ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/248018135

The solution of the linear bilevel programming problem by using the linear complementarity problem

Article

CITATIONS
25

READS
22

2 authors, including:

Ana M. Faustino
University of Porto

16 PUBLICATIONS **234** CITATIONS

SEE PROFILE

Available from: Ana M. Faustino
Retrieved on: 28 July 2016

Investigação Operacional, Vol. 8, Nº 1 (Junho/1988) 77-95

THE SOLUTION OF THE LINEAR BILEVEL PROGRAMMING PROBLEM BY USING THE LINEAR COMPLEMENTARITY PROBLEM

J. J. Júdice*
A. M. Faustino**

*Dept. de Matemática , Universidade de Coimbra
**Dept. de Matemática, Universidade do Porto

Resumo
O Problema Linear de 2 níveis (LBLP) é um problema de optimização não convexa que aparece muito frequentemente em aplicações. Bialas e Karwan desenvolveram um algoritmo SLCP para a solução do LBLP que consiste em resolver uma sucessão de Problemas Lineares Complementares. Contudo, o algoritmo não é sempre convergente. Neste artigo é proposta uma versão modificada e convergente desse algoritmo SCLP. Experiência computacional com o algoritmo SCLP modificado na solução de LBLPs de média dimensão e estrutura esparsa é também incluída e mostra que o processo é eficiente.

Abstract
The Linear Bilevel Programming Problem (LBLP) is a nonconvex optimization problem that occurs quite frequently in applications. Bialas and Karwan have designed a SLCP algorithm for the solution of the LBLP, which consists of solving a sequence of linear complementarity problems. However, the method is not convergent in general. In this paper we describe a modified convergent version of the SLCP algorithm. Computational experience with the modified SLCP algorithm on the solution of medium-scale LBLPs is also included and shows the efficiency of the approach.

Keywords: Bilevel Programming, Nonconvex Optimization, Linear Complementarity Problem, SLCP algorithms, sparse matrices.

1 - INTRODUCTION

The Bilevel Programming Problem (BLP) can be defined as

$$\left.\begin{array}{l} \text{Minimize } f_1(x,y) \\ \quad y \in R^m \\ \text{subject to } g_i(x,y) \leq 0, \quad i = 1,\ldots,\ell \end{array}\right\} \qquad (1)$$

where $x \in R^n$ is implicitly defined as the optimal solution of the program

$$\left.\begin{array}{l} \text{Minimize } f_2(x,y) \\ \quad x \in R^n \\ \text{subject to } h_i(x,y) \leq 0, \quad i = 1,\ldots,r \end{array}\right\} \qquad (2)$$

and f_1, f_2, g_i and h_i are real functions of the variables x_j and y_j.

The BLP is nowadays a well-known nonconvex optimization problem. A large number of applications of the BLP have been proposed, primarily in economic planning [6, 11, 12]. A number of algorithms [2,3,4,5,6,7,9,12] have been developed for finding a global or a local minimum for the BLP.

In [7], Bialas and Karwan study the linear BLP (LBLP), that is, the BLP in which all the functions are linear. They discuss a number of algorithms and recommend a Sequential Linear Complementarity Problem (SLCP) algorithm for the solution of LBLPs of small dimension ($n \leq 50$, $m \leq 50$). This algorithm consists of solving a sequence of linear complementarity problems (LCP) by using a Restricted Basis Entry Simplex method similar to the Wolfe-Markowitz method [13,17] for quadratic programming. However, such scheme is not convergent in general [5].

In this paper we describe a modification of the SLCP algorithm capable of finding a global minimum for the LBLP. In this procedure, the LCPs required by the sequential algorithm are solved by an extension of Al-Khayyal's enumerative method [1]. We show that the computational effort of this hybrid enumerative method is highly reduced because of the special structure of the LCPs. We also present some computational experience with medium-scale LBLPs ($n \leq 150$ and $m \leq 250$) which shows the efficiency of the modified SLCP algorithm.

The organization of the paper is as follows. In Section 2 Bialas and Karwan SLCP algorithm for the LBLP is described. In Section 3 the incorporation of the hybrid enumerative method in the SLCP algorithm is discussed. Finally computational experience on the solution of medium-scale LBLPs by the modified SLCP method is presented in Section 4.

2 - SLCP ALGORITHM FOR THE LBLP

Consider the LBLP as stated in [7]

$$\left.\begin{array}{l} \text{Minimize } c^T x + d^T y \\ y \in R_+^m \end{array}\right\} \tag{3}$$

where $R_+^m = \{ y \in R^m : y \geq 0 \}$ and $x \in R^n$ solves the linear program

$$\left.\begin{array}{l} \text{Minimize } a^T x \\ x \in R^n \\ \text{subject to } A_1 x + A_2 y \geq b, x \geq 0 \end{array}\right\} \tag{4}$$

where $A_1 \in R^{r \times n}$ and $A_2 \in R^{r \times m}$. As in [7], we assume that the set

$$H = \{(x,y) \in R^{n \times m} : A_1 x + A_2 y \geq b, x \geq 0, y \geq 0\} \tag{5}$$

is bounded, so that the LBLP has an optimal solution [7].

