Mathematical Programming 40 (1988) 15-27
North-Holland

CONVEX TWO-LEVEL OPTIMIZATION

Jonathan F. BARD

Operations Research Group, Department of Mechanical Engineering, University of Texas, Austin, TX 78712, USA

Received 13 January 1986
Revised manuscript received 20 March 1987

In this paper a model for a two-level planning problem is presented in the form of a static Stackelberg game. By assumption, play is sequential and noncooperative; however, the leader can influence the actions of the followers through a set of coordination variables while the followers' responses may partly determine the leader's payoff.

Under certain convexity assumptions, it is shown that the feasible region induced by the leader is continuous in the original problem variables. This observation, coupled with two corollary results, are used as a basis for a hybrid algorithm which clings to the inducible region until a local optimum is found. A branching scheme is then employed to located other segments of the region, eventually terminating with the global optimum. A number of examples are given to highlight the results, while the algorithm's performance is tested in a comparison with two other procedures.

Key words: Bilevel programming, Stackelberg games, branch and bound, active set strategy.

1. Introduction

In the standard formulation of a Stackelberg game the dominant player is designated the 'leader' and has control over the decision vector $x \in X \subseteq R^{n_1}$ while the 'followers' individually control the decision vectors $y_p \in Y_p \subseteq R^{n_{2p}}$, $p = 1, \ldots, P$ (see Basar and Selbuz, 1979; Simaan and Cruz, 1973; Tolwinski, 1981). It will be assumed that the leader is given the first choice and selects an $x \in \Omega(X) \subseteq X$ to minimize his objective function F. In light of this decision, the followers select a $y_p \in Y_p \cap \Omega_p(x)$ to minimize their individual objective functions f_p, where $F: X \times Y \to R^1$. $Y = Y_1 \times Y_2 \times \cdots \times Y_p \subset R^{n_2}$, $f_p: X \times Y_p \to R^1$, and the sets $\Omega(X)$ and $\Omega_p(x)$ place additional restrictions on the feasible regions of the leader and followers, respectively. For the static case, this leads to the bilevel programming problem (BLPP) (Aiyoshi and Shimizu, 1981; Bard and Falk, 1982; Bard, 1983a; Bialas and Karwan, 1984; Fortuny-Amat and McCarl, 1981) given below.

$$\min_x F(x, y(x)) \tag{1a}$$

$$\text{subject to} \quad x \in X = \{x: H(x) \geqslant 0\}, \tag{1b}$$

$$\min_{y_p} f_p(x, y_p) \tag{1c}$$

$$\text{subject to} \quad g_p(x, y_p) \geqslant 0, \tag{1d}$$

$$y_p \in Y_p = \{y_p \colon G_p(y_p) \geqslant 0\}, \tag{1e}$$

$$p = 1, \ldots, P,$$

where H, G_p, and g_p are vector valued functions of dimension m_1, m_{2p}, and m_{3p}, respectively. For convenience define $m = \sum_p (m_{2p} + m_{3p})$.

In this paper we present an active set approach (see Gill et al., 1981) within the context of branch and bound to solve the "quadratic" BLPP. In the next section assumptions and basic notation are offered along with an example to help fix ideas. The algorithm is presented in Section 3 and demonstrated with a simple example. Two problems taken from the literature are analyzed in Section 4 and our general computational experience is highlighted in Section 4. Section 5 concludes with a discussion of the results, and some suggestions for future research.

2. Terminology, assumptions, and properties

In the sequel it will be assumed that the leader has full knowledge of the followers' problems, and that cooperation is prohibited. This precludes the use of correlated strategies and side payments. The following notation will be used to facilitate the development.

BLPP Constraint Region:

$$\Omega = \{(x, y) \colon x \in X, y_p \in Y_p, g_p(x, y_p) \geqslant 0, \ p = 1, \ldots P\}.$$

Projection of Ω onto the Leader's Decision Space:

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

Follower p's Feasible Region for $x \in X$ fixed:

$$\Omega_p(x) = \{y_p \colon y_p \in Y_p, g_p(x, y_p) \geqslant 0\}.$$

Follower p's Rational Reaction Set:

$$M_p(x) = \{y_p \colon \min(f_p(x, y_p) \colon y_p \in \Omega_p(x))\}.$$

Inducible Region:

$$\mathrm{IR} = \{(x, y) \colon x \in \Omega(X), y_p \in M_p(x), p = 1, \ldots, P\}.$$

In order to ensure that (1) is well posed we make the additional assumption that Ω is nonempty and compact, and that for all decisions taken by the leader, each follower has some room to respond; i.e., $\Omega_p(x) \neq \emptyset$. The rational reaction sets, $M_p(x)$, define 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

$$\min(F(x, y) \colon (x, y) \in \mathrm{IR}). \tag{2}$$

It should be mentioned that in practice the leader will incur some cost in determining the decision space, $\Omega(X)$, over which he may operate. For example, when the BLPP is used as a model for a decentralized firm with the corporate office representing the leader and the divisions representing the followers, coordination of lower level activities by the former requires detailed knowledge of production capacities, technological capabilities, and routine operating procedures (see Bard, 1983b). Up-to-date information in these areas is not likely to be available to corporate planners without constant monitoring and oversight.

