J Glob Optim (2009) 44:313–337
DOI 10.1007/s10898-008-9325-7

DC programming techniques for solving a class of nonlinear bilevel programs

Le Thi Hoai An · Pham Dinh Tao · Nam Nguyen Canh · Nguyen Van Thoai

Received: 14 May 2006 / Accepted: 11 June 2008 / Published online: 9 July 2008

Abstract We propose a method for finding a global solution of a class of nonlinear bilevel programs, in which the objective function in the first level is a DC function, and the second level consists of finding a Karush-Kuhn-Tucker point of a quadratic programming problem. This method is a combination of the local algorithm DCA in DC programming with a branch and bound scheme well known in discrete and global optimization. Computational results on a class of quadratic bilevel programs are reported.

Keywords Bilevel programming · Nonconvex programming · DC programming · DCA · Global optimization · Branch and bound techniques

1 Introduction

A real function defined on a convex set is called DC if it can be represented as the difference of two convex functions. The subject of the present article is a class of nonlinear bilevel

L. T. Hoai An
Laboratory of Theoretical and Applied Computer Science, UFR MIM, Metz University, Ile du Saulcy, 57045 Metz, France
e-mail: lethi@sciences.univ-metz.fr

P. D. Tao (✉) · N. Nguyen Canh
Laboratory of Modelling, Optimization and Operations Research, LMI, National Institute for Applied Sciences - Rouen, BP 08, Place Emile Blondel F, 76131 Mont Saint Aignan Cedex, France
e-mail: pham@insa-rouen.fr

N. Nguyen Canh
e-mail: nguyencn@insa-rouen.fr

N. Van Thoai
Department of Mathematics, University of Trier, 54286 Trier, Germany
e-mail: thoai@uni-trier.de

programs dealing with DC functions, which is described in the form

$$\begin{array}{lll} \alpha := \min & f(x, y) & \\ \text{s.t.} & (x, y) \in Z & \\ & y \in K(x), & \end{array} \tag{1}$$

where $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ is a DC function, Z is a polyhedral convex subset of $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, and for each given $x \in \mathbb{R}^{n_1}$, $K(x)$ is the set of all Karush-Kuhn-Tucker points of the following quadratic programming problem:

$$\begin{array}{ll} \min & y^T Px + \frac{1}{2} y^T Qy + q^T y \\ \text{s.t.} & Dx + Ey + b \le 0 \end{array} \tag{2}$$

with P, Q, D and E being matrices of dimensions $(n_2 \times n_1)$, $(n_2 \times n_2)$, $(p \times n_1)$ and $(p \times n_2)$, respectively, and $q \in \mathbb{R}^{n_2}$, $b \in \mathbb{R}^p$. By replacing Q with $\frac{1}{2}(Q + Q^T)$, we can assume, without loss of generality, that Q is symmetric.

For the establishment of solution methods for the above nonlinear bilevel programming problem, we assume that the set

$$\{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : (x, y) \in Z,\ Dx + Ey + b \le 0\}$$

is nonempty and bounded.

As well known, for each given x, $K(x)$ is the set of all solutions $(y, \lambda) \in \mathbb{R}^{n_2} \times \mathbb{R}^p$ of the system

$$\begin{array}{l} Px + Qy + q + E^T \lambda = 0 \\ Dx + Ey + b \le 0 \\ \lambda^T (Dx + Ey + b) = 0 \\ \lambda \ge 0. \end{array} \tag{3}$$

Using (3) we can rewrite Problem (1) in the form

$$\begin{array}{ll} \alpha := \min & f(x, y) \\ \text{s.t.} & (x, y) \in Z \\ & Px + Qy + E^T \lambda + q = 0 \\ & Dx + Ey + b \le 0 \\ & \lambda^T (Dx + Ey + b) = 0 \\ & \lambda \ge 0. \end{array} \tag{4}$$

Clearly, Problems (1) and (4) are equivalent in the following sense:

(i) If (x^*, y^*) is an optimal solution of (1), then there exists λ^* such that (x^*, y^*, λ^*) is an optimal solution of (4), and
(ii) If (x^*, y^*, λ^*) is an optimal solution of (4), then (x^*, y^*) is an optimal solution of (1).

The nonconvex program (1) is NP-hard, even in case $f(x, y)$ is quadratic. Obviously, the complete nonlinear bilevel program, obtained from (1) in replacing $K(x)$ by the solution set of (2), is even more difficult, because the latter one is itself NP-hard.

Bilevel programming problems play a prominent role in the field of nonconvex global optimization because of its theoretical aspects as well as its wide range of applications. In [23] and [28], bibliographies of almost all references on this topic up to 1997 can be found. The most efficient methods are established for solving the linear bilevel programming problem in which the function f is linear, Z is a polyhedral convex set, and P and Q are null matrices. For some classes of nonlinear problems, solution methods can be found e.g. in [1–6,19,24,27].

In this article, we present a method for solving the nonlinear bilevel programming problem of the form (1), which is a combination of a local approach in DC programming and the well known branch and bound scheme successfully used in global optimization. To this purpose, we first apply the theory of exact penalization of mathematical programs with equilibrium constraints developed by Luo et al. [17] and the exact penalty techniques in DC programming due to Le Thi Hoai An et al. [13,14] to reformulate Problem (4) as the problem of minimizing a DC function over a polyhedral convex set. The resulting problem is then handled by a local approach in DC programming developed by Pham Dinh Tao and Le Thi Hoai An in their early works (see, [8,9,13,10–12,14,26,25]). A branch and bound scheme in combination with this local approach yields an algorithm for finding a global optimal solution of the nonlinear bilevel program under consideration.

Both DCA and branch and bound algorithms are finite in case the objective function $f(x, y)$ is convex.

The article is organized as follows. After the introduction, the mentioned DC programming and DCA (DC Algorithms) are briefly presented in Sect. 2. Section 3 deals with a special realization of DCA to the underlying nonlinear bilevel program. A combination of DCA with a branch and bound algorithm is discussed in Sect. 4. Numerical examples and computational results on some classes of nonconvex quadratic bilevel programs are reported in Sect. 5, while some conclusion is presented in the last section.

2 DC programming and DCA

First of all, to make the paper self-contained and so more comprehensive for the reader not familiar with DC programming and DCA, we will outline main theoretical and algorithmic results on the topic. It is then more natural and elegant to enhance the general results for the nonlinear bilevel program.

2.1 DC programming

Let $\Gamma_0(\mathbb{R}^n)$ denote the convex cone of all lower semicontinuous proper convex functions on $\mathbb{R}^n$. The vector space of DC functions, $DC(\mathbb{R}^n) = \Gamma_0(\mathbb{R}^n) - \Gamma_0(\mathbb{R}^n)$, is quite large to contain almost real life objective functions and is closed under all the operations usually considered in Optimization.

Consider the standard DC program

$$\alpha = \inf\{f(x) := g(x) - h(x) \; : \; x \in \mathbb{R}^n\} \quad (P_{dc})$$

with $g, h \in \Gamma_0(\mathbb{R}^n)$. Remark that the closed convex constraint set C is incorporated in the first convex DC component g with the help of its indicator function χ_C ($\chi_C\ (x) := 0$ if $x \in C, +\infty$ otherwise). Let

$$g^*(y) := \sup\{\langle x, y\rangle - g(x) \; : \; x \in \mathbb{R}^n\}$$

be the conjugate function of g. The dual problem of (P_{dc}) is defined by

$$\alpha = \inf\{h^*(y) - g^*(y) \; : \; y \in \mathbb{R}^n\} \quad (D_{dc})$$

by using the fact that every function $\theta \in \Gamma_0(\mathbb{R}^n)$ is characterized as a pointwise supremum of affine functions. More precisely

$$\theta(x) := \sup\{\langle x, y\rangle - \theta^*(y) : y \in \mathbb{R}^n\}.$$

DC programming investigates the structure of the vector space $DC(\mathbb{R}^n)$, DC duality, optimality conditions for DC programs and relations between primal and dual DC programs. They constitute the basic tools for the construction of DCA. The complexity of DC programs resides, of course, in the lack of practical optimal globality conditions. It is worth noting that our works involve the convex DC components g and h but not the DC function f itself.

A DC program (P_{dc}) is called *polyhedral* if either g or h is a polyhedral convex function. This class of DC programs, which is frequently encountered in practice and has been extensively developed in our previous works (see e.g. [9] and references therein), enjoys interesting properties (from both theoretical and practical viewpoints) concerning the local optimality and the convergence of the DCA.

Instead of impractical global optimality conditions, we developed the following necessary local optimality conditions for DC programs in their primal part (by symmetry their dual part is trivial (see [8–10,12,13,25,26] and references therein):

(i) $\partial h(x^*) \cap \partial g(x^*) \neq \emptyset$

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

(ii) $\emptyset \neq \partial h(x^*) \subset \partial g(x^*)$.

The condition (ii) is also sufficient for many classes of DC programs. In particular its sufficiency holds for the next cases quite often encountered in practice:

+ In polyhedral DC programs with h being a polyhedral convex function. Hence in this case, a critical point is almost always a local minimizer for (P_{dc}) because a polyhedral convex function is differentiable everywhere except on a set of measure zero. More precisely if

$$h(x) := \max\{a_i^T x - \gamma_i : i = 1, \ldots, m\}$$

then $\partial h(x) = co\{a_i\colon i \in M(x)\}$ where $M(x) := \{i \in M : h(x) = a_i^T x - \gamma_i\}$ where $M := \{1, \ldots, m\}$ and co denotes the convex hull. Hence h is nondifferentiale only on a union of a finite collection of affine sets of dimensions smaller than that of the whole space.Hence in this case, a critical point is almost always a local minimizer for (P_{dc}).

+ In case the function f is locally convex at x^*.

Note that a polyhedral DC function $f = g - h$ with h being polyhedral convex is locally convex wherever h is differentiable.

