# THE MIXED INTEGER LINEAR BILEVEL PROGRAMMING PROBLEM

**JAMES T. MOORE**
*Offutt Air Force Base, Omaha, Nebraska*

**JONATHAN F. BARD**
*University of Texas, Austin, Texas*

(Received April 1988; revisions received March, June 1989; accepted July 1989)

A two-person, noncooperative game in which the players move in sequence can be modeled as a bilevel optimization problem. In this paper, we examine the case where each player tries to maximize the individual objective function over a jointly constrained polyhedron. The decision variables are variously partitioned into continuous and discrete sets. The leader goes first, and through his choice may influence but not control the responses available to the follower. For two reasons the resultant problem is extremely difficult to solve, even by complete enumeration. First, it is not possible to obtain tight upper bounds from the natural relaxation; and second, two of the three standard fathoming rules common to branch and bound cannot be applied fully. In light of these limitations, we develop a basic implicit enumeration scheme that finds good feasible solutions within relatively few iterations. A series of heuristics are then proposed in an effort to strike a balance between accuracy and speed. The computational results suggest that some compromise is needed when the problem contains more than a modest number of integer variables.

Planning in hierarchical organizations is an interactive process in which a central or supremal unit coordinates lower level units to optimize overall system performance. When the subunits are afforded a degree of autonomy, the situation may be complicated by the fact that their individual objectives do not always coincide with those of the leader. Such may be the case in a decentralized firm or government agency where different divisions and departments compete for limited resources.

Mathematical programming has often provided a basis for modeling hierarchical behavior, with decomposition techniques being used to solve problems of large scale (Geoffrion 1970, Burton and Obel 1977, Rouhani et al. 1985). The latter easily lend themselves to an economic interpretation which reflects the underlying algorithms. The complete process can be viewed as a series of adjustments where the supremal unit sends tentative information to the subunits, observes their reactions, and then fashions a new response. At each iteration, the transmitted information may take the form of prices on scarce resources or resource allocations to the subunits.

A major criticism of single objective formulations is that they fail to adequately account for the conflict and lack of cooperation that normally arise in hierarchical systems. Multicriteria methods (e.g., see Geoffrion and Hogan 1972 or Chankong and Haimes 1983) offer some relief, but they too typically model only one decision maker. As an alternative, we will investigate the following two-level problem.

Let the supremal unit be designated the *leader* and have control over the decision vector $\mathbf{x} \in X \subseteq R^{n_1}$; the subunits will collectively be called the *follower* and control the decision vector $\mathbf{y} \in Y \subseteq R^{n_2}$. (This can be generalized, somewhat, at the expense of notational simplicity by treating each subunit as a separate entity.) It will be assumed that the leader is given the first choice and selects an $\mathbf{x} \in \Omega(X) \subseteq X$ to minimize the objective function $F$. In light of this decision, the follower then selects a $\mathbf{y} \in Y \cap \Omega(\mathbf{x})$ to minimize the objective function $f$, where the sets $\Omega(X)$ and $\Omega(\mathbf{x})$ place additional restrictions on the feasible regions of the leader and follower, respectively. This leads to the bilevel programming problem (BLPP) (Aiyoshi and Shimizu 1981, Bard and Falk 1982, Bialas and Karwan 1984), which has found application in such diverse areas as transportation network design (LeBlanc and Boyce 1985), national agricultural planning (Fortuny-Amat and McCarl 1981), and the decentralized management of multidivisional firms (Bard 1983).

The purpose of this paper is to investigate the mixed integer linear case from an algorithmic point of view. Within the context of branch and bound, we discuss the prospect of finding global optima, and develop a

*Subject classifications:* Games: noncooperative. Programming, integer: branch-and-bound algorithms.


0030-364X/90/3805-0911 $01.25

set of heuristics that may be used when the computations become too burdensome. To specify the model, let $x^1$ be an $n_{11}$-dimensional vector of continuous variables and $x^2$ be an $n_{12}$-dimensional vector of discrete variables, where $\mathbf{x} \equiv (x^1, x^2)$ and $n_1 = n_{11} + n_{12}$. Similarly define $y^1$ as an $n_{21}$-dimensional vector of continuous variables and $y^2$ as an $n_{22}$-dimensional vector of discrete variables, where $\mathbf{y} \equiv (y^1, y^2)$ and $n_2 = n_{21} + n_{22}$. This leads to

$$\max_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}) = c^{11}x^1 + c^{12}x^2 + d^{11}y^1 + d^{12}y^2 \tag{1a}$$

subject to

$$\mathbf{x} \in X = \{\mathbf{x}: D^1x^1 + D^2x^2 \leqslant b^1\} \tag{1b}$$

$$\max_{\mathbf{y}} f(\mathbf{y}) = d^{21}y^1 + d^{22}y^2 \tag{1c}$$

subject to

$$g(\mathbf{x}, \mathbf{y}) = A^1x^1 + A^2x^2 + B^1y^1 + B^2y^2 \leqslant b^2 \tag{1d}$$

$$\mathbf{y} \in Y = \{\mathbf{y}: C^1y^1 + C^2y^2 \leqslant b^3\} \tag{1e}$$

$$\mathbf{x} \geqslant 0, \quad \mathbf{y} \geqslant 0, \quad x^2, y^2 \text{ integer} \tag{1d}$$

where the coefficients are of conformal dimension.

From the leader's perspective, (1) can be viewed as a mathematical program with an implicitly defined constraint region given by the follower's subproblem (1c)–(1e). Once the vector $\mathbf{x}$ is chosen, though, the subunits simply face a standard optimization problem. Alternatively, the BLPP can be interpreted as a static Stackelberg game (Simaan and Cruz 1973, Basar and Selbuz 1979, Tolwinski 1981) with full information.

A significant amount of effort has been devoted to solving the linear BLPP. A comprehensive discussion can be found in Bialas and Karwan where they outline both their *high point* algorithm and complementarity approach. More recently, Bard and Moore (1990) have proposed a branch-and-bound strategy that appears to offer the best performance to date.

With the exception of a code by Moore and Bard (1987) for the zero-one case, no effort has been made to solve the complete formulation (1). In response, we present a relatively efficient implicit enumeration scheme that finds global optima for a subclass of problems (see Moore 1988 for a production planning application). In general, though, the difficulty encountered in obtaining tight upper bounds argues for the use of heuristics. Those developed are shown to greatly speed convergence without appreciably degrading the quality of the solutions.

In the next section, basic assumptions and notation are presented along with two examples to help clarify the characteristics of the problem. Two bounding theorems are given in Section 2. The branch-and-bound methodology is discussed in Section 3 where a series of heuristics is proposed in the search for a balance in performance. Computational experience with problems that contain up to 40 variables is highlighted in Section 4. Section 5 concludes with a discussion of the results, and some suggestions for future research.

## 1. PRELIMINARIES

In the model, it is assumed that the leader has full knowledge of the follower's problem, and that cooperation is prohibited. This prevents the players from negotiating side payments, and introduces the possibility of non-Pareto optimal solutions.

