AN ISOMETRIC SURFACE METHOD FOR INTEGER LINEAR PROGRAMMING*

Y. Y. NIE†, L. J. SU‡ and C. LI

Shenyang Institute of Automation, Academia Sinica; Graduate School, C.A.S. No.114, Nanta Avenue, Shenyang, 110016, P.R. China

(Received December 2002)

Based on the isometric plane method for linear programming, an algorithm for integer linear programming is proposed in this paper. The algorithm can quickly obtain the optimal integer point simultaneously using isometric planes and cutting planes derived from polyhedral-cones, rounded-minimal-balls and second-rounded-balls at the highest vertex and its neighboring vertices.

Keywords: Integer programming; Linear programming; Isometric plane; Cutting plane; Isometric surface

AMS Subject Classification: 90C05, 90C10

C.R. Categories: G.1.3

1 INTRODUCTION

We consider the following integer linear programming:

\[ \begin{align*}
  \text{max} \quad z &= c^T x, \\
  \text{s.t.} \quad Ax &\geq b, \quad x = (x_1, x_2, \ldots, x_n)^T: \text{integer vector}
\end{align*} \tag{1.1} \]

where \( A = (a_{ij}), c = (c_1, c_2, \ldots, c_n)^T, b = (b_1, b_2, \ldots, b_m)^T \) are given \( m \times n \) matrix and vectors in \( \mathbb{R}^n \) and \( \mathbb{R}^m \) respectively. The relaxation linear programming of (1.1) is

\[ \begin{align*}
  \text{max} \quad z &= c^T x, \\
  \text{s.t.} \quad Ax &\geq b.
\end{align*} \tag{1.2} \]

There is no equality in the constraint conditions of problems (1.1) and (1.2). Otherwise, for instance, an equality

\[ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i \tag{1.3} \]
can be replaced by two inequalities
\[\begin{align*}
ad_1 x_1 + ad_2 x_2 + \cdots + ad_n x_n & \geq b_1 - \delta_0, \\
-ad_1 x_1 - ad_2 x_2 - \cdots - ad_n x_n & \geq -b_1 - \delta_0, \\
\end{align*}\]
where \(\delta_0\) is a positive number near to computer zero, for example \(\delta_0 = 10^{-9}\). \((\delta_0^{-1}\) is understood as infinite.) Clearly, the replacement (1.4) does not decrease integer points of the hyperplane (1.3) in \(\mathbb{R}^n\); however, it is possible to increase integer points. When the distance (L_2 norm) from some integer point to the hyperplane (1.3) is not bigger than a multiplier of \(\delta_0\), the integer point is included in the “thin” space defined by (1.4). If this phenomenon happens to the replacement (1.4), then we approximately think of the integer point is located on the hyperplane (1.3).

Generally the following hypothesis is adopted in this paper,

**Thick-whole-point hypothesis:** Assume \(x, y \in \mathbb{R}^n\) and \(y\) is a rounded integer point of \(x\). If
\[
\begin{align*}
\|x - y\|_2 &= \left(\sum_{i=1}^{n} (x_i - y_i)^2\right)^{1/2} \leq \delta_0,
\end{align*}
\]
then the point \(x\) is considered an integer point.

Under the thick-whole-point hypothesis (1.1) has possibly several solutions, even if (1.2) has unique solution. Suppose \(x^*\) is the solution of (1.2), then all the integer points \(x^I\) that satisfy
\[
\|c^T(x^* - x^I)\|_2 \leq \delta_0
\]
are the solutions of (1.1).

Based on solution of the relaxation linear programming (1.2), the most popular methods for the integer linear programming (1.1) are branch-and-bound and cutting-plane [4] up to now. They both become quite complicated with increasing \(n\) and \(m\).

In this paper we present a method, referred to as isometric surface method, based on the isometric plane method for linear programming [3]. In order to get the solution of problem (1.1), the proposed method need not solve relaxation problems (1.2) many times in most cases.

