A Global Optimization Approach for the Linear Two-Level Program

HOANG TUY*
Institute of Mathematics, P.O. Box 631, BoHo, Hanoi, Vietnam

ATHANASIOS MIGDALAS and PETER VÄRBRAND
Department of Mathematics, Linköping Institute of Technology, S-581 83 Linköping, Sweden

(Received: 30 July 1990; accepted: 6 January 1992)

Abstract. Linear two-level programming deals with optimization problems in which the constraint region is implicity determined by another optimization problem. Mathematical programs of this type arise in connection with policy problems to which the Stackelberg leader-follower game is applicable. In this paper, the linear two-level programming problem is restated as a global optimization problem and a new solution method based on this approach is developed. The most important feature of this new method is that it attempts to take full advantage of the structure in the constraints using some recent global optimization techniques. A small example is solved in order to illustrate the approach.

Key words. Linear two-level program, global optimization, Stackelberg game, reverse convex constraint programming, polyhedral annexation method.

1. Introduction

Linear two-level programming, a special case of multi-level programming, deals with optimization problems in which the constraint region is implicitly determined by another optimization problem.

The model can be considered as a two-person game where one of the players, the leader, knows the cost function mapping of the second player, the follower, who may or may not know the cost function of the leader. The follower knows however the strategy of the leader and takes this into account when computing his own strategy. The leader can foresee the reactions of the follower and can therefore optimize his choice of strategy.

The basic leader/follower strategy was originally proposed for a duopoly by von Stackelberg [17]. Of particular interest in the range of policy problems to which the Stackelberg game is applicable are certain hierarchical decision-making systems in mixed economies where policy makers at the top level influence the decisions of private individuals and companies. In order to reduce a country's dependence on imported energy resources, for instance, a government can impose retail sales tax, import quotas and rationing. The energy consumption of in-

*The paper was completed while this author was visiting the Department of Mathematics of Linköping University.

dividuals and companies will consequently adjust their consumption in accordance with respect to the resulting availability and prices. This will in turn affect import levels, the general price level and government revenue.

With respect to land-use policies [6], a government can decide to invest in order to provide transportation and communication infrastructures. Individual farmers and traders can then make their own decisions about production and marketing.

The basic Stackelberg game involves two players. The first one chooses strategy x and the second player chooses strategy y. The cost function associated with the first player is

$$c_1^T x + d_1^T y$$

and the cost function of the second player is

$$c_2^T x + d_2^T y .$$

Designating the first player as leader and the second as follower, we have the following scenario:

For each decision x that the leader takes, the follower chooses $y = \omega(x)$, where ω is a mapping from x to y such that

$$c_2^T x + d_2^T \omega(x) \leq c_2^T x + d_2^T y \tag{I}$$

for all feasible y. The leader chooses x^* such that

$$c_1^T x^* + d_1^T \omega(x^*) \leq c_1^T x + d_1^T \omega(x) \tag{II}$$

for all feasible x.

The strategy x^* is the Stackelberg strategy for the first player while $y^* = \omega(x^*)$ is the Stackelberg strategy for the second player.

Inequalities (I) and (II) suggest the following two-level optimization problem, where the first level problem, the outer problem, is associated with the leader, while the second level problem, the inner problem, is associated with the follower.

$$\textbf{(P)} \quad \min_{x \geq 0} c_1^T x + d_1^T y \tag{1}$$

$$\text{s.t. } A_1 x + B_1 y \leq g_1 \tag{2}$$

where y solves

$$\min_{y \geq 0} c_2^T x + d_2^T y \tag{3}$$

$$\text{s.t. } A_2 x + B_2 y \leq g_2 \tag{4}$$

$$(x \in R^p,\ y \in R^q,\ g_1 \in R^{m_1},\ g_2 \in R^{m_2}) .$$

A great deal of progress has been made in developing algorithms for this problem. The first class of methods to be mentioned is based on enumeration techniques. The motivation for choosing such an approach arises from the fact that an optimal solution to problem (**P**) can be found which is a basic feasible solution of the set of all linear constraints in the model. In such a case, we need a procedure that enumerates the vertices of the feasible set in an efficient way. The most widely known algorithms based on this approach are the enumeration method by Candler and Townsley [7], "The Kth best algorithm" by Bialas and Karwan [5] and the B&B-algorithm by Moore and Bard [4].

Another straightforward approach is to replace the inner problem by its corresponding Karush–Kuhn–Tucker conditions and hence obtain an ordinary mathematical programming problem with a single objective function. The difficulty here occurs instead in the set of constraints – the complementary slackness conditions. Several approaches have been suggested to take care of this difficulty. Bard and Falk [2] suggest a branch and bound approach where the complementary slackness conditions are replaced by a set of equations giving a separable non-convex program. Fortuny and McCarl [8] suggest a transformation giving a large mixed integer programming problem.

Another class of solution methods tries to solve the linear two-level programming problem via multiple objective linear programming [3, 22]. Here the two objective functions are weighted together to give a standard linear programming problem. However, Wen and Hsu [23] have recently shown that in general, there is no such relationship between bilevel and bicriteria programming problems.

Recently, local optimization methods on nonlinear programming have been used to approach the optimal solution smoothly. Such methods are for instance, penalty or barrier function methods and direct gradient methods (see, e.g., [13, 14]).

The implicit enumeration methods mentioned above tend to generate large search trees while at the same time, an ever increasing set of constraints is encountered. Implicit enumeration methods as well as the branch-and-bound approach often fail to utilize and exploit specific structures inherent to the problem. On the other hand, because of the nonconvexity of the problem, local optimization methods do not generally guarantee a global optimal solution. The best proposed Branch and Bound procedure appears in Hansen et al. [9].

In this paper, our aim is to restate the linear two-level programming problem as a global optimization problem and develop a new solution method based on this approach. The most important feature of this new method is that it attempts to take full advantage of the structure in the constraints using some recently global optimization techniques [12].

Hopefully this will open up a new way for efficiently handling a large class of problems which would otherwise be very difficult to attack. In this connection, we should mention that a global optimization approach to bilevel programming in the nonlinear case has been earlier proposed in [1].

The paper consists of 7 sections. In Section 2 we establish some general properties which allow the problem to be restated as a reverse convex constrained program, i.e., a program which differs from a conventional linear program only by the presence of an additional reverse convex constraint. Section 3 is devoted to preliminary transformations and to the formulation of a subproblem which plays a central role in the subsequent development. In Section 4 we establish a basic structural property for the constraints and outline the new method which seems to be particularly suitable for exploiting this structural property and for significantly reducing the dimension of the problem in many circumstances. Sections 5 and 6 are devoted to the development of the algorithm. Finally, Section 7 concludes the work with an illustrative example.

2. General Properties

Observing that the inner problem involves only minimization over y (recall inequality (I)), we restate (3)–(4) as follows:

where y solves the linear program $\mathbf{R}(x)$:

$$\min\{d_2^T y\colon A_2x + B_2y \leqslant g_2\,, \quad y \geqslant 0\}\,.$$

Without loss of generality we therefore subsequently assume that $c_2 = 0$. Denote by $\varphi(x)$ the optimal value of $\mathbf{R}(x)$. Note that $\varphi(x) = +\infty$ if $\mathbf{R}(x)$ is infeasible.

PROPOSITION 1. *$\varphi(x)$ is a convex polyhedral function.*

THEOREM 1. $(\mathbf{P})$ *is equivalent to the reverse convex programming problem*:

$$(\mathbf{Q})\quad \min c_1^T x + d_1^T y$$

$$\text{s.t.}\quad A_1x + B_1y \leqslant g_1 \tag{5}$$

$$A_2x + B_2y \leqslant g_2 \tag{6}$$

$$x,\ y \geqslant 0 \tag{7}$$

$$d_2^T y \leqslant \varphi(x)\,. \tag{8}$$

Setting $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$, $g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$ we can rewrite $(\mathbf{Q})$ as

$$(\mathbf{Q})\quad \min c_1^T x + d_1^T y$$

$$\text{s.t.}\quad Ax + By \leqslant g \tag{9}$$

$$x \geqslant 0, \quad y \geqslant 0 \tag{10}$$

$$d_2^T y \leqslant \varphi(x)\,. \tag{11}$$

All constraints of $(\mathbf{Q})$ are linear, except the last one which is reverse convex. Thus, $(\mathbf{Q})$ is a linear program with an additional reverse convex constraint.

