そのために $n+1$ 次元の空間を導入して元の空間 R^n をその中に埋め込む。いま $z = (z_0, z_1, \dots, z_n)^t \in R^{n+1}$ として $x_i = z_i/z_0$, $i=1, \dots, n$ とおく。そしてポテンシャル関数 \tilde{F} を

$$(2.4) \quad \tilde{F}(z; c_0) = \frac{(\tilde{c}^t z)^{m+1}}{\prod_{i=0}^m \tilde{a}_i^t z}$$

と定義する。ここで

$$\tilde{c} = \begin{bmatrix} -c_0 \\ c \end{bmatrix}, \quad \tilde{a}_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \tilde{a}_i = \begin{bmatrix} -b_i \\ a_i \end{bmatrix}, \quad i=1, \dots, m$$

である。また \tilde{A} を

$$\tilde{A} = [\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_m] = \begin{bmatrix} 1 & & -b^t \\ & & A \\ 0 & & \end{bmatrix}$$

によって定義する。明らかに $z_0 \neq 0$ ならば

$$F(x; c_0) = \tilde{F}(z; c_0)$$

であるから

$$(2.5) \quad \tilde{\Pi} = \{z \in R^{n+1} \mid \tilde{A}^t z \geq 0\}$$

の内点を通して $\tilde{F}(z; c_0) \rightarrow 0$ となれば対応する $(x_1/z_0,$

$\dots, z_n/z_0) \rightarrow \Pi^*$ となる。したがって x の空間で F を最小化しても z の空間で \bar{F} を最小化しても同じことになる。ただし 1 つの重要な相異点がある。いま $x^* \in \Pi^*$ が最適解とすると $z = (z_0, z_0 x^*)^t$ は任意の z_0 ($\neq 0$) に対して対応する“最適解”を与える。すなわち原点と $(1, x^*)$ を結ぶ直線上のどの点も“最適解”である。

\bar{F} を 0 に近づけるために次のようにする。すなわち、

\bar{F} の分子を一定に保ちつつ分母の $\prod_{i=0}^m \bar{a}_i^t z$ を最大化する。

そこで

$$h(z) = \sum_{i=0}^m \ln(\bar{a}_i^t z)$$

とおいて問題：

$$(2.6) \text{ 最大化 } h(z), z \in \text{int } \bar{\Pi}$$

$$\text{条件 } \bar{c}^t z = \text{一定} (> 0)$$

を考える。簡単な計算によって

$$\nabla h = \bar{A} \bar{D}^{-1} e$$

$$\nabla^2 h = -\bar{A} \bar{D}^{-2} \bar{A}^t \equiv -\bar{B}$$

となることがわかる。ここで

$$\bar{D} = \text{diag}\{\bar{a}_0^t z, \dots, \bar{a}_m^t z\}, e = (1, \dots, 1)^t$$

である。したがって問題 (2.6) に Newton 法を適用するとその探索方向ベクトル $s \in R^{n+1}$ は

$$(2.7) \text{ 最大化 } -\frac{1}{2} s^t \bar{B} s + s^t \bar{A} \bar{D}^{-1} e$$

$$\text{条件 } \bar{c}^t s = 0$$

から得られる。(2.7) の解は

$$(2.8) s = \bar{B}^{-1} \bar{A} \bar{D}^{-1} e - \zeta \bar{B}^{-1} c = z - \zeta \bar{B}^{-1} c$$

$$(2.9) \zeta = \frac{\bar{c}^t \bar{B}^{-1} \bar{A} \bar{D}^{-1} e}{\bar{c}^t \bar{B}^{-1} c} = \frac{\bar{c}^t z}{\bar{c}^t \bar{B}^{-1} c}$$

