Solving linear bilevel programming by DC algorithm

M. S. Radjef[⋆] and A. Anzi [⋆⋆]

Laboratory of Modelling and Optimization
of systems (LAMOS), Béjaia, Algeria

Abstract. In this paper we propose an algorithm for solving bilevel linear programming problems, in which the second level problem is replaced by its Karush-Kuhn-Tucker optimality conditions. This algorithm is a combination of the DCA algorithm in DC programming and the exact penalty methods.

Keywords : Bilevel linear programming, DC programming, DCA algorithm, KKT optimality conditions, exact penalty.

1 Introduction

Bilevel programming is a tool for modelling two level hierarchical systems. This class of programs constitutes a branch of mathematical programming in which the constraints are, partially, determined by another optimization problem.

Bilevel programming is motivated by the static noncooperative game theory of Stackelberg. In these problems, the upper level is termed Leader and the lower level is termed Follower. The control of variables is partitioned between the decision maker's who attempt to optimize their individual objectives. The Leader goes first in order to optimize his objective function. The Follower observes the Leader's decision and constructs his decision.

Bilevel linear programming (BLP) is one of the basic models of bilevel programming where the objective function and the constraints of the upper level and the lower level problems are all linear. The (BLP) problem can be formulated as follows:

$$\max_x \ F(x,y) = c_1^t x + d_1^t y, \tag{1a}$$

$$s.t. \ \ A_1 x + B_1 y \leq b_1, \tag{1b}$$

$$x \geq 0; \tag{1c}$$

$$\max_y \ f(x,y) = c_2^t x + d_2^t y, \tag{1d}$$

$$s.t. \ \ A_2 x + B_2 y \leq b_2, \tag{1e}$$

$$y \geq 0, \tag{1f}$$

[⋆] LAMOS, University of Béjaia, Algeria.
[⋆⋆] LAMOS, University of Béjaia, Algeria.

where $x,\, c_1,\, c_2 \in \mathbb{R}^{n_1}$; $y,\, d_1,\, d_2 \in \mathbb{R}^{n_2}$; $b_1 \in \mathbb{R}^{m_1}$; $b_2 \in \mathbb{R}^{m_2}$; $A_1 \in \mathbb{R}^{m_1 \times n_1}$; $B_1 \in \mathbb{R}^{m_1 \times n_2}$; $A_2 \in \mathbb{R}^{m_2 \times n_1}$ and $B_2 \in \mathbb{R}^{m_2 \times n_2}$.
Following [9] we give these definitions:

1. *Constraint set of the problem*

$$S = \{(x, y) : A_1 x + B_1 y \le b_1, A_2 x + B_2 y \le b_2, x \ge 0, y \ge 0\}.$$

2. *Feasible set for the Follower for each fixed x*

$$S(x) = \{y \in \mathbb{R}^{n_2} : B_2 y \le b_2 - A_2 x, y \ge 0\}.$$

3. *Projection of S onto the Leader's space*

$$P(X) = \{x \in \mathbb{R}^{n_1} : \exists y \in \mathbb{R}^{n_2}, A_1 x + B_1 y \le b_1, A_2 x + B_2 y \le b_2, x \ge 0, y \ge 0\}.$$

4. *Follower's rational reactions set for $x \in P(X)$*

$$R(x) = \{y \in \mathbb{R}^{n_2} : y = \arg\max[f(x, \hat{y}) : \hat{y} \in S(x)]\}.$$

5. *Inducible region*

$$RI = \{(x, y) \in S,\, y \in R(x)\}.$$

   The inducible region represents the feasible set over which the Leader may optimize his objective.

There are mainly two ways to formulate a (BLP) : the pessimistic formulation and the optimistic one. The formulation considered in this paper is the optimistic formulation. In this case, an optimal solution of the (BLP) is defined as follows:

Definition 1 *[15] A point $(x^*, y^*) \in RI$ is an optimal solution of problem (1) if*

$$c_1^t x^* + d_1^t y^* \ge c_1^t x + d_1^t y,\ \forall (x, y) \in RI.$$

The (BLP) is a nonconvex and NP-Hard problem. Such characteristics are proper even if constraints (1b) do not exist. This is the most studied version of (BLP) . To solve this problem, many approaches have been proposed in the literature. These methods can be divided into the following categories:

(a) Methods based on vertex enumeration [8],[10],[13].
(b) Methods based on KKT reformulation [6],[14],[21].
(c) Methods based on meta-heuristics [20],[25].

