Journal of the Operations Research
Society of Japan
Vol. 28, No. 2, June 1985



# A COMPUTATIONAL METHOD FOR ALL-INTEGER INTERVAL LINEAR PROGRAMMING

Teruo Sunaga
*Kyushu University*

Alfonso J. Hayakawa M.
*Kyushu University*

Yuji Maruki
*Kyushu University*

(Received August 6, 1983: Final December 13, 1984)

*Abstract* A stable Fractional Cutting Plane method is aimed for solving All-Integer Linear Programming problems with Interval Constraints, in which a modified Interval Linear Programming method and an Interval Width Reduction procedure are availed for computational efficiency. To avoid cycling caused by degeneracy, a simple Lexicographic method is introduced under the concept of perturbation using infinitesimal random numbers. The Optimization problem is separated into a finite set of subproblems to search for integer feasible solutions. Twenty-nine well-known small size test problems are solved; the iterations data are compared with those from the usual Cutting Plane methods.

## 1. Introduction

This paper presents and evaluates a method for solving the Integer Interval Linear Programming (IILP) problem

$$(1.1) \qquad f = \sum_{j=1}^{n} c_j^{\circ} x_j \longrightarrow \text{Max},$$

s.t.

$$(1.2) \qquad l_i \leq x_i \leq u_i, \quad i = 1,\ldots,n,$$

$$(1.3) \qquad l_k \leq \sum_{j=1}^{n} a_{kj}^{\circ} x_k \leq u_k, \quad k = n+1,\ldots,m$$

$$(1.4) \qquad x_j \text{ is an integer}, \quad j = 1,\ldots,n$$

where $a_{kj}^{\circ}$, $c_j^{\circ}$, $l_i$, $u_i$, $l_k$ and $u_k$ are given integers, such that $l_k \leq u_k$,

$k = n+1,\ldots,m$ and $0 \leq l_i \leq u_i$, $i = 1,\ldots,n$. If tighter bounds are not known, we can take $l_i = 0$ and $u_i = M$, where $M$ is a sufficiently large positive integer. An unbounded solution is identified by $x_j = M$ for some $j$ in the optimal solution.

Problem (1.1)-(1.4) can be transferred to an Integer Linear Programming problem with one-sided inequality constraints, but for large values of $(m-n)$, i.e., for many constraints (1.3), the size of the problem increases significantly. Since many real world models have the same (1.1)-(1.4) formulation [1], it is better to use an algorithm for solving this model directly. Therefore, when the integrality condition (1.4) is dropped, the obtained Interval Linear Programming (ILP) problem is solved by the Charnes' ILP method [1,2], to which an Interval Width Reduction procedure is added according to Zionts' Extended Definition method [6] in order to accelerate the solution of the relaxed problem.

Considering the computational difficulties appearing in the solution of Integer Programming problems, and following the current tendency in this field where "composite" algorithms combining the good properties of the available algorithms are the actual goal [3,4], a Search method is proposed, from a computational standpoint.

The framework of this paper is as follows: In §2, the Charnes' Interval Linear Programming method with several modifications is explained. In §3, the application of Gomory's Fractional Cut method to the Integer Interval Linear Programming is given. The Search procedure is explained in §4, and finally in §5 the computational results obtained by the proposed method are compared with those results extracted from reference [5].

## 2. A Modified Interval Linear Programming Method

In this section a summary of Charnes' Interval Linear Programming method with several modifications is presented. The Relaxed ILP problem obtained from (1.1)-(1.4), when the integrality condition (1.4) is eliminated, may be written in matrix form as

$$(2.1) \qquad f = c^{o}\, x \longrightarrow \text{Max},$$

s.t.

$$(2.2) \qquad l \leq A'\, x \leq u$$

where $c^{o}$, $l$, $u$ and $x$ are vectors of conformable dimensions and $A'$ is an $(m \times n)$

matrix such that $m \geq n$. Since $A'$ is a matrix formed by the coefficients in (1.2) and (1.3), $A'$ is a full column rank matrix.

Let the column vector $y = (y_1,\ldots,y_m)^t$ be defined as

$$y = A' x \tag{2.3}$$

or equivalently

$$\begin{aligned} y_i &= x_i , \quad i = 1,\ldots,n , \\ y_i &= \sum_{j=1}^{n} a^{\circ}_{ij} x_j , \quad i = n+1,\ldots,m. \end{aligned} \tag{2.4}$$

Consider any $(n\mathrm{X}n)$ full row rank submatrix $R$ of $A'$ and let $N$ be the remaining submatrix. By rearranging the elements of $y$ so that its first $n$ elements are those associated to $R$, $y$ can be partitioned to $y^R$ and $y^N$ such that

$$y^R = R x , \quad (\text{or } x = R^{-1} y^R) \tag{2.5}$$

and

$$y^N = N x , \quad (\text{or } y^N = NR^{-1} y^R). \tag{2.6}$$

Now, let us define $J^R$ and $J^N$ as follows:

$$J^R = \{i | y_i \in y^R\}$$

and

$$J^N = \{i | y_i \in y^N\}.$$

Further, by

$$c = c^{\circ} R^{-1} = (c_j), \quad j \in J^R \tag{2.7}$$

and

$$A = NR^{-1}, \tag{2.8}$$

the problem (2.1)-(2.2) can be rewritten as

$$f = c\, y^R \longrightarrow \text{Max}, \tag{2.9}$$

s.t.

$$l^R \leq y^R \leq u^R , \tag{2.10}$$

$$l^N \leq A\, y^R \leq u^N \tag{2.11}$$

where $l^R$, $l^N$ $u^R$ and $u^N$ are obtained by rearranging the elements of $l$ and $u$ to match the elements of $y^R$ and $y^N$, and the matrix $A$ is defined as

(2.12) $A = [a_{ij}]_{i \in J^N, j \in J^R}$

which is obtained by (2.8).

$y^R$ is called the Reference vector and expressions (2.9)-(2.10) are called the Reference system, for which its optimal solution $\bar{y}^R = (\bar{y}_j)$, $j \in J^R$ is given by

$$\bar{y}_j = \begin{cases} u_j & \text{if } c_j > 0 , \\ l_j & \text{if } c_j < 0 , \end{cases} \qquad j \in J^R \tag{2.13}$$

and the objective function value $\bar{f}$ is calculated by $\bar{f} = c\bar{y}^R$.

When $c_j = 0$ for some $j \in J^R$ in (2.9), $\bar{y}_j$ is undefined in (2.13), and a stationary cycle may be caused. In order to avoid such a situation, a pseudo-perturbation method is introduced as follows:

Instead of $c_j^o$ in (2.1), we may use

$$c_j^o + \varepsilon\xi_j^o$$

where $\xi_j^o$ is a random number in [-1,1] and $\varepsilon$ is an infinitesimal positive number. However the calculations for $c_j^o$ and $\xi_j^o$ are made independently by adding the row vector $\xi^o = (\xi_1^o, \ldots, \xi_n^o)$ to the calculation table as shown in Table 1, which displays the first calculation table for (2.1)-(2.2). Therefore, $c$ in (2.7) should be redefined as

$$c = (c^o + \varepsilon\xi^o)R^{-1} = c^oR^{-1} + \varepsilon\xi^oR^{-1} ,$$

but since the calculations of $c_j^o$ and $\xi_j^o$ are made independently, we may preserve the definition of $c$ as in (2.7) and define $\xi = \xi^oR^{-1} = (\xi_j)$, $j \in J^R$, which is calculated in the Current Calculation table as shown in Table 2.

