数理解析研究所講究録
第 680 巻 1989 年 117-126

Elementary Inductive Proofs for Linear Programming

by

Komei FUKUDA 福田公明
Tomomi MATSUI 松井知己

Department of Information Sciences
Tokyo Institute of Technology
Oh-okayama, Meguro-ku
Tokyo 152, Japan

(January 1989)

1 The Fundamental Theorem

Let A be an $m \times n$ matrix, where $m \geq 1$ and $n \geq 2$. Let E be the index set of the columns of matrix A, and f, g be two distinct members of E. Here we consider the standard form linear program:

(1.1)	(P)	maximize	x_f		
(1.2)		subject to	$A\ x$	$=$	$0,$
(1.3)			x_g	$=$	$1,$
(1.4)			x_j	$\geq$	$0, \ \forall j \in E \setminus \{f, g\}.$

For any $J \subseteq E \setminus \{f,g\}$, the *deleted subproblem* (P$\backslash J$) can be obtained from (P) by dropping the nonnegative constraints $x_j \geq 0$ for all $j \in J$, and the *contracted subproblem* (P$/J$) can be obtained from (P) by adding the constraints $x_j = 0$ for all $j \in J$. For any $j \in J$, we abbreviate $\{j\}$ by j. Given two disjoint subsets J_1 and J_2 of $E \setminus \{f,g\}$, we define the subproblem ((P$\backslash J_1)/J_2$):

	((P$\backslash J_1)/J_2$)	maximize x_f	
		subject to (1.2), (1.3), and	
(1.5)		$x_j = 0$	$j \in J_2,$
(1.6)		$x_j \geq 0$	$j \notin J_1 \cup J_2.$

By using the artificial variables, we can transform the problem ((P$\backslash J_1)/J_2$) to a linear program with the form (1.1), (1.2), (1.3), and (1.4). Thus, in the rest of this section, we consider more general linear program defined by (1.1), (1.2), (1.3), (1.5), and (1.6). We define the deleted and contracted subproblems of this general form linear program in the same way.

A vector x is said to be *feasible* if it satisfies the constraints (1.2), (1.3), (1.5), and (1.6) If a linear program has a feasible solution, then it is called *feasible*, else it is called *infeasible*. For any linear program, we will refer to following three situations as *characters*:

1. it has an optimal solution,
2. it is infeasible,
3. there exists a vector x satisfying that (1.2), (1.5), (1.6), and

(P)		(P\j)		
		OPT	INF*	INF
(P/j)	OPT	OPT	INF* or OPT	——
	INF	OPT or INF	INF* or INF	INF
	INF*	——	INF*	INF and INF*

Table 1: The characters of (P\j), (P/j), and (P).

$$x_g = 0, \tag{1.7}$$
$$x_f > 0. \tag{1.8}$$

In the situation 2, a linear programming does not have an optimal solution. In the situation 3, a linear programming with a feasible solution also does not have an optimal solution. Each of these three situations is denoted by *OPT, INF,* and *INF**.

The following lemma shows the relations between the characters of linear programs P, (P\j), and (P/j). Originally, this lemma was proved in a more general setting of oriented matroid programming, see Fukuda [1].

Lemma 1 *Let P be a linear program, and* $j \in E \setminus \{f, g\}$ *be an index corresponding to a variable with nonnegative constraint* $x_j \geq 0$. *When* H_1 *is a character of (P\j), and* H_2 *is a character of (P/ j), either* H_1 *or* H_2 *is a character of (P). More precisely, the following statements hold.*

1. *If (P\j) is INF, then P is also INF.*

1*. *If (P/j) is INF*, then P is also INF*.*

2. *If both (P\j) and (P/j) are OPT, then P is OPT.*

3. *If (P\j) is OPT and (P/j) is INF, then P is either OPT or INF.*

3*. *If (P/j) is OPT and (P\j) is INF*, then P is either OPT or INF*.*

4. *If (P\j) is OPT, then (P/j) is not INF*.*

4*. *If (P/j) is OPT, then (P\j) is not INF.*

5. *If (P\j) is INF* and (P/j) is INF, then P is either INF or INF*.*

The Lemma 1 is represented by the Table 1, and it implies the following theorem.