In this paper, we consider the (BLP) with upper level constraints and use the second approach which consists in replacing the Follower's problem (1d)-(1f) with its Karush-Khun-Tucker optimality conditions. The resulting problem has the form (see [9], *proposition 5.2.2*) :

$$\max_{x,y} \; F(x,y) = c_1^t x + d_1^t y \tag{2a}$$
$$A_1 x + B_1 y + e = b_1 \tag{2b}$$
$$A_2 x + B_2 y + w = b_2 \tag{2c}$$
$$B_2^t u - v = d_2 \tag{2d}$$
$$v^t y + u^t w = 0 \tag{2e}$$
$$x \geq 0,\; y \geq 0,\; u \geq 0,\; v \geq 0,\; w \geq 0, e \geq 0. \tag{2f}$$

where $w \in \mathbb{R}^{m_2}$ and $e \geq 0$ are slack variables, $v \in \mathbb{R}^{n_2}$ and $u \in \mathbb{R}^{m_2}$ are dual variables. Then we apply an exact penalization to the nonconvex constraints (2e) in order to transform problem (2) in a concave minimization problem under linear constraints. Finally, we use DC programming and DCA algorithm to solve the resulting problem .

DC programming and DCA algorithm [3],[5] have been introduced by P.D. Tao in 1986. DCA is a primal-dual method for solving a general DC program. In general, DCA converges to a local solution, however, il was observed in practice that it converges quite often to a global one. This method has proved its efficiency from both theoretical and numerical viewpoints and has been successfully applied to a large number of nonconvex and nondifferentiable problems in various domains.

The paper is organized as follows. In section 2, we describe how to reformulate the problem via an exact penalty technique. Section 3 is devoted to DC programming and DCA algorithm for solving the resulting penalized problem. Computational results are presented in section 4, while some conclusion is presented in the last section.

2 Reformulation via exact penalty

In this section, we use an exact penalty to reformulate the problem (2) in the form of a concave minimization program. For this we first introduce some useful notations. Let

$z = \begin{pmatrix} x\; y\; e\; w\; v\; u \end{pmatrix}^t \in \mathbb{R}^n, \quad c = \begin{pmatrix} -c_1 \; -d_1 \; 0\;0\;0\;0 \end{pmatrix}^t \in \mathbb{R}^n,$

$A = \begin{pmatrix} A_1 & B_1 & I_{m_1} & 0 & 0 & 0 \\ A_2 & B_2 & 0 & I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{n_2} & B_2^t \end{pmatrix} \in R^{m \times n}, \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ d_2 \end{pmatrix} \in \mathbb{R}^m,$

$E_u = \begin{pmatrix} 0\;0\;0\;0\;0\; I_{m_2} \end{pmatrix}, \quad E_v = \begin{pmatrix} 0\;0\;0\;0\; I_{n_2}\; 0 \end{pmatrix},$

$E_w = \begin{pmatrix} 0\;0\;0\; I_{m_2}\; 0\;0 \end{pmatrix}, \quad E_y = \begin{pmatrix} 0\; I_{n_2}\; 0\;0\;0\;0 \end{pmatrix},$

where I_k is $k \times k$ identity matrix ; 0 is zero matrix with appropriate dimension for each case, with $n = n_1 + 2n_2 + m_1 + 2m_2;\; m = m_1 + m_2 + n_2$.
Using these notations, we have :

$u^t w = (E_u z)^t (E_w z) = z^t (E_u^t E_w) z = z^t D^1 z$, and $v^t y = (E_v z)^t (E_y z) = z^t (E_v^t E_y) z = z^t D^2 z$,
which gives : $u^t w + v^t y = z^t D^1 z + z^t D^2 z = z^t D z$ with $D^1 + D^2 = D$.

Note that the elements $d_{ij} (i = \overline{1,n},\ j = \overline{1,n})$ of matrix D are all nonnegative. Setting $Dz = q(z)$, problem (2) can be written as

$$\min \left\{ F(z) = c^t z, Az = b, z^t q(z) = 0, z \geq 0 \right\} \tag{3}$$

with $q(z) \geq 0,\ \forall z \geq 0$.

