Operations Research

Implicit Enumeration for the Pure Integer 0/1 Minimax Programming Problem

Moustapha Diaby,

TECHNICAL NOTE

IMPLICIT ENUMERATION FOR THE PURE INTEGER 0/1 MINIMAX PROGRAMMING PROBLEM

MOUSTAPHA DIABY

Arizona State University, Tempe, Arizona

(Received December 1989; revisions received February 1991, February 1992; accepted April 1992)

We present an implicit enumeration procedure for solving pure integer 0/1 minimax problems which arise in the context of Benders decomposition for mixed integer 0/1 linear programming problems, or in various practical settings such as the location of facilities and assembly line balancing. The procedure is an extension of the additive algorithm of E. Balas for pure integer 0/1 programming problems. We solve minimax problems directly (i.e., as minimax problems, not as mixed integer programming problems). A numerical example is used to illustrate the procedure. Extensions of the basic algorithm are discussed.

The pure integer 0/1 minimax problem can be stated as follows.

Problem MP

$\mathrm{Min}_x\{\mathrm{Max}_{i\in M}[f_i(x)]\}$

subject to

$g_t(x) \leq 0 \quad t \in K$

$x_j \in \{0, 1\} \quad j \in N,$

where

$M = \{1, 2, \ldots, m\}$ is the set of indices of objective functions;
$N = \{1, 2, \ldots, n\}$ is the set of indices of variables;
$K = \{1, 2, \ldots, k\}$ is the set of indices of constraints;
$f_i(x) = a_{i0} + \sum_{j\in N} a_{ij}x_j \quad i \in M, \quad x = (x_j);$
$g_t(x) = b_{t0} + \sum_{j\in N} b_{tj}x_j \quad t \in K, \quad x = (x_j).$

This problem is important in the context of Benders decomposition because Benders' master problem can be formulated as a pure integer 0/1 minimax programming problem when the integer variables are binary. The problem is also important in its own right, because it can be found in many practical settings. For example, it is associated with several location/allocation problems in which the objective may not be to simply maximize profits or minimize costs. Such problems often arise in the context of locating hospitals, police stations, fire departments, etc., where one wants to minimize the maximum of some factor such as response time (or the maximum of functions measuring response time), for example, with respect to different demand points (see Hax and Candea 1984 or Mirchandani and Francis 1990). Also, the problem of minimizing assembly line cycle time, which arises in the context of assembly line balancing (see Baybars 1986), is essentially a pure integer 0/1 minimax programming problem.

The standard approach to solving **MP** has been to reformulate it as the following equivalent mixed integer programming problem.

Problem MIP

Minimize z

subject to

$z \geq f_i(x) \quad i \in M$

$0 \geq g_t(x) \quad t \in K$

$x_j \in \{0, 1\} \quad j \in N$

z unrestricted.

Owing to the fact that **MIP** has only one continuous variable, it has often been suggested (see Garfinkel and Nemhauser 1972 or Nemhauser and Wolsey 1988, for example) that it (**MIP**) not be solved as a mixed integer programming problem, but rather more efficiently as a pure integer programming problem.

Subject classifications: Production/scheduling: production line balancing. Programming, integer, algorithms: Benders decomposition.
Area of review: OPTIMIZATION.

0030-364X/93/4106-1172 $01.25

However, in all research known to us, aimed at exploiting the "almost-pure integer" nature of **MIP**, **MIP** is not solved optimally (see Lemke and Spielberg 1967, Balas 1970, Gorry, Shapiro and Wolsey 1972, Minchandani and Francis 1990, and the examples in Baybars 1986).

The purpose of this paper is to propose a new enumeration procedure for optimally solving problem **MP** directly. To simplify the presentation, we first consider the unconstrained version of the problem. The extension of the procedure to handle constraints is straightforward and will be discussed subsequently.

1. SOLUTION METHOD

In developing and stating the proposed solution method we use the following notation. Let

$$a_{ij}^+ \equiv \begin{cases} a_{ij} & \text{if } a_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij}^- \equiv \begin{cases} a_{ij} & \text{if } a_{ij} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

Then, clearly $a_{ij} = a_{ij}^+ + a_{ij}^-$ for all $(i, j) \in (M, N)$.

