統計数理 (1992)
第 40 巻 第 1 号 27-44

線形計画問題の主双対内点法

統計数理研究所　水　野　眞　治

(1992 年 1 月　受付)

1. はじめに

十年ほど前まで，線形計画問題を解く方法といえば G.B. Dantzig が 1940 年代に考案したシンプレックス法のことを指していた．線形計画問題を解くソフトウエアパッケージは，シンプレックス法に基づいて開発され，実際にかなり大規模な問題を解くことができた．線形計画問題の解法として，シンプレックス法がかなり実用的であるため，他の解法を考えるということはほとんどなかった．

しかし，数理計画法の理論的な研究者の間では，シンプレックス法が指数オーダの解法であり，最悪の場合には 100 変数程度でも現在の計算機では解けない問題が存在することが知られていた．したがって，多項式オーダの解法が存在するかという問題が研究者の関心の的であった．この問題に答えたのは，Khachiyan (1979) であった．彼は，楕円体法と呼ばれる多項式オーダの解法を開発した．しかし，楕円体法は実用的な解法ではなかった．

Karmarkar (1984) は，楕円体法とは異なる多項式オーダの新解法を発表した．その解法は，理論的に優れているのみならず，シンプレックス法よりも早く例題を解くことができると発表された．Karmarkar の発表以後，その解法に変形，改良を加えたさまざまなアルゴリズムが開発された．それらのアルゴリズムは，実行可能領域の（相対的）内部に点列を生成するという共通の特徴を持つ．その特徴は，実行可能領域の端点列を生成するシンプレックス法と比較して，大きな相違点である．それゆえに，Karmarkar のアルゴリズムとそれ以後に開発されたアルゴリズムは内点法と呼ばれる．

内点法には多くの種類があり，それらの分類の仕方もさまざまである．対象とする問題により分類すると，標準形の線形計画問題（等式制約と非負変数）を解く主内点法，その双対問題（不等式制約のみ）を解く双対内点法，主問題と双対問題の組を同時に解く主双対内点法がある．また，線形計画問題のみでなく，二次計画問題，相補性問題，非線形計画問題を解く内点法もある．点列の生成方法により分類すると，射影変換法，アフィンスケーリング法，(センター) パス追跡法，点列追跡法などがある．ポテンシャル関数あるいは罰金関数を減少させる方法は，ポテンシャル減少法あるいは罰金関数法と呼ばれる．一般的な内点法についてより詳しい解説は，Todd (1989) あるいは Goldfarb and Todd (1989) などに書いてある．Kranich (1991) には 900 以上にのぼる内点法の参考文献が上げてある．

本論の目的は，主問題と双対問題の組を同時に解く主双対内点法を総括的に示すことである．主双対内点法は，Megiddo (1989) による解析結果を基礎として Kojima et al. (1989a) と Tanabe (1987a, 1987b) により提案された．Kojima et al. (1989a) のアルゴリズムは多項式オーダである．McShane et al. (1989) と Lustig et al. (1991) は，主双対内点法により開発したプログラムを使って数値実験をし，効率よく大規模な線形計画問題を解くという結果を報告した．

次章では，線形計画問題の標準形の主問題と双対問題を定義し，その最適条件を示す．3章では，一般的なスタイルで主双対内点法を示し（アルゴリズム1），そのアルゴリズムが初期点の計算，収束判定，探索方向の計算，ステップサイズの計算から構成されることを説明する．4章から7章では，上記のそれぞれの構成要素（初期点の計算，収束判定，探索方向の計算，ステップサイズの計算）についてより詳しく述べる．8章では，さまざまな主双対内点法のアルゴリズムを3章に述べるアルゴリズム1の枠組みで記述し，それぞれの大域的収束性を示す．9章では，主双対内点法の局所的収束性についてまとめ，いくつかのアルゴリズムが局所的に超一次収束あるいは二次収束することを述べる．10章では，アルゴリズム全体の計算複雑度について示し，同時に計算複雑度を下げるための行列の部分的更新について述べる．11章では，実際にプログラミングをする場合の指針を述べる．

2. 線形計画問題と双対問題

標準形の線形計画問題は次式により定式化できる：

$$
\begin{array}{ll}
\text{Minimize} & c_1x_1+c_2x_2+\cdots+c_nx_n. \\
\text{Subject to} & a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1, \\
 & a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2, \\
 & \qquad\qquad\vdots \qquad\qquad\qquad \vdots \\
 & a_{m1}x_1+a_{m2}x_2+\cdots+a_{mn}x_n=b_m, \\
 & \qquad\qquad x_1, x_2,\ldots, x_n\geq 0.
\end{array}
$$

この問題は行列とベクトルを使い次式により表せる：

$$
\begin{array}{lll}
\text{(P)} & \text{Minimize} & c^Tx. \\
 & \text{Subject to} & Ax=b, \quad x\geq 0.
\end{array}
$$

ここで，c は n 次の定数ベクトル，A は m 行 n 列の定数行列，b は m 次の定数ベクトル，x は n 次の変数ベクトルであり，T はベクトルあるいは行列の転置を表す．次の問題(D)は，問題(P)の双対問題である：

$$
\begin{array}{lll}
\text{(D)} & \text{Maximize} & b^Ty. \\
 & \text{Subject to} & A^Ty+z=c, \quad z\geq 0.
\end{array}
$$

線形計画法の双対定理により，主問題（P）と双対問題（D）がともに実行可能解を持つならば，問題（P）と（D）はそれぞれ最適解を持ち，その最適値は等しい．このことから，主問題の最適解 x^* と双対問題の最適解 (y^*, z^*) の組 (x^*, y^*, z^*) は，次の条件をみたす点である：

$$
c^Tx=b^Ty, \qquad Ax=b, \qquad A^Ty+z=c, \qquad x\geq 0, \qquad z\geq 0.
$$

ここで，次式より，第一の条件は $x^Tz=0$ と同値である：

$$
x^Tz=x^T(c-A^Ty)=c^Tx-(Ax)^Ty=c^Tx-b^Ty.
$$

$x\geq 0, z\geq 0$ のとき，$x^Tz=0$ と $x_iz_i=0$ $(i=1, 2,\ldots, m)$ は同値であるので，線形計画問題(P)と(D)の解 x と (y, z) は次の方程式系の解である．

$$
\begin{pmatrix} Xz \\ Ax-b \\ A^Ty+z-c \end{pmatrix}=0. \tag{2.1}
$$

ここで $X=\mathrm{diag}(x)$ は，x の各要素と等しい値を対角成分に持つ対角行列である．逆に，方程式系（2.1）の解 (x, y, z) が $x\geq 0, z\geq 0$ をみたすならば，x と (y, z) はそれぞれ (P) と (D) の解である．

3. 主双対内点法

この章では，一般的なスタイルで主双対内点法を示す．主双対内点法は，反復解法であり，点列 $\{(x^k, y^k, z^k)\}$ を生成する．点 (x^k, y^k, z^k) が与えられているとし，次の点 $(x^{k+1}, y^{k+1}, z^{k+1})$ の求め方を示す．

n 次のベクトル v に対して，次の方程式系を考える：

$$\begin{pmatrix} Xz-v \\ Ax-b \\ A^Ty+z-c \end{pmatrix}=0. \tag{3.1}$$

この方程式系は，$v=0$ のとき方程式系（2.1）と一致する．主双対内点法は，0 に収束する点列 $\{v^k\}$ をもちいて，各反復で $v=v^k$ とした方程式系（3.1）の近似解を求める方法である．点 (x^k, y^k, z^k) において，$v=v^k$ とした方程式系（3.1）のニュートン方向 $(\Delta x^k, \Delta y^k, \Delta z^k)$ は，次の線形方程式系の解である：

$$\begin{pmatrix} Z_k & 0 & X_k \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix}\begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{pmatrix}=-\begin{pmatrix} X_k z^k-v^k \\ Ax^k-b \\ A^Ty^k+z^k-c \end{pmatrix}. \tag{3.2}$$

ここで，$Z_k=\mathrm{diag}(z^k)$ かつ $X_k=\mathrm{diag}(x^k)$ である．次の点 $(x^{k+1}, y^{k+1}, z^{k+1})$ は，主問題の変数と双対問題の変数がそれぞれのステップサイズで探索方向 $(\Delta x^k, \Delta y^k, \Delta z^k)$ へ進んだ点である：

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \end{pmatrix}=\begin{pmatrix} x^k+\alpha_P^k\Delta x^k \\ y^k+\alpha_D^k\Delta y^k \\ z^k+\alpha_D^k\Delta z^k \end{pmatrix}. \tag{3.3}$$

ここで α_P^k と α_D^k は，それぞれ主問題と双対問題のステップサイズを表すパラメータである．一般的な主双対内点法は，下記のアルゴリズム 1 である．

アルゴリズム 1.