Consider the convex set $\mathcal{Z} = \{z \in \mathbb{R}^n : Az = b, z \geq 0\}$, and let be the function $\Psi : \mathbb{R}^n \to \mathbb{R}$ defined as $\Psi(z) = \sum\limits_{i=1}^{n} \min\{q_i(z),\ z_i\}$. Ψ is a finite concave and nonnegative function on $\mathcal{Z}$. Then we have

$$\{z \in \mathcal{Z},\ z^t q(z) = 0\} = \{z \in \mathcal{Z},\ \Psi(z) \leq 0\}.$$

Thus, the problem (3) can be rewritten in the form

$$\alpha = \min\{F(z)\ :\ z \in \mathcal{Z},\ \Psi(z) \leq 0\}. \tag{4}$$

If $\mathcal{Z}$ is a nonempty and bounded set, then ([2], *theorem 1*) there exists $k_0 \geq 0$ such that for every $k > k_0$, problem (4) is equivalent to the following penalized problem:

$$\alpha(k) = \min\{F(z) + k\Psi(z) : z \in \mathcal{Z}\}. \tag{5}$$

which is a concave minimization program under linear constraints.

3 DCA for solving problem (5)

This section is devoted to the DC decomposition of problem (5) and its resolution with DCA algorithm.

3.1 DC programming

Let $\Gamma_0(X)$ denotes the set of all lower semicontinuous proper convex functions on X. A general DC program has the form

$$\alpha = \inf\{f(x) = g(x) - h(x)\ :\ x \in X\}, \tag{6}$$

where $g,\ h \in \Gamma_0(X)$ are called DC components of the function f and $g - h$ is the DC decomposition of f.

The dual of (6) is the DC program

$$\alpha = \inf\{h^*(y) - g^*(y) : y \in Y\}, \tag{7}$$

where g^* and h^* are respectively the conjugate function of g and h :

$$g^*(y) = \sup\{\langle x, y\rangle - g(x) : x \in X\}.$$

For problem (6) we have the following necessary local optimality conditions [2]

$$\emptyset \neq \partial h(x^*) \subset \partial g(x^*) \tag{8}$$

$$\emptyset \neq \partial g(x^*) \cap \partial h(x^*). \tag{9}$$

Such a point x^* is called critical point of $g - h$.

A function g is polyhedral convex on a convex polyhedral set $C \subset \mathbb{R}^n$ if it is of the form

$$g(x) = \max\{\langle a_i, x\rangle - \beta_i : \quad i = 1, ..., m\} + \chi_C(x),$$

where $a_i \in \mathbb{R}^n, \beta_i \in \mathbb{R}, i = 1, ..., m$ and χ_C is the indicator function of C

$$\chi_C(x) = 0 \text{ if } x \in C, \quad +\infty \text{ otherwise.}$$

A DC problem is called polyhedral DC program if g or h are polyhedral convex functions. For this class of DC programs condition (8) is also sufficient.

DCA algorithm is a descent method without linesearch, consisting of the construction of the two sequences $\{x^i\}$ and $\{y^i\}$, (candidates for being primal and dual solutions, respectively), such that their corresponding limit points satisfy local optimality conditions. Recall that there are two forms of DCA: the simplified DCA (or DCA) and the complete DCA. In practice we use the first because it is less expensive.

DCA algorithm:

```
1 : $x^0$ given.
2 : Compute $y^i \in \partial h(x^i)$.
3 : Compute $x^{i+1} \in \partial g^*(y^i)$.
4 : if a convergence criterion is satisfied Stop; else $i = i + 1$ and goto 2.
```

3.2 DCA for solving (5)

We first prove that (5) is a DC program then we present DCA applied to the resulting DC program.
Denote by $\chi_{\mathcal{Z}}$ the indicator function of $\mathcal{Z}$ and let g and h given by

$$g(z) = \chi_{\mathcal{Z}}(z) \text{ and } h(z) = -F(z) - k\Psi(z). \tag{10}$$

Therefore g and h are convex functions and problem (5) is equivalent to the DC program of the form

$$\min\{g(z) - h(z) : z \in \mathbb{R}^n\}. \tag{11}$$

The application of the DCA to problem (11) consists of computing the two sequences $\{t^i\}$ and $\{z^{i+1}\}$ defined by

$$t^i \in \partial h(z^i) \ \text{ and } \ z^{i+1} \in \partial g^*(t^i).$$

Using the rules in convex analysis we compute $\{t^i\}$ and $\{z^{i+1}\}$.

Computation of $t^i \in \partial h(z^i)$ **:** we choose $t^i \in \partial\big(-c^t z^i - k\sum_{j=1}^{n} \min\{q_j(z^i),\ z^i_j\}\big)$ of the form

$$t^i = -c + k\theta^i, \tag{12}$$

where $\theta^i \in \sum_{j=1}^{n} \partial\big(\max\{-q_j(z^i),\ -z^i_j\}\big)$ and $q_j(z^i) = D_j z^i$.
Let be

$$\theta^i = -\sum_{j=1}^{n} \begin{cases} D^t_j, & \text{if } \ z^i_j > D_j z^i, \\ e_j, & \text{if } \ z^i_j < D_j z^i, \\ \gamma e_j + (1-\gamma) D^t_j, & \text{if } \ z^i_j = D_j z^i, \end{cases} \tag{13}$$

where D_j is the *j-th* line of matrix D, e_j is the *j-th* unit vector of $\mathbb{R}^n$ and $\gamma \in [0,1]$.
Hence, θ^i given by (13) is an element of $\sum_{j=1}^{n} \partial\big(\max\{-D_j z^i, -z^i\}\big)$.