(iii) The transportation of local and global solutions between (P_{dc}) and (D_{dc}) is expressed by:

$$[\cup_{y^* \in \mathcal{D}} \partial g^*(y^*)] \subset \mathcal{P}, \ [\cup_{x^* \in \mathcal{P}} \partial h(x^*)] \subset \mathcal{D}$$

where $\mathcal{P}$ and $\mathcal{D}$ denote the solution sets of (P_{dc}) and (D_{dc}) respectively.

Under technical conditions, this transportaion holds also for local solutions of (P_{dc}) and (D_{dc}).

The transportation property has led to the following design of DCA.

2.2 DCA

Based on local optimality conditions and duality in DC programming, the DCA consists in the construction of two sequences $\{x^k\}$ and $\{y^k\}$ (candidates to be solutions of (P_{dc}) and (D_{dc}) resp.) such that x^{k+1} $(resp.\ y^k)$ is a solution to the convex program $(P_k)(resp.\ (D_k))$ defined by

$$\inf\{g(x) - h_k(x) : x \in \mathbb{R}^n\} \quad (P_k)$$

where

$$h_k(x) := h(x^k) + \langle x - x^k, y^k \rangle, \tag{5}$$

$$\inf\{h^*(y) - (g^*)_k(y) \ : y \in \mathbb{R}^n\} \quad (D_k)$$

where

$$(g^*)_k(y) := g^*(y^{k-1}) + \langle y - y^{k-1}, x^k \rangle. \tag{6}$$

It is clear that $(P_k)(resp.(D_k))$ is obtained from $(P_{dc})(resp.(D_{dc}))$ by replacing h $(resp.$ $g^*)$ with its affine minorization h_k (resp. $(g^*)_k$) defined by $y^k \in \partial h(x^k)$ $(resp. x^k \in \partial g^*(y^{k-1}))$ and the DCA then yields the next scheme:

$$y^k \in \partial h(x^k); \quad x^{k+1} \in \partial g^*(y^k).$$

We will end this subsection by summarizing the main properties of DCA [8–10,12,13,25,26], and references therein.

DCA's convergence theorem: DCA is a descent method without linesearch which enjoys the following primal properties (the dual ones can be formulated in a similar way):

(1) The sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing and

- $g(x^{k+1}) - h(x^{k+1}) = g(x^k) - h(x^k)$ iff $y^k \in \partial g(x^k) \cap \partial h(x^k)$, $y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1})$ and $[\rho(g, C) + \rho(h, C)]\|x^{k+1} - x^k\| = 0$. Moreover if g or h are strictly convex on C then $x^k = x^{k+1}$.
  In such a case DCA terminates at the kth iteration (finite convergence of DCA).
  Here C (resp. D) is a convex set containing the sequence $\{x^k\}$ (resp. $\{y^k\}$) and the modulus of strong convexity of g on C, denoted by $\rho(g, C)$ or $\rho(g)$ if $C = \mathbb{R}^n$, is given by:

$$\rho(g, C) = \sup\{\rho \geq 0 : g - (\rho/2)\| \cdot \|^2 \text{ be convex on } C\}.$$

- $h^*(y^{k+1}) - g^*(y^{k+1}) = h^*(y^k) - g^*(y^k)$ iff $x^{k+1} \in \partial g^*(y^k) \cap \partial h^*(y^k), x^{k+1} \in \partial g^*(y^{k+1}) \cap \partial h^*(y^{k+1})$ and $[\rho(g^*, D) + \rho(h^*, D)]\|y^{k+1} - y^k\| = 0$. Moreover if g^* or h^* are strictly convex on D, then $y^{k+1} = y^k$.
  In such a case DCA terminates at the kth iteration (finite convergence of DCA).

(2) If $\rho(g, C) + \rho(h, C) > 0$ (resp. $\rho(g^*, D) + \rho(h^*, D) > 0$) then the series $\{\|x^{k+1} - x^k\|^2$ (resp. $\{\|y^{k+1} - y^k\|^2\}$ converges.
(3) If the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^k\}$ and $\{y^k\}$ are bounded then every limit point x^∞ (resp. y^∞) of the sequence $\{x^k\}$ (resp. $\{y^k\}$) is a critical point of $g - h$ (resp. $h^* - g^*$).
(4) DCA has a linear convergence for general DC programs.
(5) Within the natural choice of x^{k+1} and y^k by

$$\begin{aligned} x^{k+1} &\in \arg\min\{g(x) - h(x) : x \in \partial g^*(y^k)\} \\ &= \arg\min\{\langle x, y^k \rangle - h(x) : x \in \partial g^*(y^k)\} \end{aligned}$$

and

$$\begin{aligned} y^k &\in \arg\min\{h^*(y) - g^*(y) : y \in \partial h(x^k)\} \\ &= \arg\min\{\langle x^k, y \rangle - g^*(y) : y \in \partial h(x^k)\} \end{aligned}$$

the previous results are improved: nonemptiness of intersection of subdifferentials will be replaced by inclusion of subdiffrentials:

$$\partial h(x^\infty) \subset \partial g(x^\infty) \text{ and } \partial g^*(y^\infty) \subset \partial h^*(y^\infty).$$

(6) For polyhedral DC programs, DCA has a finite convergence.

(7) An in-depth analysis of DCA:
Let h^ℓ and $(g^*)^\ell$ be the polyhedral convex functions (which underestimate the convex functions h and g^* respectively) defined by

$$h^\ell(x) \ : \ = \sup\{h_i(x) : i = 0, ..., \ell\}, \ \forall x \in \mathbb{R}^n \tag{7}$$

$$(g^*)^\ell(y) = \sup\{(g^*)_i(y) : i = 1, ..., \ell\}, \forall y \in \mathbb{R}^n. \tag{8}$$

Let $k := \inf\{\ell : g(x^\ell) - h(x^\ell) = g(x^{\ell+1}) - h(x^{\ell+1})\}$. Then there hold:

(i) If k is finite then the solution computed by DCA, x^{k+1} and y^k, are global minimizers for the polyhedral DC programs

$$\beta_k = \inf\{g(x) - h^k(x) : x \in \mathbb{R}^n\} \quad (P^k)$$

and

$$\beta_k = \inf\{h^*(y) - (g^*)^k(y) : x \in \mathbb{R}^n\} \quad (D^k)$$

respectively.

(ii) If $k = +\infty$ (i.e., $g(x^\ell) - h(x^\ell) > g(x^{\ell+1}) - h(x^{\ell+1})$ for every ℓ) then x^∞ and y^∞ are global minimizers for the polyhedral DC programs

$$\beta_\infty = \inf\{g(x) - h^\infty(x) : x \in \mathbb{R}^n\} \quad (P^\infty)$$

and

$$\beta_\infty = \inf\{h^*(y) - (g^*)^\infty(y) : x \in \mathbb{R}^n\} \quad (D^\infty)$$

respectively, where the convex functions h_∞, h^∞, $(g^*)_\infty$ and $(g^*)^\infty$ are given by

$$h_\infty(x) \ : \ = h(x^\infty) + \langle x - x^\infty, y^\infty\rangle = \langle x, y^\infty\rangle - h^*(y^\infty), \ \forall x \in \mathbb{R}^n$$
$$(g^*)_\infty(y) \ : \ = g^*(y^\infty) + \langle y - y^\infty, x^\infty\rangle = \langle y, x^\infty\rangle - g(x^\infty), \ \forall y \in \mathbb{R}^n$$
$$h^\infty(x) \ : \ = \sup\{h_i(x) : i = 0, ..., +\infty\}, \quad \forall x \in \mathbb{R}^n$$
$$(g^*)^\infty(y) = \sup\{(g^*)_i(y) : i = 1, ..., +\infty\}, \quad \forall y \in \mathbb{R}^n.$$

(iii) If either of the following conditions hold (k is finite or equal to $+\infty$)
+ the functions h^k and h coincide at some optimal solution to (P_{dc})
+ the functions $(g^*)^k$ and g^* coincide at some optimal solution to (D_{dc}) then x^k (resp. y^k) is also an optimal solution to (P_{dc}) (resp. (D_{dc}))

Remark 1 (i) A DC function f has infinitely many DC decompositions which have crucial impacts on the qualities (speed of convergence, robustness, efficiency, globality of computed solutions,...) of DCA.
(ii) In practice, DCA, once well suited to treated DC programs, handles the large-scale setting and converges quite often to global solutions.
(iii) Property 7 (i) holds especially in polyhedral DC programs where DCA has a finite convergence while 7 (ii) involves infinite convergence. The hidden features reside in 7): (k is finite or equal to $+\infty$).

+ x^{k+1} (resp. y^k) is not only the solution of (P_k) (resp. (D_k) but also the solution to the more tightly approximate problem (P^k) (resp. (D^k)).

+ $\beta_k + \varepsilon_k \leq \alpha \leq \beta_k$ where $\epsilon_k := \inf\{h^k(x) - h(x) : x \in \mathcal{P}\} \leq 0$ and the more ϵ_k is near zero (i.e., the more the polyhedral convex minorization h^k is close to h over $\mathcal{P}$), the more x^{k+1} is near $\mathcal{P}$.

We shall apply *all these DC enhancement features* to solve Problem (4).

3 DCA for solving the DC program (10)

In this section we will reformulate Problem (4) as an equivalent DC program and then apply DCA for its solution. We emphasize the finite convergence of DCA and its feasibility in related polyhedral DC programs.

3.1 Reformulation of the penalty equivalent as a DC program

Let $\theta : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^p \to \mathbb{R}$ be the function defined by

$$\theta(x, y, \lambda) := \sum_{i=1}^{p} \min\{\lambda_i, -(Dx + Ey + b)_i\}. \tag{9}$$

It is clear that the negation of θ is finite polyhedral convex (and so nonsmooth) function on $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^p$, and θ is nonnegative on the feasible set of Problem (4). Since DC functions are Lipschitz on bounded sets of their domains, it follows, from Z.Luo et al. [17] and Le Thi et al. [13,14], that if Problem (4) is feasible and the polyhedral convex set $\{(x, y) \in Z : Dx + Ey + b \leq 0\}$ is bounded, then there exist positive scalars τ_1 and c such that for all scalars $\tau > \tau_1$, Problem (4) is equivalent to the following penalized program

$$\begin{array}{lll} \alpha(\tau) := \min & F_\tau(x, y, \lambda) := f(x, y) + \tau\theta(x, y, \lambda) \\ \text{s.t.} & (x, y) \in Z \\ & Px + Qy + q + E^T\lambda = 0 \\ & Dx + Ey + b \leq 0 \\ & \lambda \geq 0 \\ & \|\lambda\|_\infty \leq c, \end{array} \tag{10}$$

where $\|\cdot\|_\infty$ stands for the infinity norm $\|\cdot\|_\infty := \max\{|\lambda_i| : i = 1, \ldots, p\}$.