Theorem 2 *A linear program has at least one of the character of OPT, INF, and INF*.*

Proof. Let k be the cardinality of the index set of nonnegative constraints. We will prove this theorem by the induction on k.

1. First, we will show that a linear program P, such that the cardinality of index set of nonnegative constraints is equal to 0, satisfies the statement of the theorem. If $k = 0$, then $J_1 \cup J_2 = E \setminus \{f, g\}$. Assume that a linear program P has no optimal solution and P is not INF. Let x be a feasible solution of P. Then there exists a feasible solution x' of P satisfies that $x_f < x'_f$. The vector $x' - x$ satisfies the conditions of INF* of P. Thus P is INF*.

2. Now we will show that the theorem holds for $k = r + 1$ under the assumption that the theorem holds for all linear program satisfying $k \leq r$. Let P be a linear program such

that the cardinality of index set of nonnegative constraints is equal to $r+1$. Let j be an index of nonnegative constraint. Then from the assumption, the two subproblems $(\mathrm{P}\backslash j)$ and (P/j) has at least one of the characters of OPT, INF, and INF*. From the lemma 1, the linear program P has at least one of the characters of OPT, INF, and INF*. □

2 Dictionaries

Here, we give the definition of dictionaries. For any $J \subseteq E$, x_J denotes the subvector of x indexed by J, and A_J denotes the submatrix of A consisting of columns indexed by J. In the rest of this paper, we assume the followings.

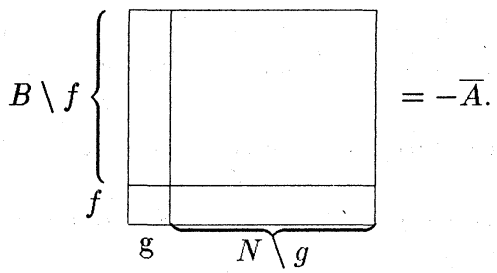
Assumption 1 *There exists a index subset B of E satisfying that $|B| = m$, $f \in B \not\ni g$, and the submatrix A_B is nonsingular.*

The above assumption implies that $n \geq m+1$. In the rest of this paper, we consider the linear programs defined by (1.1), (1.2), (1.3), and (1.4).

If B and N are subsets of the index set E, satisfying that: $|B| = m$, $f \in B \not\ni g$, $N = E \setminus B$, and the submatrix A_B is nonsingular, then we can transform the linear system (1.2) as:

(2.1) $$x_B = -A_B^{-1} A_N x_N.$$

We refer to this system (2.1) as *dictionary* $\mathcal{D}$. Each index in B is called *basic*, and each index in N is called *non-basic*. The *coefficient matrix of* $\mathcal{D}$, denoted by $\overline{A}$, is the matrix $A_B^{-1}A_N$. We consider $\overline{A}$ as the matrix whose rows (columns) are indexed by B (N), and $\overline{A} = (\overline{a}_{ij} \,:\, i \in B,\ j \in N)$. For any dictionary $\mathcal{D}$, we identify the dictionary (linear systems (2.1)) with the following matrix:



For any dictionary $\mathcal{D}$, a *basic solution with respect to* $\mathcal{D}$, is a vector indexed by E which satisfies that:

$$\begin{aligned} x_j &= 0, & \forall j \in N \setminus g, \\ x_g &= 1, & \\ x_i &= -\overline{a}_{ig}, & \forall i \in B, \end{aligned}$$

Clearly, the basic solution x satisfies the linear system (2.1). For any index $j \in N$, we denote the column of $\overline{A}$ corresponding to the index j by $\overline{A}_j$. For any index $i \in B$, we denote the row of $\overline{A}$ corresponding to the index i by ${}_i\overline{A}$.