Consider the dual of the linear program (4)

$$\left.\begin{array}{ll} \text{Maximize} & (b - A_2 y)^T u \\ \text{subject to} & A_1^T u \leq a \\ & u \geq 0 \end{array}\right\} \tag{6}$$

If α and β are the vectors of the slack variables corresponding to the primal and dual respectively, then by the complementarity slackness property [14, pages 166-167], the LBLP is equivalent to the following nonconvex optimization problem

$$\left.\begin{array}{ll} \text{Minimize} & c^T x + d^T y \\ \text{subject to} & \alpha = -b + A_1 x + A_2 y \\ & \beta = a - A_1^T u \\ & x, y, u, \alpha, \beta \geq 0, \; x^T \beta = u^T \alpha = 0 \end{array}\right\} \tag{7}$$

Hence a global minimum $(\overline{x}, \overline{y})$ of the LBLP can be found by solving the nonconvex program (7). Bialas and Karwan [7] propose a SLCP algorithm for such purpose. In this method a parameter λ is introduced and the objective function is replaced by the constraint $c^T x + d^T y \leq \lambda$ to obtain the following parametric LCP

LCP(λ):

$$\left.\begin{array}{c} \begin{bmatrix} \alpha \\ \beta \\ v_0 \end{bmatrix} = \begin{bmatrix} -b \\ a \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \lambda + \begin{bmatrix} 0 & A_1 & A_2 \\ -A_1^T & 0 & 0 \\ 0 & -c^T & -d^T \end{bmatrix} \begin{bmatrix} u \\ x \\ y \end{bmatrix} \\ \\ x, y, u, \alpha, \beta, v_0 \geq 0, \; x^T \beta = u^T \alpha = 0 \end{array}\right\} \tag{8}$$

The SLCP algorithm consists of finding the solution of the LCP($\overline{\lambda}$), where $\overline{\lambda}$ is the smallest value of λ such that LCP(λ) has a solution. To achieve this, the method solves a sequence of LCP(λ_k), where $\{\lambda_k\}$ is a decreasing sequence defined by

$$\left.\begin{array}{l}\lambda_0 = \text{upper-bound of } c^Tx+d^Ty \text{ on } H\\ \lambda_k = c^Tx^{k-1}+d^Ty^{k-1} - |\gamma_k(c^Tx^{k-1}+d^Ty^{k-1})|\end{array}\right\} \quad (9)$$

with (x^{k-1}, y^{k-1}) the solution of LCP(λ_{k-1}) and γ_k a small positive number. The method terminates in a iteration k such that LCP(λ_k) has no solution. When this occurs, the solution (x^{k-1}, y^{k-1}) of the LCP(λ_{k-1}) satisfies

$$0 \le c^Tx^{k-1}+d^Ty^{k-1} - VAL \le |\gamma_k(c^Tx^{k-1}+d^Ty^{k-1})| \quad (10)$$

where VAL is the value of the objective function at the optimal solution. Hence, if the set H defined by (5) is nonempty and bounded, the algorithm finds an ε-optimal solution of the LBLP, where

$$\varepsilon = |\gamma_k(c^Tx^{k-1}+d^Ty^{k-1})| \quad (11)$$

In practice, if γ_k is quite small, the solution (x^{k-1}, y^{k-1}) of the last LCP(λ_{k-1}) is usually the optimal solution of the LBLP. Bialas and Karwan [7] recommend the choice of γ_k = 0.01 in each iteration k and our experience has confirmed that this choice is usually the most suitable.

The steps of the SLCP algorithm are presented below.

SLCP Algorithm

STEP 0 - Let k = 0 and λ_0 be an upper-bound of c^Tx+d^Ty on H.

General Step - Solve the LCP(λ_k). If LCP(λ_k) has no solution, go to Exit.

Otherwise let (x^k, y^k) be the solution of this LCP. Set

$$\lambda_{k+1} = c^Tx^k+d^Ty^k - |\gamma_k(c^T x^k+d^T y^k)| \quad (12)$$

where γ_{k+1} is a fixed value.

Set k = k + 1 and repeat.

EXIT - If k = 0 the LBLP is infeasible. Otherwise (x^{k-1}, y^{k-1}) is an ε-optimal solution of the LBLP, where ε is given by (11).

The efficiency of the SLCP algorithm depends essentially on the procedure to solve the LCP(λ_k). Bialas and Karwan [7] recommend the use of a Basis Restricted Entry Simplex (BRES) method for this purpose. They do not provide sufficient details for the complete understanding of the algorithm, but this kind of method can not always be used to process the LCP(λ_k) [5]. Next we describe a BRES algorithm with exactly one artificial variable. Consider the LCP(λ_k) in the following form

$$\left.\begin{array}{l}w = q+Mz+Ny\ ,\ w \ge 0, z \ge 0, y \ge 0\\ z_iw_i = 0,\ i = 1,\dots,r+n\end{array}\right\} \quad (13)$$

where

$$w = \begin{bmatrix}\alpha\\ \beta\\ v_0\end{bmatrix} \in R^{r+n+1}\ ,\ z = \begin{bmatrix}u\\ x\end{bmatrix} \in R^{r+n}\ ,\ q = \begin{bmatrix}-b\\ a\\ \lambda_k\end{bmatrix} \in R^{r+n+1}$$

$$M = \begin{bmatrix}0 & A_1\\ -A_1^T & 0\\ 0 & -c^T\end{bmatrix} \in R^{(r+n+1)\times(r+n)}\ ,\ N = \begin{bmatrix}A_2\\ 0\\ -d^T\end{bmatrix} \in R^{(r+n+1)\times m}$$