PROPOSITION 2. *If* (**Q**) *is solvable, at least an optimal solution is achieved at a vertex of the polyhedron (9)–(10).*

Proof. For all (x, y) satisfying (9)–(10) we must have $\varphi(x) \leq d_2^T y$. Since an optimal solution of (**Q**) must be an optimal solution of (**P**) it follows that $d_2^T y = \varphi(x)$. Therefore, an optimal solution of (**Q**) must maximize the convex function $\varphi(x) - d_2^T y$ over the polyhedron (9)–(10). But it is known that the set of all (x, y) where this maximum is attained is a union of faces of the polyhedron (9)–(10) (see Rockafellar, Corollary 32.1.1). Since an optimal solution of (**Q**) must minimize the linear function $c_1^T x + d_1^T y$ over this union, it follows that at least one optimal solution is achieved at a vertex of the polyhedron (9)–(10). □

Methods for solving linear programs with an additional reverse convex constraint have been developed in recent years by several authors ([10, 12, 15, 18] and the references therein). However, since our problem (**Q**) has a specific structure, to solve it efficiently it is important to devise a method which could take advantage of this structure.

In the sequel we will use the technique recently developed in [21] to substantially reduce the dimension of the global optimization problem to be solved. For this, we observe that any two strategies x', x of the leader such that $A_2\,(x' - x) = 0$ will cause the same response from the follower because $R(x') = R(x)$. On the other hand, any two strategies y', y of the follower such that $d_2^T(y' - y) = 0$ will have the same effect on the objective function of the leader. Therefore, from the overall point of view, two strategies (x', y'), (x, y) are equivalent if $A_2(x' - x) = 0$ and $d_2^T(y' - y) = 0$. That is, what we are looking for is not really a strategy (x, y) but an equivalent class of strategies (x, y), with respect to the just defined equivalence relation. Consequently, instead of working in the original $(x, y)-$space, we can work in the quotient space formed by all equivalent classes with respect to the relations $A_2(x' - x) = 0$ and $d_2^T(y' - y) = 0$. Since the dimension of this quotient space is at most $1 + \mathit{rank}\ A_2$ we see that the size and hence the difficulty of the problem, depends mainly on the number of independent constraints of the subproblem $\mathbf{R}(x)$. It is in fact the presence of $\mathbf{R}(x)$ which is responsible for the nonlinearity of the problem.

3. Preliminary Transformations

Let (x^0, y^0) be an optimal basic solution of the linear program, obtained from (Q) by omitting the reverse convex constrains (11). If $\varphi(x^0) = d_2^T y^0$ then (x^0, y^0) solves (Q). Therefore we shall assume that

$$\varphi(x^0) - d_2^T y^0 < 0\,. \tag{12}$$

Introducing the slack variables $s = g - Ax - By$, we can write the system (9) (10) as

$$Ax + By + s = g \tag{13}$$

$$x \geqslant 0\,, \quad y \geqslant 0\,, \quad s \geqslant 0\,. \tag{14}$$

Setting $\tilde{x} = (x, y, s)$, $\tilde{A} = (A, B, I)$ with I the identity matrix of order $m_1 + m_2$, we obtain a more compact form

$$\tilde{A}\tilde{x} = g\,, \quad \tilde{x} \geqslant 0\,. \tag{15}$$

With $s^0 = g - Ax^0 - By^0$, the point $\tilde{x}^0 = (x^0, y^0, s^0)$ is a vertex of the polyhedron (15).

Denote by $\tilde{x}_J = (\tilde{x}_i, i \in J)$ and $\tilde{x}_N = (\tilde{x}_i, i \in N)$ the basic and nonbasic variables, respectively, relative to the basic solution $\tilde{x}^0$ of (15). (J and N are subsets of the set $\{1, \ldots, p + q + m_1 + m_2\}$). The basic variables can be expressed in terms of the nonbasic ones as

$$\tilde{x}_J = \tilde{x}_J^0 - W\tilde{x}_N \quad (\tilde{x}_J \geqslant 0\,, \quad \tilde{x}_N \geqslant 0)\,, \tag{16}$$

where W is a certain matrix. Setting now $u = \tilde{x}_N$, $\tilde{b} = \tilde{x}_J^0$ we can also write (15) (i.e. (9) (10) in the form

$$Wu \leqslant \tilde{b} \tag{17}$$

$$u \geqslant 0\,. \tag{18}$$

Furthermore, for each given u we can determine the corresponding vector $\tilde{x} = (x, y, s)$ by the formulas

$$\tilde{x}_N = u\,, \quad \tilde{x}_J = \tilde{x}_J^0 - W\tilde{x}_N = \tilde{b} - Wu \tag{19}$$

which give the affine mappings

$$x = x^0 + \xi u\,, \quad y = y^0 + \eta u\,, \tag{20}$$

where ξ and η are known matrices. Note that $u \in R^{p+q}$ (because $|N| = p + q$, $|J| = m_1 + m_2$). Setting

$$l(u) = c_1^T(x^0 + \xi u) + d_1^T(y^0 + \eta u) \tag{21}$$

$$\phi(u) = \varphi(x^0 + \xi u)\,, \quad h(u) = d_2^T(y^0 + \eta u) \tag{22}$$

we can finally rewrite (**Q**) in the form

$$(\tilde{\mathbf{Q}}) \quad \min l(u) \tag{23}$$

$$\text{s.t.} \qquad Wu \leqslant \tilde{b} \tag{24}$$

$$u \geqslant 0 \tag{25}$$

$$\phi(u) - h(u) \geqslant 0\,. \tag{26}$$

Here $l(u)$ and $h(u)$ are affine functions, $\phi(u)$ is a convex function and moreover, the data is such that

1. $u = 0$ is a vertex of the polyhedron

$$D = \{u\colon Wu \leq \tilde{b}\,, \quad u \geq 0\}\,. \tag{27}$$

2. $\phi(0) - h(0) < 0$, i.e., $u = 0$ belongs to the convex set

$$C = \{u\colon \phi(u) - h(u) < 0\}\,. \tag{28}$$

3. The closure of C is $\bar{C} = \{u\colon \phi(u) - h(u) \leq 0\}$ and we have

$$D \subseteq \bar{C}\,. \tag{29}$$

The latter property is due to the lower semi-continuity of $\varphi(x)$ (which implies that $\bar{C}$ is closed) and the fact that for any $u \in D$ the point $(x^0 + \xi u, y^0 + \eta u)$ satisfies (6), i.e., $y^0 + \eta u$ is feasible to the linear program $\mathbf{R}(x^0 + \xi u)$ (so that $\phi(u) = \varphi(x^0 + \xi u) \leq d_2^T(y^0 + \eta u) = h(u)$, hence (30)).

Thus, using simple manipulations, the original problem (**P**) can be converted to the form ($\tilde{\mathbf{Q}}$), which amounts to

$$\min l(u) \quad \text{s.t.} \quad u \in D \backslash C \tag{30}$$

where $l(u)$ is an affine function, D is a polyhedron and C is a convex set with the properties 1, 2 and 3. To solve this problem with a tolerance $\epsilon > 0$ we can proceed according to the following scheme:

Find a vertex u^1 of $D_1 = D$ that does not lie in C; then find a point u^2 of $D_2 = D_1 \cap \{u\colon l(u) \leq l(u^1) - \epsilon\}$ that does not lie in C (D_2 is obtained from D_1 by cutting off u^1); and so on, until we get a polyhedron D_r entirely contained in C (then the last vertex u^{r-1}, if any, solves our problem).

Obviously, a method for solving the following subproblem is central to this scheme:

(**SP**) *Given the convex set C defined by (28) and a polyhedron $D \subset C$ such that $0 \in D \cap C$, find a point of $D \backslash C$, if there is one, or else establish that no such point exists (i.e., $D \subset C$).*

In the next section we shall deal with this subproblem. To ease the presentation, it is convenient to define here a construction which will be needed repeatedly in the sequel.