Now we define special three dictionaries, called *terminal dictionaries.* Let J_1 and J_2 be disjoint index subsets of $E \setminus \{f, g\}$. A dictionary $\mathcal{D}$ satisfying that:

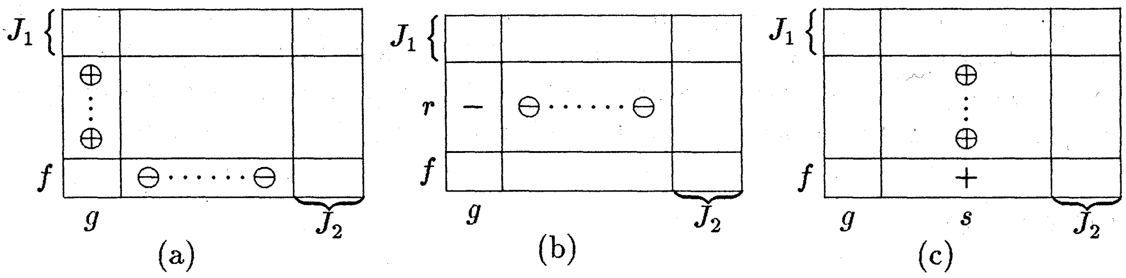


Figure 1: Terminal dictionaries.

(2.2)
$$J_1 \subseteq B,\ J_2 \subseteq N,$$
$$-\bar{a}_{ig} \geq 0,\ {}^\forall i \in B \setminus f,$$
$$-\bar{a}_{fj} \leq 0,\ {}^\forall j \in N \setminus g,$$

is an *optimal dictionary of* $((\mathrm{P} \setminus J_1)/J_2)$, (see Figure 1.a). A dictionary $\mathcal{D}$ is an *inconsistent dictionary of* $((\mathrm{P} \setminus J_1)/J_2)$, if it satisfies (2.2) and there exists an index $r \in B \setminus (J_1 \cup f)$ satisfying that;

$$-{}_r\bar{A} \leq 0,$$
$$-\bar{a}_{rg} < 0,$$

(see Figure 1.b). A dictionary $\mathcal{D}$ is an a *dual inconsistent dictionary of* $((\mathrm{P} \setminus J_1)/J_2)$, if it satisfies (2.2) and there exists an index $s \in N \setminus (J_2 \cup g)$ satisfying that;

$$-\bar{A}_s \geq 0,$$
$$-\bar{a}_{fs} > 0,$$

(see Figure 1.c). From the above definition, any dictionary $\mathcal{D}$ is an optimal dictionary of $((\mathrm{P}\setminus(B \setminus f))/(N \setminus g))$. In figures, the symbol $\oplus$ denotes a nonnegative number, $+$ denotes a positive number, $\ominus$ denotes a nonpositive number, and $-$ denotes a negative number.

From the definition of dictionary, we can easily verify the following Lemma.

Lemma 3 *For any linear program P, and two disjoint subsets* J_1, J_2 *of* $E \setminus \{f, g\}$, *the following statements hold.*

1. *If there exists an optimal dictionary* $\mathcal{D}$ *of* $((P\setminus J_1)/J_2)$, *then the basic solution of* $\mathcal{D}$ *is an optimal solution of* $((P\setminus J_1)/J_2)$.
2. *If there exists an inconsistent dictionary of* $((P\setminus J_1)/J_2)$, *then* $((P\setminus J_1)/J_2)$ *is INF.*
3. *If there exists a dual inconsistent dictionary of* $((P\setminus J_1)/J_2)$, *then* $((P\setminus J_1)/J_2)$ *is INF*.*

3 The Fundamental Theorem in Dictionary Form

In this section, we strengthen the Lemma 1 by applying the notion of dictionaries. The case 1 and 1* of Lemma 1 can be strengthened easily as follows.

