Mathematical Programming (2021) 188:461–521
https://doi.org/10.1007/s10107-020-01601-2

FULL LENGTH PAPER

Series B

Outer approximation for global optimization of mixed-integer quadratic bilevel problems

Thomas Kleinert[1,2] · **Veronika Grimm**[2,3] · **Martin Schmidt**[2,4]

Received: 9 January 2020 / Accepted: 22 November 2020 / Published online: 8 February 2021
© The Author(s) 2021

Abstract

Bilevel optimization problems have received a lot of attention in the last years and decades. Besides numerous theoretical developments there also evolved novel solution algorithms for mixed-integer linear bilevel problems and the most recent algorithms use branch-and-cut techniques from mixed-integer programming that are especially tailored for the bilevel context. In this paper, we consider MIQP-QP bilevel problems, i.e., models with a mixed-integer convex-quadratic upper level and a continuous convex-quadratic lower level. This setting allows for a strong-duality-based transformation of the lower level which yields, in general, an equivalent nonconvex single-level reformulation of the original bilevel problem. Under reasonable assumptions, we can derive both a multi- and a single-tree outer-approximation-based cutting-plane algorithm. We show finite termination and correctness of both methods and present extensive numerical results that illustrate the applicability of the approaches. It turns out that the proposed methods are capable of solving bilevel instances with several thousand variables and constraints and significantly outperform classical solution approaches.

Keywords Bilevel optimization · Outer approximation · Quadratic programming · Convex mixed-integer nonlinear optimization

✉ Thomas Kleinert
thomas.kleinert@fau.de

Veronika Grimm
veronika.grimm@fau.de

Martin Schmidt
martin.schmidt@uni-trier.de

1 Friedrich-Alexander-Universität Erlangen-Nürnberg, Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany

2 Energie Campus Nürnberg, Fürther Str. 250, 90429 Nürnberg, Germany

3 Friedrich-Alexander-Universität Erlangen-Nürnberg, Economic Theory, Lange Gasse 20, 90403 Nürnberg, Germany

4 Department of Mathematics, Trier University, Universitätsring 15, 54296 Trier, Germany

Mathematics Subject Classification 90-08 · 90C11 · 90C30 · 90C26 · 90C46

1 Introduction

Bilevel optimization problems are used in various applications, e.g., in energy markets [9,12,20,29,31,34,39], in critical infrastructure defense [11,16], or in pricing problems [15,40] to model hierarchical decision processes. As such, they embed one optimization problem, the so-called lower-level problem, into the constraints of a so-called upper-level problem. This leads to inherent nonconvexities, which render already linear bilevel problems with a linear upper- and lower-level problem NP-hard; see, e.g., [14,32,35].

In this work, we study mixed-integer quadratic bilevel problems of the form

$$\begin{aligned}
\min_{x,\bar{y}} \quad & q_u(x,\bar{y}) = \frac{1}{2}x^\top H_u x + c_u^\top x + \frac{1}{2}\bar{y}^\top G_u \bar{y} + d_u^\top \bar{y} \\
\text{s.t.} \quad & Ax + B\bar{y} \geq a, \\
& x_i \in \mathbb{Z} \cap [x_i^-, x_i^+] \quad \text{for all } i \in I := \{1,\dots,|I|\}, \\
& x_i \in \mathbb{R} \quad \text{for all } i \in R := \{|I|+1,\dots,n_x\}, \\
& \bar{y} \in \arg\min_y \left\{ q_l(y) = \tfrac{1}{2}y^\top G_l y + d_l^\top y : Cx_I + Dy \geq b,\ y \in \mathbb{R}^{n_y} \right\}, \qquad (1)
\end{aligned}$$

where $H_u \in \mathbb{R}^{n_x \times n_x}$, $G_u \in \mathbb{R}^{n_y \times n_y}$, and $G_l \in \mathbb{R}^{n_y \times n_y}$ are symmetric and positive semidefinite matrices. Furthermore, we have vectors $c_u \in \mathbb{R}^{n_x}$, $d_u, d_l \in \mathbb{R}^{n_y}$, matrices $A \in \mathbb{R}^{m_u \times n_x}$, $B \in \mathbb{R}^{m_u \times n_y}$, $C \in \mathbb{R}^{m_l \times |I|}$, $D \in \mathbb{R}^{m_l \times n_y}$, as well as right-hand side vectors $a \in \mathbb{R}^{m_u}$ and $b \in \mathbb{R}^{m_l}$. The variables $x = (x_I, x_R)$ denote the integer (x_I) and continuous (x_R) upper-level variables and y denotes the (continuous) lower-level variables. Note that we w.l.o.g. ordered the integer and continuous upper-level variables for the ease of presentation. In this setup, the upper-level problem is a convex mixed-integer quadratic problem (MIQP) and for fixed integer upper-level variables x_I, the lower level is a convex quadratic problem (QP), i.e., it is a parametric convex QP. In total, we are facing an MIQP-QP bilevel problem with the following key properties:

(i) All upper-level integer variables x_I are bounded.
(ii) All linking variables, i.e., upper-level variables that appear in the lower-level constraints, are integer.

We note that in Problem (1), we implicitly assume that all integer upper-level variables are linking variables. However, this formulation also contains the more general case of integer upper-level variables that do not appear in the lower-level problem by setting some columns in the matrix C to zero.

The main motivation of this work is to exploit the two properties above to develop a multi- and a single-tree solution approach for Problem (1) based on outer approximation for convex mixed-integer nonlinear problems (MINLP) [18,25]. The above-mentioned two properties are required by many other state-of-the-art algorithms for linear bilevel problems with purely integer linear (ILP-ILP) or mixed-integer linear upper- and lower-level problems (MILP-MILP); see, e.g. [21,23,43,50,54].

These methods successfully use bilevel-tailored branch-and-bound or branch-and-cut methods to solve quite large instances of hundreds or thousands of variables and constraints. However, they cannot directly deal with continuous lower-level problems and/or have not yet been extended to the convex-quadratic case—if this is possible at all. An extension of [54] to the general class of mixed-integer nonlinear bilevel problems with integer linking variables is proposed in [41] but the computational study therein only covers mixed-integer linear bilevel problems. In [3], the authors use multi-parametric programming techniques to solve bilevel problems with convex MIQPs on both levels. This algorithm explicitly allows for continuous linking variables. The backbone of the approach is the computation of the optimal solutions of the lower-level problem as a function of the upper-level decisions. This step is very costly and, according to the authors, it is evident that this algorithm is not intended for the solution of larger problems. The computational study included in the paper deals with problems up to 65 variables. In addition, the algorithm is only applicable for problems that possess unique lower-level solutions.

Aside from mixed-integer bilevel problems, algorithms for various classes of continuous convex-quadratic bilevel problems have been proposed. However, in contrast to algorithms for mixed-integer bilevel problems, reported numerical results for continuous convex-quadratic bilevel problems seem to cover only rather small instances. A branch-and-bound algorithm for bilevel problems with a convex upper-level problem and a strictly convex lower-level problem is proposed in [5]. The author demonstrates the effectiveness of his method for problems with up to 15 variables and 20 constraints on each level. In [6], a convex-quadratic lower-level problem is replaced by its Karush–Kuhn–Tucker (KKT) conditions and then a branching on the complementarity constraints is applied. The authors report results for problems with up to 60 upper-level and 40 lower-level variables. This approach is generalized from linear upper-level to convex upper-level problems in [19]. Two different descent algorithms for bilevel problems with a strictly convex lower level and a concave or convex upper level are proposed in [53]. However, the authors do not provide computational results. Recently, also neural networks are used to tackle continuous convex-quadratic bilevel problems; see [33,42].

To the best of our knowledge, tailored algorithms for mixed-integer quadratic bilevel problems of the form (1) are neither reported nor has their efficiency been demonstrated in a comprehensive computational study. In fact, there exist no code packages that can be used as a benchmark for our proposed solution techniques. Thus, we use well-known single-level reformulations based on KKT conditions and strong duality as a benchmark in our computational study in Sect. 4.

Our contribution is the following. We consider bilevel problems with a mixed-integer convex-quadratic upper level and a convex-quadratic lower level. For this nonconvex problem class, we provide an equivalent reformulation to a convex MINLP that uses strong duality of the lower level; see Sect. 2. Further, in Sect. 3 we propose a multi- and a single-tree solution approach that are both inspired by outer-approximation techniques for convex MINLPs. We prove the correctness of the methods and discuss further extensions. We are not aware of any other work that applies outer-approximation techniques from the area of convex MINLP to mixed-integer bilevel programming. In Sect. 4, we evaluate the effectiveness of the proposed

approaches in an extensive numerical study, in which we solve instances with up to several thousand variables and constraints. We conclude in Sect. 5.

2 A convex single-level reformulation

Most solution techniques for bilevel problems rely on a reformulation of the bilevel problem to a single-level problem [14]. For problems with a convex lower level, the lower-level problem can be replaced by its nonconvex KKT conditions. Especially for problems with linear lower-level constraints, this approach is very popular, because it allows for a mixed-integer linear reformulation of the KKT complementarity conditions using additional binary variables and big-M values; see, e.g., [26]. With this approach the bilevel problem (1) can be equivalently transformed to the following mixed-integer single-level problem

$$\min_{x,y,v,\lambda} \quad q_u(x, y) \tag{2a}$$

$$\text{s.t.} \quad Ax + By \geq a, \quad x_i \in \mathbb{Z} \cap [x_i^-, x_i^+] \quad \text{for all } i \in I, \tag{2b}$$

$$Cx_I + Dy \geq b, \tag{2c}$$

$$G_l y + d_l = D^T \lambda, \quad \lambda \geq 0, \tag{2d}$$

$$v \in \{0, 1\}^{m_l}, \quad Cx_I + Dy - b \leq M_1 v, \quad \lambda \leq M_2(1 - v), \tag{2e}$$

where Constraints (2b) and (2c) model primal feasibility of the upper and lower level, respectively. Constraint (2d) models dual feasibility of the lower-level problem with dual lower-level variables $\lambda \in \mathbb{R}^{m_l}_{\geq 0}$. Constraints (2e) ensure KKT complementarity via additional binary variables v_j and sufficiently large numbers M_1 and M_2. Similarly, a strong-duality-based reformulation can be derived by replacing KKT complementarity (2e) by the strong-duality equation of the lower level. This approach is significantly less used in practice because even for linear bilevel problems one obtains nonconvex bilinear terms due to products of primal upper-level and dual lower-level variables. These terms can only be linearized if all linking variables are integer. Recently, in [55], a numerical study is provided that compares the KKT approach with the strong-duality approach for linear bilevel problems with integer linking variables and continuous lower-level problems. The authors conclude that the strong-duality reformulation works significantly better than the KKT reformulation for problems with large lower-level problems. Other contributions that successfully apply the strong-duality reformulation to linear bilevel problems include, e.g., [2,27,28,30,40,44]. Except from Bard, who briefly sketches the idea in [5], we are not aware of any works that use strong duality for bilevel problems with quadratic lower levels. One reason might be that the resulting strong-duality equation is quadratic. Opposed to the linearized KKT reformulation, the strong-duality-based reformulation thus yields a quadratically constrained program. Despite this drawback, we derive such a strong-duality-based single-level reformulation after introducing some notation.

2.1 General notation

The bilevel constraint region is denoted by

$$P := \{(x, y)\colon Ax + By \geq a,\ Cx_I + Dy \geq b,\ x_i \in \mathbb{Z} \cap [x_i^-, x_i^+] \text{ for all } i \in I\}.$$

Throughout this paper, we assume that P is bounded. This set corresponds to the set obtained by relaxing the optimality of the lower-level problem. Its projection onto the decision space of the upper level is given by

$$P_u := \{x\colon \exists\, y \text{ such that } (x, y) \in P\}.$$

For fixed $\bar{x} = (\bar{x}_R, \bar{x}_I) \in P_u$, the lower-level feasible region is given by

$$P_l(\bar{x}_I) = \{y\colon Dy \geq b - C\bar{x}_I\}$$

and the rational reaction set of the lower level reads

$$M(\bar{x}_I) = \arg\min_y \{q_l(y) : y \in P_l(\bar{x}_I)\}.$$

Since G_l is semidefinite, the lower level may not have a unique solution, i.e., $M(\bar{x}_I)$ may not be a singleton. In such a case, we assume the optimistic bilevel solution, i.e., $\bar{y} \in M(\bar{x}_I)$ is chosen in favor of the upper level; see, e.g., Chapter 1 in [14]. In Problem (1), this is indicated by "$\min_{x,\bar{y}}$" in the upper-level objective, i.e., the upper-level minimizes over x and $\bar{y}$. We emphasize that single-level reformulations, like, e.g., Problem (2), implicitly assume the optimistic bilevel solution, such that there is no need to distinguish between y and $\bar{y}$. Further, note that our solution approach explicitly allows for $M(\bar{x}_I) = \emptyset$ for some $\bar{x} \in P_u$. Finally, the bilevel feasible set is given by

$$\mathcal{F} = \{(x, \bar{y})\colon x \in P_u,\ \bar{y} \in M(\bar{x}_I)\}.$$

If $\mathcal{F} = \emptyset$, then Problem (1) is infeasible.

2.2 Strong-duality-based nonconvex single-level reformulation

We now use strong duality of the lower-level problem to transform Problem (1) into an equivalent nonconvex single-level problem. The parametric Lagrangian dual problem of the parametric lower level

$$\min_y \quad q_l(y) \quad \text{s.t.} \quad Dy \geq b - Cx_I \tag{3}$$

with parameter x_I is given by

$$\max_{\lambda \geq 0} \quad g(x_I; \lambda) \tag{4}$$

with $g(x_I;\lambda) = \inf_y \mathcal{L}(x_I; y, \lambda)$; see [10]. In our setup, the Lagrangian $\mathcal{L}$ reads

$$\mathcal{L}(x_I; y, \lambda) = \frac{1}{2} y^\top G_l y + d_l^\top y - \lambda^\top (Cx_I + Dy - b). \tag{5}$$

Since $\mathcal{L}(x_I; y, \lambda)$ is convex and differentiable in y, the infimum is given by

$$\nabla_y \mathcal{L}(x_I; y, \lambda) = G_l y + d_l - D^\top \lambda = 0. \tag{6}$$

In order to denote the Lagrangian dual problem in its general form (4), we could use Expression (6) to obtain $y = G_l^{-1}(D^\top \lambda - d_l)$. This can then be used to substitute the primal variable y in the Lagrangian (5) to obtain

$$g(x_I;\lambda) = -\frac{1}{2}(D^\top \lambda - d_l)^\top G_l^{-1}(D^\top \lambda - d_l) - (Cx_I - b)^\top \lambda.$$

However, this only works if G_l is regular, e.g., if G_l is strictly definite. In the more general case of semidefinite matrices, we can explicitly keep the primal variable y and substitute $D^\top \lambda = G_l y + d_l$ in the Lagrangian (5). This yields the dual problem

$$\begin{aligned} \max_{y,\lambda} \quad & \bar{g}(x_I; y, \lambda) = -\frac{1}{2} y^\top G_l y - (Cx_I - b)^\top \lambda \\ \text{s.t.} \quad & G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0. \end{aligned} \tag{7}$$

Note that $\bar{g}(x_I; \cdot)$ is a concave-quadratic function in y and λ because G_l is positive semidefinite. Thus, the dual problem (7) is a parametric concave-quadratic maximization problem over affine-linear constraints. Since Problem (7) does not involve the inverse G_l^{-1}, we use formulation (7) also in the case when G_l is strictly positive definite.

In the following, we only consider $x \in P_u$, i.e., upper-level variables for which the parametric lower-level problem is feasible. The parametric lower-level problem (3) is a convex-quadratic minimization problem over affine-linear constraints. Consequently, duality conditions apply without requiring additional constraint qualifications; see Chapter 5.2.3 in [10] and Theorem 24.1 in [52]. For every primal-dual feasible point (y, λ), i.e., $y \in P_l(x_I)$ and $\lambda \geq 0$ that fulfill $G_l y + d_l = D^\top \lambda$, weak duality

$$q_l(y) \geq \bar{g}(x_I; y, \lambda) \tag{8}$$

holds. Furthermore, for optimal primal-dual points (y^*, λ^*), strong duality holds, i.e., $q_l(y^*) = \bar{g}(x_I; y^*, \lambda^*)$. Together with weak duality (8), strong duality can be enforced by the constraint $q_l(y) - \bar{g}(x_I; y, \lambda) \leq 0$, i.e.,

$$c(x_I, y, \lambda) := q_l(y) - \bar{g}(x_I; y, \lambda) = y^\top G_l y + d_l^\top y - b^\top \lambda + \lambda^\top C x_I \leq 0. \tag{9}$$

This strong-duality inequality is convex in y but the bilinear term $\lambda^\top C x_I$ is nonconvex. Using (9), the bilevel problem (1) can be recast as the equivalent single-level nonconvex mixed-integer quadratically constrained quadratic program (MIQCQP):

$$\begin{aligned} \min_{x,y,\lambda} \quad & q_u(x, y) \\ \text{s.t.} \quad & (x, y) \in P, \\ & G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\ & c(x_I, y, \lambda) \leq 0. \end{aligned} \tag{10}$$

2.3 Convexification of the strong-duality constraint

The only nonconvexity in the strong-duality inequality (9) is the bilinear product $\lambda^\top C x_I$. Since the variables x_I are integer, this product can be reformulated using a binary expansion. For the ease of presentation, we w.l.o.g. assume in the following that $x_i^- = 0$ for all $i \in I$. The products of binary and continuous variables can then be linearized by several techniques. According to the numerical study in [55], the following approach works best in a bilevel context. We express the integer variables x_j with the help of $\bar{r}_j = \lfloor \log_2(x_j^+) \rfloor + 1$ many auxiliary binary variables s_{jr}:

$$s_{jr} \in \{0, 1\}, \qquad j \in I,\ r \in [\bar{r}_j], \tag{11a}$$

$$x_j = \sum_{r=1}^{\bar{r}_j} 2^{r-1} s_{jr}, \qquad j \in I. \tag{11b}$$

With this we obtain

$$\lambda^\top C x_I = \sum_{i=1}^{m_l} \sum_{j=1}^{|I|} c_{ij} \lambda_i x_j = \sum_{i=1}^{m_l} \sum_{j=1}^{|I|} \sum_{r=1}^{\bar{r}_j} 2^{r-1} c_{ij} \lambda_i s_{jr} = \sum_{j=1}^{|I|} \sum_{r=1}^{\bar{r}_j} 2^{r-1} s_{jr} \sum_{i=1}^{m_l} c_{ij} \lambda_i.$$

Now, we replace the binary-continuous products $s_{jr} \sum_{i=1}^{m_l} c_{ij} \lambda_i$ by introducing auxiliary continuous variables w_{jr} and enforce

$$w_{jr} = s_{jr} \sum_{i=1}^{m_l} c_{ij} \lambda_i \tag{12}$$

by the additional constraints

$$w_{jr} \leq \overline{\lambda}_j s_{jr}, \qquad j \in I,\ r \in [\bar{r}_j], \tag{13a}$$

$$w_{jr} \leq \sum_{i=1}^{m_l} c_{ij} \lambda_i + \underline{\lambda}_j (s_{jr} - 1), \qquad j \in I,\ r \in [\bar{r}_j], \tag{13b}$$

$$w_{jr} \geq \underline{\lambda}_j s_{jr}, \qquad j \in I,\ r \in [\bar{r}_j], \tag{13c}$$

$$w_{jr} \geq \sum_{i=1}^{m_l} c_{ij} \lambda_i + \overline{\lambda}_j (s_{jr} - 1), \qquad j \in I,\ r \in [\bar{r}_j]. \tag{13d}$$

In this formulation, we need bounds $\overline{\lambda}_j \geq \sum_{i=1}^{m_l} c_{ij}\lambda_i$, and $\underline{\lambda}_j \leq \sum_{i=1}^{m_l} c_{ij}\lambda_i$ which are, in practice, often derived by some suitable big-M. The need for such bounds is also a major drawback in the KKT-based reformulation (2). In [46], it is shown that wrong big-Ms can lead to suboptimal solutions or points that are actually bilevel infeasible. Unfortunately, even verifying that the bounds are correctly chosen is, in general, at least as hard as solving the original bilevel problem; see [37]. Thus, if possible, big-Ms should be derived using problem-specific knowledge. However, we point out that our solution approaches introduced in Sect. 3 compute bilevel-feasible points independent of the big-Ms. Thus, in case of too small big-Ms, our algorithms may terminate with a suboptimal solution but never compute bilevel-infeasible points. We discuss this also later in Sect. 3.3.

Using (11) and (13), we rewrite Constraint (9) as

$$\hat{c}(y, \lambda, w) := y^\top G_l y + d_l^\top y - b^\top \lambda + \sum_{j=1}^{|I|} \sum_{r=1}^{\bar{r}_j} 2^{r-1} w_{jr} \leq 0, \tag{14}$$

which is convex in y and linear in λ and w. Note that $2^{r-1} \leq x_j^+$ holds, i.e., for reasonable bounds x_j^+, the exponential coefficients 2^{r-1} can be considered as numerically stable.

Finally, the single-level problem (10) can be stated equivalently as

$$\begin{aligned} \min_{x,y,\lambda,w,s} \quad & q_u(x, y) \\ \text{s.t.} \quad & (x, y) \in P, \\ & G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\ & \lambda^\top C x_I \text{ linearization: (11), (13)}, \\ & \text{Strong duality: (14)}. \end{aligned} \tag{15}$$

This is a convex MIQCQP that has more binary variables and constraints compared to the nonconvex MIQCQP (10). We denote the feasible set of Problem (15) by Ω, feasible points by $z = (x, y, \lambda, w, s) \in \Omega$ and optimal points by z^*. By construction of Problem (15), we have the following equivalence result.