For $P \subseteq N$ and a p-vector $y^p = (y_{p,r_1}, y_{p,r_2}, \ldots, y_{p,r_p})'$, where $p = |P|$, $r_j \in P$ and $y_{p,r_j} \in \{0, 1\}$ for all $j \leq p$, let $J_{1p} \equiv \{r_j \mid y_{p,r_j} = 1\}$, $J_{op} \equiv \{r_j \mid y_{p,r_j} = 0\}$, $F_p \equiv N \backslash (J_{1p} \cup J_{op})$. Then, there exists a one-to-one correspondence between incomplete (partial) and complete enumerations of components of the decision vector x, and instances of y^p. Here J_{1p} corresponds to the set of indices of components of x that have been fixed at 1; J_{op} corresponds to the set of indices of components fixed at 0; and F_p corresponds to the set of indices of components not yet restricted (i.e., the set of "free" variables).

We refer to y^p as a complete (incomplete) solution if F_p is empty (nonempty). Hence, we are using the term solution in a broad sense.

Let B^n denote the set of all 0/1 n-vectors, where $n = |N|$. For any (complete or incomplete) solution y^p, define a set X^p:

$$X^p = \{x \in B^n \mid x_j = y_{pj} \quad \text{for} \quad j \in (J_{1p} \cup J_{op})\},$$

and an m-vector V^p: $V^p = (v_{p1}, v_{p2}, \ldots, v_{pm})'$ with $v_{pi} = \min_{x \in X^p}\{f_i(x)\}$, $i \in M$.

Finally, to every V^p, attach a scalar $\|V^p\|$, defined as $\|V^p\| = \max_{i \in M}\{v_{pi}\}$. The validity of $\|V^p\|$ as a bound for implicit enumeration is established in the following propositions.

Proposition 1. *Let* $Z^p = \min_{x \in X^p}\{\max_{i \in m}\{f_i(x)\}\}$ *denote the value of solution* y^p. *Then,* $Z^p = \|V^p\|$ *if* y^p *is complete.*

Proof. Here y^p complete implies $X^p = \{y^p\}$, which implies $v_{pi} = f_i(y^p)$, for all $i \in M$; $X^p = \{y^p\}$ implies $\|V^p\| = \max_{i \in M}\{f_i(y^p)\}$; $v_{pi} = f_i(y^p)$ for all $i \in M$, implies $Z^p = \max_{i \in M}\{f_i(y^p)\}$.

Proposition 2. *Let* $Z^p = \min_{x \in X^p}\{\max_{i \in M}\{f_i(x)\}\}$ *denote the value of solution* y^p. *Then,* $Z^p \geq \|V^p\|$ *if* y^p *is incomplete.*

Proof

$$Z^p = \min_{x \in X^p}\{\max_{i \in M}\{f_i(x)\}\}$$
$$\geq \max_{i \in M}\{\min_{x \in X^p}\{f_i(x)\}\} = \|V^p\|.$$

Proposition 3. *Restricting* x_k $(k \in F_p)$ *to* 1/0 *will cause* v_{pi} *to change (increase by exactly* $|a_{ik}|$*) if and only if* a_{ik} *is positive/negative.*

Proof. From the definition of the v_{pi}'s, we have

$$v_{pi} = a_{io} + \sum_{j \in J_{1p}} a_{ij} + \sum_{j \in F_p} a_{ij}^- \quad i \in M, \quad p \leq |N|. \tag{1}$$

The proposition follows immediately from (1).

Now, referring to a solution y^t as a *continuation* of a solution y^p if $F_t \subset F_p$, $J_{1p} \subseteq J_{1t}$, and $J_{op} \subseteq J_{ot}$, we have the following results.

Corollary 1. *For any continuation of* y^t *of* y^p, *we have* $v_{ti} = v_{pi} + \sum_{j \in T(p,t)}|a_{ij}|$ *for all* $i \in M$, *where* $T(p, t) = T_1 \cup T_2$, *with*

$$T_1 = \{j \mid j \in (J_{1t} \backslash J_{1p});\ a_{ij} > 0\}, \quad \textit{and}$$

$$T_2 = \{j \mid j \in (J_{ot} \backslash J_{op});\ a_{ij} < 0\}.$$

Proof. The proof follows directly from Proposition 3.

Corollary 2. *If* y^t *is a continuation of* y^p, *then* $\|V^t\| \geq \|V^p\|$.

Proof. The proof follows directly from Corollary 1.

Propositions 1 and 2 show that $\|V^p\|$ is a lower bound on the value of **MP.** Corollaries 1 and 2 show that this lower bound is nondecreasing with respect to restrictions. Finally, Proposition 3 can be used to compute the bound efficiently.

A flowchart of our proposed enumeration scheme which uses $\|V^p\|$ as a bound follows.