Let z_0 be an artificial variable and p be a nonnegative vector satisfying $p_i > 0$ for all i such that $q_i < 0$. Consider the following program

$$\left.\begin{array}{ll}\text{Minimize} & z_0\\ \text{subject to} & w = q+z_0p+Mz+Ny\\ & z, w, y, z_0 \ge 0\\ & z_iw_i = 0,\ i = 1,\dots,r+n\end{array}\right\} \quad (14)$$

The BRES algorithm is an extension of the Phase 1 method with a single artificial variable [14, pages 136-138], and seeks the optimal solution of the program (14) by only using complementary feasible solutions (that is, solutions satisfying the constraints of (14)). To assure this condition, a nonbasic variable z_i (or w_i) with a positive reduced cost coefficient can only be candidate to be an entering variable if its complementary w_i (or z_i) is also nonbasic or becomes nonbasic in this iteration. Because of this additional criterion, the BRES algorithm may have three possible terminations:

TERM = 1 - an optimal solution $(\bar{w}, \bar{z}, \bar{y})$ of (14) with $\bar{z}_0 = 0$.

TERM = 2 - an optimal solution $(\bar{w}, \bar{z}, \bar{y})$ of (14) with $\bar{z}_0 > 0$.

TERM = 3 - a basic solution which is not optimal for (14) but there are no candidates for an entering variable.

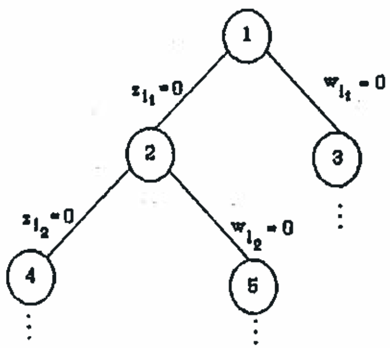
In the first case (TERM = 1) $(\bar{w}, \bar{z}, \bar{y})$ is a solution of the LCP (13). If TERM = 2 this LCP has no solution. In the latter case (TERM = 3) no conclusion can be made about the existence of a solution to the LCP(13). In appendix we present an example of a LCP of the form (13) for which there is a complementary solution but the BRES algorithm terminates with TERM = 3. This shows that the BRES method can not be used in general to solve the LCPs(λ_k) that the SLCP algorithm requires. However, the BRES method can be incorporated in a hybrid enumerative algorithm and this usually improves the efficiency of such procedure. This is discussed in the next section.

3 - A HYBRID ENUMERATIVE METHOD FOR THE LCP(λ_k)

In this section we describe a hybrid enumerative method capable of solving any LCP(λ_k) required by the SLCP algorithm. Consider a LCP(λ_k) in the form

$$w = q + Mz + Ny \tag{15}$$
$$w \geq 0, \quad z \geq 0, \quad y \geq 0 \tag{16}$$
$$z_i w_i = 0 \ , \ i = 1, \dots, r+n \tag{17}$$

where $w \in R^{r+n+1}$, $z \in R^{r+n}$ and $y \in R^m$. As in linear programming, a solution (z,w,y) satisfying the linear constraints (15) and (16) is called Feasible. A solution is Complementary if the variables z_i and w_i satisfy (17). An enumerative method attempts to find a complementary solution by only using basic feasible solutions of the system (15). To achieve this, the following tree is explored



(18)

where $i_1, i_2, \dots$ are integer numbers of $\{1, \dots, r+n\}$. In node 1 an initial feasible solution is obtained by a variant of the Phase 1 algorithm with a single artificial variable, as is discussed later. Each one of the other nodes is generated by solving a subproblem, which consists of minimizing a variable z_i or w_i subject to the linear constraints of the LCP and some constraints $z_j = 0$ or $w_j = 0$. For instance, to generate node 4 of the tree (18) it is necessary to solve the linear program

$$\left.\begin{array}{ll} \text{Minimize} & z_{i_2} \\ \text{subject to} & w = q + Mz + Ny, \ z \geq 0, \ w \geq 0, y \geq 0, \\ & z_{i_1} = 0 \end{array}\right\} \tag{19}$$

Such linear program is solved by a modification of the Phase 2 of the simplex method and two cases may occur:

(i) if the variable minimized has value equal to zero, then it is fixed at zero in all descendent paths of the tree (the variable is said to be starred).

(ii) if the minimum value of the variable is positive, then the branch is pruned and the node is fathomed.

The enumerative method attempts to solve the LCP by generating sucessive nodes of the tree according to the process explained above. The algorithm either finds a solution of the LCP (it is the first complementary feasible solution) or establishes that the LCP has no solutions (all the nodes of the tree are fathomed).