ステップ 0：$k=0$ とし，初期点 (x^0, y^0, z^0) を求める．
ステップ 1：収束条件が成立したならば停止する．
ステップ 2：ベクトル v^k を定め，方程式系（3.2）の解 $(\Delta x^k, \Delta y^k, \Delta z^k)$ を求める．
ステップ 3：パラメータ α_P^k と α_D^k の値を定め，(3.3) により $(x^{k+1}, y^{k+1}, z^{k+1})$ を求める．
ステップ 4：$k\leftarrow k+1$ としてステップ 1 へ行く．

上記の主双対内点法を実際に一つのアルゴリズムとして確定するには，以下の 4 点についてその方法を具体的に定める必要がある：

・初期点 (x^0, y^0, z^0) を求める方法．
・収束を判定する方法．

・探索方向 $(\Delta x^k, \Delta y^k, \Delta z^k)$ を計算するためにベクトル v^k を決める方法.
・パラメータ α_P^k と α_D^k の値を求める方法.

これらの方法をうまく定めれば，問題を実際に効率よく解くアルゴリズムを作成でき，理論的によい収束性を持つアルゴリズムも作成できる.

経験的に例題を効率よく解くといわれているアルゴリズムの一つ (Lustig et al. (1991)) を示す. 初期点は, $x^0>0$ と $z^0>0$ をみたす適当な点とする：たとえば $e=(1, 1,\ldots, 1)$ に対して $x^0=e$, $y^0=0$, $z^0=e$ とする. 収束判定は，十分小さな正の数 ϵ に対して,

$$x^{k^T}z^k \le \epsilon, \quad \| Ax^k - b \| \le \epsilon, \quad \| A^T y^k + z^k - c \| \le \epsilon$$

とする. ベクトル v^k は,

$$v^k = \gamma \frac{x^{k^T} z^k}{n} e,$$

$$\gamma = \begin{cases} 1/n & \text{if } n \le 5000, \\ 1/\sqrt{n} & \text{if } n > 5000 \end{cases}$$

により定める. パラメータ α_P^k と α_D^k は,

$$\begin{aligned} \alpha_P^k &= 0.9995 \hat{\alpha}_P^k, \\ \alpha_D^k &= 0.9995 \hat{\alpha}_D^k, \\ \hat{\alpha}_P^k &= \sup \{\alpha : x^k + \alpha \Delta x^k \ge 0\}, \\ \hat{\alpha}_D^k &= \sup \{\alpha : z^k + \alpha \Delta z^k \ge 0\} \end{aligned}$$

により定める. しかし，このアルゴリズムが収束するという理論的な保証はない. 実際, Kojima et al. (1990) は, $\hat{\alpha}_P^k = +\infty$ となり，このアルゴリズムが収束しない例題を示した.

4. 初 期 点

主双対内点法の初期点は，三つのタイプに分類できる：初期点を (x^0, y^0, z^0) と表せば,

タイプ 1: ベクトル x^0 と z^0 が正である，すなわち $x^0>0$, $z^0>0$.
タイプ 2: タイプ 1 の条件に加え, x^0 と (y^0, z^0) がそれぞれ問題 (P) と (D) の実行可能解である，すなわち $Ax^0=b$, $A^Ty^0+z^0=c$, $x^0>0$, $z^0>0$.
タイプ 3: (x^0, y^0, z^0) がセンターパス S の近傍 N に含まれる，すなわち $(x, y, z) \in N$. ここで，センターパス S は次の式により定義される：

$$S = \{(x, y, z): Xz = \mu e,\ \mu = x^T z/n,\ Ax = b,\ A^T y + z = c,\ x > 0,\ z > 0\}.$$

近傍 N は，多くの場合問題 (P) と (D) の実行可能領域の内部 $\{(x, y, z): Ax=b,\ A^Ty+z=c,\ x>0,\ z>0\}$ に含まれる.

任意の線形計画問題 (P) と (D) に対して，タイプ 1 の初期点は容易に求められる. たとえば $x^0=e$, $y^0=0$, $z^0=e$ とする. しかし，タイプ 2 と 3 の初期点を求めることは，特殊な場合を除くと, 元問題 (P) と (D) を解くことと同程度に難しい. タイプ 3 に使われる近傍 N として代表的なものに次の 4 種類がある：

$$\begin{aligned} N_2(\beta) = \{(x, y, z) : \| Xz - \mu e \|_2 \le \beta\mu,\ \mu = x^T z/n,\ Ax = b,\ A^T y + z = c, \\ x > 0,\ z > 0\} \text{ for } \beta \in [0, 1], \end{aligned}$$

$$N_\infty(\beta)=\{(x,y,z):\ \|Xz-\mu e\|_\infty\le\beta\mu,\ \mu=x^Tz/n,\ Ax=b,\ A^Ty+z=c,$$
$$x>0,\ z>0\}\ \text{for}\ \beta\in[0,1],$$
$$N_-(\beta)=\{(x,y,z):\ Xz\ge(1-\beta)\mu e,\ \mu=x^Tz/n,\ Ax=b,\ A^Ty+z=c,$$
$$x>0,\ z>0\}\ \text{for}\ \beta\in[0,1],$$
$$N_{\ln}(\beta)=\{(x,y,z):\ n\ln\mu-\sum_{i=1}^n\ln(x_iz_i)\le\beta,\ \mu=x^Tz/n,\ Ax=b,\ A^Ty+z=c,$$
$$x>0,\ z>0\}\ \text{for}\ \beta\ge0.$$

ここで，$\|\cdot\|_2$ と $\|\cdot\|_\infty$ は，それぞれ ℓ_2 ノルムと ℓ_∞ ノルムを表す．上記の各近傍は，$\beta=0$ のときセンターパス S と一致し，β の値が大きくなるほど広くなる．さらに，

$$N_2(\beta)\subset N_\infty(\beta)\subset N_-(\beta)$$

が任意の $\beta\in[0,1]$ に対して成立する．これらの近傍は，問題の実行可能領域に含まれ，特に $N_-(1)$ と $N_{\ln}(\infty)$ は実行可能領域の内部全体と一致する．

タイプ 2 あるいは 3 の初期点を求めるには，一般的に人工的に問題を作成する．人工問題は，明らかな初期点を持ち，さらにその問題を解けば元問題が解けるようにする．適当なベクトル $x^0>0, y^0, z^0>0$（例えば $x^0=e, y^0=0, z^0=e$）および二つの実数 ρ と σ に対して，人工問題 (P′) と (D′) を次のように定義する：

$$\begin{array}{llll}
(\mathrm{P}') & \text{Minimize} & c'^Tx'. & \\
 & \text{Subject to} & A'x'=b', & x'\ge0.\\
(\mathrm{D}') & \text{Maximize} & b'^Ty'. & \\
 & \text{Subject to} & A'^Ty'+z'=c', & z'\ge0.
\end{array}$$

