Trichotomy for Integer Linear Systems Based on Their Sign Patterns

Kei Kimura\textsuperscript{a,*}, Kazuhisa Makino\textsuperscript{b}

\textsuperscript{a}Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, 441-8580, Japan
\textsuperscript{b}Research Institute of Mathematical Sciences, Kyoto University, Kyoto, 606-8502, Japan

Abstract

In this paper, we consider solving the integer linear systems, i.e., given a matrix $A \in \mathbb{R}^{m \times n}$, a vector $b \in \mathbb{R}^m$, and a positive integer $d$, to compute an integer vector $x \in D^n$ such that $Ax \geq b$ or to determine the infeasibility of the system, where $m$ and $n$ denote positive integers, $\mathbb{R}$ denotes the set of reals, and $D = \{0, 1, \ldots, d-1\}$. The problem is one of the most fundamental NP-hard problems in computer science.

For the problem, we propose a complexity index $\eta$ which depends only on the sign pattern of $A$. For a real $\gamma$, let $\text{ILS}(\gamma)$ denote the family of the problem instances $I$ with $\eta(I) = \gamma$. We then show the following trichotomy:

- $\text{ILS}(\gamma)$ is solvable in linear time, if $\gamma < 1$,
- $\text{ILS}(\gamma)$ is weakly NP-hard and pseudo-polynomially solvable, if $\gamma = 1$,
- $\text{ILS}(\gamma)$ is strongly NP-hard, if $\gamma > 1$.

This, for example, includes the previous results that Horn systems and two-variable-per-inequality (TVPI) systems can be solved in pseudo-polynomial time.

Keywords: Integer linear system, Sign pattern, Complexity index, TVPI system, Horn system
1. Introduction

**Integer linear systems**

In this paper, we consider solving the integer linear systems, i.e., given a matrix \( A = (a_{ij}) \in \mathbb{R}^{m \times n} \), a vector \( b \in \mathbb{R}^m \), and a positive integer \( d \), to compute an integer vector \( x \in D^n \) such that \( Ax \geq b \) or to determine the infeasibility of the system, where \( m \) and \( n \) denote positive integers, \( \mathbb{R} \) denotes the set of reals, and \( D = \{0, 1, \ldots, d-1\} \). We denote the problem by ILS.

The ILS problem is one of the most fundamental and important problems in computer science, and has been studied extensively from both theoretical and practical points of view [19, 31].

The ILS problem is strongly NP-hard, but several (semi-)tractable subclasses are known to exist. For example, the problem can be solved in polynomial time, if \( n \) is bounded by some constant [25], or \( A \) is totally unimodular and \( b \) is integral [16]. Moreover, the ILS problem for the following three subclasses can be solved in pseudo-polynomial time: (1) \( m \) is bounded by some constant and \( A \) is integral [29], (2) \( A \) has at most two nonzero elements per row [2, 15] (such a system is called TVPI system), and (3) \( A \) is Horn [12, 27] (i.e., each row of \( A \) contains at most one positive element). It is also known that the problem is weakly NP-hard, even if \( m \) is bounded by some constant or \( A \) has at most two nonzero elements per row and Horn (also called monotone) [23]. The best known bounds for TVPI and Horn systems respectively require \( O(md) \) time [2, 15] and \( O(n^2md) \) time [12], where the algorithms in [2, 15] are both based on [10].

For unit linear systems, i.e., when \( A \in \{0, -1, +1\}^{m \times n} \), it is known that the problem is still strongly NP-hard, but for TVPI systems, it can be solved in \( O(nm) \) time [24] and \( O(n \log n + m) \) time [32], and for Horn systems, it can be solved in \( O(n^2m) \) time [9, 33]. Finally, for the difference constraint systems, i.e., the monotone unit systems, it is known that the problem is equivalent to the negative cycle detection in network theory and can be solved in \( O(nm) \) time [3, 11, 28] and \( O(\sqrt{nm} \log C) \) time [13], where \( C \) denotes the maximum absolute value of the negative elements in \( b \).

---

1 Note that Horn unit system without upper bounds on the variables, i.e., \( d = +\infty \), is called Horn system and Horn Constraint System in [9] and [33], respectively.
A complexity index for integer linear systems

In this paper, we introduce a complexity index $\eta$ for the ILS problem, which sharply distinguishes among the classes of easy, semi-hard and hard integer linear systems. The idea of this index originates from the work by Boros et al. [5], which introduced a complexity index for the Boolean satisfiability problem (SAT), which distinguishes between the classes of easy and hard SAT instances. Our index $\eta$ is a generalization of theirs to integer linear systems, since SAT problems can be represented as integer linear systems with unit matrices $A \in \{0, -1, +1\}^{m \times n}$, which will be discussed in Section 2.

For a real $a$, its sign is defined as

$$\text{sgn}(a) = \begin{cases} + & (a > 0) \\ 0 & (a = 0) \\ - & (a < 0), \end{cases} \quad (1)$$

and the sign of a real matrix $A \in \mathbb{R}^{m \times n}$ is the matrix $\text{sgn}(A) \in \{0, -, +\}^{m \times n}$ which is obtained from $A$ by replacing each element by its sign.

For example, for a matrix

$$A = \begin{pmatrix} 1 & -3 & 0 \\ 4 & 2 & -5 \end{pmatrix}, \quad (2)$$

we have

$$\text{sgn}(A) = \begin{pmatrix} + & - & 0 \\ + & + & - \end{pmatrix}. \quad (3)$$

Given an instance $I = (A, b, d)$ of the ILS problem, index $\eta(I)$ is the optimal value of the following linear programming problem.

$$\text{minimize} \quad Z \quad \text{subject to} \quad \sum_{j: \text{sgn}(a_{ij}) = +} \alpha_j + \sum_{j: \text{sgn}(a_{ij}) = -} (1 - \alpha_j) \leq Z \quad (i = 1, \ldots, m) \quad (4)$$

$$0 \leq \alpha_j \leq 1 \quad (j = 1, \ldots, n).$$

We note that neither numerical information of $A$ nor any information about $b$ or $d$ is used for our index $\eta(I)$, and it depends only on $\text{sgn}(A)$, i.e., two problem instances $I$ and $I'$ have $\eta(I) = \eta(I')$ if the corresponding matrices have the same sign patterns. Thus we sometimes call $\eta(I)$ index of $A$. 

3
Main results obtained in this paper

For a nonnegative real $\gamma$, let $\text{ILS}(\gamma)$ denote the family of the problem instances $I$ with $\eta(I) = \gamma$. For unit linear systems, i.e., when $A \in \{0, -1, +1\}^{m \times n}$, $\text{UILS}(\gamma)$ is defined analogously. We then have the following main result.

