MATHEMATICS OF OPERATIONS RESEARCH
Vol. 8, No. 2, May 1983
Printed in U.S.A.

JOINTLY CONSTRAINED BICONVEX PROGRAMMING*

FAIZ A. AL-KHAYYAL†‡ AND JAMES E. FALK§#

This paper presents a branch-and-bound algorithm for minimizing the sum of a convex function in x, a convex function in y and a bilinear term in x and y over a closed set. Such an objective function is called biconvex with biconcave functions similarly defined. The feasible region of this model permits joint constraints in x and y to be expressed. The bilinear programming problem becomes a special case of the problem addressed in this paper. We prove that the minimum of a biconcave function over a nonempty compact set occurs at a boundary point of the set and not necessarily an extreme point. The algorithm is proven to converge to a global solution of the nonconvex program. We discuss extensions of the general model and computational experience in solving jointly constrained bilinear programs, for which the algorithm has been implemented.

Introduction. One of the most persistently difficult and recurring nonconvex problems in mathematical programming is the bilinear program, whose general form is

$$\begin{aligned} &\underset{(x,y)}{\text{minimize}} && c^T x + x^T A y + d^T y \\ &\text{subject to} && x \in X, y \in Y \end{aligned} \tag{1}$$

where c and d are given vectors, A is a given $p \times q$ matrix, and X and Y are given polyhedra in $\mathscr{R}^p$ and $\mathscr{R}^q$, respectively.

Mills [15], Mangasarian [12], Mangasarian and Stone [13], and Altman [1] studied the problem as formulated in the bimatrix game context. Solution procedures were either locally convergent (e.g., [1], [2]) or completely enumerative (e.g., [13]).

Cabot and Francis [2] proposed an extreme point ranking procedure for the solution. Konno, in a series of papers [9]–[11], develops a cutting plane approach designed to converge locally, and in a finite number of steps to an ϵ-optimal solution.

By taking the partial dual of (1) with respect to y, the problem becomes a min-max problem wherein the constraint region available to the "inside optimizer" (i.e., the maximizer) is determined by the selection of the "outside optimizer's" (i.e., the minimizer over X) move. Falk [4] addressed this formulation and proposed a branch-and-bound solution procedure. Vaish and Shetty developed two finite procedures: one involving the extension of Tui's method of "polyhedral annexation" [20] and the other being a cutting plane procedure employing polar cuts [21]. Subsequently, Sherali and Shetty [18] improved the latter procedure by incorporating disjunctive face cuts.

Finally, Gallo and Ülkücü [7] addressed the min-max formulation by developing a cutting plane approach in connection with Tui's method.

Applications of the bilinear programming problem include constrained bimatrix games, dynamic Markovian assignment problems, multicommodity network flow

* Received July 6, 1981; revised February 8, 1982.
AMS 1980 subject classification. Primary 90C30.
OR/MS Index 1978 subject classification. Primary: 642 Programming/nonlinear/algorithms.
Key words. Bilinear programming, nonconvex programming, global convergence, branch and bound algorithm, biconvex functions.
† Georgia Institute of Technology.
‡ Research supported, in part, by the Office of Naval Research Contract No. N00014-75-C-0729.
§ The George Washington University.
# Research supported, in part, by the Army Research Office Contract No. DAAG-29-79-C-0062, and the Office of Naval Research Contracts No. N00014-75-0611 and N00014-75-C-0729.

0364-765X/83/0802/0273$01.25

problems and certain dynamic production problems. Konno [10] discusses a host of applied problems which can be formulated as bilinear programs.

In this paper, we describe a new approach to the problem which is guaranteed to converge to a global solution. The method is capable of solving a wider class of problems than addressed above. We prove convergence and discuss extensions of the general model. Finally, some computational experience is presented.

The problem. The specific problem addressed herein has the form

$$\min_{(x,\,y)} \phi(x, y) = f(x) + x^T y + g(y)$$
$$\text{subject to} \quad (x, y) \in S \cap \Omega,$$

where:

(a) f and g are convex over $S \cap \Omega$;

(b) S is a closed, convex set; and

(c) $\Omega = \{(x, y) : l \leqslant x \leqslant L, m \leqslant y \leqslant M\}$

We shall refer to this as Problem $\mathscr{P}$.

Immediate extensions of the problem are obvious. For example, a term of the form $x^T A y$ can be transformed into $x^T z$ if the (linear) constraint $z = Ay$ is included among those defining S, and if the bounds $n_j = \min\{(Ay)_j : m \leqslant y \leqslant M\}$ and $N_j = \max\{(Ay)_j : m \leqslant y \leqslant M\}$ replace the bounds on y in defining Ω. Also, a sum such as $\sum c_i x_i y_i$ is easily transformed to the form required in Problem $\mathscr{P}$. Note that this problem is not convex and, even though each term $x_i y_i$ of $x^T y$ is quasiconvex, proper local solutions are possible. For example, the simple problem $\min\{xy : -1 \leqslant x \leqslant 2, -2 \leqslant y \leqslant 3\}$ has local solutions at $(-1, 3)$ and $(2, -2)$.

The set S allows for joint constraints on (x, y), and hence we are addressing a problem of more generality than the traditional bilinear program, even when we restrict f and g above to be linear functions. The traditional bilinear program (having form (1)) will possess a solution at an extreme point (see, e.g., Falk [4]), but this property is lost in the jointly constrained case. For example, the problem

$$\begin{aligned} \underset{(x,\,y)}{\text{minimize}} \quad & -x + xy - y \\ \text{subject to} \quad & -6x + 8y \leqslant 3, \\ & 3x - y \leqslant 3, \\ & 0 \leqslant x, y \leqslant 5 \end{aligned}$$

has a solution $(7/16, 1/2)$ at a point which is not among the extreme points of its feasible region, and the extreme points of this problem are not solutions.

While a jointly constrained bilinear program may possess nonvertex solutions, we can show that such problems must have boundary solutions. We shall prove this in a somewhat more general context, by assuming the objective function is "jointly concave" and the feasible region is compact.