となる。次に \bar{A} の一般逆行列 \bar{A}^- を

$$\bar{A}^- = \bar{D}^{-2} \bar{A}^t \bar{B}^{-1}$$

によって定義し、 $w \in R^{m+1}$ を

$$w = \bar{A}^- \bar{c}$$

とおく。明らかに

$$(2.10) \bar{A} w = \bar{c}$$

である。 w の成分を $w = (w_0, y)^t$ と書くと \bar{A}, \bar{c} の定義により (2.10) は

$$w_0 = b^t y - c_0, Ay = c$$

となる。したがって $y \geq 0$ ならば y は主問題の許容解である。

3. 多項式オーダーの解法

(2.8) によって与えられる探索方向 s に沿ってステップ幅 $\alpha > 0$ だけ進み

$$(3.1) z_+ = z + \alpha s$$

とおいて次の点 $z_+ \in \text{int } \bar{\Pi}$ を得る。そこで適当なステップ幅 α を選ぶと多項式オーダーの解法が成立することを示す。そのために

$$\Delta = h(z_+) - h(z) = \sum_{i=0}^m \ln(1 + \alpha p_i)$$

$$p = \bar{D}^{-1} \bar{A}^t s$$

とおくと (2.8) より

$$p = e - \zeta \bar{D} w$$

と表わすことができ p は

$$\|p\|^2 = p^t e$$

を満たす。次の 3 つの補題によって多項式オーダーの解法であることを示すことができる。

補題 1 $z \in \text{int } \bar{\Pi}$ で $w \succ 0$ ならば $p \prec e$ である。

補題 2 $w \succ 0$ ならば $c_0 < c_0^*$ である。

補題 3 $\bar{p} = \max_i |p_i| \geq 1$ で $\alpha = 1/(1 + \bar{p})$ ならば $z_+ \in \text{int } \bar{\Pi}$ で $\Delta \geq \ln(e/2)$ となる。

いま $w \succ 0$ ならば c_0 を $w \geq 0$ を満たすように増加させることができ、このとき $p \prec e$ となるから $z_+ = z + \alpha s$ とおくと