**Theorem 1.** For a nonnegative real $\gamma$ with $\text{ILS}(\gamma) \neq \emptyset$ (equivalently, $\text{UILS}(\gamma) \neq \emptyset$), we have the following three cases:

1. $\text{ILS}(\gamma)$ is solvable in linear time, if $\gamma < 1$,
2. $\text{ILS}(\gamma)$ is weakly NP-hard and pseudo-polynomially solvable, if $\gamma = 1$,
3. $\text{ILS}(\gamma)$ is strongly NP-hard, if $\gamma > 1$.

Moreover, $\text{UILS}(\gamma)$ is polynomially solvable, if $\gamma = 1$, and strongly NP-hard, if $\gamma > 1$.

We also show that $\eta(I) < 1$, = 1, and $> 1$ can be determined in linear time. This theorem implies the previous results that TVPI systems [15, 2] and Horn systems [27] can be solved in pseudo-polynomial time, since TVPI systems and Horn systems are included in $\text{ILS}(\gamma)$ with $\gamma \leq 1$ [5]. Moreover, Theorem 1 includes polynomial solvability for TVPI unit systems [24, 32, 1, 18] and Horn unit systems [9, 33, 8], and tractability of SAT problems (i.e., the polynomial solvability of the satisfiability problem for 2-SAT [10], Horn SAT [14], renamable Horn SAT [26], and q-Horn SAT [4]).

Note that $\text{UILS}(\gamma)$ is partitioned into two classes of easy (i.e., $\gamma \leq 1$) and hard (i.e., $\gamma > 1$) instances, which corresponds to SAT problem given by Boros et al. [5], while the ILS problem is partitioned into three classes of easy, semi-hard and hard systems.

We generalize the results above by considering nonconstant $\gamma$. More precisely, we regard $\gamma$ as a function in $n$ and $d$, denoted by $\gamma(n, d)$. For such index $\gamma(n, d)$, let $\text{ILS}_{\leq}(\gamma(n, d))$ denote the family of the problem instances $I$ with $\eta(I) \leq \gamma(n(I), d(I))$, where $n(I)$ and $d(I)$ denote the input of $n$ and $d$ of $I$, respectively. We similarly define $\text{UILS}_{\leq}(\gamma(n, d))$ for unit linear systems.

We then have the following results.

**Theorem 2.** (1) $\text{ILS}_{\leq}(\gamma(n, d))$ is solvable in linear time, if $0 \leq \gamma(n, d) < 1$.

(2) $\text{ILS}_{\leq}(\gamma(n, d))$ is weakly NP-hard and pseudo-polynomially solvable, if $1 \leq \gamma(n, d) \leq 1 + (c \log_d n)/n$ for some constant $c \geq 0$.  

(3) $\text{ILS}_\leq(\gamma(n,d))$ is strongly NP-hard, if $\gamma(n,d) \geq 1 + 1/n^\delta$ for some constant $\delta < 1$.

Moreover, $\text{UILS}(\gamma)$ is polynomially solvable, if $1 \leq \gamma(n,d) \leq 1 + (c \log_d n)/n$ for some constant $c$, and strongly NP-hard, if $\gamma(n,d) \geq 1 + 1/n^\delta$ for some constant $\delta < 1$.

We note that our tractable classes defined by index $\eta$ is not comparable with the ones for totally unimodular matrices, which is shown in Section 5.

We also note that all the results obtained in this paper are applicable, even if each variable has different lower and upper bounds. For two integral vectors $l, u \in \mathbb{Z}^n$, where $\mathbb{Z}$ denotes the set of integers, consider the problem of finding an integral vector $x$ of the following system:

$$Ax \geq b, \ l \leq x \leq u. \quad (5)$$

Let $d = \max_j \{u_j - l_j\}$. For each variable $x_j$, we introduce a new variable $x'_j$. If $x_j$ appears negatively in some inequality of the system, then we replace it by $x'_j + l_j$ and add the inequality $-x'_j \geq -u_j + l_j$ to the system. Otherwise (i.e., $x_j$ appears nonnegatively in any inequality of the system), we replace it by $x'_j - d + u_j$ and add the inequality $x'_j \geq d - u_j + l_j$ to the system. Note that the resulting system has a form of the systems we consider in the paper, and it is feasible if and only if so is the original one. Moreover, these two systems have same index. These imply that our complexity results can be generalized to the systems in (5).

When some variables have no bounds, i.e., $l \in (\mathbb{Z} \cup \{-\infty\})^n$ and $u \in (\mathbb{Z} \cup \{+\infty\})^n$, we still have the polynomial results when the index is less than 1, since Algorithm SOLVE-INDEX$<1$ discussed in Section 3 correctly works if we replace $\max\{\lceil p \rceil, 0\}$ (resp., $\min\{\lfloor p \rfloor, d-1\}$) by $\lceil p \rceil$ (resp., $\lfloor p \rfloor$) if the corresponding variable has no lower (resp., upper) bound. However, for the systems with index 1, it is open whether the problem can be solved in pseudo-polynomial time. We remark that even for TVPI systems, it is still not known whether the problem can be solved in pseudo-polynomial time, as stated in [15].

We finally remark that there exists a line of research for sign solvability for linear systems [7, 30], linear programming problem [17], and linear complementarity problem [20]. They mainly study sign patterns of the input data that always determine sign patterns of solutions. Their works are motivated by the fact that the input data are uncertain but the structural properties are preserved in most practical situations. While both their and our works
concern the sign patterns of the input, ours differs from theirs in that our work studies the integer solutions and does not concern sign patterns of the solutions.

The rest of the paper is organized as follows. In Section 2, we formally define the ILS problem and introduce previous results for the SAT problem. In Section 3, we consider the case where \( \eta(I) < 1 \). Section 4 presents our results for \( \eta(I) \geq 1 \). Section 5 discusses incomparability between matrices with small index \( \eta \) and totally unimodular matrices.

### 2. Preliminaries

Let \( A \) be a matrix in \( \mathbb{R}^{m \times n} \), and let \( b \) be a vector in \( \mathbb{R}^m \), where \( m \) and \( n \) denote positive integers, and \( \mathbb{R} \) denotes the set of reals. For a positive integer \( d \), let \( D = \{0, 1, \ldots, d-1\} \). In this paper, we consider the problem of computing an integer vector \( x \) that satisfies the system

\[
Ax \geq b, \; x \in D^n, \tag{6}
\]

or determining its infeasibility. Suppose that the \( j \)th column of \( A \) is nonnegative (resp., nonpositive). Then we can fix \( \alpha_j = 0 \) (resp., \( \alpha_j = 1 \)) without increasing the optimal value of (4). Moreover, the system is feasible if and only if so is the one obtained by fixing \( x_j = d - 1 \) (resp., \( x_j = 0 \)). Thus in the following sections except for the discussions at the end of Section 3, we assume without loss of generality that \( A \) contains neither nonnegative columns nor nonpositive columns, where the end of Section 3 deals with properties of matrices \( A \) and optimal solutions of (4) when they have index \( \eta \) at most \( 1/2 \).

