

# 線形計画問題に対する単体法の計算量と 強多項式アルゴリズム

東京工業大学 水野 眞治\*

Shinji Mizuno

Tokyo Institute of Technology

## 概要

本論では、Kitahara and Mizuno [4, 5] によって得られた、線形計画問題を解くときに単体法あるいは双対単体法で生成される異なる基底解の数の上界について説明する。その後、クラメル公式を使うことにより、上界がより具体的に表現できることを示す。その上界を使い、Mizuno [7] によって得られた単体法の計算量の結果を示す。その結果は、次のように述べることができる：解くべき補助問題がすべて非退化ならば、提案される単体法で線形計画問題の最適基底を得るか、最適基底が存在しないことが判定でき、その時に必要とされる総計算量がデータの大きさの多項式で抑えることができる。線形計画問題の係数行列が全ユニモジユラならば、提案される単体法は強多項式アルゴリズムとなる。

Keywords: 線形計画問題、単体法、多項式アルゴリズム、強多項式アルゴリズム。

## 1 はじめに

線形計画問題を解く方法として、Dantzig[1] の提案した単体法、Khachiyan の楕円体法 [3]、Karmarkar の内点法 [2] がよく知られている。楕円体法と内点法は、多項式アルゴリズムであるが、単体法については多項式アルゴリズムであるかわかっていない。実際、Dantzig のピボッティング規則（最小の縮約コストをもつ非基底変数を新しく基底変数とする規則）などを使う単体法については、多項式時間で解けない問題 (Klee-Minty [6]) が知られているので、多項式アルゴリズムとはいえない。

---

\* 大学院社会理工学研究科、152-8552 東京都目黒区大岡山 2-12-1, W9-58, E-mail: mizuno.s.ab@m.titech.ac.jp.

本論では、標準形の線形計画問題

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x}, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (1)$$

を対象とする。ここで、正の整数  $m$  と  $n$  に対して、 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{c} \in \mathbb{R}^n$  はデータであり、 $\mathbf{x} \in \mathbb{R}^n$  は変数ベクトルである。行列  $\mathbf{A}$  の各要素が整数であり、 $\text{rank } \mathbf{A} = m$  であるとする。また、行列  $\mathbf{A}$  の要素の最大絶対値を

$$A_{\max} = \{|a| \mid a \text{ は行列 } \mathbf{A} \text{ の要素}\}$$

とし、正方部分行列の行列式の最大絶対値を

$$\Delta_A = \max\{|\det \mathbf{D}| \mid \mathbf{D} \text{ は行列 } \mathbf{A} \text{ の正方部分行列}\}$$

と定義する。明らかに、 $A_{\max} \leq \Delta_A$  と  $\Delta_A \leq m! A_{\max}^m$  が成立する。線形計画問題 (1) を解くアルゴリズムが多項式アルゴリズムであるということは、解くときに必要とされる計算量（あるいは基本演算の回数）が  $m, n, L_A, L_b, L_c$  の多項式で抑えることができることを意味する。ここで、 $L_A, L_b, L_c$  はそれぞれ  $\mathbf{A}, \mathbf{b}, \mathbf{c}$  のサイズをあらわし、具体的には、

$$\begin{aligned} L_A &= \sum_{i=1}^m \sum_{j=1}^n \log(|a_{ij}| + 1), \\ L_b &= \sum_{i=1}^m \log(|b_i| + 1), \\ L_c &= \sum_{j=1}^n \log(|c_j| + 1) \end{aligned}$$

である。単体法では、各反復で必要とする計算量が  $m, n$  の多項式で抑えることができるので、反復回数が  $m, n, L_A, L_b, L_c$  の多項式で抑えることができれば多項式アルゴリズムとなる。

