Optim Eng (2018) 19:187–211
https://doi.org/10.1007/s11081-017-9369-y

Efficiently solving linear bilevel programming problems using off-the-shelf optimization software

S. Pineda[1] · H. Bylling[2] · J. M. Morales[3]

Received: 6 April 2017 / Revised: 14 July 2017 / Accepted: 16 October 2017 /
Published online: 4 November 2017

Abstract Many optimization models in engineering are formulated as bilevel problems. Bilevel optimization problems are mathematical programs where a subset of variables is constrained to be an optimal solution of another mathematical program. Due to the lack of optimization software that can directly handle and solve bilevel problems, most existing solution methods reformulate the bilevel problem as a mathematical program with complementarity conditions (MPCC) by replacing the lower-level problem with its necessary and sufficient optimality conditions. MPCCs are single-level non-convex optimization problems that do not satisfy the standard constraint qualifications and therefore, nonlinear solvers may fail to provide even local optimal solutions. In this paper we propose a method that first solves iteratively a set of regularized MPCCs using an off-the-shelf nonlinear solver to find a local optimal solution. Local optimal information is then used to reduce the computational burden of solving the Fortuny-Amat reformulation of the MPCC to global optimality using off-the-shelf mixed-integer solvers. This method is tested using a wide range of randomly generated examples. The results show that our method outperforms existing general-purpose methods in terms of computational burden and global optimality.

Keywords Bilevel programming · Mathematical programming with complementarity conditions · Nonlinear programming · Mixed-integer programming · Optimization solvers

✉ S. Pineda
spinedamorente@gmail.com

[1] Department of Electrical Engineering, University of Malaga, Málaga, Spain

[2] Department of Mathematical Sciences, University of Copenhagen, Copenhagen, Denmark

[3] Department of Applied Mathematics, University of Malaga, Málaga, Spain

1 Introduction

Decentralized environments are characterized by multiple decisions makers with divergent objectives that interact with each other in a hierarchical organization. In the simplest case with only two decision makers, one player, called the *leader*, makes her decisions first and then the other player, called the *follower*, determines the optimal reaction to the leader's decisions. This non-cooperative sequential game is known as a *Stackelberg game* and was first investigated in Von Stackelberg (1952). A Stackelberg game can be mathematically formulated as a bilevel problem (BLP) as follows (Bard 1998; Dempe 2002):

$$\min_{x} \quad F(x,y) \tag{1a}$$

$$\text{s.t.} \quad G_i(x,y) \geq 0, \quad \forall i \tag{1b}$$

$$\min_{y} \quad f(x,y) \tag{1c}$$

$$\text{s.t.} \quad g_j(x,y) \geq 0, \quad \forall j \tag{1d}$$

where $F(x, y)$ and $f(x, y)$ are, respectively, the leader's and follower's objective functions, and $G_i(x,y)$ and $g_j(x,y)$ are the leader's and follower's constraint functions, respectively. Even if $F(x, y)$, $f(x, y)$, $G_i(x,y)$ and $g_j(x,y)$ are all linear functions, solving bilevel problem (1) is a very challenging task because its feasible region is non-convex in most cases. Furthermore, the BLP is proven to be NP-hard (Jeroslow 1985; Bard 1991) and therefore the solution methods to solve BLP are computationally intensive. A review of the different solution approaches to solve the bilevel problem (1) can be found in Dempe (2003) and Colson et al. (2005, 2007).

From a practical point of view, methods to solve linear bilevel problems can be divided into two main categories. The first category includes those methods that make use of dedicated solution algorithms to solve bilevel problems (Bialas and Karwan 1984; Shi et al. 2005b; Calvete et al. 2008; Li and Fang 2012; Sinha et al. 2013; Jiang et al. 2013; Bard and Falk 1982; Bard and Moore 1990; Hansen et al. 1992; Shi et al. 2006). While these methods are usually efficient and ensure global optimality, they involve substantial additional and ad-hoc coding work to be implemented in commercially available off-the-shelf optimization software such as CPLEX (The ILOG CPLEX 2015). The second category includes the methods that can be implemented in or in combination with general purpose optimization software without any further ado (Fortuny-Amat and McCarl 1981; Ruiz and Conejo 2009; Gabriel and Leuthold 2010; Siddiqui and Gabriel 2012; Scholtes 2001; Ralph and Wright 2004; White and Anandalingam 1993; Hu and Ralph 2004; Lv et al. 2007; Fletcher and Leyffer 2002, 2004). Although these methods are sometimes preferred due to their straightforward implementation, they may involve a high computational burden or only guarantee local optimality. The method proposed in this paper belongs to this second group and is shown to outperform existing methods within its category in terms of computational efficiency and global optimality.

An important property of a linear bilevel problem (LBLP) with a bounded constraint region is that its solution set contains at least one extreme point of such a constraint region (Bialas and Karwan 1984). Therefore, the first dedicated methods to solve LBLP were based on vertex enumeration. For instance, the Kth best method that computes global solutions of LBLP by enumerating the extreme points of the polyhedral constraint region is introduced in Bialas and Karwan (1984) and Candler and Townsley (1982). Shi et al. (2005b) propose an extended Kth best approach when the upper-level constraint functions are of an arbitrary linear form. Although quite robust, the Kth best method is computationally costly, especially for large-size problems.

If the lower-level problem (1c)–(1d) is convex and satisfies some constraint qualification, problem (1) can be reformulated as a one-level optimization problem by replacing the lower-level problem with its KKT optimality conditions as follows (Dempe and Zemkoho 2012; Dempe et al. 2015):

$$\min_{x,y,\lambda_j} \quad F(x,y) \tag{2a}$$

$$\text{s.t.} \quad G_i(x,y) \geq 0, \quad \forall i \tag{2b}$$

$$g_j(x,y) \geq 0, \quad \forall j \tag{2c}$$

$$\nabla_y f(x,y) - \sum_j \lambda_j \nabla_y g_j(x,y) = 0 \tag{2d}$$

$$\lambda_j \geq 0, \quad \forall j \tag{2e}$$

$$\lambda_j \cdot g_j(x,y) = 0, \quad \forall j \tag{2f}$$

where λ_j denotes the dual variable corresponding to each lower-level constraint (1d). Although (2) is the most commonly used approach, there exist alternative single-level reformulations of bilevel problems. Also under convexity assumptions, a bilevel problem (BLP) can be replaced by its primal KKT reformulation that does not need additional variables λ_j but requires determining the normal cone to the follower's feasible region for each value of x. Alternatively, problem (1) can be recast as a nonsmooth and nonconvex single-level optimization problem using an optimal value function of the lower-level problem. Further details about these two approaches can be found in Dempe et al. (2015).

Problem (2) is a mathematical program with complementarity conditions (MPCC) (Outrata 2000). As proven in Dempe and Dutta (2010), if (x^*, y^*, λ_j^*) is a global optimal solution of problem (2), and the lower-level problem (1c)–(1d) is convex and satisfies some constraint qualification, then (x^*, y^*) is a global optimal solution of the original bilevel problem (1). Besides, if the lower-level problem is convex and Slater's condition holds, the local optimal solutions of problem (2) are also local optimal solutions of the bilevel problem (1) (Dempe and Dutta 2010).

Note that these conditions are always satisfied for the linear bilevel problems analyzed in this paper.

Note also that although constraint (2d) remains affine provided that f and g_j are linear or convex quadratic functions, problem (2) is non-convex due to the nonlinear complementarity conditions (2f). Moreover, as shown in Scheel and Scholtes (2000), problem (2) violates the Mangasarian-Fromovitz constraint qualification at every feasible point of the problem, which makes both the formulation of (necessary and sufficient) optimality conditions and the computation of global optimal solutions difficult.

Taking the single-level optimization problem (2) as a starting point, we can also find methods within the two categories previously discussed. For example, some dedicated methods take advantage of the intrinsically combinatorial structure of problem (2) to handle the complementarity constraints using ad-hoc branch-and-bound algorithms as first proposed in Bard and Falk (1982) and further developed in Bard and Moore (1990), Hansen et al. (1992), Shi et al. (2006). In these methods, the root node solves the problem obtained by removing the complementarity conditions (2f). If at a given node one complementarity constraint j' is not satisfied, two new nodes are added to the tree, one with the additional constraint $\lambda_{j'} = 0$ and the other with the constraint $g_{j'}(x, y) = 0$. By repeating this process and solving the linear problems obtained after each branching, all possible combinations that satisfy the complementarity conditions are evaluated and therefore, obtaining the global optimal solution is guaranteed.