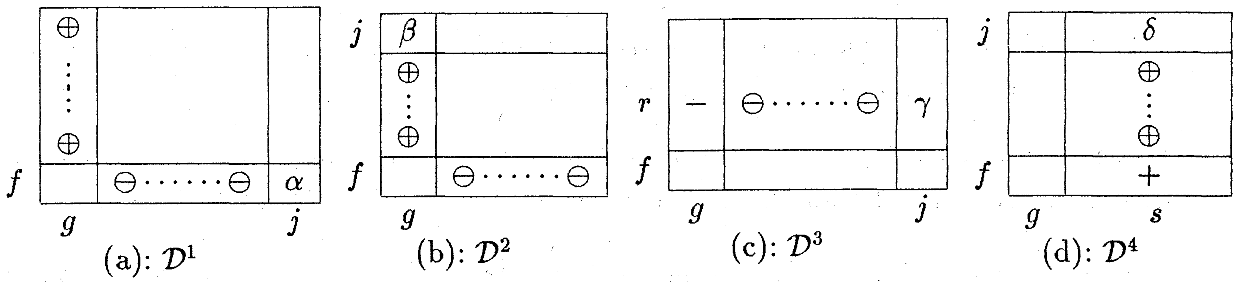


Figure 2: Terminal dictionaries of (P\j) and (P/j)

Claim 4 *Let P be a linear program and j be an index in $E\backslash\{f,g\}$. Then the following statements hold.*

1. *If $\mathcal{D}$ is an inconsistent dictionary of (P\j), then it is also an inconsistent dictionary of (P).*
2. *If $\mathcal{D}$ is a dual inconsistent dictionary of (P/j), then it is also a dual inconsistent dictionary of (P).*

Now we strengthen the other cases of Lemma 1. Given a index $j \in E \setminus \{f,g\}$, we consider the following four special dictionaries.

1. A dictionary $\mathcal{D}$ satisfying that $j \in N \setminus g$ and $\mathcal{D}$ is an optimal dictionary of (P/j), (see Figure 2.a). In this case, we denote $-\overline{a}_{fj}$ by α for simplicity.
2. A dictionary $\mathcal{D}$ satisfying that $j \in B \setminus f$ and $\mathcal{D}$ is an optimal dictionary of (P\j), (see Figure 2.b). In this case, we denote $-\overline{a}_{jg}$ by β for simplicity.
3. A dictionary $\mathcal{D}$ satisfying that $j \in N \setminus g$ and $\mathcal{D}$ is an inconsistent dictionary of (P/j), (see Figure 2.c). In this case, we denote $-\overline{a}_{rj}$ by γ for simplicity, where r is the row index of $\mathcal{D}$ which implies the inconsistency.
4. A dictionary $\mathcal{D}$ satisfying that $j \in B \setminus f$ and $\mathcal{D}$ is a dual inconsistent dictionary of (P\j), (see Figure 2.d). In this case, we denote $-\overline{a}_{js}$ by δ for simplicity, where s is the column index of $\mathcal{D}$ which implies the dual inconsistency.

Then the case 2, 3, 3*, and 5 of Lemma 1 can be strengthened as follows.

Lemma 5 *(c.f. Fukuda [1])*

Let P be a linear program and j be an index in $E \setminus \{f,g\}$. Assume that $\mathcal{D}$ is a terminal dictionary of (P\j), j is a basic index of $\mathcal{D}$, $\mathcal{D}'$ is a terminal dictionary of (P/j), and j is a non-basic index of $\mathcal{D}'$. Then either D or D' is a terminal dictionary of (P). More precisely, Claim 4 and the following statements hold.

1. *If (P) has an optimal dictionary $\mathcal{D}^1$ of (P/j) and an optimal dictionary $\mathcal{D}^2$ of (P\j), then either $\alpha \leq 0$ or $\beta \geq 0$.*
2. *If (P) has an optimal dictionary $\mathcal{D}^1$ of (P/j) and a dual inconsistent dictionary $\mathcal{D}^4$ of (P\j), then either $\alpha \leq 0$ or $\delta \geq 0$.*
3. *If (P) has an optimal dictionary $\mathcal{D}^2$ of (P\j) and an inconsistent dictionary $\mathcal{D}^3$ of (P/j), then either $\beta \geq 0$ or $\gamma \leq 0$.*

4. *If (P) has an inconsistent dictionary $\mathcal{D}^3$ of (P/j) and a dual inconsistent dictionary $\mathcal{D}^4$, of $(P\backslash j)$, then either $\gamma \leq 0$ or $\delta \geq 0$.*

Proof. For each dictionary $\mathcal{D}^k$ (for $k = 1,2,3,4$), we denote the basic solution, the basic index set, and the non-basic index set of $\mathcal{D}^k$, by x^k, B^k, and N^k.