Tardos [8] は、 $\log(A_{\max})$  が  $m, n$  の多項式で抑えられる線形計画問題 (1) に対して、強多項式アルゴリズムを提案した。強多項式アルゴリズムとは、 $m, n$  の多項式で抑えられる計算量で問題を解くことができることを意味する。 $\log(A_{\max})$  が  $m, n$  の多項式で抑えられるならば、定義より行列  $\mathbf{A}$  のサイズ  $L_A$  も  $m, n$  の多項式で抑えられる。Tardos は、 $\mathbf{b}$  と  $\mathbf{c}$  のサイズ  $L_b$  と  $L_c$  が  $\mathbf{A}$  のサイズの多項式で抑えられる複数の補助線形計画問題を解くことにより、元の問題 (1) が解けることを示した。このとき、それぞれの補助問題を多項式アルゴリズムで解けば、各補助問題は  $m, n, L_A$  の多項式で抑えられる計算量で解くことができる。また、解かれる補助問題の総数も  $m, n$  の多項式で抑えることができるので、 $\log(A_{\max})$  が  $m, n$  の多項式で抑えられるならば、全計算量が  $m, n$  の多項式で抑えることができることになる。

Tardos のアルゴリズムは、補助問題を解くときに多項式アルゴリズムを使う必要があるため、単体法を使った場合には、強多項式アルゴリズムとは言えない。Tardos のアル

ゴリズムでは、ベクトル  $c$  のサイズ  $L_c$  を  $m, n, L_A$  の多項式で抑えた補助問題を複数解く基本アルゴリズムを使い、さらにベクトル  $b$  のサイズ  $L_b$  を  $m, n, L_A$  の多項式で抑えるための工夫を加えている。一方、Mizuno [7] は、 $\Delta_A$  が  $m, n$  の多項式で抑えられる問題 (1) に対して、Tardos の基本アルゴリズムに現れる補助問題を単体法で解くことにより、強多項式アルゴリズムが得られることを示した。

本論では、Kitahara and Mizuno [4, 5] によって得られた、問題 (1) を解く主単体法あるいは双対単体法で生成される異なる基底解の数の上界について説明し、クラメルの公式を使うことにより、上界がより具体的に表現できることを示す。その上界を使い、Mizuno [7] によって得られた単体法の計算量の結果を示す。その結果は、次のように述べることができる：解くべき補助問題がすべてが非退化ならば、提案される単体法で問題 (1) の最適基底を得るか、最適基底が存在しないことが判定でき、そのときに必要とされる総計算量が  $m, n, \Delta_A$  の多項式で抑えることができる。もし、行列  $A$  が全ユニモジュラ ( $\Delta_A = 1$ ) ならば、総計算量が  $m$  と  $n$  の多項式で抑えられるので、提案される単体法は強多項式アルゴリズムとなる。

本論では、ベクトル  $a$  に対して、 $\|a\|_1$  と  $\|a\|_\infty$  は、それぞれ  $a$  の  $l_1$  ノルムと  $l_\infty$  ノルムを表す。また、実数  $a$  に対して、 $\lceil a \rceil$  は  $a$  より小さくない最小の整数を表す。

## 2 単体法で生成される異なる基底解の数

この節では、標準形の線形計画問題 (1) あるいはその双対問題

$$\begin{aligned} \max \quad & b^T y, \\ \text{subject to} \quad & A^T y + z = c, \quad z \geq 0 \end{aligned} \quad (2)$$

を対象とし、Kitahara and Mizuno [4, 5] によって得られた、主単体法あるいは双対単体法で生成される異なる基底解の数の上界について説明する。その後、クラメルの公式 (Cramer's rule) を使い、その上界をより具体的に表現する。

Kitahara and Mizuno [5] によれば、線形計画問題 (1) を Dantzig の規則を使った主単体法で解くときに生成される異なった基底解の数は

$$nm \frac{\gamma}{\delta} \log(m \frac{\gamma}{\delta}) \quad (3)$$

で抑えられる (より正確には、整数への切り上げを使う必要があるが、簡単のため略している)。ここで、 $\delta$  と  $\gamma$  は、それぞれ問題 (1) の実行可能基底解の正の成分の最小値と最大値を表している。問題が非退化ならば、上の値 (3) は、単体法の反復回数の上界にもなっている。