Alternatively, Fortuny-Amat and McCarl (1981) propose a mixed-integer reformulation of problem (2) that can be directly implemented using off-the-shelf optimization software. This approach replaces the complementarity conditions (2f) with the following set of disjunctive constraints:

$$\lambda_j \leq z_j M, \quad \forall j \tag{3a}$$

$$g_j(x, y) \leq (1 - z_j)M, \quad \forall j \tag{3b}$$

where z_j is a binary variable and M a sufficiently large positive number. Note that for the linear case, problem (2) is reformulated as a mixed-integer linear programming problem that can be solved to optimality using conventional branch-and-bound or branch-and-cut techniques available in most mixed-integer optimization solvers. For this reason, this approach is the most commonly used to solve LBLP in practical applications. Notwithstanding this, the equivalence between problem (2) and its mixed-integer reformulation using (3) is only true provided that the value of M is large enough so that constraints (3a) and (3b) are only binding for $z_j = 0$ and $z_j = 1$, respectively. On the other hand, choosing a too large constant M may create numerical instabilities due to scalability issues. Hence, finding suitable values of M a priori is a delicate task. Although some ad-hoc methods have been proposed to solve this issue for particular applications of bilevel programming (Ruiz and Conejo 2009; Gabriel and Leuthold 2010), tuning the large constants M for general LBLP requires a nontrivial trial-and-error process. In fact, many authors (Motto et al. 2005; Hasan et al. 2008; Garces et al. 2009; Baringo and Conejo 2011; Wogrin et al. 2011; Pozo and Contreras 2011; Kazempour et al. 2011, 2012; Ruiz et al. 2012; Kazempour and Conejo 2012; Baringo and Conejo 2012, 2013; Jenabi 2013; Wogrin et al. 2013; Pozo et al. 2013; Zugno et al. 2013; Pisciella et al. 2014; Baringo and Conejo 2014; Lorenczik et al. 2014; Maurovich-Horvat et al. 2014; Morales et al. 2014; Ruiz and Conejo 2014; Valinejad and Barforoushi 2015; Moiseeva 2015) solve either MPEC or bilevel problems using the Fortuny-Amat reformulation approach, but without explaining how the large constants M are determined.

Another approach to solve (2) as a mixed-integer problem consists in reformulating the complementarity conditions using Special Order Sets (SOS) (Siddiqui and Gabriel 2012). Special Order Sets of type 1 (SOS1) are sets of variables in which at most one member can be strictly positive. Therefore, constraint (2f) can be equivalently expressed as:

$$s_j(1) = \lambda_j, \quad \forall j \tag{4a}$$

$$s_j(2) = g_j(x, y), \quad \forall j \tag{4b}$$

where the pair $\{s_j(1), s_j(2)\}$ is defined as an SOS1 for each j. The main advantages of this approach are that no large constant is required and that it can be also directly solved using commercially available mixed-integer optimization solvers. On the other hand, this method can also be computationally very expensive, especially for large models, as shown in Sect. 5.

As previously mentioned, optimization problem (2) is not regular since it fails to comply with the standard Mangasarian-Fromovitz constraint qualification and therefore, off-the-shelf nonlinear solvers may even fail to find a local optimal solution. For instance, if the nonlinear solver is based on a sequential quadratic programming algorithm (SQP), the quadratic programming subproblems may be degenerate because the original problem (2) has no strictly feasible points (Fletcher and Leyffer 2004). To overcome this issue, a regularization approach to solve mathematical programs with complementarity conditions (MPCC) was first introduced in Scholtes (2001) and further investigated in Ralph and Wright (2004). This method replaces each complementarity constraint (2f) by:

$$\lambda_j \cdot g_j(x, y) \le t, \quad \forall j \tag{5}$$

where t is a small non-negative scalar. In doing so, problem (2) becomes a parametrized nonlinear optimization problem that typically satisfies constraint qualifications and is thus easier to solve. Alternatively, all inequalities in (5) can be replaced by a single inequality as follows:

$$\sum_j \lambda_j \cdot g_j(x, y) \le t \tag{6}$$

Using (6) instead of (5) may improve the numerical behavior of nonlinear solvers since the number of inequality constraints is reduced. In either case, Scholtes (2001) provides the necessary conditions under which a local minimizer of the original problem (2) is a limit point of a curve of stationary points of the parametrized nonlinear problem as t tends to 0. Although this regularization method significantly reduces the computational burden of solving problem (2), using existing nonlinear optimization techniques such as SQP only guarantees local optimal solutions of problem (2), which are not necesarily local optimal solutions of the generic bilevel problem (1) (Dempe and Dutta 2010). Another advantage of this method is that it can also be directly implemented using off-the-shelf nonlinear optimization software since it just consists of iteratively solving a set of nonlinear problems.

Some other works investigate the solution of linear bilevel problems using a penalty function. For example, the procedure proposed in White and Anandalingam (1993) disregards the complementarity conditions (2f) and adds a term to the upper-level objective function that penalizes the duality gap of the lower-level optimization problem. In the linear case, White and Anandalingam (1993) demonstrate that the proposed procedure guarantees global optimality. Further studies of penalty methods for solving LBLP can be found in Hu and Ralph (2004) and Lv et al. (2007).

Finally, some heuristic methods have been suggested in the literature to solve linear bilevel problems. For example, the procedure proposed in Hejazi et al. (2002) applies genetic algorithms to solve the KKT reformulation of the LBLP. Similarly, Calvete et al. (2008) present a solution algorithm that combines extreme point enumeration techniques with genetic search methods. Li and Fang (2012) and Sinha et al. (2013) introduce evolutionary algorithms to solve bilevel problems. The approach proposed in Jiang et al. (2013) applies particle swarm optimization to a smooth version of the KKT reformulation of the bilevel problem. Given the complexity of these approaches and the amount of extra code required to be implemented in standard optimization software, they fall into the category of dedicated methods.

In summary, dedicated methods such as the Kth best method, ad-hoc branch-and-cut algorithms, or heuristic approaches can be efficient to provide the global optimal solutions of linear bilevel problems. However, they cannot be directly coded using off-the-shelf optimization software. Among general purpose methods that can be directly implemented using optimization solvers, the mixed-integer reformulations (Fortuny-Amat or SOS1 approaches) determine global optimal solutions at the expense of drastically increasing the computational burden. On the other hand, regularization approaches to solve the KKT reformulation of the LBLP using off-the-shelf nonlinear optimization software prove to be fast but cannot guarantee neither global nor local optimality of the original bilevel problem (Dempe and Dutta 2010). In this paper we propose a new procedure that combines these two approaches to efficiently solve linear bilevel programming problems and that can be directly implemented using off-the-shelf optimization software. The contribution of this paper is thus twofold:

- We provide a computationally efficient method to solve linear bilevel programming problems using available optimization software. The proposed method uses first a regularization approach to efficiently determine a local optimal solution of the KKT reformulation of the LBLP using a nonlinear optimization solver. Then, this local optimal solution is used to significantly reduce the computational burden of solving the mixed-integer linear reformulation proposed in Fortuny-Amat and McCarl (1981) using a conventional mixed-integer optimization solver as follows. First, by setting appropriate values of the large constant M in (3) according to the order of magnitude of the primal and dual variables. Second, by providing initial values to the binary variables based on which term of the complementarity conditions is equal to 0 at the local optimal solution.

- We test the performance of the proposed method through a set of comprehensive computational studies based on a large family of randomly generated examples of different sizes. The proposed method is compared in terms of computational burden and global optimality against other general purpose methods to solve LBLP. The obtained results show that the proposed approach is an efficient generic algorithm to solve lineal bilevel problems in practice.