STEP 0. **(Initialization)**

Let $J_{1o} = \phi$; $F_o = n$
(all variables are unrestricted)
Set $Z^* = \infty$; $J^* = \phi$; $p = 0$
(Z^*: the initial incumbent value;

J^*: the set of variables fixed at 1 in the incumbent solution;
p: the level of branching (the number of variables restricted, or the node number))
Go to Step 1.

STEP 1. (**Bounding and Branching**)
Compute $\| V^p \|$
(use relation (1) if $p = 0$; use Proposition 3 otherwise)
If $\| V^p \| < Z^*$, then:
1a. If $F_p = \phi$, then:
(bound is better than current incumbent value; solution is complete: record solution as the new incumbent and backtrack).
Set $Z^* = \| V^p \|$
$J^* = J_{1p}$
Go to Step 2.
1b. Else: (i.e., $F_p \neq \phi$)
(bound is better than current incumbent value; solution is incomplete: restrict one "free" variable to zero (i.e., branch forward)).
Choose any $k \in F_p$
Set $F_{p+1} = F_p \backslash \{k\}$
$J_{1,p+1} = J_{1p}$
$p = p + 1$; $s_p = 0$
(s_p is a node "state" index: $s_p = 0$ (1) means that the "1-branch" has not been (has been) explored).
Go to Step 1.
Else (i.e., $\| V^p \| \geq Z^*$): Go to Step 2.

STEP 2. (**Backtracking**)
If $s_p = 0$, then:
(the 1-branch has not been explored (i.e., node p is "live"): explore the 1-branch (i.e., branch backward: backtrack and branch)).
2a. Let k be such that $F_{p-1} \backslash F_p = \{k\}$
Set $J_{1p} = J_{1p} \cup \{k\}$
$s_p = 1$
Go to Step 1.
Else:
2b. Set $p = p - 1$
(the 1-branch has been explored: exit if there are no more "live" nodes ($p = 0$), or keep backtracking until a live node is encountered).
If $p = 0$, Go to Step 3
Else, Go to Step 2.

STEP 3. (**Exiting**)

$$\text{Set } x_j^* = \begin{cases} 1 & \text{if } j \in J^* \\ 0 & \text{otherwise} \end{cases}$$

Stop; $x^* = (x_j^*)$ is an optimal solution to **MP**.

2. NUMERICAL EXAMPLE

Consider the problem below.

Problem MIP

Minimize z

subject to

$$z \geq 2173 - 2600x_1 - 2700x_2 - 4075x_3 - 4150x_4 = f_1(x)$$

$$z \geq 1605 + 250x_1 + 250x_2 + 50x_3 + 50x_4 = f_2(x)$$

$$z \geq 1199.5 - 1100x_1 - 2700x_2 + 50x_3 - 2075x_4 = f_3(x)$$

$$z \geq 1798 - 1100x_1 - 1200x_2 + 50x_3 + 50x_4 = f_4(x)$$

$$x_j \in \{0, 1\}; \quad j \in N = \{1, 2, 3, 4\}.$$

MIP is equivalent to the following problem.

Problem MP

$$\text{Min}_x\{\text{Max}_{i \in M}\{f_i(x)\}\}$$

$$x_j \in \{0, 1\}, \quad j \in N,$$

where $M = \{1, 2, 3, 4\}$; $N = \{1, 2, 3, 4\}$; $x = (x_j)$, $j \in N$; and the matrix of coefficients of the x_j's is:

$$\begin{bmatrix} -2{,}600 & -2{,}700 & -4{,}075 & -4{,}150 \\ 250 & 250 & 50 & 50 \\ -1{,}100 & -2{,}700 & 50 & -2{,}075 \\ -1{,}100 & -1{,}200 & 50 & 50 \end{bmatrix}$$

The steps of the search procedure are summarized in Table I, where the columns represent iterations.

Note that the components of V^p are computed according to (1) at the initial node (i.e., node 0) and according to Proposition 3 for other nodes. For example,

$$v_{o1} = a_{1o} + \sum_{j \in J_{1o}} a_{1j} + \sum_{j \in F_o} a_{1j}^-$$

$$= 2{,}173 - 2{,}600 - 2{,}700 - 4{,}075 - 4{,}150$$

$$= -11{,}352,$$

whereas for node 1 (which has $x_1 = 0$, since $1 \notin (J_{11} \cup F_1)$):

$$v_{11} = v_{o1} + |a_{11}|$$

$$= -11{,}352 + |-2{,}600| = -8{,}752.$$

Note also that the bound at a given node is equal to the greatest component of V^p at that node. For example, the bound at node 0 is $1{,}605 = \max\{-11{,}352, 1{,}605, -4675.5, -502\}$.