$$\bar{F}(z_+; c_{0+}) \leq (\bar{e}^2) \bar{F}(z; c_0)$$

を得る。

4. アルゴリズムの構造

前節で述べた射影空間上でのアルゴリズムを元の空間 R^n にもどすとどのような方法が得られるであろうか。

いま

$$d_1 = -B^{-1} c, d_2 = B^{-1} A D^{-1} e$$

とおくと $(B = A D^{-2} A^t, D = \text{diag}\{a_i^t x - b_i\})$

s に対応する探索方向ベクトル d_p は

$$d_p = d_1 + \zeta_2 d_2$$

$$\zeta_2 = \gamma (c^t \hat{x} - c_0)$$

$$\hat{x} = A^{-t} b, A^- = D^{-2} A^t B^{-1}$$

$$\gamma = 1 / \{1 + \|D^{-1}(A^t \hat{x} - b)\|^2\}$$

と表わすことができる。すなわち探索方向は 2 つのベクトル d_1 と d_2 の 1 次結合によって表わすことができる。

d_1 は良く知られているようにアフィン・スケーリングの方向である。 d_2 は色々と興味深い性質をもっている。いま

$$h(x) = \sum_i \ln(a_i^t x - b_i)$$

とおくと h の最大値を与える点を解析的中心と呼ぶが、 d_2 は h を最大化するための Newton 法のベクトルとなっている。また $-d_2$ は

$$\hat{x} = \text{argmin} \|D^{-1}(A^t x - b)\|$$

とおくと

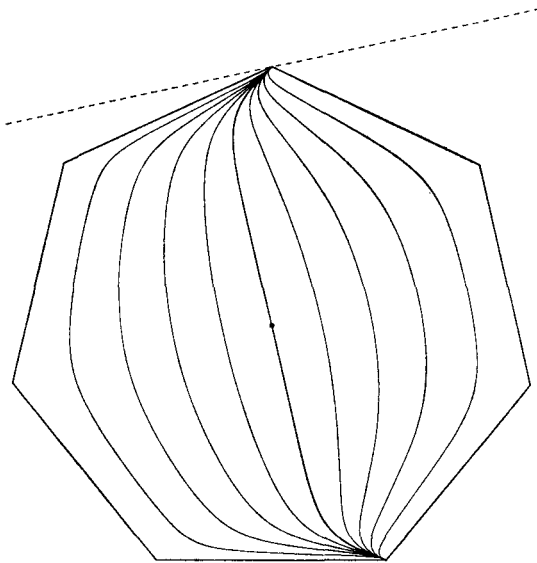


図 1 $\hat{x}=d_1$ の解のトラジェクトリ

$$\hat{x} = x - d_2$$

となるので上の重みつき線形最小乗問題の解を得るための補正ベクトルとなっている。

また興味深いことに伊理・今井のNewton法のベクトル d_N も

$$d_N = \theta_1 d_1 + d_2$$

と書くことができる。

次に c_0 の更新はどのようになるであろうか。

$$u = \gamma D^{-2} (A^t \hat{x} - b)$$

とおいて

$$y = -c_0 u + A^t c$$

$$A^2 = (I + ub^t) A^-$$

と定義する。 A^+ も A^- も A の一般逆行列である。前節の w は

$$w = \begin{bmatrix} -c_0 + b^t y \\ y \end{bmatrix}$$

となり $Ay = c$ を満たす。 $w \geq 0$ の条件で c_0 を増大させることはある y から u の方向に $b^t y$ を $y \geq 0$ の条件の下で最大化していることになる。方向 u は主問題に対するアフィン・スケーリングの方向に一致している。

以上のことよりアルゴリズムの概要は次のようになる。

アルゴリズム

ステップ0 (初期化) $x \in \text{int } H$, $c_0 \leq c_0^*$ とおく。

ステップ1 (停止規準) $c^t x - c_0 \leq \epsilon$ ならばストップ。

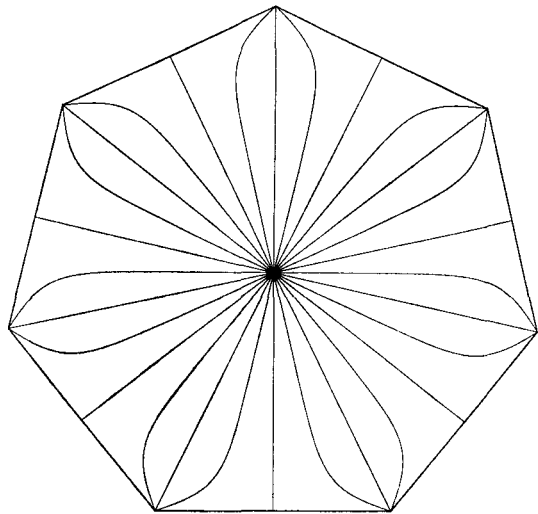


図 2 $\hat{x}=d_2$ の解のトラジェクトリ

ステップ2 (c_0 の更新) 主変数 y を

$$y = -c_0 n + A^t c$$

によって計算する。 $y \succ 0$ ならば $c_{0+} = c_0$ においてステップ3へ。さもなければ c_{0+} を

$$c_{0+} = \max \{ b^t (y + \alpha u) \mid y + \alpha u \geq 0 \}$$

によって計算する。

ステップ3 (探索方向の計算) $d_p = d_1 + \zeta_2 d_2$ とおく。

ステップ4 (直線探索) F を d_p に沿って近似的に最小化してステップ幅 α を決定する。

ステップ5 (x の更新) $x_+ = x + \alpha d_p$ においてステップ1へ。

図1~4に d_1, d_2, d_p, d_N によって生成されるトラジェクトリの例を示す。

5. インプリメンテーション

以下に上記のような内点法のアルゴリズムを大規模問題用にインプリメントする上でのいくつかの注意点を述べる。

(i) 直線探索

与えられた方向 d に沿って関数 F を最小化することは

$$f = \ln F = (m+1) \ln (c^t x - c_0) - \sum_i \ln (a_i^t x - b_i)$$

であるから

$$f'(\alpha) = \frac{(m+1)c^t d}{(c^t x - c_0 + \alpha c^t d)} - \sum_i \frac{a_i^t d}{a_i^t x - b_i + \alpha a_i^t d} = 0$$

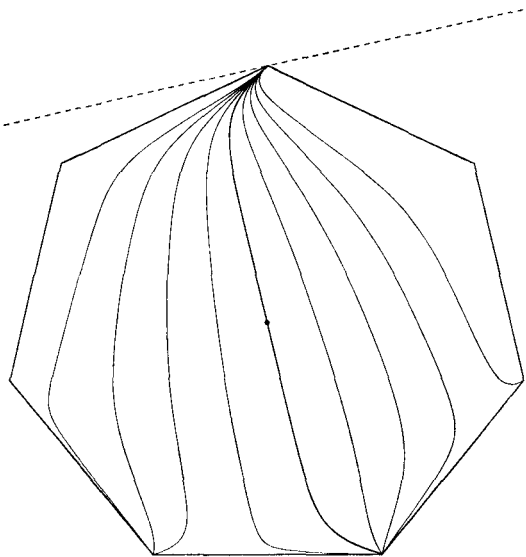


図 3 $\dot{x} = d_p$ の解のトラジェクトリ

を α について解くことを意味する。上式を

$$f'(\alpha) \simeq \phi(\alpha) = \frac{\lambda_1}{\alpha - \bar{a}_c} - \frac{\lambda_2}{\alpha - \bar{a}}$$