The isometric plane method, an interior-point method, for linear programming (1.2) considers constraints an arbitrary convex polyhedron \(\Omega^m\) in \(\mathbb{R}^n\). In \(\Omega^m\) a strictly interior point is successively moved to a higher isometric plane from a lower one along the gradient. Finally, the highest boundary point which makes the objective function value maximum is found, or unbounded objective function value is concluded. With determination of initial interior point we can know if the constraints of (1.2) are consistent.

In the description of Ref. [3] the boundary points of the convex polyhedron \(\Omega^m\) are divided into three classes: the surface point, the edge point and the vertex. The edge point is the point \(y\) which satisfies
\[
A_i^Ty = b_i, \quad i = l_1, l_2, \ldots, l_r, \quad 1 \leq r \leq n,
\]
\[
\sum_{j=1}^{r} z_j A_{ij} \neq 0 \quad \forall z_1, \ldots, z_r \in \mathbb{R}^1, \quad \sum_{j=1}^{r} z_j^2 \neq 0,
\]
\[
A_i^Ty > b_i, \quad i = l_{r+1}, \ldots, l_m,
\]
where \(A_i^T = (a_{i1}, a_{i2}, \ldots, a_{in})\), \(l_1, \ldots, l_m\), is a permutation of \(1, \ldots, m\). When \(r = 1\), an \((n - 1)\)-dimensional edge point is a surface point of \(\Omega^m\). And when \(r = n\), a 0-dimensional edge point is a vertex of \(\Omega^m\). The manifold
\[
E_y^r = \{x \in \mathbb{R}^n | A_i^T(x - y) = 0, \quad j = 1, \ldots, r\}
\]
is an \((n - r)\)-dimensional edge of \(\Omega^n\) passing through \((n - r)\)-dimensional edge point \(y\). The highest boundary point is usually a vertex, but it is possible that the highest boundary point is \((n - r)\)-dimensional edge point \((1 \leq r < n)\). Assume \(y^*\) satisfied (1.5) is such a highest edge point. In order to find a vertex on the edge \(E_{y^*}^r\), we can first let, for instance, \(x_{r+1}^* = \cdots = x_n^* = 0\), then solve (1.6) for \(x_1^*, \ldots, x_r^*\), and obtain another highest edge point \(x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T\). The \((x^* - y^*)\)-line passing through \(y^*\):

\[
L_{y^*}^x = \{z \in \mathbb{R}^n | z - y^* = t(x^* - y^*), t \in \mathbb{R}\}
\]

has generally two intersection points with boundary \(\partial \Omega^n\). (If there is no intersection point, then let

\[
t^* = \pm \frac{1}{\|y^* - y^*\|_2}
\]

and regard \(z^* = y^* + t^*(x^* - y^*)\) as the boundary points.) An intersection point \(z^*\) is a highest \((n - r - 1)\)-dimensional edge point. Using similar procedure above, we can always get a highest vertex on the edge \(E_{y^*}^r\).

If the problem (1.2) has an unbounded objective function value, then we regard

\[
d_j^T(x - x^0) = \frac{\|d_{r}\|_2}{\varepsilon_0}
\]

as a boundary hyperplane, where \(x^0\) is an initial interior point and \(d_j\) is the gradient making objective function value increase infinitely [3]. We can also find a highest vertex on the plane (1.7). It is uncertain if the integer programming (1.1) has a solution when the relaxation programming (1.2) has an unbounded solution [1].

The method presented in this paper is applicable for mixed-integer linear programming as well.

2 POLYHEDRAL-CONE AND ROUNDED-MINIMAL-BALL

Assume now we have obtained, starting from an initial interior point \(x^0\), the highest vertex \(x^*\) for relaxation linear programming (1.2) whatever the nonempty constraint polyhedron \(\Omega^n\) is bounded or unbounded. \(x^*\) is usually an intersection point of \(n\) hyper-planes. If there are more than \(n\) hyper-planes intersecting at \(x^*\), then take \(n\) hyper-planes recorded, which are linearly independent of each other and include the initial interior point \(x^0\), as intersection hyper-planes. These \(n\) hyper-planes with vertex \(x^*\) and normal vectors \(A_{lj}\) \((j = 1, 2, \ldots, n)\) form a polyhedral-cone \(C_x^n\) in \(\mathbb{R}^n\), that is,

\[
C_x^n = \{x \in \mathbb{R}^n | A_{lj}^T(x - x^*) \geq 0, j = 1, 2, \ldots, n\}
\]