The enumerative method is efficient if there are only few nodes to be visited before a complementary solution is found or it is verified that none exists. There exist some heuristic rules and some procedures that usually improve the efficiency of the algorithm. The heuristic rules are related with the choice of the node of the tree and of the pair (z_{i_k}, w_{i_k}) and are presented later. Next we describe the extensions for the LCP (13) of two efficient algorithms [10] that are usually incorporated in the enumerative method [10].

(I) A MODIFIED BRES ALGORITHM

Consider the problem of generating the node 1 of the tree (18), that is, finding a first feasible solution for the LCP (13). As stated before, this can be achieved by solving the linear program

$$\left.\begin{array}{ll} \text{Minimize} & z_0 \\ \text{subject to} & w = q + z_0 p - Mz + Ny \\ & z, w, y, z_0 \geq 0 \end{array}\right\} \quad (20)$$

where p is a nonnegative vector satisfying $p_i > 0$ for all i such that $q_i < 0$ and z_0 is an artificial variable. Since our ultimate goal is to find a complementary feasible solution, then the BRES algorithm can be used to solve such linear program. If this algorithm terminates with TERM = 1 or TERM = 2, then a solution of the LCP (13) has been found or LCP has no solutions respectively, and the enumerative method stops. If TERM = 3, the value of z_0 can still be reduced if a simplex pivot step is performed, where the pivot belongs to a column with a positive reduced cost coefficient. The modified BRES algorithm consists of performing such pivot step when TERM = 3 and reapplying the BRES method by starting with the basic solution obtained by this pivot step. The process is then repeated. This algorithm can also be used to generate any other node k, since, as discussed before, the generation of a node consists of minimizing a variable z_i or w_i.

To describe the modified BRES algorithm, we associate to each basic solution a quantity NCP defined as the number of pairs of complementary variables (z_i, w_i) such that z_i and w_i are both basic in this solution. If NCP = 0 the basic solution is complementary. The converse is true for nondegenerate basic solutions, but a degenerate basic solution may be complementary with NCP > 0. Hence the value of NCP is increased by one whenever TERM = 3 occurs in the BRES algorithm. In addition, NCP never increases and may even be reduced during the BRES procedure. Hence a complementary feasible solution may occur during this algorithm. The steps of the modified BRES algorithm are presented below.

A Modified BRES Algorithm for Generating a node k

STEP 0 - Let NCP be the number of pairs of complementary variables (z_i, w_i) such that z_i and w_i are both basic in the basic solution (NCP = 0 if k = 1).

STEP 1 - Apply the BRES algorithm. If a complementary feasible solution is found when the BRES algorithm is in use, set TERM = 1 and NCP = 0 and go to Exit. If the BRES algorithm terminates with TERM = 1 or TERM = 2, go to Exit. Otherwise (TERM = 3) go to step 2.

STEP 2 - Set NCP = NCP + 1 and perform a simplex pivot step, where the pivot belongs to a column with a positive reduced cost coefficient. Go to step 1.

EXIT - If TERM = 1 and NCP = 0, a solution of the LCP has been found and the enumerative method stops. If TERM = 1 and NCP > 0, the node k has been generated. If TERM = 2, the node k can not be generated and is fathomed.

(II) A MODIFIED REDUCED-GRADIENT (MRG) ALGORITHM

This algorithm is described in [1] and consists of finding a local star minimum of the function

$$f(z,w,y) = \sum_{i=1}^{r+n} z_i w_i \quad (21)$$

that is, a basic feasible solution $(\bar{z}, \bar{w}, \bar{y})$ such that

$$f(\bar{z}, \bar{w}, \bar{y}) \leq f(z,w,y) \quad (22)$$

for all its adjacent basic feasible solutions (z,w,y). To describe the algorithm for the case of the LCP(13), consider a basic feasible solution $(\bar{z}, \bar{w}, \bar{y})$ with Basis B. If $(\tilde{z}, \tilde{w}, \tilde{y})$ is an adjacent basic solution of $(\bar{z}, \bar{w}, \bar{y})$, then

$$(\tilde{z}, \tilde{w}, \tilde{y}) = (\bar{z}, \bar{w}, \bar{y}) + \mu_0 \, (d^z, d^w, d^y) \quad (23)$$

where μ_0 is the minimum ratio of the simplex method and $d = (d^z, d^w, d^y)$ is a feasible direction such that d^z, d^w and d^y are the vectors of the components of d corresponding to the variables z,w and y respectively. To define this direction, let F and T be the sets of the indices of the basic and nonbasic variables respectively and s be the index of the nonbasic entering variable that is increased to generate the basic feasible solution $(\tilde{z}, \tilde{w}, \tilde{y})$. Then

$$d_s = 1$$
$$d_j = 0 \quad \text{for all } j \in T - \{s\}$$

$$d_F = \begin{cases} -B^{-1}M_{.s} & \text{if } s \text{ is a column of a } z \text{ variable} \\ B^{-1}I_{.s} & \text{if } s \text{ is a column of a } w \text{ variable} \\ -B^{-1}N_{.s} & \text{if } s \text{ is a column of a } y \text{ variable} \end{cases}$$

