On the Finiteness of the Criss-Cross Method

by

Komei FUKUDA*
Graduate School of Systems Management
The University of Tsukuba
Otsuka, Bunkyo-ku
Tokyo 112, Japan

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

(April 1989, revised September 1989)

Abstract. In this short paper, we prove the finiteness of the criss-cross method by showing a certain binary number of bounded digits associated with each iteration increases monotonically. This new proof immediately suggests the possibility of relaxing the pivoting selection in the criss-cross method without sacrificing the finiteness.

Key Words: linear programming. simplex method, finite pivoting rules.

1 The Criss-Cross Method

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

$$(P) \qquad \text{maximize} \quad x_f \tag{1.1}$$

$$\text{subject to} \quad A\,x = 0, \tag{1.2}$$

$$x_g = 1, \tag{1.3}$$

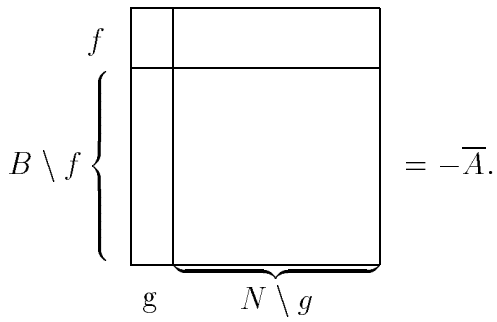
$$x_j \geq 0, \quad {}^{\forall}j \in E - \{f, g\}. \tag{1.4}$$

A vector x is said to be *feasible* if it satisfies the constraints (1.2), (1.3), and (1.4). If a linear program has a feasible solution, then it is called *feasible*, otherwise it is called *infeasible*. For any linear program, we will refer to following three situations as *characters*:

*Supported by Grant-in-Aids for Co-operative Research (63490010) of The Ministry of Education, Science and Culture

1) it has an optimal solution, 2) it is infeasible, 3) there exists a vector x satisfying (1.2), (1.4), $x_g = 0$, and $x_f > 0$. In the situation 3, a linear program is called *dual-infeasible.* Clearly, if a linear program is infeasible or dual-infeasible, it has no optimal solution. The criss-cross method, originally developed by Zionts [6] and extended to the setting of oriented matroid by Terlaky [3] [4] and Wang [5], can be considered as a finite procedure to decide which one of characters a given linear program has. Before describing the criss-cross method, we must have few definitions.

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. A *basis* B is an index subset satisfying the conditions that $|B| = m$, $f \in B \not\ni g$, and the submatrix A_B is nonsingular. In the rest of this paper, we assume that there exists at least one basis. For any basis B, we can transform the linear system (1.2) as $x_B = -A_B^{-1} A_N x_N$, where N is the *non-basis* $E - B$. We refer to this system as a *dictionary.* The *coefficient matrix* of a dictionary, 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 (respectively, N), i.e., $\overline{A} = (\overline{a}_{ij} \ : \ i \in B, \ j \in N)$. We often identify a dictionary with the following tableau:



$$= -\overline{A}.$$

Now we define special three dictionaries, called *terminal dictionaries.* A dictionary $\mathcal{D}$ is called *optimal*, if it satisfies $-\overline{a}_{ig} \geq 0$, $^\forall i \in B - f$, and $-\overline{a}_{fj} \leq 0$, $^\forall j \in N - g$, (see Figure 1.a). If $\mathcal{D}$ is optimal, then the basic solution of $\mathcal{D}$ is an optimal solution of P. A dictionary $\mathcal{D}$ is called *inconsistent*, if there exists an index $r \in B - f$ satisfying the conditions that: $-\overline{a}_{rg} < 0$, and $-\overline{a}_{rj} \leq 0$, $^\forall j \in N - g$, (see Figure 1.b). A dictionary $\mathcal{D}$ is called *dual-inconsistent*, if there exists an index $s \in N - g$ satisfying the conditions that: $-\overline{a}_{fs} > 0$, and $-\overline{a}_{is} \geq 0$, $^\forall i \in B - f$, (see Figure 1.c). The inconsistency and the dual-inconsistency of a dictionary implies the infeasibility and the dual-infeasibility of P, respectively.