THEOREM 1. *Let $\phi(x, y)$ be a continuous function defined over the nonempty compact set S, and assume that the functions $\phi(\cdot, y)$ and $\phi(x, \cdot)$ are both concave functions. Then the problem $\min\{\phi(x, y) : (x, y) \in S\}$ has a solution on ∂S, the boundary of S.*

PROOF. Assume that this problem has a solution $(\bar{x}, \bar{y}) \in \operatorname{int} S$. Let (x^*, y^*) denote a point on ∂S which minimizes the Euclidean distance between ∂S and $(\bar{x}, \bar{y})$, and let d^* denote this minimum distance.

Let $(s, t) = 2(\bar{x}, \bar{y}) - (x^*, y^*)$, so that $(\bar{x}, \bar{y})$ is the midpoint of the line segment joining (s, t) and (x^*, y^*). Note that (s, t) is feasible.

We have

$$\phi(\bar{x}, \bar{y}) = \phi\left(\tfrac{1}{2}(x^*, y^*) + \tfrac{1}{2}(s,t)\right)$$
$$\geqslant \tfrac{1}{2}\phi(x^*, \tfrac{1}{2}y^* + \tfrac{1}{2}t) + \tfrac{1}{2}\phi(s, \tfrac{1}{2}y^* + \tfrac{1}{2}t)$$
$$\geqslant \tfrac{1}{4}\left[\phi(x^*, y^*) + \phi(x^*, t) + \phi(s, y^*) + \phi(s,t)\right].$$

But $(\bar{x}, \bar{y})$ is a solution so that

$$\phi(\bar{x}, \bar{y}) \leqslant \min\{\phi(x^*, y^*), \phi(x^*, t), \phi(s, y^*), \phi(s,t)\}$$

which implies

$$\phi(\bar{x}, \bar{y}) \leqslant \tfrac{1}{4}\left[\phi(x^*, y^*) + \phi(x^*, t) + \phi(s, y^*) + \phi(s,t)\right].$$

Hence

$$\phi(\bar{x}, \bar{y}) = \phi(x^*, y^*)$$

and $(x^*, y^*) \in \partial S$ must be a solution of the problem.

The algorithm. Since Problem $\mathscr{P}$ is nonconvex, any algorithm designed to solve it must take into account the behavior of ϕ over the entire feasible region $S \cap \Omega$. We shall develop a branch-and-bound scheme to meet this requirement. The method is patterned after the general scheme of Falk and Soland [5] although the results of this reference cannot directly apply here, as we are dealing with a nonseparable objective function.

We shall employ convex envelopes to obtain the bounds required. The *convex envelope of a function f over* Ω is the pointwise supremum of all convex functions which underestimate f over Ω (see [3]), and will be denoted here by $\mathrm{Vex}_{\Omega} f(x)$.

Since

$$\mathrm{Vex}_{\Omega}\left(\sum_{i=1}^{n} x_i y_i\right) = \sum_{i=1}^{n} \mathrm{Vex}_{\Omega_i} x_i y_i$$

where $\Omega = \{(x, y) : l \leqslant x \leqslant L, m \leqslant y \leqslant M\}$, and $\Omega_i = \{(x_i, y_i) : l_i \leqslant x_i \leqslant L_i, m_i \leqslant y_i \leqslant M_i\}$, it suffices to compute $\mathrm{Vex}_{\Omega_i} x_i y_i$. In order to simplify the notation, we will temporarily drop the subscripts on the quantities $x_i y_i$, l_i, L_i, m_i and M_i, and compute $\mathrm{Vex}_{\Omega} xy$ where x and y are now *real* variables, as opposed to vectors.

THEOREM 2. *Let* $\Omega = \{l \leqslant x \leqslant L, m \leqslant y \leqslant M\} \subset \mathscr{R}^2$.
Then

$$\mathrm{Vex}_{\Omega} xy = \max\{mx + ly - lm, Mx + Ly - LM\}.$$

PROOF. The convex envelope of a function f over Ω may be equivalently defined as the pointwise supremum of all linear functions which underestimate f over Ω. Since $x - l \geqslant 0$ and $y - m \geqslant 0$, it follows that

$$xy \geqslant mx + ly - lm$$

so that $l_1(x, y) = mx + ly - lm$ underestimates xy over Ω. Likewise $l_2(x, y) = Mx + Ly - LM$ underestimates xy over Ω. Hence

$$\theta(x, y) = \max\{l_1(x, y), l_2(x, y)\}$$

underestimates xy over Ω. Note that θ is convex.

Also, simple computation shows that $\theta(x, y)$ agrees with xy at the four extreme points of the rectangle Ω, i.e., $\max\{l_1(x, y), l_2(x, y)\} = xy$ at the corners of Ω.

If θ were not the convex envelope of xy over Ω, there would be a third linear function $l_3(x, y)$ such that $\max\{l_1(\tilde{x}, \tilde{y}), l_2(\tilde{x}, \tilde{y})\} < l_3(\tilde{x}, \tilde{y}) \leqslant \tilde{x}\tilde{y}$ for some $(\tilde{x}, \tilde{y}) \in \Omega$.

But l_3 must also underestimate xy at the corners of Ω, so $l_3(x^c, y^c) \leqslant x^c y^c$ for each corner (x^c, y^c) of Ω. This implies that $l_3(\tilde{x}, \tilde{y}) \leqslant \max\{l_1(\tilde{x}, \tilde{y}), l_2(\tilde{x}, \tilde{y})\}$ which is a contradiction.

The following related result, proven easily by direct computation, is useful in the algorithm to be developed.

THEOREM 3. *Under the assumptions of the previous theorem*

$$\text{Vex}_{\Omega} xy = xy$$

for all $(x, y) \in \partial\Omega$.

The above two results directly imply the next corollary.

COROLLARY. *If* $x, y \in \mathscr{R}^n$

$$\Omega = \{(x, y) : l \leqslant x \leqslant L, m \leqslant y \leqslant M\}, \quad \textit{and}$$

$$\Omega_i = \{(x_i, y_i) : l_i \leqslant x_i \leqslant L_i, m_i \leqslant y_i \leqslant M_i\},$$

then

$$\text{Vex}_{\Omega} x^T y = \sum_{i=1}^{n} \text{Vex}_{\Omega_i} x_i y_i$$

and $x^T y = \text{Vex}_{\Omega} x^T y$ *for all* $(x, y) \in \partial\Omega$.