1. It is obvious that x^1 is optimal to $((\mathrm{P}\backslash B^1)/j)$. If $\alpha > 0$, then there exists a vector d indexed by E, satisfying that $d_j = 1$, $d_f > 0$, and $x^1 + d \in \Omega(\mathrm{P}\backslash B^1)$. It implies that $x^1_f < (x^1 + d)_f$. If $\beta < 0$, then $x^2_j < 0$, and $x^1_f \leq x^2_f$. It implies that there exists $0 < \lambda < 1$, satisfying that $x' = \lambda(x^1 + d) + (1 - \lambda)x^2$, $x'_j = 0$, and $x^1_f < x'_f$. Since $x^1 + d \in \Omega(\mathrm{P}\backslash B^1)$, $x^2 \in \Omega(\mathrm{P}\backslash B^1 \cup j)$, and $x'_j = 0$, it is obvious that $x' \in \Omega(\mathrm{P}\backslash B^1/j)$. It contradicts with the optimality of x^1 to $((\mathrm{P}\backslash B^1)/j)$.
2. Let d be the vector defined above which corresponds to $\mathcal{D}^1$. If $\delta < 0$, there exists a vector d' indexed by E, satisfying that $d'_j < 0$, $d'_f > 0$, and $x^1 + d' \in \Omega(\mathrm{P}\backslash(B^1 \cup j))$. Then it implies that there exists $0 < \lambda < 1$ satisfying that $x' = x^1 + \lambda d + (1 - \lambda)d'$, $x'_j = 0$, and $x^1_f < x'_f$. Since $x^1 + d \in \Omega(\mathrm{P}\backslash B^1)$, $x^1 + d' \in \Omega(\mathrm{P}\backslash(B^1 \cup j))$, and $x'_j = 0$, it is obvious that $x' \in \Omega(\mathrm{P}\backslash B^1/j)$. It contradicts with the optimality of x^1 to $(\mathrm{P}\backslash B^1/j)$.
3. It can be proved in the same way as in case 2.
4. From the existence of $\mathcal{D}^3$, it is obvious that $(\mathrm{P}\backslash(B^3 \setminus r)/j)$ is infeasible. If $\gamma > 0$, then there exists a vector d indexed by E, satisfying that $d_j = 1$, $d_r > 0$, $d_g = 0$, $Ad = 0$, and $d_k = 0$, $\forall k \in N^3 \setminus j$. It implies that therer exists $\lambda > 0$ such that $x' = x^3 + \lambda d$, $x'_r = 0$, $x'_j > 0$, and $x' \in \Omega(\mathrm{P}\backslash(B^3 \setminus r))$. If $\delta < 0$, then there exists a vector d' indexed by E, satisfying that $d'_s = 1$, $d'_j < 0$, $d'_g = 0$, $Ad = 0$, and $d_k \geq 0$, $\forall k \in E \setminus j$. It implies that there exists $\lambda' > 0$ such that $x'' = x' + \lambda' d'$, $x''_j = 0$, $x''_g = 1$, and $x''_k \geq x'_k$, $\forall k \in E \setminus j$. It follows that $x' \in \Omega(\mathrm{P}\backslash(B^3 \setminus r)/j)$, and it is a contradiction. □

The Lemma 5 implies the following theorem.

Theorem 6 *Under the assumption 1, every linear program P has at least one of the three terminal dictionaries.*

Proof. We will prove this theorem by induction on $n + m$. From the assumptions, it is clear that $m \geq 1$, $n \geq 2$, and $n \geq m + 1$.