Table 1. Starting Calculation Table

| | | | $y_1, \ldots, y_n$ | |
|---|---|---|---|---|
| | | | $\xi^o$ | |
| $f$ | $\bar{f}$ | $l$ \ $c$ | $c^o$ | $u$ |
| $y$ | $\bar{y}$ | $l$ | $I$ / $A^o$ | $u$ |

Table 2. Current Calculation Table

| | | | | |
|---|---|---|---|---|
| | | | $y^R$ | |
| | | | $\xi \circ R^{-1}$ | |
| $f$ | $\bar{f}$ | $l$ \ $c$ | $c \circ R^{-1}$ | $u$ |
| $y$ | $\bar{y}$ | $l$ | $R^{-1}$ | $u$ |
| | | | $A \circ R^{-1}$ | |

Since we should examine $\xi_j$ only when $c_j = 0$ and we can practically assume that $\xi_j$ is not zero, then the optimal solution for the Reference system is now given by

$$\bar{y}_j = \begin{cases} u_j & \text{if } (c_j, \xi_j)^t \succ 0 , \\ l_j & \text{if } (c_j, \xi_j)^t \prec 0 , \end{cases} \qquad j \in J^R \tag{2.14}$$

where $(c_j, \xi_j)^t \succ 0$ and $(c_j, \xi_j)^t \prec 0$ mean lexicographic positiveness and negativeness respectively for the bi-dimensional vectors $(c_j, \xi_j)^t$, $j \in J^R$.

$\bar{y}^R$ is called the Reference solution and it corresponds to a basic solution satisfying the LP optimality condition (see Fig. 1).

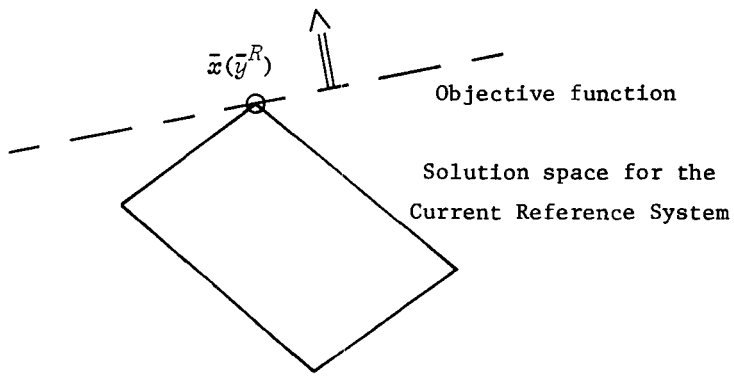


Fig. 1. Optimal Solution of the Reference System.

The corresponding solution $\bar{x}$ for (2.1)-(2.2), according to (2.5), is calculated by

$$\bar{x} = R^{-1}\bar{y}^R \tag{2.15}$$

and $\bar{y}^N$ is given by

(2.16) $\bar{y}^N = A\bar{y}^R$.

If $\bar{y}^N$ satisfies (2.11), $(\bar{y}^R, \bar{y}^N)^t$ is a feasible basic solution satisfying the LP optimality condition and hence $\bar{x}$ is the optimal solution to (2.1)-(2.2). Otherwise, one variable in $y^N$ unsatisfying (2.11) must be selected to enter the Reference vector and one from $y^R$ to leave it.

In order to accelerate the procedure, take up the element $y_k$, $k \in J^N$ for which one of its bounds, $l_k$ or $u_k$, is the most unsatisfied one and this $y_k$ enters the Reference vector. For the selection of the variable to leave the Reference vector, we have two cases when $\bar{y}^N$ is calculated by (2.16).

The surplus case. Here the calculated value $\bar{y}_k$ exceeds its upper bound $u_k$, i.e.,

(2.17) $$\bar{y}_k = \sum_{j \in J^R} a_{kj} \bar{y}_j > u_k .$$

The value of $y_k$ should be reduced in order to achieve feasibility. This is possible only when $a_{kj} > 0$ and $(c_j, \xi_j)^t \succ 0$ $(\bar{y}_j = u_j)$, and when $a_{kj} < 0$ and $(c_j, \xi_j)^t \prec 0$ $(\bar{y}_j = l_j)$. Therefore, pick up the lexicographic positive bi-dimensional ratio vectors

(2.18) $$(c_j/a_{kj}, \xi_j/a_{kj})^t \succ 0 , \quad j \in J^R$$

and arranging them in an increasing lexicographic order, the following series of $j$ is obtained:

(2.19) $$j(1), j(2), \ldots, j(l), \ldots .$$

Next, the least index $p$, which satisfies

(2.20) $$\sum_{l=1}^{p} |a_{kj(l)}| (u_{j(l)} - l_{j(l)}) \geq \bar{y}_k - u_k$$

determines the variable $y_{j(p)}$ to leave the Reference vector for this case.

The slack case. The calculated value $\bar{y}_k$ falls below its lower bound $l_k$, i.e.,

(2.21) $$\bar{y}_k = \sum_{j \in J^R} a_{kj} \bar{y}_j < l_k .$$

In a similar way as in the above case, we pick up the lexicographic negative ratio vectors

(2.22) $$(c_j/a_{kj}, \xi_j/a_{kj})^t \prec 0 , \quad j \in J^R$$

and arranging them in an increasing lexicographic order with respect to their absolute values, we get the series of $j$ given by (2.19).

Next, the least index $p$, which satisfies

$$\sum_{l=1}^{p} |a_{kj(l)}|(u_{j(l)} - l_{j(l)}) \geq l_k - \bar{y}_k \tag{2.23}$$

determines the variable $y_{j(p)}$ to leave the Reference vector.

In either the slack or the surplus cases, if the index $p$ satisfying (2.20) or (2.23) can not be found, then (2.1)-(2.2) is infeasible. Otherwise, $y_k$ enters the Reference vector and $y_{j(p)}$ leaves it. Next, execute an exchange operation in the calculation table by pivoting on element $a_{kj(p)}$ and a new Reference system is obtained. Following this procedure iteratively, a Reference system whose optimal solution satisfies all the constraints (2.11) is obtained, or infeasibility is found. A typical two-dimensional solution space for the new Reference system is as shown in Fig. 2.

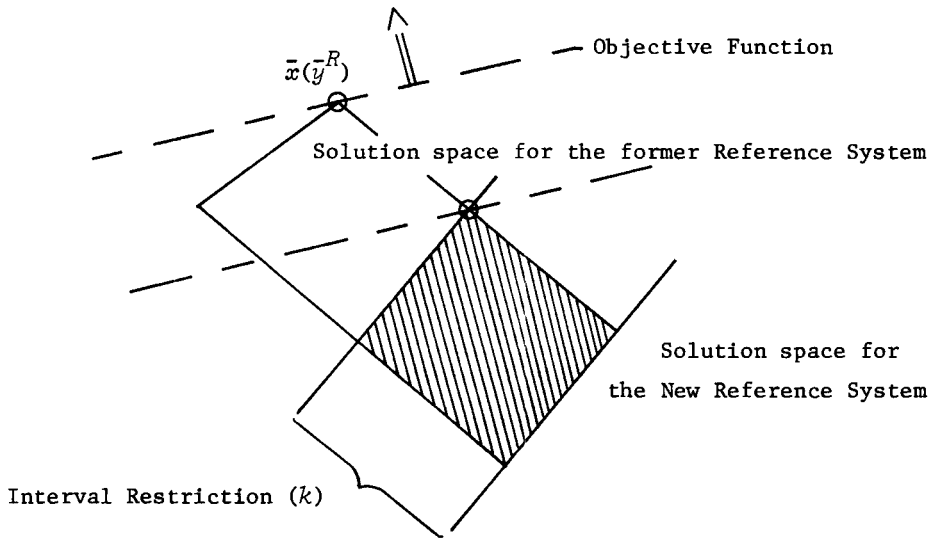


Fig. 2. Solution space for the New Reference System.