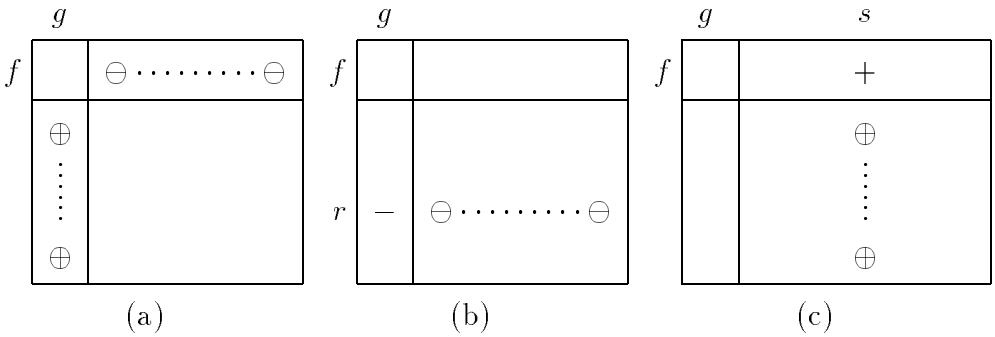


Figure 1: Terminal dictionaries.

Now we describe the criss-cross method. In our description of the algorithm, we maintain a 0-1 vector L indexed by $E-\{f,g\}$, and dictionary $\mathcal{D}$. The vector L is actually redundant for the description of the criss-cross method, but, as we will see in the next section, the finiteness of the algorithm will be proved by using the vector L. In the criss-cross method, we must assign a linear ordering on the index subset $E-\{f,g\}$ by identifying $E-\{f,g\}$ with the set of integer numbers $\{1,2,3,\cdots,|E|-2\}$. Given a 0-1 vector L indexed by $E-\{f,g\}$, and any index $j \in E-\{f,g\}$, $L+j$ is the 0-1 vector indexed by $E-\{f,g\}$, satisfying:

$$(L+j)_k = \begin{cases} 0, & \text{if } k < j, \\ 1, & \text{if } k = j, \quad (k \in E-\{f,g\}). \\ L_k, & \text{if } k > j, \end{cases}$$

The criss-cross method

Step 0. Set $\mathcal{D}$ be any dictionary.

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

Step 1. Let r be the minimal index satisfying:

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

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

If r does not exist, then stop (the current dictionary is optimal).

Step 2. Let s be the minimal index satisfying:

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

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

If s does not exist, then stop (the current dictionary is inconsistent if $r \in B - f$, and it is dual-inconsistent if $r \in N - g$).

Step 3. Pivot on (r, s) and reset the resulting dictionary as $\mathcal{D}$.

Set $L := L + \max\{r, s\}$, and go to step 1.

The Fig. 2 shows a small example. We call the index $t = \max\{r, s\}$ chosen in step 3 the *key pivot index*. In the above algorithm, it is clear that, if the algorithm terminates, one of the terminal dictionaries is obtained. Terlaky [3] [4] proved the finiteness of the algorithm by showing it cannot cycle. Here we prove it by showing the algorithm makes an explicit improvement at each iteration.

Theorem 1 *In the criss-cross method, the vector L increases monotonically in the sense of lexicographic ordering, and hence it terminates in finite steps.*

(L is *lexicographically greater* than L', if there exists an index k satisfying the conditions that $L_k > L'_k$, and $L_j = L'_j$, $^\forall j > k$.)

To prove this theorem, it suffices to show that at each iteration, the vector L (before updating) satisfies $L_t = 0$ for the key pivot index t. We will show it in the next section.

2 Proof of the finiteness

We define the following four types of dictionaries as *almost-terminal dictionaries with respect to j*.

1. A dictionary $\mathcal{D}^1$ satisfying the conditions that $j \in N - g$, the submatrix of $\mathcal{D}^1$ indexed by B and $N - j$ is optimal, and $-\overline{a}_{fj} > 0$, (see Figure 3.a).
2. A dictionary $\mathcal{D}^2$ satisfying the conditions that $j \in B - f$, the submatrix of $\mathcal{D}^2$ indexed by $B - j$ and N is optimal, and $-\overline{a}_{jg} < 0$, (see Figure 3.b).
3. A dictionary $\mathcal{D}^3$ satisfying the conditions that $j \in N - g$, the submatrix of $\mathcal{D}^3$ indexed by B and $N - j$ is inconsistent, and $-\overline{a}_{rj} > 0$ where r is an index in $B - f$ giving rise to the inconsistency, (see Figure 3.c).

