J Glob Optim (2010) 47:615–637
DOI 10.1007/s10898-009-9482-3

Solving convex quadratic bilevel programming problems using an enumeration sequential quadratic programming algorithm

Jean Bosco Etoa Etoa

Received: 22 December 2006 / Accepted: 22 October 2009 / Published online: 4 November 2009

Abstract In this paper, we present an original method to solve convex bilevel programming problems in an optimistic approach. Both upper and lower level objective functions are convex and the feasible region is a polyhedron. The enumeration sequential linear programming algorithm uses primal and dual monotonicity properties of the primal and dual lower level objective functions and constraints within an enumeration frame work. New optimality conditions are given, expressed in terms of tightness of the constraints of lower level problem. These optimality conditions are used at each step of our algorithm to compute an improving rational solution within some indexes of lower level primal-dual variables and monotonicity networks as well. Some preliminary computational results are reported.

Keywords Convex quadratic bilevel program · Enumeration sequential quadratic programming algorithm · Primal-dual monotonicity · Monotonicity network

1 Introduction

The bilevel programming problems (BLP) are hierarchical optimization problems in the sense that their constraints are defined in part by a second parametric optimization problem. We can also define BLP as a sequence of two optimization problems in which the feasible region of the upper level problem is determined implicitly by the solution set of the lower level

---

The algorithm presented in this paper was conceived during a PhD research program of the author at MAGI, École Polytechnique de Montréal, C. P. 6079, succ. Centre-ville Montréal (Québec), H3C 3A7 Canada [21].

J. B. E. Etoa (✉)
Department of Economic and Management Sciences, University of Yaounde II, BP 15, Soa, Yaounde, Cameroon
e-mail: jbetoa_etoa@hotmail.com; jbetoa3101@rogers.com

Present Address:
J. B. E. Etoa
Cameroon High Commission, 170 Clemow Avenue, Ottawa, ON K1S 2B4, Canada

problem. Early formulations of BLP can be found in [30,37]. Candler and Norton [13,14] were the first to use the term 'bilevel' or 'multilevel', to describe a development policy problem. Comprehensive overview of the historical development of BLP can be found in [18,21,41]. The properties as optimality conditions of BLP have been studied in many papers as in [4,5,9,11,20]. Jeroslow [28], Ben-Ayed and Blair [7] showed that the linear BLP is NP-hard. Hansen et al. [27] strengthened this result; they proved that BLP is strongly NP-hard. Marcotte and Savard [34] present the relationship between two specific classes of bilevel programs and well-known combinatorial problems. Applications have been a stimulating factor for the development of bilevel programming and some related algorithms. The number of papers presenting applications is constantly growing. Interesting applications include the investigation of network of oligopolies [1], as well as the determination of optimal prices, as road tolls or prices for electricity [10]. Related is the determination of optimal tax credits for biofuel production [7]. An overview on applications of bilevel programming can be found in [43].

The formulation of BLP considered in this paper follows: QCBL

$$\begin{aligned} &\min_{x,y} F(x,y) = c_1^t x + d_1^t y + (x^t, y^t) N (x^t, y^t)^t \\ &\text{s.t.} \begin{cases} x \in P = \{x \geq 0 : A_1 x \leq b_1\}, \\ \min_y f(x,y) = c_2^t x + d_2^t y + (x^t, y^t) Q (x^t, y^t)^t \\ \text{s.t.} \begin{cases} A_2 x + B_2 y \leq b_2, \\ y \geq 0. \end{cases} \end{cases} \end{aligned} \tag{1.1}$$

where $N, Q \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ are symmetric positive semidefinite matrices, $c_1, c_2 \in \mathbb{R}^{n_1}, d_1, d_2 \in \mathbb{R}^{n_2}$, F and f are the objective functions of the upper level (leader) and the lower level (follower), respectively, $A_1 \in \mathbb{R}^{m_1 \times n_1}$, $A_2 \in \mathbb{R}^{m \times n_1}$, $B_2 \in \mathbb{R}^{m \times n_2}$ and the rank of B_2 is equal to m, $b_1 \in \mathbb{R}^{m_1}, b_2 \in \mathbb{R}^m$; $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$ are the decision variables under the control of the upper level and the lower level, respectively. The polyhedron $\Omega = \{(x,y) \in P \times \mathbb{R}_+^{n_2} : A_2 x + B_2 x \leq b_2\}$ defined by both upper and lower levels is called feasible region of the problem (1.1). Given a value x of the upper level variable, $M(x)$ is the set of optimal solutions of the lower level problem define by the following program

$$\begin{aligned} &\min_y f(x,y) = c_2^t x + d_2^t y + (x^t, y^t) Q (x^t, y^t)^t \\ &\text{s.t.} \begin{cases} A_2 x + B_2 y \leq b_2, \\ y \geq 0. \end{cases} \end{aligned} \tag{1.2}$$

An element of the inducible region defined by $IR = \{(x,y) \in \Omega : y \in M(x)\}$ is called a rational solution. As most of the contributions to bilevel programming, we consider the BLP problem (1.1) in an optimistic approach i.e., for a given value of the upper level variable x, the solution of the lower level problem $y \in M(x)$ is computed as the best one according to the goal of the upper level problem. Based on the same fact, several algorithms have been proposed that compute an optimal solution using enumeration scheme (see [15]). One of a solution approach to BLP is to transform the original two level problems into a single level problem by replacing the lower level problem with its Karush–Kuhn–Tucker (KKT) optimality conditions; based on this transformation, other methods containing branch-and-bound algorithms were developed (see [3,22,25,27,39]). Local optimization procedures were also developed (see [29,38,42]).

Using monotonicity properties, the *HJS* algorithm for linear BLP (see [27]) exploits necessary conditions for some subsets of the lower level problem to contain at least one constraint that is active. That algorithm also investigates branching rules, based on logical relation expressing the conditions of the tightness of constraints which have been detected. In this paper, we propose an enumeration sequential quadratic programming algorithm to solve the problem (1.1) where F and f are convex functions. We introduce from monotonicity principles an original concept of monotonicity networks. A monotonicity path is used to express the tightness of some subsets of the primal and dual lower level constraints, depending on the current rational solution. At each step of our algorithm, monotonicity paths are used to compute an improving rational solution within the tightness of primal-dual lower level constraints. Given a feasible or rational solution in a branch and bound framework, our method performs a sequential investigation on some indexes of lower level primal-dual variables in order to compute some improved rational solutions.

The organization of the rest of the paper is as follows. The next section is devoted to the conceptual framework of our algorithm. Here, we introduce monotonicity networks from monotonicity principles to express the tightness of the lower level problem. Our algorithm is stated in Sect. 3 and illustrated by an example in Sect. 4. Finally, computational experiences are reported in Sect. 6; on one of the problem solved, we illustrate how, the solution computed by the global optimization method introduced by Muu and Quy [35] or by the genetic algorithm of Wang et al. [44] is not feasible, contrary to the solution computed by our algorithm.

2 Preliminaries

In problem (1.1), let

$$Q = \begin{bmatrix} Q_x & Q_{xy} \\ Q_{xy}^t & Q_y \end{bmatrix}$$

where $Q_x \in \mathbb{R}^{n_1 \times n_1}$, $Q_y \in \mathbb{R}^{n_2 \times n_2}$, $Q_{xy} \in \mathbb{R}^{n_1 \times n_2}$.

Then, the lower level objective function is transformed into

$$f(x, y) = c_2^t x + x^t Q_x x + (d_2 + 2Q_{xy}x)^t y + y^t Q_y y.$$

Next, as discussed in [3,39] we assume that the set of feasible solution Ω is nonempty and bounded.

The KKT formulation of problem (1.1) follows: QCBL_{KKT}

$$\begin{array}{l} \min_{x,y,u,v,w} F(x, y) = c_1^t x + d_1^t y + (x^t, y^t) N (x^t, y^t)^t \\ \text{s.t.} \begin{cases} x \in P, \\ A_2 x + B_2 y + w = b_2, \\ 2Q_{xy}x + 2Q_y y + B_2^t u - v = -d_2, \\ u^t w = v^t y = 0, \\ x, y, u, v, w \geq 0, \end{cases} \end{array} \tag{2.1}$$

where $u \in \mathbb{R}^m$ is the dual lower level variable, or the vector of KKT multipliers of the lower level problem, $w \in \mathbb{R}_+^m$ and $v \in \mathbb{R}_+^{n_2}$ are, respectively, the primal and dual slack variables. Nonconvexity arises first in the complementarity conditions $u^t w = v^t y = 0$. These conditions are usually not difficult to handle as they can be taken implicity into account in the separation framework of branch and bound algorithm. The *ESQP* algorithm uses the KKT formulation of problem (1.1) within the monotonicity analysis; the main idea of the algorithm consists in performing a sequential investigation on some subsets of indexes of complementarity variables in order to compute an improving rational solution of problem (1.1). We define the support of a vector as follows:

Definition 2.1 Let $S = \{1, 2, \ldots, n_2\}$ and $R = \{1, 2, \ldots, m\}$; the **support** of a vector $y \in \mathbb{R}_+^{n_2}$ is defined as $\mathrm{spp}(y) = \{i \in S : y_i = 0\}$; all the components of y with indexes in $\mathrm{spp}(y)$ are equal to zero, whereas for $u \in \mathbb{R}_+^m$, $\mathrm{spp}(u) = \{j \in R : u_j = 0\}$.

A rational point $(x^*, y^*) \in \Omega$ is an optimal solution of the problem (1.1) if and only if there exist $u^* \in \mathbb{R}_+^m$, $v^* \in \mathbb{R}_+^{n_2}$, $w^* \in \mathbb{R}_+^m$ such that $(x^*, y^*, u^*, v^*, w^*)$ is an optimal solution of the problem (2.1) (see ref. [4]). Let $\mathfrak{F}^{KKT}$ be the set of feasible solutions of problem (2.1); if $(x^*, y^*, u^*, v^*, w^*) \in \mathfrak{F}^{KKT}$, then from the complementarity constraints in (2.1), there exist partitions of indexes (I_*, I_*^C) in S and (J_*, J_*^C) in R such that $I_* \subseteq \mathrm{spp}(y^*)$, $I_*^C \subseteq \mathrm{spp}(v^*)$ with $I_*^C = S \backslash I_*$, and $J_* \subseteq \mathrm{spp}(u^*)$, $J_*^C \subseteq \mathrm{spp}(w^*)$ with $J_*^C = R \backslash J_*$. For a given step k of the *ESQP* algorithm, let $r^k = (y^k, u^k, v^k, w^k)$; to build subsets of indexes included in $\mathrm{spp}(\bar{y})$ and $\mathrm{spp}(\bar{u})$ corresponding to an improving prima-dual rational solution $\bar{z} = (\bar{x}, \bar{y}, \bar{u}, \bar{v}, \bar{w}) \in \mathfrak{F}^{KKT}$, sequentially we solve a family of relaxed quadratic convex problems formulated below: QCBR_T

$$\min_{x,y,u,v,w} F(x, y)$$
$$\text{s.t.} \begin{cases} x \in P, \\ A_2 x + B_2 y + w = b_2, \\ 2Q_{xy} x + 2Q_y y + B_2^t u - v = -d_2, \\ r_t = 0 \text{ for } t \in T, \\ x \geq 0, r = (y, u, v, w) \geq 0, \end{cases} \tag{2.2}$$

where $T \subseteq N(r^k)$. For $r = (y, u, v, w)$, $N(r)$ represents a subset of indexes of r whose properties are described in the next subsection.