Given a point $u \neq 0$ we denote by $Ext(u)$ the last point where the ray from 0 through u meets the boundary ∂C of C, i.e., $Ext(u) = \theta u$, where

$$\theta = sup\{\tau\colon \phi(\tau u) - h(\tau u) \leq 0\}\,. \tag{31}$$

The construction of $Ext(u)$ amounts to solving a linear program as shown by the following.

PROPOSITION 3. *θ is equal to the optimal value of the linear program*

$$\max_{y,\tau} \tau \tag{32}$$

$$\text{s.t.} \quad d_2^T(y - y^0 - \tau\eta u) \leqslant 0 \tag{33}$$

$$A_2(x^0 + \tau\xi u) + B_2 y \leqslant g_2 \tag{34}$$

$$y \geqslant 0, \quad \tau \geqslant 0\,. \tag{35}$$

Proof. If y and τ satisfy (33)–(35) then y is feasible to the linear program $\mathbf{R}(x^0 + \tau\xi u)$, hence $d_2^T y \geqslant \varphi(x^0 + \tau\xi u)$ and consequently $d_2^T(y^0 + \tau\eta u) \geqslant d_2^T y \geqslant \varphi(x^0 + \tau\xi u)$ by (33), i.e., $h(\tau u) \geqslant \phi(\tau u)$. Conversely, if $\phi(\tau u) - h(\tau u) \leqslant 0$ then there exists a y feasible to $\mathbf{R}(x^0 + \tau\xi u)$ such that $d_2^T y = \phi(\tau u) \leqslant h(\tau u) = d_2^T(y^0 + \tau\eta u)$ hence y and τ satisfy (33)–(35). □

NOTE. Assuming $\theta < +\infty$ we obviously have $Ext(u) \in \partial C$. However, since C may not be open (unless $\varphi(x)$ and hence $\phi(u)$ is continuous), $Ext(u)$ may belong to C or to $\bar{C}\backslash C$. We can then construct a point $\hat{u} \in C$ in the line segment $[u; Ext(u)]$ as follows. If $Ext(u) \in C$ we let $\hat{u} = Ext(u)$. Otherwise, it follows from the convexity of C that every point in the line segment $[u; Ext(u)]$, except $Ext(u)$, belongs to C; then we let $\hat{u}$ be any point of C in this line segment (for the efficiency of the algorithm to be developed below $\hat{u}$ should be taken as close to $Ext(u)$ as possible). To recall the construction of $\hat{u}$ from u we will write $\hat{u} \approx Ext(u)$ in the sequel.

If $\theta = +\infty$ then $\phi(\tau u) - h(\tau u) \leqslant 0$, $\forall \tau > 0$, i.e., the convex function ϕ is bounded above on the ray Γ from 0 through u. But then, by well known properties of convex functions (see, e.g., ([16]), Corollary 32.3.4), ϕ achieves its maximum over Γ at point 0, i.e., $\phi(\tau u) - h(\tau u) \leqslant \phi(0) - h(0) < 0$, $\forall \tau > 0$. In this case we set $\hat{u} = \theta_\infty u$, where θ_∞ is an arbitrarily large positive number.

4. Finding a Point of $D\backslash C$

In this section we outline a method called "polyhedral annexation" [19] (see also [12, 21]) for solving the subproblem (**SP**) formulated in the previous section.

Denote $f(u) = \phi(u) - h(u)$. Since we wish to find a point $\bar{u}$ of D such that $f(\bar{u}) = 0$, while $f(u) \leqslant 0$, $\forall u \in D$, the problem can be solved by maximizing the convex function $f(u)$ over the polyhedron D. Indeed, if this maximum is negative then no point $u \in D\backslash C$ exists; otherwise, this maximum is equal to 0 (and a maximizer can always be found which is a vertex of D).

Several methods are currently available for solving convex maximization problems (see [12]). For our purpose here, however, an efficient method should take advantage of some specific structural properties of the convex set C which we are going to show.

PROPOSITION 4. *We have $K \subset C$, where K is the cone*

$$K = \{u: A_2(\xi u) \leqslant 0\,, \quad d_2^T(\eta u) \geqslant 0\}\,. \tag{36}$$

Proof. Let $u \in K$. We have to prove that $f(u) < 0$. But it is easy to see that since $A_2(x^0 + \xi u) \leqslant A_2 x^0$ the feasible set of $\mathbf{R}(x^0 + \xi u)$ contains that of $\mathbf{R}(x^0)$. Indeed, if y is feasible to $\mathbf{R}(x^0)$, i.e., if $A_2 x^0 + B_2 y \leqslant g_2$, $y \geqslant 0$ then $A_2(x^0 + \xi u) + B_2 y \leqslant g_2$, $y \geqslant 0$, which means that y is also feasible to $\mathbf{R}(x^0 + \xi u)$. Hence, $\phi(u) \leqslant \varphi(x^0) = \phi(0)$. On the other hand, $d_2^T(y^0 + \eta u) \geqslant d_2^T y^0$ by hypothesis. Therefore $f(u) = \phi(u) - h(u) = \phi(u) - d_2^T(y^0 + \eta u) \leqslant \phi(0) - d_2^T y^0 = \phi(0) - h(0) < 0$. □

Let

$$\tilde{A}_2 = A_2 \xi\,, \quad \tilde{d}_2 = d_2^T \eta\,. \tag{37}$$

For any set $M \subset R^n$ denote by M^* the polar of M, i.e., the set of all $v \in R^n$ satisfying $v^T u \leqslant 1$, $\forall u \in M$.

PROPOSITION 5. *The polar K^* of K is the convex cone generated by the m_2 rows of $\tilde{A}_2$ and $-\tilde{d}_2$, i.e.,*

$$K^* = \{v = \tilde{A}_2^T \lambda - \lambda_0 \tilde{d}_2: \quad \lambda \in R_+^{m_2}\,, \quad \lambda_0 \in R_+\}\,.$$

Proof. See [16], Section 14. Since K is a cone, $v \in K^*$ if and only if $v^T u \leqslant 0$ for all u satisfying $\tilde{A}_2 u \leqslant 0$, $-\tilde{d}_2^T u \leqslant 0$ and the result follows by applying Farkas Lemma. □

COROLLARY 1. *The polar C^* of C is contained in the cone K^* with dim $K^* \leqslant$ rank $A_2 + 1 \leqslant m_2 + 1$.*

Proof. Since $K \subset C$ (Proposition 4) it follows that $C^* \subset K^*$ and from (37) we derive *dim* $K^* \leqslant$ *rank* $A_2 + 1$. □

Note that in most cases *rank*$A_2 + 1 \ll p + q$. This suggests that we should apply the version of polyhedral annexation method as developed in [21] to the problem (**SP**) in order to reduce the dimension of the problem to be solved.

The basic idea of polyhedral annexation is to construct adaptively a sequence of expanding polyhedrons

$$P_1 \subset P_2 \subset \dots$$

approximating the convex set C more and more closely from the interior until we obtain a polyhedron P_k such that $D \subset P_k$ or find a point $u \in D \backslash C$.

Specifically, we start from a polyhedron P_1 such that

$$L \subset P_1 \subset C\,, \tag{38}$$

where L is the lineality space of K (the largest linear space contained in K). Since $0 \in L \subset P_1$, with each facet of P_1 we can associate a vector v, normal to this facet, such that the hyperplane through this facet is described by the equation $v^T u = 0$ (if this facet contains the origin 0) or $v^T u = 1$ (if it does not). Denote by V_1 the set of all vectors v associated this way with the facets of P_1 and by V_1^* the subset of V_1 consisting of all vectors v associated with the facets that contain 0. Then P_1 is the polyhedron determined by the system of linear inequalities

$$v^T u \leqslant \beta_v \quad (v \in V_1)\,, \tag{39}$$

where $\beta_v = 0$ if $v \in V_1^*$ and $\beta_v = 1$ otherwise. We shall refer to the system (39) as the *defining system* of P_1.