The remainder of this paper is organized as follows. Section 2 formally presents the generic formulation of the linear bilevel problem under study together with some important definitions and properties. Section 3 introduces the KKT reformulation of the LBLP and explains in detail how both existing algorithms and the proposed algorithm can be used to solve it. Section 4 elaborates on how the test examples are randomly generated and sets the basis for comparing the results provided by the different methods. The main computational results are presented and discussed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Linear bilevel programming problem

Given the complexity of bilevel programming problems, in this paper we restrict ourselves to the simplest case in which the functions $F(x, y)$, $f(x, y)$, $G_i(x, y)$ and $g_j(x, y)$ are all linear. Hence, a linear bilevel problem (LBLP) is generally formulated as follows (Bard 1998; Zhang et al. 2015):

$$\min_{x} \quad F(x,y) = c_1 x + d_1 y \tag{7a}$$

$$\text{s.t.} \quad A_1 x + B_1 y \leq b_1 \tag{7b}$$

$$\min_{y} \quad f(x,y) = c_2 x + d_2 y \tag{7c}$$

$$\text{s.t.} \quad A_2 x + B_2 y \leq b_2 \tag{7d}$$

where $c_1, c_2, d_1, d_2, b_1, b_2, A_1, B_1, A_2, B_2$ are vectors and matrices of appropriate dimensions.

The induced region (IR) of the LBLP is the set of feasible points of the leader and rational responses from the follower (Bard 1998). With this notation, the LBLP can be equivalently recast as the following one-level optimization problem:

$$\min_{x,y} \quad F(x,y) \tag{8a}$$

$$\text{s.t.} \quad (x,y) \in IR \tag{8b}$$

If an explicit formulation of the IR as a polyhedron were possible and available, the solution to (7) could be obtained by solving problem (8) as a one-level linear programming problem using, for example, the simplex method. However, even for simple instances of LBLP, the IR cannot be formulated as a polyhedron, which makes (8) a very hard problem to solve (Jeroslow 1985; Bard 1991; Ben-Ayed and Blair 1990). As proven in Bard (1998), if the follower's rational reaction set is bounded and the constraint region is non-empty and bounded, then an optimal solution to the LBLP (8) exists. Therefore, unless otherwise specified, these assumptions apply to all problems presented in this paper.

One issue worth discussing is the existence of upper-level constraints that include both upper-level and lower-level variables. The validity of such joint upper-level constraints is beyond the choice of the leader and can only be validated after the follower's optimal choice is determined (Dempe et al. 2015). Mathematically, joint upper-level constraints can lead to disconnected or empty IR (Colson et al. 2005), which further complicates the solution of the linear bilevel problem as illustrated in Shi et al. (2005c). Extended approaches to apply existing solution algorithms to LBLP with upper-level constraints of arbitrary form can be found in Shi et al. (2005a, 2006), and Mersha and Dempe (2006). However, for the sake of simplicity, this paper only considers LBLP with upper-level constraints that do not include lower-level variables, i.e., $B_1 = 0$ in (7) unless otherwise stated.

Another important aspect of LBLP is the existence of multiple optimal solutions to the lower-level problem. Under such circumstances, the leader's choice has to be determined without exactly knowing the reaction of the follower, who can choose among a set of decisions that lead to the same value of her objective function. To overcome this indeterminacy, there are two main possibilities, namely, the optimistic and the pessimistic solution (Dempe 2002; Colson et al. 2005, 2007). The leader can assume that the follower can be influenced to select the solution that involves a higher leader's objective function. This is known as the optimistic solution of a LBLP. Conversely, the pessimistic solution considers that the leader has no possibility to alter the behavior of the follower, who can choose the worst solution with respect to the leader's objective function. In this paper we focus on the optimistic formulation since it is simpler, is the usual approach and has been more deeply investigated in the technical literature (Dempe et al. 2007; Strekalovsky et al. 2010a; Dempe and Franke 2014). For further details about the pessimistic formulation of a linear bilevel problem, the interested reader is referred to Dempe et al. (2014) and the references therein.

3 Solution methods

The original linear bilevel problem (7a)–(7d) can be reformulated as the single-level optimization problem (9a)–(9f) by replacing its lower-level optimization problem with its KKT optimality conditions. Note that model (9a)–(9f) is a nonlinear optimization problem because of the products $\lambda \cdot x$ and $\lambda \cdot y$ in equation (9f), where λ denotes a vector with the dual variables of the lower-level constraint (7d). All the methods presented in this section aim at solving this single-level nonlinear optimization model using different approaches. The following subsections provide the detailed steps of the solution algorithms compared in this paper.

$$\min_{x,y,\lambda} \quad F(x,y) = c_1 x + d_1 y \tag{9a}$$

$$\text{s.t.} \quad A_1 x + B_1 y \leq b_1 \tag{9b}$$

$$d_2 + \lambda B_2 = 0 \tag{9c}$$

$$b_2 - A_2 x - B_2 y \geq 0 \tag{9d}$$

$$\lambda \geq 0 \tag{9e}$$

$$\lambda(b_2 - A_2 x - B_2 y) = 0 \tag{9f}$$

3.1 Branch-and-bound approach

This method solves the single-level reformulation of the LBLP (9) using a binary tree. The method starts by solving the relaxed linear problem (9a)–(9e). If all complementarity conditions are satisfied, then this is the optimal solution to (9). Otherwise, the tree is branched in one of the violated complementarity constrains j' so that two nodes are added to the tree. A linear optimization problem is defined for each new node by adding the constraint $\lambda_{j'} = 0$ or $(A_2 x + B_2 y - b_2)_{j'} = 0$ to the problem corresponding to the predecessor node. This procedure continues until the subproblems corresponding to all ending nodes are infeasible or have an objective value larger than the current upper bound (Bard and Moore 1990).

Note that this approach only involves the solution of linear programming problems and therefore, convergence to global optimality is guaranteed. For this reason, and despite the fact that this approach belongs to the category of dedicated solution methods, the solution provided by the branch-and-bound is used to check the performance of the other general purpose methods investigated in this paper. On the other hand, applying this algorithm to solve LBLP may easily become computationally expensive, even for low size problems.

3.2 Mixed-integer approach

Given the combinatorial nature of the complementarity constraints (9f), some solution methods propose to reformulate problem (9) as a mixed-integer programming problem and directly use off-the-shelf integer optimization software. The idea of Fortuny-Amat is to rewrite these complementarity conditions using disjunctive constraints that require the use of binary variables and large enough constants (Fortuny-Amat and McCarl 1981). Problem (9) is thus reformulated as follows:

$$\min_{x,y,\lambda,u} \quad F(x,y) = c_1x + d_1y \tag{10a}$$

$$\text{s.t.} \quad A_1x + B_1y \leq b_1 \tag{10b}$$

$$d_2 + \lambda B_2 = 0 \tag{10c}$$

$$b_2 - A_2x - B_2y \geq 0 \tag{10d}$$

$$\lambda \geq 0 \tag{10e}$$

$$b_2 - A_2x - B_2y \leq (1-u)M_1 \tag{10f}$$

$$\lambda \leq uM_2 \tag{10g}$$

$$u \in \{0,1\} \tag{10h}$$

where u is a vector of binary variables of appropriate size and M_1, M_2 are large enough scalars. Note that formulation (10) is obtained from formulation (9) by simply replacing the nonlinear constraint (9f) with constraints (10f), (10g) and (10h). Problem (10) is a mixed-integer linear programming problem that can be solved using conventional branch-and-bound algorithms such as the one used by CPLEX (The ILOG CPLEX 2015).

Alternatively, SOS1 variables can be used to impose the complementarity conditions by replacing equations (10f)–(10h) with (Siddiqui and Gabriel 2012):

$$s_j(1) = (b_2 - A_2x - B_2y)_j, \quad \forall j \tag{11a}$$

$$s_j(2) = \lambda_j, \quad \forall j \tag{11b}$$

where the pair $\{s_j(1), s_j(2)\}$ is declared as SOS1 for each j. Problem (11) can also be solved using mixed-integer linear solution methods such as those in commercially available optimization software.

If the values of M_1, M_2 are properly set, both (10) and (11) can be solved to global optimality using existing mixed-integer optimization solvers. However, similarly to the branch-and-bound approach, the computational burden of solving these models dramatically increases with the size of the bilevel problem.