We also assume that there exists no row whose elements are all 0. The input length of \( A \) is measured by the number of the nonzero elements of \( A \).

System (6) is said to be TVPI if each row of \( A \) contains at most two nonzero elements, and Horn if each row of \( A \) contains at most one positive element. Let \( V \) denote the set of the indices of variables, i.e., \( V = \{1, \ldots, n\} \). For \( i = 1, \ldots, m \), we define \( P_i \) and \( N_i \) by

\[
P_i = \{ j \in V : \text{sgn}(a_{ij}) = + \} \quad \text{and} \quad N_i = \{ j \in V : \text{sgn}(a_{ij}) = - \}. \tag{7}
\]

We here describe the complexity index introduced by Boros et al. [5] for SAT problem and present some of the results in [5, 6] on the index which will be used later in this paper.
Let \( \varphi = \bigwedge_{i=1}^{m} \left( \bigvee_{j \in L_i^+} x_j \lor \bigvee_{j \in L_i^-} \overline{x_j} \right) \) be a CNF of \( n \) variables, where 
\( L_i^+, L_i^- \subseteq \{1, \ldots, n\} \) with \( L_i^+ \cap L_i^- = \emptyset \) for \( i = 1, \ldots, m \). The complexity index of the CNF \( \varphi \) is the optimal value \( Z(\varphi) \) of the following linear programming problem:

\[
\text{minimize} \quad Z \\
\text{subject to} \quad \sum_{j \in L_i^+} \alpha_j + \sum_{j \in L_i^-} (1 - \alpha_j) \leq Z \quad (i = 1, \ldots, m) \\
0 \leq \alpha_j \leq 1 \quad (j = 1, \ldots, n).
\]

(8)

Given a CNF \( \varphi = \bigwedge_{i=1}^{m} \left( \bigvee_{j \in L_i^+} x_j \lor \bigvee_{j \in L_i^-} \overline{x_j} \right) \), we construct an integer linear system as follows:

\[
\sum_{j \in L_i^+} x_j + \sum_{j \in L_i^-} (1 - x_j) \geq 1 \\
x \in \{0, 1\}^n.
\]

(9)

Namely, \( A \) is a matrix defined by

\[
a_{ij} = \begin{cases} 
1 & j \in L_i^+ \\
-1 & j \in L_i^- \\
0 & \text{otherwise},
\end{cases}
\]

(10)

\( b \) is a vector defined by

\[
b_i = 1 - |L_i^-| \quad (i = 1, \ldots, m),
\]

(11)

and \( d = 2 \).

It is not difficult to see that \( \varphi \) is satisfiable (i.e., \( \exists x \in \{0, 1\}^n : \varphi(x) = 1 \)) if and only if there exists an \( x \in \{0, 1\}^n \) such that \( Ax \geq b \). Moreover, our index \( \eta \) defined by the LP problem (4) is regarded as a natural generalization of the complexity index of SAT defined by Boros et al. [5].

Since for a CNF \( \varphi \), \( Z(\varphi) \) defined in (8) coincides with \( \eta(I) \) of the integer linear system defined in (4), the result [6] for the LP problem (8) immediately implies the following lemma.

**Lemma 1** ([6]). *It can be decided in linear time whether a given integer linear system has index at most 1, and if indeed index is at most 1, a half-integral solution with value 1 can be computed in linear time.*
By Lemma 1, the LP problem (4) has a half-integral optimal solution, if its optimal value is exactly 1.

Moreover, the following results are known for the SAT index.

**Theorem 3** ([5]). Let \( \text{SAT}_{\leq}(\gamma(n)) \) be the set of instances \( \varphi \) of SAT such that \( Z(\varphi) \leq \gamma(n(\varphi)) \), where \( n(\varphi) \) denotes the input of \( n \) of \( \varphi \). Then we have

1. \( \text{SAT}_{\leq}(\gamma(n)) \) is polynomially solvable, if \( 0 \leq \gamma(n) \leq 1 + (c \log_2 n)/n \) for some constant \( c \).

2. \( \text{SAT}_{\leq}(\gamma(n)) \) is strongly NP-hard, if \( \gamma(n) \geq 1 + 1/n^\delta \) for some constant \( \delta < 1 \).

**3. Integer linear systems with index smaller than 1**

In this section, we deal with integer linear systems with index \( \eta \) smaller than 1. We first show that the LP problem (4) does not always have a half-integral optimal solution. By Lemmas 1, 6 and 8, we note that there always exists a half-integral optimal solution if the optimal value is either at most 1/2 or exactly 1, where Lemmas 6 and 8 are shown in this section. We then provide a structural property on the systems, and by using this property, we show that it can be checked in linear time whether a given system has index smaller than 1, and the integer linear system can be solved in linear time when index is smaller than 1.

The following example shows that the LP problem (4) does not always have a half-integral optimal solution, even if the optimal value is smaller than 1. This example will also be used in the proof of Lemma 3.

**Example 1.** Let \( A \) be an \((n + 1) \times n\) matrix such that \( a_{ij} = 1 \) if \( i = j \), \(-1 \) if \( i > j \), and 0 otherwise. Note that \( A \) corresponds to Horn systems. Then the LP problem (4) is written as

\[
\begin{align*}
\text{minimize} & \quad Z \\
\text{subject to} & \quad \alpha_1 \\
& \quad (1 - \alpha_1) + \alpha_2 \leq Z \\
& \quad (1 - \alpha_1) + (1 - \alpha_2) + \alpha_3 \leq Z \\
& \quad \vdots \\
& \quad (1 - \alpha_1) + \cdots + (1 - \alpha_{n-1}) + \alpha_n \leq Z \\
& \quad (1 - \alpha_1) + \cdots + (1 - \alpha_{n-1}) + (1 - \alpha_n) \leq Z \\
& \quad 0 \leq \alpha_j \leq 1 \\
& \quad (j = 1, \ldots, n),
\end{align*}
\]
and it has a unique optimal solution

\[ Z = 1 - \frac{1}{2^n} \quad \text{and} \quad \alpha_j = 1 - \frac{2^{j-1}}{2^n} \quad (j = 1, \ldots, n). \]  

(13)