The following notation is used in the sequel.

**BLPP Constraint Region**

$$\Omega = \{(\mathbf{x}, \mathbf{y}): \mathbf{x} \in X, \mathbf{y} \in Y, g(\mathbf{x}, \mathbf{y}) \leqslant b^2\}.$$

**Projection of $\Omega$ onto the Leader's Decision Space**

$$\Omega(X) = \{\mathbf{x} \in X: \exists \mathbf{y} \text{ such that } (\mathbf{x}, \mathbf{y}) \in \Omega\}.$$

**Follower's Feasible Region for $\mathbf{x} \in X$ Fixed**

$$\Omega(\mathbf{x}) = \{\mathbf{y}: \mathbf{y} \in Y, g(\mathbf{x}, \mathbf{y}) \leqslant b^2\}.$$

**Follower's Rational Reaction Set**

$$M(\mathbf{x}) = \{\mathbf{y}: \arg\max(f(\mathbf{y}'): \mathbf{y}' \in \Omega(\mathbf{x}))\}.$$

**Inducible Region**

$$\text{IR} = \{(\mathbf{x}, \mathbf{y}): \mathbf{x} \in \Omega(X), \mathbf{y} \in M(\mathbf{x})\}.$$

In order to ensure that (1) is well posed we make the additional assumption that $\Omega$ is nonempty and compact, and that for each decision taken by the leader, the follower has some room to respond; i.e., $\Omega(\mathbf{x}) \neq \varnothing$. The rational reaction set, $M(\mathbf{x})$, defines these responses while the inducible region, IR, represents the set over which the leader may optimize. Thus, in terms of the above notation, the BLPP can be written as

$$\max(F(\mathbf{x}, \mathbf{y}): (\mathbf{x}, \mathbf{y}) \in \text{IR}). \tag{2}$$

**Definition 1.** If $\bar{\mathbf{y}} \in M(\bar{\mathbf{x}})$ then $\bar{\mathbf{y}}$ is said to be optimal with respect to $\bar{\mathbf{x}}$; such a pair will be called *bilevel feasible.*

**Definition 2.** A point $(\mathbf{x}^*, \mathbf{y}^*)$ is said to be an *optimal solution* to the BLPP if

a. $(\mathbf{x}^*, \mathbf{y}^*)$ is bilevel feasible; and,
b. for all bilevel feasible pairs $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \in \mathrm{IR}$, $F(\mathbf{x}^*, \mathbf{y}^*) \geqslant F(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$.

Note that the assumption $\Omega(\mathbf{x}) \neq \varnothing$ for all $\mathbf{x} \in X$ is somewhat restrictive in that it rules out those situations in which the players' problems are tightly coupled. Such might be the case in armed conflict where the primary goal of the leader is to prevent the follower from initiating any sort of response.

The applications that we have in mind, though, are less severe, and logically exclude this situation. To cite one example, consider the regulatory problem in which the government would like to control the behavior or output of a particular industry. Here, the government assumes the role of leader and exercises its power through a combination of taxes, fines and quotas. However, if the resultant constraints are too tight, a significant portion of the industry may be driven out of business. This situation corresponds to the case where $\Omega(\mathbf{x})$ is empty, but most likely would be forestalled by either economic or political considerations.

### 1.1. Obstacles to Algorithmic Development

Algorithms designed to solve integer programs generally rely on some form of separation, relaxation and fathoming to construct ever tighter bounds on the solution. Separation is usually accomplished by placing contradictory constraints on a single integer variable. This approach is directly applicable to the mixed integer BLPP. The natural relaxation derives from the removal of the integrality requirements on the variables. Fathoming, however, presents several difficulties. In mixed integer programming (MIP), candidate subproblems are created by separation. Relaxation follows, and some technique is employed to determine if the relaxed subproblem contains the optimal solution. Accordingly, if the subproblem does not contain a feasible solution better than the incumbent (best feasible solution yet found), the subproblem can be dismissed from further consideration. This leads to three general fathoming rules.

**Rule 1.** The relaxed subproblem has no feasible solution.

**Rule 2.** The solution of the relaxed subproblem is no greater that the value of the incumbent.

**Rule 3.** The solution of the relaxed subproblem is feasible to the original problem.

Unfortunately, only Rule 1 in its original form still holds for the mixed integer BLPP. Rule 2 requires strong qualification, and Rule 3 must be discarded altogether. The following examples demonstrate these points.

**Example 1**

$$\underset{x}{\text{Maximize}}\ F(x, y) = x + 10y$$

where $y$ solves

$$\max_{y} f(x, y) = -y$$

$$\text{subject to} \quad -25x + 20y \leqslant 30$$
$$x + 2y \leqslant 10$$
$$2x - y \leqslant 15$$
$$2x + 10y \geqslant 15$$
$$x, y \geqslant 0$$
$$x, y \text{ integer.}$$

The BLPP constraint region $\Omega$ of this example is shown in Figure 1. When the integrality requirements are relaxed the solution is $(x, y) = (8, 1)$ with $F(x, y) = 18$ (note that this point is in the inducible region). The true optimum, though, obtained by enforcing integrality is $(x^*, y^*) = (2, 2)$ with $F(x^*, y^*) = 22$. As a consequence, we have the following observations.

**Observation 1.** The solution of the relaxed BLPP does not provide a valid bound on the solution of the mixed integer BLPP.

**Observation 2.** Solutions to the relaxed BLPP that are in the inducible region cannot, in general, be fathomed.

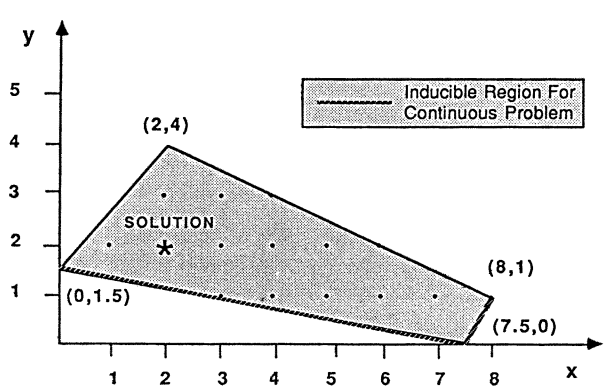


**Figure 1.** Constraint region for Example 1.

The next example illustrates how a branch-and-bound approach to the mixed integer BLPP can be thwarted if Rule 3 is applied.

**Example 2**

$$\underset{x}{\text{Maximize}}\ F(x, y) = -x - 2y$$

where $y$ solves

$$\max_{y} f(x, y) = y$$

subject to
$$-x + 2.5y \leq 3.75$$
$$x + 2.5y \geq 3.75$$
$$2.5x + y \leq 8.75$$
$$x, y \geq 0$$
$$x, y \text{ integer.}$$

The BLPP constraint region for Example 2 contains three integer points: (2, 1), (2, 2) and (3, 1). If the leader picks $x = 2$, the follower chooses $y = 2$ so $F = -6$. If the leader's decision is $x = 3$, the follower's choice is $y = 1$ so $F = -5$. Therefore, the optimal solution of the problem is $(x^*, y^*) = (3, 1)$ with $F = -5$.