3.3 Regularization approach

As shown in Scheel and Scholtes (2000), all feasible points of (9) are nonregular, which implies that most existing nonlinear optimization solvers may fail even to find a local optimal solution. If the regularization approach proposed in Scholtes (2001) and Ralph and Wright (2004) is applied to problem (9), we obtain the following formulation:

$$\min_{x,y,\lambda} \quad F(x,y) = c_1x + d_1y \tag{12a}$$

$$\text{s.t.} \quad A_1x + B_1y \leq b_1 \tag{12b}$$

$$d_2 + \lambda B_2 = 0 \tag{12c}$$

$$b_2 - A_2x - B_2y \geq 0 \tag{12d}$$

$$\lambda \geq 0 \tag{12e}$$

$$\lambda(b_2 - A_2x - B_2y) \leq t \tag{12f}$$

where t is a small non-negative scalar. Formulation (12) is derived from formulation (9) by replacing the nonlinear equality constraint (9f) with the nonlinear inequality constraint (12f). Notice that both models are, therefore, equivalent for t tending to 0. This approach consists in iteratively solving a set of nonlinear regular optimization problems. In each iteration, the value of t is reduced. The local optimal solution in one iteration is used as the initial starting point for the following iteration. While being relatively fast and presenting strong theoretical and empirical convergence properties (Scholtes 2001), this regularization approach is only guaranteed to provide local optimal solutions of the MPCC, which are also local optimal solutions of the original LBLP (Dempe and Dutta 2010).

3.4 Penalty approach

Another method to solve the nonregular problem (9) consists in penalizing the complementarity constraints in the objective function as follows (White and Anandalingam 1993; Hu and Ralph 2004; Lv et al. 2007):

$$\min_{x,y,\lambda} \quad F(x,y) = c_1x + d_1y + \frac{1}{t}\sum_j \lambda_j(b_2 - A_2x - B_2y)_j \tag{13a}$$

$$\text{s.t.} \quad A_1x + B_1y \leq b_1 \tag{13b}$$

$$d_2 + \lambda B_2 = 0 \tag{13c}$$

$$b_2 - A_2x - B_2y \geq 0 \tag{13d}$$

$$\lambda \geq 0 \tag{13e}$$

where t is also a non-negative scalar that is iteratively decreased to make the complementarity conditions tend to 0. The initial value of t is set to a large value and is reduced by a factor of $\rho > 1$ in each iteration. As in the regularization method, a nonlinear optimization problem has to be solved at each iteration.

3.5 Proposed approach

The purpose of the proposed solution method is to combine the mixed-integer and the regularization approaches presented above in order to obtain a global optimal solution while reducing the computational burden. The main issue with the regularization approach is that, albeit fast, it only ensures local optimal solutions for the MPCC reformulation. On the other hand, formulation (10) can be solved to global optimality. However, finding appropriate values of the large constants M_1, M_2 that allow solving (10) in a reasonable time is usually a difficult task. In fact, very low or very high values of M_1, M_2 may lead to infeasible, suboptimal and numerically unstable problems, respectively. The proposed approach uses the local optimal solution for the MPCC reformulation provided by the regularization method to soundly determine values of these large constants that allow us to find the optimal global solution of (10) at a low computational cost.

The proposed approach relies on nonlinear optimization solvers whose performance is significantly improved if a feasible initial point is provided. This initial feasible point is calculated by sequentially solving two linear programming problems. The first linear optimization problem is obtained by removing the nonlinear complementarity condition from model (9) to obtain a pair (x, y) that satisfies all upper- and lower-level constraints, but that is not optimal for the lower-level problem. We then fix the values of x and solve the lower-level optimization problem alone, which is also a linear programming problem, to find values of y that are also optimal for the lower-level problem. Therefore, by sequentially solving these two linear programming problems, we obtain a feasible point (x, y) that satisfies all the constraints (9b)–(9f).

The proposed approach requires the use of the following parameters:

k	Iteration counter.
t	Small non-negative scalar representing the slackness of the complementarity conditions.
ρ	Non-negative scalar used to update the value t.
$\mathcal{M}$	Non-negative scaling parameter used to compute the large enough constants.

The steps of the proposed procedure are the following:

- *Step 0 (Initialization)* Select parameters $t > 0$, $\rho > 1$, $\mathcal{M} > 1$ and the number of iterations K. Set $k \leftarrow 0$ and go to Step 1.
- *Step 1 (Feasible point)* Solve the linear programming problem (9a)–(9e) and denote the obtained leader's variables as x_0. Solve the lower-level linear programming problem (7c)–(7d) in which upper-level variables are fixed at x_0. Denote the optimal values of the primal and dual variables as y_0 and λ_0, respectively. Go to Step 2.

- *Step 2 (Iteration)* Set $k \leftarrow k+1$. Solve problem (12) taking $(x_{k-1}, y_{k-1}, \lambda_{k-1})$ as an initial point. Denote its solution as (x_k, y_k, λ_k). If $k < K$, then $t \leftarrow t/\rho$ and go to Step 2. Otherwise, go to Step 3.
- *Step 3 (Tuning)* Set $M_1 \leftarrow \mathcal{M} \max_j \{(b_2 - A_2 x_k - B_2 y_k)_j\}$ and $M_2 \leftarrow \mathcal{M} \max_j \{(\lambda_k)_j\}$. Go to Step 4.
- *Step 4 (Warming)* Set initial values of binary variables u as follows. If $(b_2 - A_2 x_k - B_2 y_k)_j > 0$, then $u_j = 0$. If $\lambda_j > 0$, then $u_j = 1$. Go to Step 5.
- *Step 5 (Solution)* Solve the mixed-integer linear problem (10) using the values of M_1, M_2 determined in Step 3 and the initial values of the binary variables computed in Step 4. Declare its solution (x^*, y^*, λ^*) as the optimal solution.

The core of the proposed approach relies on Steps 3 and 4, in which the local optimal solution provided by the regularization method is used to tune the large constants M_1 and M_2 and to compute initial values for the binary variables u, respectively. Let us explain first the reasoning behind Step 3. Note that the mixed-integer approach (10) is only valid provided that constraints (10f) and (10g) are binding if and only if $u = 1$ or $u = 0$, respectively. This is only true if the following two conditions hold: M_1 is larger than $b_2 - A_2 x - B_2 y$ for any feasible pair (x, y) and M_2 is larger than any feasible value of the dual variable λ. Even though the solution obtained in Step 2 using regularization is just locally optimal, we assume that the maximum value of $b_2 - A_2 x - B_2 y$ over all lower level constraints at the local optimal solution is a good proxy of M_1. Similarly, the maximum value of the lower-level dual variable λ_j over all constraints at the local optimal solution is also a good estimation of the large constant M_2. If large constants M_1 and M_2 are tuned based exclusively on the locally optimal solution computed in Step 2, two issues may arise. In some cases, the globally optimal solution to the original linear bilevel problem may be actually infeasible due to the bad adjustment of the large constants M_1 and M_2. For other cases, the optimal solution (10) may not be globally optimal for the original optimization problem due to the overly-constrained feasible region. To avoid these two issues, these values are multiplied by the scaling parameter $\mathcal{M} > 1$, which needs to be adjusted by trial and error bearing in mind the following trade-off: the larger the value of $\mathcal{M}$, the lower the risk that the global optimal solution becomes infeasible or suboptimal, but the higher the computational time required to solve the problem due to numerical instabilities. The intuition behind Step 4 is the following. Note that the values of the variables u obtained in Step 2 provide information about which term of each complementarity condition (9f) is equal to 0 at the locally optimal solution. Assuming that the globally optimal solution is not "too different" from the locally optimal solution obtained by the regularization approach, the terms of the complementarity conditions equal to 0 are expected to coincide for most of these constraints.

Providing initial values for the binary variables u and tuning the large constants M_1, M_2 only seeks to improve the computational performance of the mixed-integer solver without jeopardizing the optimality of the solution that the solver eventually returns. How much the computational burden of solving (10) will be reduced by taking advantage of the locally optimal information provided by (12) cannot be exactly established a priori with full guarantees. To provide some guidance on this issue, however, we conduct and present an exhaustive numerical analysis in Sects. 5.1–5.3, in which a large set of linear bilevel problems of different size, sparsity and scale are solved.