For example, for \( n = 4 \), \( A \) is given as

\[
A = \begin{pmatrix}
1 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 \\
-1 & -1 & 1 & 0 \\
-1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1
\end{pmatrix},
\]

(14)

and the LP problem (4) is written as follows:

\[
\begin{align*}
\text{minimize} & \quad Z \\
\text{subject to} & \quad \alpha_1 \leq Z \\
& \quad (1 - \alpha_1) + \alpha_2 \leq Z \\
& \quad (1 - \alpha_1) + (1 - \alpha_2) + \alpha_3 \leq Z \\
& \quad (1 - \alpha_1) + (1 - \alpha_2) + (1 - \alpha_3) + \alpha_4 \leq Z \\
& \quad (1 - \alpha_1) + (1 - \alpha_2) + (1 - \alpha_3) + (1 - \alpha_4) \leq Z \\
& \quad 0 \leq \alpha_j \leq 1 \quad (j = 1, 2, 3, 4),
\end{align*}
\]

(15)

which has a unique optimal solution \( Z = 15/16 \), \( \alpha_1 = 15/16 \), \( \alpha_2 = 7/8 \), \( \alpha_3 = 3/4 \), and \( \alpha_4 = 1/2 \).

We note that \( Z \) and \( \alpha \) in (13) is a feasible solution of the LP problem (12), where all the constraints except for \( 0 \leq \alpha_j \leq 1 \) \((j = 1, \ldots, n)\) are satisfied with equalities. Thus the optimal value is at most \( 1 - \frac{1}{2^n} \). To see that it is also a lower bound for the optimal value, let \( c_i \) denote the \( i \)th constraint in the LP problem (12). Then by calculating

\[ c_j + \sum_{i=1}^{j-1} 2^{j-i-1}c_i \quad (j = 1, \ldots, n + 1), \]

we have

\[ \alpha_j + 2^{j-1} - 1 \leq 2^{j-1}Z \quad (j = 1, \ldots, n) \]

(16)

\[ 2^n - 1 \leq 2^nZ. \]

(17)

From (17), we have \( Z \geq 1 - \frac{1}{2^n} \), which implies that \( 1 - \frac{1}{2^n} \) is the optimal value. We finally show that (13) is a unique optimal solution of the LP problem.
(12). It is clear that $Z = 1 - \frac{1}{2^k}$ is unique, since it is the optimal value. This together with (16) implies that any optimal solution satisfies $\alpha_j \leq 1 - \frac{2^{j-1}}{2^n}$ for all $j$. Moreover, if some variable $\alpha_j$ is smaller than $1 - \frac{2^{j-1}}{2^n}$, then the last constraint $c_{n+1}$ is violated. This proves the uniqueness of (13).

We next provide a structural property on the systems with index smaller than 1.

For a problem instance $I = (A, b, d)$ of ILS (i.e., an integer linear system (6)), let $\alpha$ and $Z$ denote an optimal solution of the LP problem (4). For two reals $a$ and $b$ with $a < b$, define $[a, b) = \{z \in \mathbb{R} : a \leq z < b\}$, $(a, b] = \{z \in \mathbb{R} : a < z \leq b\}$ and $[a, b] = \{z \in \mathbb{R} : a \leq z \leq b\}$.

Let $k$ be a positive integer that satisfies $k(1 - \eta(I)) \geq 1/2$, namely $\eta(I) \leq 1 - \frac{1}{2k}$, where we note that $k$ might depend on $m$ and $n$ as in Example 1. We partition $[0, 1]$ into $2k + 1$ sets $\left[\frac{\ell - 1}{2k}, \frac{\ell}{2k}\right)$, $\ell = 1, \ldots, k$, $\{1/2\}$, and $\left(1 - \frac{\ell}{2k}, 1 - \frac{\ell - 1}{2k}\right]$, $\ell = 1, \ldots, k$, i.e.,

$$[0, 1] = \bigcup_{\ell=1}^{k} \left[\frac{\ell - 1}{2k}, \frac{\ell}{2k}\right) \cup \{1/2\} \cup \bigcup_{\ell=1}^{k} \left(1 - \frac{\ell}{2k}, 1 - \frac{\ell - 1}{2k}\right]. \quad (18)$$

Then we have the following lemma, where we recall that $P_i = \{j \in V : \text{sgn}(a_{ij}) = +\}$ and $N_i = \{j \in V : \text{sgn}(a_{ij}) = -\}$, for $i = 1, \ldots, m$.

**Lemma 2.** Let $A$ be a matrix with index smaller than 1, and let $k$ be defined as above. Then for all $i = 1, \ldots, m$, the following statements hold.

(i) $j \in P_i$ implies $\alpha_j \leq 1 - \frac{1}{2k}$, and $j \in N_i$ implies $\alpha_j \geq \frac{1}{2k}$.

(ii) Let $\ell \in \{1, \ldots, k - 1\}$. If some $j$ satisfies either ($j \in P_i$ and $1 - \frac{\ell + 1}{2k} < \alpha_j \leq 1 - \frac{\ell}{2k}$) or ($j \in N_i$ and $\frac{\ell}{2k} \leq \alpha_j < \frac{\ell + 1}{2k}$), then we have $\alpha_{j'} < \frac{\ell}{2k}$ for $j' \in P_i \setminus \{j\}$ and $\alpha_{j''} > 1 - \frac{\ell}{2k}$ for $j'' \in N_i \setminus \{j\}$.

(iii) $P_i \cup N_i$ contains at most one $j$ with $\alpha_j = 1/2$.

**Proof.** (i): Assume that there exists $j \in P_i$ with $\alpha_j > 1 - \frac{1}{2k}$ for some $i$. Then we have $1 - \frac{1}{2k} < \sum_{j \in P_i} \alpha_j + \sum_{j \in N_i} (1 - \alpha_j) \leq Z (= \eta(I))$, where the last inequality follows from the $i$th constraint of the LP problem (4). This contradicts the definition of $k$. The latter case is proven in a similar way.

Similarly, if some $i$ satisfies (ii) or (iii) in the lemma, then it holds that $1 - \frac{1}{2k} < \sum_{j \in P_i} \alpha_j + \sum_{j \in N_i} (1 - \alpha_j) \leq Z (= \eta(I))$, which again contradicts the definition of $k$. \qed
Definition 1. For a matrix $A$, a linear ordering of the columns is called an elimination ordering if $A$ becomes empty by repeatedly eliminating the columns $j$ that satisfies one of the following conditions.

(i) $a_{ij} > 0$ implies $a_{ij'} = 0$ with $j' \neq j$ for all $i = 1, \ldots, m$.

(ii) $a_{ij} < 0$ implies $a_{ij'} = 0$ with $j' \neq j$ for all $i = 1, \ldots, m$.

