ACTA MATHEMATICA VIETNAMICA
Volume 22, Number 1, 1997, pp. 367–378

A RELAXATION ALGORITHM FOR SOLVING MIXED INTEGER PROGRAMMING PROBLEMS

NGUYEN VU TIEN AND LE DUNG MUU

Dedicated to Hoang Tuy on the occasion of his seventieth birthday

Abstract. We propose an algorithm for solving mixed integer linear programming problem , which is a combination of branch-and-bound and decomposition procedure. For branching and bounding we divide a rectangular domain into smaller and smaller subrectangles, and to each generated subrectangle R we associate a lower bound for the objective function with respect to $y \in R$ by using a suitable relaxation involving the continuous linear constraint but not the integer ones. This relaxation thus allows us to decompose the problem into subprograms each of them consists of a linear program and one-dimensional integer linear problems.

1. Introduction

Consider the following mixed integer linear programming problem, denoted by (P):

$$\text{(P)} \qquad \min\{f(x,y) := c^T x + d^T y : x \in X, y \in Y, (x,y) \in S,\ y \text{ integer}\},$$

where X and S are polyhedral convex sets in R^n and $R^n \times R^m$ given by a system of linear inequalities and/or equalities; Y is a rectangle in R^m. By F we shall denote the feasible region of (P).

Mixed integer linear programming of the form (P) is an important topics of mathematical programming, since it has many applications in different fields. It is recognized that Problem (P) is difficult to solve. The existing methods are efficient only with a moderate size of the integer variables. Some efforts have been made to develop solution-methods for Problem (P). The first decomposition method for solving (P) is due to Bender [1]. Another important approach to Problem (P) is by branch-

Received April 4, 1996
Key words. Optimization, relaxation, mixed linear integer problem, decomposition, branch-and-bound.
This work is partially supported by the National Research Program in Natural Sciences

and-bound method combined with other techniques such as cutting plane and decomposition. Commonly used technique for obtaining lower bound in these methods is relaxation by linear programming (see e.g. [1], [2], [5], [6]). Recently methods using "strong" valid inequalities and reformulation have been intensively study to solve specific models of mixed integer linear programming problems of the form (P) (see [6] and references therein).

The purpose of this paper is to propose an algorithm for the solution of Problem (P), which is a combination of branch-and-bound method and decomposition procedure. Similar schemes have been described in our earlier papers [3], [4] for solving an indefinite quadratic function and a convex-concave function over a convex set. Branching is performed by dividing the rectangular domain Y into smaller and smaller subrectangles, and to each generated subrectangle R we associate a real number which serves as a lower bound for the objective function with respect to $y \in R$ by using a suitable relaxation involving the constraint $(x, y) \in S$ but not the integer constraints. This relaxation allows us to decompose Problem (P) into subprograms consisting of a linear program in the space $R^n \times R^m$ together with the problem of minimizing the linear function $d^T y$ over the integer points of a rectangular domain of R^m. The latter can in turn be converted into m one-dimensional linear integer problems, and therefore it can be solved very easily. The method can be extended to the case when X, Y, S are convex compact sets, and

$$f(x, y) := h(x) + \sum g_j(y_j),$$

where h is a convex continuous function on X and each g_j is a function of one variable with $g := \sum g_j$ continuous on Y.

2. Description of the algorithm

For each subrectangle R we denote by $P(R)$ the problem

$$(P(R)) \qquad \min\{c^T x + d^T y : x \in X, y \in R, (x, y) \in S,\ y \text{ integer}\},$$

and by $B(R)$ the relaxed problem

$$(B(R)) \qquad \min\{c^T x + d^T y : x \in X, y \in R, u \in R, (x, u) \in S,\ y \text{ integer}\}.$$

By $f(R)$ and $b(R)$ we denote the optimal values of $P(R)$ and $B(R)$, respectively. We agree that an optimal value equals $+\infty$ if no feasible point exists. It is clear that the feasible region of $P(R)$ is contained in that of $B(R)$. Hence $b(R) \leq f(R)$, and therefore if $Y = \bigcup_j R_j$, then $\min_j b(R_j)$ is a lower bound for the optimal value f_* of Problem (P), i.e., $\min_j b(R_j) \leq f_*$.