Let $z^{k_0+1} = (x^{k_0+1}, y^{k_0+1}, u^{k_0+1}, v^{k_0+1}, w^{k_0+1})$ be a feasible solution of some relaxed quadratic program QCBR_T where $T \subseteq N(r^{k_0})$ at a step k of the *ESQP* algorithm such that $k_0 \geq k$, and let $z^* = (x^*, y^*, u^*, v^*, w^*) \in \mathfrak{F}^{KKT}$ be the current feasible solution of the problem (2.1). All the components of a primal-dual lower level variables in z^{k_0} with indexes in T are constrained to be equal to 0, leading then to z^{k_0+1}. From a feasible solution z^{k_0} of some program QCBR_T,$T \subseteq N(r^{k_0-1})$, and from the family of subsets of indexes $T \subseteq N(r^{k_0})$, the *ESQP* algorithm constructs sequentially a partition of indexes (I, I^C) in S and (J, J^C) in R such that $I \subseteq \mathrm{spp}(y^{k_0+1})$ $(\Rightarrow I^C \subseteq \mathrm{spp}(v^{k_0+1}))$, $J \subseteq \mathrm{spp}(u^{k_0+1})$ $(\Rightarrow J^C \subseteq \mathrm{spp}(w^{k_0+1}))$ with $F(x^{k_0+1}, y^{k_0+1}) \leq valF$ within an enumerative scheme, where $valF$ represents the current value of the upper level objective function. We now introduce the definitions of improving and improvable rational solutions related to the *ESQP* algorithm.

Definition 2.2 Let $(x^*, y^*) \in IR$ be the current rational solution of problem (1.1) and let $z^{k_0+1} = (x^{k_0+1}, y^{k_0+1}, u^{k_0+1}, v^{k_0+1}, w^{k_0+1})$ be a computed feasible solution of some relaxed quadratic program QCBR_T where $T \subseteq N(r^{k_0})$ at a given step k of the *ESQP* algorithm such that $k_0 \geq k$. If $z^{k_0+1} \in \mathfrak{F}^{KKT}$, then the solution $(x^{k_0+1}, y^{k_0+1}) \in IR$ is called **improving rational solution** for the *ESQP* algorithm whereas $(x^*, y^*) \in IR$ is an **improvable rational solution**.

Remark 2.1 The relaxed programs solved within the branch and bound algorithm in [3] are almost identical to the family of programs QCBR_T, where $|T| \geq 1$.

Let $z^* = (x^*, y^*, u^*, v^*, w^*)$ be the current solution of the problem (2.1), and let's consider the subsets of indexes $I \subset S$ and $J \subset R$. If $J^C \subseteq \mathrm{spp}(w^*)$, then the primal lower level constraints with indexes $i \in J^C$ become equalities. Similarly, if $I^C \subseteq \mathrm{spp}(v^*)$, then all of the constraints of the dual lower level problem with indexes $j \in I^C$ become equalities. A primal lower level variable y_j or dual lower level variable u_i can also be equal to zero at optimality. We find such variables as well as the family of indexes of primal-dual lower level variables $N(r^*)$ by using the monotonicity principles. These principles were introduced first by Wilde [45], while Hansen et al. [26] generalized them. The *HJS* algorithm [27] takes advantage of these principles within a branch and bound framework; it exploits the condition of tightness of the lower level constraints. But, the enumerative scheme used by the *ESQP* algorithm may not investigate all the combinations of partitions of indexes (I_*, I_*^C) and (J_*, J_*^C) defined previously in order to compute a global optimal solution of the BLP problem (1.1). Therefore, in this paper, the convergence of the *ESQP* algorithm to the global optimal solution will not be possible to investigate.

Notation In this the paper, a monotonicity property will be denoted by M-property.

2.1 Necessary optimality conditions: M-principles

From the lower level objective function and the dual of the lower level program, we state M-principles considered as the key of our algorithm. The dual lower level problem DPS(x) has the following formulation: DPS(x)

$$\begin{aligned} &\max_{\mathrm{u,y}} Df(u, y) = c_2^t x + x^t Q_x x + (A_2 x - b_2)^t u - y^t Q_y y \\ &\text{s.t.} \begin{cases} -2Q_{xy} x - 2Q_y y - B_2^t u \leq d_2, \\ y, u \geq 0. \end{cases} \end{aligned} \tag{2.3}$$

The *ESQP* algorithm uses the KKT formulation of problem (1.1), as well as the primal-dual M-principles involving the primal and dual lower level problem, contrary to the M-analysis introduced in [27] which deals only with the primal lower level problem. The M-analysis is usually applied to the lower level primal variables, whereas some components of the dual lower level variables may also be equal to zero at optimality. Therefore, the M-analysis should be applied to the dual variable u. This approach can contribute to determine analytically variables in the complementarity constraints which may be equal to zero at optimality, contrary to the treatment that Bard and Moore [3] applied to these variables in their algorithm.

Let's assume that $\bar{i}$ represents the index of the ith component of the slack variable w, whereas $\bar{j}$ represents the index of the jth component of the slack variable v. Let (B_{ij}) be the coefficients of matrix B_2. For each lower level variable y_j, $j \in S$, let us define the following sets:

$$\begin{aligned} I_{j^+} &= \left\{\bar{i} : i \in \{1, 2, \ldots, m\}, B_{ij} > 0 \text{ and } \nabla_{y_j} f(x, y) < 0, (x, y) \in \Omega\right\}, \\ I_{j^-} &= \left\{\bar{i} : i \in \{1, 2, \ldots, m\}, B_{ij} < 0 \text{ and } \nabla_{y_j} f(x, y) > 0, (x, y) \in \Omega\right\}, \\ I_{j^\pm} &= \left\{\bar{i} : i \in \{1, 2, \ldots, m\}, B_{ij} \neq 0 \text{ and } \bar{i} \notin I_{j^+} \cup I_{j^-}\right\}. \end{aligned} \tag{2.4}$$

The primal M-principles stated in the theorem below gives an indication: (i) on the components of the primal lower level variable which may be equal to zero at optimality, or (ii) on the tightness of the primal lower level constraints at optimality.

Theorem 2.1 (Primal M-principles). *Let $(\bar{x}, \bar{y})$ be a rational solution of the problem (1.1). Let's consider the lower level primal constraints; then, for all $j \in S$, (a) there exists at least one lower level active constraint with index i such that $\bar{i} \in I_{j^+} \cup I_{j^\pm}$; moreover, (b) there exists at least one lower level active constraint with index i such that $\bar{i} \in I_{j^-}$.*

Proof Assume by contradiction that for some rational solution $(\bar{x}, \bar{y})$, there exists some $j \in S$ such that $\nabla_{y_j} f(\bar{x}, \bar{y}) < 0$ and (a) does not hold. Let $\varepsilon \in \mathbb{R}$ and consider the vector $y = \bar{y} + \varepsilon e_j$ where e_j is the unit vector with the jth component equal to 1. Since f is continuous, there exists $\delta > 0$ such that $\nabla_{y_j} f(\bar{x}, y) < 0$ for all $\varepsilon < \delta$. Moreover, as $B_2 y \leq b_2 - A_2\bar{x}$, there exists δ'such that y remains feasible for all $\varepsilon < \delta'$ in all constraints with index i such that $\bar{i} \in I_{j+} \cup I_{j\pm}$ (since (a) does not hold, i.e., constraints with index i such that $\bar{i} \in I_{j+} \cup I_{j\pm}$ are satisfied as strict inequalities). It is also easy to check that $(\bar{x}, y)$ satisfies the constraints with $\bar{i} \in I_{j-}$, as the corresponding function is not increasing with respect to y_j.

Consider now a vector y with ε satisfying $0 < \varepsilon < \min\{\delta, \delta'\}$. It leads to a feasible solution with a value lower than $f(\bar{x}, \bar{y})$, which is a contradiction.
A similar reasoning can be made for (b). □

Corollary 2.1 *Let's consider the lower level primal constraints. For any optimal solution* $(\bar{x}, \bar{y})$ *of problem* (1.1)*, the tightness of the constraints of the lower level problem is such that for all* $j \in S$*, there exists, at least one active constraint with index* i *such that* $\bar{i} \in I_{j+} \cup I_{j\pm}$*; moreover, there exists at least one active constraint with index* i *such that* $\bar{i} \in I_{j-}$*.*

Proof Since optimal solutions of problem (1.1) are rational, this follows immediately from Theorem 2.1.

Let's now consider the dual of the lower level program (2.3). For a given value $x \in P$, $\Omega^*(x)$ represents the dual feasible region of the lower level problem. For each dual lower level variable u_i, $i \in R$, one defines the following index sets:

$$\begin{aligned} J_{i+} &= \left\{\bar{j} : j \in S, B_{ij} < 0 \text{ and } \nabla_{u_i} Df(u, y) > 0, u \in \Omega^*(x)\right\}, \\ J_{i-} &= \left\{\bar{j} : j \in S, B_{ij} > 0 \text{ and } \nabla_{u_i} Df(u, y) < 0, u \in \Omega^*(x)\right\}, \\ J_{i\pm} &= \left\{\bar{j} : j \in S, B_{ij} \neq 0 \text{ and } \bar{j} \notin J_{i+} \cup J_{i-}\right\}. \end{aligned} \tag{2.5}$$

The following theorem (respectively corollary) is a statement of M-principles, applied to the dual of lower level program (2.3). The proof is the same as for Theorem 2.1 (respectively corollary 2.1).