1. If $n + m = 3$, then the linear program has only one dictionary whose corresponding coefficient matrix is a scalar. From the definition, the dictionary is a trivial optimal dictionary.
2. Now we show that the theorem holds for $n + m = r + 1$ under the assumption that the theorem holds for $n + m \leq r$. From the assumption 1, there exists a dictionary $\mathcal{D}$ of P. Let B (N) be the basic (non-basic) index set of $\mathcal{D}$. Let $\overline{A}$ be the coefficient matrix of $\mathcal{D}$. First, we consider the trivial case such that $B = \{f\}$, or $N = \{g\}$. In this case, each dictionary is row or column vector. Then it is clear that each dictionary is one of the three terminal dictionaries. Next, we consider the case

that both $B \setminus f$ and $N \setminus g$ are non-empty. If the coefficient matrix $\overline{A}$ satisfies $\overline{a}_{jk} = 0$, $\forall j \in B \setminus f$, $\forall k \in N \setminus g$, then it is clear that the dictionary $\mathcal{D}$ is one of the three terminal dictionaries. Now we assume that there exists a index $j \in B \setminus f$, and $k \in N \setminus g$, satisfying that $\overline{a}_{jk} \neq 0$. Let $\tilde{A}$ be the submatrix of the coefficient matrix $\overline{A}$ whose rows are indexed by $B \setminus j$ and columns are indexed by N. Then we can construct a linear program P' as follows:

$$
\begin{array}{llrcl}
\text{(P')} & \text{maximize} & x_f & & \\
& \text{subject to (1.3) and} & & & \\
& & [\tilde{A}\ I] \quad x & = & 0, \\
(3.1) & & x_j & \geq & 0, \quad \forall j \in E \setminus \{f, g, j\},
\end{array}
$$

where the variable vector x is indexed by $E \setminus j$. From the definition of P', it is clear that P' satisfies the assumption 1. It follows that there exists a terminal dictionary of P'. Let B' be the basic index set of a terminal dictionary of P'. Then it is obvious that the submatrix $[\overline{A}\ I]_{B' \cup j}$ of $[\overline{A}\ I]$ is non-singular. From the definition, $[\overline{A}\ I]_{B' \cup j} = A_B^{-1} A_{B' \cup j}$. The non-singularity of $[\overline{A}\ I]_{B' \cup j}$ and A_B implies that $A_{B' \cup j}$ is also non-singular. Then there exists a dictionary $\mathcal{D}'$ of P with basic index set $B' \cup j$. It is clear that $\mathcal{D}'$ is a terminal dictionary of P$\setminus j$. From the assumption that $\overline{a}_{jk} \neq 0$, it is clear that by pivoting on (j, k), we can obtain a new dictionary such that the index j is non-basic and k is basic. Let $\tilde{A}'$ be the submatrix of the new dictionary whose rows are indexed by $(B \setminus j) \cup k$, and columns are indexed by $N \setminus k$. Then we can construct a linear program P" as follows:

$$
\begin{array}{llrcl}
\text{(P")} & \text{maximize} & x_f & & \\
& \text{subject to (1.3), (3.1), and} & & & \\
& & [\tilde{A}'\ I] \quad x & = & 0,
\end{array}
$$

where the variable vector x is indexed by $E \setminus j$. Then there exists a terminal dictionary of P". Let B'' be the basic index set of a terminal dictionary. We can easily prove that the matrix $A_{B''}$ is nonsingular in the same way with P'. The nonsingularity of $A_{B''}$ implies that there exists a dictionary $\mathcal{D}''$ of P whose basic index set is B''. Then $\mathcal{D}''$ is a terminal dictionary of P$/j$. The Lemma 5 implies that either D' or D'' is a terminal dictionary of P. □

4 The Criss-Cross Method

In this section we describe the criss-cross method for solving a linear program. (Terlaky [2],[3] and Wang [4]).

In the following algorithm, we maintain a 0-1 vector L indexed by $E \setminus \{f, g\}$, and dictionary $\mathcal{D}$. The vector L does not affect the performance of the algorithm, but in the next section, the finiteness of the algorithm can be proved by using the vector L.