Note that Problem $B(R)$ can be solved by solving separately the standard linear program

$$(B_1(R)) \qquad \min\{c^T x : x \in X, u \in R, (x,u) \in S\},$$

and the linear integer problem

$$(B_2(R)) \qquad \min\{d^T y : y \in R,\ y \ \text{ integer}\},$$

so that $b(R) := b_1(R) + b_2(R)$, where $b_i(R)$ denotes the optimal value of $B_i(R)$, $i = 1,2$. It is easy to see that if $R := \{y \in R^m : \beta_j \leq y_j \leq B_j,\ j = 1,\dots,m\}$, then solving Problem $B_2(R)$ amounts to solving one-dimensional integer problems of the form

$$\min\{d_j y_j : \beta_j \leq y_j \leq B_j,\ y_j \ \text{ integer}\}, (j = 1,\dots,m).$$

We are now in a position to describe the algorithm.

Algorithm.

Step 0.

Set $R_0 = Y$ and solve Problem $B_1(R_0)$ and $B_2(R_0)$.

a) If $b(R_0) = \infty$, then $f_* = \infty$; stop: Problem (P) has no feasible point.

b) If $b(R_0) < \infty$, then take a solution (x^0, u^0) of the linear problem $B_1(R_0)$ and a solution y^0 of the problem $B_2(R_0)$.

Let

$$F_0 := \{(x^0, v) : v \in \{u^0, y^0\}, (x^0, v) \in F\},$$
$$f_0 := \min\{c^T x^0 + d^T v : (x^0, v) \in F_0\}$$

and, if $F_0 \neq \emptyset$, let $(x^0, v^0) \in F_0$ such that $f_0 = f(x^0, v^0)$.

Set
$$(s^0, v^0) = (x^0, v^0), \quad \Delta_0 = \{R_0\}, k = 0.$$

Step 1.

a) If $f_k \le b(R_k)$, then terminate; f_k is the optimal value and (s^k, v^k) (if any) is an optimal solution of Problem (P). (If no (s^k, v^k) exists, then $f_* = \infty$, i.e., (P) has no feasible point.)

b) If $f_k > b(R_k)$, then bisect R_k into the two rectangles R_k^- and R_k^+ as follows:

Let $j_k \in \{1. \dots, m\}$ such that

$$|(u^k - y^k)_{j_k}| = \max_j |(u^k - y^k)_j|,$$

and set

$$\begin{aligned} R_k^- &:= \{y \in R_k,\ y_{j_k} \le r_k\}, \\ R_k^+ &:= \{y \in R_k,\ y_{j_k} \ge r_k\}, \end{aligned}$$

where $r_k := \dfrac{1}{2}(u^k + y^k)_{j_k}$.

Solve $B(R_k^-)$ and $B(R_k^+)$. For each $R \in \{R_k^-, R_k^+\}$ denote by (x^R, u^R) and y^R the obtained solutions (if any) of $B_1(R)$ and $B_2(R)$ respectively.

Let

$$\begin{aligned} F_{k+1} &:= \{(x^R, v) \in F : v \in \{u^R, y^R\}, R \in \{R_k^-, R_k^+\}\}, \\ f_{k+1} &:= \min\{f_k, \min\{f(x, v) : (x, v) \in F_{k+1}\}\}, \end{aligned}$$

and, if $f_{k+1} < \infty$, let $(s^{k+1}, v^{k+1}) \in F$ such that $f_{k+1} = f(s^{k+1}, v^{k+1})$.

Let

$$\begin{aligned} \Delta_k' &:= (\Delta_k \setminus \{R_k\}) \cup \{R_k^-, R_k^+\}, \\ \Delta_{k+1} &:= \{R \in \Delta_k' : b(R) \le f_{k+1}\}. \end{aligned}$$

Set $R_{k+1} \in \Delta_{k+1}$ such that $b(R_{k+1}) = \min\{b(R) : R \in \Delta_{k+1}\}$.

Increase k by 1 and go to Step 1.

Remarks.

1. The main operations in the just described algorithm are the solutions of Problem $B_1(R)$ and $B_2(R)$. Problem $B_1(R)$ is a standard linear program. Problem $B_2(R)$ can be solved by solving m one-dimensional linear integer problems.