In a typical depth-first branch-and-bound scheme, separation is done by forming subproblems with contradictory constraints on a single integer variable, and relaxation is accomplished by ignoring the integrality requirements. Thus, each subproblem is a mixed integer BLPP with tighter bounds placed on the integer variables. One of many branch-and-bound trees that could have arisen for this example is shown in Figure 2. At node 0, the relaxed solution is $(x, y) = (0, 1.5)$ with $F = -3$. Separating on the $y$ variable and adding the constraint $y \geq 2$ yields the subproblem at node 1. Solving the corresponding relaxation yields $(x, y) = (1.25, 2)$ with $F = -5.25$. Further separation, fathoming due to infeasibility, and backtracking, eventually puts us at node 9 whose solution is $(x, y) = (2, 1)$ with $F = -4$. This point, although integer, and in both $\Omega$ and the inducible region of the subproblem, is not bilevel feasible, and hence cannot be fathomed. The variable $x$ must be constrained further in order to uncover the optimal solution. In addition, note that the objective function value obtained at node 9 is not a valid bound; at node 7, if $y$ had been selected as the branching variable, the optimal solution would not have been found. Thus, we see the following observation.

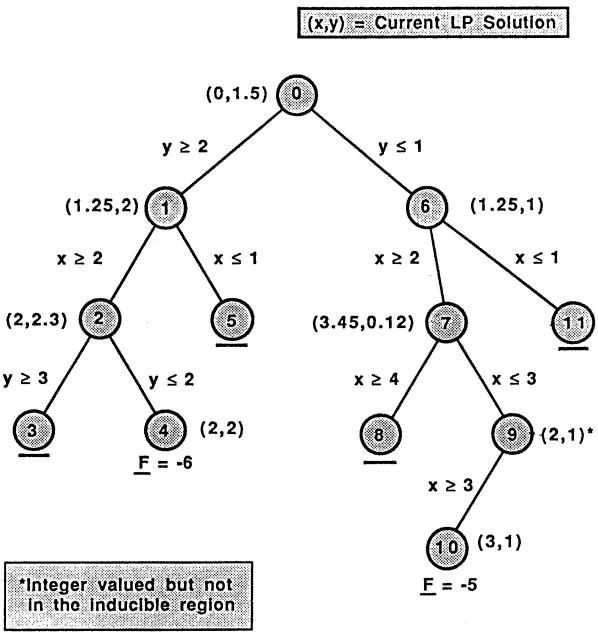


**Figure 2.** Branch-and-bound tree for Example 2.

**Observation 3.** All integer solutions to the relaxed BLPP with some of the follower's variables restricted cannot, in general, be fathomed.

### 1.2. Branch-and-Bound Notation

Before formally stating the conditions under which the second type of fathoming is applicable, some additional notation must be introduced. Let

$N = \{1, \ldots, n_1 + n_2\}$ be the index set of decision variables;
$N^1 = \{1, \ldots, n_{12}\}$ be the index set of the integer variables, $x^2$, controlled by the leader;
$N^2 = \{1, \ldots, n_{22}\}$ be the index set of integer variables, $y^2$, controlled by the follower;
$U^1$ = the $n_{12}$-dimensional vector of original upper bounds on the integer variables controlled by the leader;
$U^2$ = the $n_{22}$-dimensional vector of original upper bounds on the integer variables controlled by the follower.

If an integer variable is unbounded above, then the corresponding entry in the upper bound vector is $\infty$. The initial lower bound on each integer variable is assumed to be zero. For subproblem $k$, the sets of

bounds on the variables are

$$H_k^1 = \{(\alpha^{1k}, \beta^{1k}):$$
$$0 \leqslant \alpha_j^{1k} \leqslant x_j^8 \leqslant \beta_j^{1k} \leqslant U_j^1, j \in N^1\}$$
$$H_k^2 = \{(\alpha^{2k}, \beta^{2k}):$$
$$0 \leqslant \alpha_j^{2k} \leqslant y_j^2 \leqslant \beta_j^{2k} \leqslant U_j^2, j \in N^2\}$$

where $\alpha^{1k}$ and $\beta^{1k}$ are $n_{12}$-dimensional vectors of lower and upper bounds, respectively, placed on the integer variables controlled by the leader; $\alpha^{2k}$ and $\beta^{2k}$ are $n_{22}$-dimensional vectors of lower and upper bounds, respectively, placed on the integer variables controlled by the follower. For subproblem $k$, the notation $H_k^2(0, \infty)$ is used to indicate that no bounds other than the original bounds specified in problem (1) are placed on the integer variables controlled by the follower. In addition, by $H_l^1 \subseteq H_k^1$ we mean $\alpha^{1k} \leqslant \alpha^{1l}$ and $\beta^{1l} \leqslant \beta^{1k}$. Thus, if node $k$ is along the path to node $l$, the subproblem associated with node $l$ is derived from the subproblem associated with node $k$, implying that $H_l^1 \subseteq H_k^1$ and $H_l^2 \subseteq H_k^2$.

The index sets of the integer variables that are restricted in subproblem $k$ are

$$S_k^1 = \{j: \alpha_j^{1k} > 0 \text{ or } \beta_j^{1k} < U_j^1, j \in N^1\}$$
$$S_k^2 = \{j: \alpha_j^{2k} > 0 \text{ or } \beta_j^{2k} < U_j^2, j \in N^2\}.$$

Accordingly, a restricted variable is one that has additional lower or upper bounds placed on it other than those found in (1b) and (1e).

Define $F_k^C$ as the optimal solution of the relaxed BLPP at subproblem $k$. $F_k^C$ is found by dropping the integrality requirements and solving the resultant BLPP. After collapsing constraints (1b) and (1e) into (1d), and letting $b \equiv (b^1, b^2, b^3)^T$, the formulation is given by

$$\max_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}) = c^{11}x^1 + c^{12}x^2 + c^{21}y^1 + c^{22}y^2 \tag{3a}$$

where $y$ solves

$$\max_{\mathbf{y}} f(\mathbf{y}) = d^{21}y^1 + d^{22}y^2 \tag{3b}$$

subject to

$$a^1x^1 + A^2x^2 + B^1y^1 + B^2y^2 \leqslant b \tag{3c}$$
$$x^1, x^2, y^1, y^2 \geqslant 0 \tag{3d}$$

and the bounds $H_k^1$ and $H_k^2$. (3e)

The *high point* solution to (3) is found by solving the linear program that results when the follower's objective function (3b) is removed from the formulation. Denote this value by $F_k^H$ for subproblem $k$.

## 2. BOUNDING THEOREMS

In this section, sufficient conditions are derived that indicate when the solution of a relaxed subproblem may be used as an upper bound for the mixed integer BLPP. As a consequence, subproblems whose solutions satisfy Rule 2 may be fathomed.