In the following algorithm, we identify the index subset $E \setminus \{f, g\}$ with the set of integer numbers $\{1, 2, 3, \cdots, |E| - 2\}$. Given a 0-1 vector L indexed by $E \setminus \{f, g\}$, we define the operation $+$ as follows. For any index $j \in E \setminus \{f, g\}$, $L{+}j$ is a 0-1 vector indexed by $E \setminus \{f, g\}$, satisfying that:

$$\begin{aligned}(L+j)_k &= 0, \quad {}^\forall k < j,\\ (L+j)_j &= 1,\\ (L+j)_k &= L_k, \quad {}^\forall k > j.\end{aligned}$$

The criss-cross method

Step0. Set $\mathcal{D}$ be any dictionary.

Set L be the zero vector indexed by $j \in E \setminus \{f, g\}$.

Step1. Let r be the minimum index satisfying that:

$r \in B \setminus f,\ -\overline{a}_{rg} < 0,$ or

$r \in N \setminus g,\ -\overline{a}_{fr} > 0.$

If r does not exists, then stop.

(The current dictionary is an OPT dictionary.)

Step2. Let s be the minimum index satisfying that:

$r \in B \setminus f,\ s \in N \setminus g,\ -\overline{a}_{rs} > 0,$ or

$r \in N \setminus g,\ s \in B \setminus f,\ -\overline{a}_{sr} < 0.$

If s does not exists, then stop.

(If $r \in B \setminus f$, the current dictionary is an inconsistent dictionary.)

(If $r \in N \setminus g$, the current dictionary is a dual inconsistent dictionary.)

Step3. Pivot on (r, s) and set the resulting dictionary as $\mathcal{D}$.

Set $t := \max\{r, s\}$.

Set $L := L+t$.

Go to Step1.

The index t chosen in Step3 is called the *key pivot index*. In the above algorithm, it is clear that if the algorithm terminates, one of the terminal dictionaries is obtained. In the next section, we show the finiteness of the algorithm.

5 Discussion of the Algorithm

In this section, we show the finiteness of the criss-cross method, and discuss about the flexibility of the algorithm.

At first, we define the *lexicographic ordering*. Let L and L' be the vectors indexed by $\{1, 2, \cdots, |E| - 2\}$. Then the vector L is *lexicographically greater* than L', if there exists an index k satisfying that:

$$\begin{aligned}&L_j = L'_j, \ {}^\forall j > k,\\ &L_k > L'_k.\end{aligned}$$

Given a 0-1 vector L indexed by $\{1, 2, \cdots, |E| - 2\}$, it is obvious from the definition that if $L_j = 0$, then $L+j$ is lexicographically greater than L.

Lemma 7 *The criss-cross method terminates in finite steps.*

Proof. Now we show that the vector L increases monotonically in the sense of lexicographic ordering. For this, it suffices to show that at each iteration, the vector L (before updating) satisfies $L_t = 0$ for the key pivot index t. Suppose that $L_t = 1$ at an iteration k. In the rest of this proof, the index t denotes the key pivot index at iteration k. Let $\overline{A}$ be the coefficient matrix, and B (N) be the basic (non-basic) index set at iteration k. Let J_1 and J_2 be the index subset such that:

$$J_1 = \{j \in E \setminus \{f, g\} : j \in B,\ j \leq t\} \cup f,$$
$$J_2 = \{j \in E \setminus \{f, g\} : j \in N,\ j \leq t\} \cup g.$$

Now we define the submatrix $\tilde{A}$ of $\overline{A}$ whose rows are indexed by J_1, and columns are indexed by J_2. Then $-\tilde{A}$ is a dictionary of the following linear program:

$$(\text{P'}) \qquad \begin{array}{llcl} \text{maximize} & x_f & & \\ \text{subject to} & [\tilde{A}\ I]\ x & = & 0, \\ & x_g & = & 1, \\ & x_j & \geq & 0, \quad {}^{\forall} j \in E', \setminus\{f, g\}, \end{array}$$

where $E' = J_1 \cup J_2$, x is a variable vector indexed by E', and I is the identity matrix whose columns are indexed by J_1. From the definition of the algorithm, it is clear that if $t \in B$, then $-\tilde{A}$ is one of the terminal dictionaries of the linear program (P'$\backslash j$) and if $t \in N$, then $-\tilde{A}$ is one of the terminal dictionaries of the linear program (P'$/j$).