For the remainder of the paper it will be assumed that F, $-H$, and $-G_p$ are convex in all their arguments, $-g_p$ is convex in y for x fixed, f_p is strictly convex in y for x fixed, and that all functions are twice continuously differentiable. This assures that all solutions to the subproblems (1c)-(1e) are unique, implying that $M_p(x)$ is single-valued and that the inducible region could be replaced by a unique response function, say, $y = \Psi(x)$. As a consequence we have:

Proposition 1. *Under the above assumptions on the functions in problem* (1), *the inducible region,* IR, *is continuous.*

The basis for the proof can be found in (Hogan, 1973, Corollary 8.1); the same result was established for the linear BLPP by Bard (1984) using duality arguments. The fact that the inducible region is continuous will be exploited in the algorithm presented in the next section, as will the following result of Bard and Falk (1982).

Proposition 2. *Let f_p, $-g_p$, and $-G_p$ be continuous and convex in y_p for all $x \in X$ and assume that a constraint qualification holds for* (1c)-(1e) *with x fixed at x^*. Then a necessary and sufficient condition that (x^*, y^*) solves* (1) *is that there exists a $u^* \in R^{m_{2p}}$ and a $U^* \in R^{m_{3p}}$ such that (x^*, y^*, u^*, U^*) solves*

$$\min_{x,y,u,U} F(x, y) \tag{4a}$$

$$\textit{subject to} \quad x \in X, \tag{4b}$$

$$\nabla_{y_p} f_p(x, y_p) - u_p \nabla_{y_p} g_p(x, y_p) - U_p \nabla_{y_p} G_p(y_p) = 0, \tag{4c}$$

$$u_p g_p(x, y_p) + U_p G_p(y_p) = 0, \tag{4d}$$

$$g_p(x, y_p) \geqslant 0, \tag{4e}$$

$$G_p(y_p) \geqslant 0, \tag{4f}$$

$$u_p \geqslant 0, \quad U_p \geqslant 0, \tag{4g}$$

$$p = 1, \ldots, P,$$

where u_p and U_p are the m_{2p} and m_{3p}-dimensional vectors of Kuhn-Tucker multipliers associated with the followers' problems.

Corollary 1. *If f_p $(p=1,\ldots,P)$ is quadratic in (x, y_p) and Ω is polyhedral then the inducible region is piecewise linear.*

Proof. As x is varied the solutions to the subproblems (1c)–(1e) either occur on a face of dimension $\leqslant m-1$ of Ω or in its interior. In the latter case we have $\nabla_{y_p} f_p(x, y_p)=0$ from (4c). The result then follows from Proposition 1. □

Corollary 2. *Let $F(x, y)$ be strictly convex in (x, y), $f_p(x, y_p)$ be quadratic in (x, y_p), and Ω be polyhedral. If $z^1=(x^1, y^1, u^1, U^1)$ and $z^2=(x^2, y^2, u^2, U^2)$ are local solutions to problem (4) and both lie on the same face of Ω then that face cannot be in* IR.

Proof. Let $\bar{z}=(\bar{x}, \bar{y}, \bar{u}, \bar{U})=\alpha z^1+(1-\alpha)z^2$, $\alpha\in[0,1]$ be a line on the face common to z^1 and z^2 and assume $\bar{z}$ satisfies constraints (4b)–(4g); i.e., $(\bar{x}, \bar{y})\in$ IR. From strict convexity we have $F(\bar{z})<\alpha F(z^1)+(1-\alpha)F(z^2)$ which, for α in the neighborhood of 0 or 1, contradicts the assumption that z^1 and z^2 are local optima. In particular, (4d) must be violated. □

Corollary 3. *Let $F(x, y)$ be convex in (x, y), $f_p(x, y_p)$ be quadratic in (x, y_p), and Ω be polyhedral. Then the set of solutions to problem (1) is convex.*

Example 1

$$\min_{x\geqslant 0} F(x, y)=(x-5)^2+(2y+1)^2,$$

$$\min_{y\geqslant 0} f(x, y)=(y-1)^2-1.5xy$$

$$\begin{aligned}\text{subject to}\quad & 3x-y\geqslant 3,\\ & -x+0.5y\geqslant -4,\\ & -x-y\geqslant -7,\end{aligned}$$

Figure 1 displays the BLPP constraint region, Ω, and the inducible region, IR, for Example 1. Notice that the latter is nonconvex, and unlike the case where all the functions are linear, does not lie wholly on the faces of Ω; however, its piecewise linear nature can be observed. Nonconvexity portends the existence of local solutions which are located at (1, 0) and (5, 2), and thus suggests that even the simplest of formulations may be difficult to solve without resorting to some type of branch and bound or cutting plane procedure. Note that if (1, 0) and (5, 2) are joined by adding the constraint $0.5x-y\leqslant 0.5$ to the example, Corollary 2 states that if these points remain local optima (which they do) then the hyperplane $0.5x-y=0.5$ cannot be in IR.