Finally, note that the proposed solution algorithm can be directly implemented using off-the-shelf optimization software since it only involves solving:

- Two linear programming problems using a linear optimization solver to find a point in the induced region.
- A family of regularized nonlinear optimization problems using a nonlinear optimization solver to find a local optimal solution.
- A mixed-integer linear programming problem with appropriate large constants and initial values of the binary variables using a mixed-integer optimization solver to find the global optimal solution.

4 Test and comparison

In this section, we first describe how test bilevel problems are randomly generated and then explain how the results provided by the different solution methods are compared.

As previously discussed, the test examples considered in this paper do not include any joint upper-level constraints and therefore, matrix B_1 is empty. In order to avoid unbounded test problems, it is also imposed that both the coefficients of the upper-level and lower-level objective functions (c_1, d_1, c_2, d_2) and the variables involved (x, y) must be non-negative. For the sake of generality, the test bilevel problems include two sets of lower-level constraints: the first set of constraints involves upper- and lower-level variables, while the second only comprises lower-level variables. According to these assumptions, the vectors and matrices of bilevel problem (7) are generated as follows:

$$c_1 = |\mathcal{N}(1,n)| \quad d_1 = |\mathcal{N}(1,m)| \quad A_1 = \begin{pmatrix} \mathcal{N}(p,n) \\ -I \end{pmatrix} \quad B_1 = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad b_1 = \begin{pmatrix} \mathcal{N}(p,1) \\ \mathbf{0} \end{pmatrix}$$

$$c_2 = |\mathcal{N}(1,n)| \quad d_2 = |\mathcal{N}(1,m)| \quad A_2 = \begin{pmatrix} \mathcal{N}(q,n) \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad B_2 = \begin{pmatrix} \mathcal{N}(q,m) \\ \mathcal{N}(r,m) \\ -I \end{pmatrix} \quad b_2 = \begin{pmatrix} \mathcal{N}(q,1) \\ \mathcal{N}(r,1) \\ \mathbf{0} \end{pmatrix}$$

where $\mathcal{N}(i,j)$ denotes an $i \times j$ matrix in which each element is randomly generated according to a standard normal distribution with mean and variance equal to 0 and 1, respectively. As follows from these definitions, n and m are the number of upper- and lower-level variables, respectively. Furthermore, each random problem includes p upper-level constraints, q lower-level joint constraints and r lower-level constraints not involving upper-level variables.

Given one random problem, let l be an index for the different solution approaches presented in this paper. The optimal solution, objective function value and solver status provided by solution approach l are denoted as (x_l^*, y_l^*), z_l^* and s_l, respectively, and are computed as follows:

- *Step (1)* The bilevel problem is solved using solution method l and the optimal upper-level variables are denoted as x_l^*. If no solution is provided, set s_l to 0 and stop. Otherwise, go to Step (2).
- *Step (2)* The upper-level variables are fixed to x_l^* and the lower-level problem is solved again using linear programming to obtain the lower-level optimal variables y_l^*. If the lower-level is infeasible, set s_l to 0 and stop. Otherwise, go to Step (3).
- *Step (3)* Set s_l to 1 and compute the value of the objective function z_l^* as $c_1 x_l^* + d_1 y_l^*$.

This procedure to compare the different methods is particularly relevant for those formulations that include products of binary variables and large numbers. Note that some mixed-integer solvers may round down this product and thus yield optimal values for the binary variables different from 0 and 1 due to numerical instabilities. If this happens, the objective function obtained by these methods may be lower than the optimal one since complementarity conditions do not hold. However, if we fix the upper-level variables and then solve the lower-level problem as described above, this issue is avoided and the values of the upper-level objective function provided by different solution methods can be fairly compared. For each random problem, the true optimal solution $\hat{z}$ is defined as:

$$\hat{z} = \min\{z_l^* : s_l = 1\}$$

In most examples, $\hat{z}$ will be equal to the solution provided by the branch-and-bound and SOS1 methods, since these approaches guarantee global optimality. If these methods do not provide a solution due to time restrictions, then $\hat{z}$ will be the minimum objective function among the methods that deliver a solution. The optimality gap for the solution given by method l is thus computed as:

$$g_l = 100 \times \frac{z_l^* - \hat{z}}{\hat{z}}$$

which is only defined for those methods with $s_l = 1$.

In this paper we compare the following methods to solve linear bilevel problems:

- Branch and bound method (B&B).
- Mixed-integer solution method with SOS1 variables (SOS1).
- Mixed-integer solution method in which disjunctive constraints are modeled as proposed by Fortuny-Amat and McCarl (1981). The following 11 values for the large constants are used: 5, 10, 20, 50, 100, 200, 500, 1000, 5000, 10000, 100000. Each variant of this method is thus referred to as FA-5, FA-10, FA-20, etc.
- Regularization method proposed in Scholtes (2001) and Ralph and Wright (2004) (REG). The number of iterations (K) is set to 20, the initial value of t to 10^4, and ρ is equal to 10.

- Penalty approach proposed in White and Anandalingam (1993) (PEN). The number of iterations (K) is set to 20, the initial value of t to 1, and ρ is equal to 1.2.
- The proposed solution method, which is referred to as REG-FA. The regularized local optimization method is tuned as in REG. The following 3 values for the parameter $\mathcal{M}$ are used: 2, 5, 10. Each variant of this method is thus referred to as REG-FA-2, REG-FA-5 and REG-FA-10, respectively.

5 Computational results

This section compiles the main computational results of the methods presented in Sect. 3 to solve linear bilevel problems. First, the results of 300 test problems of different sizes are provided. Then, the impact of matrix sparsity on the performance of the different methods is investigated. Finally, we also analyze how bad scaling affects the obtained results.

All the results presented here have been obtained using CPLEX 12.6.0.1 and CONOPT 3.16C optimization solvers under GAMS 24.3.3. The simulations have been run in a cluster with 288 nodes. Each node consists of Two Intel Xeon Processor E5649 (2.53 GHz, 6 cores) and 24 GB of memory. The maximum time for each problem is set to 6 h. The code and data used for the simulations are available at www.github.com/salvapineda/bilevel.

5.1 Impact of size

The solution methods presented in this paper are tested on 100 small randomly generated problems, 100 medium randomly generated problems, and 100 large randomly generated problems. The matrices of these problems are generated according to the parameters provided in Table 1. Note that the number of upper- and lower-level variables is the same in all cases. Furthermore, the number of each type of constraint is equal to half the number of variables since a much higher or a much lower number of constraints may lead to infeasible or trivial problems, respectively. It is also worth mentioning that other works providing similar computational results consider randomly generated test cases with a maximum size of 150 upper- and lower-level variables (Strekalovsky et al. 2010a, b).

Table 2 provides the results for the 18 methods compared in this study for the three problem sizes. For each problem size and solution approach four numerical results are provided, namely:

Table 1 Parameters of randomly generated problems

	n	m	p	q	r
Small	50	50	25	25	25
Medium	100	100	50	50	50
Large	200	200	100	100	100

- The number of randomly generated problems solved to global optimality, that is, with zero optimality gap ($g_l = 0$). This is denoted as #opt.
- The number of randomly generated problems that are infeasible, that is, with $s_l = 0$. This is denoted as #inf.
- The average computational time (in seconds) for those randomly generated problems with valid solutions, that is, with $s_l = 1$.
- The average optimality gap (as a percentage) for those randomly generated problems with valid solutions, that is, with $s_l = 1$.

Therefore, 100-#opt-#inf is the number of non-optimal valid solutions.