Lemma 1 *The feasible set of Problem* (15) *projected on the* (x, y)*-space equals the bilevel feasible set of the bilevel Problem* (1)*, i.e.,* $\mathcal{F} = \text{Proj}_{(x,y)}(\Omega)$ *holds. In addition, for every global optimal solution* $z^* = (x^*, y^*, \lambda^*, w^*, s^*)$ *of Problem* (15)*,* (x^*, y^*) *is a global optimal solution for Problem* (1) *and every global optimal solution* (x^*, y^*) *of Problem* (1) *is part of an optimal solution* z^* *of Problem* (15)*.*

3 Two outer-approximation solution approaches

Problem (15) is an MIQCQP in which the single quadratic constraint is convex. Such problem classes can be solved, e.g., directly by modern solvers like Gurobi

or CPLEX. On the other hand, Problem (15) belongs to the broader class of convex MINLPs. For such problems, a variety of approaches exist, e.g., nonlinear branch-and-bound or multi- and single-tree methods based on outer approximation, generalized Benders decomposition, and extended cutting-planes; see [7] for a detailed survey of these methods. In this section, we introduce outer-approximation techniques that are tailored for mixed-integer quadratic bilevel problems. The general idea is to relax the convex-quadratic strong-duality inequality (14) of the lower level in Problem (15) to obtain an MIQP. Strong duality is then resolved by iteratively adding linear outer-approximation cuts. In its simplest form, this is a direct application of Kelley's cutting-planes approach [36], which would add linear outer-approximation cuts until the strong-duality inequality (14) is satisfied up to a certain tolerance. Our preliminary numerical results indicate that this requires an enormous amount of iterations. We thus discuss a more sophisticated multi-tree approach and its single-tree variant in Sects. 3.1 and 3.2.

3.1 A multi-tree outer-approximation approach

The well-known multi-tree outer approximation for convex MINLPs was first proposed in [18] and has been enhanced in [8,25]. It alternatingly solves a mixed-integer linear master problem and a convex nonlinear problem (NLP) as a subproblem. The master problem is a mixed-integer linear relaxation of the original convex MINLP and is tightened subsequently by adding linear outer-approximation cuts for the convex nonlinearities. The convex nonlinear subproblem results from fixing all integer variables to the solution of the master problem in the original convex MINLP. Under suitable assumptions, every feasible integer solution of the master problem is visited at most once, and the algorithm terminates after a finite number of iterations with the correct solution. The algorithm proposed in this subsection is very much inspired by this scheme. The master problem that we solve in every iteration $p \geq 0$ is given by

$$\begin{aligned} \min_{x,y,\lambda,w,s} \quad & q_u(x, y) \\ \text{s.t.} \quad & (x, y) \in P, \\ & G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\ & \lambda^\top C x_I \text{ linearization: (11), (13)}, \\ & \bar{c}(\bar{y}^\ell; y, \lambda, w) \leq 0, \quad \ell = 0, \ldots, p-1, \end{aligned} \tag{M^p}$$

where $\bar{c}(\bar{y}^\ell; y, \lambda, w) \leq 0$ is the linear outer-approximation cut that is added to the master problem after every iteration. Thus, in iteration p, the master problem (M^p) contains p outer-approximation cuts. We shed some more light on $\bar{c}$ later, but first emphasize that Problem (M^p) is a convex MIQP. This is in contrast to the standard outer-approximation literature, where the objective function is relaxed and iteratively approximated as well, resulting in a mixed-integer linear master problem. The rationale is that, in our implementation, the main working horse is a state-of-the-art solver like Gurobi or CPLEX. In recent years, these solvers made significant progress in solving

convex MIQPs effectively. Thus, we want to exploit these highly evolved solvers as much as possible.

Now, we give some more details on the linear function $\bar{c}$, which is derived using the first-order Taylor approximation of the convex strong-duality inequality (14). For a general convex function $h(v)$, the first-order Taylor approximation at $\bar{v}$ reads

$$h(\bar{v}) + \nabla_v h(\bar{v})^\top (v - \bar{v}).$$

Applied to (14), this gives

$$\bar{c}(\bar{y}; y, \lambda, w) := 2\bar{y}^\top G_l y + d_l^\top y - b^\top \lambda + \sum_{j \in I} \sum_{r=1}^{\bar{r}_j} 2^{r-1} w_{jr} - \bar{y}^\top G_l \bar{y}. \tag{16}$$

Since $\hat{c}(y, \lambda, w)$ is linear in λ and w, the first-order Taylor approximation (16) is parameterized solely by $\bar{y}$. The effectiveness of the proposed outer-approximation approach will depend on the actual selection of $\bar{y}$, but $\bar{c}(\bar{y}; y, \lambda, w) \leq 0$ is a valid inequality no matter how $\bar{y}$ is obtained. This is shown in the following lemma, in which $\mathcal{M}^p$ denotes the feasible set of (M^p).

Lemma 2 *For every iteration $p \geq 1$, $\Omega \subseteq \mathcal{M}^p \subseteq \mathcal{M}^{p-1}$ holds.*

The consequence of Lemma 2 is that every master problem (M^p) is a relaxation of the single-level reformulation (15).

Proof Let $z = (x, y, \lambda, w, s) \in \Omega$ be a feasible point of Problem (15). In particular, z fulfills strong duality, i.e., $\hat{c}(y, \lambda, w) = 0$. Obviously, $z \in \mathcal{M}^0$ holds because $\mathcal{M}^0$ corresponds exactly to Ω without the strong-duality inequality (14). In addition, since $\hat{c}$ is convex, its first-order Taylor approximation is a global underestimator at any point $\bar{y}$, i.e.,

$$\bar{c}(\bar{y}; y, \lambda, w) \leq \hat{c}(y, \lambda, w) = 0.$$

This holds in particular for the choice $\bar{y} = \bar{y}^\ell$ for any $\ell = 1, \ldots, p$ and $p \geq 1$. Hence, $z \in \mathcal{M}^p$. Further, $\mathcal{M}^p \subseteq \mathcal{M}^{p-1}$ follows by construction. □

In the following, we give details on how to select the linearization points $\bar{y}$. We therefore assume that the master problem (M^p) is solvable and denote its solution by $z^p = (x^p, y^p, \lambda^p, w^p, s^p)$. According to Kelley's cutting plane approach [36], one would add an outer-approximation cut (16) at the solution of the master problem, i.e., one would add the inequality $\bar{c}(y^p; y, \lambda, w) \leq 0$. Kelley has shown that this yields a convergent approach. However, in many cases this will turn out to be inefficient, because these cuts are rather weak. The outer-approximation methods in the spirit of [8,18,25] additionally solve a convex nonlinear subproblem that results from fixing the integer variables in the original convex MINLP (or an auxiliary feasibility problem if the subproblem is infeasible) to obtain suitable linearization points $\bar{y}$. In our context,

the subproblem is given by fixing $x_I = x_I^p$ and $s = s^p$ in the convex MINLP (15), which yields the convex quadratically constrained quadratic problem (QCQP):

$$\begin{aligned} \min_{x_R, y, \lambda, w} \quad & q_u(x_I^p, x_R, y) \\ \text{s.t.} \quad & (x_I^p, x_R, y) \in P, \\ & G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\ & w_{jr} = s_{jr}^p \sum_{i=1}^{m_l} c_{ij} \lambda_i, \quad j \in I,\ r \in [\bar{r}_j], \\ & \hat{c}(y, \lambda, w) \leq 0. \end{aligned} \tag{S^p}$$

We denote the feasible set of (S^p) by $\mathcal{S}^p$ and first assume $\mathcal{S}^p \neq \emptyset$. The infeasible case is discussed afterward. Let $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$ be the solution of the subproblem (S^p). For the correctness of our proposed algorithms, we need a technical assumption that is also used in [8,18,25].

Assumption 1 For every feasible subproblem (S^p), the Abadie constraint qualification holds at the solution $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$, i.e., the tangent cone and the linearized tangent cone coincide at $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$.

A formal description of standard cones in nonlinear optimization along with the corresponding theory that is also required in the proof of the following lemma can be found, e.g., in [45]. In theory, Assumption 1 is crucial for the termination of the methods described in this section. An indication that this assumption is not fulfilled in practice is cycling, i.e., a certain integer solution is computed more than once. In our preliminary numerical tests, cycling hardly ever occurred for any of the proposed approaches, such that we disabled an expensive cycling detection (e.g., storing all integer solutions and adding a no-good-cut if an integer solution is computed for the second time) in our computations.

We now show that it is indeed a good idea to linearize the strong-duality inequality (14) at the solution of the subproblem $\bar{y}^p$ instead of at the solutions of the master problem y^p.

Lemma 3 *Let $z^p = (x_I^p, x_R^p, y^p, \lambda^p, w^p, s^p)$ be an optimal solution of the master problem (M^p) and assume that the subproblem (S^p) is feasible and has the optimal solution $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$. Suppose further that Assumption 1 holds and consider the new master problem that is obtained by adding the outer-approximation cut $\bar{c}(\bar{y}^p; y, \lambda, w) \leq 0$ to (M^p). Then, for any feasible point of the form $z = (x_I^p, x_R, y, \lambda, w, s^p)$ of this problem the following holds:*

$$q_u(x_I^p, x_R, y) \geq q_u(x_I^p, \bar{x}_R^p, \bar{y}^p).$$

Proof We consider x_I^p and s^p fixed and assume that (S^p) is feasible with optimal solution $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$. Thus, $(\bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$ fulfills the convex strong-duality inequality (14). Since weak duality holds anyway, we obtain $\hat{c}(\bar{y}^p, \bar{\lambda}^p, \bar{w}^p) = 0$. Now,

let $z = (x_I^p, x_R, y, \lambda, w, s^p)$ be feasible for (M^p) and let $\bar{c}(\bar{y}^p; y, \lambda, w) \leq 0$, i.e., z is feasible for a suitably chosen master problem in iteration $p+1$. In the following, we abbreviate the vector $v = (y, \lambda, w)$ for the ease of presentation. Then, we have

$$0 \geq \bar{c}(\bar{y}^p; v) = \hat{c}(\bar{v}^p) + \nabla_v \hat{c}(\bar{v}^p)(v - \bar{v}^p) = \nabla_v \hat{c}(\bar{v}^p)^\top (v - \bar{v}^p),$$

i.e., $\nabla_v \hat{c}(\bar{v}^p)^\top (v - \bar{v}^p) \leq 0$, which means that $(v - \bar{v}^p)$ is in the linearized tangent cone $T^{\text{lin}}_{\mathcal{S}^p}(\bar{v}^p)$. Due to Assumption 1, $T^{\text{lin}}_{\mathcal{S}^p}(\bar{v}^p)$ equals the tangent cone $T_{S^p}(\bar{v}^p)$, which gives $(v - \bar{v}^p) \in T_{S^p}(\bar{v}^p)$. For all directions $d \in T_{S^p}(\bar{v}^p)$, we know that the property

$$\nabla_v q_u(x_I^p, \bar{x}_R^p, \bar{y}^p)^\top d \geq 0 \tag{17}$$

holds. Thus, we have

$$q_u(x_I^p, x_R, y) \geq q_u(x_I^p, \bar{x}_R^p, \bar{y}^p) + \nabla_v q_u(x_I^p, \bar{x}_R^p, \bar{y}^p)^\top (v - \bar{v}^p) \geq q_u(x_I^p, \bar{x}_R^p, \bar{y}^p).$$

The first inequality follows because q_u is convex, i.e., its first-order Taylor approximation is a global underestimator. The second inequality follows from Inequality (17). □

In contrast to the slightly different setting in [25], using the solution of the subproblem (S^p) as the linearization point of the outer-approximation inequality (14) does not explicitly cut off the related integer solution x_I^p. The reason is our modified master problem, that does not linearize and approximate the convex objective function. Nevertheless, Lemma 3 lets us conclude that every integer assignment x_I that yields a feasible subproblem (S^p) needs to be visited only once, because the objective cannot be improved by visiting such a solution for a second time. This will be one of the key properties to prove finite termination of our algorithm.

We now consider the case of an infeasible subproblem (S^p). In [18], it is argued that in order to eliminate x_I^p from further consideration, an integer no-good-cut must be introduced. In our application, this is a straightforward task. The subproblem (S^p) is fully parameterized by fixed upper-level variables x_I^p. For these variables we have a binary expansion available anyway (the variables s) so that a simple binary no-good-cut on s can be used. However, such no-good-cuts are known to cause numerical instabilities. As a remedy, in [25], it is proposed to derive cutting planes from an auxiliary feasibility problem that indeed cut off the integer solution x_I^p. The feasibility problem minimizes the constraint violations of the infeasible subproblem in some suitable sense, e.g., via the ℓ^1- or the ℓ^∞-norm. Recap that z^p is a solution of the master problem (M^p). In particular, (y^p, λ^p) is primal-dual feasible for the lower-level problem (3) with fixed x_I^p. Thus, the latter problem also has an optimal solution that fulfills strong duality. Since the subproblem (S^p) is infeasible, this optimal solution must be infeasible for the upper-level constraints. On the other hand, z^p must be feasible for the subproblem (S^p) without the strong-duality inequality, because z^p is feasible for Problem (M^p). Thus, a simple feasibility problem in the sense of [25] is

given by

$$\begin{aligned} \min_{x_R, y, \lambda, w} \quad & \hat{c}(y, \lambda, w) \\ \text{s.t.} \quad & (x_I^p, x_R, y) \in P, \\ & G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\ & w_{jr} = s_{jr}^p \sum_{i=1}^{m_l} c_{ij} \lambda_i, \quad j \in I, \ r \in [\bar{r}_j], \end{aligned} \tag{F^p}$$

whose objective value is strictly greater than zero, since otherwise the subproblem (S^p) would be feasible. For a solution of (F^p), we obtain the following lemma by adapting Lemma 1 of [25].

Lemma 4 *Let z^p be a solution of the master problem (M^p), let the subproblem (S^p) be infeasible, and let $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$ be a solution of the feasibility problem (F^p). Then, $\hat{c}(\bar{y}^p, \bar{\lambda}^p, \bar{w}^p) > 0$ and every $z = (x_I^p, x_R, y, \lambda, w, s^p) \in \mathcal{M}^p$ is infeasible for the constraint*

$$\bar{c}(\bar{y}^p; y, \lambda, w) \leq 0. \tag{18}$$

Proof Consider a fixed x_I^p and assume (S^p) to be infeasible, which means that (F^p) has an optimal solution $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$ with $\hat{c}(\bar{y}^p, \bar{\lambda}^p, \bar{w}^p) > 0$. For the ease of presentation, we again use the abbreviation $v = (y, \lambda, w)$ and we rewrite the linear constraint set of (F^p) to obtain

$$\min_{x_R, v} \quad \hat{c}(v) \quad \text{s.t.} \quad \tilde{A} x_R + \tilde{B} v \leq \tilde{b}, \ \tilde{D} v = \tilde{d}. \tag{19}$$

Problem (F^p), and hence Problem (19), minimizes a convex function over affine-linear constraints. Thus, $(\bar{x}_R^p, \bar{v}^p)$ fulfills the KKT conditions of Problem (19), i.e., primal feasibility, stationarity, non-negativity of multipliers α of inequality constraints, and complementarity. With δ denoting the multipliers of the equality constraints, the KKT conditions read

$$\tilde{A} \bar{x}_R^p + \tilde{B} \bar{v}^p \leq \tilde{b}, \ \tilde{D} \bar{v}^p = \tilde{d}, \tag{20a}$$

$$\tilde{A}^\top \alpha = 0, \tag{20b}$$

$$-\tilde{B}^\top \alpha - \tilde{D}^\top \delta = \nabla_v \hat{c}(\bar{v}^p), \tag{20c}$$

$$\alpha \geq 0, \tag{20d}$$

$$\alpha^\top \tilde{A} \bar{x}_R^p + \alpha^\top \tilde{B} \bar{v}^p = \alpha^\top \tilde{b}. \tag{20e}$$

We recap that $\bar{c}$ is derived from the first-order Taylor approximation, i.e., it holds

$$\bar{c}(\bar{y}^p; v) = \hat{c}(\bar{v}^p) + \nabla_v \hat{c}(\bar{v}^p)^\top (v - \bar{v}^p).$$

This can be expanded to

Algorithm 1 Multi-tree Outer Approximation for MIQP-QP Bilevel Problems.

Initialize $\phi = -\infty$, $\Phi = \infty$, and $p = 0$.
while $\phi < \Phi$ **do**
  Solve the master problem (M^p).
  if (M^p) is infeasible **then**
    Return "The bilevel problem is infeasible".
  else
    Let z^p be the optimal solution of (M^p) and set $\phi = q_u(x^p, y^p)$.
  end if
  Solve the subproblem (S^p), or the feasibility problem (F^p) if (S^p) is infeasible, and obtain $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$.
  if (S^p) is feasible and $q_u(x_I^p, \bar{x}_R^p, \bar{y}^p) < \Phi$ **then**
    Set $z^* = (x_I^p, \bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p, s^p)$ and $\Phi = q_u(x_I^p, \bar{x}_R^p, \bar{y}^p)$.
  end if
  Add the outer approximation cut $\bar{c}(\bar{y}^p; y, \lambda, w) \leq 0$ to (M^p).
  Set $p \leftarrow p + 1$.
end while
Return z^*.

$$\bar{c}(\bar{y}^p; v) = \hat{c}(\bar{v}^p) - (\tilde{B}^\top \alpha + \tilde{D}^\top \delta)^\top (v - \bar{v}^p) \tag{21a}$$

$$= \hat{c}(\bar{v}^p) - \alpha^\top \tilde{B} v + \alpha^\top \tilde{B} \bar{v}^p - \delta^\top (\tilde{D} v - \tilde{D} \bar{v}^p) \tag{21b}$$

$$= \hat{c}(\bar{v}^p) - \alpha^\top (\tilde{B} v - \tilde{b}) - \delta^\top (\tilde{D} v - \tilde{d}), \tag{21c}$$

by using KKT stationarity (20c) (for (21a)) and re-ordering the terms (for (21b)). Further, we replaced $\alpha^\top \tilde{B} \bar{v}^p$ by $\alpha^\top \tilde{b}$ according to (20e) and (20b) and $\tilde{D} \bar{v}^p$ by $\tilde{d}$ according to (20a) to obtain (21c). Now, let $z = (x_I^p, x_R, v, s^p) \in \mathcal{M}^p$. This implies that v must be feasible for (F^p), respectively (19). In particular, we know that $\tilde{B} v - \tilde{b} \leq -\tilde{A} x_R$ and $\tilde{D} v - \tilde{d} = 0$. Applying this to (21) and using (20b) yields

$$\bar{c}(\bar{y}^p; v) \geq \hat{c}(\bar{v}^p) + \alpha^\top \tilde{A} x_R = \hat{c}(\bar{v}^p) > 0.$$

Thus, v violates (18). □

We recap that, with the last lemma, we can derive cutting planes that cut off x_I^p if the subproblem (S^p) is infeasible. This is another key property for the finite termination of our approach. With this result, we are ready to present the multi-tree outer approximation in Algorithm 1.

In every iteration p, Algorithm 1 first solves the master problem (M^p) to obtain a solution z^p. According to Lemma 2, $\mathcal{M}^p \subseteq \mathcal{M}^{p-1}$ and we can update the lower bound ϕ by $q_u(x_I^p, x_R^p, y^p)$. Next, either the subproblem (S^p) or, in case of infeasibility, the feasibility problem (F^p) is solved. If the subproblem is feasible with solution $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$, then the point $(x_I^p, \bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p, s^p)$ is feasible for the convex single-level reformulation (15) and the upper bound is updated if $q_u(x_I^p, \bar{x}_R^p, \bar{y}^p) < \Phi$. In this case, also z^* is updated. We terminate when $\phi \geq \Phi$ is achieved and return the best solution z^*. We now show the correctness of this approach.

Theorem 1 *Algorithm 1 terminates after a finite number of iterations at an optimal solution of Problem (1) or with an indication that the problem is infeasible.*

The following proof is adapted from [25].

Proof First of all, note that all master problems are bounded since P is assumed to be bounded. We next show finiteness of Algorithm 1. If Problem (1) is feasible, then we can follow from Lemma 3 that at most one integer solution is visited twice. Whenever an integer solution is visited for a second time, $\phi \geq \Phi$ holds and the algorithm terminates. On the other hand, if Problem (1) is infeasible, then every subproblem (S^p) is infeasible. According to Lemma 4, the integer solution x_I^p is infeasible for the master problem in iteration $p+1$, which results in an infeasible master problem after a finite number of iterations. Thus, finiteness follows from the finite number of integer solutions for Problem (1).

Second, we show that Algorithm 1 always terminates at a solution of Problem (1), if it is feasible. We denote a (possibly non-unique) optimal solution of Problem (1) by (x_I^*, x_R^*, y^*) with objective function value $q_u^* = q_u(x_I^*, x_R^*, y^*)$. Now assume that Algorithm 1 terminates with a solution $(x_I', x_R', y', \lambda', w', s')$ and objective function value $\Phi = q_u(x_I', x_R', y')$. It is obvious that $(x_I', x_R', y', \lambda')$ is feasible for the non-convex single-level reformulation (10). According to Lemma 1, (x_I', x_R', y') is then feasible for the original bilevel problem, which gives $q_u^* \leq \Phi = q_u(x_I', x_R', y')$. On the other hand, Lemma 2 together with Lemma 1 state that every master problem (M^p) is a relaxation of the original bilevel problem (1), i.e., $\phi \leq q_u^*$. Since Algorithm 1 only terminates when $\phi \geq \Phi$, we obtain $q_u(x_I^*, x_R^*, y^*) = q_u(x_I', x_R', y')$. □

In the remainder of this subsection we discuss some enhancements of Algorithm 1.