上の結果から、 $\gamma/\delta$  が小さいならば、単体法により問題 (1) を効率的に解くことができることがわかる。例えば、線形計画問題の係数行列  $A$  が全ユニモジュラ (totally unimodular) ならば

$$\begin{aligned}\delta &\geq 1, \\ \gamma &\leq \|b\|_1\end{aligned}$$

が成立するので、(3) の値は、たかだか

$$nm\|b\|_1 \log(m\|b\|_1)$$

となる。したがって、ベクトル  $b$  の各成分の絶対値が大きくない整数ならば、単体法で効率よく解ける。しかし、ベクトル  $b$  の中に整数ではない実数値あるいは値の大きな整数があれば、単体法は多くの反復を必要とする可能性がある。

より一般の行列  $A$  の場合に、線形計画問題 (1) の基底解の各成分は、クラメルの公式を使うと、分数  $p/q$  で表され、分母  $q$  は行列  $A$  の正方部分行列式の値、分子  $p$  はその行列のある列をベクトル  $b$  に置き換えた行列式の値となる。したがって、

$$\begin{aligned}q &\leq \Delta_A, \\ p &\leq \Delta_A \|b\|_1\end{aligned}$$

が成立し、

$$\begin{aligned}\delta &\geq 1/\Delta_A, \\ \gamma &\leq \Delta_A \|b\|_1\end{aligned}$$

となる。このとき、(3) の値は、たかだか

$$nm\Delta_A^2 \|b\|_1 \log(m\Delta_A^2 \|b\|_1)$$

となる。もし、 $\Delta_A$  と  $\|b\|_1$  の値が小さいならば、単体法で効率よく線形計画問題 (1) を解くことができる。

同様に、Kitahara and Mizuno [4] によれば、線形計画問題 (1) を Dantzig の規則を使った双対単体法で解くときに生成される異なった基底解の数は

$$m^2 \frac{\gamma_D}{\delta_D} \log\left(m \frac{\gamma_D}{\delta_D}\right) \quad (4)$$

で抑えられる。この式の中で、 $\delta_D$  と  $\gamma_D$  は、それぞれ双対実行可能基底解  $(y, z)$  における  $z$  の要素の中の最小値と最大値を表している。双対問題 (2) では、 $\gamma_D \leq \Delta_A \|c\|_1$  かつ  $\delta_D \geq 1/\Delta_A$  となるので、その上界 (4) は、たかだか

$$m^2 \Delta_A^2 \|c\|_1 \log(m\Delta_A^2 \|c\|_1) \quad (5)$$

となる。もし双対問題 (7) が非退化ならば、双対単体法の反復回数も同様に (5) で抑えられる。

これまでの議論により、係数行列  $A$  の行列式の最大絶対値  $\Delta_A$  が小さな場合（たとえば  $A$  が全ユニモジュラ行列である場合）には、制約式の右辺ベクトル  $b$  の各要素の絶対値が小さい整数ならば単体法で、目的関数の係数ベクトル  $c$  の各要素の絶対値が小さい整数ならば双対単体法により、問題 (1) と (2) を効率よく解くことができることがわかる。しかし、 $b$  と  $c$  の要素が整数とは限らない場合、あるいは絶対値が大きな整数である場合などには、単体法で効率よく解けるとは言えない。このような場合には、Mizuno [7] が示したように、Tardos の基本アルゴリズムと単体法を組み合わせることにより、効率よく解くことが可能となることを次節以降で示す。

### 3 アルゴリズム

この節では、Mizuno [7] によって提案された、標準形の線形計画問題 (1) とその双対問題 (2) を解くアルゴリズムを説明する。そのアルゴリズムは、Tardos [8] によって提案された基本アルゴリズムを使い、その中の補助問題を解くときに楕円体法 [3] あるいは内点法 [2] ではなく、単体法を使う。