Computation of $z^{i+1} \in \partial g^*(t^i)$ **:** following [22], we can choose z^{i+1} as the solution of the following linear programming problem

$$\min\{-\langle z,\ t^i\rangle : z \in \mathcal{Z}\}, \tag{14}$$

DCABLP (DCA for (5))

```
1 : Let $z^0$ initial guess, $\epsilon > 0$, $k \in \mathbb{R}_+$, $\gamma \in [0,1]$ and $\lambda > 0$. Set $i = 0$.
2 : Compute $t^i \in \partial h(z^i)$ using (12).
3 : Compute $z^{i+1} \in \partial g^*(t^i)$ by solving (14).
4 : If $y^{i+1} \in \arg\max\{f(x^{i+1}, y) : B_2 y \le b_2 - A_2 x^{i+1},\ y \ge 0\}$, then go to 5;
    otherwise go to 7.
5 : Compute $(v^*, u^*)$, solution of the dual problem
    $\min\{u^t(b_2 - A_2 x^{i+1}) : B_2^t u - v = d_2,\ u \ge 0,\ v \ge 0\}$, hence
    $z^{i+1} = (x^{i+1}, y^{i+1}, e^{i+1}, w^{i+1}, v^*, u^*)$.
6 : If $\|z^{i+1} - z^i\|/(\|z^i\| + 1) \le \epsilon$, then stop $z^{i+1}$ is optimal solution of (5); and $(x^*, y^*)$ is optimal for (1).
    otherwise go to 7.
7 : Set $z^i = z^{i+1}$, $i = i + 1$, $k = k + \lambda$ and go to 2.
```

Remark 1 *Problem (5), with DC decomposition (10), is a polyhedral DC program since $g = \chi_{\mathcal{Z}}$ is polyhedral convex function [22]. In this case DCA applied to (5) has finite convergence [3],[5].*

Remark 2 *In step 4 of the algorithm, we test the feasibility of the solution (x^{i+1}, y^{i+1}) for the (BLP) . If the test in step 4 is satisfied we have $y^{i+1} \in$*

*$R(x^{i+1})$ (*see definition 4*). Since $(x^{i+1}, y^{i+1}) \in S$, then we have $(x^{i+1}, y^{i+1}) \in RI$ which implies that (x^{i+1}, y^{i+1}) is a feasible solution.*

Remark 3 *If $z^* = (x^*, y^*, e^*, w^*, v^*, u^*)$ is an optimal solution of (5), then the optimality of (x^*, y^*) for problem (1) is provided by step 5. In fact, suppose that z^* is an optimal solution for (5) with $k = \tilde{k}$. Then, we have*

$$g(z^*) - h(z^*) \le g(z) - h(z), \ \forall z \in \mathcal{Z}.$$

Since $g = \chi_{\mathcal{Z}}$ and $z, z^* \in \mathcal{Z}$, we have

$$0 - h(z^*) \le 0 - h(z), \ \forall z \in \mathcal{Z},$$

which gives

$$F(z^*) + \tilde{k}\Psi(z^*) \le F(z) + \tilde{k}\Psi(z), \ \forall z \in \mathcal{Z}. \tag{15}$$

$$c^t z^* + \tilde{k}\Psi(z^*) \le c^t z + \tilde{k}\Psi(z), \ \forall z \in \mathcal{Z}. \tag{16}$$

From remark 2, (x^*, y^*) is an optimal solution of the follower's problem. Moreover, from duality in linear programming, if (v^*, u^*) is the optimal solution of the follower's dual problem, then the complementary constraints are satisfied ; that is the penalty function $\Psi(z^*)$ is zero. We have then

$$c^t z^* \le c^t z, \ \forall z \in \mathcal{Z}.$$

Hence

$$-c^t z^* \ge -c^t z, \ \forall z \in \mathcal{Z},$$

which gives

$$c_1^t x^* + d_1^t y^* \ge c_1^t x + d_1^t y, \ \forall (x, y) \in RI.$$

Then (x^*, y^*) is an optimal solution of problem (1).