Additional outer-approximation cuts Since the outer-approximation cuts of the form of (16) are globally valid, we can add cuts also for points other than the solution of the subproblem. One point that comes for free is the solution z^p of the master problem, i.e., we can add a cut (16) for the point y^p. This is an outer-approximation cut in the sense of Kelley [36]. Further, we can add outer-approximation cuts for all feasible solutions that are encountered in the process of solving the master problem.

Early termination of the master problem It is sufficient for the correctness of the entire algorithm that the master problem provides a new integer-feasible point. Since the outer-approximation cuts are constructed in a way that already visited integer solutions of the previous iterations have an objective value worse or equal to the incumbent Φ, we can stop the master problem with the first improving integer-feasible solution, i.e., a solution that has an objective value better than the incumbent Φ; see also [25]. This strategy mimics the single-tree approach that is stated in Sect. 3.2.

Warmstarting the master problem Warmstarting mixed-integer problems can be a very effective strategy because it may produce a tight initial upper bound and help to keep the branch-and-bound trees small. Since the incumbent solution z^* is feasible for every master problem, it is reasonable to warmstart the master problems with this solution.

The performance of the plain Algorithm 1 as well as the effectiveness of the mentioned enhancements is evaluated in Sect. 4.

3.2 A single-tree outer-approximation approach

The multi-tree outer-approximation approach from Sect. 3.1 can also be cast into a single branch-and-bound tree. In the context of general convex MINLPs, this approach is known as LP/NLP-based branch-and-bound (LP/NLP-BB) and was first introduced in [47]. LP/NLP-BB avoids the time-consuming solution of subsequently updated mixed-integer master problems by branching on the integer variables of an initial master problem and solving continuous relaxations of the subsequently updated master problem at every branch-and-bound node. Whenever such a relaxation results in a new integer solution with a better objective value than the incumbent, the solution process is interrupted. In this event the convex nonlinear subproblem with fixed integer variables is solved and the master problem is updated by an outer-approximation cut derived from the solution of the subproblem. In this view, LP/NLP-BB can be interpreted as a branch-and-cut algorithm that requires the solution of NLPs to separate cuts. For additional implementation details we refer to [1,8].

If we apply such a single-tree approach to our setup, the initial master problem is given by Problem (M^p) for $p = 0$, i.e., this corresponds exactly to the initial multi-tree master problem. In this view, the problem that is solved at every branch-and-bound node is the continuous relaxation of Problem (M^p) with bounds $\ell, u \in \mathbb{R}^{|I|}$ on the integer variables, i.e., the QP:

$$
\begin{aligned}
\min_{x,y,\lambda,w,s} \quad & q_u(x, y) \\
\text{s.t.} \quad & Ax + By \geq a, \quad Cx_I + Dy \geq b, \\
& l \leq x_I \leq u, \\
& G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\
& \lambda^\top C x_I \text{ linearization: (11), (13)}, \\
& \bar{c}(\bar{y}^\ell; y, \lambda, w) \leq 0, \quad \ell = 0, \ldots, p-1.
\end{aligned}
\tag{$\mathrm{N}^p(l, u)$}
$$

The index p in ($\mathrm{N}^p(l, u)$) corresponds to the number of added strong-duality cuts. The specific values for l and u follow from branching. As opposed to the textbook LP/NLP-BB that solves an LP at every branch-and-bound node, we solve a QP. Thus, we rather perform a QP/NLP-BB.

We now state a tailored single-tree approach for bilevel problems of the form (1) in Algorithm 2. The rationale is the following. The algorithm subsequently solves QPs of the form ($\mathrm{N}^p(l, u)$), starting with the root-node problem for $p = 0$, $l = x^-$, and $u = x^+$. Whenever such a QP is infeasible or its objective function value can not improve the incumbent, then this problem can be removed from the set of open problems $\mathcal{O}$ once and for all. In case the solution of Problem ($\mathrm{N}^p(l, u)$) is integer feasible, then the corresponding subproblem (S^p)—or the feasibility problem (F^p) if the subproblem is infeasible—is solved. One key difference of LP/NLP-BB compared to a standard branch-and-bound is that this solved subproblem must not be pruned but needs to be updated with an appropriate outer-approximation cut. Moreover, all other open problems in $\mathcal{O}$ need to be updated as well. Finally, if Problem ($\mathrm{N}^p(l, u)$)

Algorithm 2 Single-Tree Outer Approximation for MIQP-QP Bilevel Problems.

1: Initialize $\Phi = \infty$, $p = 0$, $l = x^-$, $u = x^+$, and $z^* =$ **none**.
2: Initialize the set of open node problems $\mathcal{O}$:=$\{(\mathrm{N}^p(l, u))\}$
3: **while** $\mathcal{O} \neq \emptyset$ **do**
4: Remove a QP $(\mathrm{N}^p(l, u))$ from $\mathcal{O}$ and solve it to obtain a solution $z^{l,u}$.
5: **if** $(\mathrm{N}^p(l, u))$ is infeasible or $q_u(x^{l,u}, y^{l,u}) \geq \Phi$ **then**
6: Subtree can be pruned. Continue.
7: **else if** $z^{l,u}$ is integer feasible and $q_u(x^{l,u}, y^{l,u}) < \Phi$ **then**
8: Set $x_I^p = x_I^{l,u}$ and $s^p = s^{l,u}$ and solve the subproblem (S^p) or the feasibility problem (F^p) to obtain $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$.
9: **if** (S^p) is feasible and $q_u(x_I^p, \bar{x}_R^p, \bar{y}^p) < \Phi$ **then**
10: Set $z^* = (x_I^p, \bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p, s^p)$ and $\Phi = q_u(x_I^p, \bar{x}_R^p, \bar{y}^p)$.
11: **end if**
12: Re-add the problem: $\mathcal{O} \leftarrow \mathcal{O} \cup \{(\mathrm{N}^p(l, u))\}$.
13: Add the outer-approximation cut $\bar{c}(\bar{y}^p; y, \lambda, w) \leq 0$ to all problems in $\mathcal{O}$.
14: Set $p \leftarrow p + 1$.
15: **else**
16: Branch on a fractional x_i^p, $i \in I$, to obtain new bounds l^1, u^1 and l^2, u^2.
17: Update $\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathrm{M}^p(l^1, u^1), \mathrm{M}^p(l^2, u^2)\}$.
18: **end if**
19: **end while**
20: Return z^* or, if z^* is **none**, return "The bilevel problem is infeasible".

is feasible but the solution is not integer feasible, one branches on a fractional integer variable to obtain two new open problems. Similar to the multi-tree approach, only one integer assignment—the optimal one—is computed twice during the solution process. Finiteness follows again from the finite number of possible integer assignments. At some point all problems in the set $\mathcal{O}$ are infeasible, $\mathcal{O}$ is emptied, and the algorithm terminates. If all subproblems turned out to be infeasible, then z^* is never updated and the infeasibility of the bilevel problem is correctly detected. All together, we obtain the following correctness theorem.

Theorem 2 *Algorithm 2 terminates after a finite number of iterations with an optimal solution of Problem* (1) *or with an indication that the problem is infeasible.*

Since all arguments mainly follow from the proof of Theorem 1, we refrain from a formal proof of Theorem 2. We close this subsection by discussing some possible enhancements of Algorithm 2.

Additional outer-approximation cuts Similar to the multi-tree approach, we can enhance Algorithm 2 by adding outer-approximation cuts for all integer-feasible solutions $z^{l,u}$ with $q_u(x^{l,u}, y^{l,u}) \geq \Phi$, i.e., integer-feasible solutions that do not fulfill the if-condition in Line 7 in Algorithm 2. This can be done, e.g., by storing these non-improving integer-feasible solutions and adding outer-approximation cuts for these solutions together with the cut that is added in Line 13.

Advanced initialization Initializing the single-tree approach with a bilevel-feasible solution may be beneficial for various reasons. First, in the initial master problem (M^p) for $p = 0$, the constraints for the binary expansion (11) and for the linearization (13) are redundant, because they are not yet coupled by any outer-approximation cut. When the master problem is equipped with an initial

outer-approximation cut, however, then all parts of the model are coupled and the solver can effectively presolve the entire model before solving the root-node problem, i.e., ($\mathrm{N}^p(l, u)$) for $p = 0$, $l = x^-$, and $u = x^+$. In addition, this initial outer-approximation cut results in a tighter root-node problem.
Second, an initial bilevel-feasible solution can be used to pass an incumbent solution z^* to Algorithm 2 and to compute an initial upper bound Φ. This may allow to prune parts of the search tree right in the beginning. An initial bilevel-feasible point can be obtained, e.g., by finding a feasible or optimal point for Problem (M^p) for $p = 0$ and solving the corresponding subproblem. This mimics the first iteration of the multi-tree approach.

We evaluate the plain Algorithm 2 and these enhancements in Sect. 4.

3.3 Exploiting the bilevel structure

The two outer-approximation algorithms stated in the previous sections are an application of the approaches in [8,25] and [47], respectively. The effectiveness of both algorithms will depend, among other aspects, on the following properties:

(i) The ability to solve the master problem(s) effectively.
(ii) The number of integer-feasible solutions of the master problem that need to be evaluated.
(iii) The ability to solve the subproblems effectively.

Aspect (i) is addressed by the various enhancements stated in Sects. 3.1 and 3.2, respectively. For the latter two aspects, we can exploit the specific bilevel structure of Problem (1). We explain this on the example of the multi-tree method in Algorithm 1, but the same explanations hold for the single-tree approach in Algorithm 2 as well.

We first discuss aspect (ii). In general, the number of integer-feasible solutions of the master problem coincides with the number of subproblems that need to be solved. In the worst case, the algorithm needs to consider every integer-feasible solution of the initial master problem. However, for bilevel problems of the form (1), there is hope that one needs to evaluate only a few subproblems. The hypothesis is the following. Both the upper- and the lower-level objective functions are convex-quadratic in y and are to be minimized. This means that explicit min-max problems for which the upper level minimizes a function that the follower maximizes, cannot arise for quadratic bilevel problems of the form (1), unless all matrices H_u, G_u, and G_l are 0. Thus, the solution of the early master problems (that mainly abstract from lower-level optimality) might already be a good estimate of the optimal lower-level solution, depending on how competitive the two objective functions are. As a consequence, it might be quite likely that the first few solutions of the master problem already contain a close-to-optimal or even optimal integer upper-level decision x_I. Hence, the solution of the respective subproblem already provides a very tight upper bound Φ. This is, of course, instance-specific and we discuss this in more detail in Sect. 4.6.

We now turn to aspect (iii). Both Algorithm 1 and Algorithm 2 require the solution of the subproblem (S^p). It is easy to see that the variables w can be eliminated in this convex QCQP. The resulting problem is then equivalent to fixing $x_I = x_I^p$ directly

in the original nonconvex MINLP (10). Note that for fixed integer variables, (10) is a convex QCQP as well. While the full subproblem (S^p) is more in line with the standard literature on outer approximation for convex MINLPs and makes it easier to proof correctness, it is better to use Problem (10) with fixed integer variables in the actual implementation for the following reasons. First, Problem (10) is smaller than (S^p). Second, and more importantly, Problem (10) does not contain any big-M. As already discussed in Sect. 2, a wrong big-M can result in terminating with points that are actually bilevel-infeasible or suboptimal. When using Problem (10) as the subproblem, the former case can never appear. This is an huge advantage compared to solving the two single-level reformulations (2) and (15) directly, which may indeed terminate with points that are actually bilevel-infeasible.

Further, we can replace the subproblem (S^p) (respectively Problem (10) with fixed integers) by two easier problems. For the parametric lower-level problem (3), the optimal value function is given by

$$q_l^*(x_I) := \min_y \{q_l(y) \colon y \in P_l(x_I)\}. \tag{22}$$

It is well known that the bilevel problem (1) can be reformulated as an equivalent single-level problem using the optimal value function (22); see, e.g., [13, Chapter 5.6]:

$$\min_{x_I, x_R, y} \quad q_u(x_I, x_R, y) \tag{23a}$$

$$\text{s.t.} \quad (x_I, x_R, y) \in P, \tag{23b}$$

$$q_l(y) \leq q_l^*(x_I). \tag{23c}$$

It thus follows directly from Lemma 1 that Problem (15) and Problem (23) are equivalent in the following sense.

Lemma 5 *The feasible set of Problem* (15) *projected on the* (x, y)*-space coincides with the feasible set of Problem* (23)*. In addition, for every global optimal solution* $z^* = (x^*, y^*, \lambda^*, w^*, s^*)$ *of Problem* (15)*,* (x^*, y^*) *is a global optimal solution for Problem* (23) *and every global optimal solution* (x^*, y^*) *of Problem* (23) *can be extended to a global optimal solution* z^* *of Problem* (15)*.*

This enables to solve the subproblem in a bilevel-specific way as follows.

Remark 1 We can replace Step 9 of Algorithm 1 (or Step 8 of Algorithm 2) by first solving the parametric lower-level problem (3) with fixed integer linking variables $x_I = x_I^p$ to obtain a (possibly ambiguous) lower-level solution $\tilde{y}^p$ and the corresponding objective function value $q_l(\tilde{y}^p) = q_l^*(x_I^p)$. Then, we solve Problem (23) with fixed $x_I = x_I^p$, which is a convex QCQP, to obtain an optimistic bilevel solution $(x_I^p, \bar{x}_R^p, \bar{y}^p)$. In other words, instead of solving subproblem (S^p), we can solve a convex QP and a convex QCQP that is considerably smaller than (S^p).

In case the lower-level problem has a unique solution, Remark 1 can be strengthened.

Remark 2 If the matrix G_l is positive definite, then the lower-level problem has a unique solution and $M(\bar{x}_I)$ is a singleton. In this case we can replace Step 9 of Algorithm 1 (or Step 8 of Algorithm 2) by subsequently solving the parametric lower-level QP (3) with fixed integer upper-level variables x_I^p to obtain the unique lower-level solution $\bar{y}^p$, and solving the upper-level problem

$$\min_{x_R} \quad q_u(x_I^p, x_R, \bar{y}^p) \quad \text{s.t.} \quad A \begin{pmatrix} x_I^p \\ x_R \end{pmatrix} + B\bar{y}^p \geq a, \tag{24}$$

in which all integer upper-level variables are fixed to x_I^p and all lower-level variables are fixed to the unique solution $\bar{y}^p$. Problem (24) is a convex QP as well.

With this remark, the large QCQP (S^p) can be replaced by two considerably easier QPs. We discuss the effectiveness of Remarks 1 and 2 in Sect. 4.

4 Computational study

In this section, we provide detailed numerical results for the methods proposed in the previous sections. Besides mean and median running times and counts of solved subproblems, the evaluations and comparisons rely on performance profiles according to [17]. For every test instance i we compute ratios $r_{i,s} = t_{i,s}/\min\{t_{i,s} : s \in S\}$, where S is the set of the solution approaches and $t_{i,s}$ is the running time of a solver s for instance i, given in wall-clock seconds. Each performance profile in this section shows the percentage of instances (y-axis) for which the performance ratio $r_{i,s}$ of approach s is within a factor $\tau \geq 1$ (log-scaled x-axis) of the best possible ratio. Before we go into the details, we provide some information on the computational setup in Sect. 4.1. We then specify the test set that we use throughout the study in Sect. 4.2. We also compare the two benchmark approaches of solving the KKT-based reformulation (2) and the strong-duality-based reformulation (15) in this section. In Sects. 4.3 and 4.4, we evaluate the results for different variants of the multi- and the single-tree approach, respectively. In Sect. 4.5, we compare both methods and also test their performance against the benchmark. Finally, we evaluate the impact of different modifications of the test set on running times in Sect. 4.6.

4.1 Computational setup

We implemented all solution approaches using C++-11 and used GCC 7.3.0 as the compiler. All optimization problems, i.e., all convex (MI)(QC)QP problems, are solved by Gurobi 9.0.1 using its C interface.[1] The single-tree approach of Sect. 3.2 is realized using lazy constraint callbacks of Gurobi that are invoked whenever a new integer-feasible solution with an objective value better than the bilevel incumbent is found. Note however that using Gurobi's lazy constraint callbacks requires to set the parameter LazyConstraints to 1, which avoids certain reductions and transformations during

[1] Since Gurobi's C interface offers more flexibility compared to the C++ interface, we wrote a C++ wrapper around Gurobi's C interface.

the presolve that are incompatible with lazy constraints. For all solution approaches we set the NumericFocus parameter to a value of 3, which results in increased numerical accuracy. Further, we tightened Gurobi's integer feasibility tolerance from its default value 10^{-5} to 10^{-9} throughout all computations. The rationale is to prevent numerical inaccuracies caused by products of binary variables and big-M values, e.g., in the Constraints (13). All big-M values are fixed to 10^5. For each solution attempt, we set a time limit of 3600s. The computational experiments have been executed on a compute cluster using compute nodes with Xeon E3-1240 v6 CPUs with 4 cores, 3.7 GHz, and 32 GB RAM; see [49] for more details.

4.2 Selection of the test set and evaluation of the benchmarks

Our test set is based on a subset of the MIQP-QP test set used in [38], but is extended by the additional instance classes DENEGRE, INTER-ASSIG, and INTER-KP, which turned out to be too easy for the local optimality considerations in [38]. On the other hand, some of the instance classes used in [38] are too hard to be solved to global optimality, i.e., for most instances of the respective instance class, every tested solver exceeds the time limit. For this reason, we excluded the instance classes GENERALIZED, GK, KP, MIPLIB2010, MIPLIB2017, and OR from our test set. All instances used in this paper are based on MILP-MILP instances from the literature; see the "Ref" column in Table 1 for a reference to the original MILP-MILP test set. The MIQP-QP instances were generated by relaxing all integrality conditions in the lower-level problem and by enforcing continuous linking variables to be integer. Further, we added quadratic terms to both objective functions. To this end, we randomly generated quadratic matrices Q, R, and S of suitable sizes and entries in $[-\sqrt[4]{\sigma}, \sqrt[4]{\sigma}]$ with $\sigma = \max\{\|c_u\|_\infty, \|d_u\|_\infty\}$ (or $\sigma = \|d_l\|_\infty$). We then set $H_u = Q^\top Q$, $G_u = R^\top R$, and $G_l = S^\top S + D$, where D is a diagonal matrix with entries in $[1, \sqrt[4]{\|d_l\|_\infty}]$. This approach renders H_u and G_u positive semidefinite and G_l positive definite.[2] This allows to evaluate both Remarks 1 and 2 on the full test set. Note that for the sake of completeness, we also provide results for semidefinite matrices G_l in Sect. 4.6. The full test set $\mathcal{I}^{\text{full}}$ contains 757 instances, which is—to the best of our knowledge—the largest test set of MIQP-QP bilevel problems considered for approaches that compute global optimal solutions of these models. Before we thin out this test set to compare the various methods on a more balanced set, we first compare the two benchmark approaches introduced in Sect. 2 on the full test set $\mathcal{I}^{\text{full}}$.

We therefore briefly recap the two methods. The first approach (KKT-MIQP) solves a reformulated and linearized single-level problem, in which the lower level is replaced by its KKT conditions. The KKT complementarity conditions are linearized with a big-M formulation. This yields the convex MIQP (2), which can be solved directly using solvers such as Gurobi or CPLEX. This approach is very popular and widely used in applied bilevel optimization. Similarly, using strong duality of the lower level and convexifying the strong-duality inequality yields the convex MIQCQP (15). We call

[2] The MATLAB function that we implemented to generate the matrices and a brief documentation thereof can be found in the GitHub repository under https://github.com/m-schmidt-math-opt/qp-bilevel-matrix-generator.

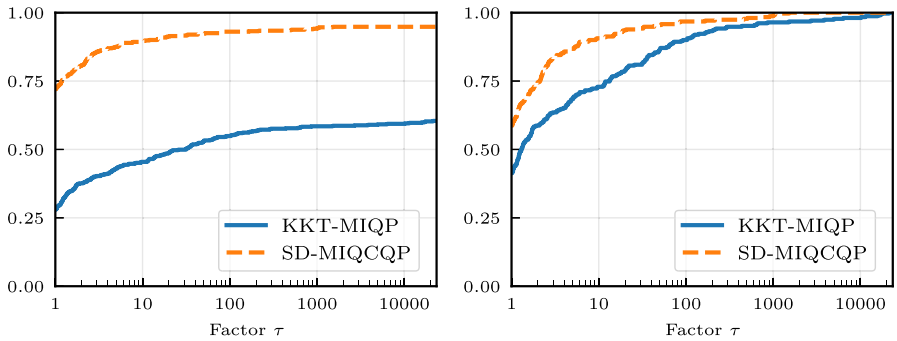

Fig. 1 Log-scaled performance profiles for the benchmark approaches KKT-MIQP and SD-MIQCQP on those instances in $\mathcal{I}^{\text{full}}$ that at least one of the two approaches solves (left) and on those instances in $\mathcal{I}^{\text{full}}$ that both approaches solve (right)

this approach SD-MIQCQP in the following. Instead of applying outer-approximation algorithms to this problem as proposed in Sect. 3, one can also solve this problem directly. Solvers like Gurobi then either apply a linear outer approximation or solve a continuous QCP relaxation at every branch-and-bound node. The exact method can be set via the MIQCPMethod parameter that we left at its default value -1. This setting automatically chooses the best strategy. In rare cases (5 instances) this resulted in unsolved node relaxations and thus a suboptimal termination. We count these instances as unsolved by SD-MIQCQP. We also emphasize again, that KKT-MIQP and SD-MIQCQP make use of a big-M value. This may result in bilevel-infeasible "solutions", which is not the case for the proposed outer-approximation algorithms. Thus, we implemented an ex-post sanity check that computes the relative strong-duality error of the lower level for a given solution (x, y, λ):

$$\chi(x, y, \lambda) := c(x_I, y, \lambda)/|q_l(y)|.$$

Whenever the error $\chi(x, y, \lambda)$ exceeds the tolerance of 10^{-4}, we consider the instance as unsolved for the respective solver. This never occurred for KKT-MIQP but happened in 19 cases for SD-MIQCQP.