Theorem 2.2 (Dual M-principles). *Let* $(\bar{x}, \bar{y})$ *be a rational solution of problem* (1.1) *and let* $\bar{u}$ *be an optimal solution of problem DPS(*$\bar{x}$*). Let's consider the lower level dual constraints; then, for all* $i \in R$*, there exists at least one active constraint with index* j *such that* $\bar{j} \in J_{i+} \cup J_{i\pm}$*. Moreover, there exists at least one active constraint with index* j *such that* $\bar{j} \in J_{i-}$*.*

Corollary 2.2 *For any optimal solution* $(\bar{x}, \bar{y})$ *of problem* (1.1) *and for any optimal solution of DPS(*$\bar{x}$*), the tightness of the constraints in problem DPS(*$\bar{x}$*) is such that for all* $i \in R$*, there exists at least one active constraint with index* $j \in J_{i+} \cup J_{i\pm}$*; moreover, there exists at least one active constraint with index* j *such that* $\bar{j} \in J_{i-}$*.*

Using only structural lower level and upper level primal and dual variables, the constraints in problem (2.1) can be written as:

$$\begin{bmatrix} A_2 & B_2 y & 0 \\ -2Q_{xy} & -2Q_y & -B_2^t \\ 0 & -I_{n_2} & 0 \\ 0 & 0 & -I_m \end{bmatrix} \begin{pmatrix} x \\ y \\ u \end{pmatrix} \leq \begin{bmatrix} b_2 \\ d_2 \\ 0 \end{bmatrix}. \tag{2.6}$$

From the formulations (2.1) and (2.6), we associate to each primal lower level constraint with index i, a boolean variable α_i such that, if $\alpha_i = 1$, this constraint becomes an equality ($i \in \text{spp}(w^*)$). To the primal lower level variable with index j, we associate a boolean variable α_{m+n_2+j} such that, if $\alpha_{m+n_2+j} = 1$, then $y_j^* = 0$ ($j \in \text{spp}(y^*)$). Assume that f is decreasing with respect to y_j and consider the set of indexes I_{j+} as in (2.4). According to Theorem 2.1, at least one of the boolean variable α_i with $\bar{i} \in I_{j+}$ is such that $\alpha_i = 1$. Then, the inequality $\sum_{I:B_{ij}>0} \alpha_i \geq 1$ holds. Now, assume that f is increasing with respect to y_j and consider the set of indexes I_{j-} as in (2.4). According to Theorem 2.1, at least one of the boolean variable α_i with $\bar{i} \in I_{j-}$ is such that $\alpha_i = 1$. As $-y_j \leq 0$ in (2.6), we may also have $\alpha_{m+n_2+j} = 1$. Then, the inequality $\sum_{I:B_{ij}<0} \alpha_i + \alpha_{m+n_2+j} \geq 1$ holds.

Note that Theorem 2.1 generalizes to bilevel programming, the first monotonicity condition in [27]. The following properties introduced by Hansen et al. [27], represent necessary optimality conditions, formulated as the tightness of the lower level constraints; it is another way to formulate Theorems 2.1 and 2.2.

Property 2.1 *For any rational solution*(x^*, y^*)*to problem* (1.1)*, the tightness of the primal lower level constraints is such that*:

$$\begin{aligned} &\sum_{i:B_{ij}>0} \alpha_i \geq 1 \text{ if } \nabla_{y_j} f(x^*, y^*) < 0, \quad (p1) \\ &\sum_{i:B_{ij}<0} \alpha_i + \alpha_{m+n_2+j} \geq 1 \text{ if } \nabla_{y_j} f(x^*, y^*) > 0, \quad (p2) \\ &\sum_{i:B_{ij}\neq 0} \alpha_i \geq 1 \text{ if } \nabla_{y_j} f(x^*, y^*) = 0, (p3) \end{aligned} \tag{2.7}$$

for $j = 1, \ldots, n_2$ *and* $i = 1, \ldots, m$.

As previous formulation in (2.6), we associate to each dual lower level constraint with index $m + j$, a boolean variable α_{m+j}. If $\alpha_{m+j} = 1$, then this constraint becomes an equality ($j \in \text{spp}(v^*)$). To the dual lower level variable with index i, we also associate a boolean variable α_{m+2n_2+j} such that, if $\alpha_{m+2n_2+i} = 1$, then $u_i^* = 0$ ($i \in \text{spp}(u^*)$).

Property 2.2 *For any rational solution* (x^*, y^*) *to problem* (1.1) *and for any optimal solution* u^* *of the dual problem* (2.3)*, the tightness of the constraints for problem* (2.3) *is such that:*

$$\begin{aligned} &\sum_{j:B_{ij}>0} \alpha_{m+j} \geq 1 \text{ if } \nabla_{u_i} Df(u,^* y^*) > 0, \quad (d1) \\ &\sum_{j:B_{ij}<0} \alpha_{m+j} + \alpha_{m+2n_2+i} \geq 1 \text{ if } \nabla_{u_i} Df(u,^* y^*) < 0, \quad (d2) \\ &\sum_{j:B_{ij}\neq 0} \alpha_i \geq 1 \text{ if } \nabla_{u_i} Df(u,^* y^*) = 0, \quad (d3) \end{aligned} \tag{2.8}$$

for $j = 1, \ldots, n_2$ *and* $i = 1, \ldots, m$.

Let $z^* = (x^*, y^*, u^*, v^*, w^*) \in \mathfrak{F}^{KKT}$ be the current feasible solution of the problem (2.1). From the logical relations (2.7) and (2.8), for each of the respective values $\nabla_{y_j} f(x^*, y^*)$ and $\nabla_{u_i} Df(u^*, y^*)$ of partial gradient of the primal and dual lower level objective functions, we introduce an M-**indexes subset** defined as follow:

$$\text{For} j \in S(\text{indexes of} y),\ J_j = \begin{cases} \{\bar{i} : B_{ij} > 0\} \text{ if } \nabla_{y_j} f(x^*, y^*) > 0 \\ \{\bar{i} : B_{ij} < 0\} \cup \{j\} \text{ if } \nabla_{y_j} f(x^*, y^*) < 0 \\ \{\bar{i} : B_{ij} \neq 0\} \text{ if } \nabla_{y_j} f(x^*, y^*) = 0. \end{cases} \tag{2.9}$$

$$\text{For } i \in R \text{ (indexes of } u\text{)}, \; J_{i+n_2} = \begin{cases} \{\bar{j} : B_{ij} < 0\} \text{ if } \nabla_{u_i} Df(u^*, y^*) > 0 \\ \{\bar{j} : B_{ij} > 0\} \cup \{i\} \text{ if } \nabla_{u_i} Df(u^*, y^*) < 0 \\ \{\bar{j} : B_{ij} \neq 0\} \text{ if } \nabla_{u_i} Df(u^*, y^*) = 0. \end{cases} \tag{2.10}$$

Note that when an M-indexes subset J_j is such that $\left|J_j\right| = 1$, then necessarily, at optimality one has $y_t = 0$ or $u_t = 0$, for $\{t\} = J_j$.

2.2 Using the M-principles in the ESQP algorithm

The M-principles are used in the *ESQP* algorithm as follows: let J_{j_0} be an M-indexes subset with the smallest cardinality, i.e., $\left|J_{j_0}\right| = \min_{1 \le j \le m+n_2} \left|J_j\right|$. We define an M-**network** G_M^0 as follows: we consider an initial enumeration of M-indexes subsets $J_{j_0}, J_{j_1}, J_{j_2}, \ldots, J_{j_{Ro}}, J_{j_{Ro+1}} \ldots, J_{j_{m+n_2-1}}$. Each element of J_{j_t} is a node of level $t \in \{0, 1, 2, \ldots, m+n_2-1\}$ in the network G_M^0. Within an initial enumeration of M-indexes subsets, let J_{j_t} and $J_{j_{t+1}}$ be two consecutive M-indexes subsets. An edge $(i_1, i_2) \in G_M^0$ is such that $i_1 \in J_{j_t}, i_2 \in J_{j_{t+1}}$ and $i_1 \neq i_2$. We now introduce the concept of M-tree and M-path as follows:

Definition 2.3 Let G_M^0 be an M-network and let J_{j_0} be some M-indexes subset with $1 \le j_0 \le m+n_2$; we define an M-**tree** with a root $i_0 \in J_{j_0}$ as a subset of vertexes $i_n \in G_M^0$ such that there exists a path called **M-path**, connecting i_0 to i_n.

Walk on M-trees: Let A_M^0 be the set of M-trees and let $T = \{i_0, i_1, \ldots, i_l\}$ be an M-path such that $i_0 \in J_{j_0}$; for $l > 0$, we have$i_l \in J_{j_l}$. An M-path T is such that $\alpha_i = 1$ for all $i \in T$ and $\alpha_i \geq 0$if$i \notin T$; this means that at a given step k of the *ESQP* algorithm, the lower level components are such that $T \subset \text{spp}(\bar{r})$ and $\bar{r}_i \geq 0$ if $i \notin T$. In other words, the lower level constraints with the slack variables index $i \in T$ become equalities at $\bar{z}$. Moreover, by construction of the M-network, if we have for all $t \geq 0 i_t \in J_{j_t} \cap T$, then$\alpha_{i_t} = 1$ and $\alpha_i \geq 0$ for all $i \in J_{j_t}$ such that $i \neq i_t$.

NB: In the *ESQP* algorithm, $N(r)$ is an M-indexes subset for any $r = (y, u, v, w)$.