**Theorem 1.** *Given $H_k^1$ and $H_k^2(0, \infty)$, let $(\mathbf{x}^k, \mathbf{y}^k)$ be the high point solution to the corresponding relaxed BLPP. Then $F_k^H = F(\mathbf{x}^k, \mathbf{y}^k)$ is an upper bound on the solution of the mixed integer BLPP at node $k$.*

**Proof.** Let $(\mathbf{x}^{*l}, \mathbf{y}^{*l})$ solve the mixed integer BLPP at node $l$ where $H_l^1 \subseteq H_k^1$ and $H_l^2 \subseteq H_k^2(0, \infty)$. Assume that $F(\mathbf{x}^{*l}, \mathbf{y}^{*l}) > F_k^H$. However, this leads to a contradiction because $(x^{*l}, y^{*l})$ is a feasible solution of the high point problem at node $k$.

This result says that the high point solution at node $k$ may be used as a bound to determine if the subproblem can be fathomed. This bound is only applicable when no restrictions on the integer variables controlled by the follower have been made along the path to node $k$ (i.e., $S_k^2 = \varnothing$). In other words, once the leader has made a decision, the follower is free to optimize the objective function without regard to any *a priori* or *artificial* restrictions.

The following theorem indicates when $F_k^H$ provides a valid upper bound for the case where $S_k^2 \neq \varnothing$.

**Theorem 2.** *Given $H_k^1$ and $H_k^2$, let $(\mathbf{x}^k, \mathbf{y}^k)$ be the high point solution to the corresponding relaxed BLPP. Then $F_k^H = F(\mathbf{x}^k, \mathbf{y}^k)$ is an upper bound on the mixed integer BLPP defined by the current path in the tree if none of the $y_j^{2k}$ are at either $\alpha_j^{2k} > 0$ or $\beta_j^{2k} < U_j^2$ for $j \in S_k^2$.*

**Proof.** Let $(\mathbf{x}^{*l}, \mathbf{y}^{*l})$ solve the mixed integer BLPP at node $l$ where $H_l^1 \subseteq H_k^1$ and $H_l^2 \subseteq H_k^2$. Assume that $F(\mathbf{x}^{*l}, \mathbf{y}^{*l}) > F_k^H$. Thus, $(\mathbf{x}^{*l}, \mathbf{y}^{*l})$ cannot be feasible to the high point problem at node $k$. Construct the line joining $(\mathbf{x}^k, \mathbf{y}^k)$ and $(\mathbf{x}^{*l}, \mathbf{y}^{*l})$ as follows. Let $(\mathbf{x}', \mathbf{y}') = \gamma(\mathbf{x}^k, \mathbf{y}^k) + (1 - \gamma)(\mathbf{x}^{*l}, \mathbf{y}^{*l})$, $\gamma \in [0, 1]$. So for $\gamma > 0$, $F(\mathbf{x}', \mathbf{y}') = F[\gamma(\mathbf{x}^k, \mathbf{y}^k) + (1 - \gamma)(\mathbf{x}^{*l}, \mathbf{y}^{*l})] = \gamma F(\mathbf{x}^k, \mathbf{y}^k) + (1 - \gamma)F(\mathbf{x}^{*l}, \mathbf{y}^{*l})$ because $F$ is linear. It is assumed that $F(\mathbf{x}^{*l}, \mathbf{y}^{*l}) > F_k^H$ so $\gamma F(\mathbf{x}^k, \mathbf{y}^k) + (1 - \gamma)F(\mathbf{x}^{*l}, \mathbf{y}^{*l}) > \gamma F(\mathbf{x}^k, \mathbf{y}^k) + (1 - \gamma)F(\mathbf{x}^k, \mathbf{y}^k) = F(\mathbf{x}^k, \mathbf{y}^k)$. Thus, $F(\mathbf{x}', \mathbf{y}') > F(\mathbf{x}^k, \mathbf{y}^k)$. However, this contradicts the optimality of $(\mathbf{x}^k, \mathbf{y}^k)$ because for $\gamma$ sufficiently small, $(\mathbf{x}', \mathbf{y}')$ is feasible to the high point problem.

In order for Theorem 2 to be applicable none of the restricted integer variables controlled by the follower may be at their bound in the high point solution. This is a fairly strong condition that may not arise frequently enough to furnish good bounds. The following corollary offers some improvement.

**Corollary 1.** *Given $H_k^1$ and $H_k^2$, let $(\mathbf{x}^k, \mathbf{y}^k)$ be the high point solution of the corresponding relaxed BLPP with the restrictions in $H_k^2$ relaxed. Then $F_k^H = F(\mathbf{x}^k, \mathbf{y}^k)$ is an upper bound on the mixed integer BLPP defined by the current path in the tree.*

**Proof.** Relaxing the restrictions in $H_k^2$ is equivalent to replacing $H_k^2$ with $H_k^2$ $(0, \infty)$. Thus, Theorem 1 may be invoked.

Unfortunately, it does not appear that any stronger bounds are available. In the BLPP, once the leader has made a decision, the follower is free to respond without regard to any a priori restrictions encountered by the leader in the branch-and-bound tree. This contrasts sharply with the standard mixed integer program where all such bounds are valid.

## 3. AN ALGORITHM FOR THE MIXED INTEGER BLPP

The algorithm developed for solving the mixed integer BLPP takes a depth-first branch-and-bound approach, incorporating the modifications just discussed. In particular, fathoming is only done when the relaxed BLPP is infeasible or when the upper bound at a node, as determined by the high point, is less than or equal to the value of the incumbent, denoted by $\underline{F}$. The necessary bookkeeping is facilitated by explicit reference to the set $H_k^1$, $H_k^2$, $S_k^1$, and $S_k^2$.

*Step 0.* (Initialization) Put $k = 0$. Set the parameters in $H_k^1$ and $H_k^2$ to the bounds of the mixed integer BLPP. Put $S_k^1 = \varnothing$, $S_k^2 = \varnothing$, $\underline{F} = -\infty$.

*Step 1.* (Upper Bounds and Fathoming) Attempt to find the high point solution of (3) and calculate $F_k^H$. If infeasible or $F_k^H \leqslant \underline{F}$, go to Step 6.

*Step 2.* (Continuous Solution) Attempt to solve the relaxed BLPP (3). If infeasible go to Step 6. If successful, label the solution $(\mathbf{x}^k, \mathbf{y}^k)$.

*Step 3.* (Branching) If integrality requirements are satisfied by $(\mathbf{x}^k, \mathbf{y}^k)$, go to Step 4. Otherwise, select an $x_j^{2k}$, $j \in N^1$, or $y_j^{2k}$, $j \in N^2$, which is fractional-valued. Place a new bound on the selected variable. Put $k \leftarrow k + 1$ and update $H_k^1$, $H_k^2$, $S_k^1$ and $S_k^2$. Go to Step 1.

*Step 4.* (Bilevel Feasible Solution) Fix $\mathbf{x}$ at $\mathbf{x}^k$ and solve the follower's problem to obtain $(\mathbf{x}^k, \hat{\mathbf{y}}^k)$. Compute $F(\mathbf{x}^k, \hat{\mathbf{y}}^k)$ and put $\underline{F} = \max[\underline{F}, F(\mathbf{x}^k, \hat{\mathbf{y}}^k)]$.