In Fig. 1 (left) we compare the running times of KKT-MIQP and SD-MIQCQP on those 561 instances in $\mathcal{I}^{\text{full}}$ that at least one of the two benchmark approaches solves. It is obvious that SD-MIQCQP is the better and more reliable approach. It solves around 95% of the instances of the subset and is the faster method for most of these instances. In contrast, KKT-MIQP is only capable of solving around 60% of the instances. Figure 1 (right) compares the running times of KKT-MIQP and SD-MIQCQP on the subset of 310 instances that both benchmark approaches solve. Also on this subset, SD-MIQCQP is the dominating approach. This is interesting, since the KKT reformulation is the most used approach in applied bilevel optimization. Note that these results are in line with the study in [55], which reveals that a strong-duality-based reformulation outperforms a KKT-based reformulation for mixed-integer linear bilevel problems with integer linking variables and considerably large lower-level problems. Due to this clear dominance, we exclude KKT-MIQP from our further considerations.

Table 1 Details of the test set $\mathcal{I}$ per instance class

			n_x		n_y		$\|I\|$		$\max_{i \in I} x_i^+$		m_u		m_l	
	References	Size	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
CLIQUE	[24]	55	546	1593	586	1653	546	1593	1	1	1	1	1326	3363
DENEGRE	[16]	25	5	15	5	15	5	15	1500	1500	0	0	20	20
IMKP	[22]	68	15	105	15	105	15	105	1	1	1	29	16	106
INT0SUM	[48]	6	110	360	110	360	110	360	10	10	44	144	44	144
INTER-ASSIG	[16]	10	25	25	25	25	25	25	1	1	1	1	45	45
INTER-CLIQUE	[51]	14	19	94	27	109	19	94	1	1	1	1	54	280
INTER-FIRE	[4]	51	22	79	44	158	22	79	1	1	1	1	41	974
INTER-KP	[16]	47	20	40	20	40	20	40	1	1	1	1	21	41
MIPLIB	[22]	8	9	90	3	253	9	90	1	1	0	0	15	331
XULARGE	[54]	49	500	1000	500	1000	500	1000	10	10	200	400	200	400
XUWANG	[21]	90	10	460	10	460	10	460	10	10	4	184	4	184

We now thin out the test set $\mathcal{I}^{\text{full}}$ to obtain a more balanced set for the outer-approximation methods that we evaluate in the following and for SD-MIQCQP. We remove 7 instances that exceed the memory limit for all above-mentioned approaches and 1 instance that is proven to be infeasible by all approaches. Further, we exclude 177 instances that are too easy, i.e., that all approaches solve within 1s. In addition, we also exclude 149 instances that cannot be solved to global optimality by any of the approaches within the time limit of 3600s. The resulting final test set $\mathcal{I}$ contains $|\mathcal{I}| = 423$ instances with up to several thousand variables and constraints. Note that we checked that the objective function values and best bounds provided by different approaches are consistent for each instance. This is not guaranteed due to possibly wrong big-M values. All instances in $\mathcal{I}$ passed this ex-post optimality check. More details on the instances in $\mathcal{I}$ can be found in Table 1.

Besides the resulting size of each instance class ("Size"), we also specify the minimum and maximum number of upper-level and lower-level variables (n_x, n_y) and constraints (m_u, m_l), as well as the minimum and maximum number of linking variables ($|I|$) and of the maximum upper bound of the linking variables ($\max_{i \in I} x_i^+$). The densities of the objective function matrices H_u, G_u, and G_l of the instances in $\mathcal{I}$ are displayed in Fig. 2.

Finally, we mention that we later also analyze the performance of the various methods on the instance set $\mathcal{I}^{\text{hard}}$ that contains those 149 instances that none of the tested approaches can solve within the time limit.

4.3 Evaluation of the multi-tree approach

We now evaluate the following different parameterizations of the multi-tree approach as described in Sect. 3.1:

MT A basic variant without any enhancements, i.e., the plain Algorithm 1.

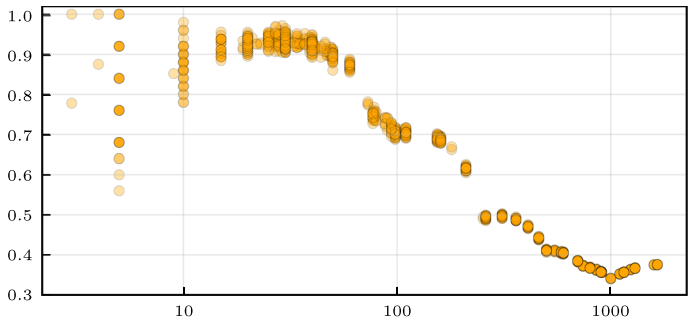

Fig. 2 Sizes n_x resp. n_y (x-axis) and densities (y-axis) of the matrices $H_u \in \mathbb{R}^{n_x \times n_x}$, $G_u \in \mathbb{R}^{n_y \times n_y}$, and $G_l \in \mathbb{R}^{n_y \times n_y}$ for the instances in set $\mathcal{I}$

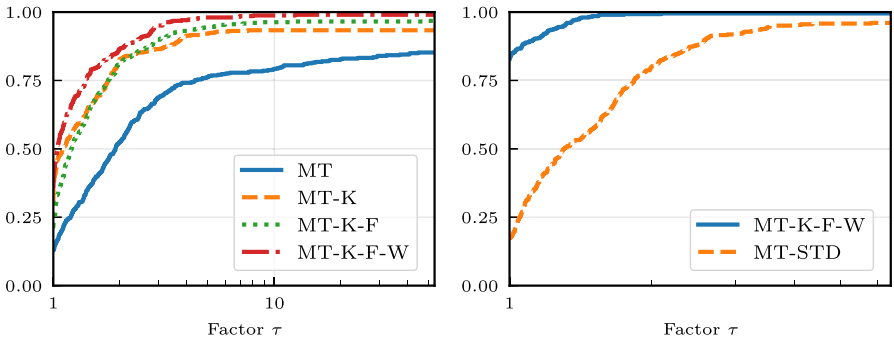

Fig. 3 Log-scaled performance profiles for different variants of the multi-tree approach that use Remark 2 on the instances in $\mathcal{I}$ that at least one multi-tree approach solves (left) and for the best multi-tree approach that uses Remark 2 compared to when the standard subproblem (S^p) is used on the instances in $\mathcal{I}$ that at least one of the two approaches solves (right)

MT-K Like MT but additional Kelley-type cutting planes ("K") are used.

MT-K-F The master problem terminates as soon as a first ("F") improving integer-feasible solution is found and after every iteration additional Kelley-type cutting planes are added for every non-improving integer-feasible solution found by the master problem.

MT-K-F-W Like MT-K-F but every master problem is warmstarted ("W") using the best available bilevel-feasible solution.

Since G_l is positive definite for the test set $\mathcal{I}$, we use Remark 2 as the default method for solving the subproblem, which dominates the method proposed in Remark 1. Later, we also compare the best setting among these four variants with an equivalent setting that uses the standard subproblem (S^p), i.e., Problem (10) with fixed integer variables; see the discussion in Sect. 3.3. Note that we assess Remark 1 on instances with positive semidefinite matrices G_l—an instance class, for which Remark 2 is not applicable—separately in Sect. 4.6.

Figure 3 (left) shows the performance profile of the four variants on the 406 instances in $\mathcal{I}$ that at least one of the four methods can solve.

Table 2 Running times (in sec), number of solved subproblems, and time spent in the subproblems for all tested multi-tree variants on the subset of instances in $\mathcal{I}$ that every multi-tree approach solves

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT	342.50	34.74	21.43	6.00	3.33	0.16
MT-K	186.15	17.88	5.85	4.00	1.63	0.08
MT-K-F	201.99	18.48	19.37	10.00	4.22	0.16
MT-K-F-W	193.75	19.18	20.34	11.00	4.20	0.15
MT-STD	347.93	23.74	20.24	11.00	152.10	3.16

It turns out that MT-K clearly outperforms MT, which means that adding Kelley-type cutting planes improves the performance. In addition, MT-K-F dominates MT-K in terms of reliability, i.e., it solves more instances. MT-K-F in turn is dominated by MT-K-F-W, which obviously outperforms all other tested variants. In fact, MT-K-F-W is the fastest method for almost 50% of the instances. Further, it is the most reliable approach and solves almost every instance that any of the multi-tree approaches solves. Overall, according to Fig. 3 (left), MT-K-F-W is the winner among the four tested variants. It is noteworthy, however, that the performance profiles suggest that the difference in MT-K, MT-K-F, and MT-K-F-W mostly lies in the reliability of the approaches, i.e., in the number of solved instances.

In Fig. 3 (right), we compare the "winner setting" MT-K-F-W with a variant with the same settings than the former approach but that uses the "standard outer-approximation" subproblem (S^p) instead of the bilevel-tailored strategy proposed in Remark 2. We label the latter approach by MT-STD as an abbreviation for MT-K-F-W-STD. Note that the underlying instance set covers those 404 instances in $\mathcal{I}$ that can be solved by at least one of the two methods. The performance profile shows that MT-STD is clearly dominated by MT-K-F-W, which is the faster method for around 85% of the instances. This highlights the usefulness of applying Remark 2 to solve the subproblem in a bilevel-tailored way.

The conclusions drawn from the performance profiles are underlined by the mean and median running times displayed in Table 2. Note that, in order to have a fair comparison, we used for the computation of the numbers in Table 2 the 339 instances in $\mathcal{I}$ that every multi-tree solver can solve. It can be seen that MT-K, MT-K-F, and MT-K-F-W have considerably shorter mean and median running times than MT. The differences across the former three approaches are however negligible. This supports the conclusion drawn from the performance profiles in Fig. 3 (left): The algorithmic enhancements mainly yield an increased number of solved instances but, on the instances that all approaches can solve, no significant differences can be observed. In fact, MT-K-F-W neither has the shortest mean running time nor the shortest median running time. Table 2 also shows the mean and median number of solved subproblems (or feasibility problems), which corresponds to the number of evaluated integer-feasible solutions (or to the number of iterations minus 1, since per construction, the last iteration computes a specific integer solution for the second time). The table reveals that adding Kelley-type outer-approximation cuts reduces the number of solved

subproblems by almost 75% in the mean (MT-K vs. MT). On the other hand, terminating the master problem early increases the number of solved subproblems again by almost 300% in the mean (MT-K-F vs. MT-K). This is expected and these additional iterations are obviously overcompensated by a reduction in the running time per iteration, i.e., the master problem terminates much faster. Note that the number of solved subproblems is more or less identical for MT-K-F-W and MT-STD. This is expected, since apart from the solution routine for the subproblem, the algorithmic setting is the same for these two approaches. However, the time spent for solving the subproblems drastically increases, if the standard subproblem is used. The mean and median times spent in the subproblems clearly justify the bilevel-tailored solution of the subproblems as proposed in Remark 2.

4.4 Evaluation of the single-tree approach

We now analyze the single-tree approach described in Sect. 3.2 in the following variants:

ST	A basic variant without any enhancements as stated in Algorithm 2.
ST-K	Additional Kelley-type cutting planes ("K") are added for every non-improving integer-feasible solution found.
ST-K-C	Like ST-K but an initial bilevel-feasible solution is computed (if available) to add an initial outer-approximation cut ("C").
ST-K-C-S	Like ST-K-C but the initial bilevel-feasible solution is used to set start values ("S") for z^* and Φ.

Again, all these variants apply Remark 2 to solve the subproblems. Later, we compare the winner setting to an equivalent setting that uses subproblem (S^p) instead, i.e., Problem (10) with fixed integer variables; see the discussion in Sect. 3.3.

We compare the running times of the four variants using the performance profiles in Fig. 4 (left) on those 409 instances in $\mathcal{I}$ that can be solved by at least one of the four approaches.

The first observation is that the plots for ST and ST-K almost match. This means that additional Kelley-type cutting planes for all non-improving integer-feasible solutions do not make a significant difference, which is in contrast to the results for the multi-tree approach. One explanation might be that the performance boost that is observed in the multi-tree method is mainly due to more effective presolving of the individual master problems when additional cuts are added. This is, of course, not possible in the single-tree approach. On the contrary, adding an initial outer-approximation cut is very beneficial. The methods without this initial cut, ST and ST-K, are clearly dominated by ST-K-C. The latter is in turn slightly dominated by ST-K-C-S, the variant that also sets starting values according to the initial bilevel-feasible solution. ST-K-C-S is the fastest method for around 50% of the instances and it solves more instances than any other single-tree variant. Thus, ST-K-C-S is the "winner setting" for the single-tree approach.

In Fig. 4 (right), we compare this winner setting to a variant with the same settings but that uses the "standard outer-approximation" subproblem (S^p). We use the label ST-STD as an abbreviation for ST-K-C-S-STD for this variant. The underlying instance

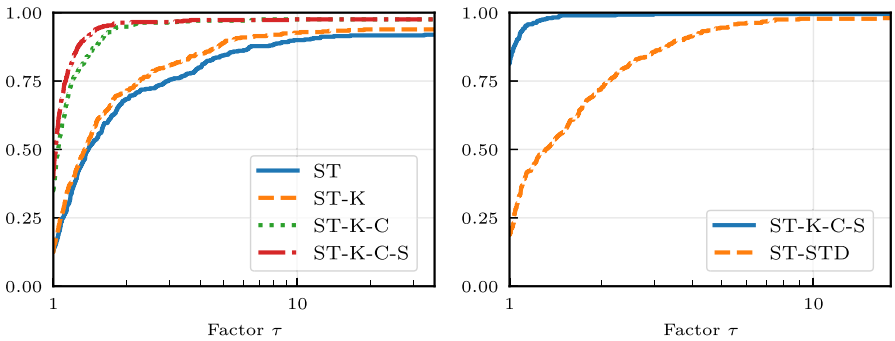

Fig. 4 Log-scaled performance profiles for different variants of the single-tree approach that use Remark 2 on the instances in $\mathcal{I}$ that at least one single-tree approach solves (left) and for the best single-tree approach that uses Remark 2 compared to when the standard subproblem (S^p) is used on the instances in $\mathcal{I}$ that at least one of the two approaches solves (right)

Table 3 Running times (in sec), number of solved subproblems, and time spent in the subproblems for all tested single-tree variants on the subset of instances in $\mathcal{I}$ that every single-tree approach solves

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
ST	301.14	13.31	105.26	18.00	5.72	1.33
ST-K	263.54	12.08	46.92	14.00	4.73	0.79
ST-K-C	188.69	10.49	45.99	14.00	4.33	0.66
ST-K-C-S	168.85	9.15	46.84	14.00	3.91	0.67
ST-STD	270.12	15.18	47.05	14.00	98.81	2.33

set consists of the 401 instances in $\mathcal{I}$ that can be solved by ST-K-C-S or ST-STD. The performance profile shows that ST-K-C-S is the faster method for around 80% of the instances and it also solves more instances than ST-STD. This suggests, that it clearly makes sense to solve the subproblem in a bilevel-tailored way.

The conclusions drawn from the performance profiles in Fig. 4 are also visible in Table 3, which displays statistics on running times and on the number of solved subproblems. The instances underlying the analysis in Table 3 consist of the 359 instances in $\mathcal{I}$ that all single-tree variants (including ST-STD) solve. Note that this renders a comparison of Tables 2 and 3 invalid, because the underlying instance sets are different. Looking at the running times in Table 3, we see that ST-K-C-S has the lowest mean running time, which is almost half of the mean running time of ST. Additionally, the median running times are dominated by ST-K-C-S, which supports the conclusion drawn from Fig. 4 (left): ST-K-C-S is the best parameterization of the single-tree approach. Looking at the number of solved subproblems, it is interesting to see that the additional Kelley-type cutting planes decrease the mean number of solved subproblems significantly, although in terms of running times only a slightly positive effect can be observed (ST-K vs. ST). This may indicate that there is a handful of instances that require to solve many subproblems when no additional Kelley-type cutting planes are

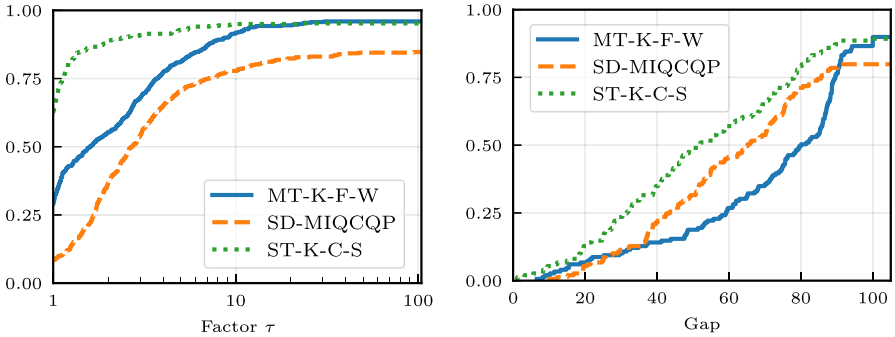

Fig. 5 Log-scaled performance profile for the winner configurations of the multi-tree and single-tree approaches compared to the benchmark SD-MIQCQP on the instances in $\mathcal{I}$ that at least one of the three methods solves (left) and an ECDF plot of the gaps on the instances in $\mathcal{I}^{\text{hard}}$ (right)

Table 4 Running times (in sec), number of solved subproblems, and time spent in the subproblems for the best outer-approximation approaches and the benchmark on the subset of instances in $\mathcal{I}$ that all the considered approaches solve

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	233.26	31.79	47.09	14.00	2.54	0.20
SD-MIQCQP	403.73	25.59	–	–	–	–
ST-K-C-S	189.20	11.04	55.90	16.00	4.55	0.80

added. In contrast, the other two algorithmic enhancements show hardly any effect on the number of solved subproblems (ST-K-C and ST-K-C-S vs. ST-K). Since ST-STD is set up in the same way as ST-K-C-S, it is clear that the mean and median numbers of solved subproblems are similar. However, the time spent for the subproblems significantly increases when (S^p) is used (ST-STD vs. ST-K-C-S), such that ST-STD has significantly longer running times in the mean and median. Again, this justifies the bilevel-tailored solution of the subproblems as proposed in Remark 2.

4.5 Comparison of the multi- and single-tree approaches with the benchmark

We now compare the best parameterizations of the multi- and single-tree approach (MT-K-F-W and ST-K-C-S) with the benchmark approach SD-MIQCQP. Figure 5 (left) shows performance profiles of the running times for those 419 instances that at least one of the three methods solves. Obviously, both outer-approximation approaches dominate the benchmark SD-MIQCQP. They are more reliable and solve around 95% of the 419 instances compared to around 85% solved by SD-MIQCQP. The outer-approximation methods are also the faster methods. In particular, ST-K-C-S is the fastest approach for more than 60% of the instances and is the dominating approach according to the performance profiles.

We analyze this in more detail by looking at mean and median running times displayed in Table 4. This table is based on those 338 instances in $\mathcal{I}$ that all three approaches can solve. Restricted to these instances, SD-MIQCQP is a factor of around 1.7 slower compared to MT-K-F-W in the mean, but it is slightly the faster method in the median. However, as seen in Fig. 5 (left), MT-K-F-W is the more reliable method. In contrast, SD-MIQCQP is a factor of more than 2 slower in the mean and almost 2.5 in the median compared to ST-K-C-S—without taking the 55 instances into account that SD-MIQCQP cannot solve within the time limit but that are solved by ST-K-C-S. Compared to MT-K-F-W, the single-tree method is almost 3 times faster in the median, although it needs to solve more subproblems in the mean and median and thus significantly spends more time in the subproblems. The reason for that may simply lie in the nature of the methods: The single-tree approach needs to search only one branch-and-bound tree. Overall, the single-tree approach ST-K-C-S is the winner approach on the test set $\mathcal{I}$. For a more detailed evaluation of the three methods, we provide performance profiles and mean and median running times per instance class as well as tables with running times and gaps per instance in Appendix 1. The figures and tables therein underline the observations discussed in this subsection.

We also evaluate the performance of the three methods on the hard instances $\mathcal{I}^{\text{hard}}$ that none approach can solve within the time limit. Therefore, we show plots of the empirical cumulative distribution functions (ECDF) of the optimality gaps of each method obtained after the time limit in Fig. 5 (right). The x-axis shows the gap in percent while the y-axis shows the percentage of instances. The figure reveals that after 3600s, ST-K-C-S has the smallest optimality gap and is thus the preferable method also on the instances in $\mathcal{I}^{\text{hard}}$. In addition, the two outer-approximation variants are more robust in the sense that they provide gaps within 100% for more instances than SD-MIQCQP does. However, the differences between the three methods are not very pronounced.

The effectiveness of our methods is also underlined in that extent that the charm of solving the MIQCQP (15) directly, i.e., applying SD-MIQCQP, lies, among other things, in the exploitation of the numerical stability of modern solvers. This contains, e.g., an elaborate numerical polishing of the outer-approximation cuts and managing these cuts in cut pools—numerical details that we mostly abstracted from in our implementation. Incorporating such aspects in our implementation would certainly not harm our results, but it can be expected that a more elaborated implementation would lead to an even greater domination of our approaches compared to the benchmark approaches.