**Infeasibility test.** Before a new Reference system is obtained, an infeasibility test must be executed as follows:

Let us define

$$a^{+}_{ij} = \begin{cases} a_{ij} & \text{if } a_{ij} \geq 0, \\ 0 & \text{if } a_{ij} < 0, \end{cases} \qquad i \in J^N, \quad j \in J^R \tag{2.24}$$

and

$$(2.25)\qquad a_{ij}^{-} = \begin{cases} 0 & \text{if } a_{ij} > 0 , \\ a_{ij} & \text{if } a_{ij} \le 0 , \end{cases} \qquad i \in J^N, \quad j \in J^R.$$

When $y_j$, $j \in J^R$ takes such values as

$$(2.26)\qquad y_j = \begin{cases} u_j & \text{if } a_{ij} > 0 , \\ l_j & \text{if } a_{ij} < 0 , \end{cases} \qquad i \in J^N, \quad j \in J^R,$$

we may write

$$(2.27)\qquad (y_i)^{+} = \sum_{j \in J^R} a_{ij}^{+} u_j + \sum_{j \in J^R} a_{ij}^{-} l_j , \quad i \in J^N,$$

and

$$(2.28)\qquad (y_i)^{-} = \sum_{j \in J^R} a_{ij}^{+} l_j + \sum_{j \in J^R} a_{ij}^{-} u_j , \quad i \in J^N$$

where $(y_i)^{+}$ and $(y_i)^{-}$ are the maximum and minimum of $y_i$ respectively with respect to the Current Reference system.

Then, if

$$(y_i)^{+} < l_i , \quad i \in J^N$$

or

$$(y_i)^{-} > u_i , \quad i \in J^N,$$

the system (2.9)-(2.11), and therefore the system (2.1)-(2.2), is infeasible and the procedure finishes (see Fig. 3).

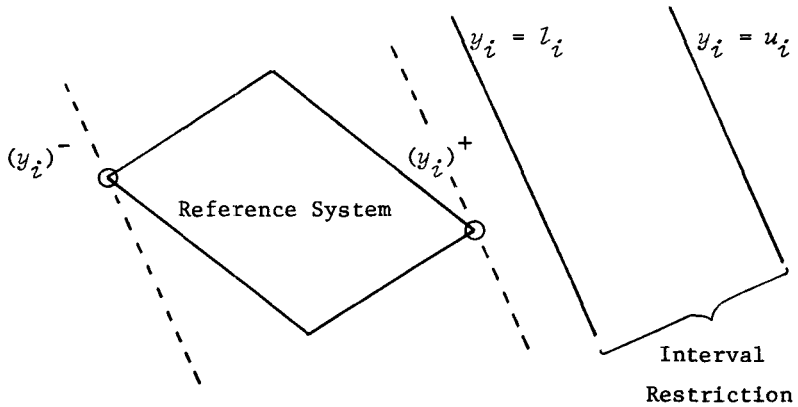


Fig. 3. Infeasibility decision

## The outline of the algorithm

We start the procedure with the following table:

Table 1.1 Calculation Table before the Pivot Iteration.

| | | | $y_1$ | . . . . | $y_n$ | |
|---|---|---|---|---|---|---|
| | | | $\xi_1^\circ$ | . . . . | $\xi_n^\circ$ | |
| | | | $c_1^\circ$ | . . . . | $c_n^\circ$ | |
| $y_1$ | $\bar{y}_1$ | $l_1$ | 1 | . . . . | 0 | $u_1$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ ⋮ ⋮ | ⋮ | ⋮ |
| $y_n$ | $\bar{y}_n$ | $l_n$ | 0 | . . . . | 1 | $u_n$ |
| $y_{n+1}$ | $\bar{y}_{n+1}$ | $l_{n+1}$ | $a^\circ_{n+1,1}$ | . . . . | $a^\circ_{n+1,n}$ | $u_{n+1}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ ⋮ ⋮ | ⋮ | ⋮ |
| $y_m$ | $\bar{y}_m$ | $l_m$ | $a_{m,1}$ | . . . . | $a_{m,n}$ | $u_m$ |

Let $R^r$ and $N^r$ be the basic and non-basic matrices respectively, after the $r$-th iteration. The first basis $R^\circ$ is the $(n\mathrm{X}n)$ unit matrix in the upper part of Table 1.1. Then $y^R = (y_1,\ldots,y_n)^t$ and $y^N = (y_{n+1},\ldots,y_m)^t$ are the Reference and Non-Reference vectors for $R^\circ$.

First, check the infeasibility by calculating $(y_i)^+$ and $(y_i)^-$ for $i \in J^N$ by (2.27) and (2.28) respectively. If $(y_i)^+ < l_i$ or $(y_i)^- > u_i$ for some $i \in J^N$, then the problem is infeasible and the procedure finishes. Otherwise, the reference solution $\bar{y}^R$ is read from the basic rows, as follows: If $c_j^\circ > 0$ then $\bar{y}_j$ is read from the right-hand side on Table 1.1, and if $c_j^\circ < 0$ then $\bar{y}_j$ is read from the left-hand side. If $c_j^\circ = 0$ then use $\xi_j^\circ$ instead of $c_j^\circ$ to decide the value of $\bar{y}_j$.

Next, calculate $\bar{y}^N$ by multiplying the non-basic row vectors by $\bar{y}^R$. If all $y_i$, $i \in J^N$ satisfy their bounds, then $\bar{x} = (R^\circ)^{-1}\bar{y}^R$ is the optimal solution for (2.1)-(2.2). Otherwise, select the variable $y_k$, $k \in J^N$ for which one of its bounds is the most unsatisfied and it is the variable to enter the Reference vector.

In order to select the variable to leave the Reference vector, compute the index $p$ for the corresponding surplus or slack case as explained above. $y_{j(p)}$ is then the variable to leave the Reference vector.

The Pivot element is $a^\circ_{k,j(p)}$, but since $J^R = \{1,\ldots,n\}$ and $J^N = \{n+1,\ldots,m\}$ in Table 1.1, we may assume that the pivot element is $a^\circ_{n+k,p}$.

Now, divide the $p$-column by the pivot element. Add the $p$-column multiplied by $-a^\circ_{n+k,1}$ to the first column, add the $p$-column multiplied by $-a^\circ_{n+k,2}$

to the second column, etc.