Let

$$\phi(x, y) = f(x) + x^T y + g(y)$$

where f and g are convex functions over $\{x : l \leqslant x \leqslant L\}$ and $\{y : m \leqslant y \leqslant M\}$, respectively. Let

$$\psi^1(x, y) = f(x) + \text{Vex}_{\Omega} x^T y + g(y).$$

Note that ψ^1 is a convex underestimator of ϕ, and furthermore, agrees with ϕ on $\partial\Omega$.

We are now in a position to describe the algorithm. Based on the branch-and-bound philosophy, the method proceeds as a sequence of *stages* $1, 2, 3, \ldots$. At stage k, there is a continuous function $\psi^k(x, y)$, which underestimates the original objective function ψ over Ω. A point (x^k, y^k) is determined as the (global) solution of the problem

$$\begin{array}{ll} \text{minimize} & \psi^k(x, y) \\ \text{subject to} & (x, y) \in S \cap \Omega \end{array} \qquad \text{(Problem } \mathscr{P}^k\text{)}$$

so that the optimal value $v^k = \psi^k(x^k, y^k)$ serves as a lower bound on the desired value v^* of problem $\mathscr{P}$.

The point (x^k, y^k) is feasible to problem $\mathscr{P}$, so that the value $V^k = \phi(x^k, y^k)$ serves as an upper bound on v^*.

At stage k, the hyperrectangle Ω will be partitioned into a set of hyperrectangles

$$\Omega^{kj} = \{(x, y) : l^{kj} \leqslant x \leqslant L^{kj}, m^{kj} \leqslant y \leqslant M^{kj}\}.$$

The (convex) functions

$$\psi^{kj}(x, y) = f(x) + \text{Vex}_{\Omega^{kj}} x^T y + g(y)$$

underestimate ϕ over Ω^{kj}, and make up the stage function ψ^k according to the rule

$$\psi^k(x, y) = \psi^{kj}(x, y) \qquad \text{if} \quad (x, y) \in \Omega^{kj}.$$

Because each of the functions ψ^{kj} agrees with ϕ along the boundary of Ω^{kj}, the above function is well defined for all $(x, y) \in \Omega$, including those points which may lie on the boundaries of two or more of the sets Ω^{kj}.

Associated with each ψ^{kj} and Ω^{kj} at stage k, there is a *convex* program

$$\begin{aligned} &\text{minimize} \quad \psi^{kj}(x, y) \\ &\text{subject to} \quad (x, y) \in S \cap \Omega^{kj} \end{aligned} \qquad \text{(Problem } \mathscr{P}^{kj}\text{)}$$

and (x^{kj}, y^{kj}) will denote a solution of this problem, with v^{kj} denoting the optimal value.

The stage value v^k is related to the value v^{kj} by

$$v^k = \min_j \{ v^{kj} \}$$

and (x^k, y^k) is any one of the points (x^{kj}, y^{kj}) which yields the value v^k.

Stage 1 consists of problem $\mathscr{P}^1$, with $\psi^1(x, y)$ defined above. Solving the convex program $\mathscr{P}^1$ yields

$$v^1 \leqslant v^* \leqslant V^1.$$

Obviously, if equality prevails throughout, (x^1, y^1) solves $\mathscr{P}$ and we are done.

If $v^1 < V^1$, then we must have

$$\operatorname{Vex}_\Omega (x^1)^T y^1 < (x^1)^T y^1,$$

i.e.,

$$\max\{ m_i x_i^1 + l_i y_i^1 - l_i m_i, M_i x_i^1 + L_i y_i^1 - L_i M_i \} < x_i^1 y_i^1$$

for some i. We choose an I which produces the largest difference between the two sides of the above inequality, and split the Ith rectangle into four subrectangles according to the rule illustrated in Figure 1.

The result of this splitting serves to set up four new subproblems, $\mathscr{P}^{21}$, $\mathscr{P}^{22}$, $\mathscr{P}^{23}$ and $\mathscr{P}^{24}$ for Stage 2. These problems may best be described in terms of the regions Ω^{2j} which define them:

$$\Omega^{2j} = \{ (x, y) : l^{2j} \leqslant x \leqslant L^{2j}, m^{2j} \leqslant y \leqslant M^{2j} \}$$

where

$$(l_i^{2j}, L_i^{2j}, m_i^{2j}, M_i^{2j}) = (l_i, L_i, m_i, M_i) \qquad \text{if} \quad i \neq I,$$

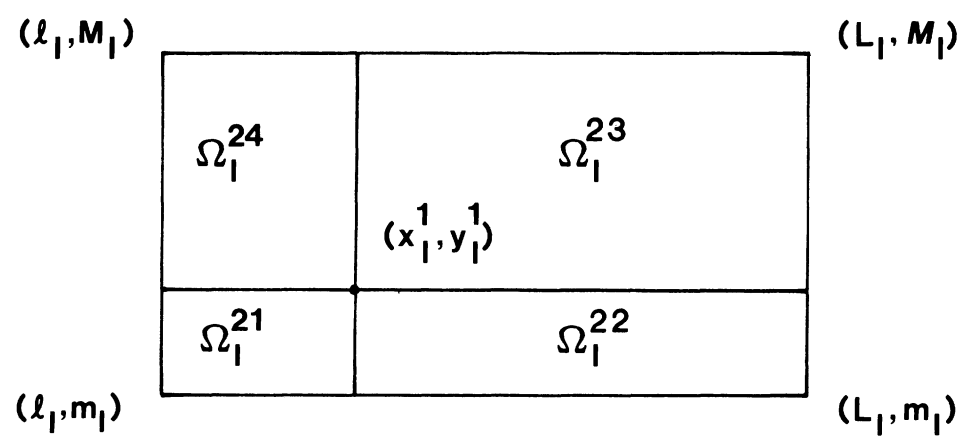

FIGURE 1. Splitting Ω_I at stage 1.

and

$$(l_I^{21}, L_I^{21}, m_I^{21}, M_I^{21}) = (l_I, x_I^1, m_I, y_I^1),$$

$$(l_I^{22}, L_I^{22}, m_I^{22}, M_I^{22}) = (x_I^1, L_I, m_I, y_I^1),$$

$$(l_I^{23}, L_I^{23}, m_I^{23}, M_I^{23}) = (x_I^1, L_I, y_I^1, M_I),$$

$$(l_I^{24}, L_I^{24}, m_I^{24}, M_I^{24}) = (l_I, x_I^1, y_I^1, M_I).$$

That is, all $\Omega_i^{2j} = \Omega_i$ for $i \neq I$, and Ω_I^{2j} are defined as in Figure 1, numbering counterclockwise starting with the lower left-hand subrectangle of Ω_I.