The optimal solution is $x^* = (0, 0, 0, 1)^t$ with a value of 1,848.

Table I
Search Procedure Steps

p	0	1	2	3	4	4	3	2	1	0
s_p		0	0	0	0	1	1	1	1	—
J_{1p}	ϕ	ϕ	ϕ	ϕ	ϕ	{4}	{3}	{2}	{1}	—
F_p	{1, 2, 3, 4}	{2, 3, 4}	{3, 4}	{4}	ϕ	ϕ	{4}	{3, 4}	{2, 3, 4}	{1, 2, 3, 4}
V^p	−11,352*	−8,752*	−6,052*	−1,977*	2,173*	−1,977	−6,052	−8,752	−11,352	—
	1,605*	1,605	1,605	1,605	1,605	1,655*	1,655*	1,855*	1,855*	—
	−4,675.5*	−3,575.5*	875.5*	875.5	1,199.5*	−875.5	−825.5*	−3,575.5	−4,675.5	—
	−502*	598*	1,798*	1,798	1,798	1,848*	1,848*	598	−502	—
$\lVert V^p \rVert$	1,605	1,605	1,798	1,798	2,173	1,848	1,848	1,855	1,855	—
Z^*	∞	∞	∞	∞	2,173	1,848	1,848	1,848	1,848	—
J^*	ϕ	ϕ	ϕ	ϕ	ϕ	{4}	{4}	{4}	{4}	—

Note that for **MIP**, the generation of the linear programming bound in a conventional branch-and-bound (BB) context requires solving one linear programming (LP) problem at each node. Each of these LP's, although not solved from scratch, could require several iterations, with up to 45 multiplications/divisions and 40 additions/subtractions per iteration (see Bazaraa, Jarvis and Sherali 1990) for the first level of branching, and possibly more for higher levels. The generation of our bound at each node requires at worst only four additions/subtractions because only the components of V^p marked by an * need be re-evaluated in the enumeration process according to Proposition 3. Also, because no divisions or multiplications are involved in our procedure, it eliminates the problem of round-off errors that can be encountered in solving an LP problem. Finally, the storage requirements for our procedure are minimal compared to that of the conventional LP-based BB.

3. EXTENSIONS/IMPROVEMENTS OF THE BASIC PROCEDURE

In this section, we discuss the extension of the basic procedure to handle constraints. We also discuss results that can be used to drop objectives from **MP**.

3.1. Extension to the General (Constrained) Minimax 0/1 Programming Problem

The extension of our procedure to handle constraints can be described as follows. Define y^p, X^p, J_{1p}, J_{op}, and F_p as in Section 1; let b_{tj}^+ and b_{tj}^- be such that:

$$b_{tj}^+ = \begin{cases} b_{tj} & \text{if } b_{tj} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$b_{tj}^- = \begin{cases} b_{tj} & \text{if } b_{tj} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

Then clearly, the b_{tj}^+ and b_{tj}^-'s are analogous to the a_{ij}^+'s and a_{ij}^-'s.

For every y^p, define a k-vector $R^p = (r_{p1}, r_{p2}, \ldots, r_{pk})^t$, where $r_{pt} = \min_{x \in X^p}\{g_t(x)\}$ for all $t \in K$. Then, r_{pt} is analogous to v_{pi}. Hence, Proposition 3 and Corollary 2, in particular, hold for r_{pt}. Proposition 3 provides an efficient way of computing r_{pt} from $r_{p-1,t}$. Clearly, if a constraint t is infeasible for a solution y^p, then it is infeasible for all continuations of y^p. Hence, the r_{pt}'s are suitable for restricting the search to feasible solutions. The basic procedure is extended to solve the full (constrained) version of **MP** by simply changing the test of Step 1 from:

"$\lVert V^p \rVert < Z^*$?" to "($\lVert V^p \rVert < Z^*$?)

and ($r_{pt} \leqslant 0$ for all $t \in K$?)".

3.2. Problem Reduction

The following results can be used to reduce problem **MP** at the preprocessing stage or during the enumeration process of a given implementation of the procedure.

Proposition 4. *Let $u_{pi} \equiv \max_{x \in X^p}\{f_i(x)\}$ for all $i \in M$. If $f_i(\cdot)$ and $f_j(\cdot)$ $(j \neq i)$ are such that $v_{pj} > u_{pi}$ for some y^p, then there exists no continuation y^t of y^p such that $\lVert V^t \rVert = v_{ti}$.*

Proof. By following a development similar to those in the proofs of Propositions 1, 2, and 3, it is easy to show that

$u_{pi} \geqslant u_{ti}$; also,

$$u_{ti} \geqslant v_{ti}; \text{ so that} \tag{2}$$

$u_{pi} \geqslant u_{ti} \geqslant v_{ti}$.