2. The lower bound computed via Problems $B_1(R)$ and $B_2(R)$ generally is not inferior to that obtained by usual the linear programming relaxation. To see this let us consider the following simple example:

$$\min\{-x + 100y :\ 1.5 \le x \le 100,\ 1.5 \le y \le 100,\ x + y \le 100,\ y \text{ integer}\}.$$

Let $R = [\ 1.5, 100\ ]$ then

$$b_1(R) = \min\{-x : 1.5 \le x, u \le 100,\ x + u \le 100\} = -98.5,$$

$$b_2(R) = \min\{100y : 1.5 \le y \le 100,\ y \ \text{ integer}\} = 200.$$

Thus $b(R) = b_1(R) + b_2(R) = 101.5$, while the lower bound by usual linear programming relaxation is

$$\begin{aligned}\min\{-x + 100y : 1.5 \le x \le 100,\ 1.5 \le y \le 100,\ x + y \le 100\}\\ = -98.5 + 150 = 51.5.\end{aligned}$$

3. Convergence

In the sequel, for simplicity of notation, we shall write x^k, u^k, y^k, b_k for x^{R_k}, u^{R_k}, y^{R_k}, $b(R_k)$ respectively. From the definitions of $P(R)$ and $B(R)$, it follows that the optimal value $b(R)$ of $B(R)$ cannot exceed the optimal value $f(R)$ of $P(R)$. Hence, for every k,

$$f_* \ge b_k := \min\{b(R); R \in \Delta_k\}.$$

This and $f_k \ge f_*$ ensure that when the stopping criterion is satisfied for some k, i.e. $f_k \le b_k$, then f_k is the optimal value of Problem (P). So if $f_k = \infty$ we deduce that (P) has no feasible point. Otherwise, the point (s^k, v^k) is an optimal solution of (P). If the algorithm is infinite, it will converge in the following sense:

Theorem. *The sequences $\{(x^k, u^k)\}$ and $\{(x^k, y^k)\}$ have a common limit point which solves Problem* (P). *Furthermore $b_k \nearrow f_*$.*

Proof. If the algorithm is infinite, there exists a decreasing sequence of rectangles. For simplicity of notation we also denote this sequence by $\{R_k\}$. Thus $R_{k+1} \subset R_k$ for all k. Let (x^*, u^*, y^*) be any limit point of the sequence $\{(x^k, u^k, y^k)\}$. By taking subsequences, if necessary, we may assume that $j_k = j_*$ for some $1 \le j_* \le m$ and that $(x^k, u^k, y^k) \to (x^*, u^*, y^*)$ as $k \to \infty$. Furthermore, since the number of integer points of

Y is finite we may assume that $y^k = y^*$ for every k. Since the set R_k is bisected into R_k^- and R_k^+, we may assume, again extracting a subsequence if necessary, that $R_{k+1} \subset R_k^-$ for all k or $R_{k+1} \subset R_k^+$ for all k.

If $R_{k+1} \subset R_k^-$ then u^{k+1} and $y^* \in R_k^-$. Hence

$$u_{j_*}^{k+1} \leq \frac{(u^k + y^*)_{j_*}}{2} \tag{1}$$

and

$$y_{j_*}^* \leq \frac{(u^k + y^*)_{j_*}}{2} \ . \tag{2}$$

If $(u^k - y^*)_{j_*} > 0$, then we use (1) to obtain

$$0 < u_{j_*}^k - y_{j_*}^* = 2\Big(u_{j_*}^k - \frac{(u_{j_*}^k + y_{j_*}^*)}{2}\Big) \leq 2(u_{j_*}^k - u_{j_*}^{k+1}) \to 0. \tag{3}$$

If $(u^k - y^*)_{j_*} \leq 0$, then we use (2) to obtain

$$0 \leq y_{j_*}^* - u_{j_*}^k = 2\Big(y_{j_*}^* - \frac{(u_{j_*}^k + y_{j_*}^*)}{2}\Big) \leq 2(y_{j_*}^* - y_{j_*}^*) = 0. \tag{4}$$

For the case $R_{k+1} \subset R_k^+$ for all k, by the same argument we also obtain (3) and (4). Thus in the both cases we have $|u_{j_*}^k - y_{j_*}^*| \to 0$. Hence $|u_{j_*}^* - y_{j_*}^*| = 0$. This and the definition of j_* imply that $u^* = y^*$. Noting that $(x^*, u^*) \in S \cap (X \times Y)$ and y^* is an integer we see that (x^*, u^*) is feasible for (P). In view of the definition of b_k this implies that

$$f_* \geq b_k := c^T x^k + d^T y^k \to c^T x^* + d^T y^* = c^T x^* + d^T u^* \geq f_*.$$

Hence $c^T x^* + d^T u^* = f_*$, and therefore $(x^*, u^*) = (x^*, y^*)$ solves (P).