Knowing the defining system of P_1 it is easy to check whether $D \subset P_1$. Indeed, for each $v \in V_1$ we can compute

$$\mu(v) = \max\{v^T u : \quad u \in D\}\,. \tag{40}$$

If it so happens that

$$\mu(v) \leqslant \beta_v\,, \quad \forall v \in V_1$$

then, obviously, $D \subset P_1$ (and consequently, $D \subset C$, i.e. no point $u \in D \backslash C$ exists). Otherwise we consider

$$v^1 \in \arg\max\{\mu(v) - \beta_v : \quad v \in V_1\}\,, \tag{41}$$

$$u^1 \in \arg\max\{(v^1)^T u : \quad u \in D\}\,. \tag{42}$$

Then $\mu(v^1) > \beta_{v^1}$ and u^1 is a vertex of D such that $u^1 \notin P_1$. If, luckily, $f(u^1) = 0$, we are done. Otherwise, since $D \subset \bar{C}$ we must have $f(u^1) < 0$, i.e., $u^1 \in C$ and since $u^1 \neq 0$ we can construct $\hat{u}^1 \approx Ext(u^1)$ (see Note following Proposition 3) and form a new polyhedron P_2 by 'annexing' $\hat{u}^1$ to P, i.e., by taking

$$P_2 = conv\ (P_1 \cup \{\hat{u}^1\}\,. \tag{43}$$

Clearly, $L \subset P_1 \subset P_2 \subset C$, so the process can now be repeated with P_2 in place of P_1.

In this way we generate a sequence of vertices of D: $u^1, u^2, \ldots$, all of which are distinct. Since the vertex set of D is finite, the procedure is guaranteed to terminate in finitely many steps.

A crucial point which should of course be clarified in the above procedure, is how to compute the defining system of P_2 given by (43). This is where the structural properties of C which have been mentioned come into play.

In fact, from (43) it is easily seen that if $\hat{u}^1 = \theta_1 u^1$ then

$$P_2^* = P_1^* \cap \left\{ v : \quad v^T u^1 \leq \frac{1}{\theta_1} \right\}, \tag{44}$$

i.e., the polar P_2^* of P_2 is obtained from the polar P_1^* of P_1 by adjoining an additional linear constraint

$$\langle u^1, v \rangle \leq \frac{1}{\theta_1} \tag{45}$$

(with the usual convention $1/\infty = 0$). Furthermore, it can be proved that (see, e.g., [16, 12]):

PROPOSITION 6. *Let P be a polyhedron containing* 0 *and let P^* be its polar. Then the defining system of P is*

$$v^T u \leq \beta_v \quad (v \in V)$$

where V is the set of nonzero generalized vertices of P^* and $\beta_v = 1$ if v is a vertex, $\beta_v = 0$ if v is an extreme direction.

(By generalized vertex we mean either a vertex or an extreme direction, i.e., a vertex "at infinity".)

Thus, V_1 is given by the generalized vertex set of P_1^*. Since P_2^* differs from P_1^* only by an additional linear constraint, the generalized vertex set V_2 of P_2^* (which yields the defining system of P_2) can be derived from V_1 by currently available procedures (see [11, 12]). As for V_1 itself, it can be considered to be known since P_1 is of our choice.

Thus, instead of working with the polyhedrons $P_1, P_2, \ldots$, it will suffice to work with their polars $P_1^*, P_2^*, \ldots$. Noting that $P_1^* \supset P_2^* \supset \ldots$ and by (38) $C^* \subset P_1^* \subset L^*$ we see that all these polars are contained in L^* which is just the linear space spanned by K^*. Therefore, the above procedure will actually operate in a space of dimension at most $rank A_2 + 1$ only, rather than in the original space (of dimension $p + q$).

To complete our description of the method for solving (**SP**) it now remains to examine how to choose the initial polyhedron P_1.

5. Construction of the Initial Polyhedron P_1

Recall that P_1 must satisfy condition (38), and must be such that the vertices of its polar P_1^* can be determined in a straightforward manner.

Let $a^1, \ldots, a^{m_2}$ be the rows of the matrix $\tilde{A}_2 = A_2 \xi$ and let $a^0 = -\tilde{d}_2 = -d_2^T \eta$ (see (36)) (all these $a^i, i = 0, 1, \ldots, m_2$, are elements of R^{p+q}). Select among $a^0, a^1, \ldots, a^{m_2}$ a maximal subset of independent vectors, for example a^i, $i \in I$, where $I \subset \{0, 1, \ldots, m_2\}$. Then each a^j $(j = 0, 1, \ldots, m_2)$ can be expressed uniquely as

$$a^j = \sum_{i\in I} \alpha_{ij} a^i \quad (j = 0, 1, \ldots, m_2) \tag{46}$$

so that any vector $v = \Sigma_{j=0}^{m_2} \lambda_j a^j$ of the space generated by $a^0, a^1, \ldots, a^{m_2}$ can be rewritten as

$$v = \sum_{j=0}^{m_2} \lambda_j \Big[\sum_{i\in I} \alpha_{ij} a^i\Big] = \sum_{i\in I} \Big(\sum_{j=0}^{m_2} \alpha_{ij}\lambda_j\Big) a^i = \sum_{i\in I} t_i a^i$$

where $t_i = \Sigma_{j=0}^{m_2} \alpha_{ij}\lambda_j$. Thus by the correspondence (isomorphism)

$$v = \sum_{j=0}^{m_2} \lambda_j a^i \leftrightarrow t = (t_i, i \in I) \quad \text{where } t_i = \sum_{j=0}^{m_2} \alpha_{ij}\lambda_j \tag{47}$$

the space generated by $a^0, a^1, \ldots, a^{m_2}$ can be identified with $R^{|I|}$. Since by Proposition 5 the polar K^* of K is the convex cone generated by $a^0, a^1, \ldots, a^{m_2}$ and a^j is represented by $\alpha^j = (\alpha_{ij}, i \in I) \in R^{|I|}$, it follows that K^* is represented by the convex cone in $R^{|I|}$ generated by the points $\alpha^0, \alpha^1, \ldots, \alpha^{m_2}$. We will construct P_1 so that $K \subset P_1 \subset C$ (which implies (38)). Then $P_1^* \subset K^*$, hence P_1^* is represented by a subset S_1 of $R^{|I|}$ such that:

$$S_1 \subset \Big\{t \in R^{|I|}: \quad t = \sum_{j=0}^{m_2} \lambda_j \alpha^j, \lambda_j \geq 0 \quad (j = 0, 1, \ldots, m_2)\Big\}. \tag{48}$$

Since $P_1 \subset P_2 \subset \ldots$, and consequently $P_1^* \supset P_2^* \supset \ldots$, we have

$$S_1 \supset S_2 \supset \ldots$$

In this manner all the S_k will be contained in $R^{|I|}$, and we will work basically in $R^{|I|}$ (recall that $I \subset \{0, 1, \ldots, m_2\}$, i.e., $|I| \leq m_2 + 1$).

Let us now describe the construction of P_1.

The simplest choice is to take $P_1 = K$, so that $P_1^* = K^*$ is the cone generated by the vectors $a^0, a^1, \ldots, a^{m_2}$, i.e.,

$$S_1 = \Big\{t \in R^{|I|}: t = \sum_{j=0}^{m_2} \lambda_j \alpha^i, \lambda_j \geq 0 \quad (j = 0, 1, \ldots, m_2)\Big\}.$$

Substituting $\alpha^j = (\alpha_{ij}, i \in I)$ we have

$$t_i = \sum_{j=0}^{m_2} \lambda_j \alpha_{ij},$$

hence, taking account of the fact $\alpha_{ii} = 1$ for $i \in I$ and $\alpha_{ij} = 0$ for $i, j \in I$, $i \neq j$,

$$t_i = \lambda_i + \sum_{j\notin I} \alpha_{ij}\lambda_j \quad (i \in I).$$

Thus, S_1 can be described as the set of all $t \in R^{|I|}$ for each of which there exists (t, λ) satisfying

$$t_i - \sum_{j \notin I} \alpha_{ij}\lambda_j \geq 0 \ (i \in I) , \quad \lambda_j \geq 0 \ (j \notin I) .$$

Consider now the polytope

$$T_1 = \left\{ t\colon t = \sum_j \lambda_j \alpha^j , \quad \sum_j \lambda_j = 1 , \quad \lambda_j \geq 0 \ \forall j \right\} .$$

Clearly the set of extreme directions of the cone S_1 can be identified with the set of vertices of T_1. To compute the latter set, we use the following

PROPOSITION 7. *Every vertex t of T_1 corresponds to a vertex (t, λ) of the polyhedron*

$$t_i - \sum_{j \notin I} \alpha_{ij}\lambda_j \geq 0 \quad (i \in I) \tag{49}$$

$$\sum_{i \in I} \left(t_i - \sum_{j \notin I} \alpha_{ij}\lambda_j \right) + \sum_{j \notin I} \lambda_j = 1 \tag{50}$$

$$\lambda_j \geq 0 \quad (j \notin I) \tag{51}$$

and vice versa.