There are \(n\) one-dimensional edges on the cone \(C_x^n\). Each 1D edge is a ray passing through \(x^*\) and satisfying

\[
A_{lj}^T(x - x^*) = 0, \quad i = 1, \ldots, j - 1, j + 1, \ldots, n; \quad A_{lj}^T(x - x^*) > 0, \quad j = 1, 2, \ldots, n.
\]

Let \(y^h = x - x^*\) be a 1D edge vector and its \(k\)th component of \(y^h\) equal 1 or \(-1\) such that \(y^h\) can be found by

\[
A_{lj}^Ty^h = 0, \quad i = 1, \ldots, j - 1, j + 1, \ldots, n; \quad y^h_k = \pm 1; \quad A_{lj}^Ty^h > 0, \quad 1 \leq j \leq n.
\]
Thus the 1D edge is the ray

$$L^y_{x^*} = \{x \in \mathbb{R}^n | x - x^* = ty^j, t \geq 0 \} \quad j = 1, 2, \ldots, n. \quad (2.3)$$

The highest vertex $x^*$ of (1.2) is not an integer point usually, otherwise the integer programming (1.1) has been solved. We can round each component of $x^*$ off to the nearest whole number, thus obtain the rounded-integer point $x^N$ nearest to $x^*$. If

$$r_{x^*} = \|x^N - x^*\|_2 \leq \varepsilon_0,$$

then $x^*$ is considered an integer point in thick-whole-point hypothesis, so then suppose $r_{x^*} > \varepsilon_0$. The ball

$$O_{x^*} = \{x \in \mathbb{R}^n | \|x - x^*\|_2 \leq r_{x^*} \} \quad (2.4)$$

is referred to as rounded-minimal-ball. Clearly, there is no integer point inside the ball $O_{x^*}$, and there is one integer point at least on the surface of $O_{x^*}$, and the radius $r_{x^*} \leq \sqrt{n}/2$.

To determine if the rounded-integer point $x^N$ belongs to $\Omega^m$, we consider the constraint set of (1.2). If

$$Ax^N \geq b - \varepsilon_0 e, \quad (2.5)$$

where $e = (1, 1, \ldots, 1)^T$ and $\varepsilon_0$ is the same as the right-hand side of (1.4), then $x^N$ is referred to on $\Omega^m$.

It is possible that there are $2^k$ integer points on the surface of $O_{x^*}$, where the fractional part of $k$ components of $x^*$ is near to 0.5, namely

$$E_k: \lfloor [x^*]^j \rfloor - x^*_j = 0.5 \pm \varepsilon_0, \quad j = j_1, \ldots, j_k, \quad 1 \leq k \leq n, \quad (2.6)$$

$[x^*_j]$ denotes the least integer not less than $x^*_j$.

However, the probability of event (2.6) is very small. Let the fractional part of independent components, which drops equally likely into an interval with length $2\varepsilon_0$ in $[0, 1]$, be a sample. Then the sample space consists of $(1/(2\varepsilon_0))^n$ samples. And the probability for the event (2.6) to happen is

$$p(E_k) = C_n^k (2\varepsilon_0)^k (1 - 2\varepsilon_0)^{n-k}, \quad 0 \leq k \leq n,$$

which belongs to binomial distribution[2]. Specially,

$$p(E_0) = (1 - 2\varepsilon_0)^n > 1 - 2n\varepsilon_0,$$

$$p(E_1) = 2n\varepsilon_0 (1 - 2\varepsilon_0)^{n-1} < 2n\varepsilon_0,$$

namely on the surface of rounded-minimal-ball $O_{x^*}$ there is one integer point with probability larger than $1 - 2n\varepsilon_0$, and there are two points with probability less than $2n\varepsilon_0$.