where $M_{.s}$, $N_{.s}$ and $I_{.s}$ represent the column s of the matrices M,N and identity matrix I respectively. By the structures of the matrices M and N of the LCP(13), we can write

$$P\,[I \;\; -M \;\; -N]\,Q = \left[\begin{array}{c|c} E & 0 \\ \hline 0 & G \end{array}\right] \tag{24}$$

where P and Q are permutation matrices and $E \in R^{(r+1)\times(r+n+m+1)}$, $G \in R^{n\times(r+n)}$ are matrices such that E contains a column of the variable $z_i(w_i)$ if and only if G contains a column of the variable $w_i(z_i)$. Hence

$$PBQ = \left[\begin{array}{c|c} B_1 & 0 \\ \hline 0 & B_2 \end{array}\right] \tag{25}$$

where $B_1 \in R^{(r+1)\times(r+1)}$ and $B_2 \in R^{n\times n}$ are nonsingular matrices, and this implies that

$$d = \begin{bmatrix} 0 \\ \hline 1 \\ B_2^{-1}G_{.s} \end{bmatrix} \quad \text{or} \quad d = \begin{bmatrix} 1 \\ B_1^{-1}E_{.s} \\ \hline 0 \end{bmatrix} \tag{26}$$

Therefore

$$d_i^w = 0 \quad \text{or} \quad d_i^z = 0 \quad \text{for each } i = 1,\dots,r+n \tag{27}$$

Computing the value of the function (21) at the adjacent basic feasible solution given by (23), we have

$$f(\hat z,\hat w,\hat y) = \sum_{i=1}^{r+n} \hat z_i \hat w_i = \sum_{i=1}^{r+n} (\bar z_i + \mu_0 d_i^z)(\bar w_i + \mu_0 d_i^w)$$

$$= \sum_{i=1}^{r+n} \bar z_i \bar w_i + \mu_0 \sum_{i=1}^{r+n} (\bar z_i d_i^w + \bar w_i d_i^z) + \mu_0^2 \sum_{i=1}^{r+n} d_i^z d_i^w$$

Hence by (27) we have

$$f(\hat z,\hat w,\hat y) - f(\bar z, \bar w, \bar y) = \mu_0 \sum_{i=1}^{r+n} (\bar z_i d_i^w + \bar w_i d_i^z) \tag{28}$$

But

$$\sum_{i=1}^{r+n} (\bar z_i d_i^w + \bar w_i d_i^z)$$

is exactly the symmetric of the reduced-cost coefficient $\bar e_s$ of the nonbasic variable of index s associated with the linear function

$$\sum_{i=1}^{r+n} (\bar w_i z_i + \bar z_i w_i) \tag{29}$$

Therefore we have shown that for the LCP(λ_k), Al-Khayyal's MRG Algorithm reduces to a simplex-type method in which the reduced-cost coefficients $\bar e_j$ of the nonbasic variables are associated with the linear function (29). The algorithm also incorporates a modification of Bland's rule [8] to avoid the occurrence of cycling and is presented below.

MRG Algorithm

STEP 0 - Let $(\bar z, \bar w, \bar y)$ be a basic feasible solution with Basis $\bar B$. Set deg = 0.

STEP 1 - Compute the reduced-cost coefficients $\bar e_j$ associated with the linear function (29). If $\bar e_j \le 0$ for all j, then $(\bar z, \bar w, \bar y)$ is a local star minimum and stop. Otherwise go to step 2.

STEP 2 - Find the entering variable as the nonbasic variable in the column s given by

$$s = \begin{cases} \min\{i : \bar e_i = \max \bar e_j\} & \text{if deg} = 0 \\ \min\{i : \bar e_i > 0\} & \text{if deg} = 1 \end{cases} \tag{30}$$

STEP 3 - Compute the minimum ratio μ_o for this entering variable and let the leaving variable be the first basic variable at which μ_o is attained. If $\mu_o = 0$ ($\mu_o > 0$) set deg = 1 (deg = 0). Perform a simplex pivot step and obtain a basic feasible solution $(\bar{z}, \bar{w}, \bar{y})$ with Basis B. Go to step 1.

As described above, the MRG algorithm always terminates in a local star minimum $(\bar{z}, \bar{w}, \bar{y})$ of the function f(z,w,y) given by (21). The method is quite suitable to be incorporated in a hybrid enumerative method, since the value of the complementarity function $\sum_{i=1}^{r+n} z_i w_i$ is usually reduced and may take the value zero, in which case a solution of the LCP(13) is found.

The hybrid enumerative method, incorporating the two procedures referred in this section and the most efficient heuristic rules, is stated below.

Hybrid Enumerative Method for Solving LCP(13)

STEP 0 - Set NODE = NNODE = 1, where NNODE is the total number of nodes to be investigated and NODE is the current node. Set STZ(NODE) = STW(NODE) = Ø, where STZ(NODE) and STW(NODE) are the sets of the indices of the z and w starred variables at node NODE respectively.

STEP 1 - Generation of node 1 (First feasible solution).

(i) Apply the modified BRES algorithm for solving the linear program (20). If TERM = 1 and NCP = 0, go to Exit. If TERM = 2 go to Exit. Otherwise (TERM = 1 and NCP > 0) go to (ii).