From Corollary 1, we have

$$v_{tj} \geqslant v_{pj}. \tag{3}$$

Combining $v_{pj} > u_{pi}$ with (2) and (3), gives

$$v_{tj} > u_{pi} \geqslant u_{ti} \geqslant v_{ti}. \tag{4}$$

Now, suppose that $\| V^t \| = v_{ti}$; then, we have:

$$v_{ti} = \max_{r \in M}\{v_{tr}\} \geq v_{tj}, \tag{5}$$

which contradicts (4). Hence, we must have $\| V^t \| \neq v_{ti}$.

Corollary 3. *If $f_i(\cdot)$ and $f_j(\cdot)$ ($j \neq i$) are such that $v_{pj} > u_{pi}$ for some solution y^p, $f_i(\cdot)$ can be eliminated from consideration for all continuations of y^p. Furthermore, if, in addition, $F_p = N$, $f_i(\cdot)$ can be deleted from the problem.*

Proof. The proof follows directly from Proposition 4.

Corollary 3 can be used to eliminate objective functions from **MP** at the initialization step of our procedure. For instance, referring back to the numerical example of Section 2, we have:

$$V^o = \begin{bmatrix} -11{,}352 \\ 1{,}605 \\ -4{,}675.5 \\ -502 \end{bmatrix} \quad \text{and} \quad U^o = \begin{bmatrix} 2{,}173 \\ 2{,}205 \\ 1{,}249.5 \\ 1{,}898 \end{bmatrix}.$$

Hence, $f_3(\cdot)$ can be deleted from the problem because $1{,}605 = v_{o2} > u_{o3} = 1{,}249.5$.

4. CONCLUSIONS

We have developed and presented a new bounding approach for the pure integer 0/1 minimax programming problem and an implicit enumeration procedure based on this bounding approach. The proposed bound can be computed much more efficiently than the conventional linear programming relaxation bound. Moreover, it requires no multiplication or division, and hence eliminates the possibility of round-off errors. In the computational experiments we conducted using randomly-generated problems with up to 200 variables, 30 objectives, and 20 constraints, the proposed overall procedure significantly outperformed the conventional linear programming branch-and-bound approach (see Diaby 1991 for details). However, in particular, further testing of the procedure is needed to determine its overall impact on convergence in a Benders decomposition framework.

ACKNOWLEDGMENT

The author is grateful to Professor Stanley Zionts of the State University of New York at Buffalo, and to Professors Alain Martel and Wilson Price of Laval University for their comments on earlier versions of this paper. An anonymous referee suggested several improvements to the notation and proofs.

REFERENCES

BALAS, E. 1970. Minimax and Duality for Linear and Nonlinear Mixed-Integer Programming. In *Integer and Nonlinear Programming*, J. Abadie (ed.). American Elsevier.

BAYBARS, I. 1986. A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Mgmt. Sci.* **32**, 909–932.

BAZARAA, M. S., J. J. JARVIS AND H. D. SHERALI. 1990. *Linear Programming and Network Flows*. John Wiley, New York, 193–194.

DIABY, M. 1991. Direct Search for Min-Max, Min-Sum Problems With Binary Variables. Working Paper No. 79, School of Business, North Carolina A&T State University, Greensboro.

GARFINKEL, R., AND G. NEMHAUSER. 1972. *Integer Programming*. John Wiley, New York.

GORRY, G. A., J. F. SHAPIRO AND L. A. WOLSEY. 1972. Relaxation Methods for Pure and Mixed Integer Programming Problems. *Mgmt. Sci.* **18**, 229–239.

HAX, A. C., AND D. CANDEA. 1984. *Production and Inventory Management*. Prentice-Hall, Englewood Cliffs, N.J., 49–58.

LEMKE, C. E., AND K. SPIELBERG. 1967. Direct Search Algorithms for Zero-One and Mixed Integer Programming. *Opns. Res.* **15**, 998–1022.

MIRCHANDANI, P. B., AND R. L. FRANCIS (eds.). 1990. *Discrete Location Theory*. John Wiley, New York, 209–262.

NEMHAUSER, G. L., AND L. A. WOLSEY. 1988. *Integer and Combinatorial Optimization*. John Wiley, New York, 409–417.