*Step 5.* (Integer Branching) If $\alpha_j^{1k} = \beta_j^{1k}$ for each $j \in N^1$, and $\alpha_j^{2k} = \beta_j^{2k}$ for each $j \in N^2$, go to Step 6. Otherwise, select an integer variable such that $\alpha_j^{1k} \neq \beta_j^{1k}$, $j \in N^1$, or $\alpha_j^{2k} \neq \beta_j^{2k}$, $j \in N^2$, and place a new bound on it. Put $k \leftarrow k + 1$ and update $H_k^1$, $H_k^2$, $S_k^1$ and $S_k^2$. Go to Step 1.

*Step 6.* (Backtracking) If no live node exists, go to Step 7. Othewise, branch to the newest live node, put $k \leftarrow k + 1$ and update $H_k^1$, $H_k^2$, $S_k^1$ and $S_k^2$. Go to Step 1.

*Step 7.* (Termination) If $\underline{F} = -\infty$, there is no feasible solution to (1). Otherwise, terminate with the point in the inducible region associated with $\underline{F}$.

A flow diagram of the algorithm is depicted in Figure 3. At Step 0, parameter values in $H_k^1$ and $H_k^2$ are initialized and the index sets, $S_k^1$ and $S_k^2$, used in the branch-and-bound tree are set to $\varnothing$. At Step 1, problem (3) is solved to find the high point solution

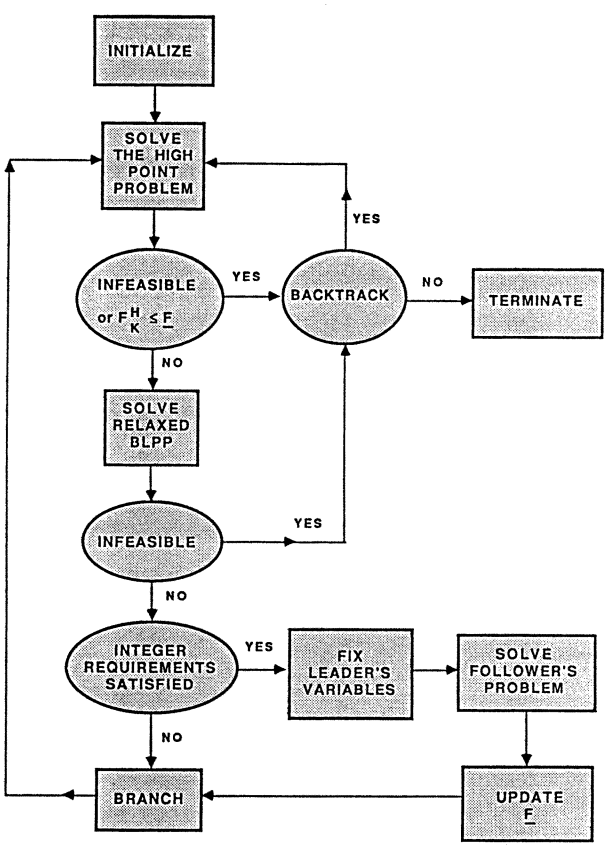


**Figure 3.** Flow chart of mixed integer BLPP algorithm.

and a potential upper bound. The relaxed BLPP is solved at Step 2 to advance the search for bilevel feasibility.

Branching occurs at Step 3 where many rules for selecting the branching variable were considered. The rule that performed best chooses the integer variable with the largest fractional part and branches in the direction that is most likely to increase the leader's objective function. For example, if $x_j^{2k}$, $j \in N^1$ is chosen at iteration $k$, and $c_j^2 \geq 0$, then the restriction $x_j^2 \geq [x_j^2] + 1$ is imposed on subproblem $k+1$ (where $[\cdot]$ denotes the largest integer less than or equal to the argument). An analysis of the 11 branching rules examined can be found in Moore. If no integer variable has a fractional value, the algorithm proceeds to Step 4 where a point in the inducible region is found. This is accomplished by fixing the variables controlled by the leader at their values arising at Step 2, and solving the following MIP.

$$\max_{\mathbf{y}} f(\mathbf{y}) = d^{21}y^1 + d^{22}y^2 \tag{4a}$$

subject to

$$B^1y^1 + B^2y^2 \leq b - A^1x^1 - A^2x^2 \tag{4b}$$

$$y^1,\ y^2 \geq 0;\quad y^2 \text{ integer.} \tag{4c}$$

Call $(\mathbf{x}^k, \hat{\mathbf{y}}^k)$ the solution of (4). If $F(\mathbf{x}^k, \hat{\mathbf{y}}^k) \geq \underline{F}$, the incumbent is updated. Next, the algorithm goes to Step 5 where an integer variable is selected for branching. Recall from Examples 1 and 2 that it is not possible to fathom an all integer solution. A list processing scheme is used to make the branching decision. If all the integer variables are fixed in the branch-and-bound tree, the algorithm goes to Step 6 and backtracks. Termination occurs at Step 7. If $\underline{F} \neq -\infty$, the optimality of the incumbent must be addressed. The following conditions are sufficient to guarantee that the true solution has been found.

**Proposition 1.** *If all the variables controlled by the leader are discrete, the algorithm finds the optimal solution to the mixed integer BLPP* (1).

**Proof.** Steps 3 and 5 ensure that all possible combinations of $x$ are implicitly considered; the subproblem solved at Step 4 ensures that all incumbent solutions are in the inducible region.

**Proposition 2.** *Assume that an optimum exists for problem* (1) *and that all the variables controlled by the follower are continuous. Then, if fathoming rules* 2 *and* 3 *are applied, the algorithm always terminates with the optimum.*

**Proof.** In this case, $N^2 = \varnothing$ so the bounds found at Step 2 are always valid. That is, the continuous BLPP solution, $F_k^c$, is always greater than or equal to the true optimum at node $k$ because the latter is always feasible to the former, and no a priori restrictions are ever placed on the follower's variables. An additional implication is that a solution that satisfies integrality can be fathomed. Thus, Rules 2 and 3 can be applied. Arguments similar to those made in the proof of Proposition 1 assure convergence.

Proposition 2 was foreshadowed by the work of Fortuny-Amat and McCarl who reformulated the continuous BLPP as an MIP by first appending the follower's Kuhn–Tucker conditions to the leader's problem, and then linearizing the resultant complementarity constraints with the aid of $m$ zero–one variables. Thus, when $N^2 = \varnothing$, the mixed integer BLPP can similarly be reformulated. By way of contrast, Proposition 1 states that when at least one of the variables controlled by the leader is continuous the algorithm may not produce an optimal solution. Fortunately, this situation can only occur when the optimum is not well defined.