The objective function of Problem (10) is nondifferentiable and nonconvex. It is actually a DC function and Problem (10) is a DC program. Note that if the objective function f of Problem (1) is convex or a DC function whose first DC component is polyhedral convex, then (10) is a polyhedral DC program for which DCA has a finite convergence.

First we have to present Problem (10) in the standard form of a DC program. Since the function f is DC (in the pair of variables (x, y)) on Z

$$f(x, y) = f_1(x, y) - f_2(x, y), \tag{11}$$

with f_1, f_2 being convex functions (in (x, y)) on Z.

The function F defined by

$$F(x, y, \lambda) := f(x, y)$$

is DC (in the triple variables $X = (x, y, \lambda)$) on $Z\times \mathbb{R}^p$ with the following DC decomposition

$$F(X) = F(x, y, \lambda) = F_1(x, y, \lambda) - F_2(x, y, \lambda),$$

where F_1, F_2 are the following convex functions (in (x, y, λ)) on $Z\times \mathbb{R}^p$

$$F_1(x, y, \lambda) := f_1(x, y), \quad F_2(x, y, \lambda) := f_2(x, y).$$

By assumption, the feasible set C of (10) is a bounded polyhedral convex set of $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^p$. Its indicator function χ_C is defined by

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

With the concavity of the function θ, Problem (10) can be rewritten as the following DC program

$$\min\{G(x, y, \lambda) - H(x, y, \lambda) : (x, y, \lambda) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^p\}, \tag{12}$$

with G, H being convex functions on $Z\times \mathbb{R}^p$ defined by $(\tau \geq \tau^*)$

$$G := F_1 + \chi_C, \qquad H := F_2 - \tau\theta.$$

Recall that if the function f is convex, then $f_2 = 0$. In this case, H is a polyhedral convex function and Problem (12) is a polyhedral DC program.

According to Sect. 3.1, performing DCA for Problem (12) amounts to computing the two sequences $\{(x^k, y^k, \lambda^k)\}$ and $\{(u^k, v^k, w^k)\}$ defined by

$$(u^k, v^k, w^k) \in \partial H(x^k, y^k, \lambda^k), \quad (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \partial G^* (u^k, v^k, w^k).$$

In other words we have to compute the subdifferentials ∂H and ∂G^*.

As usually, ∂H is often explicitly computed with the help of known rules in convex analysis. Here we have

$$\partial H(x, y, \lambda) = \partial F_2(x, y, \lambda) + \tau\partial(-\theta)(x, y, \lambda), \tag{13}$$

with the explicit computation of $\partial(-\theta)$. Indeed, since

$$\begin{aligned}
-\theta(x, y, \lambda) &= \sum_{i=1}^{p} \max\{-\lambda_i, (Dx + Ey + b)_i\} \\
&= \sum_{i=1}^{p} \max\{(-e^i)^T\lambda, (e^i)^T(Dx + Ey + b)\} \\
&= \sum_{i=1}^{p} \max\{(-e^i)^T\lambda, (D^Te^i)^Tx + (E^Te^i)^Ty + (e^i)^Tb\},
\end{aligned}$$

where $e^1, \ldots, e^p$ are the unit vectors of $\mathbb{R}^p$, it follows that

$$\partial(-\theta)(x, y, \lambda) = \begin{cases} \{(0, 0, -e^i)\} & \text{if } \lambda_i < -(D\overline{x} + Ey + b)_i, \\ \{(D^Te^i, E^Te^i, 0)\} & \text{if } \lambda_i > -(Dx + Ey + b)_i, \\ [(0, 0, -e^i),\ (D^Te^i, E^Te^i, 0)] & \text{if } \lambda_i = -(Dx + Ey + b)_i. \end{cases} \tag{14}$$

So if the computation of ∂F_2 is explicit then that of ∂H is explicit too.

As for computing $\partial G^*(u, v, t)$, we have in general to solve the following related convex program:

$$\min\{F_1(x, y, \lambda) - \langle (x, y, \lambda), (u, v, t)\rangle : (x, y, \lambda) \in C\}. \tag{15}$$

We are now in a position to give the DCA scheme to solving Problem (10). It suffices to particularize the calculation of ∂h and ∂g^* in the general scheme.

3.1.1 DCA for solving Problem (10)

It consists of the construction of two sequences $\{X^k = (x^k, y^k, \lambda^k)\}$ and $\{Y^k = (u^k, v^k, t^k)\}$ as follows:

1. Choose an initial point $X^1 = (x^k, y^k, \lambda^k)$ not necessarily in C. Set $k = 1$ and let ϵ_1 and ϵ_2 be sufficiently small positive numbers.
2. Computer $Y^k = (u^k, v^k, t^k) \in \partial H(X^k)$ by using (13) and (14).
3. Compute $X^{k+1} \in \partial g^*(G^k)$, by solving the convex program (15) with $(u, v, t) = (u^k, v^k, t^k)$.
4. If either

$$\left\| X^{k+1} - X^k \right\| \le \epsilon_1 (1 + \left\| X^k \right\|)$$

or

$$\left|(F + \tau\theta)(X^k) - (F + \tau\theta)(X^{k+1})\right| \le \epsilon_2 (\left|(F + \tau\theta)(X^k)\right| + 1)$$

then stop and $X^{k+1}$is the computed solution. Otherwise, set $k = k + 1$ and go to Step 2.

3.1.2 Special feature of DCA for Problem (10) in case f is a polyhedral DC function

It is crucial for local algorithms applied to the penalty equivalent (10) to provide feasible solutions $X^* = (x^*, y^*, \lambda^*)$ of the original nonlinear bilevel problem (4), i.e., $\theta(X^*) = 0$. We shall prove that DCA bears this feature in case the objective function $f = g - h$ of (4) is a polyhedral DC function with g being polyhedral convex. Consider first the case $g = 0$, i.e., $f = -h$.

Proposition 2 *If the objective function f of the nonlinear bilevel program* (4) *is a concave function, then there is a positive constant τ_2 such that the following holds:*

For $\tau > \max(\tau_{1,}\tau_2)$ let $\{X^j = (x^j, y^i, \lambda^j)\}$ generated by DCA applied to the penalty equivalent (10) *with penalty parameter τ_1. Then there hold*

(i) the sequence $\{X^i = (x^i, y^i, \lambda^i)\}$ is finite and contained in the vertex set $V(C)$ of the feasible set of (4).

(ii) both sequences $\{ f(x^i, y^i) + \tau\theta(X^i)\}$ and $\{\theta(X^i)\}$ are decreasing

Proof (i) The objective function of (10) then is concave function and (10) is a polyhedral DC program. It follows that the sequence $\{X^i = (x^i, y^i, \lambda^i)\}$ generated by DCA can be taken in $V(C)$.

(ii) If $V(C)$ is contained in the feasible solution set of (10) then the assertion is trivial with $\tau_2 = 0$. Otherwise let

$$\xi := \min\left\{\theta(X') - \theta(X) : (X, X') \in V(C) \times V(C), \theta(X') > \theta(X)\right\},$$
$$\eta := \max\{F(X') - F(X) : (X', X) \in V(C) \times V(C)\};$$

then $0 < \xi < +\infty$ and $0 \leq \eta < +\infty$ since $V(K)$ is a finite. Consider now the nonnegative number τ_2 defined by

$$\tau_2 := \frac{\xi}{\eta},$$

and $\tau > \tau_2$. Let $\{X^k\}$ be the sequence generated by DCA applied to (10) with this value τ. Assume for contradiction that there is $r \geq 1$ such that $\theta(X^{r+1}) > \theta(X^r)$. Then

$$\tau[\theta(X^{r+1}) - \theta(X^r)] > \tau_2[\theta(X^{r+1}) - \theta(X^r)] = \frac{\xi}{\eta}[\theta(X^{r+1}) - \theta(X^r)] \geq \xi.$$

Hence

$$\tau[\theta(X^{r+1}) - \theta(X^r)] \geq F(X^r) - F(X^{r+1}),$$

i.e.,

$$F(X^{r+1}) + t\theta(X^{r+1}) > F(X^r) + \tau\theta(X^r),$$

that contradicts the decrease of the sequence $\{F(X^k) + \tau\theta(X^k)\}$.

□

Remark 3 The proof of Proposition 2 is based on the finiteness of $V(C)$ that contains the sequences $\{X^j = (x^j, y^i, \lambda^j)\}$ generated by DCA applied to (10) with every $\tau \geq 0$. Hence it remains valid in the case $f = f_1 - f_2$ (11) with f_1 being a polyhedral convex function on Z.

Extension of this proposition to general DC functions f is still an open problem. In the case the function f is convex, the sequence $\{X^j = (x^j, y^i, \lambda^j)\}$ generated by DCA applied to (10) with every $\tau \geq 0$, is still finite but all these sequences depending on τ are no more contained in a finite subset of C independent of τ. One has in such a case the following weaker result: For $\tau \geq 0$, let

$$k := \arg\min\{j \geq 1 : F(X^{j+1}) + \tau\theta(X^{j+1}) = F(X^j) + \tau\theta(X^j)\}, \tag{16}$$
$$\tau_2 := \max\left\{\frac{f(x^j, y^j) - f(x^{j+1}, y^{j+1})}{\theta(X^{j+1}) - \theta(X^j)} : j = 1, \ldots, k \text{ and } \theta(X^{j+1}) > \theta(X^j)\right\}.$$

If $\tau > \tau_2$, then both sequences $\{f(x^i, y^i) + \tau\theta(X^i)\}$ and $\{\theta(X^i)\}$ are decreasing.