Table 2.1 Calculation Table after the Pivot Iteration.

| | | | $y_1$ | ... | $y_p$ | ... | $y_n$ | |
|---|---|---|---|---|---|---|---|---|
| | | | $\xi^\circ_1 - a^\circ_{n+k,1}\dfrac{\xi^\circ_p}{a^\circ_{n+k,p}}$ | ... | $\dfrac{\xi^\circ_p}{a^\circ_{n+k,p}}$ | ... | $\xi^\circ_n - a^\circ_{n+k,n}\dfrac{\xi^\circ_p}{a^\circ_{n+k,p}}$ | |
| | | | $c^\circ_1 - a^\circ_{n+k,1}\dfrac{c^\circ_p}{a^\circ_{n+k,p}}$ | ... | $\dfrac{c^\circ_p}{a^\circ_{n+k,p}}$ | ... | $c^\circ_n - a^\circ_{n+k,n}\dfrac{c^\circ_p}{a^\circ_{n+k,p}}$ | |
| $y_1$ | $\bar{y}_1$ | $l_1$ | 1 | ... | 0 | ... | 0 | $u_1$ |
| $y_p$ | $\bar{y}_p$ | $l_p$ | $-\dfrac{a^\circ_{n+k,1}}{a^\circ_{n+k,p}}$ | ... | $\dfrac{1}{a^\circ_{n+k,p}}$ | ... | $-\dfrac{a^\circ_{n+k,n}}{a^\circ_{n+k,p}}$ | $u_p$ |
| $y_n$ | $\bar{y}_n$ | $l_n$ | 0 | ... | 0 | ... | 1 | $u_n$ |
| $y_{n+1}$ | $\bar{y}_{n+1}$ | $l_{n+1}$ | $a^\circ_{n+1,1} - a^\circ_{n+k,1}\dfrac{a^\circ_{n+1,p}}{a^\circ_{n+k,p}}$ | ... | $\dfrac{a^\circ_{n+1,p}}{a^\circ_{n+k,p}}$ | ... | $a^\circ_{n+1,n} - a^\circ_{n+k,n}\dfrac{a^\circ_{n+1,p}}{a^\circ_{n+k,p}}$ | $u_{n+1}$ |
| $y_{n+k}$ | $\bar{y}_{n+k}$ | $l_{n+k}$ | 0 | ... | 1 | ... | 0 | $u_{n+k}$ |
| $y_m$ | $\bar{y}_m$ | $l_m$ | $a^\circ_{m,1} - a^\circ_{n+k,1}\dfrac{a^\circ_{m,p}}{a^\circ_{n+k,p}}$ | ... | $\dfrac{a^\circ_{m,p}}{a^\circ_{n+k,p}}$ | ... | $a^\circ_{m,n} - a^\circ_{n+k,n}\dfrac{a^\circ_{m,p}}{a^\circ_{n+k,p}}$ | $u_m$ |

The table obtained after applying the above pivot operation is as shown in Table 2.1, where the new solution $\bar{y}^R$ is read in the same fashion as before.

Observe that in the calculation table the elements of $y$ are not rearranged and the pivot operation affects only the coefficients' matrix and the row vectors $\xi^\circ$ and $c^\circ$. The bounds are not changed because the table has the form

$$l \le y \le u$$

where the values of $y_i$, $i = 1,\ldots,m$ are moving toward feasibility, and any change in the bounds means a redefinition of the original problem.

## 2.1 Interval width reduction procedure

This procedure was introduced by Zionts [6] in order to accelerate the solution of integer LP problems. Availing the calculation of (2.27) and (2.28) in the infeasibility test explained above, we have that, if

(2.29) $\quad (y_i)^+ < u_i \ , \quad i \in J^N \ ,$

then $(y_i)^+$ defines a new and tighter bound for the variable $y_i$, because the maximum feasible value $(y_i)^+$ for the variable $y_i$ is smaller than its upper bound $u_i$, which becomes loose and $u_i$ can be adjusted easily.

And, if

(2.30) $$(y_i)^- > l_i \ , \quad i \in J^N \ ,$$

then $(y_i)^-$ becomes the corresponding new lower bound for variable $y_i$, $i \in J^N$.

Since $y_i$, $i = 1,\ldots,m$, are integer variables for the all-integer case, the upper and lower bounds are redefined by

$$u_i \longrightarrow u_i' = [(y_i)^+] \quad \text{if (2.29) is fulfilled}$$

and

$$l_i \longrightarrow l_i' = ](y_i)^-[ \quad \text{if (2.30) is fulfilled}$$

where

$$[\alpha] = \text{Max}\{w \mid w \leq \alpha \text{ and } w \text{ is integer}\}$$

and

$$]\alpha[ = \text{Min}\{w \mid w \geq \alpha \text{ and } w \text{ is integer}\}.$$

For future computations, only the tightest bounds for each variable need be remembered. The interval width reduction means a substantial reduction in the set of feasible solutions for the relaxed problem as shown in Fig. 4.

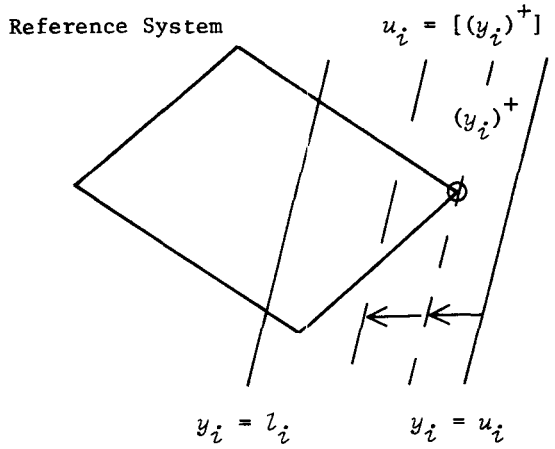


Fig. 4 Interval Width Reduction

Some simple examples can be solved by simply utilizing this procedure as shown in the following example:

Example 1

$$f = x_1 + x_2 \rightarrow \text{Max},$$

s.t.

$$0 \le x_1 - 2x_2 \le 2,$$
$$0 \le x_1 + 4x_2 \le 2,$$
$$x_1, x_2 = 0 \text{ or } 1 .$$

The relaxed problem is tabulated as shown in Table 3A.

The optimal solution for the first reference system is $\bar{y} = (\bar{y}_1,\ldots,\bar{y}_n)^t = (1,1,-1,5)$.

The maximum value of $y_3$ within the reference system, according to (2.27), is $(y_3)^+ = 1$. Hence the interval reduction $u_3 \rightarrow 1$ is performed.

At the optimum, $y_4 = 5 > 2 = u_4$ and it is the most unsatisfied restriction.

Since the positive ratios of the 4th row are $c_2/a_{42} = 1/4 < c_1/a_{41} = 1/1$, the pivot element is $a_{42}$ .

Table 3B is the resulting table after the elimination calculation. The optimal solution of the second reference system is $\bar{y} = (1,0.25,0.5,2)$ and satisfies the other interval restrictions. However, the maximum value of $y_2$, within the reference system is $(y_2)^+ = 0.5$ and then the interval reduction $u_2 \rightarrow 0$ is performed because $y_2$ is restricted to be integer.

Since $y_2$ does not satisfy the new bound $u_2 = 0$ and the unique positive ratio is $c_4/a_{24} = .25/.25 > 0$, the next pivot element is $a_{24}$ and Table 3C is obtained. From Table 3C, it can be seen that the optimal solution of the problem is $\bar{y} = (1,0,1,1)$, or $\bar{x} = (1,0)$.