Proof. It can readily be verified that (t, λ), (t', λ'), (t'', λ'') satisfy (49)–(51) and $(t, \lambda) = [(t', \lambda') + (t'', \lambda'')]/2$ if and only if t, t', t'' belong to T_1 and $t = (t' + t'')/2$. Hence, (t, λ) is a vertex of the polyhedron (49)–(51) if and only if t is a vertex of T_1. □

Thus, the generalized vertex set V_1 of S_1 can be computed by computing the vertex set of the polyhedron (49)–(51).

When the vectors a^i, $i = 0, 1, \dots, m_2$ are linearly independent (which occurs, as can easily be checked, if the matrix A_2 has full rank and $d_2 \neq 0$), the polyhedron (49)–(51) reduces to the simplex

$$t_i \geq 0 \quad (i = 0, 1, \dots, m_2) , \quad \sum t_i = 1 \tag{52}$$

and the set of its nonzero vertices is $V_1 = \{a^i, i = 0, 1, \dots, m_2\}$.

CASE WHERE $0 \in int\, C$

If $0 \in int\, C$ (which is the case when $x^0 \in intdom\ \varphi(x)$) then it is easy to construct the initial polyhedron P_1 so that S_1 is a simplex of full dimension in $R^{|I|}$ (then all the S_k will be bounded and we can set $\beta_v = 1$ in systems like (39)).

Specifically, let $\bar{a}^i = \theta\ Ext(a^i)$ (see Proposition 3) and $\bar{b} = \theta\ Ext(b)$, where b is the barycentre of the simplex spanned by $-a^i$, $i \in I$ and θ is a positive number close to, but smaller than 1. Define

$$M_1 = conv\{\bar{b}, \bar{a}^i \ (i \in I)\} , \quad L = \{u\colon \langle a^i, u\rangle = 0, \forall i \in I\} , \quad P_1 = M_1 + L .$$

PROPOSITION 8. *The above polyhedron P_1 satisfies (38) and $0 \in int\, P_1$. The set S_1 that represents its polar according to (48) is a simplex of full dimension in $R^{|I|}$ given by the system.*

$$\sum_{i\in I} t_i\langle a^i, \bar{a}^j\rangle \leq 1 \quad (j \in I), \tag{53}$$

$$\sum_{i\in I} t_i\langle a^i, \bar{b}\rangle \leq 1 . \tag{54}$$

Proof. Clearly $L \subset P_1$ and from (36) (37)

$$K = \{u:\ \langle a^i, u\rangle \leq 0 \quad (i = 0, 1, \dots, m_2)\}$$

so that L is the lineality space of K. Since obviously $0 \in relint\, M_1$, it follows from the definition of M_1 that $M_1 \subset \theta\bar{C}$, hence $M_1 + L \subset \theta(\bar{C} + K) \subset \theta\bar{C}$, which implies $P_1 \subset C$. Furthermore, since the subspace spanned by M_1 is just $L^\perp$, it is easily seen that $0 \in int\ P_1$. This proves the first assertion of the proposition.

To prove the second assertion, observe that $P_1^* = M_1^* \cap L^*$ (* denotes the polar). But $L^* = L^\perp = \{v:\ v = \Sigma_{i\in I} t_i a^i\}$, while $M_1^* = \{v:\ \langle v, \bar{a}^i\rangle \leq 1\ \ (i \in I), \langle v, \bar{b}\rangle \leq 1\}$. Therefore, P_1^* is the set of all $v = \Sigma\, t_i a^i$ such that t_i, $(i \in I)$, satisfy the system (53) (54). Finally, the system

$$\sum_{i\in I} t_i\langle a^i, \bar{a}^i\rangle = 1 \quad (j \in I) \tag{55}$$

has a unique solution since its determinant is nonzero (Gram's determinant of vectors a^i, $i \in I$). Similarly, each system obtained from (55) by substituting $\bar{b}$ for some $\bar{a}^j$, has a unique solution. This implies that the polyhedron (53) (54) is a polytope with exactly $|I| + 1$ vertices, i.e. is a simplex of dimension $|I|$. □

NOTES

(i) When $a^0, a^1, \dots, a^{m_2}$ are linearly independent (while $\in$ int C) there is a simpler way to construct P_1. Indeed, let $P_1 = \bar{w} + K$, where $\bar{w} = \tau\ Ext(w)$ for some $\tau \in (0, 1)$ and w is the unique solution of the system $\langle a^i, w\rangle = 1$ $(i \in I)$. Since $-w \in int\, K$, $K \subset C$, we have $0 \in int\ (\bar{w} + K)$, $\bar{w} + K \subset \tau\bar{C} + K = \tau(\bar{C} + K) = \tau\bar{C}$, i.e., $0 \in int\, P_1$ and $P_1 \subset C$. Furthermore, if $\bar{w} = \theta w$ then $P_1 = \{u:\ \langle a^i, u\rangle \leq \theta\ \ (i = 0, 1, \dots, m_2)\}$ and consequently, $P_1 = conv\{a^i/\theta,\ i = 0, 1, \dots, m_2\}$ so that the simplex S_1 is defined by the system

$$t_i \geq 0 \quad (i = 0, 1, \dots, m_2), \quad \sum_{i=0}^{m_2} t_i \leq \frac{1}{\theta} .$$

Obviously, the vertices of S_1 are θe^i $(i = 0, 1, \dots, m_2)$, where e^i is the i-th unit vector of $R^{|I|}$.

6. Algorithm for Solving ($\tilde{Q}$)

We now incorporate the above method for solving the basic subproblem (**SP**) into the iterative scheme outlined in Section 3 in order to obtain an algorithm for solving the original problem (**P**), or equivalently, ($\tilde{\mathbf{Q}}$).

But before describing the detailed algorithm, we observe that given a feasible solution u^1 which is a vertex of the polyhedron D it is sometimes possible to derive a better feasible solution with relatively little cost in the following way.
Since u^1 cannot be optimal for the linear program

$$\text{minimize } l(u) \quad \text{s.t.} \quad u \in D \tag{56}$$

by performing a simplex pivot on u^1 we can obtain a vertex w of D neighbouring to u^1 which has $l(w) < l(u^1)$. If this vertex w happens to satisfy $f(w) = 0$ then it provides a new feasible solution better than u^1. We can then continue this process with w replacing u^1, and so on, until we reach a vertex $\bar{u}^1$ of D such that no vertex u adjacent to $\bar{u}^1$ with $l(u) < l(\bar{u}^1)$ satisfies $f(u) = 0$. We shall refer to this improvement process as an improvement by *local moves*. Thus, before beginning the search for a feasible solution u^2 such that $l(u^2) < l(u^1) - \epsilon$, it is useful to try to improve u^1 by local moves, whenever possible.

ALGORITHM

Compute a basic optimal solution (x^0, y^0) of the linear program obtained from (**Q**) by omitting the reverse convex constraint (11). If $\varphi(x^0) = d_2^T y^0$, stop; (x^0, y^0) solves (**P**). Otherwise, rewrite the problem into the form ($\tilde{\mathbf{Q}}$), with $x = x^0 + \xi u$, $y = y^0 + \eta u$. Define $\phi(u) = \varphi(x^0 + \xi u)$, $h(u) = d_2^T(y^0 + \eta u)$, $\tilde{A}_2 = A_2 \xi$, $d_2 = d_2^T \eta$. Select a tolerance $\epsilon > 0$.