集合  $N = \{1, 2, \dots, n\}$  とする。  $N$  の部分集合  $K$  とその補集合  $\bar{K} = N - K$  に対して、必要ならば順序を入れ替えたのちに、行列  $A$  とベクトル  $c$ 、 $x$  を

$$A = (A_K, A_{\bar{K}}), c = \begin{pmatrix} c_K \\ c_{\bar{K}} \end{pmatrix}, x = \begin{pmatrix} x_K \\ x_{\bar{K}} \end{pmatrix}$$

と分解する。このとき、問題 (1) は

$$\begin{aligned} \min & \quad c_K^T x_K + c_{\bar{K}}^T x_{\bar{K}}, \\ \text{subject to} & \quad A_K x_K + A_{\bar{K}} x_{\bar{K}} = b, (x_K, x_{\bar{K}}) \geq 0 \end{aligned} \quad (6)$$

と表すことができる。

線形計画問題 (1) の各変数  $x_i (i \in N)$  には、任意の最適解で常に 0 になるもの、常に正になるものと、それ以外のものがある。いま、常に 0 となる変数  $x_i$  の添え字  $i$  の集合を

$$\bar{K}^* = \{i \mid \text{任意の最適解 } x^* \text{ で } x_i^* = 0 \text{ となる } i \in N\}$$

とし、その補集合を  $K^* = N - \bar{K}^*$  とする。このとき、問題 (1) の最適解の集合は、

$$O = \{(x_{K^*}, x_{\bar{K}^*}) \mid A_{K^*} x_{K^*} = b, x_{\bar{K}^*} = 0, x_{K^*} \geq 0\}$$

とあらわされる。したがって、集合  $\bar{K}$  が  $\bar{K}^*$  の部分集合ならば、問題 (6) において、 $\mathbf{x}_{\bar{K}} = \mathbf{0}$  とした問題を解くことにより、元の問題を解くことが可能となる。

提案するアルゴリズムでは、次のような補助問題

$$\begin{aligned} \min \quad & \bar{\mathbf{d}}_K^T \mathbf{x}_K, \\ \text{subject to} \quad & \mathbf{A}_K \mathbf{x}_K = \mathbf{b}, \mathbf{x}_K \geq \mathbf{0} \end{aligned} \quad (7)$$

を最大で  $n$  個解く。ここで、 $\bar{\mathbf{d}}_K$  は元問題のベクトル  $\mathbf{c}$  を使って求めたベクトルであり、 $\bar{\mathbf{d}}_K$  の各成分は絶対値が  $n^2 \Delta_A$  以下の整数である。問題 (7) を主問題とすると、その双対問題は、

$$\begin{aligned} \max \quad & \mathbf{b}^T \mathbf{y}, \\ \text{subject to} \quad & \mathbf{A}_K^T \mathbf{y} + \mathbf{z}_K = \bar{\mathbf{d}}_K, \mathbf{z}_K \geq \mathbf{0} \end{aligned} \quad (8)$$

とあらわされる。ここで、 $\mathbf{z}_K$  はスラック変数のベクトルである。

提案するアルゴリズムでは、最適基底解の基底変数の添え字集合  $B \subset K$  と双対問題の最適基底解

$$\mathbf{y} = (\mathbf{A}_B^T)^{-1} \bar{\mathbf{d}}_B, \mathbf{z}_B = \mathbf{0}, \mathbf{z}_{\bar{B}} = \bar{\mathbf{d}}_{\bar{B}} - \mathbf{A}_K^T (\mathbf{A}_B^T)^{-1} \bar{\mathbf{d}}_B$$

を双対単体法により計算する。ここで、 $\bar{B} = K - B$  である。このとき、主問題の最適解は、

$$\mathbf{x}_B = (\mathbf{A}_B)^{-1} \mathbf{b}, \mathbf{x}_{\bar{B}} = \mathbf{0}$$