As a result of Remark 3 and local optimality conditions in polyhedral DC programming, we get

Corollary 4 *Under the assumptions of Proposition 2 and (16), if f is convex then the sequence $\{X^i = (x^i, y^i, \lambda^i)\}$ generated by DCA applied to the penalty equivalent (10) satisfies the following properties:*

- (i) *If $X^\ell = (x^\ell, y^\ell, \lambda^\ell)$, for some $\ell \geq 1$, is feasible for (4), then $X^i = (x^i, y^i, \lambda^i)$, for every $i \geq \ell$, is feasible too and in this case the solution $X^* = (x^*, y^*, \lambda^*) = X^k$, computed by DCA, is a feasible solution of (4). Moreover the finite sequence of objective values $\{f(x^i, y^i) : i = \ell, \ldots, k\}$ for (4) is strictly decreasing and $f(x^{k+1}, y^{k+1}) = f(x^k, y^k)$.*
- (ii) *If the solution $X^* = (x^*, y^*, \lambda^*)$ computed by DCA is feasible for (4) and verifies the strict complementarity condition then it is a local minimizer for (10).*

Proof We have only to prove (ii). According to (13) and (14) the function $H = \tau(-\theta)$ is differentiable at $X^* = (x^*, y^*, \lambda^*)$ if and only if $\lambda_i > -(Dx + Ey + b)_i$ for $i = 1, \ldots, p$, i.e., the feasible solution $X^* = (x^*, y^*, \lambda^*)$ of (4) verifies the strict complementarity conditions. Since (10) is a polyhedral DC program, such an $X^* = (x^*, y^*, \lambda^*)$ is a local minimizer for (10) (See Sect. 2). □

Roughly speaking, the resulting DCA applied to (10) works in fact with the original problem (4) from the ℓ^{th} iteration if $X^\ell = (x^\ell, y^\ell, \lambda^\ell)$ is feasible to the latter one and it decreases the objective values of (4). This property is worthy to note.

3.2 Initial point and strategies of launching DCA

From the above results, it is important to find a good initial points for DCA (not only in the case DC polyhedral) applied to (10). For this, we use again DCA applied to the concave minimization problem

$$0 = \min \quad \theta(x, y, \lambda) \tag{17}$$

$$\text{subject to} \qquad (x, y) \in Z$$

$$Px + Qy + q + E^T y = 0$$

$$Dx + Ey + b \leq 0$$

$$\lambda \geq 0, \|\lambda\|_\infty \leq c$$

Problem (17) is a polyhedral DC program with known optimal value and whose solution set is exactly the feasible set (4). Fortunately, as for linear complementarity problems [11], DCA, with starting point $\overline{X} = (\overline{x}, \overline{y}, \overline{\lambda})$ not necessarily feasible but such that $\theta(\overline{x}, \overline{y}, \overline{\lambda}) = 0$, converges, almost always in practice, to a global solution of (17).

When is DCA restarted? During the algorithm, we can restart DCA to improve the current best upper bounds. As usual, an upper bound obtained when a feasible solution of the Problem (4) found, we call this upper bound as $Score$. During the algorithm, to improve the $Score$, we will restart DCA when

$$F_\tau(\tilde{X}) \leq Score(1 + 1.e - 2) \tag{18}$$

where $\tilde{X} = (\tilde{x}, \tilde{y}, \tilde{\lambda})$ is an optimal solution of the current problem estimating lower bound.

However, by using the exact penalty technique, we have another upper bound. From the equivalence of two problems (4) and (10) we can take $F_\tau(x, y, \lambda)$ as an upper bound if (x, y, λ) is a feasible point of (10) (not necessarily feasible point of (4)), we call this upper bound as UB_f (in the case (x, y, λ) is not feasible to (4)). So in our algorithm, in order to reduce the number of restart DCA, we can restart when the following condition is satisfied

$$F_\tau(\tilde{X}) \leq \min\{Score, UB_f\}(1 + 1 \cdot e - 2) \tag{19}$$

Remark 5 Since the value of τ is large, so $Score$ is often smaller than UB_f.

To check globality of solutions computed by DCA or to find better solutions for restarting the algorithm, we have recourse to global optimization techniques that we will present below.

4 Global optimization algorithm

4.1 Reformulation of Problems (4) and (10)

To establish a global algorithm for solving Problem (4), we define a new vector of variables $w \in \mathbb{R}^p$ by

$$w = -Dx - Ey - b,$$

and formulate Problems (4) equivalently as

$$\begin{array}{ll} \alpha := \min & f(x, y) \\ \text{s.t.} & (x, y) \in Z \\ & Px + Qy + E^T\lambda + q = 0 \\ & Dx + Ey + w + b = 0 \\ & \lambda^T w = 0 \\ & \lambda \geq 0 \\ & w \geq 0. \end{array} \tag{20}$$

Using the number c in Problem (10) and the assumption that the set

$$\{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : (x, y) \in Z,\ \ Dx + Ey + b \leq 0\}$$

is nonempty and bounded, we can construct two rectangles

$$\begin{array}{l} R = \{\lambda \in \mathbb{R}^p : 0 \leq \zeta \leq \lambda \leq \xi < +\infty\},\ \text{and} \\ S = \{w \in \mathbb{R}^p : 0 \leq \ell \leq w \leq u < +\infty\}, \end{array} \tag{21}$$

such that Problem (20) can be rewritten as

$$\begin{array}{ll} \alpha := \min & f(x, y) \\ \text{s.t.} & (x, y) \in Z \\ & Px + Qy + E^T\lambda + q = 0 \\ & Dx + Ey + w + b = 0 \\ & \lambda_j w_j = 0,\ \ j = 1, \ldots, p \\ & \lambda \in R \\ & w \in S. \end{array} \tag{22}$$

Accordingly, Problem (10) can be reformulated as

$$\begin{array}{ll} \alpha(\tau) := \min & f(x, y) + \tau \sum_{j=1}^{p} \min\{\lambda_j, w_j\} \\ \text{s.t.} & (x, y) \in Z \\ & Px + Qy + E^T\lambda + q = 0 \\ & Dx + Ey + w + b = 0 \\ & \lambda \in R \\ & w \in S. \end{array} \tag{23}$$

Obviously, Problems (4) and (22) are equivalent in the following sense:

(i) If (x^*, y^*) is an optimal solution of (4), then there exist λ^* and w^* such that $(x^*, y^*, \lambda^*, w^*)$ is an optimal solution of (22), and

(ii) If $(x^*, y^*, \lambda^*, w^*)$ is an optimal solution of (22), then (x^*, y^*) is an optimal solution of (4).

The rectangles R and S can be constructed by computing the vectors ζ, ξ and ℓ, u, respectively, as follows.

$$\zeta_j = \max\{0, \ \min\{\lambda_j : (x, y) \in Z, \ Dx + Ey + b \leq 0, \ Px + Qy + E^T\lambda + q = 0, \\ c \geq \lambda_i \geq 0 \ \text{ for } i \neq j\}\},$$

$$\xi_j = \max\{\lambda_j : (x, y) \in Z, \ Dx + Ey + b \leq 0, \ Px + Qy + E^T\lambda + q = 0, \\ c \geq \lambda_i \geq \zeta_i \ \text{ for } i = 1, 2, \ldots, p\},$$

$$\ell_j = \max\{0, \ \min\{(-Dx - Ey - b)_i : \ (x, y) \in Z, \ Dx + Ey + b \leq 0, \\ Px + Qy + E^T\lambda + q = 0, \ \xi \geq \lambda \geq \zeta\}\},$$

$$u_j = \max\{(-Dx - Ey - b)_i : \ (x, y) \in Z, \ Dx + Ey + b \leq 0, \\ Px + Qy + E^T\lambda + q = 0, \\ \xi \geq \lambda \geq \zeta\}.$$

Remark 6 In principle, c should be approximated by a large number. In some special cases, if the set

$$\Omega = \{(x, y, \lambda) \in \mathbb{R}^{n_1+n_2+p} : \\ (x, y) \in Z, \ Dx + Ey + b \leq 0, \ Px + Qy + E^T\lambda + q = 0, \hat{\lambda} \geq 0\}$$

is compact, then the rectangles R and S can be simply computed without using the number c.

As mentioned in the introduction, we assume in the paper that the polyhedral convex set

$$Z' := \{(x, y) \in Z : Dx + Ey + b \leq 0\}$$

is nonempty and bounded. It is clear that the boundedness of Z' holds if either Z is bounded or the objective function f is coercive, i.e.

$$\lim_{\|(x,y)\| \to +\infty} f(x, y) = +\infty.$$

On the other hand, if Z' is bounded, then the polyhedral convex set Ω is bounded if and only if its recession cone $0^+\Omega$ given by [22,7]:

$$0^+\Omega = \{(0, 0, \nu) : E^T\nu = 0, \nu \geq 0\}$$

is reduced to $\{0, 0, 0\}$.

In what follows we establish an algorithm for Problem (22) in the sense of global optimization. This is a combination of the DCA presented in the previous section and the well known branch and bound scheme. Each branch and bound algorithm consists mainly of two basic operations: bound estimation and branching procedure. We discuss these basic operations before presenting the algorithm in detail.