Initial point

In order to find an appropriate initial point to DCABLP, we use DCA to solve the following problem

$$0 = \min\{\Psi(z) : \bar{A}z = \bar{b}, z \ge 0\} \tag{17}$$

where

$$\bar{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ A_2 & B_2 & 0 & I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{n_2} & B_2^t \end{pmatrix} \quad \text{and} \quad \bar{b} = \begin{pmatrix} 0 \\ b_2 \\ d_2 \end{pmatrix},$$

which is a concave minimization problem and whose optimal value is known (equal to zero).

4 Computational results

In this section we report computational experiments with the proposed algorithm DCABLP. The algorithm has been coded in MATLAB and run on a PC Pentium IV 3.00 GHz, RAM 512Mb using the simplex algorithm for solving linear subproblems. In order to evaluate the performance of the proposed algorithm we tested it on a set of 11 problems whose optimal solutions are known. The initial point was calculated by solving problem (17). For parameter γ in (13), we took the value $\gamma = 0.5$. The algorithm is stopped when $\epsilon \leq 10^{-3}$.

The results of the algorithm are reported in table 1 where we use the following notations:

k: penalty parameter.
λ: increasing parameter.
t: the total time for the algorithm and is given in seconds.
It: the number of iterations of the algorithm.
$(x^*; y^*)$: the optimal solution of the problem.
$(F^*; f^*)$: the Leader's and the Follower's optimal value respectively.
"$-$" the algorithm failed to solve the problem.

Test problems

$$\text{P1 [24]} : \begin{cases} \max F(x,y) = 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3 \\ x \geq 0 \\ \max f(x,y) = -x_1 - 2x_2 - y_1 - y_2 - 2y_3 \\ 0x_1 + 0x_2 - y_1 + y_2 + y_3 \leq 1 \\ 2x_1 + 0x_2 - y_1 + 2y_2 - 0.5y_3 \leq 1 \\ 0x_1 + 2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1 \\ y \geq 0 \end{cases}$$

$$\text{P2 [6]} : \begin{cases} \max F(x,y) = x + 3y \\ x \geq 0 \\ \max f(x,y) = x - 3y \\ -x - 2y \leq -10 \\ x - 2y \leq 6 \\ 2x - y \leq 21 \\ x + 2y \leq 38 \\ -x + 2y \leq 18 \\ y \geq 0 \end{cases}$$

$$\text{P3 [20]} : \begin{cases} \max F(x,y) = 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3 \\ x_1 + 2x_2 - y_3 \leq 1.3 \\ x \geq 0 \\ \max f(x,y) = -2y_1 - y_2 - 2y_3 \\ 0x_1 + 0x_2 - y_1 + y_2 + y_3 \leq 1 \\ 4x_1 + 0x_2 - 2y_1 + 4y_2 - y_3 \leq 2 \\ 0x_1 + 4x_2 + 4y_1 - 2y_2 - y_3 \leq 2 \\ y \geq 0 \end{cases}$$

$$\text{P4 [10]} : \begin{cases} \max F(x,y) = x + y \\ x \geq 0 \\ \max f(x,y) = 0x - y \\ -4x - 3y \leq -19 \\ x + 2y \leq 11 \\ 3x + y \leq 13 \\ y \geq 0 \end{cases}$$

$$\text{P5 [26]} : \begin{cases} \max F(x,y) = -x - 5y \\ x \geq 0 \\ \max f(x,y) = 0x + y \\ -x - y \leq -8 \\ -3x + 2y \leq 6 \\ x + 4y \leq 48 \\ x - 5y \leq 9 \\ y \geq 0 \end{cases}$$

$$\text{P6 [23]} : \begin{cases} \max F(x,y) = -x_1 - 2x_2 - 2y_1 + y_2 \\ x_1 + x_2 + 0.5y_1 + y_2 \leq 6 \\ x \geq 0 \\ \max f(x,y) = -y_1 + 2y_2 \\ -x_1 + 2x_2 + 0y_1 + y_2 \leq 4 \\ -x_1 - x_2 + y_1 + y_2 \leq 5 \\ y \geq 0 \end{cases}$$

$$\text{P7 [14]} : \begin{cases} \max F(x,y) = -0.4x_1 - y_1 - 5y_2 + 0y_3 + 0y_4 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 0y_1 + 0.5y_2 - y_3 - 2y_4 \\ -0.1x_1 - y_1 - y_2 + 0y_3 + 0y_4 \leq -1 \\ 0.2x_1 + 0y_1 + 1.25y_2 + 0y_3 - y_4 \leq -1 \\ -x + 6y_1 + y_2 - 2y_3 + 0y_4 \leq 1 \\ y \geq 0 \end{cases}$$