Since the sequence $\{b_k\}$ is nondecreasing and, by the first part, it has a subsequence converging to f_*, it follows that $b_k \nearrow f_*$. The theorem thus is proved.

Remark. At each iteration k the points (x^k, u^k) and (x^k, y^k) are candidates for the solution of Problem (P). Each point (x^k, u^k) satisfies the constraint $(x, y) \in S$, but may satisfy the integer constraints only in the limit. On the other hand, the point (x^k, y^k) always satisfies the integer constraints but may satisfy the constraint $(x, y) \in S$ only in the limit.

4. Finite convergence and extension

By using a suitable integral bisection the above algorithm can be made finite. In fact, let R_k be the rectangle to be bisected at iteration k, and let u^k, y^k be the bisection points obtained through the bounding operation over R_k. That is.

$$y^k \in arg\min\{d^T y : y \in R^k, y \ \text{ integer }\},$$

$$(x^k, u^k) \in arg\min\{c^T x : x \in X, u \in R_k, (x, u) \in S\}.$$

Note that $u^k \neq y^k$, since otherwise the algorithm would terminate at iteration k. Then we bisect R_k into two rectangles

$$R_k^- := \{y \in R_k : y_{j_k} \leq [r_{j_k}]\}, \tag{5}$$

$$R_k^+ := \{y \in R_k : y_{j_k} \geq [r_{j_k}] + 1\}, \tag{6}$$

where r_{j_k} is the j_kth coordinate of the vector $(u^k+y^k)/2$, and $[r_{j_k}]$ denotes the largest integer which is not greater than r_{j_k}, and j_k, as before, is the index corresponding to the maximal coordinatate of the vector $u^k - y^k$, i.e.,

$$|(u^k - y^k)_{j_k}| = \max_j |(u^k - y^k)_j|.$$

It is easy to verify that with this integer bisection, every integral vector of Y is fathomed through the bounding operation. Furthermore by a similar argument to the proof of convergence, one can show that the algorithm with the integer bisection (5), (6) must terminate after finitely many iterations yielding an optimal solution.

The above algorithm can be extended to the following nonlinear mixed integer problem:

$$\min\left\{h(x) + \sum_{j=1}^m g_j(y_j) : x \in X, y \in Y, (x, y) \in S, \ y \text{ integer}\right\},$$

where X, Y are compact convex sets of R^n, R^m respectively, S is a closed convex set of $R^n \times R^m$, h is a continuous convex function on X, and each g_j is a function of one variable, and $g := \sum_{j=1}^m g_j$ is continuous on Y.

In this case we initialize the algorithm from a rectangle R_0 containing the convex compact set Y. Subsequently, for each subrectangle

$$R := \{y \in R^m : \beta_j \leq y_j \leq B_j, j = 1, \ldots, m\} \subseteq R_0,$$

the relaxed problem $B(R)$ reads

$$\min\{h(x)+\sum g_j(y_j) : x \in X, y \in R, u \in R\cap Y, (x,u) \in S,\ y_j \text{ integer } \forall j\},$$

and therefore Problem $B_1(R)$ becomes the convex program

$$\min\{h(x) : x \in X, u \in R \cap Y, (x,u) \in S\},$$

while $B_2(R)$ can be converted into m one-dimensional integer programs

$$\min\{g_j(y_j) : \beta_j \le y_j \le B_j, y_j \text{ integer}\}, (j = 1, \dots, m).$$

5. Illustrative example and preliminary computational results

We illustrate the steps of the proposed algorithm on the following simple example

$$\min f(x,y) := 2x_1 - x_2 - y_1 + 2y_2,$$

subject to

$$x \in X := \{x = (x_1, x_2) :\ x_1 + x_2 \le 4,\ x_1, x_2 \ge 0\},$$

$$y \in Y := \{y = (y_1, y_2) :\ 0 \le y_1 \le 8\ \ , 0 \le y_2 \le 6\},$$

$$(x,y) \in S := \{(x_1, x_2, y_1, y_2) : x_1 - 2x_2 + 2y_1 - y_2 \le 6\}.$$

We will use the integer bisection defined by (5) and (6).