4.2 Lower bound estimation

Our method for estimating lower bounds is based on the following problem. Let I and J be two subsets of the index set $\{1, \ldots, p\}$. Compute a lower bound for the optimal value of the following optimization problem, denoted by *P(IJ)*:

$$
\begin{array}{ll}
\min & f(x,y) \\
\text{s.t.} & (x,y) \in Z \\
& Px + Qy + E^T\lambda + q = 0 \\
& Dx + Ey + w + b = 0 \\
& \lambda_j = 0,\ \ j \in J \\
& w_j = 0,\ \ j \in I \\
& \lambda_j w_j = 0,\ \ j \in \{1,\ldots,p\}\backslash(I \cup J) \\
& \lambda \in R,\ \ w \in S.
\end{array} \qquad P(IJ)
$$

To this purpose, denote $K = \{1,\ldots,p\}\backslash(I \cup J)$, and define for each $j \in K$ the triangle T_j in $\mathbb{R}^2$ having the vertices $(0,0)$, $(\xi_j, 0)$ and $(0, u_j)$, i.e.,

$$
T_j = \{(\lambda_j, w_j) : \frac{\lambda_j}{\xi_j} + \frac{w_j}{u_j} \leq 1,\ \lambda_j \geq 0,\ w_j \geq 0\}. \tag{24}
$$

Proposition 7 *A lower bound $\mu(IJ)$ of the optimal value of Problem $P(IJ)$ can be computed by*

$$
\begin{array}{lll}
\mu(IJ) := & \min & f(x,y) \\
& \textit{s.t.} & (x,y) \in Z \\
& & Px + Qy + E^T\lambda + q = 0 \\
& & Dx + Ey + w + b = 0 \\
& & \lambda_j = 0,\ \ j \in J \\
& & w_j = 0,\ \ j \in I \\
& & (\lambda_j, w_j) \in T_j,\ \ j \in K.
\end{array} \qquad C(IJ) \tag{25}
$$

(We agree to set $\mu(IJ) = +\infty$ if Problem $C(IJ)$ is infeasible).

Proof Consider constraints $\lambda_j w_j = 0,\ (j = 1,\ldots,p),\ \lambda \in R,\ w \in S$ in Problem $P(IJ)$.

For each $j \in K = \{1, ..., p\}\backslash(I \cup J)$, constraints

$$
\lambda_j w_j = 0,\ \lambda \in R,\ w \in S
$$

is equivalent to constraints

$$
\lambda_j = 0 \text{ or } w_j = 0,\ \lambda \in R,\ w \in S. \tag{26}
$$

Obviously, T_j is the convex hull of all points (λ_j, w_j) satisfying (26).

From this, it follows that the feasible set of Problem $P(IJ)$ is contained in that one of Problem $C(IJ)$. Therefore, the proposition follows. □

Remark 8 One can without difficulty show that the DC equivalent formulation of Problem P(IJ) reads

$$
\begin{array}{lll}
\mu(IJ) := & \min & f(x,y) + \tau \sum_{j \in K} \min\{\lambda_j, w_j\} \\
& \text{s.t.} & (x,y) \in Z \\
& & Px + Qy + E^T\lambda + q = 0 \\
& & Dx + Ey + w + b = 0 \\
& & \lambda_j = 0,\ \ j \in J \\
& & w_j = 0,\ \ j \in I \\
& & (\lambda_j, w_j) \in T_j,\ \ j \in K.
\end{array} \tag{27}
$$

For each $j \in K$, if we replace the concave function $\min\{\lambda_j, w_j\}$ by its convex envelope (best convex subfunction) $\varphi_j(\lambda_j, w_j)$ on the simplex T_j, then the optimal value of the resulting problem yields a lower bound for the optimal value of Problem (27). It is well known (cf. e.g., [15,16]) that $\varphi_j(\lambda_j, w_j)$ is an affine function satisfying $\varphi_j(\lambda_j, w_j) = \min\{\lambda_j, w_j\} = 0$ at the vertices of T_j. Thus, the resulting problem is exactly Problem C(IJ).

Remark 9 Let (IJ) and $(I'J')$ be two pairs of index sets such that $I \subseteq I'$ and $J \subseteq J'$. Then it is clear that $\mu(IJ) \leq \mu(I'J')$. This monotonicity property is useful within a branch and bound procedure.

Remark 10 If $f(x, y)$ is a convex function, then Problem C(IJ) is a convex minimization over a polyhedral convex set. In general, $f(x, y)$ is a DC function given by

$$f(x, y) = f_1(x, y) - f_2(x, y)$$

with f_1, f_2 being convex functions. A promising technique for computing lower bounds of the optimal value of Problem C(IJ) is to replace the concave function $-f_2(x, y)$ by its convex envelope on some polyhedral sets having simple structures (e.g., simplices or rectangles). Methods for constructing convex envelopes for different special cases of $f_2(x, y)$ can be found e.g. in [8,9,11,15,16] and references therein.

4.3 Case f is a DC function

Rewriting as in Sect. 3.1. $f(x, y) = f_1(x, y) - f_2(x, y)$ where f_1 and f_2 are two convex functions on Z. In this case, the optimal value of Problem C(IJ) is still a lower bound for (4), but Problem C(IJ) becomes now a DC program. To compute a lower bound for Problem (4) we will introduce the following relaxation technique.

For the sake of simplicity, we shall restrict ourselves to the case (ρ being some positive number)

$$f_2(x, y) := \frac{\rho}{2} \|(x, y)\|^2 = \frac{\rho}{2} \|x\|^2 + \frac{\rho}{2} \|y\|^2$$

This important class of DC functions is large enough and contains $\mathcal{C}^{1,1}(Z)$ (the space of differentiable functions whose derivatives are Lipschitz on Z). It is worth noting that $\mathcal{C}^{1,1}(Z)$ contains the more usual space $\mathcal{C}^2(O)$,(the space of twice continuously differentiable functions on an open convex set O containing Z), and every $f \in \mathcal{C}^2(O)$ is a DC function on Z with the two following natural useful DC decompositions:

$$f = \left[f + \frac{\rho}{2} \|\cdot\|^2\right] - \frac{\rho}{2} \|\cdot\|^2 \tag{28}$$

with $\rho \geq 0$ such that the function $\left[f + \frac{\rho}{2} \|\cdot\|^2\right]$ be convex on K. This condition is satisfied if $\rho \geq -\lambda_1(\nabla^2 f(x, y)), \forall (x, y) \in O_K$ where $\lambda_1(\nabla^2 f(x, y))$ denotes the smallest eigenvalue of $\nabla^2 f(x, y))$ and $O_K \subset O$ is a closed bounded set which contains an open convex set containing K, i.e.,

$$\rho \geq \sup\{-\lambda_1(\nabla^2 f(x, y)) : (x, y) \in O_K\} = \rho_1$$

Likewise f can take another DC decomposition

$$f = \frac{\rho}{2} \|\cdot\|^2 - \left[\frac{\rho}{2} \|\cdot\|^2 - f\right], \tag{29}$$

with $\rho \geq 0$ such that the function $\left[\frac{\rho}{2}\|\cdot\|^2 - f\right]$ be convex on K.This condition is satisfied if ($\lambda_n(\nabla^2 f(x,y))$ is the greatest eigenvalue of $\nabla^2 f(x,y)$))

$$\rho \geq \sup\{\lambda_n(\nabla^2 f(x,y)) : (x,y) \in O_K\} = \rho_n.$$

In particular if f is a quadratic (nonconvex) function

$$f(x,y) = \frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T H \begin{pmatrix} x \\ y \end{pmatrix} + \langle c, (x,y)\rangle$$

then $\rho_1 = -\lambda_1(H)$ and $\rho_n = \lambda_n(H)$.

Both DC decompositions (28), (29) can be used for DCA, that leads to solving a convex quadratic program at each iteration. As for B&B, the second DC decomposition will be preferred because the separabity of its second DC component with respect to the variables (x, y) makes possible an explicit computation of convex underestimation of f on Z, as it will be shown below.

Indeed, since the set $\{(x,y) \in Z : Dx + Ey + b \leq 0\}$ is bounded, problem C(IJ) can be rewritten, with the help using the DC decomposition (28), in the form of a DC program as follows:

$$\begin{aligned}
\min\ & f_1(x,y) - \left[\frac{\rho}{2}\|x\|^2 + \frac{\rho}{2}\|y\|^2\right] \\
\text{s.t.}\ & (x,y) \in Z,\ m^x \leq x \leq M^x, m^y \leq y \leq M^y \\
& Px + Qy + E^T\lambda + q = 0 \\
& Dx + Ey + w + b = 0 \\
& \lambda_j = 0,\ \ j \in J \\
& w_j = 0,\ \ j \in I \\
& (\lambda_j, w_j) \in T_j,\ \ j \in K.
\end{aligned} \tag{30}$$

where the bounds $m^x, M^x \in \mathbb{R}^{n_1}$and $m^y, M^y \in \mathbb{R}^{n_2}$ are defined by

$$\begin{aligned}
m_j^x &= \min\{x_j : (x,y) \in Z, Dx + Ey + b \leq 0, Px + Qy + E^T\lambda + q = 0, \\
&\qquad\qquad 0 \leq \lambda_j \leq c,\ \ j = 1, ..., p\} \\
M_j^x &= \max\{x_j : (x,y) \in Z, Dx + Ey + b \leq 0, Px + Qy + E^T\lambda + q = 0, \\
&\qquad\qquad 0 \leq \lambda_j \leq c,\ \ j = 1, ..., p\} \\
m_j^y &= \min\{y_j : (x,y) \in Z, Dx + Ey + b \leq 0, Px + Qy + E^T\lambda + q = 0, \\
&\qquad\qquad 0 \leq \lambda_j \leq c,\ \ j = 1, ..., p\} \\
M_j^y &= \max\{y_j : (x,y) \in Z, Dx + Ey + b \leq 0, Px + Qy + E^T\lambda + q = 0, \\
&\qquad\qquad 0 \leq \lambda_j \leq c,\ \ j = 1, ..., p\}.
\end{aligned} \tag{31}$$