Example 2.1 We consider the following BLP program [38].

$$\begin{array}{l} \min_{\mathrm{x,y}} (x-1)^2 + 2y_1^2 - 2x \\ \text{s.t.} \begin{cases} \min (2y_1 - x)^2 + (2y_2 - 1)^2 + xy_1 \\ \text{s.t.} \begin{cases} 4x + 5y_1 + 4y_2 \le 12, \\ -4x - 5y_1 + 4y_2 \le -4, \\ 4x - 4y_1 + 5y_2 \le 4, \\ -4x + 4y_1 + 5y_2 \le 4, \\ x, y_1, y_2 \ge 0. \end{cases} \end{cases} \end{array} \tag{2.11}$$

At the following points,$\left[\mathrm{x}^0, y_1^0, y_2^0\right]$=[0.8190, 1.6129, 0.1649] , $\left[\mathrm{u}_1^0, \mathrm{u}_2^0, u_3^0, \mathrm{u}_4^0\right]$=[0.074, 0, 0, 0.477], one has the following vectors of partial gradients: $Pf^0 = [-2.2778, -2.6810]$ and $Df(u^0, y^0) = [8.7240, -0.7240, 0.7240, 7.2760]$. The following program is a relaxation of KKT formulation of problem (2.11): QCBR$_{KKT}$

$$\min_{x,y,u,v,w} (x-1)^2 + 2y_1^2 - 2x$$
$$\text{s.t.} \begin{cases} 4x + 5y_1 + 4y_2 + w_1 = 12, & (\alpha_1) \\ -4x - 5y_1 + 4y_2 + w_2 = -4, & (\alpha_2) \\ 4x - 4y_1 + 5y_2 + w_3 = 4, & (\alpha_3) \\ -4x + 4y_1 + 5y_2 + w_4 = 4, & (\alpha_4) \\ x + 8y_1 + 5u_1 - 5u_2 - 4u_3 + 4u_4 - v_1 = 16, & (\alpha_5) \\ 8\text{y}_2 + 4u_1 - 4u_2 + 5u_3 + 5u_4 - v_2 = 4, & (\alpha_6) \\ -y_1 \leq 0, & (\alpha_7) \\ -y_2 \leq 0, & (\alpha_8) \\ -u_i \leq 0, & (\alpha_{8+i}) \\ x, v_1, v_2, w_i \geq 0, i = 1, 2, 3, 4. \end{cases} \tag{2.12}$$

The primal logical relations are:

$$\begin{cases} \alpha_1 + \alpha_4 \geq 1, \\ \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \geq 1; \end{cases}$$

the corresponding subsets of M-variables are therefore: $J_1 = \{w_1, w_4\}$ and $J_2 = \{w_1, w_2, w_3, w_4\}$, while the dual logical relations are:

$$\begin{cases} \alpha_5 + \alpha_6 \geq 1, \\ \alpha_5 + \alpha_{10} \geq 1, \\ \alpha_6 \geq 1, \\ \alpha_5 + \alpha_6 \geq 1, \end{cases}$$

with corresponding subsets of M-variables:$J_3 = \{v_1, v_2\}$, $J_4 = \{u_2, v_1\}$, $J_5 = \{v_2\}$et $J_6 = \{v_1, v_2\}$.

The adjacency matrix of the M-network G_M^0 defined by Table 1 is constructed from the ordered sequence subsets of M-variables: J_6, J_3, J_5, J_2, J_4, J_1.

The *ESQP* algorithm consists in applying a branch and bound method on sequential M-trees, using the link between each M-variable r_i and the corresponding binary

Table 1 An example of adjacency-matrix of an M-network

	v_1	v_2	v_1	v_2	v_2	w_1	w_2	w_3	w_4	u_2	v_1	w_1	w_4
V_1	0	0	0	1	0	0	0	0	0	0	0	0	0
V_2	0	0	1	0	0	0	0	0	0	0	0	0	0
V_1	0	0	0	0	1	0	0	0	0	0	0	0	0
V_2	0	0	0	0	0	0	0	0	0	0	0	0	0
V_2	0	0	0	0	0	1	1	1	1	0	0	0	0
W_1	0	0	0	0	0	0	0	0	0	1	1	0	0
W_2	0	0	0	0	0	0	0	0	0	1	1	0	0
W_3	0	0	0	0	0	0	0	0	0	1	1	0	0
W_4	0	0	0	0	0	0	0	0	0	1	1	0	0
U_2	0	0	0	0	0	0	0	0	0	0	0	1	1
V_1	0	0	0	0	0	0	0	0	0	0	0	1	1
W_1	0	0	0	0	0	0	0	0	0	0	0	0	0
W_4	0	0	0	0	0	0	0	0	0	0	0	0	0

variable α_i. At a step k of the *ESQP* algorithm such that $k_0 \geq k$, let $z^{k_0+1} = (x^{k_0+1}, y^{k_0+1}, u^{k_0+1}, v^{k_0+1}, w^{k_0+1})$ be a feasible solution of some relaxed quadratic program QCBR_T where $T \subseteq N(r^{k_0})$ and $r^{k_0} = (y^{k_0}, u^{k_0}, v^{k_0}, w^{k_0})$. Let $R \cup S$ be the set of indexes of the variables (y, u), and let $r = (y, u, v, w)$ be the vector of lower level primal-dual variables. We consider the M-indexes subset J_j, $j \in R \cup S$, as well as the M-property $\sum_{i \in J_j} \alpha_i \geq 1$ attached to each subset J_j. Branching is done by fixing some binary variable(s) α_i equal to 1 or $\alpha_i \geq 0$ (i.e. $i \in \text{spp}(r)$ or $r_i \geq 0$). Therefore, the enumeration carrying on M-indexes subsets according to a **dichotomous branching** rule: sequentially, for each M-indexes subset J_j, a single index $i_0 \in J_j$ is chosen and the corresponding boolean variable α_{i_0} is fixed to be equal to 1, and $\alpha_i \geq 0$ for all $i \in J_j, i \neq i_0$. This means that, to set $i_0 \in \text{spp}(r)$ when $\alpha_{i_0} = 1$, it consists to check if this additional constraint leads to an improving rational solution. If r_{i_0} is a slack variable, the corresponding constraint is active at the optimality. At optimality, active constraints of the lower level problem correspond to an M-indexes subset $T \subset \{i_1, i_2, \ldots, i_{n_2}, i_{1+n_2}, \ldots, i_{m+n_2}\}$, where $i_j \in J_j$, $j \in R \cup S$. From M-constraints, the *ESQP* algorithm finds such M-indexes subset T which may lead to optimality.

The *ESQP* algorithm uses a branch-and-bound with depth-first search technique, within the M-trees. Let z* be the current feasible solution of problem (2.1); let A_m be an M-tree and let T be an M-path with nodes included in an M-indexes subset $N(r^*)$ such that $T = \{i_0, i_1, i_2 \ldots\}$ with $i_0 \in J_{j_0}$ and $i_t \in J_{j_t}, t \geq 1$. The quadratic problems $\text{QCBR}_{\{i_0\}}, \text{QCBR}_{\{i_0,i_1\}}, \ldots, \text{QCBR}_T$ are solved sequentially in the *ESQP* algorithm. Let $\bar{z}$ be a solution of one of the above quadratic problems. A backtracking is performed if we have one of the following conditions:

–(i) an improving rational solution $\bar{z}$ has been computed;
–(ii) one of the above quadratic programs does not have a feasible solution;
–(iii) $\bar{z} = (\bar{x}, \bar{y}, \bar{u}, \bar{v}, \bar{w})$ computed from one of the above quadratic programs such that $(\bar{x}, \bar{y})$ is a rational solution and $F(\bar{x}, \bar{y}) < F(x^*, y^*)$.

Let n_{is} be the level of a node i_s in the network G_M^0, and A_M^0 the set of M-trees. If a backtracking condition is verified at a node i_s contained in the M-path T, then the walk along the corresponding M-tree is stopped. From the node i_s, we trace back indexes in T. Let $T_B = \{i_s, i_{s-1}, \ldots, i_b\}$ be a subset of T containing backtracking nodes. Then, in the next quadratic problem to be solved in *ESQP* algorithm, the constraints $i \in \text{spp}(r)$ for$i \in T_B$ (i.e. $r_i = 0$ or $\alpha_i = 1$ for all $i \in T_B$) are relaxed and transformed into $r_i \geq 0$ for $i \in T_B$ (i.e. $\alpha_i \geq 0$ for all $i \in T_B$). From the node $i_{b-1} \in T$ including at least one successor which has not been investigated, other investigable M-paths are constructed. The investigation procedure is stopped for an M-tree if there is no more investigable M-path from the root of such M-tree. Let

$$Ro = \min_{A_m \in A_M^0} \{n_{is} : i_s \in A_m\}. \tag{2.13}$$

Ro represents the smallest value on the set of levels of nodes corresponding to backtracking nodes i_s in the network G_M^0.

Remark 2.2 Let $\sum_{i \in J_j} \alpha_i \geq 1$ be a monotonicity constraint and let J_j be an M-indexes subset. With the dichotomous branching rule, only one index $i_0 \in J_j$ is chosen and the corresponding boolean variable α_{i_0} is fixed to be equal to 1 and $\alpha_i \geq 0$ for all $i \in J_j, i \neq i_0$; hence, the M-paths for which $\alpha_{i_0} = 1$, for all $i_0 \in K$, with $K \subseteq J_j$ and $|K| \geq 2$ may not be investigated. According to the construction of *Ro* and the backtracking rule, it is easy to show that, investigations on the network consisting of M-indexes subsets $J_{j_0}, J_{j_1}, J_{j_2}, \ldots, J_{j_{Ro}}$can not improve the value of the current solution. Let $J_{j_{Ro+1}}, J_{j_{Ro+2}}, \ldots, J_{j_{m+n_2-1}}$ be the M-indexes subsets for which some indexes have not been investigated by the *ESQP* algorithm, due to the backtracking or dichotomous branching rules. An investigation on some of the indexes in the M-indexes subsets $J_{j_{Ro+1}}, J_{j_{Ro+2}}, \ldots, J_{j_{m+n_2-1}}$ may improve the solution.

According to the above remark, the construction of a new relevant M-path follows that of a sequence of M-networks $G^t_M, t = 1, 2, \ldots, m + n_2 - Ro$. We may then renew the numeration of levels of nodes in subsets $J_{j_i}, i = 0, 1, 2, \ldots, m + n_2 - 1$. Therefore, we derive an M-network G^1_M from the following ordered sequence of M-indexes subsets $J_{j_{Ro+1}}, J_{j_0}, J_{j_1}, J_{j_2}, \ldots, J_{j_{Ro}}, J_{j_{Ro+2}}, \ldots, J_{j_{m+n_2-1}}$. All nodes of level $Ro+1$ in the M-network G^0_M become nodes of level 1 in the M-network G^1_M, and nodes of level $2, 3, \ldots, Ro$ in G^1_M correspond to nodes of level $1, 2, 3, \ldots, Ro - 1$, respectively, in G^0_M. In the same way, from the network G^1_M, we derive an M-network G^2_M according to an enumeration of M-indexes subsets $J_{Ro+2}, J_{Ro+1}, J_{j_0}, J_{j_1}, \quad J_{j_2}, \ldots, J_{j_{Ro}}, J_{j_{Ro+3}}, \ldots, J_{j_{m+n_2-1}}$, etc. …, for a sequence of M-networks $G^t_M, t = 3, 4, \ldots, m + n_2 - Ro$.

2.3 Principles of an implicit enumeration algorithm

In general, the implicit enumeration may be related to search in tree algorithms. A branching algorithm consists in an implicit enumeration procedure within the set of solutions of the problem to solve. Branching algorithms are often used in integer or mixed programming. These algorithms were introduced by Little et al. [33] as well as Bertier and Roy [8]. A synthesis presentation of these notions can be found in [26]. We use an implicit enumeration principle carrying on continuous variables to solve the relaxed quadratic program QCBR_T.