4.6 Sensitivity on specific test set properties

Finally, we analyze the performance of the outer-approximation algorithms MT-K-F-W and ST-K-C-S in comparison to the benchmark SD-MIQCQP under three different modifications of the test set $\mathcal{I}$.

First, we adapt the matrix G_l to be positive semidefinite instead of positive definite for every instance in $\mathcal{I}$. We label this adapted test set by $\mathcal{I}^{\text{psd}}$. Note that with this modification, all instances in $\mathcal{I}^{\text{psd}}$ may have ambiguous lower-level solutions

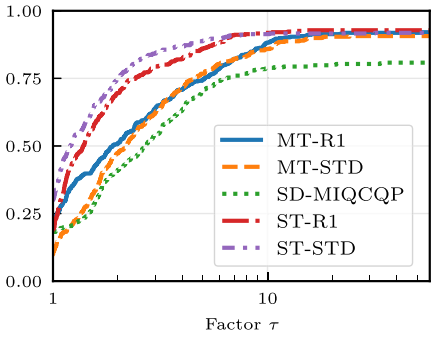

Fig. 6 Log-scaled performance profile for MT-R1, MT-STD, ST-R1, and ST-STD compared to the benchmark SD-MIQCQP on the instances in $\mathcal{I}^{\text{psd}}$ that at least one of the five approaches solves

Table 5 Running times (in sec), number of solved subproblems, and time spent in the subproblems for MT-R1, MT-STD, ST-R1, ST-STD, and the benchmark SD-MIQCQP on the subset of instances in $\mathcal{I}^{\text{psd}}$ that all five approaches solve

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-R1	223.70	30.49	37.40	13.00	20.29	2.52
MT-STD	282.87	29.24	40.34	12.00	75.09	1.42
SD-MIQCQP	340.82	23.26	–	–	–	–
ST-R1	224.14	14.36	47.38	13.00	31.65	4.20
ST-STD	229.02	12.03	46.63	13.00	81.58	2.04

such that Remark 2 is not applicable anymore. We thus equip the multi- and single-tree approaches with the subproblem routine according to Remark 1 and label these approaches by MT-R1 and ST-R1 as abbreviations for MT-K-F-W-R1 and ST-K-C-S-R1, respectively. Note that MT-STD and ST-STD, i.e., MT-K-F-W and ST-K-C-S equipped with the standard subproblem, as well as the benchmark approach SD-MIQCQP, are also applicable for the instance set $\mathcal{I}^{\text{psd}}$. We thus compare these five methods on the 417 instances in $\mathcal{I}^{\text{psd}}$ that at least one of the five methods solves using the performance profiles shown in Fig. 6.

It can be seen that the single-tree methods are still the dominating ones among the tested approaches, both in terms of running times and also in terms of reliability. Further, the multi-tree approaches still dominate the benchmark SD-MIQCQP. Thus, the outer-approximation methods outperform the benchmark, although not as pronounced as for the standard test set $\mathcal{I}$. The maximum factor τ is approximately 14 in Fig. 6 compared to 100 in Fig. 5 (left). This can be expected, since the subproblem routines are more expensive compared to approaches that make use of Remark 2. Figure 6 also suggests that using Remark 1 is not beneficial over simply using the standard outer-approximation subproblem. Thus, for instances with ambiguous lower-level solutions, ST-STD is the method of choice. This is underlined by mean and median running times as well as the numbers of solved subproblems that are shown in Table 5.

The table reveals also another interesting aspect. While the mean and median number of solved subproblems is very comparable for ST-R1 and ST-STD (as well as

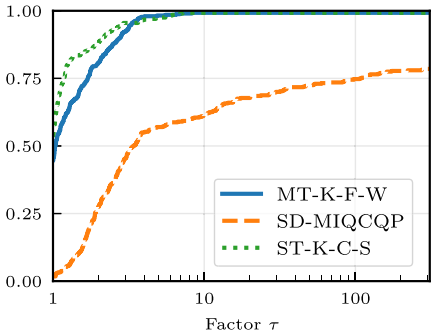

Fig. 7 Log-scaled performance profile for MT-K-F-W, ST-K-C-S, and the benchmark SD-MIQCQP on the instances in $\mathcal{I}^{\text{lc}}$ that at least one of the three approaches solves

Table 6 Running times (in sec), number of solved subproblems and time spent in the subproblems for MT-K-F-W, ST-K-C-S, and the benchmark SD-MIQCQP on the subset of instances in $\mathcal{I}^{\text{lc}}$ that all three approaches solve

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	138.30	9.14	10.34	1.00	0.50	0.14
SD-MIQCQP	501.77	20.90	–	–	–	–
ST-K-C-S	118.97	7.12	11.10	1.00	0.64	0.24

for MT-R1 vs. MT-STD), the mean and median times spent in the subproblems differ significantly. While the first aspect can be expected due to the same algorithmic setting, the latter aspect is interesting. ST-STD spends more than twice the time of ST-R1 in the subproblems in the mean, but it only spends half of the time of ST-R1 in the subproblem in the median. Thus, there seem to be few instances for which the standard subproblem (S^p) is very challenging, but for the most instances it is much easier to solve than subsequently solving the two problems (3) and (23) as proposed by Remark 1.

Second, we choose the entries of the matrices Q, R, S, and D to be in $[-\sqrt[2]{\sigma}, \sqrt[2]{\sigma}]$ respectively $[1, \sqrt{\sigma}]$ with $\sigma = \max\{\|c_u\|_\infty, \|d_u\|_\infty\}$ (or $\sigma = \|d_l\|_\infty$) instead of $[-\sqrt[4]{\sigma}, \sqrt[4]{\sigma}]$. In this setting, the coefficients of the resulting matrices $H_u = Q^\top Q$, $G_u = R^\top R$, and $G_l = S^\top S + D$ have larger absolute values and an analysis of the resulting matrices revealed that also the size of the spectrum, i.e., the range between the smallest and largest eigenvalue, increases compared to the matrices described in Sect. 4.2. We label this modified test set as $\mathcal{I}^{\text{lc}}$ and compare the methods MT-K-F-W, ST-K-C-S, and SD-MIQCQO. Performance profiles of those 400 instances in $\mathcal{I}^{\text{lc}}$ that can be solved by at least one of the three methods is shown in Fig. 7.

In comparison to the standard test set $\mathcal{I}$, see Fig. 5, the dominance of the outer-approximation approaches is even more pronounced. The reason for this behavior is that the outer-approximation methods need to solve significantly less subproblems on the test set $\mathcal{I}^{\text{lc}}$; see Table 6.

In fact, for at least half of the instances in $\mathcal{I}^{\text{lc}}$ the outer-approximation methods need to solve only 1 subproblem; see the median numbers of solved subproblems in

Table 6. In addition, there is not much difference between the multi- and the single-tree method; see Fig. 7 and the mean and median running times in Table 6. One possible explanation follows the discussion in Sect. 3.3. For many instances, the linear parts in the upper- and lower-level objective functions model a min-max structure. This structure cannot be "purely" present in our quadratic setting. Thus, choosing larger coefficients in the quadratic parts reduces the min-max structure and the two objective functions are more aligned, such that the first or second integer solution x_I of the master problem already yields the bilevel optimal solution. Overall, the properties of the involved matrices of the quadratic terms seem to have a significant impact on the effectiveness of the outer approximation algorithms.

Third, we highlight that for almost all instance classes in $\mathcal{I}$, the linking variables are binary. The reason is that most of these instances originate from MILP-MILP interdiction instances, in which the integer variables are binary by nature. In addition, for many instances in XUWANG and XULARGE that have lower and upper bounds of 0 and 10, the implied bounds are in fact 0 and 1, because the right-hand side values of these randomly generated instances limit the feasible region of the linking variables. In [54] it is pointed out that enlarging the feasible set of the original MILP-MILP instances by changing the right-hand side vectors renders the instances very hard to solve. We observed the same pattern for the MIQP-QP variants in a preliminary numerical test. In our notation, all entries in A, B, a, C, D, and b are non-positive. Decreasing the right-hand side vectors renders most of the instances too hard to be solved by any of the proposed outer-approximation or benchmark methods. On the one hand, this underlines the general hardness of MIQP-QP bilevel problems with general integer (i.e., not binary) linking variables. On the other hand, the instances XUWANG and XULARGE are randomly generated and it is not clear whether these observations generalize to "real-world" bilevel instances.

5 Conclusion

In this paper, we considered bilevel problems with a convex-quadratic mixed-integer upper level and a convex-quadratic lower level. Further, all linking variables are assumed to be bounded integers. For such problems, we proposed an equivalent transformation to a single-level convex MINLP and developed a multi- and a single-tree outer-approximation algorithm that we derived from algorithms for general convex MINLPs. We further proposed enhancements of these algorithms that exploit the bilevel-specific structure of the problem. Finally, we proved the correctness of the methods and carried out an extensive numerical study.

The study revealed that the two proposed outer-approximation algorithms outperform known benchmark approaches. For bilevel problems with unique lower-level solutions, the proposed bilevel-tailored solution of the subproblem turned out to be very effective. Even for instances with ambiguous lower-level problems, the novel algorithms perform better than the benchmark approaches. In general, the single-tree outer approximation implementation performs better than the multi-tree counterpart and is, in our opinion, the preferred method.

For both methods several questions remain open. Following the discussion in Sect. 4.6, the impact of the objective function matrices and their spectra as well as the performance on instances with non-binary linking variables needs further assessment. Up to know, this is however not possible due to a lack of bilevel instances with such properties. An interesting direction for future research can be to drop the integrality condition on the linking variables. This would require, e.g., spatial branching on linking variables but may also introduce some pitfalls like unattainable bilevel solutions; see [43]. Another question is whether one can introduce integer variables to the lower level, i.e., considering MIQP-MIQP bilevel problems. Certainly, the strong-duality-based reformulation for convex lower-level problems would not be applicable anymore, but one could use a more general single-level reformulation like the value-function reformulation, as it is done, e.g., for MILP-MILP bilevel problems in [21].

Acknowledgements This research has been performed as part of the Energie Campus Nürnberg and is supported by funding of the Bavarian State Government. We also thank the Deutsche Forschungsgemeinschaft for their support within project A05 and B08 in the "Sonderforschungsbereich/Transregio 154 Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks".

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix A: Detailed results

In this section, we provide more detailed results for the benchmark approach SD-MIQCQP as well as for MT-K-F-W and ST-K-C-S. For each instance class specified in Table 1, we provide

(i) performance profiles for all instances of the class that can be solved by at least one of the three methods;
(ii) a table with mean and median running times and number of solved subproblems for all instances of the class that could be solved by all three approaches;
(iii) a table with the exit status, gap, and running time of each solver for each instance of the class.

In the latter tables, a status "optimal" means that the instance has been solved to global optimality, "time limit" specifies that the time limit has been reached, "suboptimal" indicates a suboptimal termination due to unsolved node relaxations, and "numerics" denotes numerical issues detected in an ex-post feasibility check. Note that the last status is only relevant for SD-MIQCQP. For each instance, we mark the best running time using bold font.

A.1 Results for CLIQUE

Figure 8, Tables 7 and 8 show the results for the instance class CLIQUE.

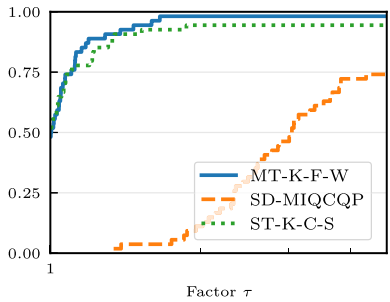

Fig. 8 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class CLIQUE that can be solved by at least one approach

Table 7 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class CLIQUE that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	606.42	406.37	1.00	1.00	2.02	1.65
SD-MIQCQP	1357.71	1198.74	–	–	–	–
ST-K-C-S	587.04	392.28	1.00	1.00	2.03	1.66

Table 8 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class CLIQUE

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
normalClique40-70-1	Optimal	0.00	373.04	Optimal	0.0	**110.05**	Optimal	0.00	113.24
normalClique40-70-2	Optimal	0.00	369.69	Optimal	0.0	100.47	Optimal	0.00	**97.42**
normalClique40-70-3	Optimal	0.00	289.83	Optimal	0.0	140.19	Optimal	0.00	**139.51**
normalClique40-70-4	Optimal	0.00	257.63	Optimal	0.0	136.47	Optimal	0.00	**127.54**
normalClique40-70-5	Optimal	0.00	350.39	Optimal	0.0	**91.68**	Optimal	0.00	92.67
normalClique40-70-6	Optimal	0.00	322.25	Optimal	0.0	**132.26**	Optimal	0.00	142.48
normalClique40-70-7	Optimal	0.00	335.94	Optimal	0.0	**109.09**	Optimal	0.00	117.05
normalClique40-70-8	Optimal	0.00	400.47	Optimal	0.0	**108.85**	Optimal	0.00	144.90
normalClique40-70-9	Optimal	0.00	366.14	Optimal	0.0	127.73	Optimal	0.00	**127.41**
normalClique40-70-10	Optimal	0.00	407.76	Optimal	0.0	119.37	Optimal	0.00	**106.55**
normalClique40-90-1	Optimal	0.00	585.19	Optimal	0.0	**250.06**	Optimal	0.00	263.49
normalClique40-90-2	Optimal	0.00	775.76	Optimal	0.0	**278.78**	Optimal	0.00	364.20
normalClique40-90-3	Optimal	0.00	521.93	Optimal	0.0	**266.40**	Optimal	0.00	319.08
normalClique40-90-4	Optimal	0.00	280.62	Optimal	0.0	219.40	Optimal	0.00	**209.49**
normalClique40-90-5	Optimal	0.00	822.14	Optimal	0.0	328.74	Optimal	0.00	**315.67**
normalClique40-90-6	Optimal	0.00	843.31	Optimal	0.0	**338.44**	Optimal	0.00	351.19
normalClique40-90-7	Optimal	0.00	1091.98	Optimal	0.0	**415.62**	Optimal	0.00	558.84
normalClique40-90-8	Optimal	0.00	656.00	Optimal	0.0	196.99	Optimal	0.00	**185.95**
normalClique40-90-9	Optimal	0.00	352.58	Optimal	0.0	272.36	Optimal	0.00	**254.31**
normalClique40-90-10	Optimal	0.00	616.73	Optimal	0.0	397.12	Optimal	0.00	**287.48**
normalClique50-70-1	Optimal	0.00	1581.87	Optimal	0.0	**613.21**	Optimal	0.00	652.19
normalClique50-70-2	Optimal	0.00	1625.99	Optimal	0.0	**534.33**	Optimal	0.00	548.84
normalClique50-70-3	Optimal	0.00	1857.00	Optimal	0.0	618.52	Optimal	0.00	**608.52**
normalClique50-70-4	Optimal	0.00	1364.65	Optimal	0.0	**317.41**	Optimal	0.00	349.81
normalClique50-70-5	Optimal	0.00	1643.38	Optimal	0.0	545.34	Optimal	0.00	**533.66**
normalClique50-70-6	Optimal	0.00	1307.56	Optimal	0.0	636.85	Optimal	0.00	**433.87**
normalClique50-70-7	Optimal	0.00	1390.13	Optimal	0.0	**418.98**	Optimal	0.00	510.08
normalClique50-70-8	Optimal	0.00	1638.54	Optimal	0.0	**609.33**	Optimal	0.00	673.89

Table 8 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
normalClique50-70-9	Optimal	0.00	1530.60	Optimal	0.0	501.33	Optimal	0.00	**420.35**
normalClique50-70-10	Optimal	0.00	1305.50	Optimal	0.0	**566.46**	Optimal	0.00	687.50
normalClique50-90-1	Optimal	0.00	3203.79	Optimal	0.0	**1463.11**	Optimal	0.00	2231.93
normalClique50-90-2	Time limit	0.00	3612.06	Optimal	0.0	1495.54	Optimal	0.00	**1161.41**
normalClique50-90-3	Optimal	0.00	2375.81	Optimal	0.0	1314.81	Optimal	0.00	**1263.55**
normalClique50-90-4	Optimal	0.00	3483.73	Optimal	0.0	3299.91	Optimal	0.00	**1990.24**
normalClique50-90-5	Optimal	0.00	2713.89	Optimal	0.0	995.94	Optimal	0.00	**862.32**
normalClique50-90-6	Optimal	0.00	2517.60	Optimal	0.0	1133.98	Optimal	0.00	**1080.39**
normalClique50-90-7	Optimal	0.00	3589.76	Optimal	0.0	**1254.83**	Optimal	0.00	1338.87
normalClique50-90-8	Optimal	0.00	2553.33	Optimal	0.0	1128.11	Optimal	0.00	**1002.23**
normalClique50-90-9	Optimal	0.00	3192.67	Optimal	0.0	1612.44	Optimal	0.00	**1369.13**
normalClique50-90-10	Optimal	0.00	3205.23	Optimal	0.0	**1361.15**	Optimal	0.00	1408.86
normalClique60-70-1	Time limit	0.07	3600.02	Optimal	0.0	1557.44	Optimal	0.00	**1388.07**
normalClique60-70-2	Time limit	0.78	3600.04	Optimal	0.0	**2283.10**	Time limit	0.19	3600.12
normalClique60-70-3	Time limit	0.47	3600.21	Optimal	0.0	**1447.52**	Optimal	0.00	2713.71
normalClique60-70-4	Optimal	0.00	2206.92	Optimal	0.0	**1190.69**	Optimal	0.00	1197.08
normalClique60-70-5	Time limit	1.12	3600.29	Optimal	0.0	**1723.88**	Optimal	0.00	2108.85
normalClique60-70-6	Time limit	0.61	3600.04	Optimal	0.0	1479.36	Optimal	0.00	**1409.20**
normalClique60-70-7	Time limit	0.33	3600.02	Optimal	0.0	2098.55	Optimal	0.00	**1904.26**
normalClique60-70-8	Time limit	1.44	3600.02	Optimal	0.0	2557.95	Optimal	0.00	**1602.48**
normalClique60-70-9	Time limit	0.00	3616.98	Optimal	0.0	1900.93	Optimal	0.00	**1874.10**
normalClique60-70-10	Time limit	0.00	3600.21	Optimal	0.0	2070.09	Optimal	0.00	**1850.07**
normalClique60-90-3	Time limit	22.66	3600.15	Optimal	0.0	**3119.34**	Time limit	0.02	3600.07
normalClique60-90-5	Time limit	26.08	3600.05	Time limit	100.0	3600.17	Time limit	5.67	3600.25
normalClique60-90-6	Time limit	22.98	3600.40	Optimal	0.0	**3269.06**	Optimal	0.00	3376.79
normalClique60-90-9	Time limit	23.68	3600.46	Time limit	100.0	3600.33	Optimal	0.00	**3381.69**
normalClique60-90-10	Time limit	22.97	3600.19	Optimal	0.0	**2948.06**	Time limit	18.74	3600.03

Bold values indicates the fastest running time

A.2 Results for DENEGRE

Figure 9, Tables 9 and 10 show the results for the instance class DENEGRE.

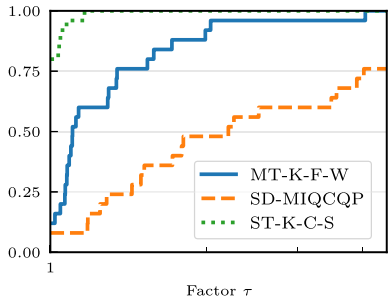

Fig. 9 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class DENEGRE that can be solved by at least one approach

Table 9 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class DENEGRE that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	11.28	4.34	10.21	9.00	0.02	0.01
SD-MIQCQP	22.65	4.10	–	–	–	–
ST-K-C-S	9.93	3.01	11.05	12.00	0.08	0.09

Table 10 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class DENEGRE

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
miblp_20_15_50_0110_10_2	Numerics	–	0.11	Optimal	0.0	0.10	Optimal	0.0	**0.09**
miblp_20_15_50_0110_10_4	Optimal	0.0	0.35	Optimal	0.0	1.10	Optimal	0.0	**0.27**
miblp_20_20_50_0110_5_1	Optimal	0.0	44.41	Optimal	0.0	18.93	Optimal	0.0	**11.03**
miblp_20_20_50_0110_5_2	Optimal	0.0	2.23	Optimal	0.0	1.90	Optimal	0.0	**1.47**
miblp_20_20_50_0110_5_4	Optimal	0.0	13.51	Optimal	0.0	12.15	Optimal	0.0	**9.04**
miblp_20_20_50_0110_5_6	Numerics	–	0.50	Optimal	0.0	0.81	Optimal	0.0	**0.75**
miblp_20_20_50_0110_5_7	Optimal	0.0	65.24	Optimal	0.0	**18.72**	Optimal	0.0	21.80
miblp_20_20_50_0110_5_9	Optimal	0.0	76.00	Optimal	0.0	22.08	Optimal	0.0	**19.47**
miblp_20_20_50_0110_5_10	Optimal	0.0	2.67	Optimal	0.0	2.31	Optimal	0.0	**2.14**
miblp_20_20_50_0110_5_12	Numerics	–	0.54	Optimal	0.0	0.36	Optimal	0.0	**0.35**

Table 10 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
miblp_20_20_50_0110_5_13	Optimal	0.0	21.19	Optimal	0.0	9.20	Optimal	0.0	**8.39**
miblp_20_20_50_0110_5_14	Optimal	0.0	12.99	Optimal	0.0	7.93	Optimal	0.0	**7.19**
miblp_20_20_50_0110_5_15	Optimal	0.0	4.10	Optimal	0.0	**1.81**	Optimal	0.0	1.88
miblp_20_20_50_0110_5_16	Optimal	0.0	44.54	Optimal	0.0	**25.89**	Optimal	0.0	27.07
miblp_20_20_50_0110_5_17	Numerics	–	38.83	Optimal	0.0	13.36	Optimal	0.0	**12.53**
miblp_20_20_50_0110_10_2	Optimal	0.0	**2.18**	Optimal	0.0	4.34	Optimal	0.0	2.34
miblp_20_20_50_0110_10_3	Optimal	0.0	**2.90**	Optimal	0.0	3.03	Optimal	0.0	3.01
miblp_20_20_50_0110_10_4	Optimal	0.0	1.74	Optimal	0.0	1.90	Optimal	0.0	**1.48**
miblp_20_20_50_0110_10_5	Optimal	0.0	3.57	Optimal	0.0	3.93	Optimal	0.0	**2.48**
miblp_20_20_50_0110_10_6	Optimal	0.0	111.70	Optimal	0.0	67.78	Optimal	0.0	**62.34**
miblp_20_20_50_0110_10_7	Optimal	0.0	17.12	Optimal	0.0	7.39	Optimal	0.0	**4.80**
miblp_20_20_50_0110_10_8	Optimal	0.0	2.01	Optimal	0.0	1.85	Optimal	0.0	**0.91**
miblp_20_20_50_0110_10_10	Optimal	0.0	1.90	Optimal	0.0	2.16	Optimal	0.0	**1.61**
miblp_20_20_50_0110_15_8	Numerics	–	0.13	Optimal	0.0	0.15	Optimal	0.0	**0.14**
miblp_20_20_50_0110_15_10	Numerics	–	0.10	Optimal	0.0	0.07	Optimal	0.0	**0.06**

Bold values indicates the fastest running time

A.3 Results for IMKP

Figure 10, Tables 11 and 12 show the results for the instance class IMKP.