と計算できる。

以上の準備のもとで、アルゴリズムは次のよう述べることができる。

アルゴリズム

ステップ0: ベクトル  $\mathbf{c}$  の各要素が整数でかつ  $\|\mathbf{c}\|_\infty \leq n^2 \Delta_A$  ならば、線形計画問題 (1) を双対単体法で解き、停止する。さもなければ、 $K = N$  として、ステップ1へ行く。

ステップ1: ベクトル  $\mathbf{c}_K$  を部分空間  $\{\mathbf{x}_K | \mathbf{A}_K \mathbf{x}_K = \mathbf{0}\}$  に射影したベクトルを  $\mathbf{c}'_K$  とする、すなわち、 $\mathbf{c}'_K = (\mathbf{I} - \mathbf{A}_K^T (\mathbf{A}_K \mathbf{A}_K^T)^{-1} \mathbf{A}_K) \mathbf{c}_K$  と計算する。もし  $\mathbf{c}'_K = \mathbf{0}$  ならば停止し、さもなければステップ2へ行く。

ステップ2: ベクトル  $\mathbf{d}_K = (n^2 \Delta_A / \|\mathbf{c}'_K\|_\infty) \mathbf{c}'_K$ ,  $\bar{d}_i = \lceil d_i \rceil$  ( $i \in K$ ) とし、 $\bar{\mathbf{d}}_i$  ( $i \in K$ ) からなるベクトルを  $\bar{\mathbf{d}}_K$  とする。次の線形計画問題

$$\begin{aligned} \min \quad & \bar{\mathbf{d}}_K^T \mathbf{x}_K, \\ \text{subject to} \quad & \mathbf{A}_K \mathbf{x}_K = \mathbf{b}, \mathbf{x}_K \geq \mathbf{0} \end{aligned} \quad (9)$$

とその双対問題

$$\begin{aligned} \max \quad & \mathbf{b}^T \mathbf{y}, \\ \text{subject to} \quad & \mathbf{A}_K^T \mathbf{y} + \mathbf{z}_K = \bar{\mathbf{d}}_K, \mathbf{z}_K \geq \mathbf{0} \end{aligned} \quad (10)$$

を考える。双対問題の実行可能領域  $F = \{(\mathbf{y}, \mathbf{z}_K) | \mathbf{A}_K^T \mathbf{y} + \mathbf{z}_K = \bar{\mathbf{d}}_K, \mathbf{z}_K \geq \mathbf{0}\}$  が空であるかチェックする。もし空ならば停止し、さもなければ、初期実行可能基底解  $(\mathbf{y}^0, \mathbf{z}_K^0) \in F$  を求め、線形計画問題 (9) を Dantzig の規則を使った双対単体法で解く。もし、問題 (9) が非有界 (無限解を持つ) ならば停止し、さもなければ最適基底解  $(\bar{\mathbf{y}}, \bar{\mathbf{z}}_K)$  と最適基底に対する添え字集合  $B \subset K$  を求める。集合  $J = \{i | d_i - \mathbf{a}_i^T \bar{\mathbf{y}} \geq n\Delta_A, i \in K\}$  を求める。ここで、ベクトル  $\mathbf{a}_i$  は行列  $\mathbf{A}_K$  の第  $i$  列を表している。この集合  $J$  に属する添え字  $i$  をすべて  $K$  から除き、ステップ 1 へ戻る。

## 4 アルゴリズムの性質

前節で説明したアルゴリズムでは、元の線形計画問題 (1) においてベクトル  $\mathbf{c}$  の各要素が整数でかつ  $\|\mathbf{c}\|_\infty \leq n^2 \Delta_A$  ならば、その問題 (1) を双対単体法で直接解く。このとき、単体法で生成される異なった基底解の数は、2 節での結果から、 $m, n, \Delta_A$  の多項式で抑えることができる。

