ACTA MATHEMATICA VIETNAMICA
Volume 22, Number 1, 1997, pp. 357–366

A NEW BOUNDING TECHNIQUE IN BRANCH-AND-BOUND ALGORITHMS FOR MIXED INTEGER PROGRAMMING

TRAN VU THIEU AND TRAN XUAN SINH

Dedicated to Hoang Tuy on the occasion of his seventieth birthday

Abstract. A branch-and-bound algorithm using a new bounding technique is presented for solving the mixed integer problem. The technique involves considering a piecewise linear and concave function of a parameter $\lambda \in \mathbb{R}^1$ whose maximum gives the largest lower bound. A parametric method is developed for finding the maximum of a such function.

1. Introduction

We consider the mixed integer programming problem, i.e. the problem of finding

$$(P) \qquad f^* = \min\{f(x,y) = c^T x + d^T y : x \in D, y \in E, (x,y) \in S, \\ x \text{ has all integer components}\},$$

where c, d are given vectors in $\mathbb{R}^p, \mathbb{R}^q$ respectively, x is a p-vector of integer restricted variables, y is a q-vector of continuous variables; D is a given polytope in $\mathbb{R}^p$, while E, S are given polyhedral convex sets in $\mathbb{R}^q, \mathbb{R}^n$ respectively ($n = p+q$). Denote by G the feasible region of (P). Without loss of generality we may assume that G is nonempty.

An important special case of (P) frequently encountered in practice is when $D = \{u \le x \le v\}$ with $u, v \in \mathbb{R}^p, E = \mathbb{R}^q_+$ and $S = \{(x,y) \in \mathbb{R}^n : Ax + By \le b\}$ with A, B, b being matrices of appropriate size.

Problem (P) has many applications in practice and has been studied in the literature for many years ago. The earliest methods for its solution

Received April 14, 1996
Key words. Mixed integer programming, parametric linear programming, piecewise linear and concave function, branch-and-bound algorithm, bounding technique.

were proposed by Gomory [3], Land and Doig [6], and Benders [1]. Subsequently, a number of other algorithms for solving the problem have been developed (for a review of them see e.g. [4]). Among them the algorithms based on the branch-and-bound technique seem to be more effective for the problem. We refer to [5] for a complete treatment of the branch-and-bound algorithms. Recently, Tien and Muu [8] proposed an algorithm of this kind for solving (P). However, its efficiency considerably depends upon the concrete rules for computing a lower bound for the objective value over a given set.

Exploiting the special structure of the problem, in the sequel we shall give an improved technique for computing a lower bound for the objective value of subproblems to be solved in the course of solving the problem by a branch-and-bound algorithm. The technique involves introducing a parameter $\lambda \in \mathbb{R}^1$ into the objective of the subproblems and we receive a piecewise linear, concave function of λ each value of which is a lower bound and whose maximum gives the largest lower bound. Some special values of such a function were chosen as a lower bound in [6] ($\lambda = 1$) and [8] ($\lambda = 0$). The breakpoints of this function can be found by parametric simplex method. However, as shown later, to compute the largest lower bound it is not necessary to generate all breakpoints in advance. Using parametric simplex method, it suffices to generate these breakpoints one by one, as needed in the course of computation.

After Introduction, we shall describe in Section 2 a new bounding technique. Then a parametric method for computing the largest lower bound will be developed in Section 3. Finally, a branch-and-bound algorithm using the new bounding technique will be presented in Section 4, along with some remarks for improving the algorithm.

2. A new bounding technique

Let $H \subset \mathbb{R}^p$ be a rectangle such that $D \subset H$. Given any rectangle $M \subset H$, let us consider the following problem, denoted by $P(M)$:

$$\min\{c^T x + d^T y : x \in D \cap M, y \in E, (x, y) \in S, \\ x \text{ has all integer components}\}.$$

Let $G(M)$ denote the feasible region and $f(M)$ denote the optimal function value of $P(M)$. Furthermore, for any fixed $\lambda \in \mathbb{R}^1$ consider a relaxation problem obtained from $P(M)$ by introducing auxiliary variables $t \in \mathbb{R}^p$

as follows:

$$\min\{\lambda c^T t + (1-\lambda)c^T x + d^T y : x \in M, t \in D \cap M, y \in E, (t,y) \in S,$$
$$x \text{ has all integer components}\}.$$

This problem is denoted by $Q(\lambda, M)$ and its optimal function value by $\mu(\lambda, M)$. As usual, for the convention we take $\mu(\lambda, M) = +\infty$ if the feasible set of $Q(\lambda, M)$ is empty.