Note that each of the problems $\mathscr{P}^{2j}$ is feasible since $(x^1, y^1) \in S \cap \Omega^{2j}$ for each j. Moreover, by the construction of ψ^{2j}, we have $\psi^1(x^1, y^1) < \psi^{2j}(x^1, y^1)$ and hence

$$\psi^1(x, y) \leqslant \psi^2(x, y) \leqslant \phi(x, y).$$

Solving each of the four convex problems $\mathscr{P}^{2j}$ in turn yields a point (x^2, y^2) with lower bound v^2 and upper bound V^2 for Stage 2. To be consistent with future notation, we shall define the *best lower bound* $v_b^2 = v^2$ and the *best upper bound* $V_b^2 = \min\{V^1, V^2\}$. Clearly,

$$v_b^1 \leqslant v_b^2 \leqslant v^* \leqslant V_b^2 \leqslant V_b^1.$$

If $v_b^2 = V_b^2$, we are done, with the solution (x^*, y^*) of problem $\mathscr{P}$ being that (x^{2j}, y^{2j}) which yields this common value.

At this point we have shown how to define Stage 1, and, from that, set up Stage 2. In general, we must show how to proceed from Stage k to Stage $(k + 1)$.

It is convenient, at this point, to refer to a *Branch-and-Bound Tree*. Figure 2 illustrates a typical case, wherein the nodes of the tree correspond to subproblems $\mathscr{P}^{kj}$.

In any implementation of the method, information describing the problems $\mathscr{P}^{kj}$ must be stored. As in any branch-and-bound procedure, the number of problems stored can become excessive. Note, however, that any subproblems $\mathscr{P}^{kj}$ from which branching has taken place need not be stored, as the subproblems created from $\mathscr{P}^{kj}$ more accurately estimate the original problem $\mathscr{P}$ than does $\mathscr{P}^{kj}$. Nodes from which no branching has yet taken place at stage k are termed *open*. There are ten open nodes in Figure 2 in Stage 4.

Note, also, that any open node $\mathscr{P}^{lj}$ for which $v^{lj} \geqslant V_b^k$ at Stage k can be dismissed from further consideration (erased from storage), since the lower bound associated

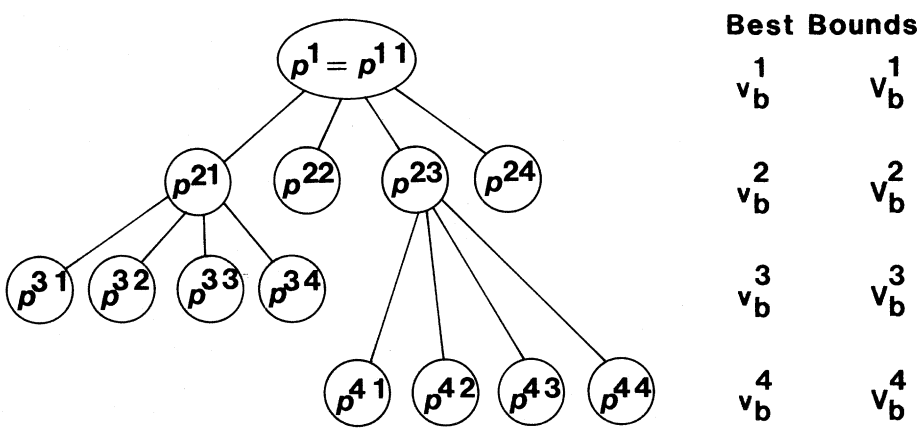

FIGURE 2. A branch-and-bound tree at stage 4.

with such a node promises no improvement over the feasible point which yielded the value V_b^k.

The functions ψ^{lj}, which correspond to open nodes at Stage k, collectively make up the stage function ψ^k.

The best lower bound at Stage k is defined by

$$v_b^k = \min\{v^{lj} : \text{node } (l, j) \text{ is open at Stage } k\}.$$

The best upper bound at Stage k is defined by

$$V_b^k = \min\{V^l : l = 1, \ldots, k\}$$

and may be expressed recursively as

$$V_b^k = \min\{V_b^{k-1}, V^{k1}, V^{k2}, V^{k3}, V^{k4}\}.$$

Moving from Stage k to Stage $(k+1)$ involves the selection of an open node at Stage k, and the creation of four new nodes from that selected node.

Selection of a branching node is accomplished by means of the *Best Bound Rule*:

> To move from Stage k to Stage $k+1$, select an open node whose lower bound v^{lj} equals v_b^k.

The creation of the four new nodes from the selected node proceeds in the same manner as $\mathscr{P}^{21}, \ldots, \mathscr{P}^{24}$ were created from $\mathscr{P}^{11}$.

It follows, immediately from the above definitions, that

$$v_b^1 \leqslant v_b^2 \leqslant \cdots \leqslant v^* \leqslant \cdots \leqslant V_b^2 \leqslant V_b^1,$$

i.e., the best lower bounds increase towards v^*, while the best upper bounds decrease towards v^*. It remains to show that the procedure converges to a global solution.

Convergence proof. In this section only, we will drop the functions f and g from the definitions of ϕ and ψ^k, i.e., we set

$$\phi(x, y) = x^T y = \sum_{i=1}^{n} x_i y_i.$$

There is clearly no loss of generality in this simplification, since both ψ^k and ϕ included the common terms f and g.

We have by construction

$$\psi^k(x, y) \leqslant \phi(x, y) \qquad \text{for all} \quad (x, y) \in \Omega$$

where ψ^k is piecewise convex and continuous over Ω.