From the assumption that $L_t = 1$ at iteration k, there exists an iteration before k at which t is also chosen as the key pivot index. Let k' be the last such iteration. Let $\overline{A}'$ be the coefficient matrix, and B' (N') be the basic (non-basic) index set at iteration k'. Then it is clear that the key pivot index chosen at each iteration between k' and k is lower than t. It implies that $B \setminus J_1 \subseteq B'$, and $N \setminus J_2 \subseteq N'$. Let $\tilde{A}'$ be the submatrix of $\overline{A}'$ whose rows are indexed by $B' \cap (J_1 \cup J_2)$, and columns are indexed by $N' \cap (J_1 \cup J_2)$. Then it is clear that, if $t \in B'$, then $-\tilde{A}'$ is one of the three terminal dictionaries of the linear program (P'$\backslash j$) and if $t \in N'$, then $-\tilde{A}'$ is one of the three terminal dictionaries of the linear program (P'$/j$).

If $t \in B$, then it is obvious that $t \in N'$. It follows that $-\tilde{A}$ is a terminal dictionary of (P'$\backslash j$), and $-\tilde{A}'$ is a terminal dictionary of (P'$/ j$). Then Lemma 5 implies that $-\tilde{A}$ or $-\tilde{A}'$ is one of the terminal dictionaries of P'. It contradicts with that t is chosen as a pivot element at iteration k and k'. If $t \in N$, then we can show the contradiction in the same way.

Thus, for each iteration the 0-1 vector L increases monotonically in the sense of lexicographic ordering. The finiteness of the number of 0-1 vectors indexed by $E \setminus \{f, g\}$ implies the finiteness of the criss-cross method. □

Now we discuss the flexibility of the criss-cross method. In the criss-cross method, the pivot elements are chosen uniquely. However, we can relax the uniqueness of the selection rule of pivot elements without violating the finiteness of the algorithm. In section 4, we assigned the integer numbers $\{1, 2, \cdots, |E| - 2\}$ to the indices in $E \setminus \{f, g\}$ before starting the algorithm. The relaxation of the selection rule of pivot elements is given by permuting the integer numbers at the entrance of step 1 at each iteration. The original algorithm can be seen as the trivial permutation is used at the entrance of step 1 at each iteration. To maintain the finiteness of the algorithm, each permutation needs to satisfy some properties described below.

Let L be the 0-1 vector indexed by the integer numbers $\{1, 2, \cdots, |E| - 2\}$ at the entrance of step 1 at an iteration. An index (integer number) $j \in \{1, 2, \cdots, |E| - 2\}$ is called *0-labeled with respect to* L when $L_j = 0$. A *0-interval with respect to* L is a subset of $\{1, 2, \cdots, |E| - 2\}$ which consists of consecutive 0-labeled integer numbers. The proof of lemma 7 does not depend on the order of the elements in any 0-labeled interval. Then it is clear that for each 0-interval, any permutation of integer numbers does not violate the proof of finiteness. Thus we can permute the integer numbers inside of any 0-interval maintaining the finiteness of the algorithm.

References

[1] K. Fukuda. *Oriented Matroid Programming.* Ph.D. Thesis, University of Waterloo, Waterloo, 1982.

[2] T. Terlaky. A convergent criss-cross method. *Math. Oper. und Stat. ser. Optimization*, 16, 683–690, 1985.

[3] T. Terlaky. A finite criss-cross method for oriented matroids. *J. Combin. Theory, Ser. B*, 42, 319–327, 1987.

[4] Z. Wang. A conformal elimination free algorithm for oriented matroid programming. *Chinese Annals of Mathematics 8 B*, 1, 1987.