ここで，

$$c'=\begin{pmatrix}c\\ \rho\\ 0\end{pmatrix},\quad b'=\begin{pmatrix}b\\ \sigma\end{pmatrix},\quad A'=\begin{pmatrix}A & b-Ax^0 & 0\\ (A^Ty^0+z^0-c)^T & 0 & 1\end{pmatrix},$$
$$x'=\begin{pmatrix}x\\ x_{n+1}\\ x_{n+2}\end{pmatrix},\quad y'=\begin{pmatrix}y\\ y_{m+1}\end{pmatrix},\quad z'=\begin{pmatrix}z\\ z_{n+1}\\ z_{n+2}\end{pmatrix}.$$

上記の (D′) は問題 (P′) の双対問題である．次の条件

$$\sigma>(A^Ty^0+z^0-c)^Tx^0,$$
$$\rho>(b-Ax^0)^Ty^0$$

が成立するならば，問題 (P′) と (D′) のタイプ 2 の初期点 x' と (y', z') は次式により求められる：

$$x'=\begin{pmatrix}x^0\\ 1\\ \sigma-(A^Ty^0+z^0-c)^Tx^0\end{pmatrix},\quad y'=\begin{pmatrix}y^0\\ -1\end{pmatrix},\quad z'=\begin{pmatrix}z^0\\ \rho-(b-Ax^0)^Ty^0\\ 1\end{pmatrix}.$$

タイプ 3 の初期点を求めるためには，若干の工夫を加える必要がある．ここでは，センターパス上の初期点を持つ人工問題を作成する．その初期点は，センターパスのすべての近傍に含まれる．正の数 M に対して，$x^0=Me, y^0=0, z^0=Me$ とする．この (x^0, y^0, z^0) について，上記の人工問題を作成する，ただし

$$\sigma=(A^Ty^0+z^0-c)^Tx^0+M^2,$$
$$\rho=(b-Ax^0)^Ty^0+M^2$$

とする．このとき，上記に定義した (x', y', z') は，$X'z'=M^2e$ をみたし，センターパス上の初期点である．

人工問題を解けば元問題も解けることを示す．人工問題 (P′) と (D′) は，ともに実行可能解を持つので，最適解を持つ．次の定理は，人工問題の最適解と元の問題の関係を示している．定理の証明は，たとえば水野 (1989) に示されている．

定理 4.1. 人工問題 (P′) と (D′) の最適解をそれぞれ $x'^*=(x^*, x^*_{n+1}, x^*_{n+2})$ と $(y'^*, z'^*)=(y^*, y^*_{m+1}, z^*, z^*_{n+1}, z^*_{n+2})$ とする．

- $x^*_{n+1}=0$ かつ $z^*_{n+2}=0$ ならば，x^* と (y^*, z^*) はそれぞれ問題 (P) と (D) の最適解である．
- 条件 $\sigma>(A^Ty^0+z^0-c)^Tx$ をみたす問題 (P) の最適解が存在し，条件 $\rho>(b-Ax^0)^Ty$ をみたす問題 (D) の最適解が存在するならば $x^*_{n+1}=0$ かつ $z^*_{n+2}=0$ である．

線形計画問題は，最適解が存在するならば，その中に基底解が存在する．線形計画問題のサイズ（問題を計算機に記述するのに必要なビット数）を L で表せば，任意の基底解は分母と分子の大きさがともに $2^{O(L)}$ の分数で表せる（たとえば伊理 (1986) 参照)．ベクトル $A^Ty^0+z^0-c$ と $b-Ax^0$ の大きさが $2^{O(L)}$ となるように x^0, y^0, z^0 を定めれば，問題 (P) と (D) の基底解 x と (y, z) に対して，$(A^Ty^0+z^0-c)^Tx$ と $(b-Ax^0)^Ty$ の大きさは $2^{O(L)}$ となる．したがって，σ と ρ（あるいは M）の大きさを $2^{\Omega(L)}$ にすれば，定理 4.1 から，人工問題を内点法で解き最適解 (x^*, y^*, z^*) を得たとき，$x^*_{n+1}>0$ または $z^*_{n+2}>0$ ならば元問題 (P) または (D) に最適解が存在せず，$x^*_{n+1}=0$ かつ $z^*_{n+2}=0$ ならば元問題 (P) と (D) の最適解を得る．

ここでは，人工問題を一つ固定して，内点法の初期点を求める方法を示した．この方法は，大きな定数 (ρ と σ) を前もって準備しなければならないという欠点を持つ．その定数が大きすぎれば計算効率が悪くなり，小さすぎれば元問題の解を得られない場合がある．Kojima et al. (1991e) は，この欠点を解消するために，小さな定数から人工問題をはじめ，その定数が小さすぎる場合をアルゴリズムの実行中に判断し，定数を大きく更新する方法を提案した．

5. 収束判定

収束判定は，実行可能内点列を生成する内点法とそれ以外の内点法（または外点法）とで異なる．点 (x^k, y^k, z^k) が実行可能 $(Ax^k=b,\ A^Ty^k+z^k=c,\ x^k\geq 0,\ z^k\geq 0)$ であるとき，$x^{k^T}z^k=0$ が成立すれば，x^k と (y^k, z^k) はそれぞれ問題 (P) と (D) の最適解である．したがって，実行可能内点列を生成する内点法の収束判定は，十分小さな正の数 ϵ に対して，

$$x^{k^T}z^k\leq\epsilon$$

とする．実行可能でない点列を生成している場合の収束判定は，十分小さな正の数 $\epsilon_1, \epsilon_2, \epsilon_3$ に対して，

$$\begin{aligned}&x^{k^T}z^k\leq\epsilon_1,\\&\|Ax^k-b\|\leq\epsilon_2,\\&\|A^Ty^k+z^k-c\|\leq\epsilon_3,\\&x^k\geq 0,\qquad z^k\geq 0\end{aligned}$$

とする．また，点列が発散する場合を判定するために十分大きな正の数 ω に対して

$$\|(x^k, y^k, z^k)\| \geq \omega$$

を収束判定に加える場合もある．

上記に現れた ϵ と ω は，実際の計算では適当な大きさにとるが，理論的に評価するとき $\epsilon=2^{-\delta L}$ と $\omega=2^{\delta L}$ とする．ここで，L は線形計画問題のサイズを表し，δ は適当な正の整数である．

6. 探索方向

3章で示したように，アルゴリズム1の探索方向 $(\Delta x^k, \Delta y^k, \Delta z^k)$ は，現在の点 (x^k, y^k, z^k) における方程式系 (3.1) のニュートン方向であり，線形方程式系 (3.2) を解き計算する．したがって，探索方向はベクトル v^k に依存して決まる．方程式系 (3.1) に現れるベクトル v^k の決め方を示す．多くの内点法では，正の数 μ に対して $v^k=\mu e$ とする．$v^k=\mu e$ と表せるとき，方程式系 (3.1) の解はセンターパス上の点である．したがって，この場合の探索方向は，センターパス上の点を求めるための方向である．

$v^k=\mu e$ とするときに，パラメータ μ の値は定数 $\gamma\in[0,1]$ に対して $\mu=\gamma x^{k^T}z^k/n$ とする．$\gamma=1$ のときの探索方向はセンタリング方向と呼ばれ，$\gamma=0$ のときの探索方向はアフィンスケーリング方向と呼ばれる．ショートステップの内点法では γ として1に近い値を使い，ロングステップの内点法では γ として小さな値を使う．

7. ステップサイズ

アルゴリズム1では，ステップサイズをパラメータ α_P^k と α_D^k により表している．このパラメータ値の決め方として下記に述べる方法がある：

・定数とする．
・実行可能領域に含まれる楕円体を使う．
・実行可能領域の境界までのパラメータ値を使う．
・実行可能領域に含まれるセンターパスの近傍 N を使う．
・ポテンシャル関数を使う．

定数とする代表的な方法は，$\alpha_P^k=\alpha_D^k=1$ とするニュートン法である．現在の点が目標とするセンターに十分近い場合にもちいる方法である．

楕円体を使いステップサイズを決める例として，定数 $\gamma\in(0,1)$ に対して，

$$\alpha_P^k=\gamma/\|X^{-1}\Delta x^k\|,$$
$$\alpha_D^k=\gamma/\|Z^{-1}\Delta z^k\|$$

とする方法がある．この方法は，主問題に対する Karmarkar 法などで使われるが，主双対内点法では余り使われていない．

実行可能領域の境界までのパラメータ値 $\hat{\alpha}_P^k$ と $\hat{\alpha}_D^k$ は，すでに3章で定義した．それを使った内点法の一つも，3章に示した．このような方法は，多くの計算実験に使われ，実際に効率よく例題を解いたという報告がある (Lustig et al. (1991), McShane et al. (1989))．

実行可能領域に含まれるセンターパスの近傍 N の例については，すでに4章にいくつか示した．近傍 N を使ってステップサイズを決める代表的な方法では，

$$\alpha_P^k=\alpha_D^k=\sup\{\alpha:(x^k,y^k,z^k)+\alpha(\Delta x^k,\Delta y^k,\Delta z^k)\in N\}$$

とする.