We have shown

**Theorem 1** Let $x^*$ be the highest vertex of relaxation linear programming (1.2), $C^y_{x^*}$ the polyhedral-cone defined by (2.1), and $O_{x^*}$ the rounded-minimal-ball with radius $r_{x^*} > \varepsilon_0$ defined by (2.4). Then, there is no integer point inside the $O_{x^*}$ and $r_{x^*} \leq \sqrt{n}/2$, there is only one integer point $x^N$ with probability larger than $1 - 2n\varepsilon_0$ on the surface of $O_{x^*}$, and the decidable condition to determine if $x^N$ belongs to constraint polyhedron $\Omega^m$ is (2.5), here, $\varepsilon_0$ is a small positive number near to computer zero.
After the rounded-minimal-ball and its \( \zeta \) rounded-integer points are considered, we may continually consider a little large rounded-ball which is referred to as the second-rounded-ball. For a rounded-integer point \( x^N = (x_1^N, x_2^N, \ldots, x_n^N)^T \) of \( O_x \), let

\[
\xi_i = x_i^N - x_i^a, \quad i = 1, 2, \ldots, n,
\]

\[
\eta_p = \begin{cases} 
\min_{1 \leq i \leq n} \{ 0.5 - |\xi_i| > \epsilon_0 \} & \zeta < 2n \\
\min_{1 \leq i \leq n} \{ 0.5 - |\xi_i| \} & \zeta \geq 2n
\end{cases}
\]

Then

\[
x^M = (x_1^N, \ldots, x_p^N, x_p^N - \text{sign}(\xi_p), x_{p+1}^N, \ldots, x_n^N)^T
\]

is also an integer point, where

\[
\text{sign}(\xi_p) = \begin{cases} 
1 & \xi_p > 0 \\
0 & \xi_p = 0 \\
-1 & \xi_p < 0
\end{cases}
\]

Evidently

\[
R_x^* = \|x^M - x^*\|_2 \geq r_x^*
\]

and \( x^M \) is next nearest integer point from \( x^* \). The ball

\[
O_x' = \{ x \in \mathbb{R}^n \| x - x^*\|_2 \leq R_x^* \}
\]

is called second-rounded-ball. There is no integer point inside the \( O_x' \) except the integer points on the rounded-minimal-ball surface. The case \( R_x' = r_x^* \) happens only if there are more than or equal to \( 2n \) rounded-integer points of \( O_x' \).

The \( n \) 1D edges of \( C_x^n \) with formula (2.3) intersect to the surface of second-rounded-ball \( O_x' \) at \( n \) points. These \( n \) intersection points define an \((n-1)\)D hyperplane \( \pi_x' \) which makes the inside of close cone \( C_x^n \) a non-integer point set except integer point on the surface of \( O_x' \) because of Theorem 1 and the convexity of ball \( O_x' \). Let \( x^{l_j} \) be the intersection point, then

\[
x^{l_j} = x^* + \frac{t_j y^{l_j}}{\|y^{l_j}\|_2}, \quad t_j = r_x^*, \quad j = 1, 2, \ldots, n.
\]

(2.7)

Suppose that the bottom plane \( \pi_{x^*} \) of \( C_x^n \) is

\[
(a_1^1, a_2^1, \ldots, a_n^1)x = \pm 1, \quad (a_1^1, a_2^1, \ldots, a_n^1)x^* < 1.
\]

(2.8)

We have clearly

\[
(a_1^1, a_2^1, \ldots, a_n^1)x^{l_j} = \pm 1, \quad j = 1, 2, \ldots, n, \quad (a_1^1, a_2^1, \ldots, a_n^1)x^* < 1.
\]

(2.9)

The linear equations (2.9) can uniquely determine the normal vector \((a_1^1, a_2^1, \ldots, a_n^1)^T\) of \( \pi_{x^*} \). Cutting-plane \( \pi_{x^*} \) cuts out a non-integer part near \( x^* \) if the rounded-integer points of \( O_x' \) is considered. We denote specially the close cone \( \overline{C}_x^n \) with bottom plane \( \pi_{x^*} \) as \( C_{x^*}^{x^*} \).