Here the second DC decomposition of f is $f_2(x,y) := \left[\frac{\rho}{2}\|x\|^2 + \frac{\rho}{2}\|y\|^2\right]$, and f_1+ co$(-f_2, [m^x, M^x] \times [m^y, M^y])$, where co$(-f_2, [m^x, M^x] \times [m^y, M^y])$ denotes the convex envelope of $-f_2$ on $[m^x, M^x] \times [m^y, M^y]$, is a convex underestimation of f on Z. On the other hand for $\omega, \nu \in \mathbb{R}^{n_1}$, $\omega < \nu$ and $\xi, \eta \in \mathbb{R}^{n_2}$, $\xi < \eta$, we have [9,18,20,21]:

$$\mathrm{co}(-f_2, [\omega, \nu] \times [\xi, \eta])(x, y) = \frac{\rho}{2} \sum_{i=1}^{n_1} \mathrm{co}(-x_i^2, [\omega_i, \nu_i]) + \frac{\rho}{2} \sum_{j=1}^{n_2} \mathrm{co}(-y_j^2, [\xi_j, \eta_j]) \quad (32)$$

and

$$\begin{aligned} \mathrm{co}(-x_i^2, [\omega_i, \nu_i])(x_i) &= -(\omega_i + \nu_i)x_i + \omega_i \nu_i, \quad i = 1, ..., n_1 \\ \mathrm{co}(-y_j^2, [\xi_j, \eta_j])(y_j) &= -(\xi_j + \eta_j)y_j + \xi_j \eta_j, \quad j = 1, ..., n_2. \end{aligned}$$

That leads to the lower bound for (4) given as the optimal value of the convex program

$$\begin{aligned} \mu(IJ) := \min \ \ & \Psi(x, y, \lambda, w) := f_1(x, y) - \frac{\rho}{2} \sum_{i=1}^{n_1} \left[(m_i^x + M_i^x)x_i + m_i^x M_i^x\right] \\ & \qquad -\frac{\rho}{2} \sum_{j=1}^{n_2} \left[(m_j^y + M_j^y)y_j - m_j^y M_j^y\right] \\ \text{s.t.} \ \ & (x, y) \in Z, \ m^x \le x \le M^x, m^y \le y \le M^y \\ & Px + Qy + E^T\lambda + q = 0 \\ & Dx + Ey + w + b = 0 \\ & \lambda_j = 0, \ j \in J \\ & w_j = 0, \ j \in I \\ & (\lambda_j, w_j) \in T_j, \ j \in K = \{1, \ldots, p\} \backslash (I \cup J) \\ & m^x \le x \le M^x, m^y \le y \le M^y \end{aligned} \quad (33)$$

(We also agree to set the optimal value equal to $+\infty$ if Problem (33) is infeasible).

In the sequel, for simplicity of notations, for two real numbers $u < v$ we consider the affine function $\phi : \mathbb{R} \to \mathbb{R}$ defined by

$$\phi(t) := -\frac{\rho}{2}[(u + v)t - uv] \quad (34)$$

On the other hand, one must note that DCA will be applied to DC programs of the form which only involve three variables (x, y, λ)

$$\begin{aligned} \alpha(\tau) = \min \ \ & f_1(x, y) - \left[\frac{\rho}{2} \|x\|^2 + \frac{\rho}{2} \|y\|^2\right] + \tau \sum_{i=1}^{p} \min\{\lambda_i, -(Dx + Ey + b)_i\} \\ \text{s.t.} \ \ & (x, y) \in Z \\ & Px + Qy + E^T\lambda + q = 0 \\ & Dx + Ey + w + b = 0 \\ & \lambda_j = 0, \ j \in J \\ & D_j x + E_j y + b_j = 0, \ j \in I \\ & \zeta_j \le \lambda_j \le \xi_j, \ j \in K = \{1, \ldots, p\} \backslash (I \cup J). \end{aligned} \quad (35)$$

where D_j, E_j are the jth rows of the matrices D and E respectively and b_j is the j^{th} component of the vector b.

4.4 Problem bisection

The algorithm begins with problem (22) which is rewritten as

$$\begin{array}{ll} \min & f(x,y) \\ \text{s.t.} & (x,y) \in Z \\ & x \in B_1^x,\ y \in B_1^y \\ & Px + Qy + E^T\lambda + q = 0 \\ & Dx + Ey + w + b = 0 \\ & \lambda_j = 0, \quad j \in J_1 \\ & w_j = 0, \quad j \in I_1 \\ & \lambda_j w_j = 0, \quad j \in \{1,\ldots,p\}\backslash(I_1 \cup J_1) \\ & \lambda \in R;\ w \in S \end{array} \qquad (P_1)$$

$$\text{where} \quad \begin{cases} B_1^x = \left[m^x, M^x\right],\ B_1^y = \left[m^y, M^y\right] \\ I_1 = \{j \in \{1,\ldots,p\} : \zeta_j > 0\}, \\ J_1 = \{j \in \{1,\ldots,p\} : \ell_j > 0\}. \end{cases} \qquad (36)$$

In general, let (P_l) be a problem generated within the algorithm with the boxes B_l^x, B_l^y and the sets I_l, j_l genertated by the branching procedure. By solving the corresponding relaxed Problem (33) we obtain a lower bound for the optimal value of Problem (P_l).

Let $(x^l, y^l, \lambda^l, w^l)$ be an optimal solution of the corresponding relaxed Problem (33).

Denote $\overline{K} := \{j \in K : \lambda_j^l w_j^l > 0\}$

Calculate

$$\delta_1 = \max_{j=1,2,\ldots,n_1} \left\{-\frac{\rho}{2}(x_j^l)^2 - \phi(x_j^l)\right\}, \max\ \text{at}\ j_x$$

$$\delta_2 = \max_{j=1,2,\ldots,n_2} \left\{-\frac{\rho}{2}(y_j^l)^2 - \phi(y_j^l)\right\}, \max\ \text{at}\ j_y$$

$$\delta_3 = \max_{j \in \overline{K}} \left\{\tau \min\{\lambda_j^l, w_j^l\}\right\}, \max\ \text{at}\ \overline{j}$$

Based on well known $\omega-$subdivision for the B&B scheme [9,18,20,21] and taking into account the linear complementarity constraints, our branching procedure is defined by the maximum of these quantities as follows:

- δ_1 is the maximum : Subdividing $B_l^{x^i}$ into two subrectangles $B_{l_1}^{x^i} := \{x \in B_l^{x^i} : x_{j_x} \leq x_{j_x}^l\}$ and $B_{l_2}^{x^i} := \{x \in B_l^{x^i} : x_{j_x} \geq x_{j_x}^l\}$ to get two subproblems (P_{l1}) and (P_{l2}).
- δ_2 is the maximum : Subdividing $B_l^{y^i}$ into two subrectangles $B_{l_1}^{y^i} := \{y \in B_l^{y^i} : y_{j_y} \leq y_{j_y}^l\}$ and $B_{l_2}^{y^i} := \{y \in B_l^{y^i} : y_{j_y} \geq y_{j_y}^l\}$ to get two subproblems (P_{l1}) and (P_{l2})
- δ_3 is the maximum : Problem (P_l) is replaced by two problems (P_{l1}) and (P_{l2}) with

$$\lambda_{\overline{j}}^l = 0 \quad : \quad I_{l1} = I_l,\ J_{l1} = J_l \cup \{\overline{j}\},\ \text{and}$$
$$w_{\overline{j}}^l = 0 \quad : \quad I_{l2} = I_l \cup \{\overline{j}\},\ J_{l2} = J_l.$$

Note that we make only one bisection (to be chosen) in case there is equality among the three quantities δ_1, δ_2 and δ_3.

Throughout the algorithm, we say that "Problem (P_l) is divided into two problems (P_{l1}) and (P_{l2}) by a problem bisection".

Remark 11 If f is convex, the bisection problem works only with λ and w and the number of bisections is finite.

As displayed above, our approaches DCA and B&B (and so their combination GOA) can be applied to general DC program (10). For the B&B scheme, it is pointed out in Remark 10 that techniques of computing lower bounds depend on structures of convex sets used in branching procedure. In the next we will describe our approaches with rectangular subdivisions which often require the separabilty of $-f_2(x, y)$ with respect to its variables. More exactly we consider the class of DC functions $f(x, y)$ presented in Sect. 4.3 since the algorithm has the same form for other cases.

Note also for this class we can use the poweful code CPLEX to solve related convex quadratic programs in DCA, B&B and their combination **GOA**.

4.5 The algorithm

Based on basic operations discussed in the previous subsections, we establish an algorithm for computing a global optimal solution of Problem (22).

Global Optimization Algorithm (GOA):

Initialization.

Construct two rectangles B^x, B^y, and the triangles T_j, $j \in K$ by (34). Define the rectangles B_1^x, B_1^y and the subsets I_1, J_1 by (36). Let $R_0 = \{P_1\}$ and solve the corresponding relaxed problem (33) to obtain an optimal solution $(x^1, y^1, \lambda^1, w^1)$ and the optimal value as the first lower bound $\mu^0 := \mu(R_0)$

If the linear complementarity constraints are satisfied, i.e.,

$$\lambda^1 w^1 = 0, \ \forall j \in \{1, \ldots, p\} \setminus \{I_1 \cup J_1\},$$

and the value objective $f(x^1, y^1) = \mu^0$ then STOP, $(x^1, y^1, \lambda^1, w^1)$ is an optimal solution and μ^0 is the optimal value.

Else, apply DCA to DC program (35) (from the starting point being the obtained solution of DCA applied to (17)) to get $(x_\tau^1, y_\tau^1, \lambda_\tau^1)$. Set $w_\tau^1 := -Dx_\tau^1 - Ey_\tau^1 - b$.

If $(\lambda_\tau^1)^T w_\tau^1 = 0$ then

Set

$$(x^*, y^*, \lambda^*, w^*) = (x_\tau^1, y_\tau^1, \lambda_\tau^1, w_\tau^1),$$
$$\gamma^0 = f(x_\tau^1, y_\tau^1) \quad (Score),$$

Else, set $\tilde{\gamma}^0 := F_\tau(x_\tau^1, y_\tau^1, \lambda_\tau^1) \quad (UB_f)$.

Set $\mathcal{R} \leftarrow \{R_0\}$, $k \leftarrow 0$.

Iteration $k \geq 0$.