主双対内点法に使われるポテンシャル関数として代表的なものは

$$f_\rho(x,z)=\rho\ln x^Tz+n\ln(x^Tz/n)-\sum_{i=1}^{n}\ln x_iz_i$$

である. ここで, $\rho>0$ は定数である. この関数を使い, ステップサイズを

$$\alpha_P^k=\alpha_D^k=\arg\min f_\rho(x^k+\alpha\Delta_x^k,\ z^k+\alpha\Delta_z^k)$$

により定める. このときの最小点は, 変数 α についての一次元探索により求められる. 一次元探索により最小点を求める代わりに, Goldstein-Armijo の規則をみたすステップサイズを求める方法もある. ここに述べたポテンシャル関数は一つの例であり, 他の関数を使用することも可能である.

8. アルゴリズムと収束性

この章では, アルゴリズム1の枠組みに当てはまるいくつかの内点法を述べ, それぞれのアルゴリズムの理論的な大域的収束性を示す. 取り上げる方法は, アフィンスケーリング法, 点列追跡法, パス追跡法, プレディクタ・コレクタ法, ロングステップ法, ポテンシャル減少法, 外点法の7種類である.

8.1 アフィンスケーリング法

アフィンスケーリング法は, 主内点法として Dikin (1967), Barnes (1986), Adler et al. (1989) が提案した. 主双対内点法としてのアフィンスケーリング法は, Monteiro et al. (1990) が提案した. アフィンスケーリング法は, 各反復において常に探索方向を $v^k=0$ として計算する方法である.

Monteiro et al. (1990) は, 初期点 (x^0,y^0,z^0) がセンターパス上 $(Ax^0=b,\ A^Ty^0+z^0=c,\ X_0z^0=\mu^0e,\ \mu^0=x^{0^T}z^0/n,\ x^0>0,\ z^0>0,\ X_0=\mathrm{diag}(x^0))$ にあり, 収束判定を $x^{k^T}z^k\le\epsilon$ とするとき, ステップサイズを

$$\alpha_P^k=\alpha_D^k=\frac{1}{\ln(n\mu^0/\epsilon)}$$

とすれば, 高々 $n(\ln(n\mu^0/\epsilon))^2$ 回の反復でアルゴリズム1が収束するということを証明した. 理論的には, $\mu^0=2^{O(L)}$ かつ $\epsilon=2^{-O(L)}$ であるので, $O(nL^2)$ 反復のアルゴリズムである.

8.2 点列追跡法

点列追跡法は, Mizuno (1989) が提案したアルゴリズムである. 線形計画問題 (P) と (D) のタイプ2の初期点 (x^0,y^0,z^0) $(Ax^0=b,\ A^Ty^0+z^0=c,\ x^0>0,\ z^0>0)$ が既知であると仮定する (あるいはその初期点を持つ人工問題を作成したとする). 収束判定は, $x^{k^T}z^k\le\epsilon$ とする. 初期点に対して, $v^0=X_0z^0$ とする, あるいは不等式 $\|X_0z^0-v^0\|\le0.3\,v_{\min}^0$ をみたす v^0 を定める. アルゴリズム1の第 k 反復では,

$$\|V_{k-1}^{-0.5}(v^{k-1}-v^k)\|\le0.3\sqrt{v_{\min}^{k-1}} \tag{8.1}$$

をみたすベクトル v^k を定める. ここで, $V_{k-1}=\mathrm{diag}(v^{k-1})$, $v_{\min}=\min\{v_i:\ i=1,2,\ldots,n\}$ であ

る．ステップサイズ・パラメータ α_P^k と α_D^k の値はともに 1 とする．このとき，生成される点列 $\{(x^k, y^k, z^k)\}$ は不等式

$$\| X_k z^k - v^k \| \leq 0.3\, v_{\min}^k, \qquad k=0, 1, \ldots$$

をみたす．したがって，$k=l$ に対して

$$e^T v^l \leq \frac{1}{1.3}\epsilon \tag{8.2}$$

が成立すれば，

$$x^{l^T} z^l \leq 1.3 e^T v^l \leq \epsilon$$

となり，収束判定が成立する．

点列追跡法は，v^0 に対して (8.1) と (8.2) をみたす点列 $\{v^k : k=0, 1, \ldots, l\}$ を定めると，(P) と (D) の近似解の組 (x^l, y^l, z^l) を求める．Mizuno (1989) は，$(1/M)e \leq v^0 \leq Me$ のとき，$l=O(\sqrt{n}\ln(M/\epsilon))$ に対して (8.1) と (8.2) をみたす点列 $\{v^k\}$ が存在することを示した．一般に，$\ln(M/\epsilon)=O(L)$ であるので，$O(\sqrt{n}L)$ 反復のアルゴリズムである．このアルゴリズムは，タイプ 2 の任意の初期点 $(Ax^0=b, A^T y^0 + z^0 = c, 2^{-L}e \leq x^0 \leq 2^L e,\ 2^{-L}e \leq z^0 \leq 2^L e)$ に対して $O(\sqrt{n}L)$ 反復が証明されている唯一の内点法である．

8.3 パス追跡法

線形計画問題 (P) と (D) のセンターパス

$$S=\{(x, y, z):\ Ax=b, A^T y + z = c, x>0, z>0, Xz=\mu e, \mu = x^T z/n\}$$

の近傍 N を一つ定める (4 章参照)．近傍 N は実行可能領域の内部に含まれているとする．近傍 N に属するタイプ 3 の初期点 (x^0, y^0, z^0) が既知であると仮定する (あるいはその初期点を持つ人工問題を作成したとする)．収束判定は，$x^{k^T} z^k \leq \epsilon$ とする．ベクトル v^k として，$\mu^k e$ を使う．ステップサイズの決め方には，二通りある．一方は定数 1 を常に使い，他方はセンターパスの近傍を使う．

Kojima et al. (1989a) のアルゴリズムは，近傍として $N_\infty(\beta)$ (β は定数) を使い，反復 k で $v^k = 0.5(x^{k^T} z^k/n)e$ を使う．ステップサイズは，現在の点に依存したセンターパスの近傍により定める．このアルゴリズムは，$O(nL)$ 反復を必要とする．

Kojima et al. (1989b) と Monteiro and Adler (1989a, 1989b) により提案されたアルゴリズムは，近傍として $N_2(\beta)$ ($\beta=0.1$) を使い，反復 k で $v^k=(1-\delta/\sqrt{n})(x^{k^T} z^k/n)e$ (δ は定数) を使い，ステップサイズとして 1 を使う．この方法により生成される点列は，近傍 $N_2(\beta)$ に含まれる．$O(\sqrt{n}L)$ 反復のアルゴリズムである．

Tanabe (1987a, 1987b) のアルゴリズムは，近傍として $N_{\ln}(\beta)$ (β は定数) を使い，反復 k で $v^k = \gamma(x^{k^T} z^k/n)e$ とし，ステップサイズを $\alpha_P^k = \alpha_D^k = \alpha$ として，近傍 $N_{\ln}(\beta)$ 内で双対ギャップ $c^T(x^k + \alpha\Delta x^k) - b^T(y^k + \alpha\Delta y^k)$ を最小にするパラメータ γ と α を二次元探索により求める．