To see this, consider Example 1 but with integer restrictions on the follower's variable only. Recall that the solution of the original problem is $(x^*, y^*) = (2, 2)$ with $F = 22$. The new solution, however, is $(x^*, y^*) = (2.5-\epsilon, 2)$ *with* $F = 22.5-\epsilon$ where $\epsilon > 0$ is arbitrarily small. Note that if the leader chooses $x = 2.5$, the follower picks $y = 1$, giving the inferior result $F = 12.5$. Thus, a discontinuity exists in the rational reaction set at $x = 2.5$. More formally, as the sequence $x^k \to \bar{x} = 2.5$ from below $(k \to \infty)$, the corresponding sequence $y^k \in M(x^k)$ has $y^k \to \bar{y} = 2$, but $2 \notin M(2.5)$. By implication, neither the point-to-set map, $M(x)$, nor the inducible region, IR, is closed at $x = 2.5$ in the sense of Hogan (1973). Consequently, *max* must be replaced by *sup* in (1a), which, for this example, is unattainable. (For an example where $M(\bar{x})$ fails to be open at $\bar{x}$ see Bard and Moore.)

The algorithm cannot recognize this situation and iterates until the normal termination conditions are met. In this example, $(x, y) = (2, 2)$ is incorrectly chosen as the optimum. Aside from the cases identified by Propositions 1 and 2, it remains an open question whether convergence can be proven for the general model.

### 3.1. Some Heuristics

For problems with more than a few integer variables, the computational burden may become excessive if good bounds are not forthcoming. But even then,

**Table I**
Basic Algorithm and Variants

| No. | Description |
|---|---|
| 1 | The basic algorithm |
| 2 | Algorithm 1 with Step 4 replaced by Step 4a |
| 3 | Algorithm 1 with Step 4 replaced by Step 4b |
| 4 | Algorithm 1 with Step 4 replaced by Step 4c |
| 5 | Algorithm 1 with Step 2 replaced by Step 2a |
| 6 | Algorithm 5 with Step 4 replaced by Step 4a |
| 7 | Algorithm 5 with Step 4 replaced by Step 4b |
| 8 | Algorithm 5 with Step 4 replaced by Step 4c |

optimality may never be confirmed as Example 3 demonstrates. (The procedure always terminates with a point in the inducible region though.) As a consequence, a series of heuristics designed to make the algorithm less enumerative has been investigated. Each represents a tradeoff between accuracy and computational effort.

To begin, we consider using the solution of the relaxed BLPP obtained at Step 2 as an alternative upper bound. This can be incorporated in the algorithm by replacing Step 2 with the following step.

*Step 2a.* Attempt to solve the relaxed BLPP. If infeasible, go to Step 6. Alternatively, label the solution $(\mathbf{x}^k, \mathbf{y}^k)$ and compute $F(\mathbf{x}^k, \mathbf{y}^k)$. If $F(\mathbf{x}^k, \mathbf{y}^k) \leq \underline{F}$, go to Step 6; otherwise go to Step 3.

Next, we introduce three new fathoming rules based on integrality. Each is described and meant as a replacement for Step 4.

*Step 4a.* Fix $\mathbf{x}$ at $\mathbf{x}^k$ and solve the follower's problem to obtain $(\mathbf{x}^k, \hat{\mathbf{y}}^k)$. Compute $F(\mathbf{x}^k, \hat{\mathbf{y}}^k)$. If $\underline{F} < F(\mathbf{x}^k, \hat{\mathbf{y}}^k)$, put $\underline{F} \leftarrow F(\mathbf{x}^k, \hat{\mathbf{y}}^k)$ and go to Step 5; otherwise go to Step 6.

*Step 4b.* Fix $\mathbf{x}$ at $\mathbf{x}^k$ and solve the follower's problem to obtain $(\mathbf{x}^k, \hat{\mathbf{y}}^k)$. Compute $F(\mathbf{x}^k, \hat{\mathbf{y}}^k)$ and put $\underline{F} = \max[\underline{F}, F(\mathbf{x}^k, \hat{\mathbf{y}}^k)]$. If $\hat{\mathbf{y}} = \mathbf{y}^k$, go to Step 6; otherwise go to Step 5.

*Step 4c.* Fix $\mathbf{x}$ at $\mathbf{x}^k$ and solve the follower's problem to obtain $(\mathbf{x}^k, \hat{\mathbf{y}}^k)$. Compute $F(\mathbf{x}^k, \hat{\mathbf{y}}^k)$ and put $\underline{F} = \max[\underline{F}, F(\mathbf{x}^k, \hat{\mathbf{y}}^k)]$. Go to Step 6.

Step 4a allows the algorithm to backtrack when the point in the inducible region derived from a solution at the current node has a value less than or equal to that of the incumbent. Step 4b sends the algorithm to Step 6 when the solution at the current node satisfies the integrality requirements and is in the inducible region. Step 4c permits backtracking simply when the integrality requirements are satisfied. Table I lists the seven additional algorithms that can be formed by using various combinations of the above steps. Each terminates with a feasible, but not necessarily, optimal solution to (1).

## 4. COMPUTATIONAL EXPERIENCE

In order to judge the comparative performance of the eight algorithms, 50 test problems were randomly generated and solved. The experimental design called for 10 classes of problems, each distinguished by the number of integer and continuous variables controlled by the players. In all cases, the coefficients of the $A$ and $B$ matrices took on values between $-15$ and 45 with approximately 25% of the entries being less than zero. Each had a density of about 40%. The coefficients of the two objective functions varied between $-20$ and 20 with approximately 50% being less than zero. The number of constraints in each problem was set at 0.4 times the total number of variables, and the right-hand side values ranged between 0 and 50 (Table II). The signs of the constraints had a 0.7 probability of being $\leq$ and a 0.3 probability of being $\geq$.

All computations were performed on an IBM 3081-D using the VS FORTRAN compiler. The continuous BLPPs arising at Step 4 were solved with the Bard-Moore code after making minor modifications to account for the complementarity conditions associated with the follower's integer variables in the branch-and-bound tree (see Moore).

### 4.1. Results

Five separate runs were made for each class of problems. Table III displays the results for the average CPU time (in seconds) required by each algorithm to reach termination. As might have been expected,

**Table II**
Number of Variables in Test Problems

| Class No. | No. of Variables $(n_1 + n_2)^a$ | No. of Follower Variables $(n_2)$ | No. of Integer Leader Variables $(n_{12})$ | No. of Integer Follower Variables $(n_{22})$ |
|---|---|---|---|---|
| 1 | 15 | 5 | 2 | 3 |
| 2 | 15 | 5 | 3 | 4 |
| 3 | 20 | 10 | 5 | 5 |
| 4 | 25 | 10 | 5 | 5 |
| 5 | 30 | 15 | 5 | 5 |
| 6 | 30 | 15 | 7 | 8 |
| 7 | 35 | 15 | 5 | 5 |
| 8 | 35 | 15 | 8 | 9 |
| 9 | 40 | 20 | 5 | 5 |
| 10 | 40 | 20 | 10 | 10 |

[a] The number of constraints $m = 0.4\ (n_1 + n_2)$.