Since the fact that (x, y) is feasible for $P(M)$ implies that $(x, t = x, y)$ is also feasible for $Q(\lambda, M)$ with the same objective function value, we have $\mu(\lambda, M) \le f(M)$ for all $\lambda \in \mathbb{R}^1$. By setting

$$\mu(M) \;=\; \max_{\lambda \in \mathbb{R}^1} \mu(\lambda, M),$$

we have $\mu(M) \le f(M)$, i.e. we can choose $\mu(M)$ as a lower bound for $f(M)$. Thus, if $D \subset \bigcup_{k \in I} M_k$, where I contains a finite number of subscripts, then $\min_{k \in I} \mu(M_k)$ is a lower bound for the optimal function value f^* of the original problem (P), i.e.

$$\min_{k \in I} \mu(M_k) \le f^*.$$

Remark 1. Problem $P(M)$ can also be relaxed as follows:

$$\min\{c^T x + d^T y : x \in M, t \in D \cap M, y \in E, (t,y) \in S,$$
$$c^T t = c^T x, x \text{ has all integer components}\}.$$

By the theory of Lagrangian duality, this problem is equivalent to

$$\min\{c^T x + d^T y + \lambda(c^T t - c^T x) : x \in M, t \in D \cap M, y \in E, (t,y) \in S,$$
$$\lambda \in \mathbb{R}^1,\ x \text{ has all integer components}\},$$

which is the same as $Q(\lambda, M)$.

It is easily seen that Problem $Q(\lambda, M)$ can be divided into two subproblems $Q_1(\lambda, M)$:

$$\mu_1(\lambda, M) \;=\; \min\{\lambda c^T t + d^T y : t \in D \cap M, y \in E, (t,y) \in S\},$$

and $Q_2(\lambda, M)$:

$$\mu_2(\lambda, M) \;=\; \min\{(1-\lambda)c^T x : x \in M \text{ and } x \text{ has all integer components}\}.$$

In addition, we get

$$\mu(\lambda, M) \;=\; \mu_1(\lambda, M) + \mu_2(\lambda, M). \tag{1}$$

Since M is a rectangle and D, E, S are polyhedra, $Q_1(\lambda, M)$ is a parametric linear program with the parameter λ in the objective function. So it can relatively easily be solved by the parametric objective simplex method (see e.g. [2]). By the theory of the parametric linear programming the optimal value $\mu_1(\lambda, M)$ is a piecewise linear and concave function on λ and there exists a finite number of breakpoints

$$-\infty < \lambda_{-s_1} < \lambda_{-s_1+1} < \cdots < \lambda_{-1} < \lambda_1 < \cdots < \lambda_{s_2-1} < \lambda_{s_2} < +\infty$$

in such a way that $\mu_1(\lambda, M)$ is linear for all $\lambda \in [\lambda_{k-1}, \lambda_k], -s_1 < k \le s_2$, or $\lambda \le \lambda_{-s_1}$ or $\lambda \ge \lambda_{s_2}$. (Here we number the breakpoints in order that breakpoints greater than 1 have positive subscripts and breakpoints smaller than 1 have negative subscripts).

Furthermore, for any given λ $Q_2(\lambda, M)$ is a problem of minimizing a linear function over a rectangle, so its optimal solution can easily be found in an explicit form. To do this let us denote $I^+ = \{i : c_i \ge 0\}$ and $I^- = \{i : c_i < 0\}$. Let $M = [u, v]$ with $u, v \in \mathbb{R}^p, u \le v$. For the purpose of the present paper we may assume that u and v have all integer components. Define

$$\begin{aligned} d_1 \;&=\; \sum_{i \in I^+} c_i u_i + \sum_{i \in I^-} c_i v_i \quad \text{and} \\ d_2 \;&=\; \sum_{i \in I^+} c_i v_i + \sum_{i \in I^-} c_i u_i. \end{aligned}$$

Evidently $d_1 \le d_2$. Denote by w^1 the vector having components u_i $(i \in I^+)$ and v_i $(i \in I^-)$, and by w^2 the vector with components v_i $(i \in I^+)$ and u_i $(i \in I^-)$. We have the following property.