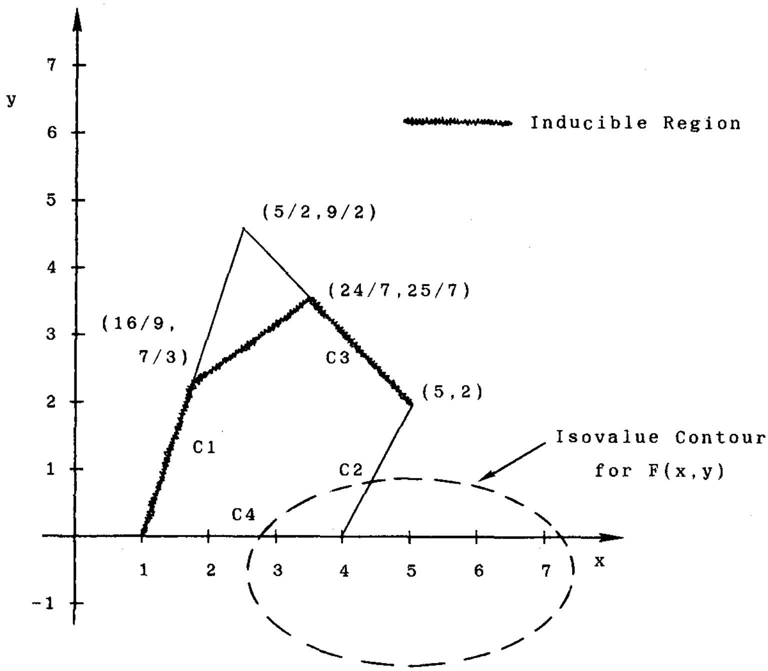

Fig. 1. Geometry and inducible region for Example 1.

3. Algorithmic developments

The implicit nature of $M_p(x)$, and hence, the inducible region, militates against solving problem (1) directly. Recognizing this fact, Aiyoshi and Shimizu (1981) used a barrier method to recast (1c)–(1e) into an unconstrained problem of the form

$$\min_{y_p} P_p^r(x, y_p) = f_p(x, y_p) + r\beta(g_p(x, y_p), G_p(y_p)), \quad r > 0, \tag{5}$$

where $\beta(g_p, G_p)$ is a barrier function appropriately defined on the interior of $\Omega(x)$. They then replaced (5) with its first order stationarity condition to obtain

$$\min_{x,y} F(x, y) \tag{6a}$$

$$\text{subject to} \quad x \in X = \{x: H(x) \geq 0\}, \tag{6b}$$

$$\nabla_{y_p} P_p^r(x, y_p) = 0, \quad p = 1, \ldots, P. \tag{6c}$$

Under the same convexity assumptions stated in Section 2, they were able to show that for the strictly null sequence $\{r^k\}$, solutions to (6) converge to a local solution of (1). Bard and Falk (1982) took a similar approach by exploiting Proposition 2, but in either case solution times became excessive as the problem size grew.

In an attempt to improve on current techniques a generalized branch and bound algorithm is proposed for the case where F is strictly convex, f is quadratic, and g

is affine. Extensions to the case where f and $-g$ are both convex in y for x fixed are straightforward. The basic idea is to first find a point in the inducible region, then iterate using an active set strategy to arrive at a local solution to (4). This will furnish a good upper bound on F that will prove valuable in subsequent fathoming. For ease of presentation it will be assumed from hereon out that $p=1$ and that $g(x, y) \equiv (g_1(x, y_1), G_1(y_1))^{\mathrm{T}}: R^{n_1} \times R^{n_2} \to R^m$. Notationally, let $I = \{1, \ldots, m\}$. Each path Π_k in the branch and bound tree will correspond to an assignment of either $u_i = 0$ or $g_i = 0$ for $i \in I_k \subseteq I$, thus identifying a partial solution. More specifically, let