と関数近似する。ここで \bar{a}_c は $f'(\alpha)$ の第 1 項の分母が 0 となる α の値で、 \bar{a} は第 2 項の和の中で分母の 0 となる α の最小値 (> 0) である。 $f'(\alpha)$ の値と $f''(\alpha)$ の値から λ_1, λ_2 を求めて $\phi(\alpha) = 0$ を解き

$$\alpha = \bar{\alpha} + \frac{\lambda_2(\bar{a} - \bar{a}_c)}{\lambda_1 - \lambda_2}$$

とおいて新しい α の値とする反復法が考えられる。この方法は非常に速い収束をすることが実験的に確かめられている。しかし大規模問題に対して上記の手続きを行なうと手間がかかりすぎるので通常は $\alpha = (0.9 \sim 0.999)\bar{\alpha}$ とすることが多い。

(ii) 探索方向の計算

ベクトル d_p を求めるためには、ある右辺ベクトル $\eta \in R^n$ に対して

$$Bd = \eta$$

を解く必要がある。ここで

$$B = AD^{-2}A^t$$

であるから B は反復の途中で構造は変化せず、 D の値が変わることによってのみ B の値も変化する。通常は

$$B = LL^t$$

と Cholesky 分解をして (L は下三角行列)

d を計算する。 B の三角分解において B の零要素であった場所に非零要素が現われる

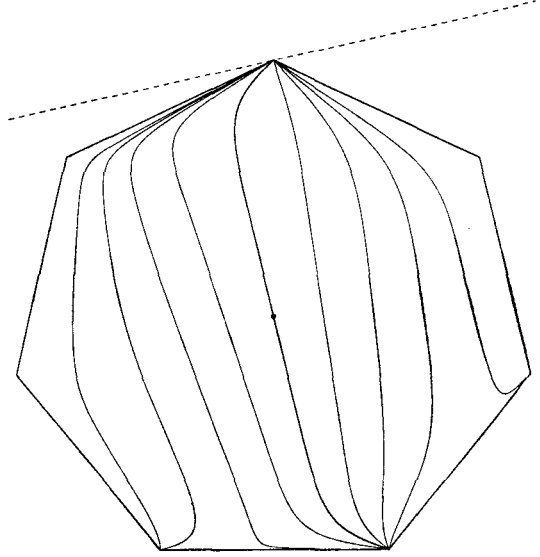


図 4 $\dot{x} = d_N$ の解のトラジェクトリ

ことを Fill-in と呼ぶが、この数が多くなると計算の効率が低下する。そこで Fill-in の減少のために前もって B の行と列の順序を入れ替えておく (オーダリング)。この方法としては最小次数 (minimum degree) 法を使用することが多い。図 5, 6 にある問題の B とそのオーダリング後の構造を示す。この問題に関してはオーダリングをすると Fill-in の数は約 1/40 に減少する。

(iii) 実験結果

表 1, 2 にわれわれの方法を標準的なテスト問題に適用して結果を Minos5 との比較で示す。使用計算機は SUN 3/160 である。

また以下のようなスケジューリングの問題 (Chvatal) を色々なサイズで解いた結果を図 7 に示す。

$$\min \sum_{j=1}^k (20y_j + 8z_j + t_j)$$

subject to

表 1 問題

問題	m	n
scsd 1	760	77
scsd 6	1350	147
scsd 8	2750	397
sctap 1	660	300
sctap 2	2500	1090
sctap 3	3340	1480

表 2 Minos との比較

問題	MINOS		本方法		比
	iter.	時間 (sec)	iter.	時間 (sec)	
scsd 1	260	48.5	12	22.2	2.18
scsd 6	508	152.2	17	48.6	3.13
scsd 8	1409	865.0	15	85.3	10.14
sctap 1	414	81.9	26	37.6	2.18
sctap 2	1494	897.1	22	220.7	4.06
sctap 3	1489	1183.5	26	326.5	3.62