**Table III**
Average CPU Time (sec) For Algorithms[a]

| Problem Class | Algorithm 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 1 | 3 | 1 | 3 | 1 | 3 | 1 |
| 2 | 210 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 186 | 27 | 105 | 17 | 49 | 12 | 35 | 9 |
| 4 | 73 | 5 | 5 | 4 | 6 | 5 | 5 | 4 |
| 5 | 678 | 77 | 837 | 70 | 683 | 57 | 730 | 50 |
| 6 | 244 | 46 | 113 | 41 | 99 | 36 | 97 | 32 |
| 7 | 170 | 42 | 33 | 28 | 41 | 38 | 28 | 24 |
| 8 | 634 | 178 | 422 | 175 | 89 | 72 | 84 | 66 |
| 9 | 366 | 103 | 235 | 81 | 118 | 78 | 107 | 59 |
| 10 | 730 | 272 | 454 | 311 | 123 | 123 | 118 | 118 |

[a] Each problem class comprises five samples.

Algorithms 4 and 8, which fathom integer solutions without regard to feasibility, performed best, while Algorithm 1 proved to be an order of magnitude slower. Note that an artificial limit of 995 nodes was placed on all problems to guard against excessive run times. The need for this restriction became apparent after a form of cycling was observed at Step 5. In particular, Algorithm 1 had a tendency to get into a loop where one variable would be continually selected for branching until all of its possible integer values were explored.

When the 995 node limit was reached, the incumbent was presented as the solution. Algorithm 1 reached this limit in eight out of the fifty problems examined. Algorithm 3, however, encountered this condition only three times, while Algorithms 5 and 7 terminated prematurely only twice. Algorithms 2, 4, 6 and 8 always ran to completion. Looping was not evident in any of the problems arising in classes 1, 7, 9 and 10.

Table IV displays the average number of nodes needed by each algorithm to reach termination, and

**Table IV**
Average Number of Nodes in Branch-and-Bound Tree

| Problem Class | Algorithm 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 86 | 21 | 56 | 18 | 51 | 21 | 49 | 18 |
| 2 | 250 | 12 | 10 | 8 | 14 | 12 | 10 | 8 |
| 3 | 541 | 60 | 313 | 54 | 169 | 32 | 166 | 26 |
| 4 | 243 | 11 | 11 | 8 | 12 | 10 | 10 | 7 |
| 5 | 483 | 49 | 415 | 42 | 343 | 26 | 362 | 20 |
| 6 | 380 | 40 | 239 | 35 | 225 | 27 | 223 | 24 |
| 7 | 90 | 15 | 15 | 11 | 13 | 11 | 9 | 6 |
| 8 | 366 | 115 | 249 | 117 | 51 | 40 | 47 | 37 |
| 9 | 136 | 30 | 84 | 25 | 41 | 20 | 38 | 15 |
| 10 | 128 | 42 | 70 | 40 | 30 | 30 | 27 | 27 |

Table V gives the average number of nodes required to find the best feasible solution. As can be seen, the results in the three aforementioned tables are highly correlated. A question remains, though, as to the quality of the final solutions.

On eleven of the fifty problems, the algorithms did not reach the same feasible solution. The bottom row of Table V indicates the number of times each algorithm failed to arrive at the presumed optimum. On four of the eleven disagreements, Algorithm 1 did *not* provide the best feasible solution. In each case, the algorithm terminated because it exceeded the node limit. Algorithms 3, 5 and 7 each terminated once at the node limit without reaching a feasible solution as good as that found by one of the other algorithms. On three of the problems that the algorithms did not reach

**Table V**
Average Number of Nodes to Find Best Feasible Solution

| Problem Class | Algorithm 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 60 | 19 | 56 | 18 | 49 | 19 | 48 | 18 |
| 2 | 44 | 10 | 10 | 8 | 12 | 10 | 10 | 8 |
| 3 | 212 | 33 | 201 | 28 | 148 | 27 | 146 | 23 |
| 4 | 10 | 7 | 6 | 6 | 7 | 7 | 6 | 6 |
| 5 | 78 | 26 | 60 | 22 | 62 | 20 | 57 | 16 |
| 6 | 216 | 23 | 216 | 21 | 209 | 22 | 208 | 21 |
| 7 | 59 | 8 | 8 | 5 | 10 | 8 | 8 | 5 |
| 8 | 94 | 90 | 36 | 96 | 46 | 34 | 44 | 34 |
| 9 | 107 | 26 | 74 | 23 | 38 | 17 | 37 | 14 |
| 10 | 36 | 20 | 27 | 20 | 20 | 20 | 20 | 20 |
| Failures | 4 | 5 | 6 | 8 | 4 | 5 | 6 | 8 |

the same solution, node limit violation is the explanation. For the fourth problem, the best feasible solution was found by Algorithm 3 alone. On this particular problem, Algorithm 3 terminated because it had reached the node limit. Algorithms 1, 5 and 7 terminated at the node limit with solutions less than the one found by Algorithm 3. The other algorithms terminated normally without achieving the same solution as Algorithm 3.

For the other seven disagreements, Algorithm 1 alone found the best feasible solution three times. Of the four remaining cases, Algorithms 1, 2, 5 and 6 found the best solution twice, Algorithms 1, 3, 5 and 7 found the best solution once, and Algorithms 1, 2, 3, 5, 6 and 7 found the best solution once.

From an examination of the individual runs, it was seen that CPU time varies greatly with both problem size and specific realization. This point is illustrated

by the results obtained for the class 5 problems. For half the algorithms, the average CPU time exceeded that for all other classes. In addition, fluctuations of up to 300 seconds within the class were observed for Algorithms 1 and 5.

### 4.2. Interpretation

To put the above discussion into tighter focus, consider Table VI which contains a qualified measure of how well each algorithm performed given that it did not find the best solution. The entries, $\delta_{c\alpha}$, are computed as

$$\delta_{c\alpha} = \sum_{i=1}^{5} \left[ \frac{F_{ic}^{*} - \hat{F}_{i\alpha}}{F_{ic}^{*} \cdot n_{c\alpha}} \right] \times 100\%, \qquad c = 1, \ldots, 10; \quad \alpha = 1, \ldots, 8 \quad (5)$$

where

$F_{ic}^{*}$ is the best solution found for problem $i$ in class $c$ ($c = 1, \ldots, 10$);
$\hat{F}_{i\alpha}$ is the best solution found for problem $i$ (in class $c$) by algorithm $\alpha$ ($\alpha = 1, \ldots, 8$); and
$n_{c\alpha}$ is the number of times algorithm $\alpha$ did not find $F_{ic}^{*}$ for the 5 problems ($i = 1, \ldots, 5$) in class $c$;

and represent conditional averages for each problem class $c$; that is, the average of the percentage difference between the best solution found for a particular problem in class $c$ and the best solution found for that problem by algorithm $\alpha$. A "—" in the table indicates that the relevant algorithm found the best solution for all five problems in the corresponding class. When agreement was not total, the algorithms generally failed only once to find the best solution for the five problems. That is, when $\delta_{c\alpha} > 0$, $n_{c\alpha}$ was usually 1. The exceptions are explicitly noted.