Mizuno et al. (1991) は，近傍として $N_\infty(\beta)$ あるいは $N_{-}(\beta)$ を使い，反復 k で $v^k = \gamma(x^{k^T} z^k/n)e$ を使い，ステップサイズを近傍の境界に定めるとき，任意の定数 $\beta \in (0, 1)$ と $\gamma \in (0, 1)$ に対して，そのアルゴリズムが $O(nL)$ 反復を必要とすることを証明した．

Mizuno et al. (1989) は，各反復において，ベクトル v^k をパラメータ μ を使い $v^k = \mu e$ と表して，ステップサイズを 1 としたときに近傍 N に含まれるような最小の μ の値を計算するア

ルゴリズムを提案した．彼らは，数値実験により，いくつかの例題を効率よく解くという結果を報告した．このアルゴリズムは，理論的に多項式オーダであり，$N=N_2(\beta)$ のとき $O(\sqrt{n}L)$ 反復を必要とする．

8.4 プレディクタ・コレクタ法

プレディクタ・コレクタ法は，プレディクタステップとコレクタステップから成る．両方のステップを合わせて一反復と見ることが一般的であるが，ここではアルゴリズム1の枠組みで示す．この方法も，パス追跡法とみなせるが，プレディクタステップとコレクタステップにおいて定めるベクトル v^k とステップサイズが異なる．プレディクタステップでは，v^k としてコレクタステップより小さなベクトルを使う．

線形計画問題(P)と(D)のセンターパスの近傍 $N_2(\beta)$ (β は定数) に属するタイプ3の初期点 (x^0, y^0, z^0) が既知であると仮定する（あるいはその初期点を持つ人工問題を作成したとする)．収束判定は，$x^{k^T}z^k \leq \epsilon$ とする．各反復の始めに，プレディクタステップを行うかコレクタステップを行うか判定する．

Ding and Li (1991) のアルゴリズムは，k が偶数のときプレディクタステップを行い，奇数のときコレクタステップを行う．プレディクタステップでは，$v^k=(1-\delta/\sqrt{n})(x^{k^T}z^k/n)e$ ($\delta=0.5$) とし，ステップサイズを1とする．コレクタステップでは，$v^k=(x^{k^T}z^k/n)e$ とし，ステップサイズを1とする．このアルゴリズムは，$O(\sqrt{n}L)$ 反復を必要とする．

Mizuno et al. (1991) のアルゴリズムも k が偶数のときプレディクタステップを行い，奇数のときコレクタステップを行う．プレディクタステップでは，$v^k=0$ とし，ステップサイズを近傍 $N_2(\beta)$ の境界に定める．コレクタステップでは，$v^k=(x^{k^T}z^k/n)e$ とし，ステップサイズを1とする．$\beta\in(0, 0.5]$ であるとき，このアルゴリズムは $O(\sqrt{n}L)$ 反復を必要とする．また，各コレクタステップの後に生成される点はセンターパスのより小さな近傍 $N_2(\beta/2)$ に属する．

8.5 ロングステップ法

ロングステップ法は，ステップサイズを長くとる方法であり，実際の計算プログラムによく使われる方法である (Marsten et al. (1990), McShane et al. (1989))．線形計画問題(P)と(D)のタイプ2の初期点 (x^0, y^0, z^0) ($Ax^0=b, A^Ty^0+z^0=c, x^0>0, z^0>0$) が既知であると仮定する．収束判定は，$x^{k^T}z^k \leq \epsilon$ とする．ベクトル v^k として，$\gamma\in[0, 1]$ に対して $\gamma(x^{k^T}z^k/n)e$ を使う．ステップサイズは，現在の点 (x^k, y^k, z^k) から実行可能領域の境界までのステップサイズ $\hat{\alpha}_P^k$ と $\hat{\alpha}_D^k$ (3章参照) を使い計算する．

Marsten et al. (1990) と McShane et al. (1989) の方法は，定数 $\theta\in(0, 1)$ を使い，

$$\alpha_P^k=\theta\hat{\alpha}_P^k,$$
$$\alpha_D^k=\theta\hat{\alpha}_D^k$$

とする．この方法は，$\hat{\alpha}_P^k$ または $\hat{\alpha}_D^k$ が ∞ となる場合に適用できない．$\hat{\alpha}_P^k$ または $\hat{\alpha}_D^k$ の少なくとも一つは理論的に有界であるので，

$$\hat{\alpha}^k=\min\{\hat{\alpha}_P^k, \hat{\alpha}_D^k\}$$

は有界である．したがって，

$$\alpha_P^k=\alpha_D^k=\theta\hat{\alpha}^k$$

とすれば，常に計算可能である．しかし，このステップサイズを使うアルゴリズム1の収束性については，v^k を定めるパラメータ γ の値をどのように設定しても，まだ不明である．

Kojima et al. (1990) は，大域的収束性を持つロングステップ法と，多項式オーダの大域的収束性を持つロングステップ法を提案した．大域的収束性を持つアルゴリズムでは，小さな正の定数 ω と α^* を固定し，各反復においてパラメータ γ を区間 [0, 0.99] から，パラメータ θ を区間 [0.01, 0.99] から任意にとり，ステップサイズを

$$c^T(x^k+\alpha_P^k\Delta x^k)-b^T(y^k+\alpha_D^k\Delta y^k)\leq(1-\omega)(c^Tx^k-b^Ty^k),\quad \alpha_P^k<\hat{\alpha}_P^k,\quad \alpha_D^k<\hat{\alpha}_D^k$$

をみたす任意の値，または

$$\alpha_P^k=\alpha_D^k=\begin{cases}\theta\hat{\alpha}^k & \text{if}\quad \hat{\alpha}^k\geq\alpha^*,\\ \theta(\hat{\alpha}^k)^2/\alpha^* & \text{if}\quad \hat{\alpha}^k<\alpha^*\end{cases}$$

に設定する．多項式オーダの大域的収束性を持つアルゴリズムでは，小さな正の定数 γ^* と α^* を固定し，各反復においてパラメータ γ を区間 $[\gamma^*, 0.5]$ から，パラメータ θ を区間 [0.5, 0.99] から任意にとり，ステップサイズを

$$\alpha_P^k=\alpha_D^k=\begin{cases}\theta\hat{\alpha}^k & \text{if}\quad \hat{\alpha}^k\geq\alpha^*,\\ \theta(\hat{\alpha}^k)^2/\alpha^* & \text{if}\quad \hat{\alpha}^k<\alpha^*\end{cases}$$

に設定する．タイプ 3 の初期点から始めれば，このアルゴリズムは $O(nL)$ 反復を必要とする．

8.6 ポテンシャル減少法

ポテンシャル減少法は，あるポテンシャル関数を減少させるようにステップサイズを決める方法である．Karmarkar (1984) のアルゴリズムもポテンシャル減少法である．Ye (1991a) と Freund (1991) は，$O(\sqrt{n}L)$ 反復のポテンシャル減少法をはじめて提案した．その方法は，主問題と双対問題の実行可能内点を生成するが，アルゴリズム 1 の枠組みには入らない主内点法である．

線形計画問題 (P) と (D) のタイプ 2 の初期点 (x^0, y^0, z^0) $(Ax^0=b, A^Ty^0+z^0=c, x^0>0, z^0>0)$ が既知であると仮定する．ただし，初期点でポテンシャル関数の値がある定数で抑えられる必要があるという意味で，タイプ 3 の初期点が必要な場合もある．収束判定は，$x^{k^T}z^k\leq\epsilon$ とする．ベクトル v^k の決め方は，ポテンシャル減少法の中にもさまざまな方法がある．単純な方法の一つは，$v^k=\gamma(x^{k^T}z^k/n)e$ (γ は定数) とする．ステップサイズは，一般に $\alpha_P^k=\alpha_D^k$ として，ポテンシャル関数を最小にする値とする．主双対内点法で使われる標準的なポテンシャル関数は，

$$g_\rho(x,z)=\frac{(x^Tz)^\rho(x^Tz/n)^n}{\prod_{i=1}^n x_iz_i},$$

またはその対数をとった

$$f_\rho(x,z)=\rho\ln x^Tz+n\ln(x^Tz/n)-\sum_{i=1}^n\ln x_iz_i$$

である．これらのポテンシャル関数は，Tanabe (1987a, 1987b) あるいは Todd and Ye (1990) により提案された．

Tanabe (1987a, 1987b) の提案したアルゴリズムは，各反復 k で $v^k=\gamma(x^{k^T}z^k/n)e$ としステップサイズを $\alpha_P^k=\alpha_D^k=\alpha$ として，ポテンシャル関数 $g_\rho(x^k+\alpha\Delta x^k, z^k+\alpha\Delta z^k)$ を最小にするパラメータ γ と α を二次元探索により求める．