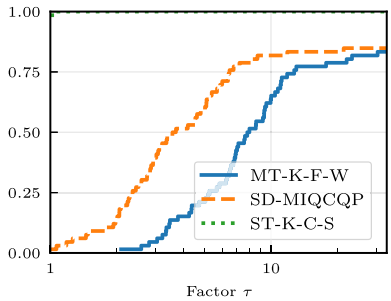

Fig. 10 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class IMKP that can be solved by at least one approach

Table 11 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class IMKP that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	375.47	54.21	186.64	130.00	0.35	0.18
SD-MIQCQP	251.66	29.50	–	–	–	–
ST-K-C-S	41.81	8.45	230.21	172.00	1.82	1.17

Table 12 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class IMKP

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
hp1.dat-0	Optimal	0.00	9.36	Optimal	0.00	23.62	Optimal	0.00	**4.56**
hp1.dat-50	Optimal	0.00	68.93	Optimal	0.00	98.39	Optimal	0.00	**15.23**
hp1.dat-100	Optimal	0.00	238.16	Optimal	0.00	1726.23	Optimal	0.00	**56.73**
hp2.dat-0	Optimal	0.00	581.87	Optimal	0.00	863.22	Optimal	0.00	**114.22**
hp2.dat-50	Optimal	0.00	604.56	Optimal	0.00	475.87	Optimal	0.00	**69.62**
hp2.dat-100	Time limit	2.41	3600.01	Time limit	25.15	3600.01	Optimal	0.00	**491.51**
pb1.dat-0	Optimal	0.00	13.92	Optimal	0.00	47.45	Optimal	0.00	**6.23**
pb1.dat-50	Optimal	0.00	54.91	Optimal	0.00	160.58	Optimal	0.00	**23.29**
pb1.dat-100	Optimal	0.00	43.58	Optimal	0.00	66.27	Optimal	0.00	**9.44**
pb2.dat-0	Optimal	0.00	998.67	Time limit	17.07	3600.00	Optimal	0.00	**330.41**
pb2.dat-50	Optimal	0.00	1864.53	Optimal	0.00	911.59	Optimal	0.00	**87.37**
pb2.dat-100	Optimal	0.00	2016.35	Optimal	0.00	3199.23	Optimal	0.00	**297.77**
pb4.dat-50	Optimal	0.00	0.92	Optimal	0.00	1.53	Optimal	0.00	**0.73**
pb5.dat-0	Optimal	0.00	53.74	Optimal	0.00	187.19	Optimal	0.00	**10.53**
pb5.dat-50	Optimal	0.00	66.40	Optimal	0.00	76.05	Optimal	0.00	**8.10**
pb5.dat-100	Optimal	0.00	104.70	Optimal	0.00	148.33	Optimal	0.00	**16.15**
pb6.dat-0	Optimal	0.00	737.00	Optimal	0.00	2415.26	Optimal	0.00	**217.65**
pb6.dat-50	Optimal	0.00	30.93	Optimal	0.00	160.34	Optimal	0.00	**12.31**
pb6.dat-100	Optimal	0.00	24.00	Optimal	0.00	180.92	Optimal	0.00	**8.22**
pb7.dat-0	Time limit	21.47	3600.01	Optimal	0.00	3170.62	Optimal	0.00	**949.22**
pb7.dat-50	Time limit	26.71	3600.00	Time limit	41.89	3600.02	Time limit	5.31	3600.00
pet3.dat-0	Optimal	0.00	0.18	Optimal	0.00	0.70	Optimal	0.00	**0.17**
pet3.dat-50	Optimal	0.00	0.24	Optimal	0.00	0.59	Optimal	0.00	**0.10**
pet3.dat-100	Optimal	0.00	0.20	Optimal	0.00	0.51	Optimal	0.00	**0.07**
pet4.dat-0	Optimal	0.00	0.66	Optimal	0.00	1.46	Optimal	0.00	**0.45**
pet4.dat-50	Optimal	0.00	0.87	Optimal	0.00	5.88	Optimal	0.00	**0.59**
pet4.dat-100	Optimal	0.00	2.58	Optimal	0.00	28.69	Optimal	0.00	**1.23**
pet5.dat-0	Optimal	0.00	40.53	Optimal	0.00	35.40	Optimal	0.00	**12.63**
pet5.dat-50	Optimal	0.00	29.50	Optimal	0.00	111.80	Optimal	0.00	**8.78**
pet5.dat-100	Optimal	0.00	54.20	Optimal	0.00	58.58	Optimal	0.00	**8.45**

Table 12 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
pet6.dat-0	Time limit	14.99	3600.01	Time limit	37.31	3600.02	Optimal	0.00	**2260.87**
pet6.dat-100	Optimal	0.00	1683.56	Optimal	0.00	3118.95	Optimal	0.00	**263.56**
weing3.dat-50	Optimal	0.00	0.36	Optimal	0.00	0.96	Optimal	0.00	**0.18**
weing4.dat-50	Optimal	0.00	**0.15**	Optimal	0.00	1.01	Optimal	0.00	0.16
weing8.dat-50	Optimal	0.00	19.49	Optimal	0.00	33.77	Optimal	0.00	**9.70**
weish01.dat-0	Optimal	0.00	11.63	Optimal	0.00	32.78	Optimal	0.00	**4.11**
weish01.dat-50	Optimal	0.00	59.58	Optimal	0.00	122.00	Optimal	0.00	**11.06**
weish01.dat-100	Optimal	0.00	104.93	Optimal	0.00	166.41	Optimal	0.00	**17.86**
weish02.dat-0	Optimal	0.00	4.28	Optimal	0.00	30.96	Optimal	0.00	**3.67**
weish02.dat-50	Optimal	0.00	7.33	Optimal	0.00	16.21	Optimal	0.00	**2.56**
weish02.dat-100	Optimal	0.00	6.12	Optimal	0.00	9.73	Optimal	0.00	**2.06**
weish03.dat-0	Optimal	0.00	6.79	Optimal	0.00	22.30	Optimal	0.00	**3.30**
weish03.dat-50	Optimal	0.00	7.18	Optimal	0.00	34.59	Optimal	0.00	**3.72**
weish03.dat-100	Optimal	0.00	8.30	Optimal	0.00	30.95	Optimal	0.00	**3.62**
weish04.dat-0	Optimal	0.00	12.11	Optimal	0.00	19.18	Optimal	0.00	**2.01**
weish04.dat-50	Optimal	0.00	13.73	Optimal	0.00	33.19	Optimal	0.00	**5.31**
weish04.dat-100	Optimal	0.00	2.75	Optimal	0.00	6.80	Optimal	0.00	**1.56**
weish05.dat-0	Optimal	0.00	5.70	Optimal	0.00	7.12	Optimal	0.00	**2.33**
weish05.dat-50	Optimal	0.00	16.08	Optimal	0.00	26.01	Optimal	0.00	**5.16**
weish05.dat-100	Optimal	0.00	2.73	Optimal	0.00	3.39	Optimal	0.00	**0.52**
weish06.dat-50	Optimal	0.00	1125.84	Optimal	0.00	328.41	Optimal	0.00	**95.03**
weish06.dat-100	Optimal	0.00	3413.24	Time limit	20.03	3600.01	Optimal	0.00	**474.51**
weish07.dat-0	Optimal	0.00	1298.08	Time limit	43.06	3600.01	Optimal	0.00	**349.98**
weish07.dat-50	Optimal	0.00	242.92	Optimal	0.00	481.89	Optimal	0.00	**48.38**
weish07.dat-100	Time limit	31.82	3600.05	Time limit	49.64	3600.01	Time limit	4.47	3600.00
weish08.dat-0	Optimal	0.00	919.85	Optimal	0.00	1502.95	Optimal	0.00	**257.36**
weish08.dat-50	Time limit	20.30	3600.01	Time limit	56.42	3600.00	Optimal	0.00	**1107.42**
weish08.dat-100	Time limit	13.62	3600.01	Time limit	58.60	3600.00	Optimal	0.00	**1743.53**
weish09.dat-0	Optimal	0.00	334.19	Optimal	0.00	263.44	Optimal	0.00	**69.56**
weish09.dat-50	Optimal	0.00	60.86	Optimal	0.00	221.16	Optimal	0.00	**19.77**
weish09.dat-100	Optimal	0.00	34.11	Optimal	0.00	54.21	Optimal	0.00	**12.44**
weish10.dat-100	Time limit	23.93	3600.01	Time limit	57.96	3600.02	Optimal	0.00	**842.24**
weish11.dat-0	Time limit	19.38	3600.01	Time limit	46.45	3600.01	Optimal	0.00	**1671.50**
weish11.dat-50	Optimal	0.00	1367.17	Optimal	0.00	1874.42	Optimal	0.00	**240.13**
weish11.dat-100	Time limit	16.52	3600.01	Time limit	56.88	3600.00	Optimal	0.00	**840.72**
weish12.dat-50	Time limit	18.56	3600.01	Time limit	26.84	3600.03	Optimal	0.00	**1111.43**
weish13.dat-0	Optimal	0.00	637.29	Optimal	0.00	471.30	Optimal	0.00	**140.42**
weish13.dat-50	Time limit	20.53	3600.01	Optimal	0.00	3227.19	Optimal	0.00	**1249.43**

Bold values indicates the fastest running time

A.4 Results for INT0SUM

Figure 11, Tables 13 and 14 show the results for the instance class INT0SUM.

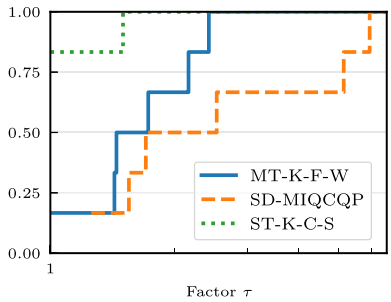

Fig. 11 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class INT0SUM that can be solved by at least one approach

Table 13 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class INT0SUM that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	375.47	54.21	186.64	130.00	0.35	0.18
SD-MIQCQP	251.66	29.50	–	–	–	–
ST-K-C-S	41.81	8.45	230.21	172.00	1.82	1.17

Table 14 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class INT0SUM

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
int0sum_i0_110	Optimal	0.0	1.15	Optimal	0.0	1.97	Optimal	0.0	**0.91**
int0sum_i0_160	Optimal	0.0	2.03	Optimal	0.0	**0.80**	Optimal	0.0	1.20
int0sum_i0_210	Optimal	0.0	2.00	Optimal	0.0	2.23	Optimal	0.0	**1.29**
int0sum_i0_260	Optimal	0.0	4.61	Optimal	0.0	6.55	Optimal	0.0	**2.71**
int0sum_i0_310	Optimal	0.0	26.04	Optimal	0.0	7.25	Optimal	0.0	**5.07**
int0sum_i0_360	Optimal	0.0	35.56	Optimal	0.0	8.66	Optimal	0.0	**5.99**

Bold values indicates the fastest running time

A.5 Results for INTER-ASSIG

Figure 12, Tables 15 and 16 show the results for the instance class INTER-ASSIG.

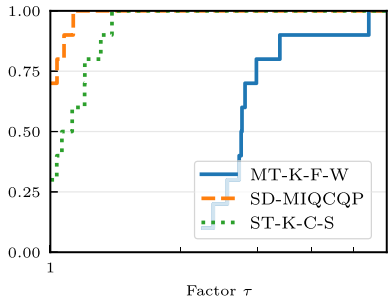

Fig. 12 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class INTER-ASSIG that can be solved by at least one approach

Table 15 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class INTER-ASSIG that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	0.72	0.59	14.30	13.00	0.03	0.03
SD-MIQCQP	0.25	0.22	–	–	–	–
ST-K-C-S	0.27	0.24	16.40	15.00	0.14	0.13

Table 16 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class INTER-ASSIG

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
2AP05-1	Optimal	0.0	0.28	Optimal	0.0	0.56	Optimal	0.0	**0.25**
2AP05-4	Optimal	0.0	**0.18**	Optimal	0.0	0.61	Optimal	0.0	0.23
2AP05-7	Optimal	0.0	0.44	Optimal	0.0	1.18	Optimal	0.0	**0.43**
2AP05-9	Optimal	0.0	**0.20**	Optimal	0.0	0.55	Optimal	0.0	0.23
2AP05-10	Optimal	0.0	**0.20**	Optimal	0.0	0.55	Optimal	0.0	0.24
2AP05-14	Optimal	0.0	0.22	Optimal	0.0	0.61	Optimal	0.0	**0.20**
2AP05-15	Optimal	0.0	**0.21**	Optimal	0.0	0.54	Optimal	0.0	0.22
2AP05-16	Optimal	0.0	**0.23**	Optimal	0.0	1.24	Optimal	0.0	0.32
2AP05-18	Optimal	0.0	**0.37**	Optimal	0.0	0.89	Optimal	0.0	0.40
2AP05-22	Optimal	0.0	**0.18**	Optimal	0.0	0.52	Optimal	0.0	0.22

A.6 Results for INTER-CLIQUE

Figure 13, Tables 17 and 18 show the results for the instance class INTER-CLIQUE.

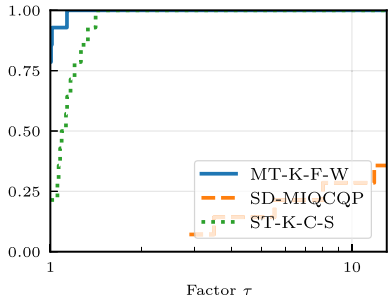

Fig. 13 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class INTER-CLIQUE that can be solved by at least one approach

Table 17 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class INTER-CLIQUE that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	0.25	0.22	1.00	1.00	0.02	0.03
SD-MIQCQP	1.37	1.19	–	–	–	–
ST-K-C-S	0.29	0.27	1.00	1.00	0.03	0.03

Table 18 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class INTER-CLIQUE

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
plusBCPIns_8_7_1.txt	Numerics	–	0.10	Optimal	0.0	**0.02**	Optimal	0.0	0.02
plusBCPIns_8_7_3.txt	Numerics	–	0.09	Optimal	0.0	**0.01**	Optimal	0.0	0.02
plusBCPIns_8_7_7.txt	Numerics	–	0.11	Optimal	0.0	**0.02**	Optimal	0.0	0.02
plusBCPIns_8_9_8.txt	Numerics	–	0.12	Optimal	0.0	0.02	Optimal	0.0	**0.02**
plusBCPIns_10_7_1.txt	Numerics	–	0.17	Optimal	0.0	**0.03**	Optimal	0.0	0.03
plusBCPIns_10_7_6.txt	Numerics	–	0.15	Optimal	0.0	**0.03**	Optimal	0.0	0.03
plusBCPIns_10_7_7.txt	Numerics	–	0.16	Optimal	0.0	**0.03**	Optimal	0.0	0.03
plusBCPIns_10_7_8.txt	Numerics	–	0.17	Optimal	0.0	**0.02**	Optimal	0.0	0.03
plusBCPIns_10_9_4.txt	Numerics	–	0.19	Optimal	0.0	0.04	Optimal	0.0	**0.03**
plusBCPIns_15_7_5.txt	Optimal	0.0	1.47	Optimal	0.0	**0.18**	Optimal	0.0	0.20
plusBCPIns_15_7_6.txt	Optimal	0.0	2.11	Optimal	0.0	**0.18**	Optimal	0.0	0.19

Table 18 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
plusBCPIns_15_7_7.txt	Optimal	0.0	1.19	Optimal	0.0	**0.22**	Optimal	0.0	0.27
plusBCPIns_15_9_3.txt	Optimal	0.0	1.04	Optimal	0.0	**0.30**	Optimal	0.0	0.42
plusBCPIns_15_9_8.txt	Optimal	0.0	1.02	Optimal	0.0	0.36	Optimal	0.0	**0.35**

Bold values indicates the fastest running time

A.7 Results for INTER-FIRE

Figure 14, Tables 19 and 20 show the results for the instance class INTER-FIRE.

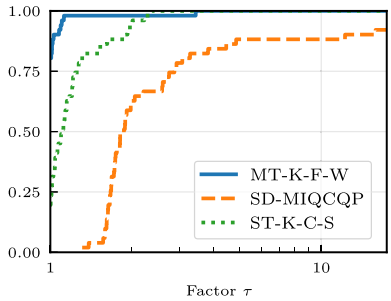

Fig. 14 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class INTER-FIRE that can be solved by at least one approach

Table 19 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class INTER-FIRE that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	3.24	2.92	1.06	1.00	0.06	0.04
SD-MIQCQP	11.25	5.03	–	–	–	–
ST-K-C-S	4.66	2.90	1.00	1.00	0.06	0.04

Table 20 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class INTER-FIRE

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
rndgraph-25_3-3-3_014	Numerics	–	0.20	Optimal	0.0	**0.09**	Optimal	0.0	0.09
rndgraph-50_1-3-3_002	Optimal	0.0	2.27	Optimal	0.0	**1.39**	Optimal	0.0	1.68
rndgraph-50_1-3-3_004	Optimal	0.0	2.05	Optimal	0.0	**1.31**	Optimal	0.0	1.61
rndgraph-50_1-3-3_007	Optimal	0.0	1.66	Optimal	0.0	**1.02**	Optimal	0.0	1.06
rndgraph-50_1-3-3_012	Optimal	0.0	1.74	Optimal	0.0	**1.08**	Optimal	0.0	1.18
rndgraph-50_1-5-3_001	Optimal	0.0	5.86	Optimal	0.0	**2.01**	Optimal	0.0	2.03
rndgraph-50_1-5-3_003	Optimal	0.0	1.78	Optimal	0.0	**0.99**	Optimal	0.0	1.25
rndgraph-50_1-5-3_011	Optimal	0.0	1.50	Optimal	0.0	0.91	Optimal	0.0	**0.90**
rndgraph-50_1-5-3_015	Optimal	0.0	3.19	Optimal	0.0	**1.69**	Optimal	0.0	1.94
rndgraph-50_3-5-5_002	Optimal	0.0	3.18	Optimal	0.0	**1.60**	Optimal	0.0	2.04
rndgraph-50_3-5-5_004	Optimal	0.0	1.82	Optimal	0.0	**1.14**	Optimal	0.0	1.29
rndgraph-50_3-5-5_008	Optimal	0.0	2.17	Optimal	0.0	**0.98**	Optimal	0.0	1.19
rndgraph-50_3-5-5_010	Optimal	0.0	1.68	Optimal	0.0	**0.93**	Optimal	0.0	1.07
rndgraph-80_1-3-3_001	Optimal	0.0	35.46	Optimal	0.0	**7.95**	Optimal	0.0	17.68
rndgraph-80_1-3-3_003	Optimal	0.0	81.51	Optimal	0.0	**6.66**	Optimal	0.0	7.56
rndgraph-80_1-3-3_007	Optimal	0.0	22.59	Optimal	0.0	**8.23**	Optimal	0.0	12.59
rndgraph-80_1-3-3_012	Optimal	0.0	25.04	Optimal	0.0	**7.64**	Optimal	0.0	18.06
rndgraph-80_2-2-2_002	Optimal	0.0	21.05	Optimal	0.0	**7.33**	Optimal	0.0	11.89
rndgraph-80_2-2-2_006	Optimal	0.0	95.75	Optimal	0.0	**6.04**	Optimal	0.0	7.26
rndgraph-80_2-2-2_009	Optimal	0.0	25.13	Optimal	0.0	**8.18**	Optimal	0.0	15.45