Lemma 1. *Under the indicated notation:*

a) w^1 is an optimal solution of $Q_2(\lambda, M)$ for all $\lambda \le 1$.

b) w^2 is an optimal solution of $Q_2(\lambda, M)$ for all $\lambda \ge 1$.

c) $\mu_2(\lambda, M)$, the optimal function value of $Q_2(\lambda, M)$, is a piecewise linear and concave function with unique breakpoint $\lambda_0 = 1$, and has the form

$$\mu_2(\lambda, M) \;=\; \begin{cases} (1-\lambda) d_1 & \text{if } \lambda \le 1, \\ (1-\lambda) d_2 & \text{if } \lambda \ge 1. \end{cases} \tag{2}$$

The proof is immediate from the fact that M is a rectangle and $d_1 \leq d_2$.

3. Parametric method for computing $\mu(M)$

The above results show that $\mu(\lambda, M) = \mu_1(\lambda, M) + \mu_2(\lambda, M)$ is also a piecewise linear and concave function. The number of breakpoints of $\mu(\lambda, M)$ exceeds that of $\mu_1(\lambda, M)$ at most one (it is the case where $\lambda_k \neq 1$ for all $k = -s_1, \dots, s_2$).

However, to compute $\mu(M) = \max_{\lambda \in \mathbb{R}^1} \mu(\lambda, M)$ it is not necessary to generate all the breakpoints of $\mu(\lambda, M)$ in advance. The following lemma shows that it suffices to generate these breakpoints one by one as needed in the course of computation.

Lemma 2. *Assume that* $h(\lambda)$ *is a piecewise linear and concave function of a single variable* $\lambda \in \mathbb{R}^1$. *Let* λ_1, λ_2 *and* λ_3 *be successive breakpoints of* h, *i.e.* $h(\lambda)$ *is linear for all* $\lambda \in [\lambda_{k-1}, \lambda_k]$, $k = 2, 3$. *Denote* $h_i = h(\lambda_i), i = 1, 2, 3, h^* = \max_{\lambda \in \mathbb{R}^1} h(\lambda)$. *Then, there are only three possibilities:*

a) if $h_2 \geq \max(h_1, h_3)$ *then* $h^* = h_2$ *and* $\arg \max_{\lambda \in \mathbb{R}^1} h(\lambda) = \lambda_2$.

b) If $h_1 < h_2 < h_3$ *then* $h^* \geq h_3$ *and* $\arg \max_{\lambda \in \mathbb{R}^1} h(\lambda) \geq \lambda_3$.

c) If $h_1 > h_2 > h_3$ *then* $h^* \geq h_1$ *and* $\arg \max_{\lambda \in \mathbb{R}^1} h(\lambda) \leq \lambda_1$.

The proof of the Lemma 2 follows immediately from the concavity and the piecewise linearity of h.

By virtue of Lemma 1 and Lemma 2 we can now describe the algorithm for computing $\mu(M)$.

Algorithm 1 (for computing the lower bound $\mu(M)$ for $f(M)$).

Start. 1. Set $\lambda_0 = 1$, solve the linear program $Q_1(\lambda_0, M)$:

a) If $\mu_1(\lambda_0, M) = +\infty$, i.e. the feasible set of $Q_1(\lambda, M)$ is empty, then stop: $\mu(M) = +\infty$.

b) Otherwise, let (t^0, y^0) be an optimal solution of $Q_1(\lambda_0, M)$. Set $h_0 = \mu_1(\lambda_0, M)$ and go to step 2.

2. Starting with $\lambda_0 = 1$, (t^0, y^0) and using parametric simplex method for solving $Q_1(\lambda, M)$ find two near breakpoints λ_{-1} and λ_1 such that

$\lambda_{-1} < \lambda_0 = 1 < \lambda_1$ (possibly $\lambda_1 = +\infty$ and/or $\lambda_{-1} = -\infty$) and

$$\mu_1(\lambda, M) = \begin{cases} \gamma + \delta\lambda & \text{for } \lambda_{-1} \le \lambda \le \lambda_0, \\ \alpha + \beta\lambda & \text{for } \lambda_0 \le \lambda \le \lambda_1. \end{cases}$$

($\alpha = \gamma$ and $\beta = \delta$ if $\lambda_0 = 1$ is not a breakpoint of $\mu_1(\lambda, M)$).