$$S_k^+ = \{i: i \in I_k \text{ and } g_i = 0\},$$

$$S_k^- = \{i: i \in I_k \text{ and } u_i = 0\},$$

$$I_k^- = \{i: i \notin I_k\} = I \backslash I_k,$$

$$J_k^- = \{i: i \in S_k^-, g_i = 0, I_k^- = \emptyset\}.$$

A completion of I_k will be an assignment of either u_i or g_i for all i in the index set I_k^- of free variables. Note that when $I_k^- = \emptyset$ we are in the inducible region. The last set, J_k^-, identifies those surfaces which are incident to the current segment of the inducible region and, by Proposition 1, candidates for extending it in the direction of decreasing F. In all, the branch and bound tree contains $2^{m+1}-1$ nodes. Upper and lower bounds on F will be represented by $\bar{F}$ and $\underline{F}$, respectively, while $S \backslash i$ will be used to denote $S \backslash \{i\}$ and $S \cup i$ to denote $S \cup \{i\}$ for any set S and element i.

Algorithm

Step 1: (Initialization) Solve $\min\{F(x, y): x \in X,\ \nabla_y f(x, y) - u \nabla_y g(x, y) = 0,\ g(x, y) \geq 0,\ u \geq 0\}$ to get (x^0, y^0, u^0), and put $\underline{F} = F(x^0, y^0)$; also put $\bar{F} = \infty$.

Step 2: (Obtaining a Feasible Point) Fix x at x^0 and solve $\min\{f(x^0, y): g(x^0, y) \geq 0\}$ to get a point $(x^0, \hat{y}^0)$ in IR. If $F(x^0, \hat{y}^0) = \underline{F}$ stop; otherwise determine the partition at $(x^0, \hat{y}^0)$ and fix S_k^+ and S_k^- for path Π_k.

Step 3: (Bounding) Solve $\min\{F(x, y): x \in X, \nabla_y f(x, y) - u \nabla_y g(x, y) = 0, g_i(x, y) = 0,\ i \in S_k^+;\ u_i = 0,\ i \in S_k^-\}$ to obtain (x^k, y^k); put $\bar{F} = \min[\bar{F}, F(x^k, y^k)]$ and stop if $\bar{F} = \underline{F}$; if no solution exists fathom the node. Go to Step 4.

Step 4: (Advancing and Fathoming) Select an $i \in J_k^-$, and call it i_1. If none exists fathom all nodes on the path Π_k such that $g_i = 0$ and go to Step 5. If the number of binding constraints is $< n_1 + n_2$ then put $k \leftarrow k+1$, $S_k^+ \leftarrow S_k^+ \cup i_1$, $S_k^- \leftarrow S_k^- \backslash i_1$, and $J_k^- \leftarrow J_k^- \backslash i_1$; otherwise select an $i \in S_k^+$, call it i_2 and put $k \leftarrow k+1$, $S_k^+ \leftarrow S_k^+ \cup i_1 \backslash i_2$, $S_k^- \leftarrow S_k^- \cup i_2 \backslash i_1$, and $J_k^- \leftarrow J_k^- \backslash i_1$. Go to Step 3.

Step 5: (Backtracking) Select a new $i \in J_t^-$ $(t = 1, \ldots, k-1)$ and call it i_1. If none exists fathom all nodes on the paths Π_t such that $g_i = 0$ and to to Step 6; otherwise select an $i \in S_t^+$, call it i_2 and put $k \leftarrow k+1$, $S_k^+ \leftarrow S_t^+ \cup i_1 \backslash i_2$, $S_k^- \leftarrow S_t^- \cup i_2 \backslash i_1$, and $J_k^- \leftarrow J_t^- \backslash i_1$. Go to Step 3.

Step 6: (Branching) Select an unfathomed node on any of the paths Π_k such that $i \in S_k^-$ and $g_i \neq 0$, call it i_3; if none exists go to Step 8; otherwise go to Step 7.

Step 7: (Fathoming) Solve $\min\{F(x, y)\colon x \in X,\ \nabla_y f(x, y) - \Sigma_i u_i \nabla_y g_i(x, y) = 0,\ g(x, y) \geq 0,\ g_{i_3}(x, y) = 0\}$ to get a point (x^0, y^0) in IR. If $F(x^0, y^0) \geq \bar{F}$, fathom all nodes on paths containing $g_{i_3} = 0$ and go to Step 5; otherwise put $k \leftarrow k+1$ and go to Step 2.

Step 8: (Degenerate Case) Attempt to solve $\min\{F(x, y)\colon x \in X,\ \nabla_y f(x, y) = 0,\ g(x, y) \geq 0,\ u_i = 0,\ \forall i\}$ to get a point (x^0, y^0) in IR. If a solution exists update $\bar{F}$. Go to Step 9.

Step 9: (Termination) Declare the feasible point in IR associated with $\bar{F}$ the optimal solution.