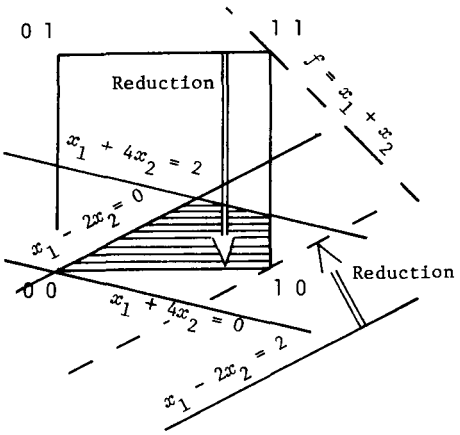


Fig. 5 Geometric explanation

Table 3A

| | | | $y_1$ | $y_2$ | |
|---|---|---|---|---|---|
| $f$ | 2 | $l$ \ $c$ | 1 | 1 | $u$ |
| $y_1$ | 1 | 0 | 1 | 0 | 1 |
| $y_2$ | 1 | 0 | 0 | 1 | 1 |
| $y_3$ | -1 | -1 | 1 | -2 | 2 → 1 |
| $y_4$ | 5 | 0 | 1 | 4 | 2 |

Table 3B

| | | | $y_1$ | $y_4$ | |
|---|---|---|---|---|---|
| $f$ | 1.25 | $l$ \ $c$ | 0.75 | 0.25 | $u$ |
| $y_1$ | 1 | 0 | 1 | 0 | 1 |
| $y_2$ | 0.25 | 0 | -0.25 | 0.25 | 1 → 0 |
| $y_3$ | 0.5 | 0 | 1.5 | -0.5 | 1 |
| $y_4$ | 2 | 0 | 0 | 1 | 2 |

Table 3C

| | | | $y_1$ | $y_2$ | |
|---|---|---|---|---|---|
| $f$ | 1 | $l$ \ $c$ | 1 | 1 | $u$ |
| $y_1$ | 1 | 0 | 1 | 0 | 1 |
| $y_2$ | 0 | 0 | 0 | 1 | 0 |
| $y_3$ | 1 | 0 | 1 | -2 | 1 |
| $y_4$ | 1 | 0 | 1 | 4 | 2 |

The geometric explanation of the calculation procedure is shown in Fig. 5. The perturbation method is not used in this example; it will be explained in a later example.

The method explained in this section will be called the Modified Interval Linear Programming (MILP) method for future references.

## 3. Gomory's Fractional Cut for IILP

This section explains the application of Gomory's Fractional Cut to Integer Interval Linear Programming.

We suppose that (2.1)-(2.2), the relaxed problem of (1.1)-(1.4), is solved by the MILP method explained in Section 2 and that its optimal solution $\tilde{x}$ in terms of $\tilde{y} = (\tilde{y}^R, \tilde{y}^N)^t$ and the coefficients $\tilde{c}_j$'s and $\tilde{a}_{ij}$'s corresponding to the last reference system (2.9)-2.10) are available.

Since $\tilde{y}^R$ is always integer, then the integrality of $\tilde{y}^N$ is the necessary and sufficient condition in order for $\tilde{x} = (\tilde{y}_1, \ldots, \tilde{y}_n)^t$ to become the optimal solution to (1.1)-(1.4).

If $\tilde{y}^N$ is not integer, select one element $y_k$, $k \in J^N$ whose value $\tilde{y}_k$ is not integer; the equation associated to $y_k$ will be referred to as the Source Row.

Consider the fractional parts $f_k$ and $f_{kj}$ such that

$$\tilde{y}_k = [\tilde{y}_k] + f_k \ , \quad 0 \le f_k < 1 \tag{3.1}$$

and

$$\tilde{a}_{kj} = [\tilde{a}_{kj}] + f_{kj} \ , \quad 0 \le f_{kj} < 1 \ , \quad j \in J^R \ . \tag{3.2}$$

Now from $y_k = \sum_{j \in J^R} \tilde{a}_{kj} y_j$, $\tilde{y}_k = \sum_{j \in J^R} \tilde{a}_{kj} \tilde{y}_j$ and the expressions (3.1)-(3.2), we have that

$$y_k = \sum_{j \in J^R} ([\tilde{a}_{kj}] + f_{kj})(y_j - \tilde{y}_j) + [\tilde{y}_k] + f_k \ . \tag{3.3}$$

Let us define $J^+$ and $J^-$ as follows:

$$J^+ = \{j \mid j \in J^R \text{ and } (c_j, \xi_j)^t > 0\}$$

and

$$J^- = \{j \mid j \in J^R \text{ and } (c_j, \xi_j)^t < 0\},$$

then from (3.3) we obtain the next expression

$$\begin{aligned} y_k - [\tilde{y}_k] - \sum_{j \in J^-} [\tilde{a}_{kj}](y_j - \tilde{y}_j) - \sum_{j \in J^+} ([\tilde{a}_{kj}] + 1)(y_j - \tilde{y}_j) \\ = \sum_{j \in J^-} f_{kj}(y_j - \tilde{y}_j) + \sum_{j \in J^+} (f_{kj} - 1)(y_j - \tilde{y}_j) + f_k \ . \end{aligned} \tag{3.4}$$

In order for $y_k$ and $y_j$, $j \in J^R$ to be integers, a necessary condition is that the right-hand side of (3.4) must also be integer.

From the determination of $\tilde{y}_j$, $j \in J^R$ by (2.14), it is easy to verify that if $j \in J^+$ then $(y_j - \tilde{y}_j) \leq 0$ and if $j \in J^-$ then $(y_j - \tilde{y}_j) \geq 0$. Hence we may write

$$(3.5) \qquad \sum_{j \in J^-} f_{kj}(y_j - \tilde{y}_j) + \sum_{j \in J^+} (f_{kj} - 1)(y_j - \tilde{y}_j) + f_k > f_k > 0$$

and the necessary condition for integrality becomes

$$(3.6) \qquad \sum_{j \in J^-} f_{kj}(y_j - \tilde{y}_j) + \sum_{j \in J^+} (f_{kj} - 1)(y_j - \tilde{y}_j) + f_k \geq 1 \ .$$

So, we get the Interval Fractional cut

$$(3.7) \qquad 1 - f_k + \sum_{j \in J^-} f_{kj}\tilde{y}_k + \sum_{j \in J^+}(f_{kj} - 1)\tilde{y}_j + \sum_{j \in J^-} f_{kj}y_j + \sum_{j \in J^+}(f_{kj} - 1)y_j \leq M$$

which is added to the current system (2.9)-(2.11). The expression (3.7) shall be referred to as an F-cut.

The left-han side of (3.7) must be an integer because all the interval bounds, $l_i$'s and $u_i$'s, must be integers in the initial calculation table. Hence we must show that

$$- f_k + \sum_{j \in J^-} f_{kj}\tilde{y}_j + \sum_{j \in J^+}(f_{kj} - 1)\tilde{y}_j$$

is an integer.

From expression (3.1) and (3.2), we have

$$\tilde{y}_k = [\tilde{y}_k] + f_k = \sum_{j \in J^R} \tilde{a}_{kj}\tilde{y}_j = \sum_{j \in J^R} [\tilde{a}_{kj}]\tilde{y}_j + \sum_{j \in J^R} f_{kj}\tilde{y}_j$$

and thus

$$[\tilde{y}_k] - \sum_{j \in J^R} [\tilde{a}_{kj}]\tilde{y}_j - \sum_{j \in J^+} \tilde{y}_j = - f_k + \sum_{j \in J^-} f_{kj}\tilde{y}_j + \sum_{j \in J^+} (f_{kj} - 1)\tilde{y}_j \ .$$

Since the left-hand side is an integer, (3.8) must be an integer.

After adding an F-cut to the current system, it is reoptimized by the MILP method and a new solution $\tilde{x}$ is obtained.