Moreover, $\{\psi^k\}$ is an increasing sequence of functions, bounded above by the continuous function ϕ. Thus $\{\psi^k\}$ converges pointwise to some function ψ:

$$\lim_{k\to\infty} \psi^k(x, y) \doteq \psi(x, y) \leqslant \phi(x, y).$$

Now let $\{(x^{k_t}, y^{k_t})\}$ converge to $(\bar{x}, \bar{y})$ where $t = 1, 2, \ldots$. We may assume $k_1 = 1$. Each (x^{k_t}, y^{k_t}) is a member of some set $\Omega^{k_t j_t}$ which we will label $\Omega^{k(t)}$. Now at least one of the sets $\Omega^{k_t+1, j}$ (j indexing the active nodes at stage $k_t + 1$)[1] contains an infinite number of members of $\{(x^{k_t}, y^{k_t})\}$. Suppose we choose one of these, and let k_{t+1} be

[1] To simplify the notation in the proof, it is convenient at this point to let j in the ordered superscript pair (k, j) be an index for the open nodes at stage k. For example, at stage $k = 4$ in Figure 2, all the open nodes can be arbitrarily relabelled P^{4j} with j varying from 1 to 10. There is no loss in generality by performing this relabelling at every stage.

the smallest integer larger than k_t such that $(x^{k_{t+1}}, y^{k_{t+1}})$ lies in the chosen set. In this way, we can assume that the sequence $\{(x^{k_t}, y^{k_t})\}_{t\in T}$ is associated with a *nested* sequence of sets $\{\Omega^{k(t)}\}_{t\in T}$, i.e.,

$$\Omega = \Omega^{k(1)} \supset \Omega^{k(2)} \supset \cdots .$$

Clearly,

$$(\bar{x}, \bar{y}) \in \bigcap_{t\in T} \Omega^{k(t)}$$

and

$$\bar{l}_1 \doteq \lim_{t\in T} l_i^{k(t)} \leqslant \bar{x}_i \leqslant \lim_{t\in T} L_i^{k(t)} \doteq \bar{L}_i,$$

$$\bar{m}_i \doteq \lim_{t\in T} m_i^{k(t)} \leqslant \bar{y}_i \leqslant \lim_{t\in T} M_i^{k(t)} \doteq \bar{M}_i$$

for each i.

Moreover, we must have:

$$\text{either } \bar{l}_I = \bar{x}_I \quad \text{or} \quad \bar{x}_I = \bar{L}_I$$

and

$$\text{either } \bar{m}_I = \bar{y}_I \quad \text{or} \quad \bar{y}_I = \bar{M}_I$$

for some I, or otherwise we could not have $\{(x^{k_t}, y^{k_t})\}$ converging to $(\bar{x}, \bar{y})$.

Now suppose

$$\bar{l}_i < \bar{x}_i < \bar{L}_i$$

and

$$\bar{m}_i < \bar{y}_i < \bar{M}_i \qquad \text{for some } i.$$

Then

$$l_i^{k(t)} < x_i^{k_t} < L_i^{k(t)}$$

and

$$m_i^{k(t)} < y_i^{k_t} < M_i^{k(t)}$$

for this i and t sufficiently large. But

$$x_i^{k_t} y_i^{k_t} - \operatorname{Vex}_{\Omega_i^{k(t)}} x_i^{k_t} y_i^{k_t} \leqslant x_{I(t)}^{k_t} y_{I(t)}^{k_t} - \operatorname{Vex}_{\Omega_{I(t)}^{k(t)}} x_{I(t)}^{k_t} y_{I(t)}^{k_t}$$

where $I(t)$ is the index on which branching takes place at stage k_t. In particular, $I(t)$ equals the above-mentioned I for infinitely many t. Hence, the right-hand side of this last inequality goes to 0 as $t \to \infty$, which yields the result

$$\psi(\bar{x}, \bar{y}) \doteq \lim_{t\in T} \psi^{k_t}(x^{k_t}, y^{k_t}) = \lim_{t\in T} \operatorname{Vex}_{\Omega^{k(t)}} x^{k_t} \cdot y^{k_t} \doteq \phi(\bar{x}, \bar{y}),$$

assuming this limit exists.

We will show that the limit exists by proving that the pointwise limit is continuous. This will be established by proving that the sequence $\{\psi^k\}$ is equicontinuous (cf. [17]). In a related paper by Horst [8], equicontinuity of a sequence of estimating functions is assumed to hold, with convergence thereby assured.

Recall that

$$\psi^k(x, y) = \operatorname{Vex}_{\Omega^{kj}} x^T y = \sum_{i=1}^{n} \operatorname{Vex}_{\Omega_i^{kj}} x_i y_i \qquad \text{if} \quad (x, y) \in \Omega^{kj}.$$

For notational convenience, let

$$H_i^{kj}(x_i, y_i) = \mathrm{Vex}_{\Omega_i^{kj}} x_i y_i,$$

$$h_{i1}^{kj}(x_i, y_i) = m_i^{kj} x_i + l_i^{kj} y_i - l_i^{kj} m_i^{kj},$$

and

$$h_{i2}^{kj}(x_i, y_i) = M_i^{kj} x_i + L_i^{kj} y_i - L_i^{kj} M_i^{kj},$$

so that

$$\psi^k(x, y) = \sum_{i=1}^{n} H_i^{kj}(x_i, y_i) = \sum_{i=1}^{n} \max\left\{ h_{i1}^{kj}(x_i, y_i), h_{i2}^{kj}(x_i, y_i) \right\}$$

when $(x, y) \in \Omega^{kj}$.

Let γ_i be the solution value and (α_i, β_i) the solution point of the maximum norm problem $\max\{\|\nabla x_i y_i\| : (x_i, y_i) \in \Omega_i\}$.

That is,

$$\gamma_i = \max\{\|\nabla x_i y_i\| : (x_i, y_i) \in \Omega_i\} = \|(\beta_i, \alpha_i)\|.$$

Note that (α_i, β_i) must be a corner point of the set Ω_i. Furthermore, for every j and all k,

$$\max\left\{ \|\nabla h_{i1}^{kj}\|, \|\nabla h_{i2}^{kj}\| \right\} \leqslant \gamma_i, \qquad i = 1, \ldots, n,$$

since the refining rule ensures that $l_i \leqslant l_i^{kj}$, $m_i \leqslant m_i^{kj}$, $L_i \geqslant L_i^{kj}$, and $M_i \geqslant M_i^{kj}$.

Consider the line $\mathscr{L}_i$ in Ω_i going through an arbitrary point $(x_i, y_i) \in \Omega_i$ in the direction (α_i, β_i). Construct the line in $\mathscr{R}^3$ having domain $\mathscr{L}_i$ and slope γ_i. The surface of revolution of this line about the vertical axis passing through the point $(x_i, y_i, x_i y_i)$ is a nonconvex cone. Truncate the cone, top and bottom, at Euclidean distances of $x_i y_i - \epsilon/n$ and $x_i y_i + \epsilon/n$ away from the x_i, y_i-plane. This yields the cone C of Figure 3.