At Step 1 a relaxed problem is solved to get a lower bound on F. The resultant value of x is then used at Step 2 to locate a point in the inducible region and to test for global optimality. At Step 3 we minimize over the active constraint surface represented by S_k^+ to get (x^k, y^k), and update the upper bound accordingly. This is an attempt to satisfy Proposition 2. At Step 4, what might be termed an 'entering constraint' is identified (i_1) which has the property that it is incident to the current segment of the inducible region; that is, $g_{i_1}(x^k, y^k) = 0$ and $i_1 \notin S_k^+$. A practical way of choosing i_1 is to solve: $\min_i[-\nabla F(x^k, y^k) \cdot \nabla g_i(x^k, y^k) / \|\nabla F(x^k, y^k)\| \cdot \|\nabla g_i(x^k, y^k)\|]$ which determines the surface on which F is decreasing most rapidly. If all the elements in J_k^- have been exhausted, then, from the continuity of the inducible region (Proposition 1), we can conclude that a local optimum has been found. Finally, the choice of the 'leaving constraint' (i_2) if necessary is problem dependent and thus fairly arbitrary. With certain qualifications the constraint associated with the smallest multiplier seems to produce good results.

At Step 4 when a local solution is reached all nodes along the paths where $g(x^k, y^k) = 0$ can be fathomed, once again due to the continuity of IR and Corollary 2. This turns out to be an extremely powerful step, trimming the branch and bound tree by a factor as large as 2^M (where M is the number of active constraints). Backtracking at Step 5 is designed to return to a previously encountered point in the inducible region and to continue minimizing, but in a different direction. When all the potential paths are explored we branch at Step 6 and then solve a new subproblem at Step 7 to return to IR. The last problem to be solved at Step 8 corresponds to the case where all u_i are zero. The procedure terminates when every node has been fathomed.

Theorem 1. *For F strictly convex in (x, y), f quadratic in (x, y), and g affine, the Algorithm terminates with a global optimum to problem* (1).

Proof. By construction, the Algorithm explores all nodes of the underlying branch and bound tree. And by hypothesis, all subproblems solved at Step 3 are convex, so for a given completion, no superior local optimum is overlooked. □

Bookkeeping — The path Π_k in the branch and bound tree can be concisely represented by an l-dimensional vector, where l is the current depth of the tree.

The order of the components is determined by their "level." Indices only appear in the vector Π_k if they are in I_k, and appear underlined if the complementary condition has already been considered. To facilitate branching to any node in the tree the following notation is used. If $i \in I_k$, let is appear in Π_k as

$$\begin{cases} i & \text{if } i \in S_k^+ \text{ and } u_i = 0 \text{ has not been considered,} \\ \underline{i} & \text{if } i \in S_k^+ \text{ and } u_i = 0 \text{ has been considered,} \\ -i & \text{if } i \in S_k^- \text{ and } g_i = 0 \text{ has not been considered,} \\ -\underline{i} & \text{if } i \in S_k^- \text{ and } g_i = 0 \text{ has been considered.} \end{cases}$$

When branching to say $g_t = 0$ from node ν_k we simply change Π_k to (Π_k, t). In backtracking, the rightmost nonunderlined entry is underlined, its sign changed, and all entries to its right erased. If this procedure is followed branching between any two nodes can occur without overlooking the optimal solution.

Demonstration — In order to see how the algorithm works let us return to Example 1 and Figure 1. Solving the relaxed problem at Step 1 yields the point (4, 0) and a lower bound $\underline{F} = 2$. Fixing x at 4 and solving the subproblem at Step 2 puts us on constraint 3 (C3) in the inducible region. Thus $S_1^+ = \{3\}$, $S_1^- = \{1, 2, 4\}$, $I_1^- = \emptyset$, and $(x^0, \hat{y}^0) = (4, 3)$. Minimizing F over the current segment of the inducible region at Step 3 gives $(x^1, y^1) = (5, 2)$, an upper bound of $\bar{F} = 25$, and $J_1^- = \{2\}$. The first path