Select R_k such that $\mu^k = \mu(R_k) = \min\{\beta(R) : R \in \mathcal{R}\}$

If $\gamma^k < \mu^k(1 + \varepsilon)$ then STOP, $(x^*, y^*, \lambda^*, w^*)$ is an ε-solution of Problem

Else, divide R_k into $R_{k_1} = \{P_{k1}\}$ and $R_{k_2} = \{P_{k2}\}$ by the problem bisection.

For each R_{k_i} $(i = 1, 2)$, if the corresponding relaxed Problem (33) is feasible, then

(i) Compute an optimal solution $(x^{ki}, y^{ki}, \lambda^{ki}, w^{ki})$, and the optimal value $\mu^{ki} = \Psi(x^{ki}, y^{ki}, \lambda^{ki}, w^{ki})$, Update $\tilde{\gamma}^k = \min\{\tilde{\gamma}^k,\ F_\tau(x^{ki}, y^{ki}, \lambda^{ki})\}\quad (UB_f)$.

(ii) If $(x^{ki}, y^{ki}, \lambda^{ki}, w^{ki})$ is feasible to Problem (22), then update the Score $\gamma^k = \min\{\gamma^k,\ f(x^{ki}, y^{ki})\}$, and update the current best feasible solution $(x^*, y^*, \lambda^*, w^*) = (x^{ki}, y^{ki}, \lambda^{ki}, w^{ki})$ such that $\gamma^k = f(x^*, y^*)$.

(iii) If condition of restarting DCA (18) is satisfied then
Apply DCA to DC program (35) from the starting point $(x^{ki}, y^{ki}, \lambda^{ki})$ to obtain the solution $(x_\tau^{ki}, y_\tau^{ki}, \lambda_\tau^{ki})$.

b) Update $\tilde{\gamma}^k = \min\{\tilde{\gamma}^k,\ F_\tau(x_\tau^{ki}, y_\tau^{ki}, \lambda_\tau^{ki})\}\quad (UB_f)$.

c) If $(x_\tau^{ki}, y_\tau^{ki}, \lambda_\tau^{ki}, w_\tau^{ki})$ is feasible to Problem (22), then update the Score $\gamma^k = \min\{\gamma^k,\ f(x_\tau^{ki}, y_\tau^{ki})\}$, and update the current best feasible solution $(x^*, y^*, \lambda^*, w^*) = (x_\tau^{ki}, y_\tau^{ki}, \lambda_\tau^{ki}, w_\tau^{ki})$ such that $\gamma^k = f(x^*, y^*)$.

Set $\mathcal{R} \leftarrow \mathcal{R} \setminus \{R_i\ :\ \mu(R_i) > \gamma^k\}$ and go to iteration $k+1$.

According to well known results in [9,16,18,20,21] about Branch-and-Bound schemes using normal rectangular subdivisions, the convergence of our BB and our combined DCA-BB (GOA) can be stated as follows

Proposition 12 *(i) If the algorithm GOA terminates at some iterations K then* (x^K, y^K) *is a global solution of (1).*

(ii) If the Algorithm is infinite then it generates a bounded sequence $\{(x^k, y^k)\}$ *every cluster point of which is a global solution of (1) and* $\gamma^k \searrow \alpha$, $\mu^k \nearrow \alpha$.

Remark 13 In the case f is a convex function, algorithm **GOA** terminates after finitely many iterations, in virtue of Remark 11.

Remark 14 (Practical choice of the parameter penalty $\tau > \tau_1$ and its use in our combined DCA and BB).

In general it is difficult to compute any upper bound of τ_1 in Problem (10). In practice we take τ sufficiently large in order for Problem (10) to be equivalent to Problem (36). To check equivalence of these problems, we use the exact penalty results [13,14]: $\alpha(\tau) \leq \alpha$ for every $\tau \geq 0$ and if an optimal solution to Problem (10) with a given $\overline{\overline{\tau}} \geq 0$ is feasible to Problem P(36) then it is also an optimal solution of the latter one and $\overline{\overline{\tau}} \geq \tau_1$.

5 Illustrative examples and computational experiments

We have implemented our algorithm in the case where f is a nonconvex quadratic function for which the DC decomposition (28) will be used for the DCA, say

$$f(x, y) := \frac{1}{2}(x^T, y^T)H\begin{pmatrix} x \\ y \end{pmatrix} + (c^1)^T x + (c^2)^T y,$$

with H being $n_1 \times n_2$ symmetric matrix, $c^1 \in \mathbb{R}^{n_1}$, $c^2 \in \mathbb{R}^{n_2}$ and

$$Z := \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \geq 0, y \geq 0,\ Ax + By + d \leq 0\}, \tag{37}$$

with A and B being matrices of dimensions $(m_1 \times n_1)$, $(m_1 \times n_2)$ respectively and $d \in \mathbb{R}^{m_1}$. The example below illustrates the algorithms when

$$m_1 = 2, \quad p = 4, \quad n_1 = 2, \quad n_2 = 2.$$

$$H = \begin{pmatrix} -3.8 & 4.4 & 1.2 & -2.2 \\ 4.4 & -2.2 & 0.6 & 1.8 \\ 1.2 & 0.6 & 0.0 & 0.4 \\ -2.2 & 1.8 & 0.4 & 0.0 \end{pmatrix}, \quad c^1 = \begin{pmatrix} 935.74474 \\ 87.53654 \end{pmatrix}, \quad c^2 = \begin{pmatrix} 121.96196 \\ 299.24825 \end{pmatrix},$$

$$A = \begin{pmatrix} 0.00000 & 3.88889 \\ -2.00000 & 8.77778 \end{pmatrix}, \quad B = \begin{pmatrix} 4.88889 & 7.44444 \\ -5.11111 & 0.88889 \end{pmatrix}, \quad d = \begin{pmatrix} -61.57778 \\ -0.80000 \end{pmatrix},$$

$$P = \begin{pmatrix} -17.85000 & 6.57500 \\ 30.32500 & 30.32500 \end{pmatrix}, \quad Q = \begin{pmatrix} 21.10204 & 11.81633 \\ 11.81633 & -14.44898 \end{pmatrix}, \quad q = \begin{pmatrix} -18.21053 \\ 13.05263 \end{pmatrix},$$

$$D = \begin{pmatrix} 5.00000 & 7.44444 \\ -8.33333 & 3.00000 \\ -8.66667 & -8.55556 \\ 6.44444 & -5.11111 \end{pmatrix}, \quad E = \begin{pmatrix} 3.88889 & 1.77778 \\ 6.88889 & 6.11111 \\ -5.33333 & -7.00000 \\ 1.44444 & 4.44444 \end{pmatrix}, \quad b = \begin{pmatrix} -39.62222 \\ -60.00000 \\ 72.37778 \\ -17.28889 \end{pmatrix}.$$

The smallest eigenvalue of H is -8.4793, so we choose $\rho = 8.48$ in (28).

Algorithm GOA

Initialization

- The first lower bound is $\mu^1 = 2251.552997$ with

$$(x(I^1J^1), y(I^1J^1))^T = (0.200001, 1.999998, 3.999999, 4.600003),$$

$$w(I^1J^1)^T = (0, 0, 0, 0.000011), \lambda(I^1J^1)^T = (9.129313, 10.617138, 63.017967, 67.375372).$$

Applying DCA to (17) (to find an initial point of DCA applied to DC program (35)) we obtain

$$(\bar{x}^0, \bar{y}^0)^T = (0.200002, 1.999997, 4.000002, 4.600002),$$

$$(\bar{w}^0)^T = (0, 0, 0, 0), : (\bar{\lambda}^0)^T = (0, 0, 36.113880, 43.251700)$$

which is a feasible point to (22).

- Apply DCA to DC program (35) from this point we get

$$(\bar{x}^1, \bar{y}^1)^T = (0.200002, 1.999997, 4.000002, 4.600002)$$

$$(\bar{w}^1)^T = (0, 0, 0, 0), : (\bar{\lambda}^1)^T = (22.764326, 22.841008, 97.375901, 99.227316)$$

and $\gamma^1 = f(\bar{x}^1, \bar{y}^1) = 2251.553704$.
Since $\gamma^1 - \mu^1 = 0.000707$, the algorithm is terminated with an ε-optimal solution $(\bar{x}^1, \bar{y}^1)$.

When applying the Branch and Bound algorithm without DCA we obtain an ε-optimal solution

$$(\tilde{x}, \tilde{y})^T = (0.200001, 1.999997, 3.999998, 4.600005),$$

$$\tilde{w}^T = (0.000007, 0, 0, 0), \ \tilde{\lambda}^T = (0, 46.900128, 111.298969, 97.180623),$$

with $\gamma^k = 2251.553551$ and $\mu^k = 2251.553416$ after 5 iterations.

We have tested the two variants of our algorithms: the combined DCA - Branch and Bound algorithm **GOA** and the Branch and Bound scheme in **GOA** without DCA denoted **BB** (i.e. the step (iii) in **GOA** is removed) on a collection of 15 problems randomly generated. The elements of matrices and vectors in these problems are random numbers in the interval $(-20, 20)$. The constraints at level 1 are generated such that the feasible set Z defined by (37) of Problem (4) is nonempty and bounded. For the boundedness of Z we take the first constraint as (A_i and B_i denote the i^{th} row of the matrices A and B respectively)

$$A_1x + B_1y + d_1 \leq 0,$$

with $A_1 > 0$, $B_1 > 0$ and $d_1 < 0$. The other constraints in level 1 are randomly generated. We check nonemptiness of generated polyhedral convex sets Z by using the CPLEX 7.5. Let $(x^0, y^0) \in Z$. The constraints at level 2 are randomly generated such that the feasible set of Problem (4) in level 2 is nonempty. For this, we randomly generate the matrices D, E and set $b = -Dx^0 - Ey^0$. Finally we use only the random data which provide nonempty bounded polyhedral convex set Ω.