Initialization. Let $a^0 = -\tilde{d}_2$, and let $a^1, \ldots, a^{m_2}$ be the rows of $\tilde{A}_2$.

Take a maximal subset $\{a^i : i \in I\}$ of independent vectors among $a^0, a^1, \ldots, a^{m_2}$.

Define S_1 to be the cone in $R^{|I|}$ generated by the vectors a^i, $i = 0, 1, \ldots, m_2$. Compute the set V_1 of extreme directions of S_1 (i.e., the vertex set of T_1 by Proposition 7). For each $t \in V_1$ define $\beta(t) = 0$.

Set $D_1 = D$, $N_1 = V_1$, $k = 1$ (k is the iteration counter; N_k is the set of newly generated vertices of S_k).

Iteration $k = 1, 2, \ldots$

k.1. For each $\bar{t} = (\bar{t}_i, i \in I) \in N_k$ solve the linear program

$$LP(\bar{t}) \quad \max\left\{\sum_{i\in I} \bar{t}_i\langle a^i, u\rangle: \quad u \in D_k\right\}$$

(see comment (i) below). Let $u(\bar{t})$ and $\mu(\bar{t})$ be a basic optimal solution and the optimal value of $LP(\bar{t})$ respectively.

k.2. Delete all $\bar{t} \in N_k$ such that $\mu(\bar{t}) \leq \beta(\bar{t})$ (see comment (ii) below). Let R_k denote the collection of the remaining members of the set V_k.
If $R_k = \emptyset$, terminate. If $D_k = D$ conclude that (**P**) is infeasible, if $D_k \neq D$, accept $(x^{k-1}, y^{k-1}) = (x^0 + \xi u^{k-1}, y^0 + \eta u^{k-1})$ as an ϵ-optimal solution.

k.3. If $f(u(\bar{t})) < 0$ for all $\bar{t} \in R_k$ then let $u^k = u^{k-1}$, $D_{k+1} = D_k$. Otherwise, $f(u(\bar{t})) = 0$ for some $\bar{t} \in R_k$, then let $u^k = u(\bar{t})$, or let u^k be any better feasible solution that can be obtained by local moves from $u(\bar{t})$, and define

$$D_{k+1} = D_k \cap \{u: \quad l(u) \leq l(u^k) - \epsilon\}$$

(see comment (iii) below).

k.4. Select $t^k \in \arg\max\{\mu(\bar{t}): \quad \bar{t} \in R_k\}$.
Compute $Ext(u(t^k))$ then θ_k such that $\theta_k u(t^k) \approx Ext(u(t^k))$ and define

$$S_{k+1} = S_k \cap \left\{t: \quad \sum_{i\in I} t_i\langle a^i, u(t^k)\rangle \leq \frac{1}{\theta_k}\right\}$$

(see comment (iv) below).
Compute the generalized vertex set V_{k+1} of S_{k+1} (from our knowledge of V_k, see comment (v) below). Let $N_{k+1} = (V_{k+1} \backslash V_k) \backslash \{0\}$ and for each $t \in N_{k+1}$ define $\beta(t) = 1$ if t is a finite vertex and $\beta(t) = 0$ if t is a vertex at infinity (an extreme direction).
Set $k \leftarrow k + 1$ and return to step k.1.

COMMENTS

(i) In Step k.1, since $\bar{t} \in N_k \subset V_k$, it corresponds to a generalized vertex of P_k^*, i.e., a facet of P_k, namely the facet whose normal is $v(\bar{t}) = \Sigma_{i\in I} \bar{t}_i a^i$. Then the linear program **LP**($\bar{t}$) is just to maximize the function $\langle v, u\rangle$ over the polyhedron D_k (see (40)).
The point $u(\bar{t})$ is the point of D_k that lies the farthest possible beyond the facet $v(\bar{t})$ and $\mu(\bar{t}) - \beta(\bar{t})$ measures the distance from $u(\bar{t})$ to the hyperplane of this facet. A positive feature of this algorithm is that for fixed k all the problems **LP**(t) have the same constraints while for different k's only the right hand side of the additional constraint $l(u) \leq l(u^k) - \epsilon$ may change.

(ii) If $\mu(\bar{t}) \leq \beta(\bar{t})$ for some $\bar{t} \in N_k$ then all of the polyhedron D_k lies strictly in the halfspace $\langle v(\bar{t}), u\rangle \leq \beta(\bar{t})$. Therefore, if $R_k = \emptyset$, then D_k lies in the intersection of all the halfspaces $\langle v, u\rangle \leq \beta_v$ that correspond to different v's in the defining system of P_k (i.e., to different facets of P_k), hence $D_k \subset P_k \subset C$. In this event, if $D_k = D$, then obviously ($\tilde{\mathbf{Q}}$) is infeasible; otherwise, a feasible solution u^{k+1} to ($\tilde{\mathbf{Q}}$) is already known that is the best so far obtained: then u^{k+1} solves $(\tilde{\mathbf{Q}})$, i.e., $(x^{k-1}, y^{k-1}) = (x^0 + \xi u^{k-1}, y^0 + \eta u^{k-1})$ solves (**P**), within the given tolerance ϵ.

(iii) In Step k.3 a vertex u^k of D_k may be identified that lies outside C (i.e., is feasible to problem $(\tilde{\mathbf{Q}})$). Then, the points $u \in D_k$ such that $l(u) > l(u^k) - \epsilon$ are no longer of interest for us, so we can restrict our further search to the part D_{k+1} of D contained in the halfspace $l(u) \leqslant l(u^k) - \epsilon$.

(iv) $t^k \in \arg\max\{\mu(\bar{t}) : \bar{t} \in R_k\}$ corresponds to that facet of P_k beyond which a vertex of D_k lying outside C has the best chance of being found. Therefore the procedure prescribes expanding P_k beyond this facet by "annexing" $\hat{u}(t^k) \approx Ext(u(t^k))$, where $u(t^k)$ is the vertex of D_k that lies the farthest from this facet. In terms of polars this "annexation" operation amounts to a restriction of S_k by means of the additional constraint $\langle v, u(t^k)\rangle \leqslant 1/\theta_k$ (the variable being v). Since $v = \Sigma_{i\in I} t_i a^i$, this constraint, in terms of the variables t_i, is:

$$\sum_{i\in I} t_i \langle a^i, u(t^k)\rangle \leqslant \frac{1}{\theta_k}\ .$$

(v) To compute the generalized vertex set V_{k+1} of $S_{k+1} = S_k \cap \{t : \Sigma_{i\in I} t_i\langle a^i, u(t^k)\rangle \leqslant 1/\theta_k\}$, observe that the generalized vertex set V_k of S_k is already known. Therefore, V_{k+1} can be derived from V_k using for example the procedure of Horst–Thoai–de Vries (see [11] or [12]).

(vi) As long as D_k is unchanged, the algorithm is a polyhedral annexation procedure for solving the subproblem (**SP**) with $D = D_k$. As seen in Section 2, this procedure is finite. Hence, after finitely many steps, either D_k changes or the procedure terminates because $R_k = \emptyset$. Since each change of D_k is connected with a decrease of the objective function value at least by $\epsilon > 0$, finiteness of the algorithm is assured.

(vii) In step k.4 the value θ_k may be taken such that $\theta_k u(t^k)$ is arbitrarily close to $Ext(u(t^k))$. This suggests that in practice one could simplify the algorithm by taking $\theta_k u(t^k)$ exactly equal to $Ext(u(t^k))$. With this simplification the algorithm will still be correct, provided in step k.2 all $\bar{t} \in N_k$ with $\mu(\bar{t}) = \beta(\bar{t})$ are retained (not deleted) and in Step k.4 if $\mu(t^k) = 1$ then $u(t^k)$ must be replaced by a basic optimal solution of the problem $\mathbf{LP}(\theta \bar{t}^k)$, with $0 < \theta < 1$ and θ very close to 1.