At Step 0 we take $R_0 = Y = \{y = (y_1, y_2) : 0 \le y_1 \le 8,\ 0 \le y_2 \le 6\}$. To compute lower bounds we solve the linear program

$$\min\{2x_1 - x_2 :\ x \in X, u \in R_0, (x,u) \in S\},$$

obtaining an optimal solution $(x,u) = (0,4,0,0)$, which is also a feasible point, and the optimal value $b_1(R^0) = -4$. Then we solve the linear integer program

$$\min\{-y_1 + 2y_2 : y = (y_1, y_2) \in R_0,\ y \text{ integer}\},$$

obtaining an optimal solution $(y_1, y_2) = (8,0)$ and the optimal solution $b_2(R_0) := -8$. Thus the lower bound for R_0 is $\beta(R^0) = -12$. The best known feasible point at this step is $(x^0, u^0) = (0, 4, 0, 0)$. Thus the best known upper is $f(x^0, u^0) = -4$.

In Step 1 of iteration $k = 0$ the rectangle R_0 is divided into two sets

$$R_0^- = \{y = (y_1, y_2) : 0 \le y_1 \le 4,\ 0 \le y_2 \le 6\},$$

$$R_0^+ = \{y = (y_1, y_2) : 4 \le y_1 \le 8,\ 0 \le y_2 \le 6\}.$$

To obtain a lower bound for R_0^-, as in Step 0, we solve the problems

$$\min\{2x_1 - x_2 :\ x \in X, u \in R_0^-, (x, u) \in S\},$$

$$\min\{-y_1 + 2y_2 : y = (y_1, y_2) \in R_0^-,\ y \text{ integer}\}.$$

For the former program we obtain an optimal solution $(x, u) = (0, 4, 0, 0)$ with optimal value is $b_1(R_0^-) = -4$. For the latter, an optimal solution and value are $y = (4, 0)$ and $b_2(R_0^-) = -4$. Thus a lower bound for R_0^- is $\beta(R_0^-) = -8$. A new feasible point is $(x, y) = (0, 4, 4, 0)$ with $f(x, y) = -8$. Thus the current best upper bound is -8. Hence R_0^- is eliminated from further consideration.

Likewise, to obtain a lower bound for R_0^+ we solve the programs

$$\min\{2x_1 - x_2 :\ x \in X, u \in R_0^+, (x, u) \in S\},$$

$$\min\{-y_1 + 2y_2 : y = (y_1, y_2) \in R_0^+,\ y \text{ integer}\}.$$

For the former program an optimal solution and the optimal value are $(x, u) = (0, 4, 7, 0)$ and $b_1(R_0^+) = -4$. For the latter we have $y = (8, 0)$ and $b_2(R_0^+) = -8$. The lower bound for this rectangle is then $\beta(R_0^+) = -12$. New feasible point and upper bound are $(x, u) = (0, 4, 7, 0)$ and -11.

In Step 1 of iteration $k = 1$ we have $\Delta_1 = \{R_0^+\}$, $f_1 = -11$, $\beta_1 = -12$ and the current best feasible point is $(x^1, v^1) = (0, 4, 7, 0)$. Thus $R_1 = R_0^+$ is divided into two rectangles

$$R_1^- = \{y = (y_1, y_2) : 4 \le y_1 \le 7,\ 0 \le y_2 \le 6\},$$

$$R_0^+ = \{y = (y_1, y_2) : 8 \le y_1 \le 8,\ 0 \le y_2 \le 6\}.$$

A lower bound for R_1^- is $\beta(R_1^-) = b_1(R_1^-) + b_2(R_1^-) = -4 - 7 = -11$ with $(x, u) = (0, 4, 7, 0)$, $y = (7, 0)$. This rectangle is eliminated.

A lower bound for R_1^+ is $\beta(R_1^+) = b_1(R_1^+) + b_2(R_1^+) = -4 - 8 = -12$ with $(x, u) = (8, 4, 8, 2)$, $y = (8, 0)$. No new feasible point is found, the upper bound thus is unchanged.