(ii) Apply the MRG algorithm to find a local star minimum $(\bar{z}, \bar{w}, \bar{y})$ of the function f(z,w,y) given by (21). If $f(\bar{z}, \bar{w}, \bar{y}) = 0$, set TERM = 1 and go to Exit. Otherwise set NCP(NODE) = NCP and go to step 2.

STEP 2 - Branching - Set NCP = NCP(NODE), STZ = STZ(NODE) and STW = = STW(NODE). Choose the pair of complementary variables (z_s, w_s) in node NODE such that

$$\bar{z}_s \bar{w}_s = \max \{ \bar{z}_i \bar{w}_i : i = 1,\dots,NCP\} \qquad (31)$$

Fathom node NODE and go to step 3.

STEP 3 - Generate NNODE+1 - Set NODE = NNODE = NNODE + 1.

(i) Apply the modified BRES algorithm for solving the linear program

$$\left.\begin{array}{ll} & \text{Minimize} \quad z_s \\ \text{subject to} & w = q + Mz + Ny, \ z,w,y \geq 0 \\ & z_j = 0, \ w_\ell = 0, \ j \in STZ, \ \ell \in STW \end{array}\right\} \qquad (32)$$

If TERM = 1 and NCP = 0, go to Exit. If TERM = 2, fathom node NODE and go to step 4. Otherwise set STZ(NODE) = STZ ∪ {s} and go to (ii).

(ii) Apply the MRG algorithm to find a local star minimum $(\bar{z}, \bar{w}, \bar{y})$ of the function f(z,w,y) given by (21). If $f(\bar{z}, \bar{w}, \bar{y}) = 0$, set TERM = 1 and go to Exit. Otherwise set NCP (NODE) = NCP, $f(\text{NODE}) = \sum_{i=1}^{NCP} \bar{w}_i \bar{z}_i$ and go to step 4.

STEP 4 - Generate node NNODE + 2 - This is step 3 with z_s, STZ and step 4 replaced by w_s, STW and step 5, respectively.

STEP 5 - Choice of the node - If all the nodes 1 ,..., NNODE are fathomed, set TERM = 2 and go to Exit. Otherwise choose the node NODE by

$$NCP(NODE) + \frac{f(NODE)}{10} = \min \left\{ NCP(j) + \frac{f(j)}{10} : j = 1,\dots, NNODE \text{ and } j \text{ is unfathomed} \right\} \qquad (33)$$

and go to step 2.

EXIT - If TERM = 1, the complementary basic feasible solution $(\bar{z}, \bar{w}, \bar{y})$ is a solution of the LCP(13). If TERM = 2, the LCP(13) has no solution.

We note that (31) and (33) are heuristic rules that we have found efficient to reduce the computational effort of the algorithm [10]. The hybrid enumerative method can be implemented for solving large and sparse LCPs. The implementation uses reinversion and updating techniques for the LU decomposition of the Basis matrices [16] and special data structures for the representation of the sets discussed above. We suggest [10] for a description of this procedure.

As discussed in section 2, the hybrid enumerative method is used to solve the

LCPs (λ_k) required by the SLCP algorithm. For any two values $\lambda_k < \lambda_{k-1}$, the LCP$(\lambda_{k-1})$ and LCP(λ_k) only differ in the . component of the vector q. Furthermore, if B is the Basis associated to the solution $(\bar{z}, \bar{w}, \bar{y})$ of the LCP(λ_{k-1}) obtained by the hybrid enumerative method and q is the right-hand side vector of the LCP(λ_k), then the vector $\bar{q} = B^{-1} q$ satisfies

$$\bar{q}_j \geq 0 \quad \text{for all} \quad j = 1, \ldots, r+n \quad \text{and} \quad \bar{q}_{r+n+1} < 0 \tag{34}$$

Hence the solution $(\bar{z}, \bar{w}, \bar{y})$ of the LCP(λ_{k-1}) can be used as the initial basic solution for the LCP(λ_k). Since the vector $B^{-1}q$ satisfies (34), then only the last component of the vector $B^{-1}p$ has to be positive. Hence we can choose in (20)

$$p = B \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{35}$$

where $0 \in R^{r+n}$ is the null vector. The choice of this initial basic solution usually provides great reductions in the hybrid enumerative method. In fact, the algorithm requires quite often a unique node and a quite small number of pivot steps to find the solution of the LCP(λ_k). This is shown by the computational results presented in the next section.

In the original SLCP algorithm the LCP(λ_o) is solved in step 0, where λ_o is an upper-bound of $c^T x + d^T y$ on H. The computation of this value of λ_o may be expensive in certain cases. Next, we present a modified step 0 of the SLCP algorithm, which does not require the computation of this upper-bound.

Modified Step 0 - Set k = 0 and solve the LCP(0) with v_o unrestricted in sign. If this LCP has no solution go to Exit. Otherwise let (x^o, y^o) be the solution of this LCP and $\lambda_o = c^T x^o + d^T y^o$ be the value of the variable v_o. Compute λ_1 by (12).