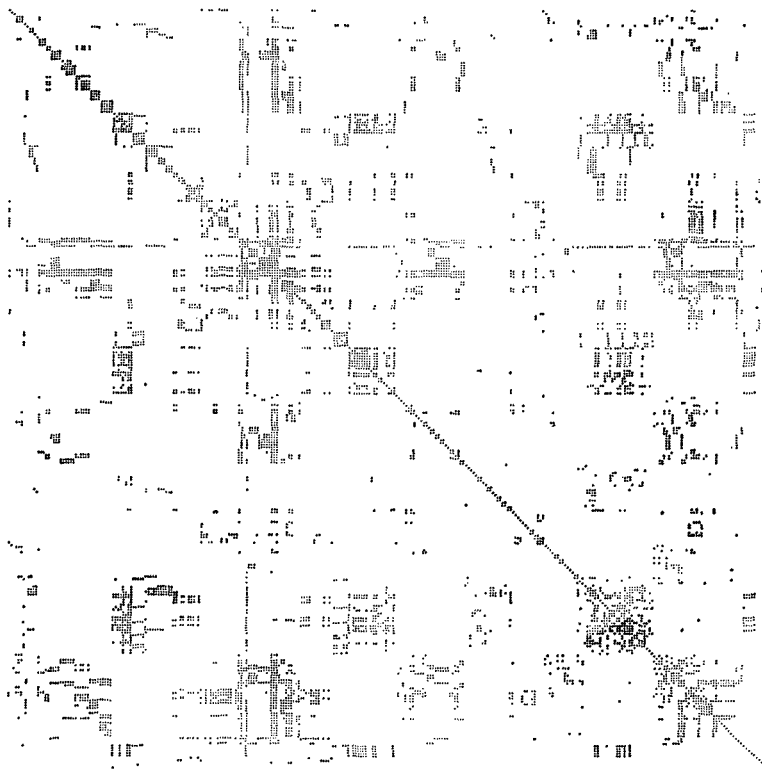


図 5 オーダリング
前の B

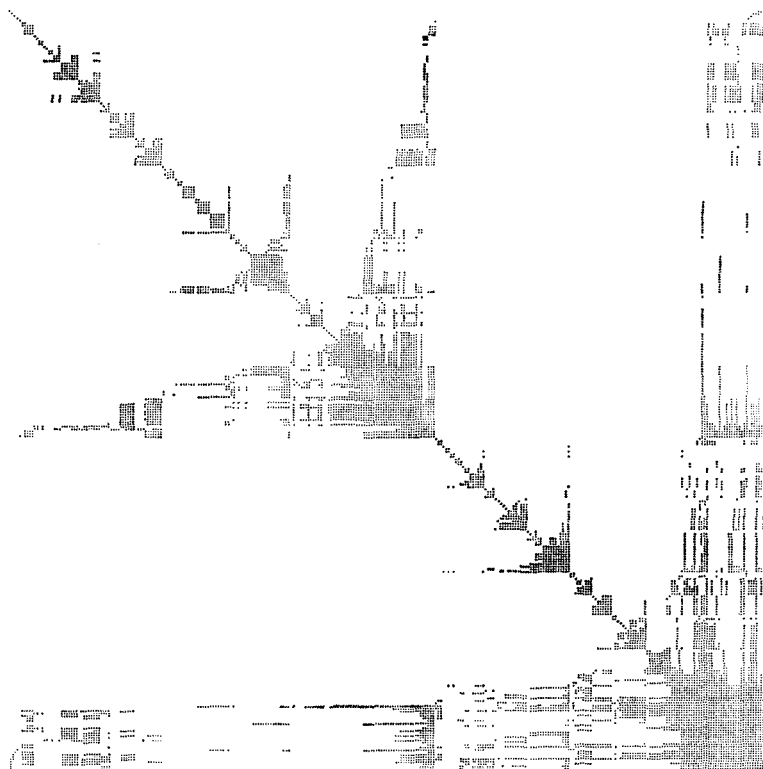


図 6 オーダリング
後の $L+L^t$

$$\begin{aligned}
& -x_j + x_{j-1} \leq 800, \\
& y_j \leq 0.3x_j, \\
& z_{j-1} + x_j + y_j = d_j + z_j \\
& t_j \geq 15(x_j - x_{j-1}), \\
& t_j \geq 21(x_{j-1} - x_j), \\
& x_0 = 5800, \\
& z_0 = 0, \\
& x_j, y_j, z_j, t_j \geq 0 \quad \text{for } j = 1, \dots, k
\end{aligned}$$