From (1) and (2) we have

$$\mu(\lambda, M) = \begin{cases} \gamma + \delta\lambda + (1-\lambda)d_1 = \gamma + d_1 + (\delta - d_1)\lambda \text{ for } \lambda_{-1} \le \lambda \le \lambda_0, \\ \alpha + \beta\lambda + (1-\lambda)d_2 = \alpha + d_2 + (\beta - d_2)\lambda \text{ for } \lambda_0 \le \lambda \le \lambda_1. \end{cases}$$

We distinguish four cases:

a) If $\beta \le d_2$ and $\delta \ge d_1$ then stop: $\mu(M) = h_0$ and (x^0, t^0, y^0) is an optimal solution of $Q(\lambda_0, M)$ with $x^0 = w^1$ or w^2, as defined in Lemma 1.

b) If $\beta > d_2$ and $\lambda_1 = +\infty$ or if $\delta < d_1$ and $\lambda_{-1} = -\infty$ then stop: $\mu(M) = +\infty$.

c) If $\beta > d_2$ and $\lambda_1 < +\infty$ then set $\mu = \alpha + d_2 + (\beta - d_2)\lambda_1, k = 1$, $(t^1, y^1) = (t^0, y^0)$ and go to *Right* step.

d) If $\delta < d_1$ and $\lambda_{-1} > -\infty$ then set $\mu = \gamma + d_1 + (\delta - d_1)\lambda_{-1}, k = -1$, $(t^{-1}, y^{-1}) = (t^0, y^0)$ and go to *Left* step.

(μ denotes the best value so far obtained for $\mu(M)$).

Right. Starting with $\lambda = \lambda_k \ge 1$, (t^k, y^k) and using parametric simplex method for solving $Q_1(\lambda, M)$ find the nearest right breakpoint $\lambda_{k+1} > \lambda_k$ (maybe $\lambda_{k+1} = +\infty$). At the same time we get (t^{k+1}, y^{k+1}) which is an optimal solution of $Q_1(\lambda, M)$ for all $\lambda \in [\lambda_k, \lambda_{k+1}]$. From (1) and (2) we have

$$\mu(\lambda, M) = \alpha_k + \beta_k\lambda + (1-\lambda)d_2 = \alpha_k + d_2 + (\beta_k - d_2)\lambda, \ \lambda_k \le \lambda \le \lambda_{k+1}.$$

There exist three cases:

a) If $\beta_k \le d_2$ then stop: $\mu(M) = \mu$ and (w^2, t^k, y^k) is an optimal solution of $Q(\lambda_k, M)$.

b) If $\beta_k > d_2$ and $\lambda_{k+1} = +\infty$ then stop: $\mu(M) = +\infty$.

c) If $\beta_k > d_2$ and $\lambda_{k+1} < +\infty$ then set $\mu = \alpha_k + d_2 + (\beta_k - d_2)\lambda_{k+1}, k+1 \leftarrow k$ and go to *'Right'* again.

Left. Starting with $\lambda = \lambda_k \le 1$, (t^k, y^k) and using parametric simplex method for solving $Q_1(\lambda, M)$ find the nearest left breakpoint $\lambda_{k-1} < \lambda_k$

(maybe $\lambda_{k-1} = -\infty$). At the same time we get (t^{k-1}, y^{k-1}) which is an optimal solution of $Q_1(\lambda, M)$ for all $\lambda \in [\lambda_{k-1}, \lambda_k]$. From (1) and (2) we have

$$\begin{aligned}\mu(\lambda, M) &= \alpha_k + \beta_k\lambda + (1-\lambda)d_1 \\ &= \alpha_k + d_1 + (\beta_k - d_1)\lambda,\ \lambda_{k-1} \le \lambda \le \lambda_k.\end{aligned}$$

There are three cases:

a) If $\beta_k \ge d_1$ then stop: $\mu(M) = \mu$ and (w^1, t^k, y^k) is an optimal solution of $Q(\lambda_k, M)$.

b) If $\beta_k < d_1$ and $\lambda_{k-1} = -\infty$ then stop: $\mu(M) = +\infty$.

c) If $\beta_k < d_1$ and $\lambda_{k-1} > -\infty$ then set $\mu = \alpha_k + d_1 + (\beta_k - d_1)\lambda_{k-1}, k-1 \leftarrow k$ and go to *'Left'* again.

Since once *'Right'* or *'Left'* is applied, a new breakpoint of $\mu(\lambda, M)$ is obtained and since $\mu(\lambda, M)$ has a finite number of breakpoints, it can easily be shown that Algorithm 1 terminates after a finite number of *'Right'* or *'Left'* applications, giving the optimal value $\mu(M) = \mu(\lambda_k, M) = \max_{\lambda \in \mathbb{R}^1} \mu(\lambda, M)$ and (x^k, t^k, y^k) is an optimal solution of $Q(\lambda_k, M)$, where $x^k = w^1$ (if $\lambda_k \le 1$) or $x^k = w^2$ (if $\lambda_k \ge 1$), as defined in Lemma 1.