Todd and Ye (1990) は，Kojima et al. (1989b) と Monteiro and Adler (1989a, 1989b) の

パス追跡法により生成される点列がポテンシャル関数 $f_\rho(x, z)$ $(\rho=\sqrt{n})$ を単調に減少させることを示し，$O(\sqrt{n}L)$ 反復のアルゴリズムであることを証明した．

Kojima et al. (1991d) は，$v^k=(n/(n+\sqrt{n}))(x^{k^T}z^k/n)e$ としステップサイズを $\alpha_P^k=\alpha_D^k=\alpha$ として，ポテンシャル関数 $f_\rho(x^k+\alpha\Delta x^k, z^k+\alpha\Delta z^k)$ $(\rho=\sqrt{n})$ を最小にするパラメータ α を一次元探索により求めるアルゴリズムを提案した．このアルゴリズムは $O(\sqrt{n}L)$ 反復を必要とする．ここで使われている探索方向は，スケーリングした空間におけるポテンシャル関数の最急降下方向を実行可能領域に射影した方向とみなすこともできる．

Kojima et al. (1991c) は，近傍とポテンシャル関数の両方を使ってステップサイズを決めるアルゴリズムを提案した．すなわち，近傍 $N_{\ln}(\beta)$ (β は定数) 内でポテンシャル関数 $f_\rho(x^k+\alpha\Delta x^k, z^k+\alpha\Delta z^k)$ を最小にするパラメータ α を一次元探索により求めるアルゴリズムである．このアルゴリズムは，β の値が小さいときにパス追跡法とみなすことができ，β の値が大きいときにポテンシャル減少法とみなすことができる．その論文では，広範囲のパラメータの設定に対してアルゴリズムの大域的収束性あるいは多項式オーダの収束性が証明されている．常に $v^k=0$ とするアフィンスケーリング法において，ステップサイズをポテンシャル関数を最小化する点により計算するアルゴリズムの収束性も証明されている．ただし，その反復回数は多項式オーダではない．

ポテンシャル減少法には，外ステップ-内ステップ (outer step - inner step) 法もある．この方法は，主内点法として Gonzaga (1991a, 1991b) が提案した．そのアルゴリズムの特徴は，外ステップで v^k を大幅に更新し，内ステップでは v^k を一定にしたままポテンシャル減少法をある条件 (たとえば $\| X_k z^k - v^k \| \leq \delta(x^{k^T}z^k/n)$, δ は定数) が成立するまで繰り返すことである．外ステップ-内ステップ法の一つをアルゴリズム1の枠組みに入れ説明する．第 k 反復で条件 $\| X_k z^k - v^k \| \leq \delta(x^{k^T}z^k/n)$ が成立したならば $v^k=0.5\,v^{k-1}$ とし，さもなければ $v^k=v^{k-1}$ とする．ステップサイズは，$\alpha_P^k=\alpha_D^k=\alpha$ として，ポテンシャル関数 $f_\rho(x^k+\alpha\Delta x^k, z^k+\alpha\Delta z^k)$ を最小にするパラメータ値とする．

8.7 外点法

いままで述べた方法は，すべてタイプ2または3の実行可能な初期点を必要とする．外点法はタイプ1の初期点から始まり，必ずしも実行可能でない点列を生成する方法である．収束判定は，5章で示したように，

$$\begin{aligned} & x^{k^T}z^k \leq \epsilon_1, \\ & \| Ax^k - b \| \leq \epsilon_2, \\ & \| A^T y^k + z^k - c \| \leq \epsilon_3, \\ & x^k \geq 0, \qquad z^k \geq 0, \end{aligned}$$

または

$$\|(x^k, y^k, z^k)\| \geq \omega$$

とする．ここで，$\epsilon_1, \epsilon_2, \epsilon_3$ は小さな正の定数であり，ω は大きな正の定数である．

Lustig (1991) は，元問題の人工問題を4章の方法により作成するとき，定数 ρ と σ を ∞ としたときの探索方向が元問題の外点法で計算される方向と一致することを証明した．彼は，人工問題の内点法において ρ と σ を ∞ にするという発想のもとに外点法を提案したが，その方法の理論的な収束性については何等得ていない．Lustig et al. (1991) は，外点法が効率よく大規模な問題を解くという計算実験の結果を報告した．

田辺 (1989) は, 各反復 k で $v^k=\gamma(x^{k^T}z^k/n)e$ としステップサイズを $\alpha_P^k=\alpha_D^k=\alpha$ として, 次の式で定義されるポテンシャル関数 $\bar{f}_\rho(x^k+\alpha\Delta x^k, y^k+\alpha\Delta y^k, z^k+\alpha\Delta z^k)$ を減少させるパラメータ γ と α を求めるアルゴリズムを提案した:

$$\bar{f}_\rho(x, y, z)=\rho\ln(x^Tz+\|Ax-b\|_1+\|A^Ty+z-c\|_1)+n\ln(x^Tz/n)-\sum_{i=1}^{n}\ln x_iz_i.$$

このアルゴリズムの収束性についてはまだ不明である.

Kojima et al. (1991a) は, 大域的に収束する外点法を提案した. そのアルゴリズムは, 第 k 反復で定数 $\gamma\in(0,1)$ に対して $v^k=\gamma(x^{k^T}z^k/n)e$ とし, 以下の条件をみたすようにステップサイズを求める:

$$\begin{aligned}
&x_i^kz_i^k\geq\delta_1x^{k^T}z^k/n,\\
&x^{k^T}z^k\geq\delta_2\|Ax^k-b\|,\\
&x^{k^T}z^k\geq\delta_3\|A^Ty^k+z^k-c\|,\\
&(x^k+\alpha_P^k\Delta x^k)^T(z^k+\alpha_D^k\Delta z^k)\leq(1-\delta_4)x^{k^T}z^k.
\end{aligned}$$

Kojima et al. (1991a) は, 上記の条件をみたす定数 $\delta_1, \delta_2, \delta_3, \delta_4$ とステップサイズ α_P^k, α_D^k が存在することを示した. このアルゴリズムでは, 元問題に実行可能内点が存在するとき有限回の反復で近似解を求め, 存在しないとき点列 $\{(x^k, y^k, z^k)\}$ が発散する.

9. 局所的収束性

この章では, 主双対内点法の局所的収束性について述べる. 主内点法の局所的収束性については, Iri and Imai (1986), Yamashita (1986), Tsuchiya and Tanabe (1990), Tsuchiya (1991) などに示されている. Tsuchiya (1991) は, 乗法的罰金法 (Iri and Imai (1986)) が非退化の仮定なしで二次収束することを証明した.