By applying iteratively the procedure explained above, after a finite number of iterations the optimal solution to (1.1)-(1.4) should be reached. However, this method has the same numerical troubles as various cutting plane methods in Integer Linear Programming. Hence it is combined into a Search procedure, whose purpose is to improve the computational efficiency.

Source row selection

After we have obtained a continuous solution $(\tilde{y}^R, \tilde{y}^N)^t$ with $\tilde{y}^N$ being not integer, a different F-cut (3.7) can be derived from each row in the current table corresponding to $y_i$ whose value is not integer. Since the rate of convergence to the integer solution depends on the Source Row selection, we describe an useful criterion to use in that selection.

Let assume there are variables $y_q$, $q = i_1,\ldots,i_Q$ whose values are not integers. From the necessary condition for integrality (3.6), we may write the hyperplane

$$(3.9)\quad \sum_{j\in J^-} (f_{qj}/(1 - f_q))(y_j - \tilde{y}_j) + \sum_{j\in J^+} ((f_{qj} - 1)/(1 - f_q))(y_j - \tilde{y}_j) = 1$$

from which we may construct an F-cut for every $y_q$, $q = i_1,\ldots,i_Q$ .

Since the vector $c$ is a normal vector to the objective function (2.9), then $y^R = (y_j)_{j\in J^R} = \tilde{y}^R + d\, c/|c|$ is the line through $y^R$ parallel to $c$. This line intersects all the $Q$ hyperplanes (3.9). Let

$$(3.10)\quad y_q^R = \tilde{y}^R + d_q c/|c|$$

be the intersection points, where $d_q$ is a real number representing the distance between the points $\tilde{y}^R$ and $y_q^R$ along the line mentioned (see Fig. 6).

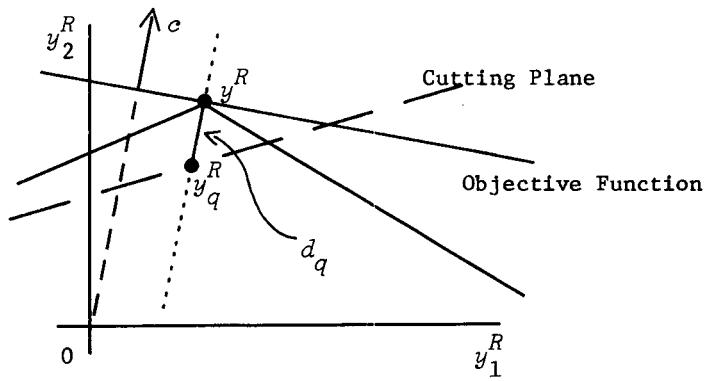


Fig. 6 Depth of a Cut.

Substituting (3.10) into (3.9), we get the expression

$$d_q = |c|/(\sum_{j\in J^-} c_j f_{qj}/(1 - f_q) + \sum_{j\in J^+} c_j(f_{qj} - 1)/(1 - f_q)) .$$

Then, the Source Row Selection rule can be written as

$$\text{Max}\{|d_q|\},\ q = i_1,\ldots,i_Q .$$

The effort required by this criterion to select the new source row can be justified by the improvement in the rate of convergence (see Section 5).

## 4. The Search Procedure

In this section we describe the Search procedure, which uses the IILP method described in sections 2 and 3. As mentioned in Section 1, the purpose of the Search procedure is to solve problem (1.1)-(1.4). The procedure consist of two steps.

First, consider that the relaxed problem (1.1)-(1.3) is solved by the MILP method and its continuous solution $\tilde{x}$ is obtained, now let us define

$$f^+ = c^\circ \tilde{x} \quad \text{and} \quad f^\circ = [f^+]$$

which are the continuous and the integer upper bounds for $f$ in (1.1) respectively.

The continuous lower bound $f^-$ for $f$ in (1.1) can be calculated by using

$$f' = c^\circ x \longrightarrow \text{Min},$$

instead of (1.1). Now let $x_{min}$ be the optimal solution to the minimization problem, and $f^-$ is defined as

$$f^- = c^\circ x_{min} .$$

### Step I

The purpose of this step is to find an upper bound $f^\circ$. Since (1.1)-(1.3) is a bounded problem, the MILP method finds its finite optimal solution or its infeasibility. If $\tilde{x}$ is obtained, the upper bound $f^\circ$ is at hand and we proceed to Step II. Otherwise infeasibility is detected, the problem (1.1)-(1.4) is also infeasible and the procedure finishes, because continuous infeasibility implies integer infeasibility.

### Step II

In this step we search for the integer optimal solution to problem (1.1)-(1.4). Let $S$ be the set of feasible solutions for problem (1.1)-(1.4), i.e.,

$$S = \{x \mid l \leq A' x \leq u \text{ and } x \text{ is integer}\}.$$

Now consider the subsets $S_t$ of $S$

$$S_t = \{x \mid l \leq A' x \leq u,\ c^\circ x = f_t \text{ and } x \text{ is integer } \}$$

where $f_t$ is an integer such that $f_1 = f^\circ$ and $f_t = f_{t-1} - 1$, for $t = 2,3,\ldots,T$, and $f_T$ is an integer lower bound for $f$ in (1.1), defined as $f_T = ]f^-[$.

Since the elements in $c^\circ$ are all-integer, then

$$S = \bigcup_{t=1}^{T} S_t$$

and the optimal solution to (1.1)-(1.4), if it exists, must be in some subset $S_t$, $t = 1,\ldots,T$.

We shall examine the subsets $S_t$ in sequence. The purpose of this examination is to get a tighter upper bound $f^\circ$, which is now $f^\circ = f_1$. If we may show that the subset $S_1$ does not contain any integer feasible solution, then we set $f^\circ = f_2$, if the subset $S_2$ does not dontain any integer feasible solution, then $f^\circ = f_3$, and so on. Therefore, when an integer feasible solution is found, this solution is the optimal one for problem (1.1)-(1.4).

In order to examine the subsets $S_t$, we may use the IILP method. For this purpose, consider the sequence of artificial optimization subproblems $P_1,\ldots,P_T$, where the subproblem $P_t$ is

(4.1) $\quad F = g(x_1,\ldots,x_n) \longrightarrow \text{Max},$

s.t.

(4.2) $\quad l_i \leq x_i \leq u_i\ ,\quad i = 1,\ldots,n,$

(4.3) $\quad l_k \leq \sum_{j=1}^{n} a^\circ_{kj} x_j \leq u_k\ ,\quad k = n+1,\ldots,m,$

(4.4) $\quad f_t \leq \sum_{j=1}^{n} c^\circ_j x_j \leq f_t\ ,$

(4.5) $\quad x_j$ is an integer $\quad ,\quad j = 1,\ldots,n$

where $g(x_1,\ldots,x_n)$ is any linear function with all-integer coefficients. For instance: $g(x) = x_i$ or $g(x) = \sum_{j=1}^{n} a^\circ_{hj} x_j$, for some $h = n+1,\ldots,m$.

Subproblems $P_1,\ldots,P_T$ are solved in sequence by the IILP method, which has only two possible results, integer feasibility or integer infeasibility. Consider the subproblem $P_t$; if an integer feasible solution is found, this solution is the optimal one for problem (1.1)-(1.4) and the procedure finishes. Otherwise integer infeasibility is found, it means that all the continuous feasible solutions in the hyperplane $c^\circ x = f_t$ were cut off by the cutting planes. Therefore, we proceed to the next subproblem $P_{t+1}$.

Since subproblems $P_1,\ldots,P_T$ are artificial optimization problems to find their integer feasible solutions, we may introduce the following procedure in order to accelerate the search of integer feasibility.