The *ESQP* algorithm consists of walks through M-paths as defined in the previous subsection. More precisely, let T be a path in an M-tree A_m. At a step k of the *ESQP* algorithm, for $k_0 \geq k$**,** let z^k be an improvable solution; if z^{k_0+1} is a solution of some relaxed program QCBR_T, $T \subset N(r^{k_0})$ and $z^{k_0+1} \in \mathfrak{F}^{KKT}$ improves the value of the upper level objective function F, we say that the **evaluation** $(z^{k_0}_t = 0, t \in T)$ corresponding to the set of variables $z_t, t \in T$ **is successful**. After what, we select another path if a backtracking condition is satisfied, or if there exists at least one M-tree which has not been investigated. At each stage of the *ESQP* algorithm, the goal is to find a subset of indexes of lower level primal-dual variables T such that $T \subset \text{spp}(r^{k_0+1})$. We write beside a node of the M-network, the index of the corresponding lower level variable, as well as the triplet $(T, z^{k_0}, F(z^{k_0}))$. An edge (i, j) has the letter S, if because of a backtracking one can no longer walk through the M-tree from the node i.

Remark 2.3 Let's consider the BPL problem (1.1).

(1) There exist $m+n_2$ M-indexes subsets, and each of them has at most $N_e = \max(m, n_2)+1$ elements.
(2) By construction, an M-network has at most $N = N_e(m+n_2)$ nodes (some may be identical). A walk on an M-tree is an elementary path of length $p(p = 1, 2, \ldots, m + n_2)$. Let N be the order of a network; according to its properties, the number of paths of length p is equal to $A^{p+1}_N = \frac{N!}{(N-p-1)!}$. Therefore, we may guess that the number of paths is equal to $O\,(N!)$.
(3) Let A_I be an M-tree. The number of children of a node $i \in A_I$, written as $d_I(i)$ is the degree of that node. By constructing an M-network, we have $d_I(i) \leq N_e - 1$.

The number N_{AM} of M-trees with nodes such that $d_I(i) = k$, is given by $N_{AM} = (N-1)^{(N-k-1)} C_{k-1}^{N-2}$ (see [17]).

3 The enumeration sequential quadratic programming algorithm (*ESQP*)

Let's consider the problem (2.1); while relaxing the complementarity constraints $u^t w = v^t y = 0$, and adding constraints $z_t = 0, t \in T$; we obtain the quadratic programming QCBR$_T$ formulated as in (2.2). To compute an initial rational solution, the following methods can be used.

Method 1. First, solve the following quadratic program:

$$\min_{x,y} F(x,y) \\ \text{s.t.} \begin{cases} x \in P, \\ A_2 x + B_2 y \leq b_2, \\ x, y \geq 0. \end{cases} \tag{3.1}$$

Let $(x\prime, y\prime)$ be an optimal solution of problem (3.1). Set $x^* = x\prime$, then the following primal lower level problem is solved:

$$\min_y f(x^*, y) \\ \text{s.t.} \begin{cases} B_2 y \leq b_2 - A_2 x^*, \\ y \geq 0, \end{cases} \tag{3.2}$$

with an optimal solution is y^*, u^* being some dual optimal solution.

Method 2. Here, the problem (3.1) of variables (x, y) is solved, while replacing the upper level objective function $F(x, y)$ with the lower level objective function. Then an optimal solution is (x^*, y^*), u^* is again some corresponding dual optimal solution.

Let (1.1) be a convex quadratic BLP problem; $valF$ represents the current value of the upper level objective function, while a current rational solution is $s^* = (x^*, y^*)$. Using the notations introduced previously, first we present the procedure $INVEST(G_M^t, s^*)$ used in the *ESQP* algorithm in order to investigate an M-network G_M^t; let A_M^t be the sets of M-trees in G_M^t, then the procedure $INVEST(G_M^t, s^*)$ follows:

Procedure $INVEST(G_M^t, s^*)$.
If $t = 0$, set $Ro = m + n_2$. Set $M_T = A_M^t$(set of trees). While $M_T \neq \emptyset$, execute A-procedure.

A-Procedure:
Select an M-tree $A_m \in M_T$, then set $M_T = A_M^t \backslash A_m$ and
$CH_T = \{Ch \in A_m : Ch \text{ is an elementary path}\}$. While $CH_T \neq \emptyset$, execute B-procedure.

B-Procedure:
Select a path $Ch \in CH_T$, then set $P = Ch$,$CH_T = CH_T \backslash \{Ch\}$and $T = \emptyset$.
While (i) $P \neq \emptyset$ or (ii) a backtracking condition is not satisfied, execute C-procedure.

C-Procedure:
Select an index $i \in P$of level n_i in G_M^t, then set $P = P \backslash \{i\}$ and $T = T \cup \{i\}$.
Let z^{k+1} be a solution of some program QCBR$_T$, $T \subset N(r^k)$ (if it exists).
Backtracking is performed on a node i_0 within the path Ch if:
(i) the family of programs QCBR$_T$, $T \subset N(r^k)$has no solution or
$F(x^{k+1}, y^{k+1}) \geq valF$.

(ii) $F(x^{k+1}, y^{k+1}) < valF$. Set $valF = F(x^{k+1}, y^{k+1})$,$k = k + 1$. If $z^{k+1} \in \mathfrak{F}^{KKT}$, then compute
$d^* = (\nabla_y f(x^*, y^*), \nabla_u Df(u^*, y^*))$, set $z^* = z^{k+1}$and update the M-constraints and the M-indexes
subsets. Remove from Ch the node i_0 as well as its children, and stop C-procedure.

End of C-procedure.

End B-procedure.

If$t = 0$, set $Ro = \min(Ro, n_{i_0})$.

End of A-procedure. (Logout with a rational solution $s^* = (x^*, y^*)$and the number Ro if $t = 0$).

Now, the *ESQP* algorithm is as follows:

The *ESQP***algorithm** (solving the problem (2.1)).

Step 0 (Initialization).

Set $k = 0$, compute $z^* = (x^*, y^*, u^*, v^*, w^*)$ and set $valF = F(x^*, y^*)$ and $z^* = z^0$.

For $j \in R \cup S$, compute the list L_M of all M-indexes subsets J_j.

For any subset J_j such that $\left|J_j\right| = 1$, set $z_t = 0$,with $\{z_t\} = J_j$; remove the subset J_j from the list L_M, then construct the M-network G_M^0.

Execute procedure *INVEST*(G_M^0, s^*) to compute an improving rational solution z^*, the numbers Roand d^*.

For $t = 1, 2, \ldots, m + n_2 - Ro$, construct each of the M-network G_M^t.

Step 1 (Computing an improving rational solution).

For $t = 1, 2, \ldots, m + n_2 - Ro$, execute the procedure *INVEST*(G_M^t, s^*).

Step 2: (Optimality).

Stop when Step 1 is completed, (x^*, y^*) is an optimal solution of the problem (QCBL).

When the current solution $z^* = (x^*, y^*, u^*, v^*, w^*)$ of the problem (2.1) is such that (x^*, y^*) is an improvable rational solution of the problem (1.1), the following proposition shows that, to compute an improving rational solution, the *ESQP* algorithm (i) may compute an improving primal-dual rational solution $\bar{z} \in \mathfrak{F}^{KKT}$, and (ii) solves a finite number of quadratic programming problems.

Proposition 3.1 *We consider the convex BLP* (1.1) *such that the set of feasible solution* Ω *is nonempty and bounded. At a given step k of the ESQP algorithm, let* $z^* = (x^*, y^*, u^*, v^*, w^*)$ *be a feasible solution of the KKT formulation* (2.1)*. Let* $k_0 \geq k$*; if* (x^*, y^*) *is an improvable solution of problem* (1.1)*, then the ESQP algorithm may compute an improving rational solution* $\bar{z} = z^{k_0+1}$ *of problem* (2.1) *by solving a finite number of quadratic programs.*

Proof Assume that the current solution (x^*, y^*) of problem (1.1) is an improvable rational solution at a step k of the *ESQP* algorithm, and let $k_0 \geq k$. The *ESQP* algorithm may then compute an improving rational solution $z^{k_0+1} \in \mathfrak{F}^{KKT}$ by means of solving the family of quadratic programs QCBR$_T$, $T \subset N(r^{k_0})$. If in any path T in any M-tree, any programs QCBR$_T$,$T \subset N(r^{k_0})$ does not have a solution, then there is no primal-dual improving rational solution $\bar{z} \in \mathfrak{F}^{KKT}$ such that $T \subset \text{spp}(\bar{z})$ and $F(x^*, y^*) < F(\bar{x}, \bar{y})$, which is absurd. Therefore there exists at least a path T in an M-tree such that $\bar{z} \in \mathfrak{F}^{KKT}$is an optimal solution of one of the program QCBR$_T$. So the *ESQP* algorithm may compute an improving rational solution when the current rational solution (x^*, y^*) is an improvable rational solution of the problem (1.1).

Now, according to remarks 2.3, every M-network G_M^j includes a finite number of trees. Moreover, each of these trees has a finite number of paths. On each path, we solve a finite number of quadratic programs of type QCBR_T in order to compute an improving rational solution. Hence, from an improvable rational solution $(x^*, y^*) \in IR$ at given step k of the *ESQP* algorithm, one computes an improving rational solution $\bar{z} = z^{k_0+1}$ of the problem (2.1) by solving a finite number of quadratic programs where $k_0 \geq k$. □

Given the current rational solution (x^*, y^*), the iterative process of the *ESQP* algorithm stops, if at the end of the enumeration carried on every path of all of the M-trees, we can not compute an improving rational solution $\bar{z}$, which is a feasible solution of the problem (2.1) such that $F(\bar{x}, \bar{y}) < F(x^*, y^*)$. The *ESQP* algorithm solves the problem (1.1) in a finite number of steps. Let $z^* = (x^*, y^*, u^*, v^*, w^*)$ be a feasible solution of the KKT formulation of the BLP (1.1) and let's consider the set of M-trees. Then one has the following situations:

(a) For any M-tree A_I and for any path T inA_I such that $T \subseteq N(r^*)$, the quadratic program QCBR_T has no solution. (x^*, y^*) is an optimal solution of the problem (1.1).
(b) Let $\bar{z} = (\bar{x}, \bar{y}, \bar{u}, \bar{v}, \bar{w})$ be the optimal solution of some quadratic program QCBR_T. The following two cases are foreseeable:
   (i) For any M-tree A_I and for any path T in A_I such that $T \subseteq N(r^*)$, $\bar{z}$ is not an improving rational solution. Therefore, (x^*, y^*) is an optimal solution of the problem (1.1) computed by the *ESQP* algorithm.
   (ii) For a given M-tree A_I, there exists a path T in A_I such that $T \subseteq N(r^*)$, $\bar{z}$ is an improving rational solution. Therefore, (x^*, y^*) is not an optimal solution of the problem (1.1).