Note that (i) is equivalent to the condition that all the constraints $i$ with $a_{ij} > 0$ are one-variable, and (ii) is equivalent to the condition that all the constraints $i$ with $a_{ij} < 0$ are one-variable.

Suppose that column $j$ satisfies (i) in Definition 1. Then by using all the constraints $i$ with $a_{ij} > 0$, we can compute the lower bound $p$ for $x_j$, since all such constraints have exactly one variable $x_j$. Since the other constraints have nonpositive coefficient for $x_j$, we can substitute $x_j$ by $\max\{|p|, 0\}$. Similarly, if $j$ satisfies (ii) in Definition 1, then by using all the constraints $i$ with $a_{ij} < 0$, we can compute the upper bound $p$ for $x_j$ and substitute $x_j$ by $\min\{|p|, d-1\}$. Therefore, if matrix $A$ admits an elimination ordering (i.e., there exists an elimination ordering of the columns of $A$), then we can solve the ILS problem by solving one-variable constraints one by one.

The following lemma characterizes index $\eta$ in terms of elimination orderings.

Lemma 3. A matrix $A$ has index less than 1 if and only if it admits an elimination ordering.

Proof. We here recall that $A$ contains neither nonnegative columns nor nonpositive columns by the assumption on $A$. For the only-if part, consider a column ordering $(j_1, \ldots, j_n)$ such that $\max\{\alpha_{j_s}, 1 - \alpha_{j_s}\} \geq \max\{\alpha_{j_i}, 1 - \alpha_{j_i}\}$ if $s < t$. For $\ell = 0, \ldots, k-1$, we inductively show that any column $j$ with $\frac{\ell}{2k} \leq \alpha_j < \frac{\ell + 1}{2k}$ or $1 - \frac{\ell + 1}{2k} < \alpha_j \leq 1 - \frac{\ell}{2k}$ can be eliminated.

For $\ell = 0$, let $j$ be a column with $\alpha_j > 1 - \frac{1}{2k}$. Then by (i) in Lemma 2, we have $j \not\in P_i$ for all $i$, which implies that column $j$ is nonpositive. Similarly, (i) in Lemma 2 implies that any column $j$ with $\alpha_j < \frac{1}{2k}$ is nonnegative. Therefore, no column $j$ satisfies $\alpha_j < \frac{1}{2k}$ or $\alpha_j > 1 - \frac{1}{2k}$, by the assumption on $A$.

Assuming that all the columns $j$ with $\alpha_j < \frac{\ell}{2k}$ or $\alpha_j > 1 - \frac{\ell}{2k}$ have been already eliminated, we consider columns $j$ with $\frac{\ell}{2k} \leq \alpha_j < \frac{\ell + 1}{2k}$ or $1 - \frac{\ell + 1}{2k} < \alpha_j \leq 1 - \frac{\ell}{2k}$. Suppose that $j$ satisfies $1 - \frac{\ell + 1}{2k} < \alpha_j \leq 1 - \frac{\ell}{2k}$. Therefore, the ILS problem by solving one-variable constraints one by one.
It follows from the assumption on \( A \) that some \( P_i \) contains \( j \). Then (ii) in Lemma 2 implies that either \( \alpha_{j'} < \frac{\ell}{2k} \) or \( \alpha_{j'} > 1 - \frac{\ell}{2k} \) holds for \( j' \neq j \) with \( a_{ij'} \neq 0 \). Namely, all such \( j' \) have been already eliminated. Thus we can eliminate column \( j \). Similarly, it follows from (ii) in Lemma 2 that columns \( j \) with \( \frac{\ell}{2k} \leq \alpha_j \leq \frac{\ell+1}{2k} \) can be eliminated.

Finally, by (iii) in Lemma 2, any \( j \) with \( \alpha_j = 1/2 \) satisfies (i) and (ii) in Definition 1. This completes the proof of the only-if part.

For the if part, assume that \( A \) admits an elimination ordering \((1, \ldots, n)\). We note that index \( \eta \) of \( A \) is equal to the one of the matrix obtained from \( A \) by changing the sign of some columns, and hence we assume that any column is eliminated by (i) in Definition 1, which is possible by replacing all the columns \( a_j \) eliminated by (ii) in Definition 1 by \(-a_j\). Then for any column \( j, a_{ij} > 0 \) implies that \( a_{ij'} \leq 0 \) for \( j' < j \) and \( a_{ij'} = 0 \) for \( j' > j \), which follows from the definition of elimination ordering. Thus, the \( i \)th constraint of the LP problem (4) is dominated by \( \sum_{j'=1}^{j-1}(1-\alpha_{j'}) + \alpha_j \leq Z \), since \( \alpha_h \leq 1 \) holds for all \( h \). Hence the feasible region of (4) contains the one of (12). This implies that \( Z \) and \( \alpha \) in (13) form a feasible solution of (4). Therefore, index \( \eta \) of \( A \) is smaller than 1.

We remark that the proof of the if part of Lemma 3 implies that index is at most \( 1 - 1/2^n \) if it is smaller than 1.

Based on Lemma 3, we can construct an algorithm that decides in polynomial time whether index \( \eta \) is smaller than 1. Moreover, the following algorithm (Algorithm FIND-EO in the next page) shows that it can be done in linear time.

Algorithm FIND-EO computes an elimination ordering \((\pi(1), \ldots, \pi(n))\) if index \( \eta \) of \( A \) is smaller than 1. We first describe the outline of Algorithm FIND-EO. Algorithm FIND-EO has two steps: (i) remove single variable constraints and (ii) while there is a variable with no positive (resp., negative) coefficients, eliminate it and remove any resulting single variable constraint. The first step clearly can be done in linear time. The second step can be done in linear time by using the following data structure. In the data structure, each variable has two lists of constraints, one stores the constraints in which the variable occurs positively and the other stores the constraints in which the variable occurs negatively. Once a variable list becomes empty, the corresponding variable can be eliminated. Moreover, each constraint has a list of variables whose coefficient is nonzero in the constraint, and when the list contains one variable, the constraint can be removed. By connecting
Algorithm Find-EO

1: Set $S^+ := S^- := \emptyset$, $s := 1$, $R^+_{\text{init}}[j] := R^-_{\text{init}}[j] := \{i : a_{ij} > 0\}$, $R^-_{\text{init}}[j] := \{i : a_{ij} < 0\}$; $C[i] := \{j : a_{ij} \neq 0\}$

2: for each $i$ call DELETE-CONSTRAINT($i$);

3: for each $j \in S^+ \cup S^-$ do

4: $\pi[s] := j$ and $s := s + 1$

5: if $j \in S^+$ (resp., $j \in S^-$) then

6: $S^+ := S^+ \setminus \{j\}$ (resp., $S^- := S^- \setminus \{j\}$)