$$\text{P8 [9]} : \begin{cases} \max F(x,y) = -x + 4y \\ x \geq 0 \\ \max f(x,y) = 0x - y \\ -x - y \leq -3 \\ -2x + y \leq 0 \\ 2x + y \leq 12 \\ -3x + 2y \geq -4 \\ y \geq 0 \end{cases}$$

$$\text{P9 [24]} : \begin{cases} \max F(x,y) = 2x_1 - x_2 - 0.5y_1 \\ x_1 + x_2 \leq 2 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 0x_2 + 4y_1 - y_2 \\ -2x_1 + y_1 - y_2 \leq -2.5 \\ x_1 - 3x_2 + y_2 \leq 2 \\ y \geq 0 \end{cases}$$

$$\text{P10 [24]} : \begin{cases} \max F(x,y) = -x + 4y \\ x \geq 0 \\ \max f(x,y) = 0x - y \\ -2x + y \leq 0 \\ 2x + 5y \leq 108 \\ 2x - 3y \leq -4 \\ y \geq 0 \end{cases}$$

$$\text{P11 [8]} : \begin{cases} \max F(x,y) = 2x_1 - x_2 - x_3 + 2x_4 + x_5 - 3.5x_6 - y_1 - 1.5y_2 + 3y_3 \\ x \geq 0 \\ \max f(x,y) = 0x_1 + 2x_2 + 0x_3 + 0x_4 - x_5 + 0x_6 + 3y_1 - y_2 - 4y_3 \\ -x_1 + 0.2x_2 + 0x_3 + 0x_4 + x_5 + 2x_6 - 4y_1 + 2y_2 + y_3 \leq 12 \\ x_1 + 0x_2 + x_3 - 2x_4 + 0x_5 + 0x_6 + 0y_1 - 4y_2 + y_3 \leq 10 \\ 5x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 + 3.2x_6 + 2y_1 + 2y_2 + 0y_3 \leq 15 \\ 0x_1 - 3x_2 + 0x_3 - x_4 + x_5 + 0x_6 - 2y_1 + 0y_2 + 0y_3 \leq 12 \\ -2x_1 - x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0y_1 - y_2 + y_3 \leq -2 \\ 0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 - y_1 - 2y_2 - y_3 \leq -2 \\ 0x_1 - 2x_2 - 3x_3 + 0x_4 - x_5 + 0x_6 + 0y_1 + 0y_2 + 0y_3 \leq -3 \\ y \geq 0 \end{cases}$$

Pb	*(k; λ)*	$(x^*; y^*)$	$(F^*; f^*)$	*t*	*It*
	(1;0.1)	-	-	-	-
	(1;0.5)	(0 0.9 0 0.6 0.4)	(29.2; -3.2)	4.06	19
	(1;1)	(0 0.9 0 0.6 0.4)	(29.2; -3.2)	2.48	11
1	(5;1)	(0 0.9 0 0.6 0.4)	(29.2; -3.2)	2.07	7
	(5;5)	(0 0.9 0 0.6 0.4)	(29.2; -3.2)	1.26	3
	(10;5)	(0 0.75 0 0.5 0)	(23; -2)	0.7	2
	(0.01;0.01)	-	-	-	-
	(0.1;0.01)	(16 11)	(49; -17)	3.14	3
	(0.1;0.1)	(16 11)	(49; -17)	6.92	6
	(1;1)	(16 11)	(49; -17)	0.65	2
2	(5;1)	(12 3)	(21; 3)	0.17	2
	(5;5)	(12 3)	(21; 3)	1.23	2
	(10;5)	(12 3)	(21; 3)	0.18	2
	(5;5)	-	-	-	-
3	(10;5)	(0 0.78 0 0.43 0.26)	(21.36; -0.95)	1.18	2
	(15;5)	(0 0.78 0 0.43 0.26)	(21.36; -0.95)	1.44	2
	(20;5)	(0 0.78 0 0.43 0.26)	(21.36; -0.95)	1.20	2
	(0.1;0.1)	-	-	-	-
	(1;0.1)	(4 1)	(5; -1)	0.09	2
	(1;1)	(4 1)	(5; -1)	1.09	4
4	(5;1)	(1 5)	(6; -5)	0.7	2
	(5;5)	(1 5)	(6; -5)	0.6	2
	(10;5)	(1 5)	(6; -5)	0.6	2
	(0.1;0.1)	-	-	-	-
	(1;0.1)	(2 6)	(-32; 6)	1.5	3
	(1;1)	(2 6)	(-32; 6)	1.7	3
5	(5;1)	(2 6)	(-32; 6)	0.5	2
	(5;5)	(2 6)	(-32; 6)	0.6	2
	(10;5)	(2 6)	(-32; 6)	0.56	2
	(0.1;0.01)	(4,10)	(4; 10)	1.15	2
	(1;0.1)	(4,10)	(4; 10)	1.01	2
6	(5;1)	(4,10)	(4; 10)	1.3	2
	(5;5)	(4,10)	(4; 10)	0.96	2
	(10;5)	(4,10)	(4; 10)	0.5	2