7. Illustrative Example

For illustration we consider the following example taken from [2] (Example 2):

$$\begin{aligned} (\mathbf{P})\quad \min\quad & -2x_1 + x_2 + 0.5y_1 \\ \text{s.t.}\quad & x_1 + x_2 \leqslant 2 \\ & x_1, x_2 \geqslant 0 \end{aligned}$$

where y solves

$$
\begin{aligned}
(\mathbf{R(x)}) \quad \min \quad & -4y_1 + y_2 \\
\text{s.t.} \quad & -2x_1 + y_1 - y_2 \leqslant -2.5 \\
& x_1 - 3x_2 + y_2 \leqslant 2 \\
& y_1, y_2 \geqslant 0 .
\end{aligned}
$$

I. Preliminary transformations

- Solve the linear program

$$
\begin{aligned}
\min \quad & -2x_1 + x_2 + 0.5y_1 \\
\text{s.t.} \quad & -2x_1 + y_1 + y_2 \leqslant -2.5 \\
& x_1 - 3x_2 + y_2 \leqslant 2 \\
& x_1 + x_2 \leqslant 2 \\
& x_1, x_2, y_1, y_2 \geqslant 0 .
\end{aligned}
$$

A basic optimal solution of this linear program is

$$x^0 = (2, 0), \quad y^0 = (0, 0)$$

with $\varphi(x^0) = \min\{-4y_1 + y_2\colon -y_1 + y_2 \geqslant -1.5;\ -y_2 \geqslant 0;\ y_1 y_2 \leqslant 0\} = -6 < d_2^T y = 0.$

- Write the problem in the form $(\tilde{\mathbf{Q}})$:
Slack variables: s_1, s_2, s_3. Basic variables: x_1, x_2, s_1,
Setting $y_1 = u_1$, $y_2 = u_2$, $s_2 = u_3$, $s_3 = u_4$, we have

$$
\begin{array}{lllll}
x_1 = 2 & & -0.25u_2 & -0.25u_3 & -0.75u_4 \\
x_2 = 0 & & +0.25u_2 & +0.25u_3 & -0.25u_4 \\
s_1 = 1.5 & -u_1 & +0.5u_2 & -0.5u_3 & -1.5u_4
\end{array}
$$

and so $x = x^0 + \xi u$, $y = y^0 + \eta u$ with

$$\xi = \begin{pmatrix} 0 & -0.25 & -0.25 & -0.75 \\ 0 & 0.25 & 0.25 & -0.25 \end{pmatrix}, \quad \eta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The problems becomes

$$(\tilde{\mathbf{Q}}) \quad \min l(u) \quad \text{s.t.} \quad u \in D , \quad f(u) = 0 ,$$

where

$$
\begin{aligned}
& l(u) = -4 + 0.5u_1 + 0.75u_2 + 0.75u_3 + 1.25u_4 \\
D = \quad & \{u\colon\ 0.25u_2 + 0.25u_3 + 0.75u_4 \leqslant 2 \\
& \qquad - 0.25u_2 - 0.25u_3 + 0.25u_4 \leqslant 0 \\
& u_1 \quad - 0.5u_2 + 0.5u_3 + 1.5u_4 \leqslant 1.5
\end{aligned}
$$

$$u = (u_1, u_2, u_3, u_4) \geq 0\}$$

$$f(u) = \varphi(u) + 4u_1 - u_2 .$$

Also

$$\varphi(u) = \{-4u_1 + u_2 \colon \quad \begin{array}{l} 0.5u_2 + 0.5u_3 + 1.5u_4 + y_1 + y_2 \leq 1.5 \\ -u_2 \quad -u_3 \qquad\qquad\quad + y_2 \leq 0 \end{array}$$

$$y_1, y_2 \geq 0\}$$

$$a^0 = (4, -1, 0, 0) \quad a^1 = (0, 0.5, 0.5, 1.5),$$

$$a^2 = (0, -1, -1, 0)$$

We now solve the problem within tolerance $\epsilon = 0.1$.

II. Initialization

Since a^0, a^1, a^2 are linearly independent, we take

$$S_1 = \{t = (t_0, t_1, t_2) : \quad t_i \geq 0 \quad (i = 0, 1, 2)\} .$$

Then $V_1 = \{(1, 0, 0)^*;\ (0, 1, 0)^*;\ (0, 0, 1)^*\}$, where the asterisk indicates an extreme direction

III. Iteration 1

$$D_1 = D , \quad N_1 = \{(1, 0, 0) ; \quad (0, 1, 0)^* ; \quad (0, 0, 1)^*\} .$$

Step 1.1.
For each $\bar{t} \in N_1$ solve $\mathbf{LP}(\bar{t})$ where

$$LP(\bar{t}): \quad \max_{u \in D_1} \{\bar{t}_0(4u_1 - u_2) + \bar{t}_1(0.5u_2 + 0.5u_3 + 1.5u_4) + \bar{t}_2(-u_2 - u_3)\} .$$

This yields:

$\bar{t}$:	$(1, 0, 0)^*$	$(0, 1, 0)^*$	$(0, 0, 1)^*$
$u(\bar{t})$:	$(5.5, 8, 0, 0)$	$(0, 1.5, 0, 1.5)$	$(0, 0, 0, 0)$
$\mu(\bar{t})$:	14	3	0

Step 1.2
$\bar{t} = (0, 1, 0)^*$ is deleted. $R_1 = \{(1, 0, 0),\ (0, 1, 0)\}$.

Step 1.3.
For $\bar{t} = (0, 1, 0)^*$, $f(u(\bar{t})) = 0$, hence $u(\bar{t}) = (0, 1.5, 0, 1.5)$ is feasible with $l(u(\bar{t})) = -1$.

A simplex pivot (for minimizing $l(u)$ over D) performed from $u(\bar{t})$ yields $u^1 = (1.5, 0, 0, 0)$ with $f(u^1) = 0$ and $l(u^1) = -3.25$. Thus, the current best solution is $u^1 = (1.5, 0, 0, 0)$, which corresponds to $x^1 = (2, 0)$, $y^1 = (1.5, 0)$. (Note that since $\min\{l(u):\ u \in D\} = -4$ the optimal value of **(Q)** lies in the interval $(-4, -3.25]$).

Define

$$D_2 = D_1 \cap \{u:\ 0.5u_1 + 0.75u_2 + 0.75u_3 + 1.25u_4 \leq 0.65\}\ .$$

Step 1.4
Select $t^1 = (1, 0, 0)^* \in \arg\max\ \{\mu(t):\ t \in N_1\}$. We have $u(t^1) = (5.5, 8, 0, 0)$ and $Ext(u(t^1)) = \theta_1 u(t^1)$ with $\theta_1 = \frac{15}{4}$, so

$$S_2 = S_1 \cap \{t:\ 7t_0 + 2t_1 - 4t_2 \leq \tfrac{4}{15}\}\ ,$$

$$V_2 = \{(\tfrac{2}{105}, 0, 0);\ (\tfrac{1}{15}, 1, 0);\ (0, 0, 1)^*;\ (0, 2, 1)^*;\ (4, 0, 7)^*\}\ ,$$

$$N_2 = \{(\tfrac{2}{105}, 0, 0);\ (\tfrac{1}{15}, 1, 0);\ (0, 2, 1)^*;\ (4, 0, 7)^*\}\ .$$

IV. Iteration 2

Step 2.1.
For each $\bar{t} \in N_2$ solve

$$LP(\bar{t}):\ \max_{u \in D_2}\{\bar{t}_0(4u_1 - u_2) + \bar{t}_1(0.5u_2 + 0.5u_3 + 1.5u_4) + \bar{t}_2(-u_2 - u_3)\}\ .$$

This yields for the vertices in the list N_2:

$$\mu(\bar{t}) \leq \tfrac{2}{105} \times 4 \text{ and } \mu(\bar{t}) \leq \tfrac{1}{15} \times 3\ .$$

(see Step 1.1). Hence $\mu(\bar{t}) < 1 = \beta(\bar{t})$ for these $\bar{t}$'s.

For the other elements of N_2 we have

$\bar{t}$	:	$(0, 2, 1)^*$	$(4, 0, 7)^*$
$u(\bar{t})$	:	$(0, 0.325, 0, 0.325)$	$(1.3, 0, 0, 0)$
$\mu(\bar{t})$	:	0.975	20.8
$f(u(\bar{t}))$	:	-5.025	-0.8

Step 2.2.
The two vertices in the list N_2 are deleted. So

$$R_2 = \{(0, 2, 1)^*,\quad (4, 0, 7)^*\}\ .$$

Step 2.3.
Since $f(u(\bar{t})) < 0$ for all $\bar{t} \in R_2$, we let $D_3 = D_2$, $u^2 = u^1$.