7: for each $i \in R^-_{\text{init}}[j]$ (resp., $i \in R^+_{\text{init}}[j]$) do

8: $C[i] := C[i] \setminus \{j\}$

9: call DELETE-CONSTRAINT($i$)

10: end for

11: end if

12: end for

13: if $s = n + 1$ then output “Index $\eta$ is smaller than 1” and halt

14: else output “Index $\eta$ is at least 1” and halt;

15: end if

16: Procedure DELETE-CONSTRAINT($i$)

17: if $C[i] = \{j'\}$ holds for some $j'$ then

18: if $a_{ij'} > 0$ (resp., $a_{ij'} < 0$) then

19: $R^+_{\text{init}}[j'] := R^+_{\text{init}}[j'] \setminus \{i\}$ (resp., $R^-_{\text{init}}[j'] := R^-_{\text{init}}[j'] \setminus \{i\}$)

20: if $R^+_{\text{init}}[j'] = \emptyset$ (resp., $R^-_{\text{init}}[j'] = \emptyset$) then

21: $S^+ := S^+ \cup \{j'\}$ (resp., $S^- := S^- \cup \{j'\}$)

22: end if

23: end if

24: end if
these lists properly by pointers, we obtain a linear time algorithm.

We now define the data structure formally. Vectors $R^+_{\text{init}}$ and $R^-_{\text{init}}$ store the information of $A$. Namely, for each column $j$, let $R^+_{\text{init}}[j] = \{ i : a_{ij} > 0 \}$ and $R^-_{\text{init}}[j] = \{ i : a_{ij} < 0 \}$. For each row $i$, $C[i]$ stores the set of columns $j$ with $a_{ij} \neq 0$ that have not been eliminated so far, and for each column $j$, $R^+[j]$ (resp., $R^-[j]$) stores the set of rows $i$ such that i) $a_{ij} > 0$ (resp., $a_{ij} < 0$) and ii) there exists a non-eliminated column $j'$ with $a_{ij'} \neq 0$ and $j' \neq j$. The vector $C$ is used to compute $R^+$ and $R^-$. By the definition of $R^+$ and $R^-$, $R^+[j] = \emptyset$ (resp. $R^-[j] = \emptyset$) if and only if $j$ satisfies (i) (resp., (ii)) in Definition 1, which implies that $j$ can be eliminated. $S^+$ and $S^-$ keep such columns, and once we choose $j \in S^+ \cup S^-$ as the next column of the ordering, we update $C$, $R^\pm$, and $S^\pm$ accordingly. By properly storing $R^+_{\text{init}}[j]$, $C[i]$, and $S^\pm$, e.g., if each is stored as a double linked list, $j \in C[i]$ and $i \in R^+_{\text{init}}[j]$ are connected by pointers, and we keep the rows $i$ with $|C[i]| = 1$, we can execute Algorithm $\text{FIND-EO}$ in linear time, where an example of the data structures for $R^+_{\text{init}}[j]$ and $C[i]$ are represented in Figure 1.

Note that the first line of Algorithm $\text{FIND-EO}$ initializes $R^+_{\text{init}}[j]$ and $C[i]$, and after the line 2, $R^\pm[j]$ and $S^\pm$ are initialized for the input matrix $A$, e.g., $S^\pm$ stores the columns which can be eliminated from the input $A$. Note that these can be done in linear time by our data structures and by keeping rows $i$ with $|C[i]| = 1$. The second for-loop tries to compute an elimination ordering. If $s$ becomes $n + 1$, i.e., the second for-loop found a linear ordering, then by Lemma 3, we can conclude that index $\eta$ is smaller than 1. Otherwise, we conclude that index $\eta$ is at least 1. Since $j \in C[i]$ and $i \in R^+_{\text{init}}[j]$ are connected by pointers as Figure 1, the second for-loop is also possible in linear time.

**Lemma 4.** Given a matrix $A$, there exists an algorithm that can decide in linear time whether it has index smaller than 1.

Let us then consider solving the integer linear systems with index smaller than 1. By Lemma 3, $A$ admits an elimination ordering $\pi$. Our algorithm fixes variables one by one according to the ordering $\pi(1), \ldots, \pi(n)$. Assume that we have already fixed variables $x_{\pi(\ell)}$ for $\ell = 1, \ldots, j - 1$, and consider variable $x_{\pi(j)}$. If $\pi(j)$ is eliminated by (i) in Definition 1, then let $qx \geq r$ denote an inequality of the current system with $d_{\pi(j)} > 0$. We note that it has no variable other than $x_{\pi(j)}$. By solving such inequalities, we have a lower bound $x_{\pi(j)} \geq p$. If $p > d - 1$, then we can conclude that the system
Figure 1: An example of data structures for $R_{\text{init}}^\pm(j)$ and $C[i]$. 
is infeasible. Otherwise, we substitute $x_{\pi(j)}$ by $\max\{\lceil p \rceil, 0\}$. Similarly, if
the $\pi(j)$th column is eliminated by (ii) in Definition 1, then by solving the
inequalities with negative $x_j$ in the current system, we have an upper bound
$x_{\pi(j)} \leq p$. If $p < 0$, then we can conclude that the system is infeasible.
Otherwise, we substitute $x_{\pi(j)}$ by $\min\{\lfloor p \rfloor, d-1\}$. If the substitution produces
no inconsistent inequality (with no variable), then we proceed to the next
column.

Formally, it can be described as Algorithm SOLVE-INDEX<1. Similarly
to Algorithm FIND-EO, it can be implemented in linear time.

**Algorithm SOLVE-INDEX<1**

1: compute an elimination ordering $(j_1, \ldots, j_n)$
2: for $s := 1$ to $n$ do
3:  if the $j_s$th column satisfies (i) in Definition 1 then
4:    compute a lower bound $x_{j_s} \geq p$ by solving the inequalities in $\{i : a_{ij_s} > 0\}$
5:    if $p \leq d - 1$ then $x_{j_s} := \max\{\lceil p \rceil, 0\}$
6:    else output “infeasible” and halt;
7:  end if
8:  if the $j_s$th column satisfies (ii) in Definition 1 then
9:    compute an upper bound $x_{j_s} \leq p$ by solving the inequalities in $\{i : a_{ij_s} < 0\}$
10:   if $p \geq 0$ then $x_{j_s} := \min\{\lfloor p \rfloor, d-1\}$
11:   else output “infeasible” and halt;
12: end if
13: if the resulting system has an inconsistent inequality with no variable then
14:   output “infeasible” and halt
15: else remove inequalities with no variable from the system;
16: end for
17: output a solution $x$, and halt.