Table 20 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
rndgraph-80_2-2-2_014	Optimal	0.0	20.92	Optimal	0.0	**7.81**	Optimal	0.0	15.06
rndgraph-80_3-3-3_001	Optimal	0.0	15.00	Optimal	0.0	**5.77**	Optimal	0.0	8.74
rndgraph-80_3-3-3_002	Optimal	0.0	31.15	Optimal	0.0	**6.4**	Optimal	0.0	12.63
rndgraph-80_3-3-3_003	Optimal	0.0	14.76	Optimal	0.0	**5.71**	Optimal	0.0	7.36
rndgraph-80_3-3-3_006	Optimal	0.0	28.84	Optimal	0.0	**7.55**	Optimal	0.0	15.15
tree-25_2-2-2_005	Numerics	–	0.21	Optimal	0.0	**0.10**	Optimal	0.0	0.10
tree-25_2-2-2_009	Numerics	–	0.21	Optimal	0.0	**0.10**	Optimal	0.0	0.10
tree-25_3-3-3_012	Numerics	–	0.18	Optimal	0.0	0.08	Optimal	0.0	**0.08**
tree-50_1-3-3_004	Optimal	0.0	1.29	Optimal	0.0	0.8	Optimal	0.0	**0.79**
tree-50_1-3-3_007	Optimal	0.0	1.29	Optimal	0.0	**0.75**	Optimal	0.0	0.80
tree-50_1-3-3_012	Optimal	0.0	1.24	Optimal	0.0	**0.70**	Optimal	0.0	0.71
tree-50_1-3-3_014	Optimal	0.0	1.23	Optimal	0.0	**0.76**	Optimal	0.0	0.85
tree-50_1-5-5_005	Optimal	0.0	1.29	Optimal	0.0	2.68	Optimal	0.0	**0.78**
tree-50_1-5-5_006	Optimal	0.0	1.47	Optimal	0.0	0.80	Optimal	0.0	**0.72**
tree-50_1-5-5_007	Optimal	0.0	1.36	Optimal	0.0	**0.73**	Optimal	0.0	0.88
tree-50_1-5-5_010	Optimal	0.0	1.48	Optimal	0.0	**0.79**	Optimal	0.0	0.92
tree-50_3-3-1_003	Optimal	0.0	1.09	Optimal	0.0	0.69	Optimal	0.0	**0.63**
tree-50_3-3-1_007	Optimal	0.0	1.17	Optimal	0.0	**0.61**	Optimal	0.0	0.62
tree-50_3-3-1_008	Optimal	0.0	1.21	Optimal	0.0	**0.61**	Optimal	0.0	0.68

Table 20 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
tree-80_2-2-2_001	Optimal	0.0	6.40	Optimal	0.0	**3.63**	Optimal	0.0	4.10
tree-80_2-2-2_002	Optimal	0.0	5.75	Optimal	0.0	**3.35**	Optimal	0.0	3.36
tree-80_2-2-2_007	Optimal	0.0	6.00	Optimal	0.0	4.93	Optimal	0.0	**4.58**
tree-80_2-2-2_012	Optimal	0.0	6.16	Optimal	0.0	**3.72**	Optimal	0.0	3.87
tree-80_3-3-1_002	Optimal	0.0	4.88	Optimal	0.0	**3.52**	Optimal	0.0	3.61
tree-80_3-3-1_004	Optimal	0.0	5.96	Optimal	0.0	**3.30**	Optimal	0.0	3.51
tree-80_3-3-1_005	Optimal	0.0	5.36	Optimal	0.0	**3.18**	Optimal	0.0	3.47
tree-80_3-3-1_015	Optimal	0.0	6.00	Optimal	0.0	**3.17**	Optimal	0.0	4.25
tree-80_3-3-3_004	Optimal	0.0	5.40	Optimal	0.0	3.30	Optimal	0.0	**3.21**
tree-80_3-3-3_007	Optimal	0.0	5.90	Optimal	0.0	**3.54**	Optimal	0.0	3.74
tree-80_3-3-3_009	Optimal	0.0	5.03	Optimal	0.0	2.92	Optimal	0.0	**2.90**
tree-80_3-3-3_010	Optimal	0.0	5.48	Optimal	0.0	3.52	Optimal	0.0	**3.45**

Bold values indicates the fastest running time

A.8 Results for INTER-KP

Figure 15, Tables 21 and 22 show the results for the instance class INTER-KP.

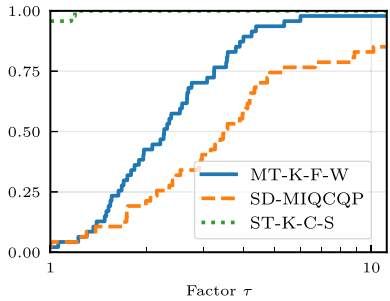

Fig. 15 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class INTER-KP that can be solved by at least one approach

Table 21 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class INTER-KP that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	342.80	40.38	77.35	50.00	0.08	0.05
SD-MIQCQP	580.29	50.54	–	–	–	–
ST-K-C-S	149.14	12.71	90.08	45.00	0.53	0.28

Table 22 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class INTER-KP

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
K5020W01.KNP	Optimal	0.00	1.06	Optimal	0.00	1.19	Optimal	0.0	**0.61**
K5020W02.KNP	Optimal	0.00	1.18	Optimal	0.00	1.18	Optimal	0.0	**0.46**
K5020W04.KNP	Optimal	0.00	0.55	Optimal	0.00	0.77	Optimal	0.0	**0.39**
K5020W05.KNP	Optimal	0.00	2.19	Optimal	0.00	4.52	Optimal	0.0	**1.27**
K5020W06.KNP	Optimal	0.00	**0.24**	Optimal	0.00	0.62	Optimal	0.0	0.29
K5020W08.KNP	Optimal	0.00	3.71	Optimal	0.00	5.15	Optimal	0.0	**1.87**
K5020W10.KNP	Optimal	0.00	0.33	Optimal	0.00	0.45	Optimal	0.0	**0.27**
K5020W11.KNP	Optimal	0.00	1.09	Optimal	0.00	1.46	Optimal	0.0	**0.35**
K5020W13.KNP	Optimal	0.00	0.91	Optimal	0.00	1.17	Optimal	0.0	**0.27**
K5020W14.KNP	Optimal	0.00	0.54	Optimal	0.00	1.02	Optimal	0.0	**0.54**

Table 22 continued

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
K5020W15.KNP	Optimal	0.00	0.25	Optimal	0.00	0.30	Optimal	0.0	**0.19**
K5020W16.KNP	Optimal	0.00	0.66	Optimal	0.00	0.58	Optimal	0.0	**0.38**
K5020W17.KNP	Optimal	0.00	0.90	Optimal	0.00	1.18	Optimal	0.0	**0.37**
K5020W19.KNP	Optimal	0.00	4.24	Optimal	0.00	4.10	Optimal	0.0	**1.97**
K5020W20.KNP	Optimal	0.00	0.56	Optimal	0.00	0.74	Optimal	0.0	**0.33**
K5030W01.KNP	Optimal	0.00	334.26	Optimal	0.00	62.91	Optimal	0.0	**37.71**
K5030W02.KNP	Optimal	0.00	82.33	Optimal	0.00	45.20	Optimal	0.0	**27.73**
K5030W03.KNP	Optimal	0.00	97.08	Optimal	0.00	80.82	Optimal	0.0	**24.94**
K5030W04.KNP	Optimal	0.00	256.26	Optimal	0.00	122.60	Optimal	0.0	**53.03**
K5030W05.KNP	Optimal	0.00	1347.74	Optimal	0.00	242.66	Optimal	0.0	**132.69**
K5030W06.KNP	Optimal	0.00	26.62	Optimal	0.00	20.14	Optimal	0.0	**9.15**
K5030W07.KNP	Optimal	0.00	403.59	Optimal	0.00	460.04	Optimal	0.0	**76.46**
K5030W08.KNP	Optimal	0.00	2901.78	Optimal	0.00	2347.61	Optimal	0.0	**665.08**
K5030W09.KNP	Optimal	0.00	15.95	Optimal	0.00	12.07	Optimal	0.0	**7.77**
K5030W10.KNP	Optimal	0.00	64.58	Optimal	0.00	48.35	Optimal	0.0	**15.70**
K5030W11.KNP	Optimal	0.00	100.67	Optimal	0.00	135.82	Optimal	0.0	**25.36**
K5030W12.KNP	Optimal	0.00	60.57	Optimal	0.00	8.41	Optimal	0.0	**6.88**
K5030W13.KNP	Optimal	0.00	50.75	Optimal	0.00	45.94	Optimal	0.0	**17.24**
K5030W14.KNP	Optimal	0.00	18.16	Optimal	0.00	17.06	Optimal	0.0	**7.56**
K5030W15.KNP	Optimal	0.00	231.28	Optimal	0.00	131.68	Optimal	0.0	**55.31**
K5030W16.KNP	Optimal	0.00	89.72	Optimal	0.00	67.24	Optimal	0.0	**18.94**
K5030W17.KNP	Optimal	0.00	208.46	Optimal	0.00	131.98	Optimal	0.0	**49.46**
K5030W18.KNP	Optimal	0.00	45.92	Optimal	0.00	42.77	Optimal	0.0	**11.31**
K5030W19.KNP	Optimal	0.00	2141.94	Optimal	0.00	2408.28	Optimal	0.0	**618.34**
K5030W20.KNP	Optimal	0.00	50.32	Optimal	0.00	37.99	Optimal	0.0	**14.12**
K5040W02.KNP	Optimal	0.00	3558.61	Optimal	0.00	1631.27	Optimal	0.0	**1091.81**
K5040W03.KNP	Optimal	0.00	1804.02	Optimal	0.00	1666.88	Optimal	0.0	**417.12**
K5040W04.KNP	Time limit	14.98	3600.00	Time limit	18.27	3600.47	Optimal	0.0	**1792.87**
K5040W06.KNP	Time limit	11.18	3600.00	Optimal	0.00	1545.70	Optimal	0.0	**1189.43**
K5040W09.KNP	Optimal	0.00	2792.56	Optimal	0.00	1657.71	Optimal	0.0	**1189.71**
K5040W10.KNP	Time limit	11.09	3600.00	Optimal	0.00	**971.59**	Optimal	0.0	1129.36
K5040W12.KNP	Time limit	15.48	3600.00	Optimal	0.00	1113.04	Optimal	0.0	**817.67**
K5040W13.KNP	Optimal	0.00	2917.00	Optimal	0.00	1467.52	Optimal	0.0	**844.68**
K5040W14.KNP	Time limit	18.86	3600.00	Optimal	0.00	2422.98	Optimal	0.0	**2288.78**
K5040W16.KNP	Optimal	0.00	3593.20	Optimal	0.00	794.46	Optimal	0.0	**537.96**
K5040W18.KNP	Time limit	10.68	3600.00	Optimal	0.00	2902.33	Optimal	0.0	**1236.55**
K5040W20.KNP	Time limit	13.46	3600.00	Optimal	0.00	1708.49	Optimal	0.0	**957.17**

Bold values indicates the fastest running time

A.9 Results for MIPLIB

Figure 16, Tables 23 and 24 show the results for the instance class MIPLIB.

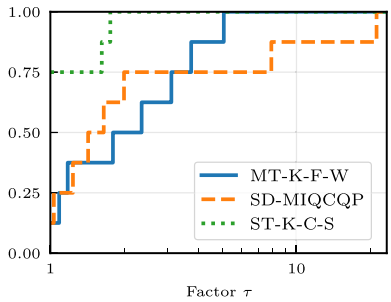

Fig. 16 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class MIPLIB that can be solved by at least one approach

Table 23 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class MIPLIB that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	9.15	2.24	7.62	4.50	0.17	0.04
SD-MIQCQP	4.21	1.85	–	–	–	–
ST-K-C-S	2.29	1.05	9.88	8.50	0.35	0.13

Table 24 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class MIPLIB

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
enigma-0.100000	Optimal	0.0	**0.66**	Optimal	0.0	0.77	Optimal	0.0	1.15
enigma-0.500000	Optimal	0.0	12.13	Optimal	0.0	4.76	Optimal	0.0	**1.53**
enigma-0.900000	Optimal	0.0	1.05	Optimal	0.0	2.76	Optimal	0.0	**0.74**
lseu-0.900000	Optimal	0.0	0.98	Optimal	0.0	0.65	Optimal	0.0	**0.60**
p0033-0.100000	Optimal	0.0	2.52	Optimal	0.0	**0.12**	Optimal	0.0	0.19
p0201-0.900000	Optimal	0.0	11.96	Optimal	0.0	58.66	Optimal	0.0	**11.58**
p0282-0.900000	Optimal	0.0	3.18	Optimal	0.0	3.75	Optimal	0.0	**1.59**
stein45-0.100000	Optimal	0.0	1.18	Optimal	0.0	1.72	Optimal	0.0	**0.96**

Bold values indicates the fastest running time

A.10 Results for XULARGE

Figure 17, Tables 25 and 26 show the results for the instance class XULARGE.

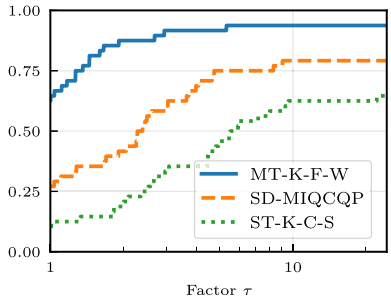

Fig. 17 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class XULARGE that can be solved by at least one approach

Table 25 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class XULARGE that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	334.39	262.69	22.75	21.00	19.27	15.42
SD-MIQCQP	648.82	424.62	–	–	–	–
ST-K-C-S	1172.12	1017.47	22.75	22.50	32.39	25.93

Table 26 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class XULARGE

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
xuLarge500-1	Optimal	0.00	**121.00**	Optimal	0.00	175.12	Optimal	0.00	537.25
xuLarge500-2	Optimal	0.00	1714.55	Optimal	0.00	**362.34**	Optimal	0.00	365.64
xuLarge500-3	Optimal	0.00	151.09	Optimal	0.00	138.98	Optimal	0.00	**118.73**
xuLarge500-4	Optimal	0.00	212.83	Optimal	0.00	103.97	Optimal	0.00	**103.68**
xuLarge500-5	Optimal	0.00	1459.84	Optimal	0.00	**161.21**	Optimal	0.00	380.26
xuLarge500-6	Optimal	0.00	263.79	Optimal	0.00	**86.42**	Optimal	0.00	256.90
xuLarge500-7	Time limit	76.61	3600.57	Optimal	0.00	675.98	Optimal	0.00	**127.10**
xuLarge500-8	Optimal	0.00	1562.00	Time limit	74.98	3600.01	Optimal	0.00	**336.60**
xuLarge500-9	Optimal	0.00	765.19	Optimal	0.00	**292.37**	Optimal	0.00	2102.37

Table 26 continued

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
xuLarge500-10	Optimal	0.00	1791.55	Time limit	100.50	3600.03	Optimal	0.00	**215.41**
xuLarge600-1	Time limit	0.00	3612.50	Optimal	0.00	**225.28**	Optimal	0.00	1809.43
xuLarge600-2	Optimal	0.00	**207.46**	Optimal	0.00	611.78	Optimal	0.00	1021.83
xuLarge600-3	Optimal	0.00	**237.78**	Optimal	0.00	379.92	Optimal	0.00	1063.74
xuLarge600-4	Optimal	0.00	384.68	Optimal	0.00	**226.74**	Optimal	0.00	575.84
xuLarge600-5	Optimal	0.00	274.75	Optimal	0.00	**120.33**	Optimal	0.00	714.46
xuLarge600-6	Optimal	0.00	**331.57**	Optimal	0.00	420.66	Optimal	0.00	696.44
xuLarge600-7	Optimal	0.00	191.48	Optimal	0.00	**63.90**	Optimal	0.00	612.92
xuLarge600-8	Optimal	0.00	**256.13**	Optimal	0.00	266.16	Optimal	0.00	789.83
xuLarge600-9	Optimal	0.00	2080.12	Optimal	0.00	**559.52**	Optimal	0.00	1013.10
xuLarge600-10	Optimal	0.00	1191.29	Optimal	0.00	**523.90**	Optimal	0.00	1051.19
xuLarge700-1	Time limit	112.82	3622.08	Optimal	0.00	**1507.66**	Time limit	80.89	3600.08
xuLarge700-2	Optimal	0.00	**433.36**	Optimal	0.00	589.76	Optimal	0.00	2362.92
xuLarge700-3	Time limit	0.00	3601.56	Optimal	0.00	**815.84**	Optimal	0.00	2107.97
xuLarge700-4	Optimal	0.00	**1443.02**	Time limit	150.16	3600.03	Optimal	0.00	1923.37
xuLarge700-5	Optimal	0.00	415.87	Optimal	0.00	**104.44**	Optimal	0.00	2314.45
xuLarge700-6	Optimal	0.00	494.81	Optimal	0.00	**259.23**	Optimal	0.00	1541.30
xuLarge700-7	Time limit	128.47	3600.05	Optimal	0.00	**2158.58**	Time limit	101.57	3600.09
xuLarge700-8	Time limit	79.22	3619.03	Optimal	0.00	**1345.66**	Optimal	0.00	2460.28
xuLarge700-9	Optimal	0.00	**435.40**	Optimal	0.00	833.66	Suboptimal	121.18	1761.38
xuLarge700-10	Optimal	0.00	528.58	Optimal	0.00	**215.11**	Optimal	0.00	1035.78
xuLarge800-1	Time limit	75.22	3606.32	Optimal	0.00	**628.20**	Time limit	38.31	3600.16
xuLarge800-3	Optimal	0.00	875.94	Optimal	0.00	**844.45**	Optimal	0.00	2258.26
xuLarge800-4	Time limit	198.99	3600.13	Optimal	0.00	**2476.00**	Time limit	98.35	3600.10
xuLarge800-5	Suboptimal	118.06	396.01	Optimal	0.00	**568.34**	Suboptimal	62.31	2471.99
xuLarge800-6	Optimal	0.00	1731.05	Optimal	0.00	**720.83**	Time limit	135.52	3600.03
xuLarge800-7	Optimal	0.00	**450.49**	Optimal	0.00	747.31	Optimal	0.00	2010.81
xuLarge800-8	Optimal	0.00	620.22	Optimal	0.00	**247.94**	Optimal	0.00	2309.24
xuLarge800-9	Optimal	0.00	638.98	Optimal	0.00	**498.95**	Time limit	117.47	3600.21
xuLarge800-10	Optimal	0.00	1908.76	Optimal	0.00	**523.82**	Optimal	0.00	2894.05
xuLarge900-2	Optimal	0.00	**758.57**	Optimal	0.00	2039.72	Time limit	150.34	3600.15
xuLarge900-3	Optimal	0.00	**696.52**	Optimal	0.00	1009.42	Time limit	102.17	3600.25
xuLarge900-4	Optimal	0.00	1639.29	Optimal	0.00	**1481.57**	Time limit	120.23	3600.11
xuLarge900-5	Time limit	139.56	3600.05	Time limit	189.09	3600.03	Time limit	140.06	3600.19
xuLarge900-6	Optimal	0.00	**949.70**	Optimal	0.00	1206.07	Time limit	144.54	3600.37
xuLarge900-8	Time limit	674.08	3624.87	Optimal	0.00	**491.74**	Time limit	96.86	3600.04

Table 26 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
xuLarge900-10	Optimal	0.00	**1570.63**	Optimal	0.00	1748.17	Time limit	151.39	3600.09
xuLarge1000-3	Optimal	0.00	1903.67	Optimal	0.00	**1137.92**	Time limit	121.02	3600.27
xuLarge1000-7	Optimal	0.00	1601.50	Optimal	0.00	**708.71**	Time limit	126.23	3600.29
xuLarge1000-8	Optimal	0.00	2046.20	Optimal	0.00	**500.56**	Time limit	84.41	3600.13

Bold values indicates the fastest running time

A.11 Results for XUWANG

Figure 18, Tables 27 and 28 show the results for the instance class XUWANG.