主双対内点法 (アルゴリズム 1) により実行可能点列 $\{(x^k, y^k, z^k)\}$ を生成し, 双対ギャップ (duality gap) 列 $\{x^{k^T}z^k\}$ が 0 に収束するとき,

$$\lim_{k\to\infty}\frac{x^{{k+1}^T}z^{k+1}}{x^{k^T}z^k}=0$$

が成立するならば, 双対ギャップ列が 0 に (局所的に) 超一次収束する, あるいはその内点法が超一次収束するといい, 正の整数 $l>0$ に対して

$$\limsup_{k\to\infty}\frac{x^{{(k+1)l}^T}z^{(k+1)l}}{(x^{{kl}^T}z^{kl})^2}<+\infty$$

が成立するならば双対ギャップ列が l 回の反復ごとに 0 に (局所的に) 二次収束する, あるいはその内点法が l 回の反復ごとに二次収束するという. $l=1$ のとき単に二次収束するという. また双対ギャップ列の Q オーダは,

$$\sup\left\{\sigma>1:\ \lim_{k\to\infty}\frac{x^{{k+1}^T}z^{k+1}}{(x^{k^T}z^k)^\sigma}=0\right\}$$

によって定義される. 二次収束すればQオーダは 2 以上であるが, Q オーダが 2 でも二次収束するとは限らない.

Kojima et al. (1991b) は, 主双対線形計画問題を一般化した非線形相補性問題に対する内点法の局所的収束性を調べた. 線形計画問題の場合について述べれば, 問題 (P) と (D) がそれぞ

れ唯一つの最適解 x^* と (y^*, z^*) を持つ場合に，ある種のパス追跡法が局所的に二次収束することを示した．

Zhang et al. (1990) は，アルゴリズム1により実行可能内点列 $\{(x^k, y^k, z^k)\}$ を生成するとき，以下の条件が成立すれば超一次収束することを証明した：

- ある最適解 (x^*, y^*, z^*) に対して $(x^k, y^k, z^k) \to (x^*, y^*, z^*)$.
- ある定数 $\gamma \in (0,1)$ とすべての k に対して，$X_k z^k \geq \gamma \left(\dfrac{x^{k^T} z^k}{n} \right) e$ が成立する．
- 0に収束する点列 $\{\sigma^k\}$ $(\sigma^k \in (0,1))$ と1に収束する点列 $\{\tau^k\}$ $(\tau^k \in (0,1))$ に対して $v^k = \sigma^k \left(\dfrac{x^{k^T} z^k}{n} \right) e$, $\alpha_P^k = \alpha_D^k = \tau^k \min\{\hat{\alpha}_P^k, \hat{\alpha}_D^k\}$ とする．

彼らは，さらに以下の条件が成立すれば二次収束することも証明した：

- $\sigma^k = O(x^{k^T} z^k)$ かつ $1 - \tau^k = O(x^{k^T} z^k)$.
- x^* は非退化な端点である．

Zhang and Tapia (1991) は上記において条件 $(x^k, y^k, z^k) \to (x^*, y^*, z^*)$ を $x^{k^T} z^k \to 0$ としても超一次収束することを示し，さらに二次収束に非退化条件が不必要であることを示した．Zhang and Tapia (1990) は，Zhang et al. (1990) の条件のもとで，大域的に $O(nL)$ の反復回数を必要とし局所的に超一次収束するアルゴリズムを提案した．

Mehrotra (1991) と Ye et al. (1991) は，別々にしかもほぼ同時期に Mizuno et al. (1991) のプレディクタ・コレクタ法が非退化の仮定なしで局所的に2反復（1回のプレディクタステップとコレクタステップ）ごとに二次収束することを証明した．Ye (1991b) は，そのプレディクタ・コレクタ法を改良し，Qオーダが2となる内点法を提案した．

10. アルゴリズムの計算複雑度

この章では，アルゴリズム全体の計算複雑度についてまとめる．アルゴリズム1は，方程式系 (3.2) を解くので，一反復ごとに $O(n^3)$ の基本演算を必要とする．したがって，$O(nL)$ 反復のアルゴリズムは $O(n^4 L)$ の計算量を必要とし，$O(\sqrt{n}L)$ 反復のアルゴリズムは $O(n^{3.5}L)$ の計算量を必要とする．

Karmarkar (1984) は，彼の提案した $O(nL)$ 反復の主内点法に，部分的に逆行列を更新する方法を導入することにより，全体として $O(n^{3.5})$ のアルゴリズムを提案した．Gonzaga (1988) と Vaidya (1990) は，Renegar (1988) の提案した $O(\sqrt{n}L)$ 反復の主内点法を改良し $O(n^3 L)$ のアルゴリズムを構築した．

主双対内点法も，探索方向の計算時に逆行列の部分的更新を利用することにより計算複雑度を下げられる．Kojima et al. (1989b) と Monteiro and Adler (1989a, 1989b) は，$O(\sqrt{n}L)$ 反復と $O(n^3 L)$ の計算量を必要とするパス追跡法を提案した．Mizuno (1990) は，$O(\sqrt{n}L)$ 反復と $O(n^3 L)$ の計算量を必要とする点列追跡法を提案した．

Bosch and Anstreicher (1990) は，Kojima et al. (1991d) のポテンシャル減少法に Goldstein-Armijo の規則と逆行列の部分的更新を利用し $O(n^3 L)$ のアルゴリズムを提案した．

11. おわりに

本論では，主双対内点法を一般的なスタイル（アルゴリズム1）で述べ，その枠組みでさまざまなアルゴリズムを説明した．実際にプログラミングして例題を解くときに，それらのアルゴリズムのなかでどれを使えばよいかという問題がある．このとき二つの面から判断する必要がある．一つは理論的に判断する方法であり，もう一つは標準的な例題を解いた経験から判断する方法である．

理論的な結果は，多くの場合に最悪の例題に対する評価である．したがって，理論的に計算複雑度の低い方法がすべての例題を効率よく解くわけではない．Mizuno et al. (1991) は，ある仮定のもとで期待的な反復回数を評価し，最悪の場合よりよい結果を得た．また，大域的な収束オーダだけでなく，局所的に超一次あるいは二次収束することも効率よいアルゴリズムの条件であろう．

標準的な例題を解いた数値実験の結果は，効率よいアルゴリズムの一つの判断材料である．しかし，新しい問題を解くときには，過去の経験で効率のよいアルゴリズムが，必ずしも効率よいとは限らない．ある種の決まったタイプの問題でデータの値のみ変わる問題を多く解く場合には，数値実験結果がよい指標となるであろう．しかし，現在効率よいといわれている Lustig et al. (1991) の方法は，理論的に収束性が不明であり，新しい問題に適用した場合にまったく収束しない可能性もある．したがって，理論的にある程度収束性が保証されているアルゴリズムの中で経験的に計算効率のよいアルゴリズムを採用することが得策と思われる．

謝　辞

本研究は部分的に文部省科学研究費一般研究（c）(03832017) および奨励研究 A (03740125) の補助を受け行われた．

参考文献

Adler, I., Karmarkar, N.K., Resende, M.G.C. and Veiga, G. (1989). An implementation of Karmarkar's algorithm for linear programming, *Math. Programming,* **44**, 297-335.

Barnes, E.R. (1986). A variation on Karmarkar's algorithm for solving linear programming problems, *Math. Programming,* **36**, 174-182.

Bosch, R.A. and Anstreicher, K.M. (1990). On partial updating in a potential reduction linear programming algorithm of Kojima, Mizuno, and Yoshise, Department of Operations Research, Yale University, New Haven, Connecticut.

Dikin, I.I. (1967). Iterative solution of problems of linear and quadratic programming, *Soviet Math. Dokl.,* **8**, 674-675.

Ding, J. and Li, T.-Y. (1991). A polynomial time predictor corrector algorithm for a class of linear complementarity problem, *SIAM Journal on Optimization,* **1**, 83-92.

Freund, R.M. (1991). Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function, *Math. Programming,* **51**, 203-222.

Goldfarb, D. and Todd, M.J. (1989). Linear programming, *Handbooks in Operations Research and Management Sciences, Volume 1, Optimization* (eds. G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd), 73-170, North-Holland, Amsterdam.

Gonzaga, C.C. (1988). An algorithm for solving linear programming programs in $O(n^3L)$ operations, *Progress in Mathematical Programming, Interior Point and Related Methods* (ed. N. Megiddo), 1-28, Springer, New York.

Gonzaga, C.C. (1991a). Large step path-following methods for linear programming, part I : barrier function method, *SIAM Journal on Optimization,* **1**, 268-279.