この実験の使用計算機は SUN4/280 である。
 $n = 4608$ において MINOS の約30倍の効率を得ている。

参考文献

[1] I. Adler, N. Karmarkar, M. G. C. Resende and G. Veiga, : An Implementation of Karmarkar's Algorithm for Linear Programming. Working Paper, Operations Research Center, University of California, Berkeley, California, May 1986.

[2] I. I. Dikin, : Iterative Solution of Problems of Linear and Quadratic Programming. Soviet Mathematics Doklady, 8, 674—675, 1967.

[3] R. M. Freund. Polynomial-time Algorithms for Linear Programming Based Only on Primal Scaling and Projected Gradients of a Potential Function. OR 182—88, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988.

[4] de G. Ghellinck and J.-P. Vial, : A Polynomial Newton Method for Linear Programming. Algorithmica, 1, 425—454, 1986.

[5] G. C. Gonzaga, : Polynomial affine Algorithm for Linear Programming. ES-139/88, Dept. of System Engineering and Computer Sciences, COPPE-Federal University of Rio de Janeiro, Brasil, 1988.

[6] M. Iri and H. Imai, : A Multiplicative Barrier Function Method for Linear Programming. Algorithmica, 1, 455—482, 1986.

[7] N. Karmarkar, : A New Polynomial-Time Algorithm for Linear Programming. Combinatorica, Vol. 4 (1984), 373—395.

[8] M. Kojima, S. Mizuno, and A. Yoshise, : A Primal-dual Interior Point Algorithm for Linear

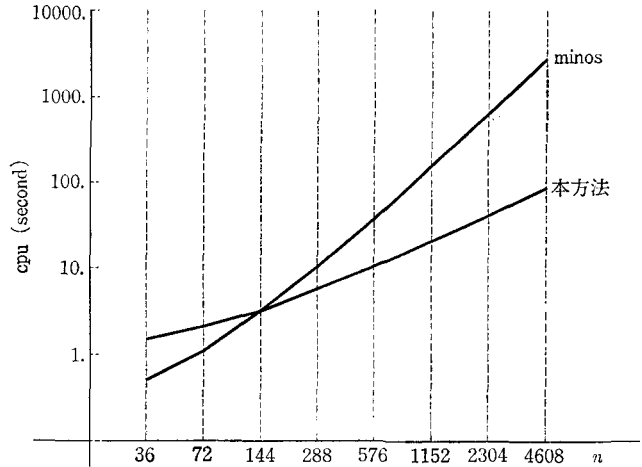


図7 MINOS との時間の比較

Programming. Dept. of Information Sciences Tokyo Institute of Technology, B-188, 1987.

[9] M. Kojima, S. Mizuno, and A. Yoshise, : An $O(\sqrt{n}L)$ Iteration Potential Reduction Algorithm for Linear Complementarity Problems. Dept. of Information Sciences Tokyo Institute of Technology, B-217, 1988.

[10] J. Renegar, : A Polynomial-Time Algorithm, Based on Newton's Method, for Linear Programming. MSRI 07118-86, Mathematical Sciences Research Institute, Berkeley, California, June 1986.

[11] K. Tanabe, : Complementarity-enforcing Centered Newton Method for Linear Programming, manuscript at the Symposium, "New Methods for Linear Programming", The Institute of Statistical Mathematics, Tokyo 1987.

[12] T. Tsuchiya, and K. Tanabe, : Local Convergence Properties of New Methods in Linear Programming. The Institute of Statistical Mathematics, Tokyo 1988.

[13] H. Yamashita, : Polynomially and Quadratically Convergent for Linear Programming, : Mathematical Systems Institute, Inc. Tokyo 1986.

[14] H. Yamashita, and T. Arakawa, : An Implementation of Polynomially and Quadratically Convergent method for Linear Programming-Dense Case. Manuscript at the Symposium, "New Methods for Linear Programming", The Institute of Statistical Mathematics, Tokyo 1987.