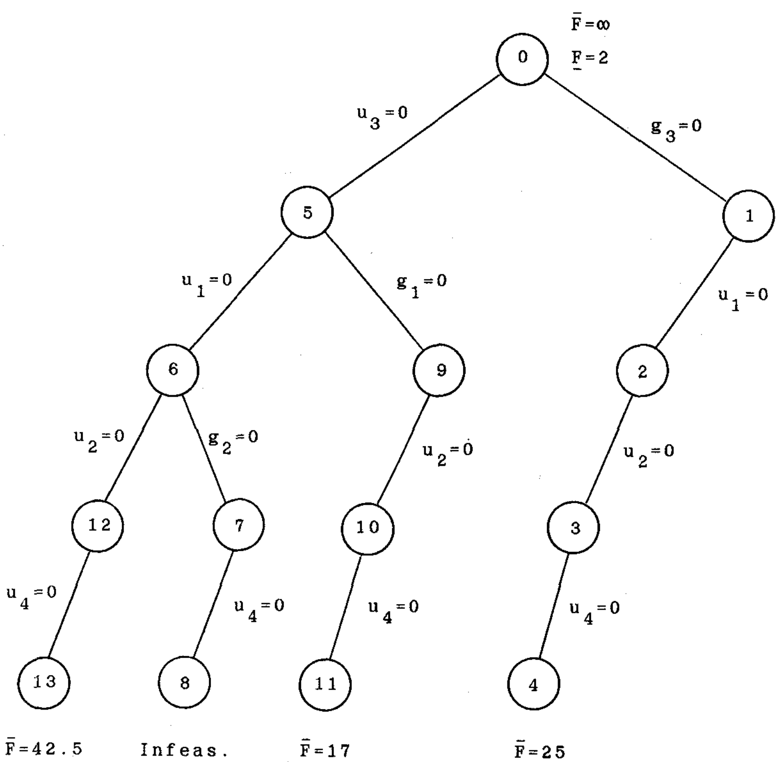

Fig. 2. Branch and bound tree for Example 1.

in the branch and bound tree as shown in Figure 2 terminates at node 4, and may be represented by the vector $\Pi_1 = (3, -1, -2, -4)$. At Step 4 we select $i_1 = 2$ and $i_2 = 3$ and return to Step 3 with $k = 2$ and $\Pi_2 = (-\underline{3}, -1, \underline{2}, -4)$. This path corresponds to node 8. An attempt to solve this problem fails due to infeasibility so the fathoming rule is applied. After eliminating all nodes on paths that contain $g_3 = 0$ and $g_2 = 0$ we backtrack to J_1^- which is now empty. Consequently, we branch to an unfathomed node at Step 6. The choices are those which are along the paths containing $g_1 = 0$ or $g_4 = 0$. Arbitrarily selecting the first and then solving the relaxed problem at Step 7 we arrive at the point $(1, 0)$ with $F = 17$. Setting $k = 3$ and returning to Step 2 we see that this point is in IR. Step 3 gives $\bar{F} = 17$ at node 11 in the tree with $\Pi_3 = (-\underline{3}, \underline{1}, -\underline{2}, -4)$; Step 4 indicates that this is a local optimum, and proceeding, all paths containing $g_1 = 0$ and $g_4 = 0$ are fathomed. Skipping to Step 8, the only subproblem remaining is associated with $u_i = 0$, $\forall i$. The solution occurs at $(\frac{16}{9}, \frac{7}{3})$ in Figure 1 with $F = 42.5$ and $\Pi_4 = (-\underline{3}, -\underline{1}, -\underline{2}, -\underline{4})$ so the corresponding node (13) is fathomed and the algorithm terminates. In all, 4 out of a possible 16 subproblems had to be solved, and 6 out of a possible 31 nodes had to be explored.

4. Computational experience

Before presenting the results of an analysis in which the above algorithm was compared with two other approaches, two examples will be given to highlight its performance. The first was prepared by Aiyoshi and Shimizu (1981) and the second by Bard (1984).

Example 2

$$\max F(y(x)) = (200 - y_{11} - y_{21})(y_{11} + y_{21}) + (160 - y_{12} - y_{22})(y_{12} + y_{22})$$

$$\begin{aligned}
\text{subject to}\quad & x_{11} + x_{12} + x_{21} + x_{22} \leqslant 40,\\
& 0 \leqslant x_{11} \leqslant 10,\ 0 \leqslant x_{12} \leqslant 5,\\
& 0 \leqslant x_{21} \leqslant 15,\ 0 \leqslant x_{22} \leqslant 20,\\
& \min f_1(y_1) = (y_{11} - 4)^2 + (y_{12} - 13)^2\\
& \text{subject to}\quad 0.4y_{11} + 0.7y_{12} \leqslant x_{11},\\
& \qquad\qquad\quad 0.6y_{12} + 0.3y_{22} \leqslant x_{12},\\
& \qquad\qquad\quad 0 \leqslant y_{11}, y_{12} \leqslant 20,\\
& \min_{y_2} f_2(y_2) = (y_{21} - 35)^2 + (y_{22} - 2)^2\\
& \text{subject to}\quad 0.4y_{21} + 0.7y_{22} \leqslant x_{21},\\
& \qquad\qquad\quad 0.6y_{21} + 0.3y_{22} \leqslant x_{22},\\
& \qquad\qquad\quad 0 \leqslant y_{21}, y_{22} \leqslant 40.
\end{aligned}$$