4 An example

We now consider the formulation (2.11) of the Example 2.1; the relaxation of its KKT formulation is given by (2.12).The initial solution is:

$$\begin{aligned}(x^*, y^*) &= \left[\mathrm{x}^0, y_1^0, y_2^0\right] = [0.8190, 1.6129, 0.1649]\,, u^* = \left[\mathrm{u}_1^0, \mathrm{u}_2^0, \mathrm{u}_3^0, \mathrm{u}_4^0\right] \\ &= [0.0739, 0, 0, 0.477]\,; v^* = \left[v_1^0, v_2^0\right] = [0, 0]\,; w^* = \left[w_1^0, w_2^0, \mathrm{w}_3^0, w_4^0\right] \\ &= [0, 6.681, 6.351, 0]\,,\end{aligned}$$

while the primal and dual gradients of the lower level objective functions are given by:

$$\begin{aligned}\left[Pf_1^0, Pf_2^0\right] = [-2.277, -2.681]\,; &\left[Df_1^0, Df_2^0, Df_3^0, Df_3^0\right] \\ &= [8.724, -0.724, 0.724, 7.276]\,.\end{aligned}$$

From the adjacency-matrix of the M-network G_M^0 on Table 1, we construct M-trees on Fig. 1. On the tree (1) in Fig. 1, the primal-dual solutions of the relaxed program $\mathrm{QCBR}_{\{v_1\}}$ follow:

$$\begin{aligned}(x^*, y^*) &= \left[\mathrm{x}^1, y_1^1, y_2^1\right] = [1.5185, 1.1852, 0]\, and u^* = \left[\mathrm{u}_1^0, \mathrm{u}_2^0, \mathrm{u}_3^0, \mathrm{u}_4^0\right] = [1, 0, 0, 0]\,, \\ valF &= [0.0412].\end{aligned}$$

The primal-dual partial gradients of the lower level objective function are

$$\left[Pf_1^0, Pf_2^0\right] = [-5, -4]\,; \left[Df_1^0, Df_2^0, Df_3^0, Df_3^0\right] = [5.9259, 2.0741, -2.0741, 10.0741]\,.$$

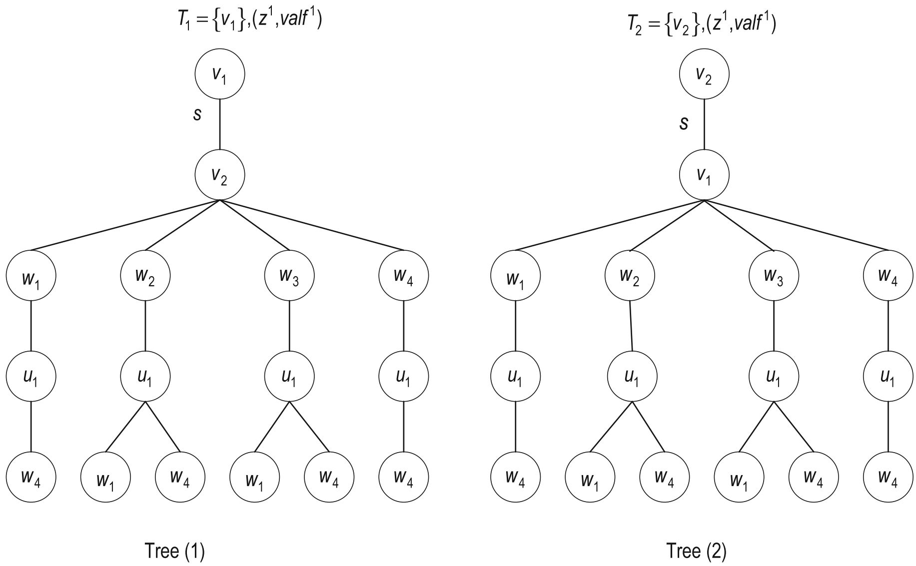

Fig. 1 M-trees of the M-network G_M^0

Walking on that tree is stopped.

Moreover, on tree (2) of Fig. 1, the program $\mathrm{QCBR}_{\{v_2\}}$ does not have a solution. Walking on that tree is stopped. We have $Ro = 1$, as an improving rational solution is computed on a node of level 1 on the network G_M^0. The M-constraints are updated, then we have:

– the primal logical relations:

$$\begin{cases} \alpha_1 + \alpha_4 \geq 1, \\ \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \geq 1, \end{cases}$$

with subsets of M-variables: $J_1 = \{w_1, w_4\}$ et $J_2 = \{w_1, w_2, w_3, w_4\}$;

– the dual logical relations:

$$\begin{cases} \alpha_5 + \alpha_6 \geq 1, \\ \alpha_6 \geq 1, \\ \alpha_5 + \alpha_{11} \geq 1, \\ \alpha_5 + \alpha_6 \geq 1, \end{cases}$$

with subsets of M-variables:$J_3 = \{v_1, v_2\}$, $J_4 = \{v_2\}$, $J_5 = \{u_3, v_1\}$et $J_6 = \{v_1, v_2\}$.

From the ordered sequence of subsets of M-variables $(J_1, J_2, J_3, J_4, J_5, J_6)$, we construct the M-network G_M^1 represented by the Table 2. The corresponding trees are indicated in the Figs. 2 and 3. Note that, at optimality, we have $v_2 = 0$ as $J_4 = \{v_2\}$. However, in order to illustrate how the *ESQP* algorithm works, we do not take into account the constraint $v_2 = 0$ at optimality.

The results of walks on these M-trees are as follow:

On the tree (1) of Fig. 2, solving the relaxed program $\mathrm{QCBR}_{\{w_1\}}$ leads to the following rational primal-dual solution:

Table 2 Adjacency-matrix of the second M-network G_M^1

	w_1	w_4	w_1	w_2	w_3	w_4	v_1	v_2	v_2	u_3	v_1	v_1	v_2
W_1	0	0	0	1	1	1	0	0	0	0	0	0	0
W_4	0	0	1	1	1	0	0	0	0	0	0	0	0
W_1	0	0	0	0	0	0	1	1	0	0	0	0	0
W_2	0	0	0	0	0	0	1	1	0	0	0	0	0
W_3	0	0	0	0	0	0	1	1	0	0	0	0	0
W_4	0	0	0	0	0	0	1	1	0	0	0	0	0
V_1	0	0	0	0	0	0	0	0	1	0	0	0	0
V_2	0	0	0	0	0	0	0	0	0	0	0	0	0
V_2	0	0	0	0	0	0	0	0	0	1	1	0	0
U_3	0	0	0	0	0	0	0	0	0	0	0	1	1
V_1	0	0	0	0	0	0	0	0	0	0	0	0	1
V_1	0	0	0	0	0	0	0	0	0	0	0	0	0
V_2	0	0	0	0	0	0	0	0	0	0	0	0	0

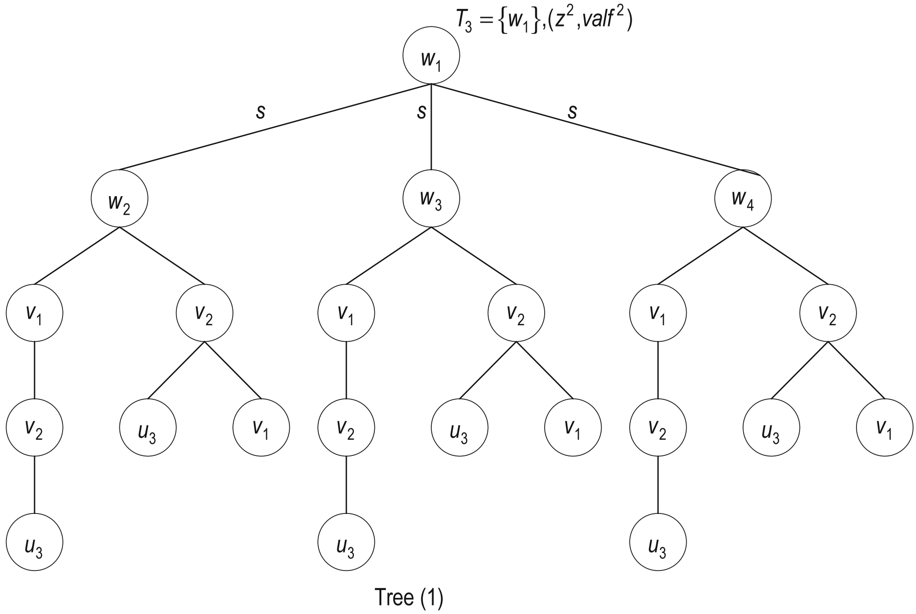

Fig. 2 First M-tree in the M-network G_M^1

$(x^*, y^*) = \left[x^2, y_1^2, y_2^2\right] = [1.8889, 0.8889, 0]$, and $u^* = \left[u_1^0, u_2^0, u_3^0, u_4^0\right] = [1.4, 0, 0, 0]$,
$ValF = -1.4074$

The primal and dual gradients of the lower level objective functions are

$\left[Pf_1^0, Pf_2^0\right] = [-7, -4]$; $\left[Df_1^0, Df_2^0, Df_3^0, Df_3^0\right] = [4.444, 3.5556, -3.5556, 11.5556]$.