もしステップ 2 で実行可能領域  $F$  が空ならば、 $\mathbf{c}'_K, \mathbf{d}_K, \bar{\mathbf{d}}_K$  の定義から、双対問題 (2) が実行不能であることがわかる。このとき、主問題 (1) は非有界である (無限解を持つ) か、あるいは実行不能である。2 段階単体法を使えば、その第 1 段階で、実行可能領域  $F$  が空であるかどうか判定することができ、空でない場合には初期の実行可能基底解  $(\mathbf{y}^0, \mathbf{z}_K^0) \in F$  を求めることができる。双対問題が非退化ならば、第 2 段階で、初期解  $(\mathbf{y}^0, \mathbf{z}_K^0) \in F$  から双対単体法を実行することにより、最適基底解  $(\bar{\mathbf{y}}, \bar{\mathbf{z}}_K)$  とそのときの最適基底  $B \subset K$  をみつけるか、あるいは問題が非有界であることを判定できる。最適基底が見つかった場合には、主問題の最適基底解  $\bar{\mathbf{x}}_K$  も簡単に計算できる。ステップ 2 における  $J$  の定義と、相補性条件から、任意の  $i \in J$  に対して、 $\bar{x}_i = 0$  となる。したがって、アップデートした新しい  $K$  に対する  $\bar{\mathbf{x}}_K$  は、次の反復における問題 (9) の実行可能解となる。

ステップ 2 において、双対問題 (10) が非有界であることが判明した場合には、主問題 (9) は実行不能である。また、主問題の最適基底解  $\bar{\mathbf{x}}_K$  は次の反復における主問題 (9) の実行可能解なので、主問題が実行不能となりえるのは、第 1 反復のみである。したがっ

て、この場合には、元の主問題 (1) が実行不能であることがわかる。

Mizuno [7] は、Tardos [8] の結果を使うことにより、任意の  $i \in J$  に対して、線形計画問題 (1) の任意の最適基底解で  $x_i^* = 0$  となることを示し (下の Lemma 1)、各反復で集合  $J$  が少なくとも一つの要素を持つことを示した (下の Lemma 2)。アルゴリズムにおいて、ステップ 1 で  $c'_K = \mathbf{0}$  が成立し停止した場合には、条件  $\mathbf{A}_K \mathbf{x}_K = \mathbf{b}$  と  $\mathbf{x}_K \geq \mathbf{0}$  をみたく任意の実行可能解  $\mathbf{x}_K^*$  に  $\mathbf{x}_K^* = \mathbf{0}$  を付け加えた解が問題 (1) の最適解となる。前の反復で求めた主問題の最適基底解  $\bar{\mathbf{x}}_K$  がその条件をみたす解であるので、問題 (1) の最適基底と最適解を得ることができる。もし最初の反復で  $c'_K = \mathbf{0}$  となったならば、問題 (1) の任意の実行可能解が最適解となる。したがって、目的関数の係数ベクトル  $\mathbf{c}$  を絶対値が  $n^2 \Delta_A$  以下となる要素からなる任意のベクトルに置き換えてから、問題 (1) を双対単体法で解くことにより、最適基底を得ることができる。

ここで、Mizuno [7] と Tardos [8] による結果を二つの Lemma として述べる。

**Lemma 1** (Lemma 1.1 in Tardos [8] and Lemma 1 in [7]) 問題 (9) と (10) の最適基底解をそれぞれ  $\bar{\mathbf{x}}_K$  と  $(\bar{\mathbf{y}}, \bar{\mathbf{z}}_K)$  とする。このとき、問題

$$\begin{aligned} \min \quad & \mathbf{d}_K^T \mathbf{x}_K, \\ \text{subject to} \quad & \mathbf{A}_K \mathbf{x}_K = \mathbf{b}, \mathbf{x}_K \geq \mathbf{0} \end{aligned} \quad (11)$$

の任意の最適解  $\mathbf{x}_K^*$  において、次のこと

$$d_i - \mathbf{a}_i^T \bar{\mathbf{y}} \geq n \Delta_A \text{ implies } x_i^* = 0 \text{ for any } i \in K,$$