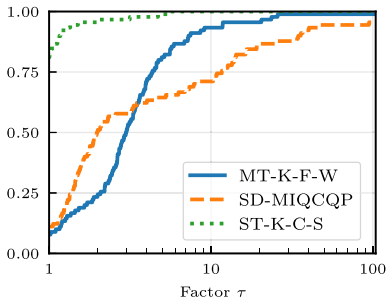

Fig. 18 Log-scaled performance profile for the instances in $\mathcal{I}$ belonging to the class XUWANG that can be solved by at least one approach

Table 27 Running times (in sec), number of solved subproblems and time spent in the subproblems (in sec) for the instances in $\mathcal{I}$ belonging to the class XUWANG that can be solved by every approach

	Running time		Solved subproblems		Time in subproblems	
	Mean	Median	Mean	Median	Mean	Median
MT-K-F-W	145.01	46.09	21.77	18.50	3.34	1.78
SD-MIQCQP	336.72	25.09	–	–	–	–
ST-K-C-S	43.12	13.48	22.94	22.00	6.41	3.88

Table 28 Exit status, gap, and running time for every instance in $\mathcal{I}$ belonging to the class XUWANG

	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
Instance	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
bmilplib_10_10	Optimal	0.00	1.53	Optimal	0.00	1.44	Optimal	0.0	**1.15**
bmilplib_60_1	Optimal	0.00	1.43	Optimal	0.00	3.45	Optimal	0.0	**0.77**
bmilplib_60_2	Optimal	0.00	1.22	Optimal	0.00	2.83	Optimal	0.0	**1.07**
bmilplib_60_3	Optimal	0.00	1.14	Optimal	0.00	2.20	Optimal	0.0	**0.73**
bmilplib_60_4	Optimal	0.00	1.27	Optimal	0.00	1.61	Optimal	0.0	**0.97**
bmilplib_60_5	Optimal	0.00	2.06	Optimal	0.00	3.83	Optimal	0.0	**1.16**
bmilplib_60_6	Optimal	0.00	1.80	Optimal	0.00	3.23	Optimal	0.0	**0.70**
bmilplib_60_8	Optimal	0.00	4.00	Optimal	0.00	6.76	Optimal	0.0	**2.32**
bmilplib_60_9	Optimal	0.00	**2.15**	Optimal	0.00	2.76	Optimal	0.0	2.49
bmilplib_60_10	Optimal	0.00	2.80	Optimal	0.00	2.95	Optimal	0.0	**0.87**
bmilplib_110_1	Optimal	0.00	1.27	Optimal	0.00	2.12	Optimal	0.0	**1.12**
bmilplib_110_2	Optimal	0.00	**0.83**	Optimal	0.00	2.19	Optimal	0.0	0.98
bmilplib_110_3	Optimal	0.00	3.03	Optimal	0.00	4.86	Optimal	0.0	**2.09**
bmilplib_110_4	Optimal	0.00	5.45	Optimal	0.00	14.26	Optimal	0.0	**3.62**
bmilplib_110_5	Optimal	0.00	7.09	Optimal	0.00	10.62	Optimal	0.0	**3.38**
bmilplib_110_6	Optimal	0.00	4.85	Optimal	0.00	8.13	Optimal	0.0	**2.46**
bmilplib_110_7	Optimal	0.00	4.62	Optimal	0.00	13.63	Optimal	0.0	**3.58**
bmilplib_110_8	Optimal	0.00	4.65	Optimal	0.00	21.53	Optimal	0.0	**3.77**
bmilplib_110_9	Optimal	0.00	5.73	Optimal	0.00	12.50	Optimal	0.0	**4.62**
bmilplib_110_10	Optimal	0.00	**13.01**	Optimal	0.00	89.19	Optimal	0.0	13.44
bmilplib_160_1	Optimal	0.00	6.51	Optimal	0.00	11.60	Optimal	0.0	**4.62**

Table 28 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
bmilplib_160_2	Optimal	0.00	16.39	Optimal	0.00	29.64	Optimal	0.0	**9.24**
bmilplib_160_3	Optimal	0.00	13.48	Optimal	0.00	22.30	Optimal	0.0	**6.52**
bmilplib_160_4	Optimal	0.00	14.68	Optimal	0.00	50.45	Optimal	0.0	**9.03**
bmilplib_160_5	Optimal	0.00	9.12	Optimal	0.00	10.03	Optimal	0.0	**4.83**
bmilplib_160_6	Optimal	0.00	3.67	Optimal	0.00	8.66	Optimal	0.0	**3.11**
bmilplib_160_7	Optimal	0.00	3.43	Optimal	0.00	3.58	Optimal	0.0	**2.33**
bmilplib_160_8	Optimal	0.00	8.22	Optimal	0.00	13.49	Optimal	0.0	**5.22**
bmilplib_160_9	Optimal	0.00	13.57	Optimal	0.00	24.58	Optimal	0.0	**9.29**
bmilplib_160_10	Optimal	0.00	22.33	Optimal	0.00	54.50	Optimal	0.0	**13.52**
bmilplib_210_1	Optimal	0.00	23.35	Optimal	0.00	30.86	Optimal	0.0	**6.70**
bmilplib_210_2	Optimal	0.00	14.19	Optimal	0.00	73.24	Optimal	0.0	**9.78**
bmilplib_210_3	Optimal	0.00	**4.66**	Optimal	0.00	12.88	Optimal	0.0	4.95
bmilplib_210_4	Optimal	0.00	22.91	Optimal	0.00	98.31	Optimal	0.0	**19.05**
bmilplib_210_5	Optimal	0.00	17.97	Optimal	0.00	37.61	Optimal	0.0	**12.62**
bmilplib_210_6	Optimal	0.00	**20.26**	Optimal	0.00	79.91	Optimal	0.0	24.10
bmilplib_210_7	Optimal	0.00	33.56	Optimal	0.00	68.06	Optimal	0.0	**18.81**
bmilplib_210_8	Optimal	0.00	**3.16**	Optimal	0.00	13.50	Optimal	0.0	4.27
bmilplib_210_9	Optimal	0.00	90.72	Optimal	0.00	509.33	Optimal	0.0	**55.98**
bmilplib_210_10	Optimal	0.00	4.21	Optimal	0.00	8.77	Optimal	0.0	**3.98**
bmilplib_260_1	Optimal	0.00	27.78	Optimal	0.00	18.77	Optimal	0.0	**8.51**
bmilplib_260_2	Optimal	0.00	**10.18**	Optimal	0.00	40.43	Optimal	0.0	11.46
bmilplib_260_3	Optimal	0.00	29.11	Optimal	0.00	7.95	Optimal	0.0	**7.24**

Table 28 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
bmilplib_260_4	Optimal	0.00	28.24	Optimal	0.00	48.96	Optimal	0.0	**13.44**
bmilplib_260_5	Optimal	0.00	9.84	Optimal	0.00	24.12	Optimal	0.0	**7.93**
bmilplib_260_6	Optimal	0.00	24.06	Optimal	0.00	52.41	Optimal	0.0	**12.05**
bmilplib_260_7	Optimal	0.00	**31.14**	Optimal	0.00	103.96	Optimal	0.0	98.20
bmilplib_260_8	Optimal	0.00	12.04	Optimal	0.00	21.11	Optimal	0.0	**8.56**
bmilplib_260_9	Optimal	0.00	26.12	Optimal	0.00	147.00	Optimal	0.0	**17.06**
bmilplib_260_10	Optimal	0.00	60.70	Optimal	0.00	381.18	Optimal	0.0	**32.32**
bmilplib_310_1	Optimal	0.00	111.01	Optimal	0.00	115.62	Optimal	0.0	**24.45**
bmilplib_310_2	Optimal	0.00	600.51	Optimal	0.00	56.85	Optimal	0.0	**16.24**
bmilplib_310_3	Optimal	0.00	826.70	Optimal	0.00	424.93	Optimal	0.0	**58.59**
bmilplib_310_4	Optimal	0.00	458.54	Optimal	0.00	127.59	Optimal	0.0	**33.10**
bmilplib_310_5	Optimal	0.00	3496.34	Optimal	0.00	182.80	Optimal	0.0	**36.89**
bmilplib_310_6	Optimal	0.00	361.80	Optimal	0.00	59.86	Optimal	0.0	**34.87**
bmilplib_310_7	Optimal	0.00	564.48	Optimal	0.00	337.97	Optimal	0.0	**46.94**
bmilplib_310_8	Optimal	0.00	**18.35**	Optimal	0.00	108.24	Optimal	0.0	19.04
bmilplib_310_9	Optimal	0.00	840.19	Optimal	0.00	137.61	Optimal	0.0	**24.67**
bmilplib_310_10	Optimal	0.00	171.84	Optimal	0.00	31.15	Optimal	0.0	**12.74**
bmilplib_360_1	Optimal	0.00	1630.29	Optimal	0.00	2009.82	Optimal	0.0	**84.98**
bmilplib_360_2	Optimal	0.00	1184.33	Optimal	0.00	1482.72	Optimal	0.0	**72.34**
bmilplib_360_3	Optimal	0.00	898.61	Optimal	0.00	**122.30**	Optimal	0.0	669.18
bmilplib_360_4	Optimal	0.00	723.10	Optimal	0.00	47.00	Optimal	0.0	**17.95**

Table 28 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
bmilplib_360_5	Optimal	0.00	561.20	Optimal	0.00	1335.51	Optimal	0.0	**51.69**
bmilplib_360_6	Optimal	0.00	484.41	Optimal	0.00	**19.09**	Optimal	0.0	24.84
bmilplib_360_7	Optimal	0.00	377.72	Optimal	0.00	94.60	Optimal	0.0	**22.73**
bmilplib_360_8	Optimal	0.00	74.64	Optimal	0.00	32.43	Optimal	0.0	**22.44**
bmilplib_360_9	Optimal	0.00	470.55	Optimal	0.00	**59.19**	Optimal	0.0	279.03
bmilplib_360_10	Optimal	0.00	203.50	Optimal	0.00	45.19	Optimal	0.0	**16.03**
bmilplib_410_1	Suboptimal	36.08	126.03	Optimal	0.00	**79.13**	Optimal	0.0	91.11
bmilplib_410_2	Optimal	0.00	74.11	Optimal	0.00	32.92	Optimal	0.0	**32.18**
bmilplib_410_3	Optimal	0.00	1772.29	Optimal	0.00	109.47	Optimal	0.0	**49.31**
bmilplib_410_4	Optimal	0.00	1905.55	Optimal	0.00	144.60	Optimal	0.0	**40.35**
bmilplib_410_5	Optimal	0.00	638.63	Optimal	0.00	111.14	Optimal	0.0	**81.60**
bmilplib_410_6	Optimal	0.00	1163.65	Optimal	0.00	964.60	Optimal	0.0	**81.48**
bmilplib_410_7	Optimal	0.00	1145.95	Optimal	0.00	89.56	Optimal	0.0	**33.10**
bmilplib_410_8	Optimal	0.00	488.11	Optimal	0.00	90.08	Optimal	0.0	**77.49**
bmilplib_410_9	Optimal	0.00	79.66	Optimal	0.00	**34.35**	Optimal	0.0	35.82
bmilplib_410_10	Optimal	0.00	874.24	Optimal	0.00	479.64	Optimal	0.0	**166.07**
bmilplib_460_1	Optimal	0.00	1192.67	Optimal	0.00	276.10	Optimal	0.0	**115.61**
bmilplib_460_2	Suboptimal	55.53	909.42	Time limit	26.02	3600.02	Optimal	0.0	**50.05**
bmilplib_460_3	Optimal	0.00	154.56	Optimal	0.00	95.09	Optimal	0.0	**80.09**
bmilplib_460_4	Optimal	0.00	1839.37	Optimal	0.00	**157.49**	Optimal	0.0	237.64
bmilplib_460_5	Optimal	0.00	126.42	Optimal	0.00	187.71	Optimal	0.0	**94.22**
bmilplib_460_6	Optimal	0.00	1060.16	Optimal	0.00	**153.38**	Optimal	0.0	303.81

Table 28 continued

Instance	SD-MIQCQP			MT-K-F-W			ST-K-C-S		
	Status	Gap	Time	Status	Gap	Time	Status	Gap	Time
bmilplib_460_7	Suboptimal	59.84	107.20	Optimal	0.00	160.66	Optimal	0.0	**133.84**
bmilplib_460_8	Suboptimal	42.21	793.25	Optimal	0.00	438.44	Optimal	0.0	**185.23**
bmilplib_460_9	Optimal	0.00	1576.36	Optimal	0.00	252.00	Optimal	0.0	**82.15**
bmilplib_460_10	Optimal	0.00	**132.28**	Optimal	0.00	244.73	Optimal	0.0	133.08

Bold values indicates the fastest running time

References

1. Abhishek, K., Leyffer, S., Linderoth, J.: FilMINT: an outer approximationbased solver for convex mixed-integer nonlinear programs. INFORMS J. Comput. **22**(4), 555–567 (2010). https://doi.org/10.1287/ijoc.1090.0373
2. Arroyo, J.M.: Bilevel programming applied to power system vulnerability analysis under multiple contingencies. IET Gener. Transm. Distrib. **4**(2), 178–190 (2010). https://doi.org/10.1049/iet-gtd.2009.0098
3. Avraamidou, S., Pistikopoulos, E.N.: A Multi-Parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. Comput. Chem. Eng. **125**, 98–113 (2019). https://doi.org/10.1016/j.compchemeng.2019.01.021
4. Baggio, A., Carvalho, M., Lodi, A., Tramontani, A.: Multilevel approaches for the critical node problem. Technical report, École Polytechnique de Montréal (2016)
5. Bard, J.F.: Convex two-level optimization. Math. Program. **40**(1), 15–27 (1988). https://doi.org/10.1007/BF01580720
6. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. SIAM J. Sci. Stat. Comput. **11**(2), 281–292 (1990). https://doi.org/10.1137/0911017
7. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. Acta Numerica **22**, 1–131 (2013). https://doi.org/10.1017/S0962492913000032
8. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., et al.: An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optim. **5**(2), 186–204 (2008). https://doi.org/10.1016/j.disopt.2006.10.011
9. Böttger, T., Grimm, V., Kleinert, T., Schmidt, M.: The Cost of Decoupling Trade and Transport in the European Entry-Exit Gas Market. Technical report (2020). http://www.optimization-online.org/DB_HTML/2020/06/7851.html
10. Boyd, S., Vandenberghe, L.: Convex Optim. (2004). https://doi.org/10.1017/cbo9780511804441
11. Caprara, A., Carvalho, M., Lodi, A., Woeginger, G.J.: Bilevel knapsack with interdiction constraints. INFORMS J. Comput. **28**(2), 319–333 (2016). https://doi.org/10.1287/ijoc.2015.0676
12. Daxhelet, O., Smeers, Y.: The EU regulation on cross-border trade of electricity: a two-stage equilibrium model. Eur. J. Oper. Res. **181**(3), 1396–1412 (2007). https://doi.org/10.1016/j.ejor.2005.12.040
13. Dempe, S.: Foundations of Bilevel Programming. Springer, Berlin (2002). https://doi.org/10.1007/b101970
14. Dempe, S., Kalashnikov, V., Pérez-Valdés, G.A., Kalashnykova, N.: Bilevel Program. Problems (2015). https://doi.org/10.1007/978-3-662-45827-3
15. Dempe, S., Zemkoho, A.B.: Bilevel road pricing: theoretical analysis and optimality conditions. Ann. Oper. Res. **196**(1), 223–240 (2012). https://doi.org/10.1007/s10479-011-1023-z
16. DeNegre, S.: Interdiction and discrete bilevel linear programming. Ph.D. thesis. Lehigh University (2011)
17. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). https://doi.org/10.1007/s101070100263
18. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**(3), 307–339 (1986). https://doi.org/10.1007/BF02592064
19. Edmunds, T.A., Bard, J.F.: Algorithms for nonlinear bilevel mathematical programs. IEEE Trans. Syst. Man Cybern. **21**(1), 83–89 (1991). https://doi.org/10.1109/21.101139
20. J. Egerer, V. Grimm, T. Kleinert, M. Schmidt, G. Zöttl. The Impact of Neighboring Markets on Renewable Locations, Transmission Expansion, and Generation Investment. Eur. J. Oper. Res. (2020). https://doi.org/10.1016/j.ejor.2020.10.055
21. Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M.: A new general-purpose algorithm for mixed-integer bilevel linear programs. Oper. Res. **65**(6), 1615–1637 (2017). https://doi.org/10.1287/opre.2017.1650
22. Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M.: Interdiction games and monotonicity, with application to knapsack problems. INFORMS J. Comput. **31**(2), 390–410 (2019). https://doi.org/10.1287/ijoc.2018.0831
23. Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M.: On the use of intersection cuts for bilevel optimization. Math. Program. **172**(1-2), 77–103 (2018). https://doi.org/10.1007/s10107-017-1189-5
24. Fischetti, M., Monaci, M., Sinnl, M.: A dynamic reformulation heuristic for generalized interdiction problems. Eur. J. Oper. Res. **267**(1), 40–51 (2018). https://doi.org/10.1016/j.ejor.2017.11.043

25. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Program. **66**(1), 327–349 (1994). https://doi.org/10.1007/BF01581153
26. Fortuny-Amat, J., McCarl, B.: A representation and economic interpretation of a two-level programming problem. J. Oper. Res. Soc. **32**(9), 783–792 (1981). https://doi.org/10.1057/jors.1981.156
27. Garcés, L.P., Conejo, A.J., García-Bertrand, R., Romero, R.: A bilevel approach to transmission expansion planning within a market environment. IEEE Trans. Power Syst. **24**(3), 1513–1522 (2009). https://doi.org/10.1109/TPWRS.2009.2021230
28. Garcia-Herreros, P., Zhang, L., Misra, P., Arslan, E., Mehta, S., Grossmann, I.E.: Mixed-integer bilevel optimization for capacity planning with rational markets. Comput. Chem. Eng. **86**, 33–47 (2016). https://doi.org/10.1016/j.compchemeng.2015.12.007
29. Grimm, V., Grübel, J., Schewe, L., Schmidt, M., Zöttl, G.: Nonconvex equilibrium models for gas market analysis: failure of standard techniques and alternative modeling approaches. Eur. J. Oper. Res. **273**(3), 1097–1108 (2019). https://doi.org/10.1016/j.ejor.2018.09.016
30. Grimm, V., Orlinskaya, G., Schewe, L., Schmidt, M., Zöttl, G.: Optimal Design of Retailer-Prosumer Electricity Tariffs Using Bilevel Optimization. In: Omega (2020). https://doi.org/10.1016/j.omega.2020.102327
31. Grimm, V., Schewe, L., Schmidt, M., Zöttl, G (2018) A multilevel model of the European entry-exit gas market. Math. Methods Oper. Res. https://doi.org/10.1007/s00186-018-0647-z
32. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Stat. Comput. **13**(5), 1194–1217 (1992). https://doi.org/10.1137/0913069
33. He, X., Li, C., Huang, T., Li, C.: Neural network for solving convex quadratic bilevel programming problems. Neural Netw. **51**, 17–25 (2014). https://doi.org/10.1016/j.neunet.2013.11.015
34. Hu, X., Ralph, D.: Using EPECs to model bilevel games in restructured electricity markets with locational prices. Oper. Res. **55**(5), 809–827 (2007). https://doi.org/10.1287/opre.1070.0431
35. Jeroslow, R.G.: The polynomial hierarchy and a simple model for competitive analysis. Math. Program. **32**(2), 146–164 (1985). https://doi.org/10.1007/BF01586088
36. Kelley Jr., J.E.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**(4), 703–712 (1960). https://doi.org/10.1137/0108053
37. Kleinert, T., Labbé, M., Plein, F., Schmidt, M.: There's no free lunch: on the hardness of choosing a correct Big-M in bilevel optimization. Oper. Res. (2020). https://doi.org/10.1287/opre.2019.1944
38. Kleinert, T., Schmidt, M.: Computing feasible points of bilevel problems with a penalty alternating direction method. INFORMS J. Comput. (2020). https://doi.org/10.1287/ijoc.2019.0945
39. Kleinert, T., Schmidt, M.: Global optimization of multilevel electricity market models including network design and graph partitioning. Discrete Optim. **33**, 43–69 (2019). https://doi.org/10.1016/j.disopt.2019.02.002
40. Labbé, M., Marcotte, P., Savard, G.: A bilevel model of taxation and its application to optimal highway pricing. Manag. Sci. **44**(12), 1608–1622 (1998). https://doi.org/10.1287/mnsc.44.12.1608
41. Lozano, L., Smith, J.C.: A value-function-based exact approach for the bilevel mixed-integer programming problem. Oper. Res. **65**(3), 768–786 (2017). https://doi.org/10.1287/opre.2017.1589
42. Lv, Y., Chen, Z., Wan, Z.: A neural network for solving a convex quadratic bilevel programming problem. J. Comput. Appl. Math. **234**(2), 505–511 (2010). https://doi.org/10.1016/j.cam.2009.12.041
43. Moore, J.T., Bard, J.F.: The mixed integer linear bilevel programming problem. Oper. Res. **38**(5), 911–921 (1990). https://doi.org/10.1287/opre.38.5.911
44. Morales, J.M., Pinson, P., Madsen, H.: A transmission-cost-based model to estimate the amount of market-integrable wind resources. IEEE Trans. Power Syst. **27**(2), 1060–1069 (2012). https://doi.org/10.1109/TPWRS.2011.2177281
45. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Berlin (2006). https://doi.org/10.1007/978-0-387-40065-5
46. Pineda, S., Morales, J.M.: Solving linear bilevel problems using Big-Ms: not all that glitters is gold. IEEE Trans. Power Syst. (2019). https://doi.org/10.1109/TPWRS.2019.2892607
47. Quesada, I., Grossmann, I.E.: An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Comput. Chem. Eng. **16**(10–11), 937–947 (1992). https://doi.org/10.1016/0098-1354(92)80028-8
48. Ralphs, T.: Cor@l: Bilevel Optimization Problem Library. http://coral.ise.lehigh.edu/data-sets/bilevel-instances/. Accessed 12 Dec 2019
49. Regionales Rechenzentrum Erlangen. Woodcrest Cluster. https://www.anleitungen.rrze.fau.de/hpc/woody-cluster/. Accessed 12 Dec 2019

50. Tahernejad, S., Ralphs, T.K., DeNegre, S.T.: A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. Math. Program. Comput. **12**, 529–568 (2020). https://doi.org/10.1007/s12532-020-00183-6
51. Tang, Y., Richard, J.-P.P., Smith, J.C.: A class of algorithms for mixedinteger bilevel min-max optimization. J. Global Optim. **66**(2), 225–262 (2015). https://doi.org/10.1007/s10898-015-0274-7
52. Vanderbei, R.J.: Linear Program. (2014). https://doi.org/10.1007/978-1-4614-7630-6
53. Vicente, L., Savard, G., Júdice, J.: Descent approaches for quadratic bilevel programming. J. Optim. Theory Appl. **81**(2), 379–399 (1994). https://doi.org/10.1007/BF02191670
54. Xu, P., Wang, L.: An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. Comput. Oper. Res. **41**, 309–318 (2014). https://doi.org/10.1016/j.cor.2013.07.016
55. Zare, M.H., Borrero, J.S., Zeng, B., Prokopyev, O.A.: A note on linearized reformulations for a class of bilevel linear integer problems. Ann. Oper. Res. **272**(1-2), 99–117 (2019). https://doi.org/10.1007/s10479-017-2694-x

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.