After a non-integer part near \( x^* \) is cut out with \( \pi_{x^*} \), some of intersection points \( x^{l_1}, x^{l_2}, \ldots, x^{l_n} \) are possibly vertices of \( \Omega^n \). If \( x^{l_j}(1 \leq j \leq n) \) is such a non-integer vertex, then we can also consider the rounded-minimal-ball \( O_{x^{l_j}}^* \) with rounded-integer points \( x_N^j \) and radius \( r_{x^{l_j}} \).
Now let us suppose that an integer point \( x^N \) of some rounded-minimal-ball surface belongs to \( \hat{\Omega}^m \). There are at most \( n \) rays from \( x^N \) along the coordinates such that the objective function of (1.1) is increasing. Set

\[
e_i = (0, \ldots, 0, \text{sign}(c_i), 0, \ldots, 0)^T,
\]

then

\[
c^T(x^N + t_i e_i) \geq c^T x^N \quad \forall t_i \geq 0.
\]

Let for \( i = 1, 2, \ldots, n \)

\[
t_i = \begin{cases} 0 & c_i = 0 \\ \min_j \left\{ \frac{b_j - A_j^T x^N}{A_j^T e_i} > 0 \right\} & c_i \neq 0' \\
\end{cases}
\]

then \( x^N + [t_i] e_i \) is clearly an integer point, were \( [t_i] \) denotes the biggest integer less than or equal to \( t_i \). Assume that

\[
c^T(x^N + [t_p] e_p) = \max_{1 \leq i \leq n} c^T(x^N + [t_i] e_i).
\]

If \( [t_p] \geq 1 \), then record

\[
x^N := x^N + [t_p] e_p.
\]

If \( [t_p] \geq c_0^{-1} \), then \( x^N \) is infinite. The procedure similar to (2.10)–(2.12) may be performed for new finite integer points \( x^N \) until \( [t_p] = 0 \).

### 3 DESCRIPTION OF THE ALGORITHM

**Step 1** Record the solution \( x^H = (-1/(c_0 \|c\|_2))c \) of integer linear programming (1.1). Let \( k_n = k_m = 0 \).

**Step 2** Let \( ks = kc = kd = ke = 0 \). Check if the constraint polyhedron \( \Omega^m \) of relaxation linear programming (1.2) is empty. If it is then go to exit, else get an initial interior point \( x^0 \).

**Step 3** Make the rounded-minimal-ball \( O^x \) with rounded-integer points \( x^N \). If \( x^N \notin \Omega^m \) then go to Step 4, else record \( x^H = x^N, x^0 = x^N \), print \( x^H \) and if \( k_n = 0 \) then append the inequality \( c^T x \geq c^T x^N \) to the constraint set of (1.2) and let \( k_n = k_m = 1 \), else append new inequality instead of old one and let \( k_n = k_m = 1 \).

**Step 4** Find the highest vertex \( x^* \) of (1.2) using isometric plane method [3]. If \( \| c^T(x^* - x^H)/\| c \|_2 \| \leq e_0 \), then go to exit.

**Step 5** Make the rounded-minimal-ball \( O^x \) with rounded-integer points \( x^N \) and radius \( r_x^* \). If \( r_x^* \leq e_0 \), then \( x^* \) is the solution of (1.1) in thick-whole-point hypothesis, record \( x^H = x^*, k_n = 1 \), and go to exit; If \( c^T x^* \leq -\|c\|_2/e_0 \), then there is no solution for the problem (1.1), go to exit.

**Step 6** Find the polyhedral-cone \( C^m_x \) and its 1D edges \( L_{\alpha_x} \) using (2.1) and (2.3).
We consider the constraint polyhedron \( \Omega^m \). If \( x^N \in \Omega^m \) and \( c^T x^N \) is the largest one, then let \( ks = 1 \) and go to Step 12.