4. A BRANCH-AND-BOUND ALGORITHM FOR MIXED INTEGER PROGRAMMING

Basing on the above developed bounding technique, we are now in a position to propose a branch-and-bound algorithm for solving the mixed integer programming problem (P).

Algorithm 2 (for solving (P)).

Initial step. First choose an initial rectangle $H \subset \mathbb{R}^p$ such that the polytope D is entirely contained in H. Let $H = [u^0, v^0]$ with $u^0, v^0 \in \mathbb{R}^p, u^0 \le v^0$. Without loss of generality we may assume that u^0 and v^0 have all integer components. Then, let $M_0 = H, \mathcal{M}_0 = \{M_0\}, \mathcal{R} = \{M_0\}$. For each rectangle $M \in \mathcal{M}_0$ compute (by the parametric method presented in the previous section) a lower bound $\mu(M)$ for the optimal value $f(M)$ of problem $P(M)$. Set $\gamma_0 = +\infty, G_0 = \emptyset$ or $\gamma_0 = f(x^0, y^0), G_0 = \{(x^0, y^0) \in G\}$ if we already have a feasible solution (x^0, y^0) for (P). Set $k = 0$.

Step $k = 0, 1, 2, \ldots$. At this step we already have γ_k (the best value so far obtained for f^*) and, if $\gamma_k < +\infty$, (x^k, y^k) with $\gamma_k = f(x^k, y^k) =$

$c^T x^k + d^T y^k$ (the best feasible solution of (P) so far obtained), $\mathcal{M}_k$ (the collection of rectangles that remain to be examined) and for each rectangle $M \in \mathcal{M}_k$ a number $\mu(M) \leq \min\{f(x,y) : (x,y) \in G(M)\}$ (the estimated lower bound for $f(x,y)$ over $G(M)$).

a) Delete all rectangles $M \in \mathcal{M}_k$ with $\mu(M) \geq \gamma_k$. Let $\mathcal{R}_k$ be the collection of all remaining rectangles.

b) If $\mathcal{R}_k = \emptyset$, stop: (x^k, y^k) is an optimal solution of (P). (If there is no incumbent, the conclusion is that the problem (P) has no feasible solution).

In the opposite case choose a rectangle $M_k \in \mathcal{R}_k$ such that

$$\mu(M_k) \;=\; \min\{\mu(M) : M \in \mathcal{R}_k\}$$

(M_k with the smallest lower bound $\mu(M_k)$ among M in $\mathcal{R}_k$).

c) Subdivide M_k by the longest edge into two subrectangles as follows.

Let $M_k = [u^k, v^k]$ with $u^k, v^k \in \mathbb{R}^p, u^k \leq v^k$ and u^k_j, v^k_j are integer for all $j = 1, 2, \ldots, p$. Select $j_k \in \{1, 2, \ldots, p\}$ such that

$$\ell_k \;=\; v^k_{j_k} - u^k_{j_k} \;=\; \max_{1 \leq j \leq p} (v^k_j - u^k_j). \tag{3}$$

Let $\delta_k = u^k_{j_k} + [\ell_k/2]$ ($[x]$ denotes the integer part of x) and define

$$M_k^- \;=\; \{x \in M_k : x_{j_k} \leq \delta_k\}, \tag{4}$$

$$M_k^+ \;=\; \{x \in M_k : x_{j_k} \geq \delta_k + 1\}. \tag{5}$$

d) Compute $\mu(M_k^-)$ and $\mu(M_k^+)$ (by the parametric method presented in Section 3). For each $M \in \{M_k^-, M_k^+\}$ let us denote (t^M, y^M) the solution (if there exists) of $Q_1(\lambda, M)$.

e) Set

$$\begin{aligned} G_{k+1} \;&=\; \{(t^M, y^M) : t^M \text{ has all integer components, } \; M \in \{M_k^-, M_k^+\}\}, \\ \gamma_{k+1} \;&=\; \min(\gamma_k, \min\{f(x,y) : (x,y) \in G_{k+1}\}), \end{aligned}$$

and if $\gamma_{k+1} < +\infty$ let (x^{k+1}, y^{k+1}) belong to G in such a way that $\gamma_{k+1} = f(x^{k+1}, y^{k+1})$.

f) Set $\mathcal{M}_{k+1} \;=\; (\mathcal{R}_k - M_k) \bigcup \{M_k^-, M_k^+\}$ and go to step $k+1$.