	(0.1;0.01)	-	-	-	-
	(0.1;0.1)	(0 0 1 0 2.25)	(-5; -4)	0.68	2
	(1;1)	(0 0 1 0 2.25)	(-5; -4)	0.6	2
7	(5;1)	(0 0 1 0 2.25)	(-5; -4)	0.56	2
	(5;5)	(0 0 1 0 2.25)	(-5; -4)	1.06	2
	(10;5)	(0 0 1 0 2.25)	(-5; -4)	0.96	2
	(0.1;0.1)	-	-	-	-
	(1;0.1)	(4 4)	(12; -4)	3.96	8
	(1;1)	(4 4)	(12; -4)	1.51	3
8	(5;1)	(2 1)	(2; -1)	0.56	2
	(5;5)	(2 1)	(2; -1)	0.21	2
	(10;5)	(2 1)	(2; -1)	0.65	2
	(0.1;0.1)	-	-	-	-
	(1;0.1)	(2 0 1.5 0)	(3.25; 6)	2.12	2
	(1;1)	(2 0 1.5 0)	(3.25; 6)	1.40	4
9	(5;1)	(2 0 1.5 0)	(3.25; 6)	1.42	2
	(5;5)	(2 0 1.5 0)	(3.25; 6)	1.39	2
	(10;5)	(2 0 1.5 0)	(3.25; 6)	1.43	2
	(0.1;0.1)	-	-	-	
	(1;0.1)	(19 14)	(37; -14)	0.64	2
10	(1;1)	(19 14)	(37; -14)	0.17	2
	(5;1)	(19 14)	(37; -14)	1.28	6
	(5;5)	(19 14)	(37; -14)	0.5	2
	(0.1;0.1)	-	-	-	-
	(0.5;0.1)	(0 4 0 15 9.2 0 0 0 2)	(41.2; -9.2)	1.17	2
11	(1;1)	(0 2 0 11 19.6 0 2 0 0)	(37.6; -13.6)	0.7	2
	(5;1)	(1 0 0 6 21 0 2 0 0)	(33; -19)	2.64	6
	(5;5)	(1 0 0 6 21 0 2 0 0)	(33; -19)	1.2	2
	(10;5)	(0.5 0 0 0 3.93 3.92 0 1 0)	(-8.04; -4.93)	0.6	2

TAB. 1: Numerical results for DCABLP

Comments: from numerical experiments, we observe that

- the algorithm with the starting procedure is efficient: in most problems, with a good choice of the penalty parameter, it computed the global solution.
- the algorithm terminates rapidly; the average number of iterations is 2.
- the total time of the algorithm is small; this result is normal because there are only linear programs to solve at each iteration.
- the computational requirements (choice of penalty parameter and increasing parameter) are greatly dependent on the problem structure.

In table 2 we give the comparison between our results and the results in some references: (/ : means that the execution time is not given in the reference)

	parameters	results in the paper				result in the references			
	(k; λ)	$(x^*; y^*)$	F^*	t	*It*	$(x^*; y^*)$	F^*	t	*It*
1 [24]	(5;1)	(0 0.9 0 0.6 0.4)	29.2	2.07	7	(0 0.89 0 0.59 0.39)	25.92	0.047	22
2 [6]	(1;1)	(16 11)	49	0.65	2	(16 11)	49	/	28
5 [26]	(5;5)	(2 6)	-32	0.6	2	(2.0002 5.9999)	-31.9999	/	34
6 [23]	(5;5)	(4 10)	4	0.96	2	(4 10)	4	/	11
9 [24]	(5;5)	(2 0 1.5 0)	3.25	1.39	2	(2 0 1.5 0)	3.25	0.037	107
10 [24]	(5;5)	(19 14)	37	0.5	2	(18.92 13.95)	36.88	0	8

TAB. 2: Comparison results

From table 2 the results in the paper accord with the results in the references. In addition, we can see that our algorithm gives the exact solution.