Since in this step the variable v_o is unrestricted in sign and is originally basic, then it remains basic during the solution of the LCP(0). Hence this LCP is exactly the LCP(λ_o) without the constraint $c^T x + d^T y \leq \lambda_o$. Since the value of λ_o obtained by the Modified Step 0 is usually smaller than the upper-bound chosen by the original SLCP algorithm, then this modification reduces the number of LCPs to be solved and improves the efficiency of the SLCP algorithm. This is another important reason for the use of the Modified Step 0.

4. COMPUTATIONAL EXPERIENCE

In this section we present some computational experience with the SLCP

algorithm described in this paper on the solution of some medium-scale sparse LBLPs. The matrix $A = [A_1, A_2]$ of the test problems has been constructed by an extension for sparse matrices of the technique described in [15]. This modification consists of introducing a quantity NZ, representing the number of nonzero elements of the matrix A, and randomly distributing these nonzero elements in a scheme similar to [15]. We have also considered in the matrix A a dense row of ones to make bounded the set H defined by (5). We associate to each matrix A two test problems, which differ in the cost coefficients. In the problems denoted by TPN, all the cost coefficients c,d, and a of the two linear functions are nonnegative. In the problems denoted by TPG, the cost coefficients c and d of the function of the first level are nonnegative but the linear function of the second level has some negative cost coefficients. By doing this we simulate LBLPs with conflicting (problems TPG) and nonconflicting (problems TPN) objectives.

The characteristics of the test problems and the results of the experiences of solving these problems by the SCLP algorithm are presented in Table 1. All the tests have been performed on a CDC CYBER 180-830 of the University of Porto. In these tables there are some parameters with the following meanings:

n = number of second level variables x

m = number of first level variables y

r = number of constraints = number of rows of $A = [A_1, A_2]$

rsm = relative sparsity of $A = [A_1, A_2] = \dfrac{\text{number of nonzeros of A}}{n+m}$

nrow = number of rows of the LCP(λ_k) = r + n + 1

ncol = number of columns of the LCP(λ_k) = r + n + m

NLCP = number of LCPs(λ_k) to be solved by the SLCP algorithm.

NI = Total number of simplex pivot steps required by the SLCP algorithm.

NIMIN = minimum number of pivot steps that a LCP(λ_k) has required.

NIMAX = maximum number of pivot steps that a LCP(λ_k) with a solution has required.

T = CPU time in seconds for the SLCP algorithm.

The results presented in Table 1 show that it is worthwhile to solve medium-scale LBLPs by the SLCP algorithm discussed in this paper. As expected the efficiency of the

TP	n	m	r	rsm	nrow	ncol	NLCP	NI	NIMIN	NIMAX	T
TPN1	30	50	30	3.5	61	110	14	54	3	5	.96
TPG1							55	249	3	8	5.5
TPN2				5.0			27	315	3	40	10.3
TPG2							80	475	3	42	15.
TPN3	50	120	50	5.2	101	220	44	230	3	51	9.8
TPG3							124	529	3	7	32.5
TPN4				7.5			120	964	3	13	59.4
TPG4							163	1547	3	97	124.
TPN5	100	300	100	5.3	201	500	139	1009	3	70	107.
TPG5							247	2207	3	203	308.
TPN6				7.1			126	682	3	17	79.9
TPG6							268	1899	3	20	380.
TPN7	150	250	150	5.3	301	550	77	1934	3	115	128.
TPG7							214	3775	3	450	889.
TPN8				7.1			127	4740	3	1365	726.
TPG8							189	9170	3	194	2198.

TABLE 1

SLCP algorithm decreases with an increase of the dimension and density of the LBLP. The method performs much better for the LBLPs with nonconflicting objectives (problems TPN).

In all the experiences presented in Table 1, we have fixed $\gamma_k = 0.01$ in the SLCP algorithm. We have also tested larger values of γ_k for the first LCPs(λ_k). This usually reduces the number of LCPs to be solved by the SLCP algorithm. However, the solutions of these LCPs usually require more computational work and this reduces the overall efficiency of the SLCP algorithm.

The results of the experiences also show that the number of LCPs to be solved by the SLCP algorithm is not too large. Furthermore the values of NIMIN and NIMAX are small and show that the hybrid enumerative method is efficient for solving the LCPs required by the SLCP algorithm. As expected the hybrid enumerative method performs

much worse when the last LCP(λ_k) is solved. Since this LCP has no solution, then the algorithm stops only when all the nodes are fathomed and this requires a great amount of tree search. Our experience has showed that the computational effort for solving this last LCP is similar to the computational work performed by the SLCP algorithm on the solution of all the remaining LCPs for the smaller Bilevel programs and is even bigger for the larger problems.

As a final conclusion, we can claim that the SLCP method described in this paper is an efficient and convergent algorithm for solving the LBLP. The efficiency of the SLCP method can be substantially improved if it is possible to design an efficient procedure to establish that the last LCP(λ_k) has no solution. This last LCP(λ_k) may even be unnecessary if the SLCP algorithm is seen as a process to provide a good incumbent for the branch-and-bound method described in [5]. These are two topics of our current research.