Let us first analyze the results provided by the SOS1 method. Note that for small instances, this method achieves the optimal solution in 98 of the 100 cases in around 1 second, the remaining 2 cases being infeasible. For the medium instances, 90 are solved to optimality while the average solution time is increased to 1.3 h. The average gap of 0.27% is due to the fact that some problems were not solved to

Table 2 Results: impact of size

	Small (n = 50)				Medium (n = 100)				Large (n = 200)			
	#opt	#inf	Time (s)	Gap (%)	#opt	#inf	Time (s)	Gap (%)	#opt	#inf	Time (s)	Gap (%)
B&B	92	2	2761	0.66	44	0	13,556	2.89	8	0	20,211	2.54
SOS1	98	2	1	0.00	90	0	4656	0.27	27	0	17,652	2.05
FA-5	11	24	12	7.65	8	7	5385	7.15	0	20	21,415	4.57
FA-10	72	4	3	0.91	68	3	2798	0.18	44	0	16,976	0.27
FA-20	95	4	3	0.01	92	1	3914	0.05	51	0	16,981	0.32
FA-50	98	2	5	0.00	94	2	5495	0.04	53	1	16,873	0.31
FA-100	97	2	9	0.00	93	0	6678	0.02	45	1	18453	0.60
FA-200	93	2	15	0.09	81	0	7716	0.19	37	0	18,634	0.68
FA-500	85	2	36	0.14	78	2	9229	0.15	43	0	19,251	0.38
FA-1000	79	2	55	0.29	66	1	10,847	0.33	28	0	19241	0.51
FA-5000	49	2	80	2.28	26	4	7578	3.54	2	0	15,517	3.63
FA-10000	37	2	36	3.95	13	0	4361	7.71	2	0	6782	10.34
FA-100000	27	2	0	6.44	11	0	0	10.19	2	0	1	9.86
REG	61	2	0	0.55	41	0	1	0.52	30	1	14	0.29
PEN	34	2	0	0.85	10	0	17	1.03	12	0	30	0.78
REG-FA-2	94	2	2	0.02	96	0	660	0.01	82	2	10,657	0.07
REG-FA-5	98	2	2	0.00	99	1	767	0.00	71	11	10,770	0.08
REG-FA-10	98	2	5	0.00	99	0	2353	0.00	72	0	13,258	0.10

optimality after 6 h. Finally, for the large instances, the SOS1 method only achieved the optimal solution in 27 cases and the average solution time is 4.9 h. The increase in the computational time required by this method with the size of the problem is thus apparent. Like the SOS1 method, the branch-and-bound method guarantees global optimality. Note, however, that the number of optimal solutions, the average computational time and the average optimality gap are worse for the branch-and-bound method for all problem sizes. Therefore, the SOS1 method is considered in this analysis as a benchmark.

Regarding the Fortuny-Amat method, the following general observations are in order. Both for very low and very large values of the large constant M, the number of examples solved to optimality is very low although for different reasons. While small values of M lead to a high number of infeasible problems, high values of M create numerical instabilities in the solution algorithm. Note also that the value of M that results in the largest number of test problems solved to optimality is equal to 50 for the three sets of examples, being the average time equal to 5 s, 90 min and 4.5 h for small, medium and large problems, respectively. Observe that for large instances, the maximum number of optimal solutions achieved by the best Fortuny-Amat method is only 53.

Despite being very fast, the regularization method only provides the global optimal solution in a low number of cases, which decreases as the dimension of the problems increases. Note that for large problems, in only 30 examples the local optimal solution found by this method is also global optimal. Observe as well that the results provided by the penalty method are even worse than those of the regularization method in terms of global optimality, computational time and optimality gap.

For the three problem sizes, the proposed approach provides very similar results for the three values of $\mathcal{M}$ in terms of number of optimal cases, computational time, and optimality gap. This shows that selecting an appropriate value of $\mathcal{M}$ for the proposed approach is substantially less critical than choosing a high enough value of M for the Fortuny-Amat approach. Let us then focus on the results for $\mathcal{M} = 10$, for example. For small problems, REG-FA-10 also results in 98 instances solved to optimality, but with an average time higher than that of the SOS1 method. Given the low number of binary variables, optimization solvers such as CPLEX are quite efficient in solving problems of this size and that implies that the pre-calculations of the proposed method significantly increase the computational time in comparative terms. On the other hand, for medium problems, REG-FA-10 is able to find the optimal solution in 99 cases in an average time of 40 min, thus outperforming the SOS1 method (90 optimal cases, 1.3 h) and the best Fortuny-Amat method (94 optimal cases, 1.5 h). These results demonstrate, therefore, the computational efficiency of the solution method proposed in this paper. For large problems, REG-FA-10 obtains 72 optimal cases in 3.6 h, versus the 27 optimal cases and 4.9 h of the SOS1 method, and the 53 optimal cases and 4.6 h of the best Fortuny-Amat method. Notice also that the average gap corresponding to the non-optimal cases is equal to 0.10, 2.05 and 0.31% for REG-FA-10, SOS1 and FA-50, respectively.

It should be noted that the discussion above is based on comparing the proposed approach with the Fortuny-Amat method providing the best results. However, the value of M that performs best is not known in advance and can only be determined after a trial-and-error process similar to the extensive testing done in this paper, which makes our method even more advantageous than what this analysis already reveals.

5.2 Impact of sparsity

All the randomly generated matrices for the analysis of the previous subsection are full matrices. In order to investigate the performance of the proposed solution algorithm for more sparse bilevel problems, three additional sets of 100 randomly generated problems are solved using the different methods in this section. For this study, half of the elements of each vector and matrix are randomly set to 0. The rest of the parameters to generate the random problems are equal to those provided in Table 1. Table 3 contains the results corresponding to the bilevel problems with 50% sparsity.

As in Table 2, we can observe that although the SOS1 method outperforms the B&B method for all problem sizes, this method provides a number of optimal cases and an average computational time that drastically worsen as the problem dimension increases. It is also shown that the results of the Fortuny-Amat method highly depend on the value of M, being the best value around 50. Again, the results provided by the proposed method are not very sensitive to the value of $\mathcal{M}$ and hence, we focus on those of REG-FA-10 to make the following comparison analysis. For small problems, the results of the proposed method are similar to those of the SOS1 and the best Fortuny-Amat. For medium problems, the proposed method achieves 97 optimal cases in 27 min, versus the 86 optimal cases in 72 min of the SOS1 method and the 92 optimal cases in 71 min of the best Fortuny-Amat. Finally, for large problems, our method provides 61 optimal cases in 3.5 h, versus the 29 optimal cases in 4.5 h of the SOS1 approach and the 48 optimal cases and 5 h of the best-tuned Fortuny-Amat method. Note also that our method attains the lowest average gap (0.07–0.08%) for the non-optimal cases.

5.3 Impact of scaling

Real-life optimization problems often have parameters with different orders of magnitude. For example, some parameters may have values around 10^3, while other parameters may take on values around 1. Such problems are badly scaled and are difficult to solve with optimization solvers. In order to investigate the impact of bad scaling on the proposed solution method, the elements of matrices and vectors $c_1, d_1, A_1, B_1, b_1, c_2, d_2, A_2, B_2, b_2$ are multiplied by 10^z, where z follows a discrete uniform distribution with values 0, 1, 2, 3 and probability 0.25 each. In doing so, one fourth of the elements is multiplied by 1, one fourth by 10, one fourth by 100, and one fourth by 1000. Table 4 contains the results of the randomly generated badly-scaled examples for the three sizes considered.

The first observation is that, although B&B and SOS1 still perform reasonably well for small and medium problems, none of the large problems are solved to