5 Conclusion

We have presented a DC optimization approach for solving bilevel linear optimization problems. The resulted DC program is polyhedral and the DCA algorithm has a finite convergence. Computational experiments show that the procedure of calculating the starting point is efficient, but the search of global solution remains sensitive to the choice of the penalty parameter. The proposed algorithm is fast, since it solves only linear subproblems at each iteration.

References

1. F.B. Akoa: Approches de points intérieurs et de la programmation DC en optimisation non convexe. Codes et simulations numériques industrielles. Thèse de Doctorat, Institut national des sciences appliquées de Rouen. (2005)
2. L.T.H. An and P.D. Tao: A continuous approach for globally solving linearly constrained quadratic zero-one programming problems. Optimization, 50: 93 − 120, (2001)
3. L.T.H. An and P.D. Tao: Convex analysis approach to dc programming: theory and applications. Acta Mathematica Vietnamica, 22: 289 − 355, (1997)
4. L.T.H. An and P.D. Tao: Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. J. Glob. Optim., 11: 253 − 285, (1997)
5. L.T.H. An and P.D. Tao: The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. Annals Oper. Res., 133: 23 − 46, (2005)
6. G. Anandalingam and D. J. White: A solution for the linear static Stackelberg problem using penalty functions. IEEE Trans. on Aut. Cont., 35: 1170 − 1173, (1990)
7. G. Anandalingam and T. L. Friesz: Hierarchical optimization : An introduction. Annals of Oper. Res., 34: 1 − 11, (1992)
8. J.F. Bard: An efficient point algorithm for a linear two stage optimization problem. Oper. Res., 31: 670 − 684, (1983)
9. J.F. Bard: Practical bilevel optimization : algorithms and applications. Kluwer academic publishers, Dordrecht (1998)
10. O. Ben-Ayed and C. E. Blair: Computational difficulties of bilevel linear programming. Oper. Res., 38: 556 − 560, (1990)
11. W.F. Bialas: Multilevel mathematical programming : An introduction. Technical report, Department of Industrial Engineering. University at Buffalo (2002)
12. W.F. Bialas and M.H. Karwan: Multilevel linear programming. Technical Report, Oper. Res. Program, Dept. of Industrial Eng., State University of New York at Buffalo, 1 − 78, (1978)
13. W.F. Bialas and M.H. Karwan: Two-level linear programming. Management Science, 30(8) :1004 − 1020, (1984)
14. M. Campêlo, S. Dantas and S. Scheimberg: A note on a penalty function approach for solving bilevel linear programs. J. Glob. Optim., 16: 245 − 255, (2000)

15. B. Colson, P. Marcotte and G. Savard: Bilevel programming : A survey. 4OR A Quarterly, J. Oper. Res., (2007)
16. S. Dempe: Foundations of Bilevel Programming. Kluwer academic publishers, Dordrecht (2000)
17. S. Dempe and A.G. Mersha. Linear bilevel programming with upper level constraints depending on the lower level solution. App. Math. and Comp., 180: $247-254$ (2006)
18. J. Fortuny-Amat, B. McCarl: A representation and economic interpretation of a two level programming problem. Oper. Res., 321: $783-792$, (1981)
19. A. Haurie, G. Savard, and D.J. White. A note on : An efficient point algorithm for a linear two stage optimization problem. Operations Research, 38(3) :$553-555$, (1990)
20. S.R. Hejazi, A. Memariania, G. Jahanshahloob and M.M. Sepehria. Linear bilevel programming solution by genetic algorithm. Comp. and Oper. Res., 29: $1913-1925$, (2005)
21. Y. Lv, T. Hu, G. Wang and Z. Wan. A penalty function method based on Kuhn–Tucker condition for solving linear bilevel programming. App. Math. and Comp., (2006)
22. R.T. Rockafellar: Convex analysis. Princeton, USA (1970)
23. N.V. Thoai, Y. Yamamoto and A. Yoshise: Global optimization method for solving mathematical programs with linear complementarity constraints. J. Opt. Theory and App., 124(2) :$467-490$, (2005)
24. H. Tuy, A. Migdalas and N. T. Hoai-Phuong: A novel approach to Bilevel nonlinear programming. J. Glob. Optim., 38: $527-554$, (2007)
25. G. Wang, Z. Wan, and X. Wang: Solving method for a class of bilevel linear programming based on genetic algorithms. Supported by the National Natural Science Foundation and the Doctoral Foundation in Ministry of Education of China.
26. D.L Zhu, Q. Xu and Z. Lin: A homotopy method for solving bilevel programming problem. J. Nonlinear Analysis, 57: $917-928$, (2004)