Gonzaga, C.C. (1991b). Large step path-following methods for linear programming, part II: potential reduction method, *SIAM Journal on Optimization,* **1**, 280-292.

伊理正夫 (1986). 『線形計画法』, 共立出版, 東京.

Iri, M. and Imai, H. (1986). A multiplicative barrier function method for linear programming, *Algorithmica,* **1**, 455-482.

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

Khachiyan, L.G. (1979). A polynomial algorithm in linear programming, *Soviet Math. Dokl.,* **20**, 191-194.

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

Kojima, M., Mizuno, S. and Yoshise, A. (1989b). A polynomial-time algorithm for a class of linear complementarity problems, *Math. Programming,* **44**, 1-26.

Kojima, M., Megiddo, N. and Mizuno, S. (1990). Theoretical convergence of large-step primal-dual interior point algorithms for linear programming, Research Report RJ 7872, IBM Thomas J. Watson Research Center, New York.

Kojima, M., Megiddo, N. and Mizuno, S. (1991a). A primal-dual exterior point algorithm for linear programming, Research Report RJ 8500, IBM Thomas J. Watson Research Center, New York.

Kojima, M., Megiddo, N. and Noma, T. (1991b). Homotopy continuation methods for nonlinear complementarity problems, *Math. Oper. Res.,* **16**, 754-774.

Kojima, M., Megiddo, N., Noma, T. and Yoshise, A. (1991c). A unified approach to interior point algorithms for linear complementarity problems, *Lecture Notes in Comput. Sci.,* Springer, New York.

Kojima, M., Mizuno, S. and Yoshise, A. (1991d). An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems, *Math. Programming,* **50**, 331-342.

Kojima, M., Mizuno, S. and Yoshise, A. (1991e). A little theorem of the big M in interior point algorithms, Research Reports on Information Sciences B-239, Department of Information Sciences, Tokyo Institute of Technology, Tokyo.

Kranich, E. (1991). Interior point methods for mathematical programming: a bibliography, Diskussionbeitrag Nr. 171, Wirtschaftwissenschaften und Operations Research, Fern Universität Hagen, Germany.

Lustig, I.J. (1991). Feasibility issues in a primal-dual interior point method for linear programming, *Math. Programming,* **49**, 145-162.

Lustig, I.J., Marsten, R.E. and Shanno, D.F. (1991). Computation experience with a primal-dual interior point method for linear programming, *Linear Algebra Appl.,* **152**, 191-222.

Marsten, R., Subramanian, R., Saltzman, M., Lustig, I. and Shanno, D. (1990). Interior point methods for linear programming: just call Newton, Lagrange, and Fiacco and McCormick!, *Interfaces,* **20**, 105-116.

McShane, K.A., Monma, C.L. and Shanno, D. (1989). An implementation of a primal-dual interior point method for linear programming, *ORSA Journal on Computing,* **1**, 70-83.

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

Mehrotra, S. (1991). Quadratic convergence in a primal-dual method, Tech. Report 91-15, Department of Industrial Engineering and Management Sciences, Northwestern University, Illinois.

水野眞治 (1989). 線形計画問題に対する内点法について,『数理計画法の最近の進歩と知的所有権』, 第1回 RAMP シンポジウム, 15-25.

Mizuno, S. (1989). A new polynomial time method for a linear complementarity problem, Tech. Report No. 16, Department of Management Science and Engineering, Tokyo Institute of Technology, Tokyo.

Mizuno, S. (1990). An $O(n^3L)$ algorithm using a sequence for a linear complementarity problem, *J. Oper. Res. Soc. Japan,* **33**, 66-75.

Mizuno, S., Yoshise, A. and Kikuchi, T. (1989). Practical polynomial time algorithms for linear complementarity problems, *J. Oper. Res. Soc. Japan,* **32**, 75-92.

Mizuno, S., Todd, M.J. and Ye, Y. (1991). On adaptive-step primal-dual interior-point algorithms for linear programming, Tech. Report No. 944, School of Operation Research and Industrial Engineering, College of Engineering, Cornell University, Ithaca, New York.

Monteiro, R.C. and Adler, I. (1989a). Interior path following primal-dual algorithms, part 1: linear programming, *Math. Programming,* **44**, 27-42.

Monteiro, R.C. and Adler, I. (1989b). Interior path following primal-dual algorithms, part 2: convex quadratic programming, *Math. Programming,* **44**, 43-66.

Monteiro, R.C., Adler, I. and Resende, M.G.C. (1990). A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension, *Math. Oper. Res.,* **15**, 191-214.

Renegar, J. (1988). A polynomial-time algorithm based on Newton's method for linear programming, *Math. Programming,* **40**, 59-94.

Tanabe, K. (1987a). Centered Newton method for linear programming and quadratic programming, *Proceedings of the 8th Mathematical Programming Symposium, Japan*, 131-152.

Tanabe, K. (1987b). Complementarity-enforcing centered Newton method for mathematical programming: global method, ISM Cooperative Research Report 5, 118-144.

田辺國士 (1989). Centered Newton method for linear programming: exterior point method, 統計数理, **37**, 146-148.

Todd, M.J. (1989). Recent developments and new directions in linear programming, *Mathematical Programming, Recent Developments and Applications* (eds. M. Iri and K. Tanabe), 109-157, Kluwer, London.

Todd, M.J. and Ye, Y. (1990). A centered projective algorithm for linear programming, *Math. Oper. Res.,* **15**, 508-529.

Tsuchiya, T. (1991). Quadratic convergence of Iri and Imai's algorithm for degenerate linear programming problems, Research Memo. 412, The Institute of Statistical Mathematics, Tokyo.

Tsuchiya, T. and Tanabe, K. (1990). Local convergence properties of new methods in linear programming, *J. Oper. Res. Soc. Japan,* **33**, 22-45.

Vaidya, P.M. (1990). An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations, *Math. Programming,* **47**, 175-202.

Yamashita, H. (1986). A polynomially and quadratically convergent method for linear programming, Tech. Report, Mathematical Systems Institute, Inc., Tokyo.

Ye, Y. (1991a). An $O(n^3L)$ potential reduction algorithm for linear programming, *Math. Programming,* **50**, 239-258.

Ye, Y. (1991b). On the Q-order of convergence of interior-point algorithms for linear programming, Tech. Report No. 91-17, Department of Management Sciences, The University of Iowa, Iowa City.

Ye, Y., Güler, O., Tapia, R.A. and Zhang, Y. (1991). A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming, Working Paper No. 91-14, The College of Business Administration, The University of Iowa, Iowa City.

Zhang, Y. and Tapia, R.A. (1990). A quadratically convergent polynomial primal-dual interior-point algorithm for linear programming, TR90-40, Department of Mathematical Sciences, Rice University, Houston, Texas.

Zhang, Y. and Tapia, R.A. (1991). Superlinear and quadratic convergence of primal-dual interior-point methods for linear programming revisited, TR91-27, Department of Mathematical Sciences, Rice University, Houston, Texas.

Zhang, Y., Tapia, R.A. and Dennis, J.E. (1990). On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms, TR90-6, Department of Mathematical Sciences, Rice University, Houston, Texas.

 Proceedings of the Institute of Statistical Mathematics Vol. 40, No. 1, 27-44 (1992)

A Primal-dual Interior Point Method for Linear Programming

Shinji Mizuno

(The Institute of Statistical Mathematics)

In this paper, we summarize various primal-dual interior point algorithms for linear programming. We first show a generic primal-dual interior point algorithm, which consists of 4 main steps ; finding an initial solution, convergence criteria, computing search directions and determining step sizes. Then the detail of each step is shown. We explain various primal-dual interior point algorithms in the framework of the generic algorithm and show the computational complexity of each algorithm. The algorithms are affine scaling algorithms, sequence following algorithms, path following algorithms, predictor-corrector algorithms, long-step algorithms, potential reduction algorithms and exterior point algorithms. We also show recent results of asymptotic behavior of primal-dual interior point algorithms.

Key words: Interior point method, linear programming, duality theorem, potential function, path of centers.