The projection of C onto the x_i, y_i-plane is the set $\{(u_i, v_i) : \|(x_i, y_i) - (u_i, v_i)\| < \epsilon/n\gamma_i\}$. Let $\gamma = \max\gamma_i$ and $\delta = \epsilon/n\gamma$. Then we can write: given $\epsilon > 0$, for any $(x_i, y_i), (u_i, v_i) \in \Omega_i$,

$$\|(x_i, y_i) - (u_i, v_i)\| < \delta \Rightarrow |x_i y_i - u_i v_i| < \epsilon/n$$

for every $i = 1, \ldots, n$.

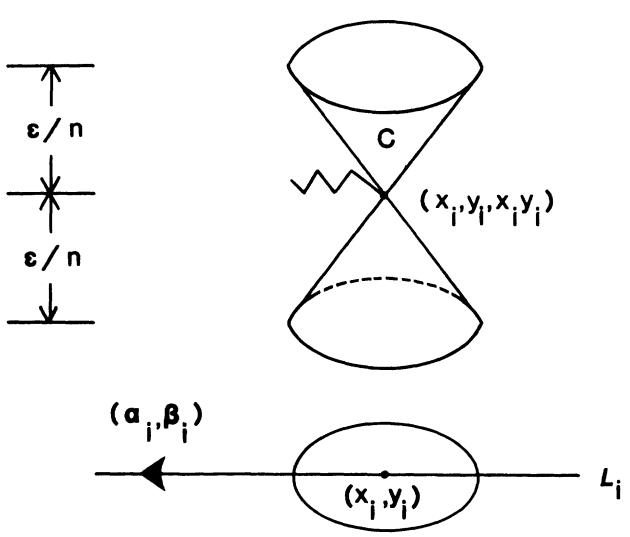

FIGURE 3. Cone generated by line in R^3 having domain $\mathscr{L}_i$ and slope γ_i.

If the cone C is moved along the vertical axis until its center rests on the point $(x_i, y_i, H_i^{kj}(x_i, y_i))$, then it can be shown that

$$|H_i^{kj}(x_i, y_i) - H_i^{kj}(u_i, v_i)| \leq |x_i y_i - u_i v_i|$$

for all i, j, k, and any (x_i, y_i) and (u_i, v_i) satisfying $\|(x_i, y_i) - (u_i, v_i)\| < \delta$. This follows from the fact that any (piecewise linear) line segment on the surface of H_i^{kj}, in any direction, cannot intersect the boundary of the cone C (see Figure 3). This is guaranteed by γ_i being the maximum slope of any *line* tangent to the graph of $x_i y_i$, and H_i^{kj} defining a piecewise affine underestimating function for $x_i y_i$. Notationally, this may be expressed as

$$\max_j \{ \|\nabla h_{i1}^{kj}\|, \|\nabla h_{i2}^{kj}\| \} \leq \gamma_i \leq \gamma$$

for every i and all k, where j indexes the active nodes at Stage k.

Hence, we can write: given $\epsilon > 0$, for any $(x, y), (u, v) \in \Omega$,

$$\begin{aligned} \|(x, y) - (u, v)\| < \delta &\Rightarrow \|(x_i, y_i) - (u_i, v_i)\| < \delta \\ &\Rightarrow |H_i^{kj}(x_i, y_i) - H_i^{kj}(u_i, v_i)| < \epsilon/n \\ &\Rightarrow |\psi^k(x, y) - \psi^k(u, v)| < \epsilon \end{aligned}$$

for all k. This is precisely the definition of equicontinuity on a set, which completes the proof of convergence.

Computational considerations. The proof of convergence only requires S to be closed. The convexity of S has been assumed only to ensure that each problem $\mathscr{P}^{kj}$ is a convex program, thus making global solutions computationally feasible. The objective function, however, is nondifferentiable. Although problem $\mathscr{P}^{kj}$ can be solved by nondifferentiable optimization algorithms, an alternative approach would be to solve the equivalent problem $\overline{\mathscr{P}}^{kj}$:

$$\begin{aligned} \underset{(x, y, z)}{\text{minimize}} \quad & f(x) + e^T z + g(y) \\ \text{subject to} \quad & h_{i1}^{kj}(x_i, y_i) \leq z_i, \qquad i = 1, \ldots, n, \\ & h_{i2}^{kj}(x_i, y_i) \leq z_i, \qquad i = 1, \ldots, n, \\ & (x, y) \in S \cap \Omega \end{aligned}$$

where $e^T = (1, 1, \ldots, 1)$ is the summation vector.

Suppose (x^k, y^k) is generated by problem $\mathscr{P}^{kq}$ at stage k. The set Ω^{kq} is then partitioned into four regions, each having a subproblem defined on it. It is clear that (x^k, y^k) is feasible to each of these four problems. However, if (x^k, y^k, z^k) denotes a solution of problem $\overline{\mathscr{P}}^{kq}$, then this point will not necessarily be feasible to the subproblems. On the other hand, it is easy to show that the point (x^k, y^k, ζ^k), where $\zeta_I^k = x_I^k y_I^k$ and $\zeta_i^k = z_i^k$, $i \neq I$ (I is the branching index), is feasible to the latter problems and may hence be used as a starting point for these subproblems.

A final comment on computational considerations concerns the infinite convergence property of the algorithm. It is clear that the algorithm can be terminated at a prespecified ϵ degree of accuracy whenever

$$\phi(x^k, y^k) - \psi^k(x^k, y^k) < \epsilon.$$

While a rigorous proof has only been given for finite convergence to ϵ-optimum solutions, the branch-and-bound structure of the algorithm is flexible enough to

incorporate acceleration of convergence procedures. Such a procedure has been designed for the bilinear case and will be discussed in the next section. Although this procedure found the exact solutions of all sample problems in a finite number of stages, it is still an open problem whether or not acceleration of convergence techniques exist that guarantee finite convergence for specialized cases of the general problem.