To see how to interpret the data in Table VI, consider the column Algorithm 3. For the five problems in class 3 (third row), Algorithm 3 failed to find the best solution twice; the average disagreement was 21%. In addition, it failed to find the best solution once for each of the five problems in classes 6, 7 and 8. (For the one problem in class 8, the discrepancy was 29,770%!)

**Table VI**
Average Percent Difference Between Best Solution Found and the Best Solution Found By a Particular Algorithm[a]

| Problem Class | Algorithm | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | — | — | — | — | — | — | — | — |
| 2 | 4 | — | — | — | — | — | — | — |
| 3 | — | — | 21[b] | 662[c] | — | — | 21[b] | 662[c] |
| 4 | — | — | — | — | — | — | — | — |
| 5 | 58 | 15[b] | — | 15[b] | 4 | 23[b] | 4 | 23[b] |
| 6 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 7 | — | 17 | 18 | 18 | 17 | 17 | 18 | 18 |
| 8 | 29770 | 0.08 | 29770 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| 9 | — | — | — | — | — | — | — | — |
| 10 | — | — | — | — | — | — | — | — |

[a] A "—" indicates that the algorithm always found the best solution; unless otherwise noted, $n_{c\alpha} = 1$.
[b] Represents an average of two problems ($n_{c\alpha} = 2$).
[c] Represents an average of three problems ($n_{c\alpha} = 3$).

### 4.3. Assessment

An approximate ranking of the algorithms by CPU time from fastest to slowest yields 8, 6, 4, 2, 7, 5, 3 and 1. Of course, reduced accuracy is likely to accompany an increase in speed. Algorithms 4 and 8, for example, failed to reach the best available solution in eight of the fifty instances. And for the three problems in class 3 a discrepancy was observed, and to a lesser degree for the three problems in classes 5 and 7, the results were less than satisfactory (see Table VI). These algorithms backtrack when a point satisfying the integrality requirements is uncovered. Similarly, Algorithms 2 and 6 failed to reach the best feasible solution 10% of the time. They backtrack when no improvement is provided by a solution which satisfies the integrality requirements. Algorithms 3 and 7 failed to find the best feasible solution on six of the problems, while Algorithm 5 failed on four. Algorithms 3 and 7 backtrack when the solution of (3) is in the inducible region, and Algorithm 5 backtracks when the solution of (3) is no better than the incumbent.

In summary then, if it is desirable to find the best feasible solution regardless of CPU time, Algorithm 1 without a node limit is the proper choice. However, if a tradeoff can be made between the quality of the solution and CPU time, then Algorithm 5 offers a good compromise. If CPU time is of primary importance, Algorithm 8 is recommended. Nevertheless, the best feasible solutions uncovered by each of the algorithms invariably arise well within 995 nodes, and in many instances, before the 50th subproblem is solved.

## 5. CONCLUSIONS

This paper represents a first step in the development of procedures for solving the mixed integer BLPP. The difficulties that arise in obtaining exact solutions stem from the inappropriateness of using common relaxation techniques to derive tight upper bounds. As a result, subproblems within a branch-and-bound tree

can only be fathomed when they are shown to be infeasible.

These points, as well as the fact that it is not possible to verify optimality for a large class of problems, were demonstrated with a number of simple examples. Consequently, a series of heuristics was proposed to find approximate solutions. To some extent each sacrifices accuracy for speed, but in almost all cases, good feasible solutions emerged within a relatively few iterations. Nevertheless, the challenge remains to devise an algorithm that can guarantee optimality in a computationally efficient manner. To this end, a variety of approaches based on cutting plane and decomposition techniques are currently being explored.

## ACKNOWLEDGMENT

This work was partially supported by a grant under the Advanced Research Program of the Texas Higher Education Coordinating Board.

## REFERENCES

AIYOSHI, E., AND K. SHIMIZU. 1981. Hierarchical Decentralized Systems and Its New Solution by a Barrier Method. *IEEE Trans. Syst. Man Cybern.* **SMC-11,** 444–449.

BARD, J. F. 1983. Coordination of a Multidivisional Organization Through Two Levels of Management. *OMEGA* **11,** 457–468.

BARD, J. F., AND J. E. FALK. 1982. Explicit Solution to the Multi-Level Programming Problem. *Comput. Opns. Res.* **9,** 77–100.

BARD, J. F., AND J. T. MOORE. 1990. A Branch and Bound Algorithm for the Bilevel Programming Problem. *SIAM J. on Scientific and Statistical Computing* **11,** 281–292.

BASAR, T., AND H. SELBUZ. 1979. Closed Loop Stackelberg Strategies With Applications in Optimal Control of Multilevel Systems. *IEEE Trans. Auto. Control* **AC-24,** 166–178.

BIALAS, W. F., AND M. H. KARWAN. 1984. Two-Level Linear Programming. *Mgmt. Sci.* **30,** 1004–1020.

BURTON, R. M., AND B. OBEL. 1977. The Multilevel Approach to Organizational Issues of the Firm: A Critical Review. *OMEGA* **5,** 395–444.

CHANKONG, V., AND Y. Y. HAIMES. 1983. *Multiobjective Decision Making: Theory and Methodology.* North-Holland, New York.

FORTUNY-AMAT, J., AND B. MCCARL. 1981. A Representation and Economic Interpretation of a Two-Level Programming Problem. *J. Opnl. Res. Soc.* **32,** 783–792.

GEOFFRION, A. M. 1970. Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems. *Opns. Res.* **18,** 375–403.

GEOFFRION, A. M., AND W. W. HOGAN. 1972. Coordination of Two-Level Organizations With Multiple Objectives. In *Techniques of Optimization,* A. V. Balakrishnan (ed.). Academic Press, New York.

HOGAN, W. W. 1973. Point-To-Set Maps in Mathematical Programming. *SIAM Rev.* **15,** 591–603.

LEBLANC, L. J., AND D. E. BOYCE. 1985. A Bilevel Programming Algorithm for Exact Solution of the Network Design Problem With User-Optimal Flows. *Trans. Res.* **20B,** 259–265.

MOORE, J. T. 1988. Extensions to the Multilevel Linear Programming Problem. Ph.D. Dissertation, Department of Mechanical Engineering, University of Texas, Austin.

MOORE, J. T., AND J. F. BARD. 1987. An Algorithm for the Zero-One Bilevel Programming Problem. Department of Mechanical Engineering, University of Texas, Austin.

ROUHANI, R., L. LASDON, W. LEBOW AND A. D. WARREN. 1985. A Generalized Benders Decomposition Approach to Reactive Source Planning in Power Systems. *Math. Prog. Study* **25,** 62–75.

SIMAAN, M., AND J. B. CRUZ, JR. 1973. On the Stackelberg Strategy in Nonzero-Sum Games. *J. Optim. Theory Appl.* **11,** 533–555.

TOLWINSKI, B. 1981. Closed-Loop Stackelberg Solution to a Multistage Linear-Quadratic Game. *J. Optim. Theory Appl.* **34,** 485–501.