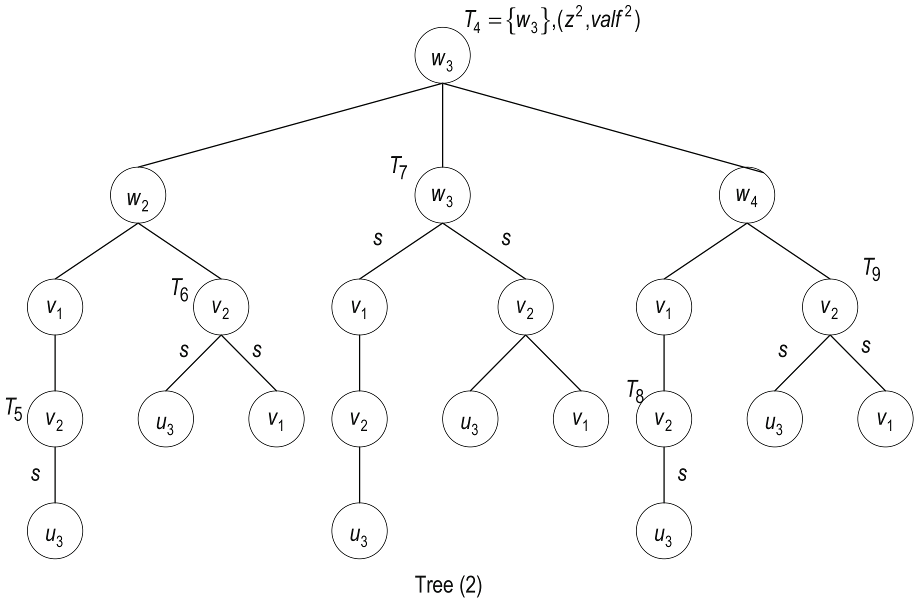

Fig. 3 Second M-tree in the M-network G_M^1

Then the walk on tree (1) is stopped.

On the tree (2) of Fig. 3, one has:

- The value $ValF^2$of the upper level objective function after solving the relaxed program QCBR_{T_2} with $T_2 = \{w_4, w_1, v_1, v_2\}$ is such that $ValF > valF^2$. A backtracking is performed.
- The previous result is gotten for program QCBR_{T_3} where $T_3 = \{w_4, w_1, v_2\}$.
- The programs QCBR_{T_4}, QCBR_{T_5} and QCBR_{T_6} where $T_4 = \{w_4, w_2\}$, $T_5 = \{w_4, w_3, v_1, v_2\}$ and $T_6 = \{w_4, w_1, v_2\}$ do not have solution.

The walk on tree (2) is finished.

Each walks on the M-trees is built from the following ordered sequences of subsets of M-variables

$$(J_2, J_1, J_3, J_4, J_5, J_6), (J_3, J_2, J_1, J_4, J_5), \quad (J_4, J_3, J_2, J_1, J_5, J_6),$$
$$(J_5, J_4, J_3, J_2, J_1, J_6), (J_6, J_5, J_4, J_3, J_2, J_1)$$

do not improve the value of the current solution. Then the optimal solution is

$$(x^*, y^*) = \left[\mathrm{x}^1, y_1^1, y_2^1\right] = [1.8889, 0.8889, 0] \text{ and } ValF = \text{-}1.4074.$$

5 Numerical experiences of the *ESQP* algorithm

The *ESQP* algorithm was coded in MATLAB and all computations were performed on a PC Pentium 4 (processor 3.2 GHZ, 1.24 GB of RAM). We illustrate our algorithm in two ways:

- First, we solve small size problems known in the literature.

– Then, to solve medium size problems, the data are randomly generated using MATLAB generator. We compare the results computed by *ESQP* algorithm with those computed by the algorithm of Bard and Moore [3] (*BM*). To solve medium size problems, we consider the general model of a quadratic BLP problem (1.1) where

$$Q = \begin{bmatrix} Q_x & Q_{xy} \\ Q_{xy}^t & Q_y \end{bmatrix} \text{ and } N = \begin{bmatrix} N_x & N_{xy} \\ N_{xy}^t & N_y \end{bmatrix},$$

with $Q_x = O_{n_1 \times n_1} \in \mathbb{R}^{n_1 \times n_1}$,$Q_y = I_{n_2} \in \mathbb{R}^{n_2 \times n_2}$, $Q_{xy} = O_{n_1 \times n_2} \in \mathbb{R}^{n_1 \times n_2}$,$c_1 = c_2 = O_{n_1}$, $d_1 = d_2 = O_{n_2}$ $N_x = I_{n_1} \in \mathbb{R}^{n_1 \times n_1}$, $N_y = I_{n_2} \in \mathbb{R}^{n_2 \times n_2}$, $N_{xy} = O_{n_1 \times n_2} \in \mathbb{R}^{n_1 \times n_2}$, O_t is a null matrix, or a null vector while I_t is the identity matrix.

Data are randomly generated as in [2], with a density of 20%: the elements of B_2 are chosen between -20 and 20, while those of A_1andA_2 are taken from interval [0, 20]. The coefficients of c^1 belong to the interval $[-10, 10]$ and those of d^1 belong to [10, 20]. The elements of d^2 are chosen between 0 and 10. The elements of b_1 and b_2 are randomly chosen between -40 and 40. Finally, the additional lower level constraint $\sum_{j=1}^{n_1} x_j + \sum_{j=1}^{n_2} y_j \leq n_1 + n_2$ is added to obtain a bounded relaxed problem. The vector (n, n_2, m) characterizes the size of each problems solved.

Let's recall that the *ESQP* algorithm solves two types of quadratic programs at every stage:

– A relaxed program of type QCBR_T. The total number of such programs solved for a given problem is designated by *ND*; this quantity represents the number of programs of the same type solved by the *BM* algorithm.
– If $\bar{z}$ is a solution of some relaxed program QCBR_T, we verify that this solution is rational; then we solve the lower level problem. *NDI* represents the number of times this verification occurs for a given problem, whereas *NDA* represents the number of the rational solutions contributing to improve the value of the upper level objective function. The columns of the table representing the values of *ND* and *NDI* include two numbers: the smaller value corresponds to the stage of the algorithm computing the optimal solution, whereas the higher value designates the total number of programs solved. Performance measures include *CPU* time (seconds) represented in a same way.

Small size problems used to test the *ESQP* algorithm follow:

Example 5.1 [35].

$$\min_{\mathrm{x,y}} (x_1 - 30)^2 + (x_2 - 20)^2 - 20y_1 + 20y_2$$
$$\text{s.t.} \begin{cases} x_1 + 2x_2 \leq 30, \\ x_1 + x_2 \geq 20, \\ 0 \leq x_i \leq 15, \quad i = 1, 2, \\ \min_{\mathrm{y}} (x_1 - y_1)^2 + (x_2 - y_2)^2, \\ \text{s.t. } 0 \leq y_i \leq 15, \ \ i = 1, 2. \end{cases}$$

Example 5.2 [35].

$$\min_{\mathrm{x,y}} y_1^2 + y_2^2 - y_1 y_3 - 4y_2 - 7x_1 + 4x_2$$
$$\text{s.t.} \begin{cases} x_1 + x_2 \leq 1, \\ x_1, x_2 \geq 0, \\ \min_y y_1^2 + 0.5y_2^2 + 0.5y_3^2 + y_1 y_2 + (1 - 3x_1)y_1 + (1 + x_2)y_2 \\ \text{s.t.} \begin{cases} 2y_1 + y_2 - y_3 + x_1 - 2x_2 + 2 \leq 0, \\ y^t = (y_1, y_2, y_3) \geq 0. \end{cases} \end{cases}$$

For Example 5.1, the Lagrange function of the lower level problem is:

$$L(y, \lambda) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \lambda_1(y_1 - 15) + \lambda_2(y_2 - 15),$$

and the corresponding first order optimality conditions follow:

$$\begin{array}{l}\text{(a) } \frac{\partial L(y,\lambda)}{\partial y_i} = -2(x_i - y_i) + \lambda_i \begin{cases} = 0 & \text{if } y_i > 0, \\ \leq 0 & \text{else.} \end{cases} \\ \text{(b) } \lambda_i(y_i - 15) = 0, \ \text{ for } i = 1, 2.\end{array}$$

But, the solution computed by Muu and Quy [35] is $(x_1^*, x_2^*) = (15, 7.5)$ and $(y_1^*, y_2^*) = (10, 7.5)$.

$$\text{(b)} \Rightarrow \lambda_1^* = \lambda_2^* = 0 \text{ and (a)} \Rightarrow x_1^* = y_1^* \text{ and } x_2^* = y_2^*.$$

This solution does not satisfy optimality conditions for the lower level problem; it is nearly identical to the solution computed by the genetic algorithm of Wang et al. [44], contrary to the solution computed by our algorithm. For small size problems, the *BM* algorithm computes the same solution as our algorithm (see Tables 3, 4 and 5).

Tables 6 and 7 include the results computed, respectively, by the *BM* and *ESQP* algorithms, on the convex quadratic BLP of medium size problems. We have $n = n_1 + n_2$. Note that the *BM* algorithm was not able to solve the problem of size $(n, n_1, m_1) = (50, 25, 9)$ contrary to the *ESQP* algorithm; the *ESQP* algorithm was not able to solve the problem of size $(n, n_1, m_1) = (46, 23, 9)$ contrary to the *BM* algorithm. These failures may be attributed to MATLAB environment. For problems of size $n \geq 30$, we see that the *BM* algorithm proved to be slower than the *ESQP* algorithm.

Finally, the *ESQP* algorithm constitutes a compromise computing a good evaluation of an optimal solution of a convex quadratic BLP at a least cost. On most of the examples used, the *ESQP* computed a globally optimal solution.

Table 3 Results on Examples 5.1 and 5.2 (*ESQP*)

Example no	Results in the *ESQP* algorithm (x, y)	F	f	*ND*	*NDI*	*NDA*	*Ro*
5.1	(15, 7.5, 15, 7.5)	231.25	0	6	2	2	1
				7	2		
5.2	(0.611, 0.389, 0, 0, 1.833)	0.6389	1.6806	2	2	2	1
				77	10		

Table 4 Examples 5.1 and 5.2 (Results in the references)

Example no	Results in the references (x, y)	F	f
2.1	(1.889, 0.889, 0)	−1.407	7.617
5.1	(15,7.5,10,7.5)	331.25	25
5.2	(0.609, 0.391, 0, 0, 1.828)	0.6426	1.6708

Table 5 Results on Examples 5.1 and 5.2 (*BM* algorithm)

Example no	Results in the *BM* algorithm (x, y)	*F*	*f*	*ND*	*NDI*	*NDA*
5.1	(15, 7.5, 15, 7.5)	231.25	0	7	2	2
				13	2	
5.2	(0.611, 0.389, 0, 0, 1.833)	0.6389	1.6806	6	2	2
				7	3	

Table 6 Results with *BM* algorithm—problems of Audet et al. [2]

n	n_1	m_1	*ND*	*NDI*	*NDA*	*CPU*	F_{BM}^{opt}
12	6	5	5	2	2	2	30
			59	6		28	
16	8	6	3	2	2	0.4	160
			69	16		38	
20	10	13	16	3	2	20	240
			155	37		163	
30	10	6	50	7	3	157	510
			237	32		402	
40	20	8	6	2	2	11	880
			1,099	198		5,001	
46	23	9	10	2	2	79	1,150
			1,088	273		5,001	
50	25	9	1	1	1	0	625
			5	2		8	

Table 7 Results with the *ESQP* algorithm—problems of Audet et al. [2]

n	n_1	m_1	*Ro*	*ND*	*NDI*	*NDA*	*CPU*	F_{PQEI}^{opt}
12	6	5	6	13	2	2	2.4	30
				88	11		29.5	
16	8	6	1	13	2	2	48	160
				36	4		63	
20	10	13	2	77	3	2	64	240
				88	4		72	
30	10	6	2	62	23	5	57	510
				123	23		158	
40	20	8	1	133	6	2	260	880
				179	20		407	
46	23	9	1	1	1	1	0	1,058
				83	1		345	
50	25	9	1	120	6	3	751	1,350
				183	31		992	

6 Conclusion and final remarks

The *ESQP* algorithm using the M-analysis, investigates some active constraints of the KKT formulation of problem (2.1). Facchinei et al. [23] present a new technique which identifies active constraints in a neighborhood of a solution of a nonlinear program with inequality constraints. This technique may be applied to the KKT formulation of the general convex bilevel programming problem, but it cannot remove the combinatorial aspect of this problem. However, it may be interesting first to define an appropriate identification function (see [23,32]) to compute the solution of the problem (2.1); this will be done in a further research.