**Lemma 5.** The integer linear system problem can be solved in linear time if
index $\eta$ is smaller than 1.

Before concluding this section, we show that the LP problem (4) has a
half-integral optimal solution, if index $\eta$ is at most half. Here we do not
assume that $A$ contains neither nonnegative nor nonpositive columns to analyze
a relationship between matrices $A$ and their indices $\eta$. 
Let us first consider the case in which index $\eta$ is smaller than $1/2$. Then we see that the sign pattern of $A$ is rather simple in this case. In fact, the following lemma holds.

**Lemma 6.** If the LP problem (4) has the optimal value $\eta(I) < 1/2$, then $\eta(I) = 0$ holds, and it has a unique 0–1 optimal solution.

**Proof.** Consider an optimal solution $(Z, \alpha_1, \ldots, \alpha_n)$. There exists no $j \in V$ with $\alpha_j = 1/2$, since otherwise we have $Z \geq 1/2$, a contradiction. If $\alpha_j > 1/2$ for some $j \in V$, then the $j$th column of $A$ is nonpositive, since the optimal value is smaller than 1/2. This implies that we can replace $\alpha_j$ by 1 without increasing the objective value. Similarly, if $\alpha_j < 1/2$, we can replace $\alpha_j$ by 0, since the $j$th column of $A$ is nonnegative. Note that the replacement above implies that the optimal value is 0. Indeed, it is a unique optimal solution, since each column is nonzero. □

The above lemma immediately implies the following result.

**Lemma 7.** Let $I = (A, b, d)$ be a problem instance. Then there exists an algorithm that can decide whether $\eta(I) < 1/2$ in linear time, and, if so, the problem can be solved in linear time.

**Proof.** Note that $\eta(I) < 1/2$ (and hence $\eta(I) = 0$ by Lemma 6) holds if and only if each column of $A$ is either nonnegative or nonpositive, which can be checked in linear time. In order to solve the integer linear system, consider the $n$-dimensional vector $y$ such that $y_j = d - 1$ if the $j$th column of $A$ is nonnegative, and 0, otherwise (i.e., if the $j$th column of $A$ is nonpositive). Then we can see that the system is feasible if and only if $y$ is a feasible solution, since $Ay \geq Ax$ for all $x \in D^n$. Thus we can solve the problem in linear time. □

We finally consider the case in which index $\eta$ is $1/2$.

**Lemma 8.** If the LP problem (4) has the optimal value $\eta(I) = 1/2$, then it has a half-integral optimal solution.

**Proof.** If $\alpha_j > 1/2$ (respectively, $\alpha_j < 1/2$) for some $j \in V$, then $\eta(I) = 1/2$ implies that the $j$th column of $A$ is nonpositive (respectively, nonnegative). Define a vector $\alpha^* \in \mathbb{R}^n$ by $\alpha^*_j = 1$ if the $j$th column of $A$ is nonpositive, 0 if the $j$th column of $A$ is nonnegative, and 1/2, otherwise. Then it follows from $\eta(I) = 1/2$ that each row $i$ contains at most one nonzero element $a_{ij}$ with $\alpha_j = 1/2$. Thus $\alpha^*$ is also an optimal solution of the LP problem (4). □
By the above lemma, we have the following result.

**Lemma 9.** Let $I = (A, b, d)$ be a problem instance. Then there exists an algorithm that can decide whether $\eta(I) = 1/2$ in linear time, and if so, the problem can be solved in linear time.

**Proof.** By Lemma 8, the integer linear system has index $1/2$ if and only if $Z = 1/2$ and $\alpha^*$ constructed in the proof of Lemma 8 is an optimal solution. Therefore, it can be checked in linear time. To solve the system, we can fix $x_j = 0$ for all $j \in V$ with $\alpha^*_j = 1$, and $x_j = d - 1$ for all $j \in V$ with $\alpha^*_j = 0$. Then all the inequalities in the resulting system contain at most one variable, and hence it can be solved in linear time. \qed

4. Integer linear systems with index $\geq 1$

4.1. The case of $\eta(I) = 1$

In this section, we consider integer linear systems of index 1, and prove Theorem 1 for this case. The proof is in the same line of Boros et al. [5], but it does not directly follow from [5]. Hence we provide an outline of the proofs, where the proof can be found in [21].

Let $(Z, \alpha_1, \ldots, \alpha_n)$ be an optimal solution of the LP problem (4). It is known [4, Remark 3] that $\alpha \in \{1/2, 1\}^n$ can be assumed. In fact, by Lemma 1, we can assume that $\alpha \in \{0, 1/2, 1\}^n$ and removal of 0’s can be done by replacing the variable $x_j$ to a new variable $x_j' = d - 1 - x_j$ for each $j$ with $\alpha_j = 0$. Let $Q = \{j \in V : \alpha_j = 1/2\}$ and $H = \{j \in V : \alpha_j = 1\}$. By $\alpha \in \{1/2, 1\}^n$, $V$ can be partitioned into $Q$ and $H$:

$$V = Q \cup H,$$

which is called a $QH$-partition of $V$. Then we have the following properties, where we recall that $P_i = \{j \in V : \text{sgn}(a_{ij}) = +\}$ and $N_i = \{j \in V : \text{sgn}(a_{ij}) = -\}$ for $i = 1, \ldots, m$.

**Lemma 10 ([5]).** Let $Q$ and $H$ be defined as above. Then they satisfy the following three conditions:

(a) Each row $i$ of $A$ contains at most two nonzero elements $a_{ij}$ with $j \in Q$. Or equivalently, $|(P_i \cup N_i) \cap Q| \leq 2$ holds for all $i = 1, \ldots, m$.

(b) Each row $i$ of $A$ contains at most one positive element $a_{ij}$ with $j \in H$. Or equivalently, $|P_i \cap H| \leq 1$ holds for all $i = 1, \ldots, m$. 18
(c) If a row \(i\) of \(A\) contains a positive element \(a_{ij}\) with \(j \in H\), then the elements \(a_{ik}\) with \(k \in Q\) are all zeros. Or equivalently, if \(P_i \cap H \neq \emptyset\) then \((P_i \cup N_i) \cap Q = \emptyset\).