A solution to the relaxed problem at Step 1 yields an upper (rather than lower because we are maximizing) bound $\bar{F}=6600$ at $x^0=(8.24, 4.86, 10.76, 16.14)$, $y_1^0=(3.1, 10)$ and $y_2^0=(26.9, 0)$. Because complementary slackness is not satisfied at this point, we must solve the subproblem at Step 3 to find a point in IR. After advancing at Step 4, we arrive at $x^1=(7.91, 4.37, 11.09, 16.63)$, $y_1^1=(2.29, 10)$ and $y_2^1=(27.21, 0)$ with an objective function value equal to the upper bound; that is, $F(x^1, y^1)=6600$, so we terminate with $f_1=11.92$ and $f_2=55$. It is interesting to note that $x=(7, 3, 12, 18)$, $y_1=(0, 10)$ and $y_2=(30, 0)$ is also optimal, providing the followers with payoffs of $f_1=25$ and $f_2=29$. Corollary 3 implies that any convex combination of these points is likewise optimal so some ambiguity still exists in the model; however, for any of these points chosen by the leader, the followers' solutions seem to lie on the efficient frontier. In general, though, this will not be the case.

Example 3

$$\min_x F=-x_1^2-3x_2-4y_1+y_2^2$$

$$\begin{aligned}
\text{subject to}\quad & x_1^2+2x_2\leqslant 4,\\
& x_1\geqslant 0,\ x_2\geqslant 0,\\
& \min_y 2x_1^2+y_1^2-5y_2\\
& \text{subject to}\quad x_1^2-2x_1+x_2^2-2y_1+y_2\geqslant -3,\\
& \qquad\qquad\quad x_2+3y_1-4y_2\geqslant 4,\\
& \qquad\qquad\quad y_1\geqslant 0,\ y_2\geqslant 0.
\end{aligned}$$

Because the first constraint in this example is nonlinear, the Algorithm has to be modified to ensure that no local optima are overlooked. In particular, it is no longer possible to fathom all nodes containing binding constraints at a local solution unless they are all linear. Therefore, rather than try to adhere to the inducible region at all costs, the more traditional strategy of backtracking and solving a relaxed problem is adopted once a local minimum is found.

To begin, the computations at Step 1 yield $x^0=(0, 2)$ and $y^0=(4, 1)$ with a lower bound $\underline{F}=-21$. Executing Steps 2 and 3 gives an upper bound $\bar{F}=-14.13$ at $x^1=(1.45, 0.95)$ and $y^1=(1.88, 0.64)$ along the path $\Pi_1=(2, -1, -3, -4)$, with $J_1^-=\{1\}$. Because there are only 3 binding constraints at this point and $n_1+n_2=2+2=4$, we solve the problem associated with the path $\Pi_2=(2, \underline{1}, -3, -4)$ to arrive at a new upper bound $\bar{F}=-14.36$, giving $J_2^-=\emptyset$. Ordinarily, all nodes on paths containing $g_1=0$ and $g_2=0$ would now be fathomed, but this would be incorrect as mentioned above. Instead, because it seems that $g_2=0$ produces a good upper bound, we branch to the top of the tree and solve the relaxed problem where $u_2=0$ on path $\Pi_3=(-\underline{2})$. The resultant formulation is infeasible so all successor nodes are fathomed. An attempt to branch to a different part of the inducible region at Step

6 with $g_2 = 0$ and either $g_3 = 0$ or $g_4 = 0$ fails and the algorithm terminates with what appears to be the global optimum. However, because a number of the subproblems encountered at Step 3 were nonconvex, global optimality is not guaranteed (although in this case it was attained).

Comparative Results — In an attempt to more precisely gauge the efficiency of the Algorithm with respect to the underlying branch and bound tree, as well as other numerical procedures, a range of sample problems were generated and solved. Specifically, the Algorithm was compared with the barrier method of Aiyoshi and Shimizu, and a version of GRG2 (Lasdon et al., 1978) applied directly to problem (4). The results are presented in Table 1. In all cases, nonnegativity of the variables was assumed, but not reflected in the size of m which only counts the number of constraints of the form $g(x, y) \geqslant 0$.

Table 1
Comparative results

Procedure	Problem size (n_1, m_1, n_2, m)[a]	Average number of iterations	Range of iterations	Percent of problems solved[b]
Branch and bound algorithm	(5, 5, 5, 5)	24	2–84	100%
	(10, 12, 10, 15)	77	21–174	100%
	(15, 20, 15, 20)	209	62–743	100%
Barrier method	(5, 5, 5, 5)	53	37–157	60%
	(10, 12, 10, 15)	156	122–412	50%
	(15, 20, 15, 20)[c]	—	—	—
GRG2	(5, 5, 5, 5)	31	11–97	50%
	(10, 12, 10, 15)	99	47–241	30%
	(15, 20, 15, 20)	238	108–871	30%

[a]Nonnegativity of the variables assumed in all cases; m_1 refers to the number of constraints in X, and m to the number of constraints of the form $g(x, y) \geqslant 0$.
[b]Global optimum found.
[c]No apparent convergence for most problems after 10 minutes of CPU time.