Since the cutting planes lose effectiveness gradually, for some problems even though $\tilde{x}$ is near to the optimal solution $x^*$, it still takes a long time until $x^*$ is recognized. Hence, we examine the feasibility of the rounded solution $z$, which is defined as

$$z_i = \begin{cases} [\tilde{x}_i] + 1 & \text{if } f_i' \geq 0.5 \\ [\tilde{x}_i] & \text{if } f_i' < 0.5 \end{cases}$$

where $f_i'$ is

$$\tilde{x}_i = [\tilde{x}_i] + f_i' \ ; \quad 0 \leq f_i' < 1.$$

If $z$ satisfies (4.3) and (4.4), then $z$ is an integer feasible solution and the procedure finishes. Otherwise, we proceed to introduce a new cutting plane to search for a new continuous solution $\tilde{x}$.

## Algorithm

STEP I. Solve the relaxed problem (1.1)-(1.3) by the MILP method in Section 2, here cutting planes are not used. One of the following three cases is obtained:

i) INTEGRALITY. The optimal solution $\tilde{x}$ for the relaxed problem is integer. Hence $\tilde{x}$ is the optimal solution to (1.1)-(1.4) and the procedure finishes.

ii) INFEASIBILITY The relaxed problem is infeasible. Since continuous infeasibility means integer infeasibility, the procedure finishes and (1.1)-(1.4) is found infeasible.

iii) CONTINUOUS FEASIBILITY. $\tilde{x}$ is not integer. Then

$$f^+ = \sum_{j=1}^{n} c_j^\circ \tilde{x}_j$$

is an upper bound for $f$ in (1.1). Solve the minimization problem mentioned above, whose solution provides $f^-$. Set $t = 1$ and go to step II.

STEP II. Solve subproblem $P_t$, i.e., problem (4.1)-(4.5) by the IILP method. As described previously, evaluate the rounded solution $z$ from every continuous solution $\tilde{x}$ obtained. Here one of the following two cases becomes:

i) INTEGER FEASIBILITY. An integer solution $x^*$ for (4.1)-(4.5) is obtained. Since $f^* = c^\circ x^* = f_t$ is an upper bound for $f$ in (1.1), $x^*$ is the optimal solution to (1.1)-(1.4).

ii) INTEGER INFEASIBILITY. The subproblem $P_t$ is found to be integer infeasible. If $f_t$ is equal to the lower bound $f_T$, the procedure finishes and (1.1)-(1.4) is infeasible. Otherwise, $f_{t+1} = f_t - 1$ becomes a new upper bound for $f$, because for higher values of $f$ there is no integer feasible solution. Set $t = t + 1$ and go to the starting point of step II.

Example 2

Consider the problem

$$f = 5x_1 + 13x_2 + 5x_3 + 4x_4 + 3x_5 + 7x_6 \longrightarrow \text{Max},$$

s.t.

$$3x_1 + 4x_2 + 16x_3 + 7x_4 + 7x_5 + 6x_6 \leq 25 ,$$

$$x_i = 0 \text{ or } 1, \quad i = 1,\ldots,6 .$$

STEP I, The continuous relaxed problem is formulated as an ILP problem and solved by the MILP method. The continuous solution $\tilde{x}$ is obtained and the objective function is $f^+ = 31.143$.

STEP II. Consider the subproblem $P_t$

$$F = x_i \longrightarrow \text{Max},$$

s.t.

$$0 \leq x_i \leq 1 , \quad i = 1,\ldots,6,$$

$$0 \leq 3x_1 + 4x_2 + 16x_3 + 7x_4 + 7x_5 + 6x_6 \leq 25 ,$$

$$f_t \leq 5x_1 + 13x_2 + 5x_3 + 4x_4 + 3x_5 + 7x_6 \leq f_t ,$$

$x_i$ is an integer, $i = 1,\ldots,6$.

Searching for integer feasibility in $f_1 = f^\circ = 31$ and $f_2 = 30$, by the IILP method, we find that both subproblems $P_1$ and $P_2$ are integer infeasible. Hence, following the Search procedure, subproblem $P_3$ with $f_3 = 29$ is examined.

Table 4A shows the optimal continuous solution for subproblem $P_3$, where in the last row a cut is added because the rounded solution $z = (1,1,0,1,1,1)$ is infeasible. Finally, Table 4B shows the integer optimal solution.

Table 4A

| | | | | $y_1$ | $y_8$ | $y_3$ | $y_4$ | $y_5$ | $y_2$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\xi$ | 0.208 | 0.077 | -.407 | 0.439 | 0.620 | -.268 | |
| | $F$ | 1 | $l$ \ $c$ | 1 | 0 | 0 | 0 | 0 | 0 | $u$ |
| $x_1$ | $y_1$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| $x_2$ | $y_2$ | 1 | 1* | 0 | 0 | 0 | 0 | 0 | 1 | 1* |
| $x_3$ | $y_3$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $x_4$ | $y_4$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $x_5$ | $y_5$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| $x_6$ | $y_6$ | 4/7 | 0 | -5/7 | 1/7 | -5/7 | -4/7 | -3/7 | -13/7 | 1 |
| | $y_7$ | 171/7 | 17 | -9/7 | 6/7 | 82/7 | 25/7 | 31/7 | -50/7 | 25[1] |
| | $y_8$ | 29 | 29* | 0 | 1 | 0 | 0 | 0 | 0 | 29* |
| | $y_9$ | -32/7 | -4 | -2/7 | -1/7 | 5/7 | -3/7 | -4/7 | 6/7 | 1000[2] |

* Indicates null interval width

[1] Source row

[2] Cut

Table 4B

| | | | | $y_1$ | $y_8$ | $y_9$ | $y_3$ | $y_6$ | $y_2$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\xi$ | -.051 | -.104 | -1.161 | 0.496 | 0.102 | 0.917 | |
| | $F$ | 1 | $l$ \ $c$ | 1 | 0 | 0 | 0 | 0 | 0 | $u$ |
| $x_1$ | $y_1$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| $x_2$ | $y_2$ | 1 | 1* | 0 | 0 | 0 | 0 | 0 | 1 | 1* |
| $x_3$ | $y_3$ | 0 | 0* | 0 | 0 | 0 | 1 | 0 | 0 | 0* |
| $x_4$ | $y_4$ | 1 | 1* | -2 | 1 | 3 | -5 | -4 | -10 | 1* |
| $x_5$ | $y_5$ | 0 | 0* | 1 | -1 | -4 | 5 | 3 | 9 | 0* |
| $x_6$ | $y_6$ | 1 | 1* | 0 | 0 | 0 | 0 | 1 | 0 | 1* |
| | $y_7$ | 20 | 17 | -4 | 0 | -7 | 16 | -1 | -3 | 23 |
| | $y_8$ | 29 | 29* | 0 | 1 | 0 | 0 | 0 | 0 | 29* |
| | $y_9$ | -4 | -4 | 0 | 0 | 1 | 0 | 0 | 0 | -3 |

## 5. Computational Results

To evaluate the performance of the proposed method, we use twenty-nine test problems from four groups,

A) Nine Allocations Problems

B) Ten Fixed-Charge Problems of Haldi

C) Four Combinatorial Problems in Graph Theory

D) Six "IBM" Test Problems of Haldi,

which are explicitly reproduced in C.A. Trauth, JR. and R.E. Woolsey [5]. We shall evaluate our results based on the results published in the above reference, where the used codes and other features are detailed. Here we reproduce only some features which help us to evaluate the performance of the Search procedure.