$$
\begin{array}{lrcrcrcrcrcl}
\text{maximize} & x_f & & & & & & & & & & \\
\text{subject to} & x_f & - & 3x_1 & - & 5x_2 & & & + & 25x_g & = & 0 \\
& & & x_1 & + & x_2 & + & x_3 & - & 7x_g & = & 0 \\
& & & x_1 & + & x_2 & + & x_4 & - & 4x_g & = & 0 \\
& & & 3x_1 & + & 4x_2 & + & x_5 & - & 18x_g & = & 0 \\
& & & & & & & & & x_1,\ x_2,\ x_3,\ x_4,\ x_5 & \geq & 0
\end{array}
$$

The initial value of the sequence $L = (L_5, L_4, L_3, L_2, L_1) = (0,\ 0,\ 0,\ 0,\ 0)$.

	x_g	x_1	x_2
x_f	-25	3	5
x_3	7	-1^*	-1
x_4	4	-1	-1
x_5	18	-3	-4

$(r, s) = (1, 3)$

$\max(r, s) = 3$

$L := L + 3 = (0,\ 0,\ 1,\ 0,\ 0)$

	x_g	x_3	x_2
x_f	-4	-3	2
x_1	7	-1	-1^*
x_4	-3	1	0
x_5	-3	3	-1

$(r, s) = (2, 1)$

$\max(r, s) = 2$

$L := L + 2 = (0,\ 0,\ 1,\ 1,\ 0)$

	x_g	x_3	x_1
x_f	10	-5	-2
x_2	7	-1	-1
x_4	-3	1^*	0
x_5	-10	4	1

$(r, s) = (4, 3)$

$\max(r, s) = 4$

$L := L + 4 = (0,\ 1,\ 0,\ 0,\ 0)$

	x_g	x_4	x_1
x_f	-5	-5	-2
x_2	4	-1	-1
x_3	3	1	0
x_5	2	4	1

Figure 2: An example.

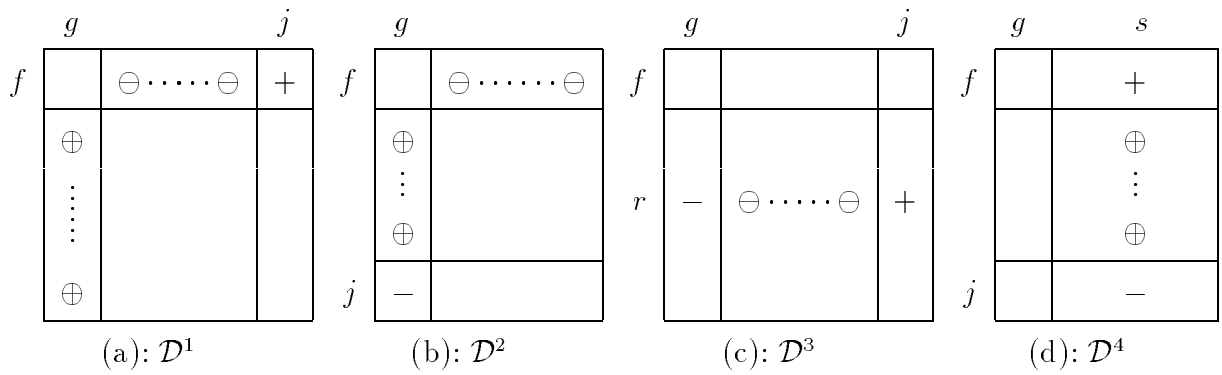


Figure 3: Almost terminal dictionaries

4. A dictionary $\mathcal{D}^4$ satisfying the conditions that $j \in B - f$, the submatrix of $\mathcal{D}^4$ indexed by $B - j$ and N is dual inconsistent, and $-\overline{a}_{js} < 0$ where s is an index in $N - g$ giving rise to the dual inconsistency, (see Figure 3.d).

For the proof of Theorem 1, the following lemma plays an important role.

Lemma 2 *(c.f. Bland [1], Theorem 5.1) Let P be a linear program and j be an index in $E - \{f, g\}$. Then P has at most one type of almost terminal dictionaries with respect to j.*

To prove the above lemma, we define subproblems of a linear program. For any $J \subseteq E - \{f, g\}$, the *deleted subproblem* $(P \backslash J)$ is defined to be the linear program obtained from (P) by dropping the nonnegative constraints $x_j \geq 0$ for all $j \in J$, and the *contracted subproblem* (P/J) the one 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 $E - \{f, g\}$, we define the subproblem $(P \backslash J_1 / J_2)$:

$$
\begin{array}{lll}
(P \backslash J_1 / J_2) & \text{maximize} \quad x_f & \\
& \text{subject to (1.2), (1.3), and} & \\
& \qquad x_j = 0 & j \in J_2, \qquad (2.1) \\
& \qquad x_j \geq 0 & j \notin J_1 \cup J_2. \qquad (2.2)
\end{array}
$$

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). We define the deleted and contracted subproblems of this general form linear program in the same way. For any linear program P', $\Omega(P')$ denotes the set of all feasible solutions.