**Step 7** With (2.5) check \( 2n \) times at most if different rounded-integer point \( x^N \) belongs to \( \Omega^m \). If \( x^N \in \Omega^m \) and \( c^T x^N \) is the largest one, then let \( ks = 1 \) and go to Step 12.

**Step 8** Construct the second-rounded-ball \( O_j^r \) and its bottom plane \( \pi_{v^r} \) of \( \tilde{C}_{n^v}^r \) using (2.2), (2.7) and (2.8). Append the inequality \((a_1^r, a_2^r, \ldots, a_r^r)x \geq \pm 1\) to the constraint set of (1.2).

**Step 9** Record \( kc := kc + 1 \). With (2.5) check if \( x^{1j} \) belongs to \( \Omega^m \), where \( x^{1j} \) is defined by (2.7). If \( x^{1j} \notin \Omega^m \), then let \( kd = 1 \) and go to Step 12.

**Step 10** Make the rounded-minimal-ball \( O_{x^{1j}} \) with rounded-integer points \( x^N \) and radius \( r_{x^{1j}} \). If \( r_{x^{1j}} \leq \varepsilon_0 \), then record \( x^N := x^{1j} \), let \( ks = 1 \) and go to Step 12. With (2.5) check \( 2n \) times at most if different rounded-integer point \( x^N \) belongs to \( \Omega^m \). If \( x^N \in \Omega^m \) and \( c^T x^N \) is the largest one, then let \( ks = 1, ke = 1 \) and go to Step 12.

**Step 11** Let \( ke = 0 \).

**Step 12** If \( kc < n \) and \( kd = 1 \), then let \( kd = 0 \) and go to Step 9. If \( ks = 0 \), then go to Step 16.

**Step 13** Compare \( x^N \) with \( x^H \), if \( x^N \neq x^H \) then perform the procedures similar to (2.11)–(2.13) and record \( x^N := x^N + \lfloor b_p \rfloor e_p \). If \( x^N \) is infinite, then go to exit.

**Step 14** If \( k_n = 0 \) then append the inequality \( c^T x \geq c^T x^N \) to the constraint set of (1.2), else append new inequality instead of old one. If \( c^T x^N > c^T x^H + \varepsilon_0 \) then record \( x^H = x^N \), print \( x^H \) and let \( k_n = k_m = 1 \), else record \( k_n := k_n + 1 \) and if \( x^N \neq x^H \) then record \( k_m := k_m + 1, x^H = x^N \) and print \( x^H \).

**Step 15** Let \( ks = 0 \). If \( k_n > n \), then go to exit.

**Step 16** If \( ke = 1 \) then go to Step 11, else if \( kc = n \) then go to Step 2, else if \( kc = 0 \) then go to Step 8 else go to Step 9.

The conclusion on the algorithm is

**Theorem 2** The algorithm will be terminated in finite steps. And the solutions of integer programming (1.1) are under the thick-whole-point hypothesis or nearly located in the highest plane of polyhedron cut out.

**Proof** We consider the constraint polyhedron \( \Omega^m \) of the relaxation linear programming (1.2). If original \( \Omega^m \) is empty, then the algorithm is stopped at step 2, and the record \( x^H = (-1/(\varepsilon_0 \| c \|_2))c \) means that (1.1) has no solution. So, assume original \( \Omega^m \) is nonempty.

Each time to perform Step 5–Step 13, we append some cutting-planes \( \pi_{v^r} \) to \( \Omega^m \) which cut out a part of \( \Omega^m \) containing fractions near to \( x^* \), and if \( x^N \in \Omega^m \), we also record \( x^H = x^N \) and append an isometric plane \( P_{x^N} = \{ x \in R^n \mid c^T(x - x^N) = 0 \} \) to \( \Omega^m \). The algorithm thus can get vertex series \( x^1, x^2, \ldots, x^i, \ldots \) and whole-point series \( x^{H1}, x^{H2}, \ldots, x^{Hj}, \ldots \) such that

\[
\begin{align*}
    c^T x^1 & \geq c^T x^2 \geq \cdots \geq c^T x^i \geq \cdots, \\
    c^T x^{H1} & \leq c^T x^{H2} \leq \cdots \leq c^T x^{Hj} \leq \cdots.
\end{align*}
\]
And the equality signs cannot always hold on the first sequence of inequalities. Therefore the algorithm will be terminated at finite steps when the constraint polyhedron $\Omega^m$ of (1.2) become empty or $\|c^T(x^{s_i} - x^{H})/\|c\|_2 \leq \epsilon_0$.