Table 3 Results: impact of sparsity

	Small (n = 50)				Medium (n = 100)				Large (n = 200)			
	#opt	#inf	Time (s)	Gap (%)	#opt	#inf	Time (s)	Gap (%)	#opt	#inf	Time (s)	Gap (%)
B&B	81	3	5100	1.81	42	0	14,019	3.19	9	0	19,964	2.53
SOS1	97	3	2	0.00	86	0	4293	0.48	29	0	16,061	2.19
FA-5	13	24	21	7.14	7	12	4370	8.75	0	11	21,366	5.50
FA-10	67	11	5	1.12	67	5	2123	0.62	34	9	15,709	0.35
FA-20	90	6	8	0.01	88	3	3109	0.12	44	10	15,803	0.21
FA-50	97	3	31	0.00	92	2	4283	0.02	48	0	16,049	0.34
FA-100	97	3	40	0.00	87	6	4918	0.08	45	7	16,364	0.38
FA-200	94	3	94	0.12	81	6	5498	0.09	37	5	17,198	0.45
FA-500	89	3	119	0.24	72	6	6671	0.31	47	3	17,801	0.31
FA-1000	83	3	189	0.39	61	13	6641	0.33	26	13	17,672	0.42
FA-5000	54	3	121	2.26	30	11	6968	2.77	8	9	15,808	1.67
FA-10000	34	3	134	4.04	15	3	5255	6.42	1	3	13,094	6.54
FA-100000	20	3	0	8.91	10	0	0	10.58	1	0	16	9.77
REG	61	3	0	0.52	45	0	1	0.67	22	0	11	0.30
PEN	28	3	0	1.41	11	0	11	1.27	5	0	18	0.90
REG-FA-2	91	3	2	0.08	95	2	453	0.00	76	11	10,409	0.07
REG-FA-5	96	3	3	0.01	97	3	536	0.00	73	14	10,795	0.07
REG-FA-10	97	3	3	0.00	97	1	1644	0.01	61	15	12,512	0.08

optimality and the average gap is 57.79% and 57.06%, respectively. Note also that, for values of M below 1000, the Fortuny-Amat approach was infeasible for all cases of the three problem sizes. Moreover, for larger values of M, the number of optimal cases is always below 10. The regulation and penalty methods also exhibit a very small number of optimal cases. On the other hand, the proposed method for $\mathcal{M} = 5$ achieves the lowest objective function in 91, 80 and 51 cases for small, medium and large problems, respectively. Furthermore, the average solution time for these sizes is 6 s, 1.5 and 6 h, in that order. This means that none of the random problems with $n = 200$ was finished before 6 h. For this reason, the results for large problems should be interpreted with caution, since few methods are able to provide solutions in most cases. Therefore, the average gap of 57.06% linked to the SOS1 method should be understood as the gap between the best solution provided by this method and the solution given by the proposed method after 6 h of running time. The results in Table 4 clearly prove that the proposed solution approach is superior to the existing ones for badly-scaled problems.

Table 4 Results: impact of scaling

	Small (n = 50)				Medium (n = 100)				Large (n = 200)			
	#opt	#inf	Time (s)	Gap (%)	#opt	#inf	Time (s)	Gap (%)	#opt	#inf	Time (s)	Gap (%)
B&B	55	3	11,535	54.58	12	0	19,170	47.84	0	0	21,605	57.79
SOS1	97	3	12	0.00	56	0	18,419	7.02	0	0	21,601	57.06
FA-5	0	100	–	–	0	100	–	–	0	100	–	–
FA-10	0	100	–	–	0	100	–	–	0	100	–	–
FA-20	0	100	–	–	0	100	–	–	0	100	–	–
FA-50	0	100	–	–	0	100	–	–	0	100	–	–
FA-100	0	100	–	–	0	100	–	–	0	100	–	–
FA-200	0	100	–	–	0	100	–	–	0	100	–	–
FA-500	0	100	–	–	0	100	–	–	0	100	–	–
FA-1000	0	95	4408	251.92	0	100	–	–	0	100	–	–
FA-5000	3	22	340	165.59	1	15	14,822	57.00	3	4	21,603	37.09
FA-10000	9	10	257	177.58	4	11	7878	121.04	0	3	20,801	126.56
FA-100000	10	3	0	251.70	3	0	1	202.40	0	0	27	201.24
REG	12	37	1	9.25	4	41	4	6.68	4	85	36	2.51
PEN	3	77	1	38.02	0	89	9	30.60	0	98	16	2.80
REG-FA-2	76	3	4	1.36	68	3	3474	0.92	29	38	20,764	1.90
REG-FA-5	90	3	6	0.75	80	11	5637	0.25	51	10	21,397	1.61
REG-FA-10	91	3	15	0.54	82	6	10702	0.16	31	0	21,788	3.51

6 Conclusions

Linear bilevel problems are non-convex and NP-hard and therefore, finding their optimal solution is computationally costly. In this paper we focus on methods that allow to directly solve LBLP using off-the-shelf optimization software. Among these methods, mixed-integer reformulations provide global optimal solutions at the expense of drastically increasing the computational time, which implies that they can only be applied to small problems. On the other hand, regularization approaches based on iteratively solving nonlinear optimization problems can efficiently solve large bilevel problems, but only guarantee local optimality of the MPCC reformulation.

In this paper we propose a new solution method that combines the advantages of the two aforementioned approaches. First, the regularization approach is used to efficiently find a local optimal point of the MPCC reformulation. Local optimal information is then used in the mixed-integer reformulation of the problem to (1) provide initial values for the binary variables and (2) tune the large-enough constants. The results provided by this method have been compared with those obtained by other general purpose methods when solving a set of 900 randomly generated linear bilevel problems with different size, sparsity and scaling. These results show that the proposed method substantially outperforms the others in terms of number of cases solved to global optimality, average computational time and average optimality gap. For the largest examples, the proposed method achieved the optimal solution in 50% more cases than all the other methods, with an average time 30–95% lower, and an average optimality gap lower than 3.5% in all cases. Finally, it is worth highlighting that the proposed method does not require the adjustment of any large enough constant, and that setting the scaling parameter $\mathcal{M}$ to 5 or 10 is good enough to solve a wide set of different problems.

As future research, it must be investigated how to adapt the proposed methodology so that it can be applied to linear bilevel problems with upper-level constraints that involve both upper- and lower-level variables. Likewise, how to solve bilevel problems with an upper-level objective function that includes dual variables of the lower-level problem requires further research. Moreover, the fact that the coefficients of the upper- and lower-level objective functions are all positive implies that the angle between the objective function vectors is statically small, which, in turn, may reduce the computational burden of solving the LBLP. Therefore, further investigation is required to analyze how the proposed method performs for arbitrary objective function parameters. The results presented in this paper could also be complemented by comparing the computational performance of different commercial solvers, such as GUROBI. Finally, testing the performance of the proposed solution approach in specific real applications is also left for future research.

Acknowledgements This work was supported in part by the Spanish Ministry of Economy, Industry and Competitiveness through Project ENE2016-80638-R and in part by the Research Funding Program for Young Talented Researchers of the University of Málaga through Project PPIT-UMA-B1-2017/18.

References

Bard JF (1991) Some properties of the bilevel programming problem. J Optim Theory Appl 68(2):371–378

Bard JF (1998) practical bilevel optimization: algorithms and applications. Springer, Berlin

Bard JF, Falk JE (1982) An explicit solution to the multi-level programming problem. Comput Oper Res 9(1):77–100

Bard J, Moore J (1990) A branch and bound algorithm for the bilevel programming problem. SIAM J Sci Stat Comput 11(2):281–292

Baringo L, Conejo AJ (2011) Wind power investment within a market environment. Appl Energy 88(9):3239–3247

Baringo L, Conejo AJ (2012) Wind power investment: a benders decomposition approach. IEEE Trans Power Syst 27(1):433–441

Baringo L, Conejo AJ (2013) Risk-constrained multi-stage wind power investment. IEEE Trans Power Syst 28(1):401–411

Baringo L, Conejo AJ (2014) Strategic wind power investment. IEEE Trans Power Syst 29(3):1250–1260

Ben-Ayed O, Blair CE (1990) Computational difficulties of bilevel linear programming. Oper Res 38:556–560

Bialas WF, Karwan MH (1984) Two-level linear programming. Manag Sci 30:1004–1020

Calvete HI, Galé C, Mateo PM (2008) A new approach for solving linear bilevel problems using genetic algorithms. Eur J Oper Res 188(1):14–28

Candler W, Townsley R (1982) A linear two-level programming problem. Comput Oper Res 9(1):59–76

Colson B, Marcotte P, Savard G (2005) Bilevel programming: a survey. 4OR 3(2):87–107

Colson B, Marcotte P, Savard G (2007) An overview of bilevel optimization. Ann Oper Res 153(1):235–256

Dempe S (2002) Foundations of bilevel programming. Springer, Berlin

Dempe S (2003) Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization 52(3):333–359

Dempe S, Dutta J (2010) Is bilevel programming a special case of a mathematical program with complementarity constraints? Math Program 131(1–2):37–48

Dempe S, Franke S (2014) Solution algorithm for an optimistic linear Stackelberg problem. Comput Oper Res 41:277–281

Dempe S, Zemkoho AB (2012) The bilevel programming problem: reformulations, constraint qualifications and optimality conditions. Math Program 138(1–2):447–473