Each problem set was characterized by the number of x and y variables, and the number of independent and joint contraints. Ten cases were run for each set with performance being measured by the 'average number of iterations', the 'range of iterations,' and whether or not the global optimum was found. Computation times are not reported because the work was done on different machines using codes embodying disparate levels of sophistication.

An iteration for the branch and bound algorithm is defined as the solution to one subproblem at a node in the tree[1]. Considering the first problem set where $m + n_2 = 10$,

[1] At this step, the actual computations were performed with a reduced gradient algorithm, itself requiring a number of iterations. This number, though, was relatively small since the previous solution provided a good starting point.

this implies a total of $2^{11}-1$ nodes. For the ten problems investigated 24 iterations on average were required, indicating that only about 1% of the tree had to be searched. The results were not nearly as positive for the barrier method where an iteration is defined as a solution to problem (6) for a particular value of r. Here β was taken to be logarithmic and GRG2 was used for the computations. The code was terminated when the objective function failed to improve by more than 2 percent in five consecutive iterations, or when the equivalent of 10 minutes of CPU time on an IBM 3081-D had elapsed. Notice that none of the problems in the third set met the convergence criterion within the allotted time, and for the first two sets global optimality proved elusive. For GRG2 an iteration is defined as the calculation of the reduced gradient and the solution to the corresponding line search problem. Although performance was relatively good, the global optimum was rarely uncovered, and in some cases, the code stopped with only a point in IR.

5. Discussion and conclusions

The branch and bound algorithm presented above is designed to efficiently solve the BLPP when the leader's objective function and constraint set are convex, and the followers' problems are quadratic. By exploiting the properties of the inducible region an active set strategy permits us to quickly develop a tight upper bound, and thus markedly reduce the number of subproblems that must be set up and solved. The numerical results tend to bear this out; in all cases, only a small fraction of the nodes in the underlying branch and bound tree had to be examined.

By way of comparison two other procedures were investigated but did not fare quite so well, primarily due to the sharp nonlinearities associated with the BLPP. In approximately 50 percent of the test cases they were unable to find the true solution, and in all cases, global optimality could never be independently confirmed.

As suggested by Example 3, the Algorithm can readily be modified to accommodate more general functional forms, but not without suffering some decline in efficiency. Further, if the rational reaction sets, $M_p(x)$, are single-valued, then the Algorithm will also solve the linear BLPP with a minor modification to deal with the potential for multiple optima in (4). With regard to the nonlinear case, if the followers' problems are convex, it may be possible to replace (1c)—(1e) with their equivalent Lagrangian duals, write the first order stationarity conditions of each to get (4c), and then devise an iterative scheme that will converge at least locally.

References

E. Aiyoshi and K. Shimizu, "Hierarchical decentralized systems and its new solution by a barrier method," *IEEE Transactions on Systems, Man, and Cybernetics* SMC-11 (1981) 444-449.

J.F. Bard, "An algorithm for solving the general bilevel programming problem," *Mathematics of Operations Research* 8(2) (1983a) 260-272.

J.F. Bard, "Coordination of a multidivisional organization through two levels of management," *Omega* 11 (1983b) 457-468.

J.F. Bard, "Optimality conditions for the bilevel programming problem," *Naval Research Logistics Quarterly* 31 (1984) 13-26.

J.F. Bard and J.E. Falk, "An explicit solution to the multi-level programming problem," *Computers and Operations Research* 9(1) (1982) 77-100.

T. Basar and H. Selbuz, "Closed loop Stackelberg strategies with applications in optimal control of multilevel systems," *IEEE Transactions on Automatic Control* AC-24 (1979) 166-178.

W.F. Bialas and M.H. Karwan, "Two-level linear programming," *Management Science* 30 (1984) 1004-1020.

J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *Journal of the Operational Research Society* 32 (1981) 783-792

P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization* (Academic Press, London 1981).

W.W. Hogan, "Point-to-set maps in mathematical programming," *SIAM Review* 15 (1973) 591-603.

L.S. Lasdon, A.D. Warren, A. Jain and M. Ratner, "Design and testing of a generalized reduced gradient code for nonlinear programming," *ACM Transactions on Mathematical Software* 4 (1978) 34-50.

M. Simaan and J.B. Cruz, Jr., "On the Stackelberg strategy in nonzero-sum games," *Journal of Optimization Theory and Applications* 11 (1973) 533-555.

B. Tolwinski, "Closed-loop Stackelberg solution to a multistage linear-quadratic game," *Journal of Optimization Theory and Applications* 34 (1981) 485-501.