If the algorithm stops with $\|x^{s_i} - x^{H}\|_2 \leq \epsilon_0$ and the solution $x^{s_i}$ of the last relaxation programming is unique, then the problem (1.1) has a unique solution.

Generally, if the same isometric plane $c^T x = c^T x^H$ is recorded continually over $n$ times ($k_n > n$), then the integer points ($k_m \geq 1$) are the solutions of (1.1).

The arithmetical-operation quantity making cutting-planes is dependent on mainly steps 6, 8 and 10. In order to solve $n$ linear equations (2.2), we need to find inverse of an $n \times n$ matrix. An $n \times n$ matrix inversion needs $O(n^3)$ arithmetical-operations which is the same estimation with solving linear equations (2.9). To check inequalities (2.5) needs $O(mn^2)$ arithmetical-operations which is like completing an iterative cycle in the isometric plane method [3].

The algorithm need not solve relaxation linear programming (1.2) many times except the original constraint polyhedron $\Omega^m$ of (1.2) is unbounded and the integer programming (1.1) has no solution.

4 NUMERICAL EXAMPLES

In order to investigate the efficiency of the algorithm, we have done some numerical experiments, and obtained satisfactory results. Two typical examples are given as follows. The program is run using Matlab under Windows 98. The running time required on a PC is given in seconds. It includes almost everything, such as the time used to calculate and to print out the results.

**Example 1** Solve the linear equations formed by Hilbert matrix [3]

$$Ax = b, \quad a_{ij} = \frac{1}{i+j}, \quad b_i = \sum_{i=1}^{n} a_{ij}.$$  \hfill (4.1)

The exact solution is $x_i = 1, (i = 1, 2, \ldots)$.

Change (4.1) into the following two forms:

$$\min \sum_{i=1}^{n} x_i, \quad \text{s.t. } Ax \leq b + \xi, \quad x \geq 0.75 e, \quad x \text{ is an integer vector};$$  \hfill (4.2)

and

$$\max \sum_{i=1}^{n} x_i, \quad \text{s.t. } Ax \leq b + \xi, \quad x \geq 0, \quad x \text{ is an integer vector};$$  \hfill (4.3)

here, $e = (1, \ldots, 1)^T$, $\xi = (\xi_1, \xi_2, \ldots)^T$ is a pseudo-random vector in $[0, 1]$.

The problem (4.2) was solved by using the proposed method with $n = 20, 30, \ldots, 100$. For each $n$, the computed solution $x$ equals $e$. And it was found that for each $n$, the relaxation programming was used twice and one isometric plane and one cutting plane were appended to each relaxation programming.

Similarly, the problem (4.3) was solved using the proposed method with $n = 20, 30, \ldots, 70$. The computed solutions had the form $(0, \ldots, 0, w_\xi)^T$ with $w_\xi$ being relative to $\xi$. It was also
found that for each \( n \), the relaxation programming was used four times. In Table II, \( w_0 \) represents the solution of problem (4.3) when \( \xi = 0 \).

**Example 2** Consider a cutting-stock problem \([3]\)

\[
\begin{align*}
\min \quad & z = \sum_{j=1}^{37} x_j, \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, 2, 3, 4, \\
& x_j \geq 0, \quad j = 1, 2, \ldots, 37, \ x: \text{an integer vector}
\end{align*}
\]

where \( b_1 = 97, b_2 = 610, b_3 = 395, b_4 = 211, \) and non-zero \( a_{ij} \) (\( i = 1, 2, 3, 4, 1 \leq j \leq 37 \)) are as follows (see \([3]\)):