Dempe S, Dutta J, Mordukhovich BS (2007) New necessary optimality conditions in optimistic bilevel programming. Optimization 56(5–6):577–604

Dempe S, Mordukhovich BS, Zemkoho AB (2014) Necessary optimality conditions in pessimistic bilevel programming. Optimization 63(4):505–533

Dempe S, Kalashnikov V, Pérez-Valdés GA, Kalashnikova N (2015) Bilevel programming problems: theory, algorithms and applications to energy networks. Energy systems. Springer, Berlin

Fletcher R, Leyffer S (2002) Numerical experience with solving MPECs as NLPs. Technical report, Department of Mathematics and Computer Science, University of Dundee, Dundee

Fletcher R, Leyffer S (2004) Solving mathematical programs with complementarity constraints as nonlinear programs. Optim Methods Softw 19(1):15–40

Fortuny-Amat J, McCarl B (1981) A representation and economic interpretation of a two-level programming problem. J Oper Res Soc 32(9):783–792

Gabriel SA, Leuthold FU (2010) Solving discretely-constrained MPEC problems with applications in electric power markets. Energy Econ 32(1):3–14

Garces LP, Conejo AJJ, Garcia-Bertrand R, Romero R, Garcés LP (2009) A bilevel approach to transmission expansion planning within a market environment. IEEE Trans Power Syst 24(3):1513–1522

Hansen P, Jaumard B, Savard G (1992) New branch-and-bound rules for linear bilevel programming. SIAM J Sci Stat Comput 13(5):1194–1217

Hasan E, Galiana FD, Conejo AJ (2008) Electricity markets cleared by merit order—part I: finding the market outcomes supported by pure strategy nash equilibria. IEEE Trans Power Syst 23(2):361–371

Hejazi SR, Memariani A, Jahanshahloo G, Sepehri MM (2002) Linear bilevel programming solution by genetic algorithm. Comput Oper Res 29(13):1913–1925

Hu XM, Ralph D (2004) Convergence of a penalty method for mathematical programming with complementarity constraints. J Optim Theory Appl 123(2):365–390

Jenabi M, Ghomi SM, Smeers Y (2013) Bi-level game approaches for coordination of generation and transmission expansion planning within a market environment. IEEE Trans Power Syst 28(3):2639–2650

Jeroslow RG (1985) The polynomial hierarchy and a simple model for competitive analysis. Math Program 32(2):146–164

Jiang Y, Li X, Huang C, Xianing W (2013) Application of particle swarm optimization based on CHKS smoothing function for solving nonlinear bilevel programming problem. Appl Math Comput 219(9):4332–4339

Kazempour SJ, Conejo AJ (2012) Strategic generation investment under uncertainty via benders decomposition. IEEE Trans Power Syst 27(1):424–432

Kazempour SJ, Conejo AJ, Ruiz C (2011) Strategic generation investment using a complementarity approach. IEEE Trans Power Syst 26(2):940–948

Kazempour SJ, Conejo AJ, Ruiz C (2012) Strategic generation investment considering futures and spot markets. IEEE Trans Power Syst 27(3):1467–1476

Li H, Fang L (2012) An evolutionary algorithm for solving bilevel programming problems using duality conditions. Math Probl Eng. https://doi.org/10.1155/2012/471952

Lorenczik S, Malischek R, , Trüby J (2014) Modeling strategic investment decisions in spatial markets. Technical Report 14/09, Köln

Lv Y, Tiesong H, Wang G, Wan Z (2007) A penalty function method based on Kuhn–Tucker condition for solving linear bilevel programming. Appl Math Comput 188(1):808–813

Maurovich-Horvat L, Boomsma TK, Siddiqui AS (2014) Transmission and wind investment in a deregulated electricity industry. IEEE Trans Power 30(3):1633–1643

Mersha AG, Dempe S (2006) Linear bilevel programming with upper level constraints depending on the lower level solution. Appl Math Comput 180(1):247–254

Moiseeva E, Hesamzadeh MR, Biggar DR (2015) Exercise of market power on ramp rate in wind-integrated power systems. IEEE Trans Power Syst 30(3):1614–1623

Morales JM, Zugno M, Pineda S, Pinson P (2014) Electricity market clearing with improved scheduling of stochastic production. Eur J Oper Res 235(3):765–774

Motto ALL, Arroyo JMM, Galiana FDD (2005) A mixed-integer LP procedure for the analysis of electric grid security under disruptive threat. IEEE Trans Power Syst 20(3):1357–1365

Outrata J (2000) On mathematical programs with complementarity constraints. Optim Methods Softw 14(1):117–137

Pisciella P, Bertocchi M, Vespucci MT (2016) A leader-followers model of power transmission capacity expansion in a market driven environment. Comput Manag Sci 13:87–118

Pozo D, Contreras J (2011) Finding multiple nash equilibria in pool-based markets: a stochastic EPEC approach. IEEE Trans Power Syst 26(3):1744–1752

Pozo D, Sauma E, Contreras J (2013) A three-level static MILP model for generation and transmission expansion planning. IEEE Trans Power Syst 28(1):202–210

Ralph D, Wright SJ (2004) Some properties of regularization and penalization schemes for MPECs. Optim Methods Softw 19(5):527–556

Ruiz C, Conejo AJ (2009) Pool strategy of a producer with endogenous formation of locational marginal prices. IEEE Trans Power Syst 24(4):1855–1866

Ruiz C, Conejo AJ (2014) Robust transmission expansion planning. Eur J Oper Res 242:390–401

Ruiz C, Conejo AJ, Smeers Y (2012) Equilibria in an oligopolistic electricity pool with stepwise offer curves. IEEE Trans Power Syst 27(2):752–761

Scheel H, Scholtes S (2000) Mathematical programs with complementarity constraints: stationarity, optimality, and sensitivity. Math Oper Res 25(1):1–22

Scholtes S (2001) Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. SIAM J Optim 11(4):918–936

Shi C, Jie L, Zhang G (2005a) An extended Kuhn–Tucker approach for linear bilevel programming. Appl Math Comput 162(1):51–63

Shi C, Jie L, Zhang G (2005b) An extended Kth-best approach for linear bilevel programming. Appl Math Comput 164(3):843–855

Shi C, Zhang G, Jie L (2005c) On the definition of linear bilevel programming solution. Appl Math Comput 160(1):169–176

Shi C, Jie L, Zhang G, Zhou H (2006) An extended branch and bound algorithm for linear bilevel programming. Appl Math Comput 180(2):529–537

Siddiqui S, Gabriel SA (2012) An SOS1-based approach for solving MPECs with a natural gas market application. Netw Spat Econ 13(2):205–227

Sinha A, Malo P, Deb K (2013) Efficient evolutionary algorithm for single-objective bilevel optimization. arXiv:1303.3901

Strekalovsky AS, Orlov AV, Malyshev AV (2010a) On computational search for optimistic solutions in bilevel problems. J Glob Optim 48(1):159–172

Strekalovsky AS, Orlov AV, Malyshev AV (2010b) Numerical solution of a class of bilevel programming problems. Numer Anal Appl 3(2):165–173

The ILOG CPLEX (2015) http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html

Valinejad J, Barforoushi T (2015) Generation expansion planning in electricity markets: a novel framework based on dynamic stochastic MPEC. Int J Electr Power Energy Syst 70:108–117

Von Stackelberg H (1952) The theory of the market economy. Oxford University Press, Oxford

White DJ, Anandalingam G (1993) A penalty function approach for solving bi-level linear programs. J Glob Optim 3(4):397–419

Wogrin S, Centeno E, Barquín J (2011) Generation capacity expansion in liberalized electricity markets: a stochastic MPEC approach. IEEE Trans Power Syst 26(4):2526–2532

Wogrin S, Barquin J, Centeno E (2013) Capacity expansion equilibria in liberalized electricity markets: an EPEC approach. IEEE Trans Power Syst 28(2):1531–1539

Zhang G, Lu J, Gao Y (2015) Multi-level decision making: models, methods and applications. Intelligent systems reference library. Springer, Berlin

Zugno M, Morales JM, Pinson P, Madsen H (2013) Pool strategy of a price-maker wind power producer. IEEE Trans Power Syst 28(3):3440–3450