Step 2.4.
Select $t^2 = (4, 0, 7)^*$. Then $Ext(u(t^2)) = \theta_2 u(t^2)$ with $\theta = 60/52$,

$$S_3 = S_2 \cap \{t:\ 6t_0 \leq 1\}$$

$$V_3 = \{(\tfrac{2}{105}, 0, 0);\ (\tfrac{1}{15}, 1, 0);\ (\tfrac{1}{6}, 0, \tfrac{7}{24});\ (\tfrac{1}{6}, 0, \tfrac{31}{120});\ (\tfrac{1}{6}, \tfrac{1}{15}, \tfrac{7}{24});\ (0,0,1)^*;\ (0,2,1)^*\}$$

$$N_3 = \{(\tfrac{1}{6}, 0, \tfrac{7}{24});\ (\tfrac{1}{6}, 0, \tfrac{31}{120});\ (\tfrac{1}{6}, \tfrac{1}{15}, \tfrac{7}{24})\}\ .$$

V. Iteration 3

Step 3.1.
Solving $LP(\bar{t})$ for each $\bar{t} \in N_3$ we obtain $\mu(\bar{t}) = 13/15$ for the first element of N_3 and $\mu(\bar{t}) < 1$ for the two other elements.

Step 3.2.
All the elements of N_3 are deleted. So

$$R_3 = \{(0, 2, 1)^*\}\ .$$

Step 3.3.
$D_4 = D_3,\ u^3 = u^2\ .$

Step 3.4.
$t^3 = (0, 2, 1)^*$ with $u(t^3) = (0, 0.325, 0, 0.325)$ (see step 2.1). We have $Ext(u(t^3) = \theta_3 u(t^3)$ with $\theta_3 = 60/13$, therefore

$$S_4 = S_3 \cap \{t:\ -t_0 + 2t_1 - t_2 \leq \tfrac{4}{3}\}$$

$$V_4 = V_3 \backslash \{(0, 2, 1)^*\} \cup N_4$$

$$N_4 = \{(0, \tfrac{4}{9}, \tfrac{2}{9});\ (\tfrac{2}{105}, \tfrac{32}{35}, \tfrac{16}{35});\ (0, \tfrac{19}{45}, \tfrac{8}{45});\ (\tfrac{1}{12}, \tfrac{3}{4}, \tfrac{2}{3});\ (\tfrac{1}{6}, \tfrac{131}{180}, \tfrac{28}{45});\ (\tfrac{1}{6}, \tfrac{41}{20}, \tfrac{77}{60});\ (0, 1, 2)^*\}\ .$$

VI. Iteration 4

Step 4.1.
Solving $LP(\bar{t})$ for each $\bar{t} \in N_4$ yields $\mu(\bar{t}) < 1$ for every vertex and $\mu(\bar{t}) = 0$ for the extreme direction in this list.

Step 4.2.
All the elements of N_4 are deleted. Hence, $R_4 = \emptyset$ and we conclude that $u^3 = (1.5, 0, 0, 0)$ solves $(\tilde{Q})$, i.e., a (global) optimal solution to (P), within tolerance 0.1 is

$$\bar{x} = (2, 0)\ , \quad \bar{y} = (1.5, 0)$$

with objective function value -3.25. From the computations, it can even be checked that this is actually an exact optimal solution. (Incidentally, we see that the solution $x = (1, 0)$, $y = (0.5, 1)$ indicated in ([2]), with objective function value -1.75, is not an optimal one).

Also note than an optimal solution has already been found at the second iteration but the algorithm has to go through two more iterations to prove the optimality of this solution.

Computational results with this algorithm as well as comparisons with an alternative branch and bound algorithm will be discussed in a subsequent paper.

Acknowledgement

Research supported in part by the Swedish Transport Research Board (TFB).

References

1. Al-Khayyal, F. A., R. Horst, and P. M. Pardalos (1991), Global Optimization of Concave Functions Subject to Quadratic Constraints Is in an Application in Nonlinear Bilevel Programming, forthcoming in *Anals of Oper. Res.*
2. Bard, J. F. and J. E. Falk (1982), An Explicit Solution to the Multi-Level Programming Problem, *Comput. & Ops. Res.* **9**(1), 77–100.
3. Bard, J. F. (1983), An Algorithm for Solving the General Bi-Level Programming Problem, *Math. of Ops. Res.* **8**(2), 260–272.
4. Moore, J. T. and J. F. Bard (1990), The Mixed Integer Linear Bilevel Programming Problem, *Ops. Res.* **38**(5), 911–921.
5. Bialas, W. F. and M. H. Karwan (1982), On Two-Level Optimization, *IEEE Trans. Auto. Cont.* AC-**27**(1), 211–214.
6. Blundet, W. R. and J. A. Black (1984), *The Land-Use/Transport System*, 2nd Edition, Pergamon Press, Australia.
7. Candler, W. and R. Townsley (1982), A Linear Two-Level Programming Problem, *Comput. & Ops. Res.* **9**(1), 59–76.
8. Fortuny-Amat, J. and B. McCarl (1981), A Representation of a Two-Level Programming Problem, *J. Ops. Res. Soc.* **32**, 783–792.
9. Hansen, P., B. Jaumard, and G. Savard (1990), New Branching and Bounding Rules for Linear Bilievel Programming, forthcoming in *SIAM Journal on Scientific and Statistical Computing.*
10. Hillestad, R. J. and S. E. Jacobsen (1980), Linear Programs with an Additional Reverse Convex Constraint, *Applied Mathematics and Optimization* **6**, 257–269.
11. Horst, R., N. V. Thoai, and J. de Vries (1988), On Finding New Vertices and Redundant Constraints in Cutting Plane Algorithms for Global Optimization, *Operations Research Letters* **7**, 85–90.
12. Horst, R. and H. Tuy (1990), *Global Optimization: Deterministic Approach*, Springer Verlag, Berlin.
13. Kolstad, C. D. and L. S. Lasdon (1986), *Derivative Evaluation and Computational Experience with Large Bi-level Mathematical Programs*, Faculty Working Paper No. 1266, College of Commerce and Business Administration, University of Illinois at Urbana-Champaign.
14. Loridan, P. and J. Morgan (1989), New Results of Approximate Solutions in Two-Level Optimization, *Optimization* **20**(6), 819–836.
15. Muu, L. D. (1985), A Convergent Algorithm for Solving Linear Programs with an Additional Reverse Convex Constraint, *Kybernetica* **21**, 428–435.
16. Rockafellar, R. T. (1970), *Convex Analysis*, Princeton Univ. Press.
17. von Stackelberg, H. (1952), *The Theory of the Market Economy*, William Hodge and Company Limited, London.
18. Thuong, N. V. and H. Tuy (1984), A Finite Algorithm for Solving Linear Programs with an Additional Reverse Convex Constraint, *Lecture Notes in Economics and Mathematical Systems* **225**, 291–302.

19. Tuy, H. (1990), Polyhedral Annexation Method for Concave Minimization, in Leifman, Lev. J. and J. B. Rosen (eds.), *Functional Analysis, Optimization and Mathematical Economics*, Oxford University Press, New York, 248–260.
20. Tuy, H. (1991), Effect of Subdivision Strategy on Convergence and Efficiency of Some Global Optimization Algorithms, *Journal of Global Optimization* **1**(1), 23–36.
21. Tuy, H. (1991), Polyhedral Annexation, Dualization and Dimension Reduction Technique in Global Optimization, *Journal of Global Optimization* **1**(3), 229–244.
22. Ünlü, G. (1987), A Linear Bi-Level Programming Algorithm Based on Bicriteria Programming, *Comput. Ops. Res.* **14**(2), 173–179.
23. Wen, U.-P. and S.-T. Hsu (1989), A Note on a Linear Bilevel Programming Algorithm Based on Bicriteria Programming, *Comput. & Ops. Res.* **16**(1), 79–83.