が成立する。ここで、 $\mathbf{a}_i$  は行列  $\mathbf{A}_K$  の第  $i$  列のベクトルを表している。

**Lemma 2** (Lemma 1.2 in Tardos [8] and Lemma 2 in [7]) アルゴリズムの各反復のステップ 2 で定義された集合  $J$  は少なくとも一つの要素  $i$  を含む。

以上の議論をまとめると、前節で提案したアルゴリズムは、各反復で集合  $K$  の要素数が少なくとも 1 つ減るので、たかだか  $n$  回の反復で終了する。また、集合  $K$  から除かれる要素は、最適解において値が 0 となる要素のみなので、アルゴリズムがステップ 1 で  $c'_K = \mathbf{0}$  となって停止した場合には、最適基底と最適解を求めることができる。ステップ 2 で停止した場合には、線形計画問題 (1) は実行不能であるか非有界であり、最適解を持たないことがわかる。補助問題では目的関数の係数ベクトル  $\bar{\mathbf{d}}_K$  の各成分の絶対値が  $n^2 \Delta_A$  以下の整数であるので、双対問題が非退化であるとき、問題 (7) を解く双対単体法の反復回数は、式 (5) と  $\|\mathbf{c}\|_1 \leq n^3 \Delta_A$  より、たかだか

$$m^2 n^3 \Delta_A^3 \log(mn^3 \Delta_A^3)$$



で抑えられる。また、各反復で2段階単体法を適用するので、解くべき補助問題の数は、たかだか  $2n$  である。この結果より、係数行列  $A$  が全ユニモジユラ ( $\Delta_A = 1$ ) である場合には、反復回数が  $n$  の多項式で抑えられるので、強多項式アルゴリズムとなる。

## 5 おわりに

本論では、単体法で線形計画問題 (1) を解くときに、生成される異なる解の数あるいは反復回数について議論した。その結果は、Kitahara and Mizuno [4, 5] によって得られた上界にクラメル公式を適用することにより得ることができる。また、Mizuno [7] によって提案されたように、Tardos [8] の基本アルゴリズムにおいて、補助問題を単体法で解いた場合の計算量を求めた。その結果、線形計画問題の係数行列  $A$  が全ユニモジユラならば、そのアルゴリズムが強多項式となることを示した。なお、単体法で各問題が確実に解けることを保証するためには、解くべき補助問題がすべて非退化であることを仮定する必要がある。理論的に少し強い条件であるが、実際に単体法で問題を解くときに巡回に陥ることはめったにないので、実用的には大きな問題とはならないと思われる。また、辞書的順序などを導入すれば、巡回を避けることが可能であるが、その場合の反復回数については不明であり、今回の結果を直接適用することはできない。

謝辞：本研究の一部は、科学研究費基盤研究 (A)26242027 の補助を受けて行われました。

## 参考文献

- [1] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, New Jersey, 1963.
- [2] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica*, 4, 373–395 (1984)
- [3] L. G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Math. Dokl.*, 20, 191–194 (1979)
- [4] T. Kitahara and S. Mizuno, On the number of solutions generated by the dual simplex method, *Oper. Res. Lett.*, 40, 172–174 (2012)
- [5] T. Kitahara and S. Mizuno, A bound for the number of different basic solutions generated by the simplex method, *Math. Pro.*, 137, 579–586 (2013)
- [6] V. Klee and G.J. Minty, How good is the simplex method, in: O. Shisha (Ed.),

- Inequalities III, Academic Press, New York, 1972, 159-175.
- [7] S. Mizuno: A strongly polynomial simplex method for totally unimodular LP, Technical Report 2014-3, Department of Industrial Engineering and Management, Tokyo Institute of Technology (2014).
- [8] É. Tardos, A strongly polynomial algorithm to solve combinatorial linear programs, *Oper. Res.*, 34, 250–256 (1986)