Computational results. The branch-and-bound algorithm described in the preceding sections has been implemented for jointly constrained bilinear programs of the form

$$\begin{aligned} \underset{(x,\,y)}{\text{minimize}} \quad & \phi(x, y) = c^T x + x^T y + d^T y \\ \text{subject to} \quad & A\begin{pmatrix} x \\ y \end{pmatrix} = a, \\ & B\begin{pmatrix} x \\ y \end{pmatrix} \leqslant b, \\ & 0 \leqslant l \leqslant x \leqslant L, \\ & 0 \leqslant m \leqslant y \leqslant M \end{aligned}$$

where A and B are $n_1 \times 2n$ and $n_2 \times 2n$ matrices, respectively. A FORTRAN IV code implementing the algorithm was developed using the SEXOP, Version 4, package of Marsten [14] to solve the linear programming subproblems.

To improve the convergence of the algorithm for nonvertex solutions, an acceleration of convergence procedure was incorporated into the code. For each problem $\mathscr{P}^{kj}$ such that $v^{kj} < V_b^k$, a first order approximation of ϕ about the point (x^{kj}, y^{kj}) is minimized over the constraint set $S \cap \Omega^{kj}$. The solution, say $(\tilde{x}^{kj}, \tilde{y}^{kj})$, of this linear program is a vertex of $S \cap \Omega^{kj}$, whereas (x^{kj}, y^{kj}) is only guaranteed to be on an edge or a face of the polyhedron $S \cap \Omega^{kj}$. This step of the procedure will be recognized as an application of the Frank-Wolfe method [6]. The next step is to minimize ϕ over the line segment connecting the points (x^{kj}, y^{kj}) and $(\tilde{x}^{kj}, \tilde{y}^{kj})$. The latter problem is a one-dimensional quadratic program that admits a closed form solution. The new point thus generated yields a higher value of ψ^{kj} than does (x^{kj}, y^{kj}) and is hence a better candidate for future branching. For the sample problems solved, this procedure located all known nonvertex solutions in a finite number of steps.

Two sample problems are exhibited below. These problems were randomly generated using pseudorandom numbers between -10 and 10 for the constraint matrices and the cost vectors. The upper bounds and right-hand-side vectors were generated as pseudorandom numbers between 0 and 99.

EXAMPLE 1.

$$\begin{aligned} \underset{(x,\,y)}{\text{Minimize}} \quad & (2, -4, 8, 4, 9)x + x^T y + (3, -1, -2, -4, 5)y \\ \text{subject to} \quad & B^1 x \leqslant b^1, \qquad B^2 y \leqslant b^2, \\ & 0 \leqslant x_i \leqslant 20, \qquad 0 \leqslant y_i \leqslant 20 \end{aligned}$$

where

$$b^1 = (1, 3, 5, 4, 0)^T, \qquad b^2 = (0, 7, 3, 6, 2)^T.$$

$$B^1 = \begin{bmatrix} -8 & 0 & -6 & 7 & -7 \\ -6 & 2 & -3 & 9 & -3 \\ 6 & 0 & -7 & -8 & 2 \\ -1 & 1 & -8 & -7 & -5 \\ 4 & -7 & 4 & 5 & 1 \end{bmatrix}$$

and

$$B^2 = \begin{bmatrix} 0 & 5 & -4 & 9 & -7 \\ 7 & 4 & 3 & 7 & 5 \\ 6 & 1 & -8 & 8 & 0 \\ -3 & 2 & 7 & 0 & 1 \\ -2 & -3 & 8 & 5 & -2 \end{bmatrix}.$$

The algorithm found the minimum -45.38 at the point $x^* = (4.5667, 20, 3.2, 0, 0)$ and $y^* = (0, 0, 0.19565, 0.086957, 0)$, after exploring five nodes, four of which were deleted before solving the extra problem called for by the acceleration of convergence technique. When the problem was resolved using the objective function $(2, 4, 8, -1, -3)x + x^T y + (-2, -1, -2, -4, 5)y$, the algorithm located the minimum -42.963 at the point $x^* = (0, 5.9821, 0, 4.375, 20)$ and $y^* = (0.80645, 0, 0.45161, 0, 0)$ after solving only one linear program.

EXAMPLE 2.

$$\begin{aligned} \underset{(x, y)}{\text{Minimize}} \quad & (-1, -2, -3, -4, -5)x + x^T y + (-1, -2, -3, -4, -5)y \\ \text{subject to} \quad & B\begin{pmatrix} x \\ y \end{pmatrix} \leqslant b, \\ & x_1 + x_2 + x_3 - 2x_4 + x_5 + y_1 + y_2 + 4y_3 + y_4 + 3y_5 \leqslant 200, \\ & \sum_{i=1}^{5} x_i \geqslant 1, \qquad \sum_{i=1}^{5} y_i \geqslant 2, \\ & 0 \leqslant x_i \leqslant 100, \qquad 0 \leqslant y_i \leqslant 100, \end{aligned}$$

where

$$b = (80, 57, 92, 55, 76, 14, 47, 51, 36, 92)^T$$

and

$$B = \begin{bmatrix} 1 & 7 & 5 & 5 & 0 & -6 & -3 & -3 & 5 & -7 \\ -3 & 3 & 8 & 7 & -9 & -7 & -9 & 0 & 8 & -7 \\ 1 & 0 & 1 & 3 & 8 & 9 & 0 & 9 & -7 & -8 \\ -1 & -2 & 2 & 0 & 9 & 5 & -3 & 1 & -1 & -5 \\ -5 & 8 & -8 & 0 & 3 & 0 & 4 & -5 & -2 & 9 \\ 4 & -1 & 6 & -4 & -7 & -8 & -7 & 6 & -2 & -9 \\ 0 & 7 & 4 & 0 & 9 & 0 & 0 & -6 & -5 & -5 \\ -5 & -1 & 0 & 7 & -1 & 2 & 5 & -8 & -5 & 2 \\ -4 & -7 & 0 & -9 & 2 & 6 & -9 & 1 & -5 & 0 \\ -2 & 6 & 0 & 8 & -6 & 8 & 8 & 5 & 2 & -7 \end{bmatrix}.$$

Several problems were solved using the same B matrix and b vector. After each solution, the cost vector and feasible region bounding constraints were arbitrarily adjusted in an attempt to produce a more challenging problem. The problem exhibited above was the result of this experiment. The algorithm found the minimum -794.86 at the point $x^* = (100, 0, 0, 80.94, 0)$ and $y^* = (0, 0, 17.828, 0, 63.523)$. For this problem, the extra linear program was solved at seven of the thirteen nodes explored.