In reference [5], the following five codes are tested: IPM 3 and LIP 1, which are based on the Gomory's fractional cut, and ILP 2-1, ILP 2-2 and IPSC, which are based on the Gomory's all-integer cut.

Tables 5, 6, 7 and 8 show the results published in the above reference and the results obtained by the Search procedure, for each group of problems tested, where iteration means a pivot iteration.

For the Search procedure, the random number series are arbitrarily chosen and $F = x_1$ in Step II is used for all the tested problems. We use two experimental codes for the Search procedure; the first one is written in BASIC and a personal computer NEC PC-8801 (P) is used and the reported CPU time for this code includes the display time. The second code is written in FORTRAN 77 and it is run on a large scale computer FACOM M382 (L). For this case, the reported CPU time includes compilation.

Considering differences among machines, a comparison on basis of calculation time lacks meaning. However the CPU time is reproduced to give an idea of the invested effort.

The Search procedure managed to solve all the above test problems, especially the 7-point combinatorial problem, which became infeasible.

The Search procedure shows computational stability for small size problems as seen in the test problems solved. However, in future, applications to different classes of larger integer models should be planned.

Table 5. Allocation Problems

| | Code | IPM 3 | LIP 1 | ILP 2-1 | ILP 2-2 | IPSC | Search Procedure | | |
|---|---|---|---|---|---|---|---|---|---|
| Problem | Size | Iter. | Iter. | Iter. | Iter. | Iter. | Iter. | Time | Machine |
| 1 | 1X10 | 14 | 19 | 54 | 51 | 46 | 5 | 71 | P |
| 2 | 1X10 | 31 | 55 | 163 | 77 | 64 | 32 | 380 | P |
| 3 | 1X10 | 30 | 41 | 168 | 59 | 71 | 22 | 197 | P |
| 4 | 1X10 | 18 | 19 | 192 | 48 | 62 | 12 | 118 | P |
| 5 | 1X10 | 11 | 12 | 139 | 32 | 50 | 3 | 52 | P |
| 6 | 1X10 | 18 | 40 | 157 | 54 | 81 | 25 | 256 | P |
| 7 | 1X10 | 61 | 81 | 504 | 119 | 131 | 39 | 404 | P |
| 8 | 1X10 | 21 | 51 | 307 | 57 | 102 | 29 | 280 | P |
| 9 | 1X10 | 12 | 12 | 201 | 34 | 44 | 4 | 120 | P |

Time in seconds

Size means $(m-n)Xn$

Table 6. Fixed-Charge Problems

| | Code | IPM 3 | LIP 1 | ILP 2-1 | ILP 2-2 | IPSC | Search Procedure | | |
|---|---|---|---|---|---|---|---|---|---|
| Problem | Size | Iter. | Iter. | Iter. | Iter. | Iter. | Iter. | Time | Machine |
| 1 | 4X5 | 54 | 24 | 135 | 36 | 32 | 22 | 171 | P |
| 2 | 4X5 | 81 | 15 | 94 | 47 | 45 | 19 | 120 | P |
| 3 | 4X5 | 37 | 26 | 154 | 104 | 56 | 22 | 132 | P |
| 4 | 4X5 | 91 | 18 | 93 | 18 | 22 | 19 | 120 | P |
| 5 | 5X5 | +7000 | 158 | +7000 | +7000 | 6104 | 124 | 800 | P |
| 6 | 5X5 | +7000 | 123 | +7000 | 311 | 3320 | 101 | 540 | P |
| 7 | 3X5 | +7000 | 159 | +7000 | +7000 | +7000 | 124 | 800 | P |
| 8 | 3X5 | +7000 | 126 | +7000 | 306 | +7000 | 101 | 540 | P |
| 9 | 6X6 | 118 | 42 | +7000 | 298 | 339 | 58 | 485 | P |
| 10 | 10X12 | 1369 | 102 | +7000 | +7000 | +7000 | 94 | 2170 | P |

Time in seconds

+7000 means that after 7000 iterations the problem is not solved.

Table 7. Combinatorial Problems

| | Code | IPM 3 | LIP 1 | ILP 2-1 | ILP 2-1 | IPSC | Search Procedure | | |
|---|---|---|---|---|---|---|---|---|---|
| Problem | Size | Iter. | Iter. | Iter. | Iter. | Iter. | Iter. | Time | Machine |
| 4-Point | 4X6 | 4 | 11 | 4 | 4 | 5 | 6 | 56 | P |
| 5-Point | 4X10 | 140 | 83 | 132 | 74 | 86 | 52 | 1080 | P |
| 6-Point | 65X15 | 42 | 27 | 114 | 29 | 70 | 51 | 2.5 | L |
| 7-Point | 140X21 | +7000 | +7000 | +7000 | +7000 | +7000 | 2830 | 24.0 | L |

Time in seconds

Table 8. "IBM" Test Problems

| | Code | IPM 3 | LIP 1 | ILP 2-1 | ILP 2-2 | IPSC | Search Procedure | | |
|---|---|---|---|---|---|---|---|---|---|
| Problem | Size | Iter. | Iter. | Iter. | Iter. | Iter. | Iter. | Time | Machine |
| 1 | 7X7 | 8 | 11 | 9 | 11 | 9 | 13 | 129 | P |
| 2 | 7X7 | 17 | 32 | 13 | 15 | 16 | 26 | 240 | P |
| 3 | 3X7 | 22 | 53 | 23 | 14 | 14 | 65 | 338 | P |
| 4 | 15X15 | 24 | 73 | 41 | 18 | 17 | 40 | 737 | P |
| 5 | 15X15 | 1144 | 351 | +7000 | 842 | 1020 | 224 | 2.7 | L |
| 9 | 35X15 | 6758 | 953 | +7000 | 1105 | 752 | 562 | 3.7 | L |

Time in seconds

## Acknowledgment

The authors are grateful to the referees for their valuable comments on this paper. The second author is partially supported by the National Council of Sciences and Technology of the Mexican Government (CONACYT) and by the Durango's Institute of Technology (Mexico) during his studies in Japan as a schoolarship student of the Japanese Ministry of Education.

## References

[1] Ben-Israel, A. and Charnes, A.: An Explicit Solution of a Special Class of Linear Programming Problems. *Operations Research*, Vol.16, No.6 (1968), 1166-1175.

[2] Charnes, A., Granot, F. and Phillips, F.: An Algorithm for Solving Inter-

val Linear Programming Problems. *Operations Research*, Vol.25, No.4 (1977), 688-695.

[3] Garfinkel, R. and Nemhauser, G.L.: A Survey of Integer Programming Emphasizing Computation and Relation among Models. *Mathematical Programming* (eds. T.C. Hu and S.M. Robinson). Academic Press, 1973, 77-155.

[4] Taha, H.A.: *Integer Programming, Theory, Applications and Computations*. Academic Press, 1975.

[5] Trauth, C.A. and Woolsey, R.E.: Integer Linear Programming: A Study in Computational Efficiency. *Management Sciences*, Vol.15 (1969), 481-493.

[6] Zionts, S.: Toward a Unifying Theory for Integer Linear Programming. *Operations Research*, Vol.17, (1969), 359-367.

Teruo SUNAGA: Department of Mechanical Engineering, Faculty of Engineering, Kyushu University, Hakozaki 6 - 10 - 1 Higashi-ku, Fukuoka, 812, Japan.