APPENDIX

In this appendix we present an example in which the BRES algorithm fails to find a solution of a LCP(λ_k). Consider the LBLP stated in [7]. Since $x_1 \geq 0$, then 0 is an upper-bound of the objective function $-x_1$ and the LCP(0) is firstly solved. To solve this problem by the BRES algorithm, we consider $p = (1,0,0,0,0,0,0)^T$ and obtain the following initial tableau

BASIC VARIABLES	NONBASIC VARIABLES								BASIC VALUES
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	y_1	
w_1	-1	0	0	0	0	0	-2	-1	-10
w_2	0	0	0	0	0	0	-2	1	6
w_3	0	0	0	0	0	0	-1	2	21
w_4	0	0	0	0	0	0	2	1	38
w_5	0	0	0	0	0	0	2	-1	18
w_6	0	2	2	1	-2	-2	0	0	1
w_7	0	0	0	0	0	0	-1	0	0

A complementary basic feasible solution for the program (14) is obtained by performing the simplex pivot step whose pivot is circled in the tableau. The tableau associated with this basic solution is given below.

	v_1	z_1	z_2	z_3	z_4	z_5	z_6	y_1	
z_0	-1	0	0	0	0	0	2	1	10
v_2	0	0	0	0	0	0	-2	1	6
v_3	0	0	0	0	0	0	-1	2	21
v_4	0	0	0	0	0	0	2	1	38
v_5	0	0	0	0	0	0	2	-1	18
v_6	0	2	2	1	-2	-2	0	0	1
v_7	0	0	0	0	0	0	-1	0	0

Since we wish to minimize the value of the variable z_0, then the variables z_6 and y_1 are elligible to be the entering variable. However z_6 can not be a candidate, since w_6 is basic and does not become nonbasic if z_6 is chosen as entering variable. Hence y_1 must be the entering variable. After finding the pivot (entry circled in the tableau) by the usual minimum ratio criterion and performing the corresponding pivot step, we obtain another complementary basic feasible solution given by the following tableau

	v_1	z_1	z_2	z_3	z_4	z_5	z_6	v_2	
z_0	-1	0	0	0	0	0	4	-1	4
y_1	0	0	0	0	0	0	-2	1	6
v_3	0	0	0	0	0	0	3	-2	9
v_4	0	0	0	0	0	0	4	-1	32
v_5	0	0	0	0	0	0	0	1	24
v_6	0	2	2	1	-2	-2	0	0	1
v_7	0	0	0	0	0	0	-1	0	0

In this iteration only z_6 is elligible to be the entering variable. As before this variable can not be a candidate, whence the termination TERM = 3 occurs. However, the LCP(0) has at least the solution $z = \left(\frac{1}{2}, 0,0,0,0,1\right)$, $y_1 = 8$.

REFERENCES

[1] F.A.Al-Khayyal, An implicit enumeration procedure for the general linear complementarity problem, Mathematical Programming Studies 31 (1987) 1-20.

[2] J.F. Bard, An efficient point algorithm for a linear two-stage optimization problem, Operations Research 31 (1983) 670-684.

[3] J.F. Bard, Convex two-level optimization, Mathematical Programming 40(1988) 15-28.

[4] J.F. Bard and J.E. Falk, An explicit solution to the multi-level programming problems, Computers and Operations Research 9 (1982) 77-100.

[5] J.F. Bard and J.J. Moore, A branch-and-bound algorithm for the Bilevel linear programming problem, Technical Report, Department of Mechanical Engineering, University of Texas at Austin, 1988.

[6] O. Ben-Ayed, C.E. Blair and D. E. Boyce, Solving a real world highway network design problem using bilevel linear programming, BEBR Faculty Working Paper 1463, University of Illinois at Urbana-Champaign, 1988.

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

[8] R.G. Bland, New finite pivoting rules for the simplex method, Mathematics of Operations Research 2 (1977) 103-107.

[9] W. Candler and R. Townsley, A linear two-level programming problem, Computers and Operations Research 9(1982) 59-76.

[10] J.J. Júdice and A.M. Faustino, An experimental investigation of enumerative methods for the linear complementarity problem, Computers and Operations Research 15 (1988) 417-426.

[11] C.D. Kolstad, A review of the literature on Bilevel Mathematical Programming, Technical Report LA-10284-MS, Los Alamos National Laboratory, 1985.

[12] C.D. Kolstad and L.S. Lasdon, Derivative evaluation and computational experience with large Bilevel mathematical programs, Faculty Working Paper 1266, College of Commerce and Business Administration, University of Illinois at Urbana-Champaign, 1986.

[13] H. M. Markowitz, The optimization of a quadratic function subject to linear constraints, Naval Research Logistics Quarterly 3 (1956) 111-133.

[14] K.G. Murty, Linear and Combinatorial Programming, Wiley, 1976.

[15] B. Ramarao and C.M. Shetty, Application of disjunctive programming to the linear complementarity problem, Naval Research Logistics Quarterly 31 (1984) 589-600.

[16] J.K. Reid, A sparsity-exploitation variant of the Bartels-Golub decomposition for linear programming bases, Mathematical Programming 24 (1982) 55-69.

[17] P. Wolfe, The simplex method for quadratic programming, Econometrica 27 (1959) 382-398.