Proof of Lemma 2. (see Fukuda and Matsui [2]). For each almost terminal dictionary $\mathcal{D}^i$ (for $i = 1,2,3,4$), we denote the coefficient matrix, the basic solution, the basic index set, and the non-basic index set of $\mathcal{D}^i$, by $\overline{A}^i$, x^i, B^i, and N^i. For each dictionary $\mathcal{D}^i$ $(i = 1,3)$, we define the vector d^i indexed by E as follows:

$$d_k^i = \begin{cases} -\overline{a}_{kj}^i & k \in B^i \cup f \\ 1 & k = j \\ 0 & k \in (N^i \cup g) - j. \end{cases}$$

For dictionary $\mathcal{D}^4$, we define d^4 be the vector indexed by E as follows:

$$d_k^4 = \begin{cases} -\overline{a}_{ks}^i & k \in B^4 \cup f \\ 1 & k = s \\ 0 & k \in (N^4 \cup g) - s. \end{cases}$$

The existence of $\mathcal{D}^3$ implies that there exists $\widetilde{x^3}$ satisfying the conditions that $\widetilde{x^3} = x^3 + \lambda d^3$, $\lambda > 0$, $\widetilde{x^3}_r = 0$, and $\widetilde{x^3}_j > 0$.

In the followings, we show that for each pair of almost terminal dictionaries, at most one of them exists.

1. Assume both $\mathcal{D}^1$ and $\mathcal{D}^3$ exist. Then the linear program (P/j) has an optimal solution x^1 and it is infeasible. It is impossible.

2. Assume both $\mathcal{D}^2$ and $\mathcal{D}^4$ exist, then the linear program $(P\backslash j)$ has an optimal solution x^2 and feasible solution $x^2 + d^4$. Then $x_f^2 < (x^2 + d^4)_f$ and it is a contradiction.

3. Assume both $\mathcal{D}^1$ and $\mathcal{D}^2$ exist. It is obvious that x^1 is optimal to $(P\backslash B^1\ /j)$. From the definition of d^1, it is easy to show that $x^1 + d^1 \in \Omega(P\backslash B^1)$, $x_f^1 < (x^1 + d^1)_f$, and $(x^1 + d^1)_j > 0$. The basic solution x^2 satisfies $x^2 \in \Omega(P\backslash(B^1 \cup j))$, $x_f^1 \leq x_f^2$, and $x_j^2 < 0$. Then there exists a scalar λ satisfying $0 < \lambda < 1$, $x' = \lambda(x^1 + d) + (1 - \lambda)x^2$ and $x'_j = 0$. The vector x' satisfies $x' \in \Omega(P\backslash B^1/j)$ and $x_f^1 < x'_f$. It contradicts with the optimality of x^1 to $(P\backslash B^1/\mathrm{j})$.

4. Assume both $\mathcal{D}^1$ and $\mathcal{D}^4$ exist. As in the case 3, we have $x^1 + d^1 \in \Omega(P\backslash B^1)$, $x_f^1 < (x^1 + d^1)_f$, and $(x^1 + d^1)_j > 0$. From the definition of d^4, it is also easy to show that $x^1 + d^4 \in \Omega(P\backslash(B^1 \cup j))$, $x_f^1 < (x^1 + d^4)_f$, and $(x^1 + d^4)_j < 0$. Then there exists $0 < \lambda < 1$, satisfying $x' = \lambda(x^1 + d^1) + (1 - \lambda)(x^1 + d^4)$ and $x'_j = 0$. The vector x'

satisfies $x' \in \Omega(P\backslash B^1/j)$ and $x^1_f < x'_f$. It contradicts with the optimality of x^1 to $(P\backslash B^1/\mathrm{j})$.

5. Assume both $\mathcal{D}^2$ and $\mathcal{D}^3$ exist. It is obvious that $(P\backslash(B^3-r)/j)$ is infeasible. The basic solution x^2 satisfies $x^2 \in \Omega(P\backslash((B^3-r)\cup j))$, $x^2_r \geq 0$, and $x^2_j < 0$. From the definition of $\widetilde{x^3}$, it is also easy to show that $\widetilde{x^3} \in \Omega(P\backslash(B^3-r))$, $\widetilde{x^3}_r = 0$, and $\widetilde{x^3}_j > 0$. Then there exists $0 < \lambda < 1$, satisfying $x' = \lambda x^2 + (1-\lambda)\widetilde{x^3}$ and $x'_j = 0$. The vector x' satisfies $x'_r \geq 0$ and $x' \in \Omega(P\backslash(B^3-r)/j)$. It contradicts with the infeasibility of $(P\backslash(B^3-r)/j)$.