\[
\begin{align*}
& a_{11} = 2, \ a_{12} = 1, \ldots, a_{19} = 1; \ a_{22} = 1, \ a_{23} = 1, \ a_{2,10} = 2, \\
& a_{2,11} = 2, \ a_{2,12} = 2, \ a_{2,13} = 1, \ldots, a_{2,21} = 1; \ a_{34} = 1, \ a_{35} = 1, \ a_{3,13} = 2, \ a_{3,14} = 1, \\
& a_{3,15} = 1, \ a_{3,16} = 1, \ a_{3,22} = 3, \ a_{3,23} = 2, \ a_{3,24} = 2, \ a_{3,25} = 2, \ a_{3,26} = 1, \ldots, a_{3,30} = 1; \\
& a_{42} = 1, \ a_{44} = 1, \ a_{46} = 3, \ a_{47} = 2, \ a_{48} = 1, \ a_{4,10} = 2, \ a_{4,11} = 1, \ a_{4,14} = 2, \ a_{4,15} = 1, \\
& a_{4,17} = 4, \ a_{4,18} = 3, \ a_{4,19} = 2, \ a_{4,20} = 1, \ a_{4,23} = 2, \ a_{4,24} = 1, \ a_{4,26} = 4, \ a_{4,27} = 3, \\
& a_{4,28} = 2, \ a_{4,29} = 1, \ a_{4,31} = 7, \ a_{4,32} = 6, \ a_{4,33} = 5, \ a_{4,34} = 4, \ a_{4,35} = 3, \ a_{4,36} = 2, \\
& a_{4,37} = 1.
\end{align*}
\]

Transform (4.4) into the canonical form:

\[
\begin{align*}
\max \quad & -z = -\sum_{j=1}^{37} x_j, \\
\text{s.t.} \quad & \sum_{j=1}^{37} a_{ij}x_j \geq b_i - \epsilon, \quad \epsilon = 10^{-9}, \\
& -\sum_{j=1}^{37} a_{ij}x_j \geq -b_i - \epsilon, \quad i = 1, 2, 3, 4; \\
& x_j \geq 0, \quad j = 1, 2, \ldots, 37, \ x: \text{an integer vector}
\end{align*}
\]

There are many solutions for this problem. Using isometric plane method for the relaxation linear programming, we get a solution: \( x_1 = 13.325, x_2 = 36.125, x_3 = 34.225, x_4 \sim x_9 = 0, x_{10} = 58.925, \ x_{11} = 57.025, \ x_{12} = 55.125, \ x_{13} = 197.5, \ x_{14} \sim x_{37} = 0; \ \max (-z) = -452.25. \) However, the problem (4.5) require that \( x_j \ (j = 1, 2, \ldots, 37) \) must be integer. Using the isometric surface algorithm, we get an integer solution: \( x_1 = 13, x_2 = 36, x_3 = 32, x_4 = 0, x_5 = 3, x_6 \sim x_9 = 0, x_{10} = 59, x_{11} = 57, x_{12} = 57, x_{13} = 196, \)

**TABLE I** Running Time for the Form (4.2) with Solution \( x = e \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds</td>
<td>4.56</td>
<td>7.8</td>
<td>12.74</td>
<td>19.77</td>
<td>29.22</td>
</tr>
</tbody>
</table>

**TABLE II** Running Time and Solution for the Form (4.3).

<table>
<thead>
<tr>
<th>( n )</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_0 )</td>
<td>5.16</td>
<td>6.93</td>
<td>10.43</td>
<td>15.38</td>
<td>21.91</td>
</tr>
<tr>
<td>( w_0 )</td>
<td>44</td>
<td>61</td>
<td>73</td>
<td>86</td>
<td>97</td>
</tr>
</tbody>
</table>

\[TABLE I\] Running Time for the Form (4.2) with Solution \( x = e \).
$x_{14} \sim x_{37} = 0$; $\max(-z) = -453$. The running time is 2.53 seconds, and the relaxation programming time is 2. Since the relaxation programming of (4.5) is a multi-solution problem, the running time did not include the time to solve equations (1.6) for finding a vertex on the highest edge.

**References**