We briefly explain a pseudo-polynomial time algorithm for the problem. For a \(QH\)-partition of \(V\), let \(S\) denote the set of rows \(i\) of \(A\) such that \(a_{ij} = 0\) for all \(j \in Q\). Let \(A[S, H]\) denote the submatrix of \(A\) whose row and column sets are \(S\) and \(H\), respectively, and let \(b_S\) and \(x_H\) respectively denote the restriction of \(b\) to \(S\) and \(x\) to \(H\). Then by Lemma 10 (b), the linear system \(A[S, H]x_H \geq b_S\) is Horn, i.e., each row of \(A[S, H]\) contains at most one positive element. It is known that any Horn system has a unique minimal solution if it is feasible. Here \(x^*\) is called a unique minimal solution of the system if it is a solution that satisfies \(x^* \leq x\) for all solutions \(x\) to the system. Assume that \(A[S, H]x_H \geq b_S\) is feasible and let \(x_H^* \in D_H\) be a unique minimal solution for \(A[S, H]x_H \geq b_S\). Since Lemma 10 (c) implies that any element \(a_{ij}\) with \(i \notin S\) and \(j \in H\) is nonpositive, the original integer linear system is feasible if and only if so is the system obtained from it by substituting \(x_H = x_H^*\). We note that a unique minimal solution of a Horn system can be obtained in pseudo-polynomial time [12, 27]. Since the resulting system is TVPI, we can solve it in pseudo-polynomial time [15, 2]. We summarize the algorithm in Algorithm Solve-Index=1.

**Algorithm Solve-Index=1**

1: compute a \(QH\)-partition \(V = Q \cup H\) of \(V\)
2: if the system of \(x_H \in D_H\) and \(A[S, H]x_H \geq b_S\) is infeasible then
3: output “infeasible” and halt
4: else compute a unique minimal solution \(x_H^*\) and \(x_H := x_H^*\);
5: if the resulting system is infeasible then
6: output “infeasible” and halt
7: else
8: compute a solution \(x_Q^* \in D_Q\) of the resulting system
9: output a solution \((x_H^*, x_Q^*)\) and halt;

**Lemma 11.** Algorithm Solve-Index=1 solves the integer linear system with index 1 in time polynomial in \(n, m\) and \(d\).
Proof. Since the correctness of Algorithm Solve-Index−1 follows from the discussion before describing the algorithm, we discuss its time complexity only.

It is known [6] that a $QH$-partition can be computed in linear time. Horn and TVPI systems can be solved in $O(n^2md)$ time [12] and $O(md)$ time [2, 15], respectively. Therefore, in total, the algorithm requires polynomial time in $n$, $m$, and $d$.

This immediately implies the following corollary.

**Corollary 1.** For a unit matrix $A$, Algorithm Solve-Index−1 solves the integer linear system with index 1 in polynomial time.

**Proof.** The lemma follows from the fact that Horn [8] and TVPI [18] integer linear systems are solvable in polynomial time, if $A$ is unit.

We next show the weak NP-hardness of the problem.

**Lemma 12.** ILS(1) is weakly NP-hard.

**Proof.** It is known that solving TVPI system is weakly NP-hard [23] and TVPI systems have index at most 1 [6]. For a TVPI system, let us add two inequalities $y - z \geq 0$ and $z - y \geq 0$, where $y$ and $z$ are new variables. Then the resulting system is feasible if and only if so is the original system, since $y = z = 0$ satisfies the two inequalities. Note that the resulting system has index exactly 1. This completes the proof.

Before concluding this section, we mention that the pseudo-polynomiality for ILS$_\leq$ and the polynomiality for UILS$_\leq$ hold, even if $1 < \eta(I) \leq 1 + (c \log_d n)/n$ for some constant $c$. Since the proofs are in the same line of the proof of Theorem 3 (1), we skip the proof, which can be found in [22].

4.2. The case of $\eta(I) > 1$

In this section, we consider the case in which our index is greater than 1. As mentioned in Section 2, index for CNF coincides with index for the ILS problem. By combining this with (2) in Theorem 3, we can prove (3) in Theorem 1.

Note that they imply that for any constant $\gamma > 1$, ILS$_\leq(\gamma)$ and UILS$_\leq(\gamma)$ are strongly NP-hard. We now show (3) in Theorem 1.
Lemma 13. Let $\gamma$ be a constant with $\gamma > 1$ and $\text{UILS}(\gamma) \neq \emptyset$. Then $\text{UILS}(\gamma)$ is strongly NP-hard.

Proof. We reduce $\text{UILS}_{\leq}(\gamma)$ to $\text{UILS}(\gamma)$. Let $A$ be an $m \times n$ unit matrix with the optimal value $\gamma$ of the LP problem (4). Note that such a matrix exists, since $\text{UILS}(\gamma) \neq \emptyset$.

For a problem instance $A'x' \geq b'$ of $\text{UILS}_{\leq}(\gamma)$, let $\gamma'$ be its complexity index. Then consider the following integer linear system:

$$
\begin{pmatrix}
A & 0 \\
0 & A'
\end{pmatrix}
\begin{pmatrix}
x \\
x'
\end{pmatrix}
\geq
\begin{pmatrix}
0 \\
b'
\end{pmatrix},
$$

where 0 denotes a zero matrix (or vector) of appropriate size. This system has a solution if and only if $A'x' \geq b'$ has a solution, because $x = 0$ clearly satisfies $Ax \geq 0$. Since $\gamma \geq \gamma'$, this completes the proof.

5. Discussion

In this section, we show that the class of matrices with low index are incomparable with the one of totally unimodular matrices, where total unimodularity is well known for offering polynomial time algorithms for the integer linear programming problem. Namely, we present two matrices $A_1$ and $A_2$ such that $A_1$ has index 0 and arbitrarily high determinant and $A_2$ is totally unimodular of arbitrarily high index.

Example 2. Let $B$ be a $3 \times 3$ matrix defined as

$$
B = 
\begin{pmatrix}
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0
\end{pmatrix},
$$

and $A_1$ be a $3n \times 3n$ matrix defined as

$$
A_1 = 
\begin{pmatrix}
B & \cdots & 0 \\
0 & \cdots & B
\end{pmatrix},
$$

that is, $A_1$ consists of $n$ diagonal blocks $B$. Then we note that $A_1$ has index 0 and $\det A_1 = 2^n$. In particular, the latter implies that $A_1$ is not totally unimodular.
Example 3. Let $C$ be a $2 \times 2$ matrix defined as
\[
C = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},
\]
and $A_2$ be a $2 \times 2n$ matrix defined as
\[
A_2 = \begin{pmatrix} C & \cdots & C \end{pmatrix}.
\]
Then $A_2$ is totally unimodular and has index $n$.

Acknowledgment

The authors are indebted to the reviewers for their valuable comments and suggestions, which have considerably improved the presentation of the results.

This work is supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan. The first author is supported by the Global COE “The Research and Training Center for New Development in Mathematics,” the Research Fellowship of the Japan Society for the Promotion of Science for Young Scientists, and by JST, ERATO, Kawarabayashi Large Graph Project.

References