In Step 1 of iteration $k = 2$ we have $\Delta_2 = \{R_1^+\}$, $f_2 = -11$, $\beta_2 = -12$ and the current best feasible point is $(x^2, v^2) = (0, 4, 7, 0)$. Set $R_2 = R_1^+$ and divide it into

$$R_2^- = \{y = (y_1, y_2) : 8 \leq y_1 \leq 8,\ 0 \leq y_2 \leq 1\},$$

$$R_2^+ = \{y = (y_1, y_2) : 8 \leq y_1 \leq 8,\ 1 \leq y_2 \leq 6\}.$$

A lower bound for R_2^- is $\beta(R_2^-) = b_1(R_2^-) + b_2(R_2^-) = 1 - 8 = -7$ with $(x, u) = (0, 4, 7.5, 1)$, $y = (8, 0)$.

A lower bound for R_2^+ is $\beta(R_2^+) = b_1(R_2^+) + b_2(R_2^+) = -4 - 6 = -10$ with $(x, u) = (0, 4, 8, 2)$, $y = (8, 1)$.

Since $\beta(R_2^-) = -7 > f_2 = -11$ and $\beta(R_2^+) = -10 > f_2 = -11$, both these sets are deleted. The procedure terminates yielding an optimal solution $(x^*, y^*) = (x^2, v^2) = (0, 4, 4, 7)$ with the optimal value $f(x^*, y^*) = -11$.

To obtain a preliminary evaluation of the performance of the proposed algorithm, the algorithm was coded in PASCAL7 and run on a MATH 486 personal computer. The computed code used the ordinary simplex method for solving the linear programs called for by the algorithm. The computational results on fourteen randomly generated problems are reported in Table 1. In this table we use the following notations:

- n, p : numbers of continuous and integer variables respectively,
- m: number of constraints defined the polytope X (without $x \geq 0$),
- ite: number of iterations,
- s: number of the rectangles stored in the memory,
- $time$: CPU time (in second).

The results show that the algorithm could be used for solving Problem (P) with a number of integer variables up to $p = 6$ on a personal computer 486. The number of continuous variables x may be much larger. It appears that the running time is much more sensitive to the growth of the number of integer variables y than to the growth of the number of constraints or continuous variables x. The required memory however increases slowly as the program runs, since a large percentage of generated rectangles is eliminated from further consideration. We report here only computational results with the integer bisection, because in almost test problems this subdivision performs better than the bisection via the midpoint $r^k = (y^k + y^k)_{jk}$.

Table 1. The computational results with the integer bisection

Prob.	m	n	p	ite	s	time (second)
1	5	7	3	64	11	8.03
2	5	15	3	142	99	19.91
3	5	20	3	331	61	22
4	10	20	3	124	93	21.01
5	10	30	3	132	112	61.2
6	10	50	3	241	62	58.92
7	8	10	4	54	13	11.43
8	10	10	4	108	64	66.15
9	10	20	4	944	205	188
10	5	10	5	959	219	221
11	6	20	5	1009	129	340.6
12	10	10	5	718	247	571.0
13	5	15	6	659	119	228
14	7	20	6	709	208	441.3

Acknowledgements

The authors are grateful to the referees for their helpful comments and remarks. Thanks are also due to Mr. N. H. Son (University of Hue) for his cooperation in writing the code.

References

1. J. F. Benders, *Partitioning procedures for solving mixed variables programming problems*, Numerische Mathematik **4** (1962), 238-252.
2. M. Grötschel, M. Junger and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Operations Research **32** (1984), 1195-1220.
3. L. D. Muu and W. Oettli, *An algorithm for indefinite quadratic programming with convex constraints*, Operations Research Letters **10** (1989), 323-327.
4. L. D. Muu and W. Oettli, *A method for minimizing a convex-concave function over a convex set*, J. of Optimization Theory and Applications **70** (1990), 377-384.
5. T. J. Van Roy and L. A. Wolsey, *Solving mixed 0-1 programs*, Discrete Applied Mathematics **14** (1986), 199-213.
6. L. A. Wolsey, *Strong formulations for mixed integer programming: a survey*, Mathematical Programming **45** (1989), 173-192.

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