Extensions. With a little extra work the algorithm described in this paper can be used to solve programs having a general bilinear form in the objective function; namely,

$$\begin{aligned} \underset{(x, y)}{\text{minimize}} \quad & \phi(x, y) = f(x) + x^T A y + g(y) \\ \text{subject to} \quad & (x, y) \in S \cap \Omega \end{aligned}$$

where A is a general $p \times q$ matrix, and all functions and sets are as defined earlier. The extra work involved takes the form of exploiting any structure that A may have and then making a coordinate change that transforms the bilinear term x^TAy into an inner product term in the new coordinate variables. These coordinate transformations correspond to the introduction of new bases for the range space and domain space of A. Care must be taken to compute upper and lower bounds on the new variables, preferably as tight as possible. We will illustrate the approach by describing how certain A matrices could be handled.

When A is nonsingular and is also isotone (or antitone), making the transformation $z = Ay$ immediately leads to a canonical form accepted by the algorithm. If A is only nonsingular, a hyperrectangle bounding the z-vector can be computed with little extra work. The most general problem arises when A is nonsquare with rank ρ. For this case we appeal to matrix factorization techniques that have been extensively used in large-scale linear programming procedures. In particular, the $A = QRZ$ decomposition and the singular-value decomposition lead to a suitable transformed problem. The theory and numerical techniques involved are discussed in Stewart [19] and Noble [16]. These matrix decomposition procedures yield an inner product in the new coordinates with only ρ nonzero terms. Thus the added work in decomposing A is exploited by the algorithm, which now has only ρ convex envelopes and branching indices to account for—a considerable savings when ρ is small.

Another extension of the algorithm is to permit f and g to be separable concave functions. That is, the objective function can be expressed as

$$\phi(x, y) = \sum_{i=1}^{n} f_i(x_i) + x^Ty + \sum_{i=1}^{n} g_i(y_i)$$

where all f_i and g_i are concave. Convex envelopes of f_i and g_i over the bounded variables are easily computed. Moreover, the convex enevelope of ϕ over Ω, not just a convex underestimating function as we had before, is now available as a result of the following.

THEOREM 4. *Let f_i and g_i be concave; then*

$$\mathrm{Vex}_{\Omega}\phi(x, y) = \sum_{i=1}^{n} \{\mathrm{Vex}_{[l_i, L_i]} f_i(x_i) + \mathrm{Vex}_{\Omega_i} x_i y_i + \mathrm{Vex}_{[m_i, M_i]} g_i(y_i)\}.$$

The proof of this theorem is a direct extension of the arguments used in Theorem 2.

As defined above, the function ϕ is biconcave. With minor adjustments to the branching rule and convergence proof, our approach can be applied to the problem of minimizing a biconcave function over $S \cap \Omega$, the solution of which is known to lie on the boundary of the feasible set. Moreover, the matrix factorization procedures can be invoked to handle biconcave objective functions having a general bilinear term.

References

[1] Altman, M. (1968). Bilinear Programming. *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phy.* **19** 741–746.

[2] Cabot, A. Victor and Francis, Richard L. (1970). Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points. *Oper. Res.* **18** 82–86.

[3] Falk, James E. (1969). Lagrange Multipliers and Nonconvex Programs. *SIAM J. Control* **7** 534–545.

[4] ———. (1973). A Linear Max-min Problem. *Math. Programming* **5** 169–188.

[5] ——— and Soland, Richard M. (1969). An Algorithm for Separable Nonconvex Programming Problems. *Management Sci.* **15** 550–569.

[6] Frank, Marguerite and Wolfe, Philip. (1956). An Algorithm for Quadratic Programming. *Naval Res. Logist. Quart.* **3** 95–110.

[7] Gallo, Giorgio and Ülkücü, Aydin. (1977). Bilinear Programming: An Exact Algorithm. *Math. Programming* **12** 173–194.

[8] Horst, Reiner. (1976). An Algorithm for Nonconvex Programming Problems. *Math. Programming* **10** 312–321.
[9] Konno, Hiroshi. (1971). Bilinear Programming: Part I. An Algorithm for Solving Bilinear Programs. Technical Report No. 71-9, Operations Research House, Stanford University.
[10] ———. (1971). Bilinear Programming: Part II. Applications of Bilinear Programming. Technical Report No. 71-10, Operations Research House, Stanford University.
[11] ———. (1976). A Cutting Plane Algorithm for Solving Bilinear Programs. *Math. Programming* **11** 14–27.
[12] Mangasarian, Olvi L. (1964). Equilibrium Points of Bimatrix Games. *SIAM J.* **12** 778–780.
[13] ——— and Stone, H. (1964). Two Person Nonzero-sum Games and Quadratic Programming. *J. Math. Anal. Appl.* **9** 345–355.
[14] Marsten, Roy E. (1974). User's Manual for SEXOP (Subroutines for Experimental Optimization). Release 4. University of Arizona.
[15] Mills, H. (1960). Equilibrium Points in Finite Games. *SIAM J.* **8** 397–402.
[16] Noble, Ben and Daniel, James W. (1977). *Applied Linear Algebra*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
[17] Rudin, Walter. (1964). *Principles of Mathematical Analysis*, second edition. McGraw-Hill, Inc., New York.
[18] Sherali, Hanif D. and Shetty, C. M. (1980). A Finitely Convergent Algorithm for Bilinear Programming Problems Using Polar Cuts and Disjunctive Face Cuts. *Math. Programming* **19** 14–31.
[19] Stewart, G. W. (1973). *Introduction to Matrix Computation*. Academic Press, Inc., New York.
[20] Vaish, Harish and Shetty, C. M. (1976). The Bilinear Programming Problem. *Naval Res. Logist. Quart.* **23** 303–309.
[21] ——— and ———. (1977). A Cutting Plane Algorithm for the Bilinear Programming Problem. *Naval Res. Logist. Quart.* **24** 83–94.

AL-KHAYYAL: SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING, GEORGIA INSTITUTE OF TECHNOLOGY, ATLANTA, GEORGIA 30332
FALK: DEPARTMENT OF OPERATIONS RESEARCH, THE GEORGE WASHINGTON UNIVERSITY, WASHINGTON, D.C. 20052