The test calculations with the *ESQP* algorithm presented in this paper seem to indicate that the proposed algorithm is better than algorithms proposed by Muu and Quy [35] or Wang et al. [44]. To solve problem (1.1), the algorithm of Muu and Quy [35] uses a merit function technique to transform the initial problem and approximates the resulting problem which is then solved within a branch and bound framework. This algorithm, may not lead to a rational solution for problem (1.1) as shown in Example 5.1 because of the approximation method used.

The *ESQP* algorithm computes almost same solutions as Bard and Moore's algorithm [3]. However, it would be useful to find a method leading analytically to an optimal investigation of M-indexes subsets. With the promising results of our computational results, principles of the *ESQP* algorithm would be used to solve the general convex BLP problem; the *ESQP* algorithm would be then combined to a trust region method.

If one can prove that walks on M-trees explore all the combinations of indexes leading to an improving rational solution, then, the *ESQP* algorithm may be considered as an exact globally convergent algorithm, even though it is built from necessary optimality conditions, as well as the *HJS* algorithm in [27]. However, the *ESQP* algorithm may be considered as a heuristic computing in at least one local solution of a BLP problem.

Acknowledgments The author would like to thank anonymous referees for their constructive review and useful comments or remarks that help him very much in revising and improving the paper.

References

1. Abdou-Kandil, H., Bertrand, P.: Government-private sector relations as a stackelberg game: a degenerate case. J. Econ. Dyn. Control **11**, 513–517 (1987). doi:10.1016/S0165-1889(87)80004-0
2. Audet, C., Hansen, P., Jaumard, B., Savard, G.: Links between linear bilevel and mixed 0–1 programming problems. J. Optim. Theory Appl. **93**(2), 273–300 (1997). doi:10.1023/A:1022645805569
3. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. SIAM J. Sci. Stat. Comput. **11**, 281–292 (1990). doi:10.1137/0911017
4. Bard, J.F.: Convex two-level programming. Math. Program. **40**(1), 15–28 (1988). doi:10.1007/BF01580720
5. Bard, J.F.: Optimality conditions for the bilevel programming problem. Nav. Res. Logist. Q. **31**, 13–26 (1984). doi:10.1002/nav.3800310104
6. Bard, J.F., Plummer, J.C., Sourie, J.C.: A bilevel programming approach to determining tax credits for biofuel production. Eur. J. Oper. Res. **120**, 30–43 (2000). doi:10.1016/S0377-2217(98)00373-7
7. Ben-Ayed, O., Blair, C.: Computational difficulties of bilevel linear programming. Oper. Res. **38**, 556–560 (1990). doi:10.1287/opre.38.3.556
8. Bertier, P., Roy, B.: Procédures de résolution pour une classe de problème pouvant avoir un caractère combinatoire. Cahier du centre d'études de recherche opérationnelle, 6 (1964)
9. Bialas, W.F., Karwan, M.H.: Two-level linear programming. Manage. Sci. **30**(8), 1004–1020 (1984). doi:10.1287/mnsc.30.8.1004
10. Brotcorne, L., Labbé, M., Marcotte, P., Savard, G.: A bilevel model and solution algorithm for a freight tariff setting problem. Transp. Sci. **34**, 289–302 (2000). doi:10.1287/trsc.34.3.289.12299

11. Calvete, H.I., Gale, C.: On the quasiconcave bilevel programming problem. J. Optim. Theory Appl. **98**(3), 613–622 (1998). doi:10.1023/A:1022624029539
12. Campêlo, M., Dantas, S., Scheimberg, S.: A note on a penalty function approach for solving bilevel linear programs. J. Glob. Optim. **16**, 245–255 (2000). doi:10.1023/A:1008308218364
13. Candler, W., Norton, R.: Multi-level programming and development policy. World bank development research center discussion paper, vol. 258, Washington, May (1977b)
14. Candler, W., Norton, R.: Multi-level programming. World bank development research center discussion paper, vol. 20, Washington, January (1977a)
15. Candler, W., Townsley, R.: A linear two-level programming problem. Comput. Oper. Res. **9**, 57–76 (1982). doi:10.1016/0305-0548(82)90006-5
16. Chen, Y., Florian, M.: The nonlinear bilevel programming problem: formulation, regularity and optimality conditions. Optimization **32**, 193–309 (1995). doi:10.1080/02331939508844048
17. Clarke, L.E.: On cayley's formula for counting trees. Proc. Camb. Philos. Soc. **59**, 509–517 (1963). doi:10.1017/S0305004100037178
18. Colson, B., Marcotte, P., Savard, G.: Bilevel programming: a survey. $4OR$ Q. J. Oper. Res. **4**(R 3), 87–107 (2005)
19. Dempe, S.: A simple algorithm for the linear bilevel programming problem. Optimization **18**, 373–385 (1987). doi:10.1080/02331938708843247
20. Dempe, S.: A necessary and sufficient optimality for bilevel programming problem. Optimization **2**, 341–354 (1992). doi:10.1080/02331939208843831
21. Dempe, S.: Annoted bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization **52**, 333–359 (2003). doi:10.1080/0233193031000149894
22. Etoa, E.J.B.: Contribution à la résolution des programmes mathématiques à deux niveaux et des programmes mathématiques avec contraintes d'équilibre, PhD Thesis, École Polytechnique de Montréal, December (2005)
23. Facchinei, F., Fischer, A., Kanzow, C.: On the accurate identification of active constraints. SIAM J. Optim. **9**, 14–32 (1998). doi:10.1137/S1052623496305882
24. Falk, J.E., Liu, J.: On bilevel programming, part I: general nonlinear case. Math. Program. **70**, 47–72 (1995)
25. Gümüs, Z.H., Floudas, C.H.: Global optimization of nonlinear bilevel programming problems. J. Glob. Optim. **20**, 1–31 (2001). doi:10.1023/A:1011268113791
26. Hansen, P., Jaumard, B., Lu, S.H.: A frame work for algorithms in globally optimal design. ASME J. Mech. Autom. Transm. **111**, 353–360 (1989)
27. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Stat. Comput. **13**, 1194–1217 (1992). doi:10.1137/0913069
28. Jeroslow, R.G.: The polynomial hierarchy and simple model for competitive analysis. Math. Program. **32**, 146–164 (1985). doi:10.1007/BF01586088
29. Kolstad, C.D., Lasdon, L.: Derivative evaluation and computational experience with large bilevel mathematical programs. J. Optim. Theory Appl. **65**, 485–499 (1990). doi:10.1007/BF00939562
30. Kornaj, J., Liptak, T.: Two-level planning. Econometrica **33**, 141–169 (1965). doi:10.2307/1911892
31. Labbé, M., Marcotte, P., Savard, G.: A bilevel model of taxation and its application to optimal highway pricing. Manage. Sci. **44**, 1608–1622 (1998). doi:10.1287/mnsc.44.12.1608
32. Lin, G.H., Fukushima, M.: Hybrid algorithms with active set identification for mathematical programs with complementarity constraints, Technical Report 2002–2008, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, Japan (2003)
33. Little, J.D.C., Murty, K.G., Sweeney, D.W., Karel, C.: An algorithm for the traveling salesman problem. J. OSRA, **11** (1963)
34. Marcotte, P., Savard, G.: Bilevel programming: a combinatorial perspective, graph theory and combinatorial Optimization. In: Avis, D., Hertz, A., Marcotte, O. (eds.) Springer, Berlin (2005)
35. Muu, L.D., Quy, N.V.: A global optimization method for solving convex quadratic bilevel programming problems. J. Glob. Optim. **26**, 199–219 (2003). doi:10.1023/A:1023047900333
36. Rockafellar, R.T.: Convex analysis. Princeton University Press, Princeton (1970)
37. Ross, S.A.: The economic theory of agency: the principal's problem. AER **63**, 134–139 (1973)
38. Savard, G., Gauvin, J.: The steepest descent direction for the nonlinear bilevel programming problem. Oper. Res. Lett. **1**, 265–272 (1994). doi:10.1016/0167-6377(94)90086-8
39. Savard, G.: Contribution à la programmation mathématique à deux niveaux, PhD Thesis, École Polytechnique de Montréal (1989)
40. Still, G.: Linear bilevel problems: genericity results and efficient method for computing local minima. Math. Methods Oper. Res. **5**, 383–400 (2002). doi:10.1007/s001860200189
41. Vicente, L.N., Calamai, P.H.: Bilevel and multilevel programming, a bibliography review. J. Glob. Optim. **5**(3), 291–306 (1994). doi:10.1007/BF01096458

42. Vicente, L.N., Savard, G., Jùdice, J.: Descent approach for quadratic bilevel programming. J. Optim. Theory Appl. **81**, 379–399 (1994). doi:10.1007/BF02191670
43. Wang, S., Lootsma, F.A.: A hierarchical optimization model of resource allocation. Optimization **28**, 351–365 (1994). doi:10.1080/02331939408843928
44. Wang, G.M., Wan, Z.P., Wang, X.J.: Genetic algorithm for solving convex quadratic bilevel programming problems, working paper. AMS, MOS subject classifications: 90C30 (2003)
45. Wilde, D.: Monotonicity and dominance in optimal hydraulic cylinder design. J. Eng. Ind. **97**(4), 13–26 (1975)