The algorithm is written in Visual C++ and the code is run on a Intel Centrino 2 GHz with 1 Go of RAM. The software CPLEX 7.5 has been used for solving convex quadratic programs. We take $\epsilon_1 = \epsilon_2 = 10^{-5}$ for DCA. The stopping criterion of **GOA** is $\gamma_*^k - \mu^k \leq 0.05\mu^k$ (we replace the condition $\gamma_*^k = \mu^k$ in step i) of iteration k by this inequality). In step v) we restart DCA when $F_\tau((x(I_i^k J_i^k), y(I_i^k J_i^k), \lambda(I_i^k J_i^k)) < (1 + 0.001)\gamma^k$.

The performance of **BB** is presented in Table 1. In Table 2 we report the results of **GOA**: updating the upper bound deals only with feasible solutions of Problem (22).

We use the following notations:

- N°: Problem's number;
- UB: the upper bound (the optimal value given by the algorithm);
- LB: the lower bound.
- #Iter: number of iterations of the algorithm;
- #DCA: number of restarting DCA;

Table 1 Computational results of **BB**

No.	Dim				B&B				
	n_1	n_2	m_1	p	#Iter	UB	LB	$t(s)$	
1	50	40	40	30	505	462644.5247	448955.1444	242.20	
2	55	45	50	40	10000	–	–	–	**F**
3	60	40	45	30	2492	−28265449.1373	−28272922.7411	2182.47	
4	60	45	50	40	1503	−611438.5458	−642009.5275	1238.36	
5	65	45	45	45	2562	2046166.8028	2019211.5721	3196.06	
6	70	50	40	40	2515	−1089775.0821	−1111945.6653	2904.92	
7	75	55	60	70	10000	–	–	–	**F**
8	80	55	70	50	10000	–	–	–	**F**
9	85	65	50	70	1802	601955.0399	585978.1117	3872.19	
10	90	50	35	40	4266	−38156237.9444	−38181803.5565	6379.34	
11	100	60	50	60	955	−689439.8551	−723899.9328	2126.77	
12	100	70	60	60	10000	–	–	–	**F**
13	110	75	50	60	2859	−1756770.8377	−1816249.7601	6980.55	
14	115	80	50	60	10000	–	–	–	**F**
15	120	100	80	80	10000	–	–	–	**F**

F : Failed to optimize—cannot find a global solution after 10000 iterations

Table 2 Computational results of **GOA**

No.	Dim				GOA				
	n_1	n_2	m_1	p	#Iter	UB	LB	#DCA	$t(s)$
1	50	40	40	30	12	459822.8581	439066.0691	4	20.36
2	55	45	40	40	37	−1041541.4271	−1066550.3169	2	41.64
3	60	40	45	30	2	−27552404.9379	−28923435.2544	1	20.22
4	60	45	50	40	11	−632930.3141	−656901.7298	2	30.44
5	65	45	45	45	63	2029743.5968	1962483.0682	96	284.52
6	70	50	40	40	31	−1100516.7247	−1155224.0662	2	71.05
7	75	55	60	70	273	824830.4224	786643.1045	2	556.36
8	80	55	70	50	9276	−574659.5627	−603391.6106	36	16162.67
9	85	65	50	70	141	596288.0777	568695.1210	18	451.27
10	90	50	35	40	26	−37872305.7633	−39742103.3698	16	108.24
11	100	60	50	60	251	−696833.4655	−731633.8059	2	495.47
12	100	70	60	60	207	−1738223.7518	−1815521.9879	2	527.58
13	110	75	50	60	27	−1772929.7039	−1840109.5365	15	212.91
14	115	80	50	60	118	1534483.3384	1500965.6830	2	488.56
15	120	100	80	80	407	2230626.8007	2121577.4023	13	2371.28

We have chosen the penalty parameter $\tau = 10^5$ for these test problems. The optimal solution of the convex quadratic program Problem will be used as initial point for DCA in our combined algorithm GOA. All CPU are computed in seconds.

Note that #Iter is limited to 10000. Beyond this threshold we consider that the algorithm fails: neither feasible solution is found nor the quality of obtained solution is less than 5%.

Comments: from the numerical results on these test problems we observe that

(i) The combined DCA-Branch and Bound algorithms **GOA** always provided ϵ-optimal (feasible) solutions to quadratic bilevel programs. While the Branch and Bound algorithm cannot provide ϵ-optimal (feasible) solutions in many cases

(ii) Not surprisingly, **GOA** is much better than **BB**. The superiority of **GOA** with respect to **BB** increases with the dimension. The efficiency of **GOA** is clearly due to DCA.

(iii) DCA finds rapidly a feasible solution to (4), by the way it improves considerably the best current feasible solution during the restarting process and therefore accelerates the convergence of **GOA**.

(iv) An interesting issue is how to restart DCA. Our strategy seems to be quite natural and efficient because the number of restarting DCA is relatively small, except for some problems finding feasible solutions (of the original problem) requires a more important number of restarting DCA.

6 Conclusion

We have proposed a combination of DCA (efficient local algorithm capable of handling large scale DC programs) and an adapted Branch-and-Bound scheme (**BB**) to globally solving nonlinear bilevel problems in which the leader's problem is a linearly constrained DC program while the follower's problem concerns the set of all Karush-Kuhn-Tucker points of quadratic programming problems. The problem is beforehand reformulated as a DC program with the help of exact penalty techniques in DC programming and suitable DC decompositions are investigated in order to get appropriate Branch-and-Bound methods. The very original

DCA's feasibility (i.e. the sequence of points generated by DCA, applied to the penalized equivalent DC program, is contained in the feasible set of the original problem) is pointed out as well as the strategy of restarting DCA with better feasible solutions given by **BB.** Numerical simulations show the effectiveness and efficiency of DCA and its combination GOA with **BB** in order to reach global solutions of the problems under consideration. Our future work is dealing with solution of the complete nonlinear bilevel obtained from (1) by replacing the KKT—point set $K(x)$ of (2) by its optimal solution set.

References

1. Bard, J.F.: An algorithm for solving the general bilevel programming problems. Math. Oper. Res. **8**, 260–272 (1983)
2. Bard, J.F.: Convex two-level optimization. Math. Program. **40**, 15–27 (1988)
3. Bard, J.F., Falk, J.E.: An explicit solution to the multi-level programming problem. Comput. Oper. Res. **9**, 77–100 (1982)
4. Falk, J.E., Liu, J.: On bilevel programming, Part I: general nonlinear cases. Math. Program. **70**, 47–72 (1995)
5. Gumus, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. J. Glob. Optim. **20**, 1–31 (2001)
6. Hermanns, P., Van Thoai, N.: Method for global optimization of a class of nonlinear bilevel programming problems. Research Report No. 02-14. Department of Mathematics, University of Trier (2002)
7. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms, Parts I&II. Springer Verlag (1991)
8. Hoai An, L.T.: Contribution à l'Optimisation Non Convexe et l'Optimisation Globale: Théorie, Algorithmes et Applications. Habilitation, National Institute for Applied Sciences, Rouen (1997)
9. Hoai An, L.T., Tao, P.D.: Solving a class of linearly constrained indefinite quadratic problems by DC algorithms. J. Glob. Optim. **11**, 253–285 (1997)
10. Hoai An, L.T., Tao, P.D.: Large scale global molecular optimization from exact distance matrices by a DC optimization approach. SIAM J. Optim. **14**(1), 77–116 (2003)
11. Hoai An, L.T., Tao, P.D.: DC programming approaches and DCA for globally solving linear complementarity problems. Research Report, National Institute for Applied Sciences, Rouen, Submitted (2004)
12. Hoai An, L.T., Tao, P.D.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Ann. Oper. Res. **133**, 23–48 (2005)
13. Hoai An, L.T., Tao, P.D., Muu, L.D.: Exact penalty in DC programming. Vietnam J. Math. **27**(2), 169–179 (1999)
14. Hoai An, L.T., Tao, P.D., Van Ngai., H.: Exact penalty techniques in DC programming. Research Report, National Institute for Applied Sciences, Rouen, Submitted (2004)
15. Horst, R., Pardalos, P.M., Van Thoai, N.: Introduction to Global Optimization, 2nd edn. Kluwer Academic Publishers (2000)
16. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. 3rd edn. Springer, Berlin (1996)
17. Luo, Z.-Q., Pang, J.-S., Ralph, D., Wu, S.-Q.: Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints. Math. Program. **75**, 19–76 (1996)
18. Muu, L.D., Phong, T.Q., Tao, P.D.: Decomposition methods for solving a class of nonconvex programming problems dealing with bilinear and quadratic functions. Comput. Optim. Appl. **4**, 203–219 (1995)
19. Muu, L.D., Van Qui, N.: A global optimization method for solving convex quadratic bilevel programming problems. J. Glob. Optim. **26**, 199–219 (2003)
20. Phong, T.Q., Hoai An, L.T., Tao, P.D.: Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems. Oper. Res. Lett. **17**, 215–222 (1996)
21. Phong, T.Q., Hoai An, L.T., Tao, P.D.: On globally solving linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method. RAIRO Recherche Opérationnelle **30**(1), 31–49 (1996)
22. Rockafellar, R.T.: Convex Analysis. Princeton University Press, NJ (1970)
23. Shimizu, K., Ishizuka, Y., Bard, J.F.: Nondifferentiable and Two-Level Mathematical Programming. Kluwer Academic Publishers (1997)
24. Tanino, T., Ogawa, T.: An algorithm for solving two-level convex optimization problems. Int. J. Syst. Sci. **15**, 163–174 (1984)

25. Tao, P.D., Hoai An, L.T.: Convex analysis approaches to DC programming: theory, algorithms and applications. Acta Math. Vietnam. **22**(1), 287–367 (1997)
26. Tao, P.D., Hoai An, L.T.: DC optimization algorithms for solving the trust region subproblem. SIAM J. Optim. **8**, 476–507 (1998)
27. Van Thoai, N., Yamamoto, Y., Yoshise, A.: Global optimization method for solving mathematical programs with linear complementarity constraints. J. Optim. Theory Appl. (Forthcoming) (2002)
28. Vicente, L.N., Calamai, P.H.: Bilevel and multilevel programming: a bibliography review. J. Glob. Optim. **5**, 291–306 (1994)