6. Assume both $\mathcal{D}^3$ and $\mathcal{D}^4$ exist. As in the case 5, we have $\widetilde{x^3} \in \Omega(P\backslash(B^3-r))$, $\widetilde{x^3}_r = 0$, and $\widetilde{x^3}_j > 0$. From the definition of d^4, it is also easy to show that $x^3 + d^4 \in \Omega(P\backslash((B^3-r)\cup j))$, $(x^3+d^4)_r \geq 0$, and $(x^3+d^4)_j < 0$. Then there exists $0 < \lambda < 1$, satisfying $x' = \lambda\widetilde{x^3} + (1-\lambda)(x^3+d^4)$ and $x'_j = 0$. The vector x' satisfies $x'_r \geq 0$ and $x' \in \Omega(P\backslash(B^3-r)/j)$. It contradicts with the infeasibility of $(P\backslash(B^3-r)/j\ )$. □

Now, we prove the finiteness of the criss-cross method.

Proof of Theorem 1. We 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 some iteration k. Let $\overline{A}$ be the coefficient matrix, and B (respectively, N) be the basis (respectively, non-basis) at iteration k. Let J_1 and J_2 be the index subsets satisfying the conditions that: $J_1 = \{j \in E - \{f,g\} : j \in B,\ j \leq t\} \cup f$, and $J_2 = \{j \in E - \{f,g\} : j \in N,\ j \leq t\} \cup g$. Now we define $\widetilde{A}$ as the submatrix of $\overline{A}$ whose rows are indexed by J_1, and columns are indexed by J_2. Then $-\widetilde{A}$ is a dictionary of the following linear program:

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

where $E' = J_1 \cup J_2$, and x is indexed by E'. From the definition of the algorithm, it is clear that $-\widetilde{A}$ is an almost terminal dictionary of P' with respect to t.

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' (respectively, N') be the basis (respectively, non-basis) at

iteration k'. Since $L_t = 1$ at all iterations from k' to k, the key pivot index chosen at each iteration between k' and k is lower than t. It implies that $B - J_1 \subseteq B'$, and $N - J_2 \subseteq N'$. Let $\widetilde{A}'$ be the submatrix of $\overline{A}'$ whose rows are indexed by $B' \cap E'$, and columns are indexed by $N' \cap E'$. Then it is clear that $-\widetilde{A}'$ is an almost terminal dictionary of P' with respect to t.

If $t \in B$, then obviously $t \in N'$, and if $t \in N$, then $t \in B'$. Thus the almost terminal dictionaries $-\widetilde{A}$ and $-\widetilde{A}'$ are of different types, and this contradicts to lemma 2.

Therefore, at 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 $E - \{f, g\}$ implies the finiteness of the criss-cross method. □

3 Flexibility of the Criss-Cross Method

In this section, 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 unique pivot selection without losing the finiteness of the algorithm. In section 1, we assigned the integer numbers $\{1, 2, \cdots, |E| - 2\}$ to the indices in $E - \{f, g\}$ before starting the algorithm. The relaxation of the selection rule of pivot elements is to permute the integer numbers at the each entrance of step 1, and thus to reorder the indices in $E - \{f, g\}$. The original algorithm can be seen as the one in which the trivial permutation is always used. 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 an entrance of step 1. An 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 set of consecutive 0-labeled numbers. The proof of theorem 1 does not depend on the order of the elements in any 0-labeled interval. Thus using any permutation which is a product of permutations of 0-intervals does not affect the proof of finiteness. In particular, when the minimal index r (s) is chosen in step 1 (step 2), we can select any index in the maximal 0-interval containing r (s) in a relaxed rule, as long as it satisfies the remaining "terminal-violating" conditions.

References

[1] R. G. Bland. A combinatorial abstraction of linear programming. *J. Combin. Theory, Ser. B*, 23 pp.33–57, 1977.

[2] K. Fukuda and T. Matsui. Elementary inductive proofs for linear programming. In *RIMS Kokyuroku 680*, pp. 117–127. Res. Inst. Math. Sci., Univ. Kyoto, 1989.

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

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

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

[6] S. Zionts. The criss-cross method for solving linear programming problems. *Management Science*, 15 pp.426–445, 1969.