The finiteness of Algorithm 2 follows immediately from the fact that the rectangle $H \supset D$ has a finite number of integer points.

Remark 2. In the course of computing $\mu(M)$ by Algorithm 1 we can stop when we get $\mu \geq \gamma_k$. (Recall that μ is the best value so far obtained for $\mu(M) = \max_{\lambda \in \mathbb{R}^1} \mu(\lambda, M)$ and γ_k is an upper bound so far obtained for f^*). Furthermore, as indicated in Section 2, $\mu(\lambda, M) \leq f(M)$ for all $\lambda \in \mathbb{R}^1$. So in order to reduce computation efforts we can stop computing $\mu(M)$ by Algorithm 1 when we have gotten λ_k with $\mu(\lambda_k, M) \geq \max(\mu(0, M), \mu(1, M))$. For so doing, it suffices to stop at *Right* step k with $\lambda_k \geq 1$ or at *Left* step k with $\lambda_k \leq 0$. If, for example, at *Start* step the situation c) occurs then we can take $\mu(\lambda_1, M)$ as a lower bound for $f(M)$. In this situation we guarantee $\mu(\lambda_1, M) \geq \mu(\lambda, M)$ for all $\lambda \leq \lambda_1$.

Remark 3. To subdivide M_k chosen at step *k-b*) we can apply the adaptive subdivision presented in [7]. Namely, instead of selecting the index j_k, the longest edge of M_k, by (3), we choose the longest edge of $\hat{M}_k \subset M_k$, where $\hat{M}_k$ is a smallest 'integer' rectangle which contains x^k and t^k. (Recall that the triplet (x^k, t^k, y^k) is found by Algorithm 1 when computing $\mu(M)$). $\hat{M}_k$ is constructed as follows. $\hat{M}_k = [x^k, \hat{t}^k]$ with

$$\hat{t}^k_j = \begin{cases} t^k_j & \text{if } t^k_j \text{ is integer,} \\ [t^k_j] & \text{if } t^k_j \text{ is not integer and } t^k_j < x^k_j, \\ [t^k_j] + 1 & \text{if } t^k_j \text{ is not integer and } t^k_j > x^k_j. \end{cases}$$

The index $j_k \in \{1, 2, \ldots, p\}$ is now selected by

$$|x^k_{j_k} - \hat{t}^k_{j_k}| = \max_{1 \leq j \leq p} |x^k_j - \hat{t}^k_j|. \tag{3'}$$

Let $\delta_k = [(x^k_{j_k} + \hat{t}^k_{j_k})/2]$ and define M^-_k, M^+_k by (4), (5) respectively.

Remark 4. In the course of computing $\mu(M)$ by Algorithm 1, we always have (t^k, y^k) as a feasible solution for $Q_1(\lambda, M)$. If t^k has all integer components, i.e. (t^k, y^k) is also feasible for $P(M)$ then we can use $c^T t^k + d^T y^k$ to improve γ_k, the best value so far obtained for f^* :

$$\gamma_k \longleftarrow \min\{\gamma_k, c^T t^k + d^T y^k\}$$

and stop computing $\mu(M)$ when $\mu \geq \gamma_k$ (μ is the best value so far obtained for $\mu(M)$).

References

1. J. F. Benders, *Partitioning procedures for solving mixed variables programming problems*, Numerische Mathematik **4** (1962), 238-252.

2. S. I. Gass, *Linear Programming (Methods and Applications)*, Fifth Edition, McGraw-Hill, New York - Toronto - London 1985.

3. R. E. Gomory, *An algorithm for the mixed integer problem*, Rand. Corp., P-1885, Santa Monica, California. Feb. 22 (1960).

4. F. S. Hillier and G. J. Lieberman, *Introduction to Mathematical Programming*, Second Edition. McGraw-Hill, New York - London - Singapore 1995.

5. R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer–Verlag, Berlin–Heidelberg–New York, 1990.

6. A. H. Land and A. G. Doig, *An automatic method of solving discrete programming problems*, Econometrica **28** (1960), 497-520.

7. L. D. Muu and W. Oettli, *A method for minimizing a convex-concave function over a compact set*, Journal of Optimization and Applications **70** (1990), 377-384.

8. N. V. Tien and L. D. Muu, *A relaxation algorithm for solving mixed integer programming problems*, Acta Math. Vietnam. **22** (1997), 361-372.

Institute of Mathematics,
P.O. BOX 631 Bo Ho, Hanoi, Vietnam

Department of Mathematics, Vinh University,
Vinh city, Vietnam